



Universidade do Estado do Rio de Janeiro

Centro de Educação e Humanidades

Faculdade de Comunicação Social

Ivan Mussa Tavares Gomes

**O jogador descentralizado: comunicação humano-videogame no contexto
pós-Minecraft**

Rio de Janeiro

2018

Ivan Mussa Tavares Gomes

**O jogador descentralizado: comunicação humano-videogame no contexto pós-
Minecraft**

Tese apresentada como requisito parcial para obtenção do título de Doutor, ao Programa de Pós-graduação em COMUNICAÇÃO, da Universidade do Estado do Rio de Janeiro. Área de concentração: Comunicação Social.

Orientador: Prof. Dr. Vinicius Andrade Pereira

Rio de Janeiro
2018

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CEH/A

M989 Mussa, Ivan.
 O jogador descentralizado: comunicação humano-videogame no contexto
 pós-Minecraft / Ivan Mussa Tavares Gomes. – 2018.
 212 f.

 Orientador: Vinicius Andrade Pereira.
 Tese (Doutorado) – Universidade do Estado do Rio de Janeiro. Faculdade de
 Comunicação Social.

 1. Comunicação Social – Teses. 2. Videogames – Teses. 3. Jogador – Teses.
 I. Pereira, Vinicius Andrade. II. Universidade do Estado do Rio de Janeiro.
 Faculdade de Comunicação Social. III. Título.

es CDU 316.77

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta
tese, desde que citada a fonte.

Assinatura

Data

Ivan Mussa Tavares Gomes

O jogador descentralizado: comunicação humano-videogame no contexto pós-Minecraft

Tese apresentada como requisito parcial para obtenção do título de Doutor, ao Programa de Pós-graduação em COMUNICAÇÃO, da Universidade do Estado do Rio de Janeiro. Área de concentração: Comunicação Social.

Aprovado em 27 de abril de 2018.

Banca examinadora:

Prof. Dr. Vinícius Andrade Pereira (orientador)
Universidade do Estado do Rio de Janeiro - UERJ

Prof^a. Dr^a. Fátima Cristina Regis Martins de Oliveira
Universidade do Estado do Rio de Janeiro - UERJ

Prof^a Dr^a. Letícia Perani Soares
Universidade Federal de Juiz de Fora - UFJF

Prof. Dr. Thiago Pereira Falcão
Universidade Federal do Maranhão - UFMA

Prof. Dr. Emmanoel Martins Ferreira
Universidade Federal Fluminense - UFF

Rio de Janeiro
2018

DEDICATÓRIA

À memória da minha vó Marly, que me ensinou a brincar.

AGRADECIMENTOS

Agradeço antes de tudo ao meu orientador, Vinicius Andrade Pereira, não apenas pelos comentários e leitura cuidadosos, mas pela amizade e companheirismo dos últimos quatro anos. Este trabalho é fruto dessa colaboração.

I also thank my academic supervisor during my visiting period at the IT University of Copenhagen, Prof. Espen Aarseth, for his insightful comments on my research, but also for the welcoming environment of his research group. For that, I also thank the remaining researchers and students of the Center for Computer Games Research, specially Pawel Grabarczyk, Anders Løvlie, Hans-Joachim Backe, Rune Nielsen, Joleen Blom and Michael Debus.

Agradeço à minha família de acadêmicos: minha mãe, Margarida; meu pai, Hélio e meu irmão, Vítor. Talvez seja possível vislumbrar um pouco de urbanismo, políticas públicas e ciências sociais nesse trabalho. Se sim, são eles os “culpados”.

Aos funcionários e professores do PPGCOM Uerj, o lugar que veio a se tornar minha segunda (por vezes, primeira) casa no Rio de Janeiro nos últimos seis anos. Agradeço a Amanda Neves, Luzinete Cardozo e particularmente à minha heroína, Eliana Siciliano.

Agradeço também à Profa. Denise Siqueira, ao Prof. Ricardo Freitas e à Profa. Cíntia Sanmartin Fernandes, cujos esforços foram fundamentais – em diferentes momentos – para que eu e outros colegas de turma desfrutássemos da oportunidade única de fazer o doutorado sanduíche.

Aproveito para agradecer à Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior (CAPES), por ter concedido tanto a bolsa de doutorado quanto a de doutorado sanduíche, que financiaram essa pesquisa.

Devo agradecimentos especiais aos professores da linha de pesquisa em Tecnologias da Comunicação e Cultura da Uerj. Em específico, agradeço ao Prof. Márcio Gonçalves e à Profa. Fátima Régis pelo comando das disciplinas de metodologia de pesquisa e seminário de tese, respectivamente. Ambas foram essenciais para que eu chegasse ao final da pesquisa com o resultado que desejava.

Agradeço com carinho especial aos meus colegas do PPGCOM Uerj (e agregados). Em especial, Rafael Barbosa, Filipe Feijó, Ana Camila García, Zé Messias, Alessandra Maia, Yuri Garcia, Ramon Bezerra, João Guilherme, Camila Augusta, Tetê Mattos, Gustavo Audi e tantos outros. Que nossa pesquisa conjunta continue acontecendo nos bares.

Aos amigos-mestres-companheiros pesquisadores dos *game studies* no Brasil. Letícia Perani, Mariana Amaro, Emmanoel Ferreira, Thiago Falcão, Daniel Marques, Luiz Adolfo Andrade, entre muitos outros – e que venham muitos mais.

Aos meus companheiros farofeiros de Copenhague: Bia Provasi, Luiz Adolfo (novamente), Nina, Sally, Kemilla, Tiago, Isabela, Elver, Rafael e Victor. I also thank my good friends, Rainy James and my roommate Oscar, for making me feel welcome in this beautiful city.

Special thanks for Mark Johnson for giving me the opportunity of working as a beta tester in his game, *Ultima Ratio Regum*.

Um agradecimento de apoio e admiração a Alexandra Elbakyan, fundadora do SciHub, bem como aos mantenedores do Libgen.org. É um dos maiores absurdos imagináveis que ainda existam forças interessadas em sequestrar conhecimento e cobrar resgate em troca de acesso. Minhas saudações a qualquer pessoa que combata esses parasitas. Todos os artigos desbloqueados pelo SciHub usados nessa pesquisa estão nas referências com seus endereços DOI. Basta colá-los no site para ter acesso grátis. Ver: <https://www.reddit.com/r/scihub/>.

Uma nota de agradecimento pessoal aos game designers que pensam por meio de seus jogos e que inspiram em grande parte este trabalho: Brian Moriarty, Jonathan Blow, Tarn Adams, Stephen Lavelle (thanks for the coin!), Derek Yu, e à comunidade persistente e insistente de desenvolvedores *roguelikes*.

E, finalmente, agradeço a Flávia, por ser uma fonte inesgotável de energia e bons sentimentos.

We do not stand in the shoulders of giants. But if you are lucky, like me, you may one day experience the sublime humiliation of standing in the shadow of the Moon.

Brian Moriarty

RESUMO

MUSSA, Ivan. *O jogador descentralizado: comunicação humano-videogame no contexto pós-Minecraft*. 2018. 212 f. Tese (Doutorado em Comunicação) – Faculdade de Comunicação Social, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2018.

Este trabalho objetiva investigar um aspecto observado no jogo *Minecraft*. Sua disseminação em massa ocorre apesar de um traço divergente da camada mais alta da indústria dos videogames: o jogo não faz qualquer esforço para trazer o jogador ao centro de seu funcionamento. Chamamos esse atributo de descentralização do jogador, e o definimos como objeto de estudo, estabelecendo o objetivo de descrever de que modo funciona. Sob o prisma do contexto pós-*Minecraft*, separamos esse objetivo em três capítulos: o primeiro almeja definir um escopo e diretriz empírica, caracterizando a descentralização sob a ótica das dinâmicas sensoriais que se desenham na comunicação entre mundo de jogo (conjunto de estímulos manifestados pelo videogame) e os sentidos humanos. O segundo capítulo faz uso dessas definições de modo a investigar a história da descentralização nos jogos que influenciaram e foram influenciados pela emergência de *Minecraft*. Finalmente, no terceiro capítulo, selecionamos dois jogos que levam a descentralização a seu extremo, de modo a vasculhar seus mundos de jogo para investigar como se posiciona o jogador em meio a sua dinâmica.

Palavras-chave: Videogames. Comunicação. Jogador.

ABSTRACT

MUSSA, Ivan. *The decentralized player: human-game communication in the post-Minecraft context*. 2018. 212 f. Tese (Doutorado em Comunicação) – Faculdade de Comunicação Social, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2018.

This work aims to investigate an aspect observed in the game *Minecraft*. Its mass dissemination occurs despite a feature that diverges from the practices of the highest layer of the video game industry: the game makes no effort to bring the player to the center of its functioning. We call this attribute decentralization of the player, and define it as our object of research, setting the goal of describing how it works. Through the prism of the post-*Minecraft* context, we separate this objective into three chapters: the first aims to define an empirical scope and guideline, characterizing the decentralization from the point of view of the sensorial dynamics that unfold from the communication between the gameworld (a set of stimuli displayed by the game) and the human senses. The second chapter makes use of these definitions to investigate the history of the decentralization in the games that influenced and were influenced by the emergence of *Minecraft*. Finally, in the third chapter, we select two games that lead to extreme decentralization, aiming to sift through their game worlds to investigate how the player positions himself amidst their dynamics.

Keywords: Video games. Communication. Player.

LISTA DE FIGURAS

Figura 1 - Pac-Man (1980)	17
Figura 2 - Estado do mundo do jogo de Pac-Man após algumas mudanças causadas pelo processo de jogo.	18
Figura 3 - Representação do processo de colisão no jogo Braid (2008)	18
Figura 4 - Propriedades observáveis na porção abaixo	20
Figura 5 - Jogadores de Minecraft.....	27
Figura 6 - Modelo de Jogo Michael Nitsche	28
Figura 7 - Uma caverna em Minecraft	32
Figura 8 - A associação entre dispositivos de input e o lócus de manipulação é mediada por um método de input.....	37
Figura 9 - Super Mario Bros e Minecraft	39
Figura 10 - Visualização de Minecraft em primeira pessoa (esquerda) e terceira pessoa (direita).	47
Figura 11 - Cenário de Minecraft sob uma perspectiva aérea.	48
Figura 12 - Modelo extensivo da perspectiva.....	53
Figura 13 - Modelo intensivo da perspectiva e de sua interseção com o jogador e o ambiente.....	54
Figura 14 - Mapas.....	55
Figura 15 - A divisão do mundo de jogo segundo as habilidades do jogador em Super Mario Bros.	57
Figura 16 - Casa construída sobre a água em Minecraft.	60
Figura 17 - Uma das fases do jogo 1001 Spikes, pontuada com anotações do game designer Jonathan Blow	64
Figura 18 - Portal - 2	65
Figura 19 - Mapas de Minecraft gerados segundo seus algoritmos, na versão 1.8.	68
Figura 20 - Mundo de SimCity: perspectiva onipresente do ambiente dinâmico	70
Figura 21 - Maniac Mansion	73
Figura 22 - The Clolonel's Bequest:	75
Figura 23 - Lista dos 10 melhores tempos de Super Mario Bros. (visualizada em 26/11/2017).....	79
Figura 24 - Cutscene de Super Mario Bros. que pode ser pulada, caso o jogador provoque um glitch no jogo.	80

Figura 25 - “Descobrimto” da velocidade superior por parte dos jogadores.....	82
Figura 26 - Três das técnicas mais simples de otimização de mineração em minecraft. ..	84
Figura 27 - Inifniminer (2009): o espaço tridimensional dividido em blocos se assemelha ao de Minecraft.	86
Figura 28 - Dwarf Fortress (2006):	87
Figura 29 - À esquerda, GTA III, e em destaque o minimapa. À direita, mini-mapa de GTA IV (2006), com ícones de orientação destacados.....	91
Figura 30 - Sistema de “waypoints” nas três principais franquias de mundo aberto da Ubisoft:	92
Figura 31 - Elementos da HUD de Don’t Starve e sua relação com o restante do mundo de jogo.	109
Figura 32 - Composição possível (mas não única) de uma base.	110
Figura 33 - Subnautica: jogador explora o oceano.	113
Figura 34 - Mapa de Subnautica.....	114
Figura 35 - Perspectiva aérea do mapa de Factorio.....	117
Figura 36 - <i>Factorio</i>	119
Figura 37 - Rogue (1980)	123
Figura 38 - Fotografias de flocos de neve – Wilson Bentley	125
Figura 39 - Simulação computacional da formação de flocos de neve, executadas no software Visions of Chaos.	126
Figura 40 - Dungeon (1979).....	127
Figura 41 - Beneath Apple Manor (1978)	128
Figura 42 - Dragon Maze (1977-8)	129
Figura 43 - Seis permutações labirínticas de Dragon Maze	131
Figura 44 - Um dos níveis de Rogue, ainda parcialmente inexplorado. O jogador (@) enfrenta uma das criaturas do jogo, um “kobold” (K).....	132
Figura 45 - À esquerda, Moria (1983) e à direita Hack (1985), descendentes diretos de Rogue.	133
Figura 46 - Angband (1990)	134
Figura 47 - Zangband (1994):.....	135
Figura 48 - Tales of Maj’Eyal (2013).....	136
Figura 49 - Spelunky (2009):	138
Figura 50 - Crypt of the Necrodancer (2013): o jogo mistura conceitos de jogos musicais com o funcionamento tradicional dos roguelikes.	140

Figura 51 - Crypt of the Necrodancer: parte da sequência de ações que compõe a dança Red Dragon Dance.....	141
Figura 52 - À esquerda, Brogue (2009); à direita, Unexplored (2017)	141
Figura 53 - NetHack (1987)	144
Figura 54 - UnReal World (1992):	147
Figura 55 - Cataclysm: Dark Days Ahead.....	148
Figura 56 - Caves of Qud (2011):.....	149
Figura 57 - Go (esquerda) e xadrez (direita).	154
Figura 58 - Go:	156
Figura 59 - Mapa territorial de um dos mundos gerados por Ultima Ratio Regum	160
Figura 60 - Dois quadros exibindo as características de duas civilizações do mundo em questão.	161
Figura 61 - Mapas de três das 17 capitais do mundo gerado para a pesquisa.	163
Figura 62 - À esquerda, uma favela circundada por árvores; à direita, uma fortaleza nômade.....	164
Figura 63 - Distrito central de Whitemound.....	164
Figura 64 - À direita, jogador diante da catedral de Whitemound.	165
Figura 65 - Ídolos religiosos da seita Teachers of the Seven Metals, compostos por pequenas runas e símbolos.....	166
Figura 66 - Caso encontremos um personagem de URR, podemos examinar sua vestimenta.	166
Figura 67 - Tela de conversação entre o jogador (direita) e uma sacerdotisa encontrada na catedral (esquerda).	168
Figura 68 - Ilasil, “The Absolute Planes”, no ano 250 de sua história.	169
Figura 69 - Porção limitada do mapa exibindo a configuração hidrográfica de Ilasil.....	171
Figura 70 - Mapa de biomas do mundo de Ilasil:	172
Figura 71 - Distribuições de assentamentos e civilizações de Ilasil	173
Figura 72 - Parte da cidade de Tempestboot, cidade élfica onde nosso personagem iniciou o jogo.	175
Figura 73 - Personagens e nosso avatar circundam uma árvore que serve como uma das tavernas da cidade de Tempestboot.	176
Figura 74 - Humanos declaram Guerra contra elfos (seta azul).....	177
Figura 75 - Relações comerciais de Tempestboot.	178

SUMÁRIO

	INTRODUÇÃO.....	14
1	DESCENTRALIZAÇÃO DO JOGADOR: A EMERGÊNCIA DO MUNDO DE JOGO EM <i>MINECRAFT</i>	27
1.1	Percepção e ação: a autossuficiência dos mundos de jogo em relação à ficção e à representação	28
1.2	Lócus de manipulação: A autossuficiência dos mundos de jogo em relação aos dispositivos de input e output	36
1.3	Perspectiva e ambiente: a dimensão espacial da descentralização do jogador	45
1.4	Vagueamento e Dinâmica(s): a dimensão temporal da conexão jogador-jogo	50
1.5	Conectividade: o jogador implicado no ambiente do jogo	59
1.5.1	<u>Limitação da conectividade como mecanismo de centralização do jogador</u>	65
1.6	Instabilidade: as escalas de incerteza dos videogames	70
1.6.1	<u>Instabilidade como fundamento dos mundos de jogo: <i>speedrunning</i> e <i>spelunking</i></u>	77
2	MUNDOS ERRANTES: CONECTIVIDADE E INSTABILIDADE NOS SURVIVAL GAMES, <i>ROGUELIKES</i> E <i>ROGUELITES</i>	86
2.1	Exploração extensiva: a abertura dos mundos de jogo	91
2.2	Adaptação e espaço intensivo: jogos de sobrevivência e a metamorfose do <i>game loop</i>	98
2.2.1	<u>Don't Starve: a dinâmica básica da adaptação</u>	103
2.2.2	<u>Subnautica: biomas em tensão dinâmica</u>	111
2.2.3	<u>A topologia de Factorio: ciclos progressivos de adaptação</u>	116
2.3	Origens errantes: conectividade e instabilidade na camada subalterna dos videogames	122
2.3.1	<u>Cristalização: intensificação da conectividade na gênese roguelike</u>	125
2.3.2	<u>Transformações topológicas: a emergência dos roguelites</u>	136
2.3.3	<u>Regimes de instabilidade em <i>Hack</i>, <i>NetHack</i> e seus herdeiros</u>	143
3	ULTRADINÂMICAS: UM MERGULHO EM <i>ULTIMA RATIO REGUM</i> E <i>DWARF FORTRESS</i>	152

3.1	Esquema dinâmico: dois tipos de regras e suas reverberações sensoriais....	153
3.2	Ultraconectividade em <i>Ultima Ratio Regum</i>	159
3.3	Ultrainstabilidade em <i>Dwarf Fortress</i>	169
	CONSIDERAÇÕES FINAIS	182
	REFERÊNCIAS.....	188
	APÊNDICE - Ludografia comentada	195

INTRODUÇÃO

Uma pessoa joga o videogame *Minecraft* (2009): quais processos irradiam-se desse encontro? Para apenas arranhar a superfície dessa aglomeração de camadas, basta uma rápida revisão bibliográfica. Encontraremos desde análises de usabilidade e programação, passando por questões de gênero, até aplicações educacionais¹. A multiplicidade que aflora do contato lúdico entre ser humano e computador não parece conhecer fronteiras: mistura-se aos campos político, social, econômico e científico.

Esta tese tem como objetivo abordar um dos processos responsáveis por germinar essa propagação: a comunicação entre o aparato sensorial do indivíduo que joga e os estímulos expressáveis pelo videogame. As características desse processo comunicativo diferem de jogo para jogo, o que nos leva a definir um de seus aspectos como objeto de pesquisa: a *descentralização dos jogadores*.

O que significa dizer que um jogador é descentralizado pelo videogame que ele próprio opera? Antes de responder, cabe defender um escopo empírico que servirá de base para evidenciar tanto o funcionamento dessa descentralização quanto a sua relevância como objeto de pesquisa. Não por acaso, escolhemos falar de *Minecraft*, já que é este jogo que sintetiza não só os critérios de seleção para o corpus do estudo, como também justifica a necessidade dele. Como pode um jogo produzido sem qualquer financiamento, durante as pausas para o almoço de um programador solitário, tornar-se o segundo título mais vendido da história dos videogames, atrás apenas de *Tetris* (1984)? Como explicar seu impacto global, que se alastrou para além dos videogames, sendo utilizado amplamente como ferramenta educacional e assistido no YouTube diariamente por pessoas que nem mesmo possuem o hábito de jogá-lo?

As estranhezas não param por aí: além de ter sido fruto de quase nenhum investimento, *Minecraft* chega ao topo lançando mão de uma estratégia de game design avessa ao que prega a cartilha da grande indústria. Muito já se falou, por exemplo, de como opera o conglomerado dominado por empresas como Electronic Arts e Ubisoft: busca-se vender jogos não pelo que fazem, mas por seus atributos visuais². No limite, a questão vai bem além disso: o custoso investimento necessário para entrar na corrida rumo à perfeição gráfica implica um rígido comprometimento financeiro. É necessário controlar cada aspecto da experiência dos

¹ Fonte: <https://education.minecraft.net/>.

² Essa foi a mesma conclusão a que chegou Chris Crawford, um dos criadores da GDC (*Game Developers Conference*), já manifestada em seu artigo *Process Intensity* (1989): os dados exibidos pelo jogo (imagens, áudio, vídeo, texto, entre outros tipos) deveriam, para o autor ser lidos em conjunção com os processos de que participavam (um processo seria, para Crawford, a transformação cont[ínua dos dados]).

jogadores, nivelar o campo de possibilidades do jogo de modo que este se reduza àquilo que, supõe-se, apetece o público alvo. Assim, garante-se o lucro por meio de uma cartada dupla: facilitar a previsão dos resultados do consumo e demarcar um campo de possibilidades mais estreito que, portanto, é mais simples de ser polido visualmente.

Minecraft subverte as expectativas de quem crê na imperatividade do modelo acima, pois não só ignora qualquer pretensão de perfeição gráfica como faz o possível para que a experiência de seus milhões de jogadores sejam as mais diversas possíveis. Principalmente em sua incipiência, quando se espalha como uma epidemia, não se preocupava nem mesmo em apresentar um tutorial que introduzisse as habilidades básicas necessárias para que o jogador se familiarizasse com suas regras. Pelo contrário: esses eram atirados em um mundo hostil sem qualquer preparação, do qual deveriam extrair recursos, construir ferramentas e abrigo – enfim: sobreviver. Dito de outro modo, o mundo de *Minecraft* não gira em torno de quem joga, e é isso, basicamente, o que chamamos de “descentralização do jogador”.

O objetivo deste trabalho é, em termos preliminares, compreender de que modo se dá essa descentralização. É claro, tal objetivo carece de um polimento teórico-epistemológico que defina com lucidez o escopo e material empírico que guiará essa análise. Primeiro, portanto, delinearemos nosso objeto de pesquisa – que fatores exatamente abordaremos em *Minecraft*? Seu código? Seus personagens? Seus jogadores? Em seguida, será necessário demonstrar o arcabouço epistemológico que proverá recursos descritivos aplicáveis a estes fatores.

Respondendo a primeira pergunta: esta investigação recorta seu objeto de análise nas manifestações visuais, auditivas e cinéticas dos videogames, cujas interpenetrações serão chamadas aqui de *mundos de jogo*. Este termo, tal qual forjado por Kristine Jørgensen (2013), designa uma espécie de interface, um fluxo de imagens e sons em mutação que funciona como metáfora para os circuitos internos do hardware e as camadas de abstração do software.

Evidentemente, o funcionamento do *sistema interno* (uma continuidade que se estende pelo menos das correntes elétricas até códigos programados) do jogo é determinante para seu funcionamento. No entanto, ele se dispõe ao acesso humano apenas indiretamente. O que de fato abre os estados deste sistema à alteração por parte de um jogador é a sua manifestação sensível em uma superfície de estímulos moventes, que configuram um mundo de jogo.

Sendo assim, existe um contato *direto* entre os mundos de jogo e o que chamamos de “jogador”. Este último termo, tal qual comumente empregado, é um ponto de convergência de linhas de investigação nem sempre compatíveis: como o jogador imagina um mundo de jogo e como imagina a si mesmo nesse mundo? Como compartilha essas representações e identidades entre seus pares? O recorte desta pesquisa abandona qualquer preocupação com estas questões,

circunscrevendo o termo “jogador” à função de referir-se a um *conjunto de capacidades de ação, investíveis no contato com um mundo de jogo*.

Ao jogar um videogame como *Minecraft*, o *aparelho sensório-motor* humano canaliza suas capacidades de modo a atuar sobre um conjunto de *dispositivos de input*: controles, teclados, sensores de movimento, entre outros. Estas capacidades sensíveis, naturalmente, envolvem não só as disposições táteis e manipuláveis dos controles, como também a atenção que o jogador investe no jogo: vê-se e ouve-se para controlar, e vice-versa. Uma vez que esta acoplagem entra em circuito – o que requer adaptações contínuas – constitui-se uma miríade de propriedades e capacidades novas e mutáveis, dotadas da potência de intervir no mundo de jogo: constitui-se um *jogador*.

Da mesma forma que só nos preocuparemos com o sistema interno do jogo na medida em que se prolonga a uma superfície de estímulos sensoriais, a complexa conexão do corpo biológico humano à máquina que roda o videogame também será abordada apenas no que tange sua respectiva atualização sensível nos mundos de jogo. Portanto, encaramos esses mundos como zonas limítrofes, onde são detectáveis tanto traços maquínicos como humanos: um lócus onde a *relação* entre os polos jogo-jogador desenvolve-se e deixa seus traços.

Apesar disso, não há como negar o fato de que os aspectos desta relação são, em alguma medida, difíceis de serem materialmente tratados: como enxergar essas associações “invisíveis”? De que modo é possível dizer que, de um lado, os algoritmos do sistema e, de outro, as capacidades de ação humanas são observáveis nas transformações audiovisuais da tela? Para responder essa pergunta, seguimos para a segunda parte dessa introdução: a apresentação das ferramentas que compõem nosso arcabouço epistemológico, cuja finalidade é dar um caráter palpável a essas associações “invisíveis”.

Na composição dinâmica *visuaudiomotora* (PEREIRA, 2008) de um videogame, cores, formas, e mutações visíveis e audíveis são forças certamente relevantes: é por meio delas que se estabelece o contato mais fundamental entre ser humano e jogo. No entanto, existem forças invisíveis que *processam* o que antes era estável e geram a possibilidade de mudança. Estamos lidando, de modo mais simples, com as *propriedades, capacidades e tendências* tanto do jogador quanto do mundo de jogo. Tais grandezas não são diretamente perceptíveis, mas há modos de intuí-las a partir da observação do sistema em continuidade.

Quando vamos além da imagem e do som e lemos, também, suas respectivas transformações, as propriedades e capacidades dos elementos dos mundos de jogo revelam-se gradativamente. Passamos a perceber os modos pelos quais um conjunto de pixels vizinhos oscila para criar um “personagem” que se locomove e que, por sua vez, pode alcançar um

segundo conjunto de pixels (mais um personagem) e fazê-lo desaparecer. Trata-se de uma capacidade: a de eliminar o adversário de acordo com o ângulo de contato, ou *colisão*, entre os grupos de pixels. Eis uma rápida ilustração:

Figura 1 - Pac-Man (1980)



A imagem acima apresenta um único quadro do jogo *Pac-Man*. Observando seus componentes visuais estáticos, podemos prontamente detectar uma série de aspectos de seu mundo de jogo. Quatro fantasmas coloridos se posicionam no centro da tela, enquanto um círculo amarelo repousa alguns pixels abaixo. Pequenos pontos se enfileiram ao longo de corredores bifurcantes, intercalados por pontos maiores. Estes caminhos são definidos por bordas e paredes azuladas, que dão forma a um labirinto. Um letreiro central (“*Ready!*”) alerta o jogador. Para além dos limites do labirinto, encontram-se dois ícones amarelos e um em forma de fruta na faixa inferior; além de dois indicadores textuais e numéricos na superior.

O mundo do jogo de *Pac-Man* não se encerra com esta descrição. Quando o jogo começa, parte dos pixels que formam as imagens passam a oscilar, simulando o movimento de algumas dessas entidades. A observação dos deslocamentos na tela permite a detecção de mais propriedades do mundo do jogo: os quatro fantasmas centrais desatam a se locomover pelos corredores, enquanto o jogador pode controlar o círculo amarelo. Se tentar percorrer o caminho trilhado pelos pontos, o jogador notará que estes desaparecerão à medida em que o círculo amarelo os “devora”. Se alcançar um dos círculos maiores, os quatro fantasmas mudam de cor e seu padrão de movimento torna-se temporariamente mais lento e previsível, possibilitando que o jogador também os “devore”. Caso se choque com um dos fantasmas sem ativar essa função, o jogador “morre”, o que faz sumir um dos ícones indicados na secção inferior da tela.

Figura 2 - Estado do mundo do jogo de Pac-Man após algumas mudanças causadas pelo processo de jogo.



Grande parte dos pontos brancos foi devorada; os fantasmas estão “atorreados” após o jogador comer um círculo grande. Os indicadores superiores (pontuação) e inferiores (frutas) também mudaram.

Um exame ainda mais cuidadoso das mudanças na tela pode fornecer outras muitas informações sobre as propriedades e capacidades presentes no mundo do jogo. O jogador mais atento perceberá que o fantasma vermelho é o mais agressivo, deslocando-se diretamente na direção do personagem amarelo, independentemente da situação geral do sistema. O fantasma rosa, por outro lado, tenta flanquear sua vítima, circundando o jogador por meio de rotas mais longas. A ação individual de cada fantasma orienta as ações do jogador que, por sua vez, influenciará de maneira particular o comportamento de cada fantasma.

Capacidades nascem do encontro de duas propriedades: a *ofensividade* (contida no personagem 1) e a *vulnerabilidade* (contida no personagem 2). Afirmar que um personagem tem a “propriedade” de ser vulnerável soa, a princípio, absurdo, já que equiparamos uma consequência positiva a uma negativa. No entanto, a ideia, neste momento, é abrir mão de julgar o valor teleológico dos resultados das interações expressas na tela. O ponto-chave é que duas ou mais entidades do mundo de jogo empregam suas propriedades para processar diferentes capacidades.

Figura 3 - Representação do processo de colisão no jogo Braid (2008)



Quando observamos uma dada distribuição de propriedades em um cenário jogável, é possível intuir parte das capacidades ali presentes³. No jogo *Super Mario Bros.* (1985), logo se percebe que certas superfícies são capazes de oferecer tanto sustentação – pode-se repousar o personagem ou fazê-lo caminhar sobre elas – quanto interrupção do movimento (não se pode atravessá-las livremente da mesma forma que uma porção vazia do espaço⁴). Monstros também são capazes de colidir com o jogador, por exemplo, em diferentes ângulos (alguns fatais para os monstros, outros para o jogador), itens são coletáveis, reconfigurando as capacidades do jogador – como o cogumelo que aumenta seu tamanho (propriedade), permitindo que destrua certos blocos (capacidade).

³ Como será visto adiante, os jogos possuem a liberdade de quebrar com a consistência do seu campo de possibilidades: uma parede que usualmente não possui a capacidade de ser destruída por uma explosão pode, em um ponto aleatório de sua extensão, guardar essa capacidade secretamente, como no jogo *The Legend of Zelda* (1989). Em uma de suas sequências, *The Legend of Zelda: A Link to the Past* (1991) exhibe uma rachadura nas superfícies que cedem à força das explosões. Tal relação é quebrada intencionalmente no jogo de 1989 e “corrigida” no jogo de 1991. A mesma consistência, porém, pode ser quebrada também por glitches, bugs, entre intervenções na estabilidade do mundo de jogo, como será visto no capítulo 1.

⁴ A própria percepção de que uma porção do espaço está vazia nasce, no mesmo jogo, da continuidade do movimento de personagens: quando esta continuidade é interrompida, há três razões possíveis. A primeira, é que há um objeto obstruindo o movimento. A segunda é que foi encontrado um limite para o espaço jogável (como o limite da tela, por exemplo). A última possibilidade é que ocorreu um glitch/bug no sistema. A primeira, por exemplo, é relativamente mais comum e mais fácil de ser diagnosticada, já que os objetos sólidos em *Super Mario Bros.* possuem qualidades visuais (cores, formas, etc.) discerníveis do fundo vazio (que permanece imóvel e relativamente homogêneo).

Figura 4 - Propriedades observáveis na porção abaixo



Solidez (do chão, plataformas e blocos), ofensividade e vulnerabilidade (do jogador e do monstro, dependendo do ângulo de colisão), destrutibilidade (dos blocos escuros) e indestrutibilidade (dos blocos claros). As capacidades são bem mais numerosas, já que envolvem a combinação destas e outras propriedades entre si.

Uma vez rastreadas, as propriedades do jogador e do mundo de jogo fornecem ferramentas para a compreensão do processo de comunicação entre ambos. No entanto, não basta listá-las: é preciso definir o modo como se interrelacionam. Certas propriedades distribuem-se *geometricamente*, são quantificáveis. O número de pixels utilizado pelo mundo de jogo é a primeira destas propriedades quantificáveis, que é acumulada justamente para gerar outras: caminhos longos e curtos, personagens altos ou baixos, etc. As *extensões* destes elementos do mundo de jogo são *reduzíveis às suas respectivas justaposições de pixels*, que são elementos discretos, contáveis e divisíveis. Dito de outro modo, a exibição de uma linha unidimensional – ou mesmo de uma figura bidimensional – não adiciona à superfície de pixels da tela nada que ela já não possuísse de antemão: apenas revela as propriedades extensivas contidas em sua malha bidimensional.

Outras propriedades, no entanto, são fruto de processos que *não estão contidos nos pixels, mas sim surgem da relação entre eles*. Em outras palavras, elas *emergem* das diferenças *qualitativas* expressadas por conjuntos de pixels e sua transformação coordenada. A propriedade de solidez dos blocos, por exemplo, atualiza-se quando um personagem ou item descansa sobre eles e não sofre queda livre. A ofensividade de um monstro, similarmente, só se atualiza se este atingir algo ou alguém sob um determinado ângulo. Tais eventos não estão contidos previamente na justaposição de pixels que os manifestam visualmente e, portanto, não são reduzíveis a eles.

Essa diferença entre *propriedades extensivas e intensivas* (DELANDA, 2002; p. 54-93) permite que elaboremos um método para atribuir *qualidades* aos processos e objetos perceptíveis nos mundos de jogo. Como propriedades *extensivas* são geometricamente mensuráveis, expressam apenas as diferenças de grau no sistema. No entanto, tais extensões relacionam-se a propriedades intensivas: densidade, variabilidade, velocidade, aceleração, entre outras, são produto das diferenças qualitativas do sistema.

O ponto ainda mais fundamental deste raciocínio é que estas qualidades e intensidades são inseparáveis das capacidades e propriedades perceptivas do ser humano que joga. Afinal de contas, como é possível que apontemos movimento, velocidade, massa, aceleração e outras grandezas “físicas” onde há apenas pixels e vibração sonora? No hardware e no software, essas propriedades existem apenas em estado potencial, e só se atualizam quanto operam em comunhão com nossos sentidos. As sensações resultantes constroem-se por meio de fenômenos análogos ao processo que possibilita a sensação de movimento causada pelos quadros estáticos do cinema quando exibidos em continuidade. Vemos um filme como movimento graças a uma acoplagem entre a nossa percepção visual e a alta velocidade de exibição de quadros estáticos⁵. Graças ao mesmo efeito, temos a “ilusão” de ver um personagem se movimentar, correr, acelerar e pular.

Mais: “sentimos” seu peso e inércia quando apertamos um botão para interromper este mesmo movimento, o que dura um determinado lapso de tempo, já que um corpo em movimento tende a continuar em movimento. A percepção dessa desaceleração na tela não causa sensações apenas visuais, mas associa-se sinestesticamente à nossa capacidade de manipulação tátil, dando-nos a sensação de inércia quando nos esforçamos para interromper a corrida de nosso avatar. Tal inércia, bem como outras propriedades intensivas dos mundos de jogo, não estão nem nos pixels que a exhibe nem na nossa percepção pura, mas no encontro entre ambas.

Justamente por isso, são propriedades que só fazem sentido quando observamos o sistema em movimento e/ou mutação – ou ao menos quando conjecturamos suas futuras transformações. Sob este aspecto, a análise geométrica de um mundo de jogo é improdutiva se não vier acompanhada de uma análise *topológica*, ou seja, uma apreciação destes mundos que

⁵ Embora comumente atribuída a um ou alguns fenômenos, como “persistência retiniana” ou “movimento beta”, a percepção de movimento na tela pressupõe um conjunto de operações perceptivas composta por “séries complexas de transformações sistemáticas, que evidenciam múltiplos níveis de codificação” (ANDERSON; ANDERSON, 1988, p. 93). Livre tradução de: “complex series of systematic transformations and evidences multiple levels of coding”.

envolva não só o modo como seus elementos se conectam espacialmente, mas como se comportam ao desenvolverem-se no tempo.

Na topologia de jogos de plataforma, como *Super Mario Bros.*, por exemplo, a interpenetração de propriedades intensivas e as transformações que estas provocam no desenrolar temporal do jogo geram certas tensões. Uma série homogênea de blocos sólidos por onde nossa personagem caminha – o solo – oferece pouquíssima tensão. Mas se um espaço vazio – um buraco – interrompe essa continuidade, uma diferença qualitativa tensiona o personagem a pular, gerando uma *tendência*. Essa tendência muito simples quando agrupada a outras várias, constitui o repertório de estímulos do jogo. Em *Super Mario Bros.*, estas tensões são orquestradas cuidadosamente de modo a colocar o jogador em situações que se complexificam progressivamente.

As tensões decorrentes das diferenças qualitativas entre as propriedades do mundo de jogo podem, portanto, desenhar padrões. Se os padrões de *Super Mario Bros.* pudessem ser traduzidos em frases e parágrafos, estaríamos diante de um texto sobre as habilidades de um jogador que, aos poucos, percorre o mundo de jogo e progride em sua trajetória. Cada um dos blocos, itens e monstros, bem como suas respectivas propriedades e capacidades são cuidadosamente pensados e posicionados de modo a arquitetar um percurso de aprimoração. Cada etapa deste percurso exige uma ou mais habilidades do jogador, incrementando continuamente seu aprendizado. O jogador, portanto, é o parâmetro principal que guia a construção do mundo de jogo: é o centro daquele mundo.

É nesse sentido, finalmente, que *Minecraft* diverge fundamentalmente de *Super Mario Bros.* e da tradição hegemônica que representa. Se tentássemos ler a topologia de *Minecraft* como uma série de percursos centrados no letramento progressivo do jogador, certamente nos decepcionaríamos com sua falta de organização. Não há posicionamento de tensões simples que se intensificam gradativamente, nem são introduzidas regras em contextos controlados para depois inseri-las em conjunto com outras regras. Essa decepção, no entanto, diz mais sobre as expectativas do leitor do que sobre o objeto de sua análise: *Minecraft* não pode ser lido dessa maneira porque o jogador não é o critério principal que orienta o posicionamento das propriedades, capacidades e tensões que geram em seu mundo. *Minecraft* descentraliza seus jogadores.

As forças intensivas que trafegam no mundo de *Minecraft* não são desenhadas pela justaposição pré-definida de elementos. Cada jogador começa sua jornada em um mundo próprio, e se cartografássemos um mapa representativo da distribuição de objetos em cada um destes mundos, encontraríamos praticamente nenhuma coincidência entre suas coordenadas.

Sendo assim, não devemos abordá-los como formações fixas, mas sim como resultados de parâmetros de distribuição dinâmicos. Além disso, ao contrário de *Super Mario Bros.*, onde a maioria esmagadora dos objetos possui lugar estável, *Minecraft* admite *conectividade*: podemos desconectar um bloco de carvão da montanha a que pertencia e reconectá-lo a uma fornalha que refina minérios de ferro (que, por suas vezes, foram desconectados de suas respectivas minas subterrâneas).

Metaforicamente, poderíamos afirmar que as fases de *Super Mario Bros.* são como melodias clássicas, progressões de notas relativamente fixas e encadeadas. Há certa liberdade de percurso por essas notas: podemos executá-las mais ou menos rápido, nos demorando mais ou menos em certos pontos. Mas este espaço de improviso é fortemente circunscrito e parametrizado por uma estrutura. No jogo, analogamente, as conexões entre os componentes possuem uma rigidez que exerce forças: seu limite de tempo, suas ameaças e recompensas circunscrevem uma zona de influência sobre o jogador, uma continuidade de tendências.

Por mais que admita experimentação, a zona de tendências de *Super Mario Bros.* possui uma estrutura fixa, ao contrário da distribuição de tendências em *Minecraft*, que se parece mais com um tema de jazz sobre o qual se pode improvisar. Mesmo no jazz, existem restrições: temas possuem uma certa progressão de tons, padrões rítmicos e campos harmônicos que devem ser respeitados. No entanto, o conjunto aberto de melodias possíveis nessas estruturas nutre um potencial de conexões que tende mais à *instabilidade* do que à *estabilidade*. *Minecraft* também possui muitas limitações: oceanos não podem flutuar e florestas não podem desaparecer. Porém, dentro de suas regularidades, há relativamente mais espaço para a variabilidade.

A partir da metáfora musical, podemos progredir em direção a uma explicação concreta de como essa diferença se desenha. Neste sentido, podemos finalmente reformular a questão-chave desta pesquisa a partir das informações sintetizadas até aqui. Primeiro, concebemos mundos de jogo como conjuntos de estímulos projetados por uma plataforma sensível. Depois, concluímos que da conexão desta sensibilidade às capacidades sensoriais humanas, contitui-se tanto o mundo jogável propriamente dito quanto o jogador enquanto força capaz de atuar em seu contexto. Notamos também que estes mundos são formados por propriedades extensivas – mensuráveis, divisíveis e redutíveis aos seus elementos básicos – e intensivas – indivisíveis, emergentes e irredutíveis aos elementos discretos que as manifestam. Estas últimas, além disso, só são atualizadas quando estímulos se acoplam às condições perceptivas humanas. Certas configurações de propriedades e capacidades de mundos de jogo têm o jogador como parâmetro conectivo principal, enquanto outras deslocam-o de seu centro.

Em outros termos, *por meio de que mecanismos os mundos de jogo descentralizam seus jogadores?* Nessa pergunta, estão implicadas todas as questões que moverão este estudo. Possuímos um escopo, uma *perspectiva de análise*: concentramo-nos na plataforma sensível dos mundos de jogo e o modo como incorporam a sensorialidade humana e o sistema de regras interno. Selecionamos um *corpus empírico*: lidaremos com jogos cuja dinâmica inverte a hierarquia tradicional que, por motivos culturais e econômicos, coloca o jogador no centro das teias de forças dos mundos de jogo. E, finalmente, manipularemos este *corpus* segundo os princípios de um *arcabouço epistemológico*: o mapeamento topológico de propriedades intensivas e seu contraste em relação à geometria das propriedades extensivas. Por último, restamos elencar nossos objetivos principais, ou seja, as etapas da exploração de nossa questão-chave. Estas serão distribuídas em três capítulos.

No primeiro deles, “Descentralização do jogador: a emergência do mundo de jogo em *Minecraft*”, estudaremos este jogo no sentido de compreender as forças que atuam na adaptação mútua e contínua entre humano e videogame. O caminho argumentativo será trilhado por meio do contraste entre duas fontes: a) as tipologias e categorizações dos *game studies* referentes ao contato jogo-jogador, e b) descrições deste mesmo processo adaptativo em *Minecraft*. Seguindo os rastros apontados pelos inevitáveis atritos gerados por essas comparações, será possível chegar a um conceito de mundo de jogo que se sustente como uma dimensão autônoma da experiência. O objetivo deste capítulo é, portanto, realizar uma demonstração do método de análise que será empregado nos jogos apresentados nos dois capítulos subsequentes. Partiremos das ideias já apresentadas nessa introdução, aprofundando seu rigor teórico e pondo-as em teste de modo concreto com *Minecraft*.

Uma vez munidos de ferramentas afiadas e alinhadas para pensar a dinâmica sensorial entre jogo e jogador, avançaremos para o segundo capítulo: “Uma arqueologia dos ambientes errantes”. Mais do que desenhar uma história dos videogames seguindo as pistas da descentralização do jogador, a intenção é escavar sua trajetória de modo a rastrear os mecanismos usados neste processo. A partir daqui, será necessário desviarmos o foco de *Minecraft* – ainda que este seja um desvio orientado pelas pistas obtidas no primeiro capítulo – no sentido de captarmos os diferentes operadores de descentralização que compõem as dinâmicas dos jogos. Perceberemos que não são novidade na história dos videogames: suas manifestações divergem em direções múltiplas, que não obedecem a suposta linearidade da corrida em direção à perfeição gráfica.

Com uma descrição dos diferentes *mecanismos de descentralização* observáveis nos videogames, nos restará dar conta dos graus de complexidade que suas acoplagens podem

atingir. Elegemos três jogos cujas dinâmicas levam às últimas consequências a descentralização dos jogadores, semeando mundos onde estes tornam-se forças quase insignificantes. *Dwarf Fortress* (2006) e *Ultima Ratio Regum* (2011) tensionam as dinâmicas de seus mundos de tal forma que capturar sequer uma fração de seu potencial envolve uma atitude técnica necessariamente maleável.

Alguns alertas de ordem pragmática: todos os jogos analisados nessa pesquisa são, prioritariamente, para um jogador. No caso do próprio *Minecraft*, o jogo pode ser jogado tanto solitariamente quanto *online*, com outras pessoas. Concentraremos as descrições e análises no primeiro caso. Escolhemos esse caminho de modo a melhor dar conta do contato imediato entre mundo de jogo e jogador, sabendo que a inserção de outros jogadores incrementa a complexidade da situação. Como será defendido no primeiro capítulo, acreditamos que, mesmo assim, as conclusões dessa pesquisa podem ser adaptadas a jogos com múltiplos jogadores.

Outra informação a ser ressaltada é a plataforma escolhida para analisar os jogos. Preferencialmente, optamos pela plataforma PC (Microsoft Windows), de modo a eliminar a maior parte possível de inconsistências entre suportes técnicos e físicos (mouses, teclados, controles, etc.). Como será mostrado, boa parte dos jogos que serão alvo de escrutínio mais delicado encontram-se nessa plataforma. Por exemplo, *Minecraft* possui versões para consoles como Playstation 4 e Xbox One, o que não é o caso dos *roguelikes* abordados no segundo capítulo. Mesmo assim, defenderemos teoricamente – novamente no primeiro capítulo – que embora particularidades de suporte físico alterem as dinâmicas de jogo, isso não significa que as propriedades de um jogo sejam *completamente* perdidas de plataforma para plataforma.

Finalmente, cabe frisar que muitos jogos foram analisados durante o período de pesquisa, embora apenas um pequeno número deles tenha sido alvo de um olhar mais detalhado. Nas referências, estão incluídas fichas técnicas de todos os jogos citados, enquanto os mais relevantes para a pesquisa em si receberam um breve comentário descritivo. Além disso, designamos o endereço de *playlists* do Youtube que foram usadas para analisar os jogos (juntamente da experiência direta de jogo por parte do autor). Esses são jogadores que escolheram compartilhar livremente suas experiências. Utilizamos dois critérios para selecionar as listas: primeiro, evitamos jogadores com um grande número de espectadores (no máximo 5 mil visualizações por vídeo), de modo a minimizar a tensão entre o jogo espontâneo e a postagem de vídeos por dinheiro (também evitamos canais monetizados). Além disso, demos preferência a jogadores que descreveram suas experiências como “*blind let’s play*”, o que quer dizer que nunca haviam jogado aquele jogo específico antes.

Por fim, cabe frisar que escolhemos estudar uma forma específica de comunicação entre jogador e jogo – não se trata de uma abordagem melhor ou pior que outras. Alguém poderá questionar, no entanto: uma abordagem que retira o jogador do centro do desenho da experiência aparenta ser, a princípio, algo absurdo. Afinal, todo jogo só se completa com um jogador. Um ser humano precisa ligar o aparelho, escolher o jogo e, finalmente, *jogar*. A tela, os alto-falantes, os controles foram feitos para atender as demandas ergonômicas de um corpo humano. Acima de tudo e de uma forma ou de outra, todas as nuances do hardware, software e dos mundos de jogo (por mais complexas que sejam) não fariam sentido sem que acabassem chegando ao jogador. Nessa perspectiva, este *sempre* seria o centro do jogo, correto?

Imaginemos, em contrapartida, uma analogia que mostre a insuficiência deste raciocínio. Quando entramos em uma casa bem arquitetada e construída, de preferência uma que tenha sido projetada segundo as necessidades específicas daqueles que nela vivem, percebemos que cada superfície, textura e cor foram medidos e modelados para se adequarem a seus moradores. Sua estrutura mantém-se em pé graças a vigas e concreto armado, mas o que orienta as conexões entre tijolos, azulejos e pisos é a vida humana que preencherá a construção.

A situação é bem diferente quando chegamos a uma floresta ou a um país desconhecido: nada nestes lugares parece ter sido moldado de acordo com nossas habilidades, hábitos e sensorialidades. Por mais que tentemos, de início não conseguiremos dominar as forças em movimento nestes mundos: no primeiro temos dificuldade em achar abrigo e fontes de hidratação e alimentação; no segundo, somos traídos pelo idioma estrangeiro e pelos códigos culturais misteriosos. De um jeito ou de outro, somos protagonistas da nossa própria experiência e, ainda assim, somos atravessados por potências que divergem do nosso repertório de hábitos, capacidades e propriedades.

Nos videogames, podemos imaginar um espectro semelhante, que vai da estabilidade total (o lar cuidadosamente arquitetado) à instabilidade absoluta (a floresta ou país estrangeiro). Na realidade, mesmo nestes modelos, ainda podemos encontrar intercâmbio: a casa pode apresentar um defeito em sua estrutura (instabilidade) e a floresta pode esconder uma caverna (estabilidade). Seria mais adequado falar que cada jogo apresenta *tendências* à estabilidade ou à instabilidade e, por mais que em uma de suas extremidades a experiência seja necessariamente voltada para o jogador, ainda podemos apontar com rigor o grau de descentralização que este sofre. Basta que nos voltemos para seu contato com o mundo de jogo e mostremos as forças divergentes que o atravessam.

1 DESCENTRALIZAÇÃO DO JOGADOR: A EMERGÊNCIA DO MUNDO DE JOGO EM *MINECRAFT*

Figura 5 - Jogadores de Minecraft



Minecraft é consumido em casas, escolas e lan houses. Tablets, smartphones, computadores e laptops. Em TVs, monitores, óculos de realidade virtual e nos hologramas do *HoloLens*⁶, da Microsoft. Pode abrigar um, dois ou dezenas de jogadores simultâneos, no mesmo ambiente físico ou pela internet. Estes podem ser crianças, adultos ou idosos, cada qual com seus próprios traços psíquicos e físicos individuais. Querendo ou não, o encontro entre jogador e *Minecraft* envolve todas estas variáveis, suas sobreposições e reverberações. Descartar qualquer uma delas é um ato delicado, que precisa ser justificado com precisão.

Como dito antes, a abordagem deste trabalho consistirá em selecionar, dessa complexidade, a conexão entre duas superfícies concretas que, de alguma forma, estabelecem um circuito de adaptação mútua: 1) O aparelho sensório-motor humano e 2) a expressão visual, sonora e cinética do sistema que roda o jogo. O ponto mais delicado dessa escolha é especificar que não se trata de analisar estes dois pólos, mas sim sua *comunicação* propriamente dita, aquilo que deixa de existir caso eles se separem. Esse movimento teórico é o primeiro passo para darmos corpo à ideia abstrata de “conexão jogador-jogo”. Este capítulo tem como objetivo detalhar as dinâmicas envolvidas neste processo comunicativo, ressaltando uma variação específica que nos servirá de objeto: a descentralização dos jogadores nos videogames.

Dividiremos este objetivo nos seguintes subcapítulos: primeiro, discutiremos modelos que tomam esse tipo de experiência em seu nível simbólico, narrativo e representativo. Uma contrapartida será defendida: a de que existe uma dimensão sensorial que, embora seja

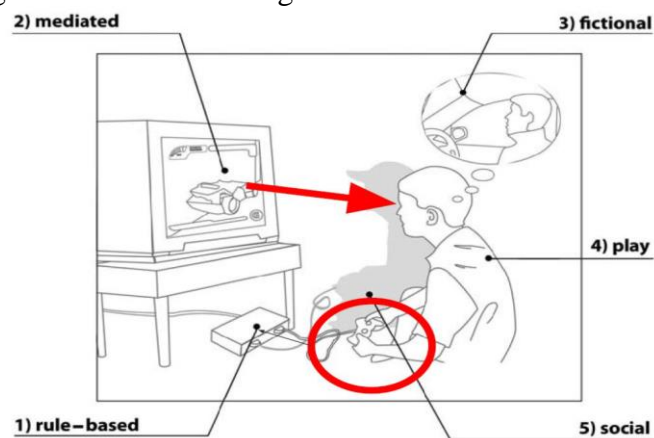
⁶ Dispositivo de realidade aumentada anunciado pela empresa. Fonte: <https://www.microsoft.com/en-us/hololens>.

penetrada por dinâmicas representativas e/ou ficcionais, possui autossuficiência em relação a elas. Depois, continuaremos a esmiuçar esse contato sensorial, que instaura uma dimensão material do jogo, composta por ritmos, movimentos e comportamentos que, novamente, são autossuficientes em relação a variações nos dispositivos de input e output. Subsequentemente, identificaremos como este contato se desenha no espaço e, depois, no tempo do jogo. Por fim, trataremos de dois atributos que contribuem para que o mundo de jogo descentralize o jogador: seus graus de conectividade e de instabilidade.

1.1 Percepção e ação: a autossuficiência dos mundos de jogo em relação à ficção e à representação

De modo a trabalhar exclusivamente com o contato sensorial entre ser humano e mundo de jogo, precisamos deixar de lado não só a variedade de ambientes físicos, plataformas tecnológicas e sociais, mas também as dimensões políticas, sociais e culturais que interferem qualitativamente nos processos de jogo. Não se trata de desprezar sua importância, mas de operar um recorte no objeto de estudo. Para que este recorte não seja arbitrário, é preciso que justifiquemos nossas opções e mostremos que os mundos de jogo se sustentam enquanto uma *dimensão autossuficiente* de um conjunto de elementos interligados e em afetação contínua. O objetivo deste subcapítulo é introduzir gradualmente esta operação, iniciando-a com a comparação entre o que chamamos de mundo de jogo e as dimensões simbólicas e representativas que se fazem presentes na experiência lúdica.

Figura 6 – Modelo de Jogo Michael Nitsche



Em preto e branco, o modelo da experiência de jogo segundo Michael Nitsche (2008, p. 15). Em vermelho, fizemos duas intervenções com o objetivo de explorar suas lacunas: a seta refere-se à expressão sensorial do mundo de jogo, e o círculo ao contato tátil entre jogador e dispositivo de input.

Apesar das incontáveis formas de jogo admitidas por um produto como *Minecraft* – ou qualquer videogame, para sermos justos –, não há carência de modelos que ambicionam dar conta da suposta totalidade desta trama. Com uma amplitude singular, o modelo de Michael Nitsche (2008) busca abranger o fenômeno de jogo a partir de cinco categorias numeradas. Para o autor, o jogo separaria-se nas seguintes camadas: 1) baseada em regras; 2) mediada; 3) ficcional; 4) lúdica e 5) social.

Este modelo nos parece ideal para discutir a composição de um mundo de jogo a partir de estímulos sensoriais por um motivo muito específico: ele deixa esta camada totalmente de fora da equação. Nele, o que mais se aproximaria de indicar uma constituição sensorial do processo de jogo, poderíamos dizer, é a camada que Nitsche chama de “espaço mediado” (p. 16). No entanto, como o próprio título da categoria já insinua, ela atua muito mais como algo que entremeia duas outras coisas do que como uma dimensão autossuficiente e produtiva dos jogos eletrônicos:

O espaço mediado [...] consiste em todos os outputs que o sistema pode manifestar de modo a apresentar o universo baseado em regras ao jogador. No caso dos videogames comerciais, esta camada consiste principalmente em outputs audiovisuais e táteis que providenciam uma forma de apresentação. O jogador é confrontado com esta apresentação, imaginando um mundo com as informações apresentadas: o espaço ficcional. Baseando-se no espaço ficcional, os jogadores decidem como agirão para afetar o espaço de jogo.⁷ (NITSCHKE, 2008, p. 16)

Como fica claro no trecho acima, definir algo como “aquilo que está entre outras duas coisas” diz muito mais dessas duas coisas do que do algo que queríamos definir. Neste caso, a “camada mediada” posiciona-se entre dois outros atores: o sistema de regras e o mundo ficcional, e sua única função é transmitir um suposto conteúdo armazenado pelo primeiro para o recipiente simbólico do segundo. À primeira vista, parece razoável afirmar que os processos que ocorrem na tela (ou em qualquer dispositivo de *output*) “traduzem” o sistema para a apreciação simbólica humana. Subliminarmente, porém, esta ideia pressupõe duas concepções: a) a mediação é um processo caracterizado pela inserção de uma camada entre uma anterior e uma posterior, e b) esse posicionamento implica em uma função de tradução.

Seguindo este raciocínio, poderíamos afirmar que o código de programação é uma camada mediada: este traduz o trabalho do sistema operacional em *outputs* na tela. Por suas vezes, os parâmetros de programação de baixo nível, como as linguagens de máquina e

⁷ Livre tradução de: “the mediated space [...] consists of all the output the system can provide in order to present the rule-based game universe to the player. In the case of commercial video games this layer consists mainly of audiovisual and tactile output that provides a form of presentation. The player is confronted with this presentation and imagines a world from the provided information—the fictional space. Based on the fictional world players decide on actions to affect the game space.”

*assembly*⁸, também se posicionam entre os circuitos elétricos e o sistema operacional, traduzindo justaposições de registradores em comandos programáveis e, depois em um operador de tarefas. Enfim: olhando por este prisma, todas as camadas materiais do computador são mediadas porque posicionam-se entre outras duas. Além disso, todas elas permitem que suas vizinhas se comuniquem entre si: ou seja, todas são tradutoras. Isso nos leva ao primeiro grande problema do modelo de Nitsche: se levarmos seu conceito de “mediação” ao extremo, ele não possui capacidade de diferenciar a camada que chama de “mediada” de outras camadas que também mediam.

Uma das discussões recentes dos *media studies* acerca da ideia de mediação apresenta a crise pela qual o conceito passa, justamente por esse motivo: não se sabe ao certo onde fincar as fronteiras do que ele designa. Richard Grusin (2015) diagnostica uma espécie de preciosismo do campo em considerar que apenas o que costumamos chamar de “mídias” são capazes de realizar o processo que se acostumou em chamar de “mediação”. Na contracorrente, o autor procura sustentar o conceito de “mediação radical” – a lógica ubíqua e múltipla da mediação –, não como aspecto totalizante que une tudo que é heterogêneo, mas percebendo que, de acordo com cada contexto, qualquer coisa pode desempenhar papel mediador:

A mediação radical insiste que, de cima a baixo, temos mediação. Mesmo nos mais básicos e menores componentes existem mediações, que, ao subirmos de escala, pode ser remediados em entidades maiores, assim como descendo de escala, eventos e entidades maiores e mais complexos podem ser remediados em contrapartes menores ou menos complexas⁹. (GRUSIN, 2015, p. 146).

De modo a validar o papel da “camada mediada”, Michael Nitsche afirmará ainda que, para que o jogador compreenda as regras (ou algo que o oriente sobre como atuar no jogo), é necessário que estas sejam recompostas pelo jogador em um universo ficcional. Como o sistema baseado em regras não oferece acesso direto a essa composição, seria necessário algo que amarrasse as duas pontas do modelo: papel atribuído aos outputs do sistema. Este é o segundo problema que demonstra a insuficiência de seu modelo.

Essa concepção parece pressupor duas coisas: a) o universo ficcional tal qual pensado pelo autor não possui contornos materiais: é algo projetado na imaginação do jogador; e b) as dimensões do processo de jogo – pelo menos estas três: sistema, outputs e mundo ficcional –

⁸ Sobre estas duas em específico, conferir o décimo capítulo de “Gödel, Escher, Bach: An Eternal Golden Braid” (HOFSTADTER, 1979, p. 279-343), no qual o autor demonstra que a linguagem computacional consiste em “níveis de descrição”, e o que cada nível acrescenta ao anterior é justamente uma tradução para uma camada de abstração mais alta.

⁹ Livre tradução de: “Radical mediation insists that it is mediation all the way down. Even the smallest or most basic components are mediations, which by scaling up can be remediated into larger entities, just as by scaling down, larger or more complex entities and events can be remediated into smaller or less complex ones.”

encadeiam-se linearmente e teleologicamente até que formem um espaço ficcional fornecedor de informações úteis para que o jogador decida como agir. Para mostrar as limitações desta equação, bastaria indicar algo muito evidente em um processo concreto de jogo: a capacidade do jogador de intuir (parte de) sua lógica sem necessariamente formular um universo ficcional. Ou, melhor dizendo, comprovar empiricamente que a construção mental de um universo ficcional é acessória em relação ao processo de tomadas de decisão no jogo. Sua ligação é antecipada por um terceiro elemento: o circuito sensorial formado por jogo e jogador.

No nível mais básico, poderíamos apontar estudos que testam habilidades sensoriais ligadas à visão e seu respectivo aprimoramento por meio da repetida exposição a experiências de jogo. Dorval e Pépin (1986) constataram que jogadores apresentam melhor desempenho em testes de percepção e transformação mental de objetos tridimensionais após repetidas sessões de jogo envolvendo estas competências. Griffith et al. (1983) apresentam a primeira evidência empírica de que jogadores assíduos possuem maior habilidade de coordenação mão-olho. Orosy-Fildes e Allan (1989) tiraram conclusões semelhantes em testes de tempo de reação. Boot et al. (2008) ainda procuram medir a diferença de aptidão entre jogadores experientes e novatos a uma série de tarefas, como acompanhar movimento de objetos e outras mudanças em seus aspectos.

O que interessa aqui não é apontar se videogames aprimoram ou não tais capacidades cognitivas, já que mesmo o aprimoramento de funções isoladas não garante o aperfeiçoamento de seu uso concomitante (GREEN; BAVELIER, 2004). O ponto fundamental é apontar a existência concreta de uma série de dinâmicas sensoriais que ocorrem nos videogames e que informam as ações e reações de jogadores – sejam elas mais ou menos eficientes. Se uma variedade de estudos do campo da psicologia cognitiva procura mensurá-las e categorizá-las, preferimos aqui procurar indícios de sua atuação em uníssono, o que na nossa hipótese configura *materialmente* o que estamos chamando de mundo de jogo. Tais mundos apresentam atributos próprios, anteriores a uma possível dimensão ficcional, simbólica ou mesmo representativa da experiência (ou ao menos independente destas).

Podemos, a partir disso, operar mais um recorte no objeto, nos livrando das representações ficcionais e tratando os elementos dos videogames apenas na sua condição de componentes da dinâmica de estímulos provenientes do mundo de jogo. Assim, quando daqui em diante fizermos uso das palavras “personagem”, “objeto”, “avatar”, “monstro”, “caverna”, etc., reforçamos que estaremos nos referindo a conjuntos de estímulos – estáticos ou não – que são percebidos de modos específicos pelo jogador.

Figura 7 - Uma caverna em Minecraft



Podemos toma-la como uma representação ficcional, mas também como um conjunto de capacidades e propriedades no mundo de jogo, conjunto este com sua própria dinâmica, autossuficiente em relação às dinâmicas ficcionais e/ou representativa.

Por mais que as palavras usadas para designar estas entidades (“caverna”, por exemplo), remetam a representações (ficcionalis ou não), elas podem remeter também a conjuntos de propriedades e capacidades. Uma caverna real, por exemplo, é um ente sólido e isolante térmico: essas propriedades geram capacidades, como a de abrigar alguém da chuva e do frio. Quando representamos uma caverna visualmente – desenhando em um papel, por exemplo – reimaginamos as texturas, cores, ângulos e curvas que fundam os aspectos visuais de uma caverna natural. De modo análogo, a caverna em um mundo de jogo como o de *Minecraft* reimagina essa mesma caverna visualmente, mas também dinamicamente.

É o que a pesquisadora Rikke Nørgård (2009) chama de “jogo instrumental” (*instrumental play*), alegando que o circuito que se forma entre nossas funções motoras e os estímulos sensoriais configura o cerne da experiência *ativa* de jogo. Descrevendo o processo concreto de uma criança que joga, a autora chama atenção para o fato de que a adaptação e a prática do jogo tomam forma graças à projeção da percepção em conjunção com a manipulação. Assim, “qualquer uso contingente da linguagem funciona de modo parasitário em relação às ações instrumentais ocorrentes, por exemplo ‘Como consigo ir para o outro lado?’ ou ‘Olha, estou fazendo bolos!’” (NØRGÅRD, 2009, p. 2).

Fica claro que o uso de frases como essas duas referem-se, mesmo que não necessariamente a uma relação “parasitária”, pelo menos a uma codeterminação entre a interação sensorial com o mundo de jogo e as demais dimensões da experiência (ficcional, computacional, social, etc.). Por isso, a versão lúdica de uma caverna é mais que um conjunto de códigos que se traduz representação ficcional: é uma dinâmica de propriedades e capacidades que se atualiza no *circuito sensorial* entre jogo e jogador.

Este circuito constitui uma dimensão ativa do jogo, o que remonta à ideia de Alexander Galloway (2008) de que a categoria ontológica essencial para nos referirmos aos elementos que constituem o jogo é a de *ação*¹⁰ (p. 1-38). Certamente a construção de um mundo ficcional pode fazer parte desta dimensão ativa e, de certa forma, pode informar o que um jogador fará ou não em um jogo (Cf. CALLEJA, 2011, p. 119). No entanto, a abertura que permite que estes dois âmbitos conversem não implica a subordinação da dinâmica ativa/perceptiva à construção ficcional ou representativa. A modalidade de ação à qual Galloway se refere opera na materialidade da mídia e nas sensações que ela provoca, um arranjo de dispositivos expressivos cujas propriedades e capacidades físicas originam modos específicos de percepção e comunicação.

Sendo assim, ao invés de nos referirmos aos processos sensoriais empregados na relação entre humano e jogo como “tempo de reação”, “percepção espacial bi ou tridimensional” ou outros termos encontrados nos estudos experimentais da psicologia, adaptaremos estes termos para adequarem-se às múltiplas manifestações particulares aos videogames. A manipulação ou transformação de um objeto tridimensional (DORVAL; PÉPIN, 1986) em *Minecraft* pode se manifestar como a extração de madeira ou a coleta de lava; a coordenação mão-olho (GRIFFITH et al., 1983) pode ser requerida na execução cuidadosa de um salto sobre um buraco, ou na tentativa de alvejar um inimigo com o uso de um arco; O tempo de reação humano (OROSY-FILDES; ALLAN, 1989) pode responder ao som de um zumbi que se aproxima sorrateiramente ou à fuga de uma ovelha ou porco que foge do abate.

Tal junção das capacidades perceptivas humanas às capacidades expressivas dos mundos de jogo pode ser tratada sob a ótica a princípio informal da ideia de *game feel*, elaborada por Steve Swink, e definida como "o controle em tempo real de objetos em um espaço simulado, com a interação enfatizada por polimento" (SWINK, 2008, p. 6). O pulo de um personagem em um jogo bidimensional, como *Super Mario Bros.* (1985), por exemplo, faz uso de uma série de parâmetros independentes que, quando disparados sincronicamente, causam uma determinada sensação de (des)controle sobre a ação.

Em *Donkey Kong* (1983), apertamos o botão de pulo e podemos direcioná-lo para a direita ou para a esquerda. Ou seja, controlamos dois parâmetros: o instante e a direção do pulo. Em *Super Mario Bros.*, porém, podemos controlar cinco parâmetros: o instante, altura, aceleração, desaceleração e extensão do pulo (este último indiretamente, por meio da velocidade do personagem no instante em que apertamos o botão).

¹⁰ Seguindo o raciocínio do próprio autor, existem ações do jogador e do jogo, ações diegéticas ou adiegéticas. No entanto, por mais que as classifiquemos e rotulemos, não há diferença ontológica: todas são ações.

É claro que cada um destes parâmetros, na condição de variável definida na escala do código, são vetores independentes. Porém, quando sua atuação sincrônica se depara com o nosso corpo, recrutamos uma miríade de capacidades de ação e percepção capazes de interrelacionar esses parâmetros e, com o tempo, agrupá-los e concatená-los em uma competência dinâmica:

Um motorista precisa aprender uma variedade de ações independentes para dirigir, como dar a partida, virar, dar marcha-a-ré, controlar, acelerar, frear, checar os espelhos, trocar de marchas, monitorar o tráfego, ler sinais de trânsito, manter a velocidade, notar nomes de rua, entre outras. Estas sub-habilidades são, muitas vezes, autodidáticas, autoensaiadas e autoavaliadas sob uma supervisão mais geral. O resultado é uma corrente de sistemas de hábitos incrivelmente complexa¹¹. (DONALD apud. CLARK, 2001).

A tarefa de controle de múltiplas funções (movimento em dois eixos, velocidade, *timing* do pressionamento de botões, etc.), assim como na construção da capacidade de dirigir um carro, convergem em uma única habilidade: a de controlar o personagem na tela.

No entanto, ainda poderíamos estar deixando de lado a especificidade da aquisição de competências de controle exclusivamente em artefatos digitais. Afinal de contas, o movimento de um carro afeta-nos diretamente por meio das sensações físicas de deslocamento, velocidade e aceleração do veículo. Enquanto isso, jogos provocariam essas sensações como em uma espécie de ilusão: não haveria movimento *real* de personagens e entidades, mas sim conjuntos de pixels deslocando-se em sincronia. O mesmo valeria para a aceleração, desaceleração, saltos, quedas, explosões, e quaisquer outros fenômenos espaço-temporais que supostamente possam ser manifestados no mundo de jogo.

Para ultrapassarmos este impasse, basta notar que a *sensação* de movimento, aceleração, etc., é mais relevante para nossas intenções que indagar se este movimento está ou não acontecendo fisicamente na “realidade”. Estudos costumam apontar e mensurar, por exemplo, a percepção de movimento em dispositivos imagéticos, digitais ou não (SPERLING, 1976; McCAULEY, 1992), bem como a sensação de peso e massa de objetos em ambientes virtuais (DOMINJON et al., 2005). Em um jogo de corrida como *Gran Turismo 6* (2013), podemos ver um carro de corrida acelerar com mais ou menos intensidade e, de acordo com a sensação causada, modular o movimento do controle de input de modo a fazer a curva à frente. Assim, a ideia intuitiva de *game feel* ganha consistência empírica, influenciando, por exemplo, o estudo

¹¹Livre tradução de: A driver must learn a whole range of somewhat independent actions for driving, such as starting, turning, backing-up, steering, accelerating, braking, checking the mirror, shifting gears, monitoring traffic, reading road signs, maintaining speed, keeping track of directions and street names, and so on. These sub-skills are usually self-taught, self-rehearsed, and self-evaluated with some overall guidance. The result is an amazingly complex chain of habit systems

específico da mensuração da sensação de controle sobre o pulo de personagens em jogos de plataforma 2D (FOSTERHOLDT, 2015). A realidade do movimento é menos importante que a inegável presença da *sensação* de movimento.

Esse tipo de sensação é uma composição que emerge do contato entre as múltiplas funções psicomotoras humanas e um sistema que pode mudar seus aspectos de múltiplas formas (direção, [des]aceleração, velocidade, etc.). Os objetos, personagens e eventos ocorridos durante o jogo não se apresentariam a nós caso não ocorresse essa emergência. O circuito que a faz funcionar é mais do que metafórico: iniciamos ações que começam em nossas mãos e dedos (no caso de joysticks clássicos), passam pelo processamento interno e tornam-se estímulos audiovisuais que reforçam as sensações de peso, inércia e força. Não há separação rígida entre ação humana e maquínica durante o jogo.

Poderia-se alegar, contra esta afirmação, que uma vez que nossas ações atravessam a fronteira entre o humano e o maquínico, elas deixam de ser nossas e passam a ser exclusivamente das máquinas. Essa visão aparentemente coerente desmorona quando observamos, mais uma vez, um processo concreto de jogo: uma vez munidos da capacidade de controle, passamos a usá-las para produzir novos mecanismos perceptivos. Em outras palavras, transformamos partes do mundo de jogo em acessórios do nosso aparelho sensório-motor.

Por mais esdrúxula que esta ideia pareça, é o que os pesquisadores David Kirsch e Paul Maglio (1992) constataram ao investigar o comportamento de jogadores de *Tetris*. Duas peças do jogo são assimétricas: as em formato de “L” e “Z”, que possuem variações espelhadas (peças em forma de “L” e “Z” invertidos). Quando elas aparecem para jogadores experientes, imediatamente estes apertam botões para a) girá-las; ou b) encostá-las na extremidade mais próxima do campo de jogo. Essas ações permitem que os jogadores percebam imediatamente se ganharam uma peça “destra” ou “canhota”. O reposicionamento das peças gera padrões visuais mais reconhecíveis (O “L” destro encosta na parede de forma diferente do canhoto), fazendo com que os jogadores economizem o tempo que gastariam para processar mentalmente a orientação da peça.

Em suma, Kirsch e Maglio apontam que *os movimentos processados no mundo de jogo constituem ferramentas perceptivas mais eficientes que a tarefa de analisar mentalmente as peças*. Usamos o ambiente – mesmo que ele seja digital – para auxiliar-nos na execução de processos cognitivos. As propriedades visuaudiomotoras¹² (PEREIRA, 2008) de um sistema, portanto, projetam-se em direção à nossa percepção, mas, pelo caminho inverso, nossas

¹² Além das audiovisualidades do sistema, linguagens visuaudiomotoras acrescentam à mistura a tatilidade e a manipulação, gerando um arranjo sensorial qualitativamente distinto.

capacidades perceptivas também são incrementadas pelas propriedades dos mundos de jogo. Este exemplo mostra que tanto a força perceptiva humana quanto o espectro de estímulos digitais encontram-se em um território misto. Neste terreno, um se apropria do outro, como em uma composição química formada por elementos heterogêneos. Sua união gera novas funções e capacidades e, uma vez misturados, só podem ser dissipados por um processo especial que interrompe sua comunicação.

1.2 Lócus de manipulação: A autossuficiência dos mundos de jogo em relação aos dispositivos de input e output

Essa adaptação, analogamente ao exemplo da direção de um automóvel, envolve o recrutamento de múltiplas habilidades, e pede que as façamos trabalhar em uníssono de modo a dominar um sistema que pode variar em diversos eixos. O choque de forças acarreta na transformação de capacidades sensoriais em capacidades lúdicas, já que estas alteram o mundo de jogo. Por suas vezes, estímulos tornam-se qualidades, movimentos e intensidades percebidas. Surge o mundo de jogo:

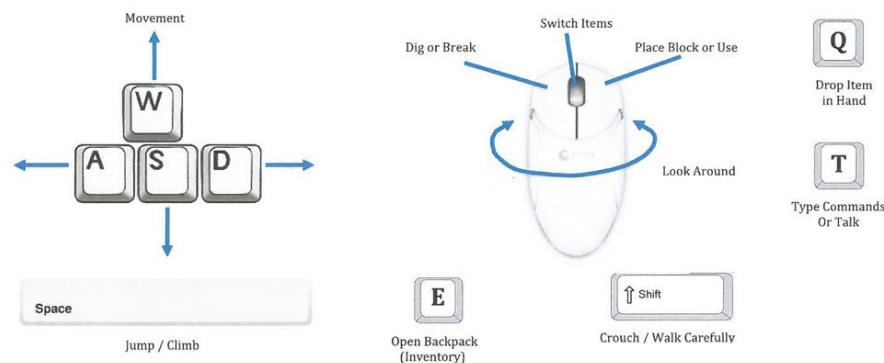
Mundos de jogo são ambientes virtuais e artificiais, desenhados com o processo de jogo em mente. Como mundos, possuem [...] qualidades que os tornam únicos e distinguíveis de outros mundos e os atribuem um certo grau de autossuficiência, que demandam propriedades de navegação e possibilidades de interação específicas (JØRGENSEN, 2013, p. 3).

Até este momento da discussão, percebemos que o modelo da experiência de jogo proposto por Michael Nitsche ignora pelo menos alguns aspectos do quadro geral do fenômeno de jogo. Sobretudo, chamamos atenção para a dimensão ativa da relação entre capacidades perceptivas do jogador e da dinâmica de exibição de estímulos do videogame. Procuramos embasar a decisão de tomar essa dimensão da experiência como um território autossuficiente da estética dos videogames, tal qual apontado por Kristine Jørgensen (2013) e pelo conceito de *gameworld interface* (que aqui traduzimos como “mundo de jogo”). Assim, não só delimitamos um campo para que investiguemos a descentralização dos jogadores, como preparamos o terreno para entender o papel da dimensão sensorial neste processo.

Podemos, portanto, continuar a recortar nosso modelo de mundo de jogo, selecionando as forças que constituem a dinâmica responsável por engendrar sua conexão ao aparelho sensorio-motor humano. Ao contrário do modelo de Michael Nitsche, o de Zagal et al. (2010), contempla a conexão jogo/jogador neste âmbito. Para os autores, estes dois pólos conectam-se por meio de pelo menos outras duas plataformas intermediárias: a primeira consiste no

acoplamento do jogador a um *dispositivo de input*, enquanto a segunda trata da conexão deste mesmo dispositivo primeiro a um *método de input* e, em seguida, ao *locus de manipulação* (Ibid., 2010, p. 9). Jogar *Minecraft* em um computador, sob esse modo de análise, requer a conexão tátil ao mouse e ao teclado (dispositivos de input), que mobilizam uma variedade de métodos de input¹³.

Figura 8 - A associação entre dispositivos de input e o locus de manipulação é mediada por um método de input



Fonte: ZAGAL et al., 2010

O primeiro componente, o dispositivo de input, é o mais evidente nos exemplos dos quais vínhamos tratando: para interferir no estado da composição de estímulos do jogo, é necessário que disparemos sinais elétricos por meio de movimentos corporais. Isso pode se dar sob um espectro de dispositivos distintos, desde joysticks analógicos, passando por botões digitais até a chegada de sensores de movimento, câmeras, entre outros aparelhos capazes de detectar perturbações externas e traduzir em informação.

Logo em seguida, na etapa dos métodos de input, nos defrontamos justamente com o modo como o sistema traduzirá nossos inputs em mudanças audiovisuais correspondentes. Os autores descrevem dois métodos de input: direto e indireto. O direto assimila o movimento contínuo humano a um movimento correspondentemente contínuo na tela; por exemplo, quando arrastamos o mouse horizontalmente e nossa perspectiva gira para a direita ou esquerda em ritmo análogo, para contemplarmos o cenário em *Minecraft*. Se manipularmos o mouse verticalmente, poderemos olhar para cima e para baixo, e assim por diante. O método indireto associa inputs aleatórios a resultados também aleatórios; por exemplo, quando apertamos “Q”

¹³ Os autores mencionam exemplos de dois métodos possíveis de input: direto e indireto. O mouse em *Minecraft* ativa inputs diretos, ou seja, associam o movimento do jogador a um movimento projetado na tela – o caso, movimentando continuamente a visão em primeira pessoa, bem como os cursores que manipulam o inventário e o menu. O teclado faz uso tanto de manipulação direta (é usado para ativar o movimento contínuo do avatar, por exemplo) quanto indireta (comandos específicos que ativam funções pré-definidas (como apertar “Q” para derrubar um item no chão).

para jogar fora um objeto que carregamos em nossas mãos: não há relação direta entre a tecla e o resultado que gera, trata-se de uma associação arbitrária, convencional.

É quando chegamos ao lócus de manipulação que encontramos aquilo que poderia dar forma a um mundo de jogo assimilável. Aqui, adentramos o terreno do que os autores chamam de “entidades”, também referidas como “objetos” (Ibid., p. 8). Como o nome sugere, estes objetos (personagens, ferramentas e itens em geral) serão manipulados de modo que seus estados sejam modificados. O lócus de manipulação, portanto já se estende além da atividade tátil de manipular mouse e teclado, prolongando essa intervenção até o software e ao hardware e, por continuidade, ao mundo de jogo e sua composição de estímulos.

Embora os autores não afirmem isso diretamente em seu texto, quando observamos um processo concreto de jogo, algo fica muito claro: a manipulação de entidades/objetos da qual estamos falando não é algo localizável em um quadrante específico do modelo que discutimos. Apertar teclas ou mover o mouse é, sem dúvida, um processo de manipulação, mas não necessariamente de personagens ou ferramentas em um jogo, já que o computador ao qual se conectam pode estar desligado, por exemplo. Da mesma forma, as mudanças internas do sistema representam modificações nas propriedades de objetos, mas não é possível que chamemos essa mutabilidade de manipulação a não ser que esteja ligada a inputs provocados por um ser humano.

Nesse sentido, o lócus de manipulação tal qual pensado por Zagal et al. assemelha-se com a dimensão sensorial da experiência de jogo que aqui procuramos apresentar. Assim como as sensações de peso e (des)aceleração que conferem qualidades distintas aos saltos de personagens de *Donkey Kong* ou *Super Mario Bros.*, o lócus de manipulação só parece fazer sentido enquanto um território limítrofe. É resultado da comunicação entre 1) aparelho sensório-motor, 2) dispositivo de input e 3) a exibição de mudanças audiovisuais por parte do jogo. Caso rompêssemos com o contato de um destes elementos com suas contrapartes, seria impossível afirmar que ainda existe a capacidade de manipulação. Em outras palavras, o lócus de manipulação *emerge*¹⁴ da conexão entre três conjuntos distintos de capacidades. Como tal, faz nascer uma nova gama de propriedades e capacidades no decorrer da adaptação mútua entre jogo e jogador.

¹⁴ O conceito de emergência será discutido com maior atenção nos capítulos 2 e 3. Por enquanto, basta apontar que, aqui, trataremos a ideia de emergência como processo que faz surgir propriedades e capacidades em uma escala que não estavam presentes na escala anterior. Como mostraremos adiante, esta definição relativamente simples pode se adequar às nuances detalhadas pelos trabalhos dos filósofos Andy Clark, Manuel Delanda e Douglas Hofstadter.

Aqui, porém, já é possível apontar uma ligação não endereçada pelo sistema de análise: a via que expressa o lócus de manipulação diretamente ao aparelho sensório-motor do jogador. Embora o conceito de lócus de manipulação pareça apontar justamente essa via, não é o que é defendido pelo modelo. Os autores admitem que há uma dimensão ontológica do jogo ligada ao modo como as regras manifestam-se audiovisualmente, alocando esta função em uma categoria totalmente separada, que chamam de “*presentation*” (p. 10). No entanto, acabam por definir que essa categoria se resume a três subcamadas: hardware, software e cardinalidade. Apenas esta última dá conta de um processo sensorial, já que as outras duas não apresentam-se diretamente aos sentidos de quem joga.

Optamos, portanto, por operar uma modificação no conceito dos autores, associando a ideia de lócus de manipulação não a objetos abstratos sujeitos a regras abstratas, mas sim ao mundo de jogo e sua dinâmica expressiva. O lócus de manipulação, nesse sentido, pode ser pensado como a projeção das condições de input (dispositivo + método) na dinâmica sensorial do jogo. Em *Minecraft*, este lócus consiste no conjunto de superfícies e forças passíveis de serem alteradas pelo jogador: tudo que for destrutível, coletável e reposicionável; personagens que possam ser atacados, recursos que admitam conexão e ferramentas que possam ser construídas e utilizadas, entre inúmeras outras.

Há de se notar, porém que, por mais que esse conceito abarque uma enorme gama de processos, ele não consiste necessariamente na totalidade do mundo de jogo: o sol em *Minecraft*, por exemplo, não faz parte do lócus de manipulação, já que não admite alteração direta por parte dos jogadores. O mundo de *Minecraft*, portanto, também é composto por estímulos que não se conectam imediatamente ao dispositivo e método de input. Não obstante, estes estímulos não-manipuláveis ainda são, na hipótese mais branda, fontes de informação; ou, como no caso do sol em *Minecraft*, entidades que afetam o mundo mas não podem sofrer afetações.

Figura 9 – Super Mario Bros e Minecraft



Setas vermelhas apontam, em Super Mario Bros. (esquerda) e Minecraft (direita) os componentes do mundo de jogo que indicam informações mas não podem ser alvo de interação direta.

Este é um dos pontos que dão consistência à afirmação de Kristine Jørgensen de que a composição sensorial do jogo dá origem a um mundo dotado de autossuficiência, uma “interface entre o sistema e o jogador”. No entanto, é uma interface que opera uma *concretização* do sistema (JØRGENSEN, 2013, p. 5). Segundo a autora, essa concretização funciona nos moldes de uma metáfora, transformando um sistema de nível mais abstrato em um conjunto de parâmetros assimiláveis sensorialmente. Assim como em uma metáfora linguística, não há limites discerníveis para metáforas interfaciais: um mesmo sistema abstrato poderia ser concretizado de diferentes modos, acarretando no surgimento de mundos de jogo distintos.

Um caso extremo que demonstra o poder criativo da concretização dos mundos de jogo é a prática de *modding*¹⁵, especificamente aquelas que reconfiguram os jogos de tal maneira que o resultado final separa-se completamente do “original”. Por exemplo, o estudo de José Messias (2016) a respeito das práticas de modificação de comunidades brasileiras nas franquias *Guitar Hero* e *Pro Evolution Soccer* comprova o poder produtivo das interfaces gráficas. As modificações colaborativas dos hackers davam-se, sobretudo, nesta camada do jogo, alterando cores, texturas e textos para que estes se adequassem a outras demandas estéticas. O resultado, segundo o autor, é uma experiência de jogo qualitativamente distinta.

Começamos a perceber, portanto, que não só o fenômeno de jogo como um todo é multifacetado, mas também que se o abordarmos apenas no que tange o contato sensorial humano-videogame, descobriremos diversas camadas de funcionamento. Havíamos chamado

¹⁵ O termo refere-se a modificações – encorajadas ou não pelos desenvolvedores originais do jogo – sobre o software, interface ou qualquer atributo do jogo. Mods podem alterar desde elementos ficcionais, passando por aprimoramentos ou mudanças funcionais de usabilidade ou intervenções na performance ou processamento do código.

atenção para o acoplamento sensorial aos dispositivos de input mas, ao introduzir a ideia de concretização, percebemos também que há uma continuidade que se desenvolve pelo lado oposto: a do sistema interno em direção aos nossos sentidos. É possível, assim, que selecionemos ainda de modo ainda mais restrito o objeto de análise e desprezemos tanto as variações nos dispositivos de input/output (estamos jogando *Minecraft* em um computador ou em um tablet?) e no aparelho sensório-motor (uma criança ou um adulto está jogando? É um jogador experiente ou iniciante?).

É importante chamar atenção para o fato de que desprezar as variações nestes dois âmbitos não é o mesmo que tê-los como insignificantes. Pelo contrário: eles são de extrema relevância para abordarmos o contato sensorial entre jogador e mundo de jogo. Diferentes dispositivos de input configuram plataformas sensoriais de ordens qualitativamente distintas. O exemplo mais claro e recente foi a popularização dos controles de movimento durante a era *Nintendo Wii* (Cf. JUUL, 2009). No *Wii*, certos jogos podem ser jogados tanto por meio dos *Wii Remotes* (joysticks equipados com acelerômetros que detectam movimento em três eixos) ou pelos *Wii Classic Remotes* (joysticks de estilo tradicional, dotados de botões simples e direcionais analógicos e digitais).

Um jogo de tiro como *Call of Duty: Modern Warfare 3*, por exemplo, exigiria movimentos constantes do *Wii Remote*, no qual o jogador muitas vezes precisa engajar músculos ligados a movimentos do braço, pulso e cotovelo. Enquanto isso, o *Wii Classic Remote* pode ser manipulado quase que a despeito destes músculos, envolvendo essencialmente o movimento das mãos e dedos. No entanto, acreditamos que, ainda assim, o jogo possui regularidades que tornam possível que analisemos sua manifestação sensorial de modo independente de que tipo de controle administra os inputs.

Por exemplo, a velocidade dos personagens, o poder de fogo de cada arma, a velocidade de seus projéteis, os pontos-cegos em cada arena de batalha, o efeito de câmera lenta causado pela explosão de granadas de efeito moral, entre outros processos intensivos: todos eles permanecem estáveis, mesmo que despluguemos o controle de movimento e joguemos com o clássico. Sendo assim, a agência desta aglomeração de propriedades possui autossuficiência em relação aos dispositivos de input e seu próprio conjunto de propriedades e capacidades: eles comunicam-se e se entrecruzam, mas possuem um grau suficiente de clausura. Optamos por focar no primeiro grupo de propriedades, ainda que levando em conta seu contato com o segundo. Assim, acreditamos ser possível contribuir e colaborar, por exemplo, com um estudo detalhado da influência de diferentes dispositivos e métodos de input na experiência de jogo.

O último passo que nos resta é apontar a serventia deste mesmo raciocínio para ratificar a autossuficiência dos mundos de jogo também em relação aos dispositivos de exibição destes mundos, ou seja, os *dispositivos de output*. A questão torna-se especialmente complexa quando percebemos que as próprias materialidades destes dispositivos influenciam suas contrapartes: os inputs possíveis no sistema (LUNDGREN e BJORK, 2003). Por exemplo, os óculos de realidade virtual *Rift* da Oculus e *Vive* da HTC/Valve consistem simultaneamente em dispositivos de input e output: ao mesmo tempo que possuem telas duplas – uma por olho – que causam a sensação de profundidade, lançam mão de giroscópios e acelerômetros para rastrear movimento da cabeça do usuário, permitindo que este olhe o cenário a sua volta.

Seria possível, então, ignorar a individualidade de cada modo de exibição de outputs no sentido de estudar os mundos de jogo isoladamente? O que seriam esses mundos de jogo, para início de conversa, sem um veículo concreto que os manifestem sensivelmente? Estas questões apontam perigos que pretendemos evitar ao lidar com a sutil diferença entre ignorar os dispositivos de input e, por outro lado, tomá-los como um conjunto de propriedades comuns. Enquanto na primeira abordagem estaríamos abrindo mão de examinar a materialidade dos mundos de jogo, no segundo reuniríamos mecanismos processuais postos em movimento por uma multiplicidade de dispositivos de outputs diferentes.

A chave da questão é percebê-los não como entidades estáticas, mas enquanto *modos de funcionamento*. As operações que colocam em prática são materialmente semelhantes, mesmo nos casos mais extremos: vinculam segmentos discretos para compor imagens e sequenciam composições para simular movimento. A maioria dos monitores de computador e displays televisivos fazem isso a partir de pixels, elementos discretos de um plano cartesiano bidimensional. O Atari VCS, por sua vez, não se utiliza de pixels, pelo menos não do mesmo modo: ele sintetiza imagens linha por linha da horizontal (192 linhas em uma TV NTSC convencional) e, na vertical, essas linhas são operadas temporalmente em mudanças nas instruções que trafegam do processador ao adaptador de interface (MONTFORT e BOGOST, 2009, p. 28). Isso implica em mudanças na composição de imagens, mas também no efeito como o movimento das imagens é exposto, por meio de sequências de composições individuais. Finalmente, estas afetam qualitativamente a visualização das imagens e de seus respectivos movimentos.

Não obstante, tanto o Atari VCS quanto computadores contemporâneos e até mesmo óculos de realidade virtual usam segmentos discretos para compor imagens e sequenciação de imagens para simular movimento. Mesmo assim, para fins de praticidade, seguiremos o rastro das plataformas que rodam *Minecraft*: todas elas utilizam-se de pixels, mas possuem, por

exemplo, tamanhos de tela distintos. Smartphones, tablets, notebooks, televisores, monitores de computador: suas áreas de display variam de algumas polegadas a dezenas delas. Variam em suas propriedades como brilho, contraste, nitidez, calibre de cor, entre muitas outras especificações técnicas. Para não dizer as discrepâncias na experiência sonora: alto-falantes e headphones de infindáveis tipos configuram uma miríade de experiências auditivas qualitativamente distintas.

O que causa, no entanto, a diferença de experiência entre uma tela maior e uma menor, ou entre um headphone com isolamento acústico e um alto-falante de smartphone? Certamente não são apenas e tão somente as especificações técnicas do sistema, embora estas indiquem parcialmente o que podemos esperar do resultado final. De fato, o que muda a qualidade sensorial da experiência é a *relação* entre nossas capacidades perceptivas e a atuação contínua destes aparelhos e seus respectivos atributos.

Ainda assim, a possibilidade de deixar de lado a individualidade de cada dispositivo de output pode dar a falsa sensação de que nos distanciaríamos da materialidade dos videogames. Contra esta ideia, cabe voltar às pesquisas que citamos anteriormente, que verificam empiricamente a sensação de movimento, peso, aceleração e outras propriedades de objetos digitais. Mesmo que não possamos tocar ou observar essas sensações, podemos experimentá-las e, como nos estudos, constatar sua indiscutível materialidade. É certo que elas são alteradas pelas capacidades específicas de cada dispositivo de output que as expressa. No entanto, a recíproca é verdadeira: a teia de forças dos mundos de jogo também afeta materialmente a superfície que atualiza seus estímulos; não modificando sua constituição física, mas impondo ritmos sensoriais aos pixels e alto-falantes anteriormente inertes.

Gilbert Simondon (2008, p. 154-159) abordaria este processo segundo sua própria concepção de informação, influenciada pelo estudo de Claude Shannon, Warren Weaver e Arthur Burks (1951), à qual incorpora sua própria ontologia da associação humano-máquina. Enquanto a máquina articula *formas*, o ser vivente articula *informações*¹⁶, a segunda sendo

¹⁶ Neste ponto, cabe uma breve digressão com o objetivo de determinar a diferença entre o uso da palavra “informação” no trabalho de Simondon e, em um contexto totalmente distinto, o emprego da mesma palavra no campo da comunicação (Cf. SODRÉ, 2014). Neste, temos uma visão que se vale da ideia de “transmissão de informação”, que passa por um canal, para chamar atenção para uma dimensão mais complexa, que Muniz Sodré chamará de formação do “comum”. Simondon, no entanto, não diria que a máquina trabalha com informação, e sim com formas: por exemplo, quando um servidor envia um pacote de dados para outro. A máquina, segundo o autor, não sabe associar e tecer o contexto das formas com as quais trabalha (nesse caso, bits). Ela apenas as manipula segundo instruções específicas, repetindo-as segundo um gesto humano que atribui a máquina um conjunto de parâmetros. Assim, da mesma forma que seu conceito de indivíduo pressupõe uma dinâmica de individuação, o que Simondon chama de informação vai além da simples manipulação de dados: implica a inserção da variabilidade das formas em um contexto tecido pela ação e

justamente o acréscimo de uma margem de imprevisibilidade sobre a estabilidade da primeira. Informação, para Simondon, é o efeito simultâneo da relação simbiótica entre determinação (sinal) e caos (ruído); é algo que se move entre o determinado e o indetermindado, e “o que aparta uma série de estados imprevisíveis, novos, que não fazem parte de nenhuma série discernível de antemão”¹⁷ (SIMONDON, 2008, p. 152).

Sob esta perspectiva, as formas limitadas de monitores e auto-falantes precisam acomodar uma amplitude infundável de combinações para que possuam capacidade de informar. O mesmo conjunto de pixels de uma tela precisa “saber” expressar todos os milhares de vídeos armazenados no YouTube todos os dias, enquanto amplificadores de som – exemplo do próprio Simondon (p. 153) – precisam ter a capacidade de vibrar todas as frequências presentes nas músicas que ainda serão compostas e gravadas. Essa aparente incompatibilidade entre as formas pré-definidas e restritas dos suportes e a virtualidade sem fundo do conjunto de composições que podem ser veiculadas por eles só pode ser explicada de um ponto de vista que ultrapasse a dicotomia sujeito-objeto. Ou, melhor, dizendo, a dicotomia suporte-conteúdo.

De igual para igual, os ritmos do videogame e as superfícies que os manifestam entram em consonância para dar corpo ao aspecto sensorial da experiência de jogo. Pela última vez, portanto, apontamos a preferência deste trabalho pela primeira dimensão em relação à segunda, embora ambas atuem em proximidade estreita. Este foco permitirá que mais atenção seja direcionada ao que, por enquanto, estamos chamando de dimensão rítmica do jogo. Mostraremos que *Minecraft* possui uma miríade de forças rítmicas, que se manifestam de modo múltiplo nas suas várias versões e plataformas. Tomadas em sua autossuficiência, podem ser pensadas como os elementos constitutivos das propriedades intensivas dos mundos de jogo. A partir do próximo item, continuaremos a abordar o contato entre jogo e jogador a partir do arcabouço empírico que esboçamos até aqui. Isso exigirá que intensifiquemos o nível de detalhamento de nossa discussão, mostrando que não só os mundos de jogo modificam materialmente os dispositivos de output: eles projetam-se até o jogador, interferindo nos modos como sua agência se manifesta no jogo.

percepção humanas, distanciando-se (embora não negando totalmente) a teoria matemática da informação de Shannon e Weaver (Cf. ILIADIS, 2013).

¹⁷ Livre tradução de: lo que aparta una serie de estados imprevisibles, nuevos, que no forman parte de ninguna serie ddinible de antemano”

1.3 Perspectiva e ambiente: a dimensão espacial da descentralização do jogador

O desafio, a partir deste ponto, consiste em atravessar o limiar do contato entre o que havíamos chamado de lócus de manipulação (conjunto de operações potencialmente jogáveis) e o mundo de jogo como um todo. Aqui, já podemos introduzir a ideia aparentemente controversa de que esse lócus *coincide* com o que chamaremos de “jogador”. Claus Pias (2011) introduz este mesmo princípio de maneira, no mínimo, curiosa: para o autor, o jogo transforma o ser humano em dispositivo, ao passo que o ser humano apreende o sistema em sua dimensão mais humanizada: a interface. Não é o ser humano que está no centro, mas uma entidade que se posiciona como plataforma onde duas naturezas distintas passam a operar em consonância:

“...algo como uma interface move-se para este centro e media a contradição entre máquina e ser humano, hardware e wetware, ao mesmo tempo criando e formatando aquilo que o ser humano é enquanto exerce a condição de usuário” (PIAS, 2011, p. 166).

No caso dos videogames, esta interface é o conjunto de estímulos que movimentam mundo de jogo. Enquanto Kristine Jørgensen limita-se a afirmar que um mundo de jogo é uma interface “porque é representacional em diversos níveis, e porque é o principal veículo de informação e aquilo que permite interação com o jogo”¹⁸ (JØRGENSEN, 2013, p. 2), Pias parece se interessar pelos processos subliminares que tornam essa operação possível. Para o autor, a interface homem-máquina atua no limiar do contato por meio de dois tipos de esforço: um de “visibilidade” e outro de “lentidão” (PIAS, 2012, p. 179). A interface visibiliza processos ao mesmo tempo que os desacelera.

De certa forma, estes processos remetem à ideia de concretização apresentada por Jørgensen – a conversão dos processos internos do sistema em um regime sensorial inteligível. O que Pias acrescenta ao modelo, no entanto, é que essa concretização se dá espacialmente (visibilidade) e temporalmente (lentidão). Ao mesmo tempo que transforma processos lógicos em movimentos audiovisuais, discerníveis por seres humanos, o computador desacelera sua velocidade de processamento para que experimentemos ritmos mais cadenciados, também adequados à nossa percepção. Caso mantivessem na sua velocidade máxima, mesmo os processadores mais antigos ultrapassariam em muito a nossa capacidade de acompanhar a mudança de estados do sistema.

¹⁸ Livre tradução de: “because it is representational on several levels, and because it is the primary carrier of information and what allows interaction with the game”.

Do lado oposto, o ser humano converte-se em uma “corporificação de um modo particular de comunicação entre dispositivos” (PIAS, 2011, p. 173) – um jogador –, sincronizando-se aos ritmos relativamente lentos e às oscilações audiovisuais do computador. Isso quer dizer que o ser humano, ao posicionar-se na condição de operador de inputs, responde ao computador como se fosse um servidor ou um programa. Pias defende sua hipótese por meio de um resgate histórico dos fundamentos de programação: chamamos de “ping”, um envio de dados que um programa manda a outro para saber se a comunicação está ativa. O segundo programa, caso receba a chamada, responde enviando dados correspondentes, o que comumente é chamado de “pong”. O ser humano age como este segundo programa: o jogo nos envia dados e respondemos com nossas próprias ações. Nesse caso, o jogo nos envia dados sensoriais e respondemos com ações que viram dados digitais. Cada jogo, antigo ou recente, emula, portanto, a estrutura básica de um dos primeiros videogames comerciais: *Pong* (1972).

Tanto jogo quanto jogador, ao encontrarem-se em um território de dinâmicas espaço-temporais, podem atuar em suas respectivas zonas de liberdades e constrangimentos que chamamos simplesmente de regras (JUUL, 2005, p. 36). Chegou a hora de observarmos as nuances específicas destas zonas espaço-temporais. Abordaremos, primeiro, a sua dimensão espacial, para depois acrescentar a lógica temporal do contato jogo-jogador.

Novamente segundo Zagal et al., nosso contato com o jogo é filtrado por um “ponto de vista” (*point of view*) (p. 10). Muito embora os autores não cheguem a desenvolver o significado deste termo em sua hierarquia de categorias, um conceito de ponto de vista nos jogos é elaborado com mais atenção na tipologia de Aarseth et al., (2003). No artigo, os autores encaram o objetivo de desenvolver parâmetros para definir gêneros de jogos segundo critérios objetivos. A abordagem dos autores procura dar conta especificamente da dimensão espacial dos videogames (e de jogos em geral, na realidade). Em outras palavras, privilegiam jogos “baseados em movimento espacial” (AARSETH et al., 2003, p. 48). Assim, construirão uma espécie de tabela tipológica com uma série de categorias, cada uma das quais com suas respectivas subcategorias.

É justamente pela categoria intitulada “Espaço” que o texto começa a traçar sua planilha tipológica. Dentro desta categoria-mestra, encontramos três subcategorias: “Perspectiva” (que pode ser onipresente ou vagueante), “Topografia” (que pode ser geométrica ou topológica) e ambiente (que pode ser classificado como estático ou dinâmico). É na ideia de perspectiva (que remete à de “ponto de vista”) que é identificaremos uma ferramenta conceitual para explorarmos o modo como o jogo filtra seu contato com o jogador.

A perspectiva em um videogame, segundo os autores, pode ser “Onipresente” ou “Vagueante” (p. 49). Em *Minecraft* – exemplo de perspectiva vagueante – a princípio, temos acesso a uma porção ínfima do espaço total do jogo. Os limites desta porção coincidem com o enquadramento móvel da “câmera” em primeira pessoa através da qual contemplamos o cenário. Ao longo da história dos videogames, tornou-se comum distinguirmos entre jogos em “primeira pessoa” ou “terceira pessoa”. Um jogo de tiro, por exemplo, é em terceira pessoa se conseguimos ver o personagem que controlamos durante o processo de jogo, e em primeira pessoa se a tela funciona como uma câmera subjetiva cinematográfica¹⁹. É importante perceber, porém, que a subcategoria não pretende distinguir entre diferentes relações entre, e sim entre *modos como o espaço se apresenta ao jogador*.

Figura 10 - Visualização de Minecraft em primeira pessoa (esquerda) e terceira pessoa (direita).



Mesmo se trocarmos para uma perspectiva em terceira pessoa, na qual nosso avatar é visível na tela, ainda teríamos acesso a uma parcela relativamente inexpressiva da totalidade de informações e processos que circulam no mundo de jogo. Portanto, não basta que classifiquemos a perspectiva segundo estas categorias, pois elas não possuem “poder diferenciante” (Ibid., p. 49). A polaridade onipresente/vagueante, pelo contrário, permitiria que olhemos para certos jogos e os diferenciemos sua perspectiva qualitativamente.

¹⁹ No segundo capítulo de seu livro “Gaming: Essays on Algorithmic Culture”, Alexander Galloway (2006, p. 39-69) realiza um estudo comparativo entre a origem da câmera subjetiva cinematográfica e a visão em primeira pessoa nos videogames. Essencialmente, trata-se da mesma perspectiva: a de transformar a tela em um filtro visual que imita a visão humana. No entanto, como o autor concluirá, as impressões estéticas são bem distintas entre uma mídia e outra: enquanto no cinema a câmera objetiva costuma ser empregada no sentido de causar a sensação de estranhamento não-identificação, no videogame o efeito será exatamente o inverso: em geral, a visão em primeira pessoa será fonte de sensação de controle e identificação com a perspectiva que nos é fornecida. Galloway atribuirá essa diferença justamente ao aspecto ativo dos videogames, ou seja, ao fato de controlarmos o movimento (vagueamento) da nossa visão. Enquanto isso, no cinema, o estranhamento seria causado pelo contraste entre uma perspectiva alinhada com nosso olhar, mas que não se movimenta segundo nossa vontade: não estamos no controle.

Figura 11 - Cenário de Minecraft sob uma perspectiva aérea.



A versão acima possui instalação de mods (modificações amadoras) para aumentar a qualidade das texturas e diversificar os biomas do ambiente.

A perspectiva em *Minecraft* é vagueante justamente porque é necessário vagar pelo cenário, percorrer as coordenadas de sua grade tridimensional de modo a captar o maior volume possível de informação. Em outra conjuntura, poderíamos intalar em *Minecraft* uma modificação que nos permitisse ver o cenário de cima, como em um panorama aéreo, por meio do qual saberíamos onde há terra e onde há mar, onde há montanhas e onde estas dão lugar a florestas. Tenderíamos, portanto, ao outro extremo: a *onipresença*. O mundo de jogo curvaria-se cada vez mais aos nossos poderes de associar informações e agir de modo a evitar o desequilíbrio. Por princípio, a perspectiva do jogador que precisa vaguear para compreender o ambiente inverte a tensão: o jogo esconde mais do que mostra, e o encolhimento do mapa informacional/sensorial intensifica a incerteza de quem joga.

Mas isto é ainda mais verdadeiro se nos atentarmos para uma segunda categoria-mestra da tipologia: a de “Ambiente”, que pode ser classificada simplesmente como estático ou dinâmico (AARSETH et al., 2003, p. 50). Ambientes estáticos, como o do jogo *The Witness* (2016), só se movimentam quando o jogador ativa alavancas ou abre portas: nenhum personagem, máquina ou objeto se move sem que ativemos um comando que provoque este movimento. No mesmo jogo, a navegação se dá, assim como em *Minecraft*, por uma perspectiva vagueante. No entanto, a ausência de movimento espontâneo altera qualitativamente a experiência: se voltarmos ao mesmo lugar que exploramos meia hora atrás, as coisas estarão do modo como as deixamos. Em *Minecraft* e seu ambiente dinâmico, se sairmos do abrigo que construímos e deixarmos a porta aberta, podemos, mais tarde, nos deparar com invasores. Principalmente se o dia der lugar à noite, quando as criaturas mais hostis perambulam pelo mundo.

Minecraft, a princípio, caracteriza-se por ser um jogo de *perspectiva vagueante em ambiente dinâmico*. Essa classificação nos ajuda a entender as tensões únicas que o acoplamento da perspectiva do jogador à dinâmica do jogo pode gerar. É também este acoplamento que nos permite avançar ainda mais em direção à descentralização do jogador e de seus aspectos sensoriais: a perspectiva relativamente limitada restringe o acesso a informações e estímulos, requerendo movimento, exploração. Enquanto isso, a dinâmica do ambiente exige que o jogador lide não só com coordenadas e entidades estáticas, mas com processos que se desencadeiam independentemente de sua presença vagueante.

No entanto, ainda há questões não respondidas pelas classificações de Aarseth et al. Comparemos *Minecraft* a outro jogo que, segundo a tipologia, também se enquadraria nos mesmos quadrantes: *Shenmue* (1999). O jogo não é estático: se nos mantivermos imóveis, direcionando nossa perspectiva a uma determinada rua ou casa, poderemos perceber personagens indo e vindo: o mundo está em movimento. Caso simplesmente chamássemos seu ambiente de “dinâmico”, estaríamos ignorando uma diferença crucial *Shenmue* e *Minecraft*: por mais que o primeiro seja dinâmico, sua movimentação contínua atinge um estágio de equilíbrio inexistente no segundo:

Quando jogos como *Shenmue* são abandonados, eles tendem a entrar em estado de equilíbrio. Não em *loop* ou em um ritmo marcado, mas em um estado de descanso. O jogo está lentamente andando no mesmo lugar, movendo-se de um lado a outro e voltando ao centro. Ele está rodando, jogando a si mesmo, talvez. O jogo está em um estado de ambiência, um *ato de ambiência*. Nem todos os jogos possuem essa função, mas quando possuem, podem existir em um ato de ambiência indefinidamente. Nenhum estímulo proveniente do jogo perturbará o personagem-jogador²⁰ (GALLOWAY, 2006, p. 10).

A conclusão evidente a que se pode chegar é que diferentes jogos possuirão diferentes dinâmicas: algumas mais intensas, outras menos. Não se trata, portanto, de apontar quais jogos são dinâmicos ou estáticos: não estamos diante de um binarismo simples. Ao contrário, trata-se de espectros de intensidades dinâmicas que podem variar de jogo pra jogo, e até serem combinadas e atuarem simultaneamente. A diferença entre *Shenmue* e *Minecraft* não pode ser decifrada pela tipologia de Aarseth et al. porque os autores partem do espaço como referência: o ambiente é dinâmico quando uma ou mais entidades (que não sejam o jogador) mudam de coordenada ou alteram o estado do sistema. A questão é: como essas entidades fazem isso? Essa

²⁰ Livre tradução de: “When games like *Shenmue* are left alone, they often settle into a moment of equilibrium. Not a tape loop, or a skipped groove, but a state of rest. The game is slowly walking in place, shifting from side to side and back again to the center. It is running, playing itself, perhaps. The game is in an ambient state, an ambience act. Not all games have this action, but when they do, they can exist in an ambience act indefinitely. No significant stimulus from the game environment will disturb the player character.”

questão só pode ser respondida quando paramos de submeter à espacialidade uma função essencial do jogo: o tempo.

1.4 Vagueamento e Dinâmica(s): a dimensão temporal da conexão jogador-jogo

De fato, a mesma tipologia de Aarseth et al. elenca, longo depois da categoria “Espaço”, a segunda categoria-mestra “Tempo”. Três subcategorias dividem-se dela: “Ritmo”, “Representação” e “Teleologia”. Reservemos a primeira para uma discussão posterior; enquanto isso podemos imediatamente descartar a segunda de nosso escopo, pois procura definir se o tempo de um determinado jogo imita (tempo “mimético”) ou não (tempo “arbitrário”) o tempo do “mundo real” (p. 51). Como já demonstramos, os videogames certamente possuem uma camada representacional. Mas, do ponto de vista do mundo de jogo, essa camada é irrelevante ou, pelo menos, secundária. As forças que compõem um mundo de jogo – as velocidades das entidades, suas colisões, transformações e oscilações, etc. – podem ser compreendidas em si mesmas²¹, sem que subjuguemos seu estudo àquilo que supostamente procuram “mimetizar”. Em outras palavras, é possível falar de *Pong* sem nos referir ao tênis.

Por sua vez, a subcategoria “teleologia” nos parece quase inócua para descrever o tipo de experiência lúdica que encontramos em *Minecraft*, ou mesmo *Shenmue*. Sua função é determinar se o jogo possui (finito) ou não (infinito) um final, ou seja, algo que decreta o encerramento da experiência e ofereça um objetivo final a ser perseguido. Foi exatamente o que *Minecraft* ganhou com seu lançamento oficial em 18 de novembro de 2011 (embora a esta altura o jogo já fosse um fenômeno mundial com dezenas de milhões de cópias vendidas). Um inimigo chamado de “Ender Dragon” foi adicionado ao mundo de jogo e que, talvez até de forma jocosa, agia como uma espécie de objetivo final em um jogo anteriormente desprovido de qualquer meta pré-determinada. O jogador que achar e matar o Ender Dragon “vence” *Minecraft*, como se a criatura fosse o último chefe. O monstro não muda nada no restante do jogo: as ferramentas, os biomas, animais e materiais continuam exercendo e possuindo as mesmas propriedades. O que leva à pergunta: podemos afirmar que, apenas pela adição de um único inimigo, *Minecraft* deixa de ser um jogo “infinito” para se tornar “finito”?

²¹ Novamente, cabe frisar que ainda podemos chamar estas entidades de “personagens”, “cavernas”, “pedra” e “carvão” sem submeter sua atuação no mundo de jogo a uma matriz “real”. No entanto, não podemos confundir a palavra com o que ela designa: o conjunto de propriedades e capacidades que chamamos de “pedra” em *Minecraft* só é uma representação de uma pedra natural até o momento que passamos a entender sua atuação no contexto da dinâmica que surge da sua interação com outros elementos do jogo. Assim, percebemos que o nome “pedra” não passa de uma convenção, e que perderemos de vista muitos de seus atributos caso nos limitemos a descrever de que forma ela representa uma pedra natural.

O problema parece ser justamente o abandono do que Galloway (2006, p. 11) chama de “processo puro”. Aqui, nos cabe perceber o aspecto temporal do jogo não em sua dimensão teleológica: essa dá ao tempo a função de um recipiente estéril, “previamente dividido em instantes de determinadas extensões ou durações” (DELANDA, 2002, p. 84). Em suma, *espacializa-se* o tempo, descrevendo-o em termos de instantes homogêneos e discretos. Embora essa estratégia seja útil para a vida prática, ela não esgota o potencial criativo dos ritmos temporais.

O filósofo Henri Bergson (2006) ilustra essa crítica ao falar da impressão que uma certa melodia nos causa. Uma melodia pode ser decomposta em notas, e é assim que podemos codificá-la em uma partitura, por exemplo. No entanto, a impressão concreta deixada pela audição da melodia – que inclusive relaciona-se à nossa capacidade de reconhecer uma música ou outra – só nos atinge uma vez que estas notas ressoam em *continuidade*. Da mesma forma, os eventos que se desdobram no jogo não podem ser compreendidos plenamente se atribuirmos a eles status independente da continuidade que os faz agir no tempo. Ou seja, a teleologia do jogo finito é apenas uma nota perdida em meio à continuidade de notas do jogo, e não pode ser pensada fora desse contexto.

Sendo assim, no nosso objetivo de pensar a temporalidade dos mundos de jogo, a teleologia perderia espaço para uma descrição da combinação de *ritmos* presentes em cada um deles. Podemos, para tanto, retornar às ideias de lentidão e visibilidade presentes no trabalho de Claus Pias: enquanto as velocidades relativamente baixas da interface produzem ritmos, a visibilidade dá a estes mesmos ritmos aspecto visível em forma de *movimento* (ou, seguindo Galloway, *ações*). Novamente, cabe aqui recorrer ao princípio da continuidade apontado por Bergson: o movimento – no jogo ou em qualquer âmbito – não pode ser reduzido à mera sucessão de estados autocontidos. Uma série de “ações” estáticas, privadas do fio contínuo que as constitui em processo, não nos diz mais de um jogo do que uma série de quadros parados nos apresenta um filme ou uma série de notas solitárias nos remete a uma melodia. Em suma, o movimento do jogo Pong é mais que a soma dos “frames” individuais que nos causam a sensação de movimento da “bola” e das barras.

Para sair do terreno das analogias, podemos fazer uso do estudo de Brigid Mary Costello a respeito do papel do ritmo em *Minecraft* e *Don't Starve* (2013). A autora observa os jogos sob diferentes escalas temporais, percebendo que as ações desempenhadas pelos jogadores são condicionadas por ciclos, fluxos e desvios rítmicos, tanto macro quanto microscópicos:

No nível macro, os ritmos de ambos os jogos são pontuados pelos seus ciclos de dia e noite. *Minecraft* possui ciclos de 10 min. para o dia e 10 min. para a noite, enquanto

Don't Starve possui 4 min., para o dia e 4 min. para a noite. Nos dois casos, os monstros são mais ativos à noite. Isto impacta os tipos de ação performados e, portanto, também alteram o fluxo rítmico de ações dentro do jogo.²² (COSTELLO, 2016, p. 8).

Logo percebemos que não há como separar, na continuidade do jogo, os ciclos temporais das capacidades e propriedades que seu tempo faz operar. O dia de dez minutos de *Minecraft* pode ser cronometrado no relógio, mas também pode ser expressado pelos processos que se desdobram durante esse tempo. Seu conjunto de ritmos, por mais que se estenda pelos mesmos dez minutos da noite, são mais lentos: menos criaturas hostis povoam florestas e montanhas, e o momento é ideal para modificar o terreno e construir a abrigo ou estruturas em geral. À noite, o jogador tende a se confinar em lugares fechados e protegidos e, caso saia destes, estará submetido a ritmos muito mais intensos: esqueletos e zumbis, por exemplo, sairão à sua caça, enquanto a escuridão dificultará a visibilidade e redobrará a necessidade de atenção. Os dez minutos do dia não são os mesmos da noite.

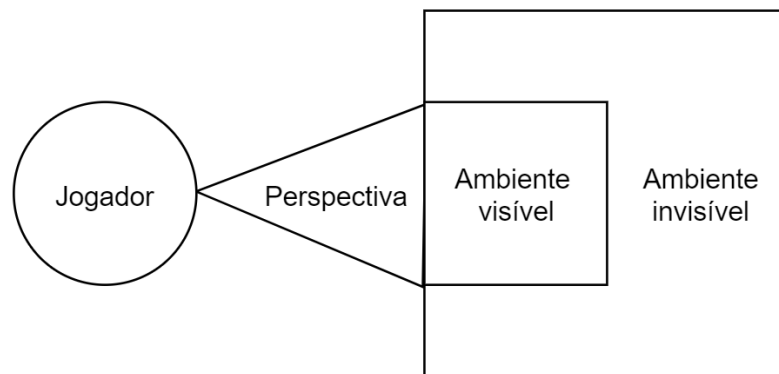
Poderíamos afirmar, porém, que a diferença é apenas uma ilusão: quantitativamente, pode ser medida pela quantidade de ações individuais (x) que ocorrem em um dado intervalo de tempo (t). Se medíssemos a atividade dos inimigos e monstros, por exemplo, e dividíssemos o valor pelos 10 minutos, teríamos um valor (i) que traduziria a intensidade maior ou menor dos processos naquele tempo ($x/t = i$). Nesse caso, as ações seriam unidades homogêneas de medida, indiferenciáveis entre si. Um golpe de picareta em uma pedra possuiria o mesmo valor de uma flecha que atinge o jogador e o mata, já que ambas as ações duram pouco menos de meio segundo ($1 \text{ ação}/0.5s = 2 \text{ unidades de intensidade}$).

O absurdo dessa hipótese pode ser constatado quando essas ações deixam de ser encaradas como fragmentos homogêneos e passam a ser vistas de acordo com o contexto de sua continuidade. Um único golpe de picareta contra uma pedra é uma ação quase totalmente inócua, como uma nota de uma melodia solitária. Já uma flecha que atinge o jogador em combate é como o acorde final de uma grande sinfonia: não pode ser entendido sem que compreendamos a pletora de ritmos que a contextualizam desde o início da “música”: a busca por um item valioso, a percepção de um inimigo a espreita, a preparação para o combate, a tentativa fracassada de golpear o adversário e, finalmente, a flecha da qual o jogador não consegue esquivar.

²² Livre tradução de: “At a macro-level, the rhythms of the two games are punctuated by their day/night cycles. *Minecraft* has a 10-min day and 10-min night and *Don't Starve* has a 4-min day and 4-min night. In both cases, monsters are more prevalent at night. This impacts the types of actions that are performed and hence also impacts the rhythmic flow of activity within the game”.

Esse contexto nada mais é do que o conjunto de propriedades, capacidades e tendências que operam no mundo de jogo (ou em um processo específico), e que não podem ser percebidas ou mesmo imaginadas sem um desenvolvimento contínuo no tempo. Sob essa ótica, tanto a perspectiva quanto o ambiente devem ser entendidos não apenas como um espaço e uma projeção deste espaço, mas como *modos de gerenciamento de capacidades, propriedades e tendências*.

Figura 12 - Modelo extensivo da perspectiva.

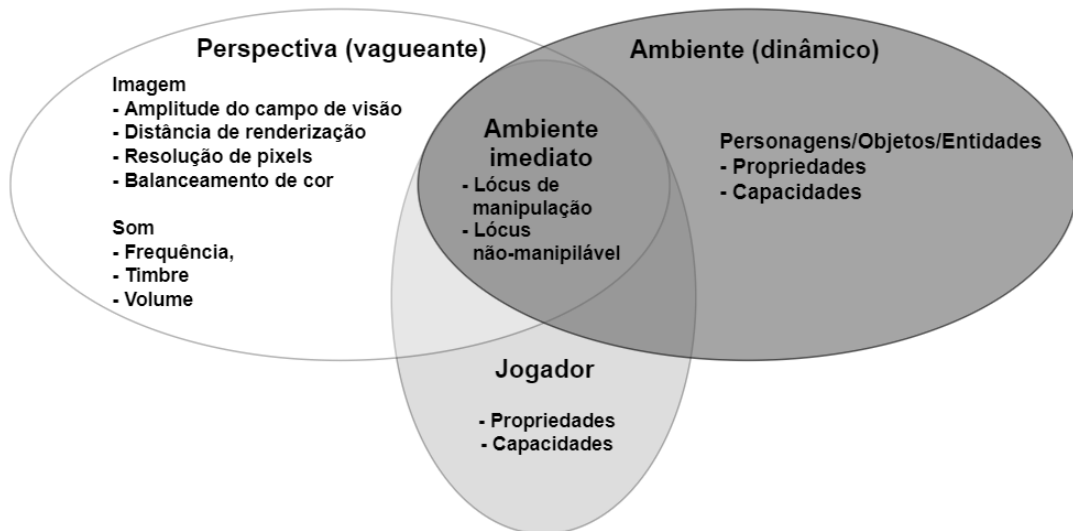


Se antes, sob o prisma do espaço, pensávamos em propriedades extensivas, agora a temporalidade deve ser tratada por meio da indicação das propriedades intensivas dos mundos de jogo. Voltemos a *Shenmue*: espacialmente, poderíamos dizer sua perspectiva se equipara à de *Minecraft*, já que, do ponto de vista extensivo, os dois jogos oferecem uma visão relativamente limitada de um espaço relativamente amplo. Temporalmente, no entanto, a perspectiva não apresenta a mesma dinâmica de ritmos, acarretando em uma diferença qualitativa na nossa necessidade de vaguear pelo ambiente. Por exemplo, se em *Minecraft* o dia inteiro dura dez minutos, o relógio de *Shenmue* leva seis minutos apenas para que uma hora do dia se passe, resultando em ciclos dia-noite de 2h24 minutos.

Em *Shenmue*, quando a noite dá lugar ao dia, a mudança de ritmo é sutil: alguns personagens trancam-se em casa, enquanto outros perambulam no porto da cidade ou se acomodam nos fliperamas de Yokosuka. Tanto os personagens quanto os ambientes são bem mais estáveis que os de *Minecraft*, obedecendo ritmos mais lentos e regulares, desempenhando ações com menos capacidade de perturbar o equilíbrio à sua volta. Além disso, embora o jogo se movimente à despeito do que o jogador esteja fazendo, uma série de eventos-chave que fazem a trama se desenvolver só podem ser disparados caso o jogador esteja no lugar certo, na hora certa. Em suma, estes e outros fatores contribuem para a que o vagueamento do jogador seja

mais estável e menos suscetível a perturbações. A perspectiva em *Shenmue*, portanto, é temporalmente e intensivamente diferente da perspectiva em *Minecraft*, já que processam dinâmicas distintas de maneiras distintas.

Figura 13 - Modelo intensivo da perspectiva e de sua interseção com o jogador e o ambiente



Aqui, seguindo o modelo intensivo de perspectiva (figura 13), percebemos que é possível mergulhar na categoria e descobrir quais parâmetros ajudam a configurá-la como um agente ativo do jogo. O campo de visão, ou seja, a amplitude visual do espaço do jogo aumenta ou diminui o espectro de movimento visível por parte do jogador – o que pode dificultar ou facilitar a localização de um monstro que se aproxima, por exemplo. A distância de renderização²³ possibilita que avistemos marcos na paisagem distante, alterando a capacidade do jogador de se orientar e navegar (para voltar ao abrigo antes da noite cair, por exemplo). Balanço de cor e resolução dão nitidez e contraste a objetos valiosos em uma caverna ou a um buraco que pode ou não ser percebido na escuridão, condicionando a necessidade de que o jogador acenda uma fonte de luz, como uma tocha. Os parâmetros de perspectiva unem-se dinamicamente às propriedades e capacidades do mundo de jogo, tornando qualquer separação entre elas apenas convencional e, na pior das hipóteses, arbitrarias.

²³ Em *Minecraft*, a distância de renderização determina quantas porções (*chunks*) do mundo podem ser vistas por vez pelo jogador. Cada porção possui 16 blocos de largura, por 16 de comprimento e 256 de altura. Quanto maior a distância de renderização, mais porções serão exibidas no monitor e, portanto, mais será exigido das unidades de processamento do computador (processador e placa de vídeo). Uma distância de renderização menor ajuda o jogador a processar o jogo com mais rapidez às custas da perda de exatidão na sua perspectiva. Isso demonstra que, na concretude do processo de jogo, aspectos técnicos e regras se misturam e se influenciam – até porque, principalmente na atualidade, muitos jogos permitem que parâmetros como distância de renderização sejam manipulados em menus pelo usuário.

Sendo assim, no seu desenvolvimento temporal, a perspectiva é mais que um recorte parcial de uma totalidade extensiva: é um modo de *processamento* de propriedades e capacidades em transformação. Ao contrário do modelo espacial anterior (figura 12), a dimensão temporal da perspectiva deve ser definida pelo acesso contínuo do jogador de parâmetros de exibição de estímulos audiovisuais (Perspectiva), que processam momento a momento as propriedades e capacidades do mundo de jogo (Ambiente). A interseção entre o potencial de ação do jogador e as capacidades do ambiente manifestadas pela perspectiva constituem o lócus de ação imediato, bem como as informações e/ou entidades não manipuláveis.

Figura 14 - Mapas



Esquerda: Mapa do mundo de jogo em Minecraft oferece informações a respeito do lócus não-manipulável; Direita: minimapa em Grand Theft Auto V (2013) cumpre o mesmo papel, mas sem constituir um elemento ficcional do jogo.

O sol, como já afirmamos, é uma entidade não manipulável, mas que interfere no processo de jogo: seu movimento indica a passagem do dia e a aproximação da noite – e, com ela, a chegada de monstros. Os mapas são objetos manipuláveis que permitem que o jogador cartografe uma área limitada do mundo total de jogo, mas que, uma vez desenhada, oferece informações referentes a áreas inalcançáveis pela perspectiva momentânea: o jogador expande seu campo de conhecimento do mundo de jogo com a ajuda de informações externas. É por essa razão que Kristine Jørgensen (2013, p. 4) identifica os elementos não-ficcionais como parte integrante do mundo de jogo, assim como Alexander Galloway alega que, embora possamos perceber a existência de ações diegéticas ou adiegéticas nos processos de jogo, “os dois tipos de ação são ontologicamente equivalentes” (GALLOWAY, 2006, p. 5). Do ponto de vista do puro processo, o que importa é como a camada sensorial do jogo processa a dinâmica, resultando em uma perspectiva movente e mutante.

O ato de vaguear – ou mudar de perspectiva – pode levar ao encontro de dinâmicas instáveis que rompem com o domínio do jogador sobre o mundo de jogo, mas pode, também,

ter o efeito inverso: fazer com que o jogador apreenda os ritmos do jogo e crie “uma performance rítmica expressiva” (COSTELLO, 2016, p. 15). O que isso significa, em termos práticos, é que o jogador passa a dominar o que antes era instável e desenvolver técnicas para reduzir sua imprevisibilidade, intensificando sua soberania sobre a dinâmica do jogo.

É neste ponto que podemos apontar a diferença brutal entre *Minecraft* e outros tipos de jogos para um jogador. Na introdução, falamos de *Super Mario Bros.*, mas, aqui, caberia compará-lo a um outro jogo tridimensional e com perspectiva vagueante: *Half-Life 2* (2004). Poderíamos descrever tanto *Minecraft* quanto *Half-Life 2* como jogos onde a perspectiva processa a dinâmica do jogo momento a momento, apresentando novas capacidades que, à primeira vista, parecem desafiadoras, mas, com o andamento da experiência, podem ser apreendidas e aprendidas. Ao pararmos por aí, no entanto, estaríamos ignorando uma diferença crucial entre os dois jogos: o modo como tecem suas dinâmicas ao longo do tempo.

Embora as perseguições, tiroteios e explosões de *Half-Life 2* nos deem a impressão de instabilidade e dinâmica constante, basta que desliguemos o jogo e comecemos a experiência do início para perceber que a mesma sequência de inimigos nos atacará, o mesmo veículo aparecerá em um momento oportuno de nossa fuga e as mesmas explosões serão disparadas nos mesmos lugares. Em uma sequência de jogo ideal, jamais perceberíamos estas repetições: passaríamos pela sequência de eventos de forma fluida, como em um filme. Este ideal, no entanto, possui pouca influência concreta, já que, seja em *Half-Life 2* ou *Minecraft*, a repetição de trechos ou mesmo da totalidade da experiência é parte integrante do processo lúdico.

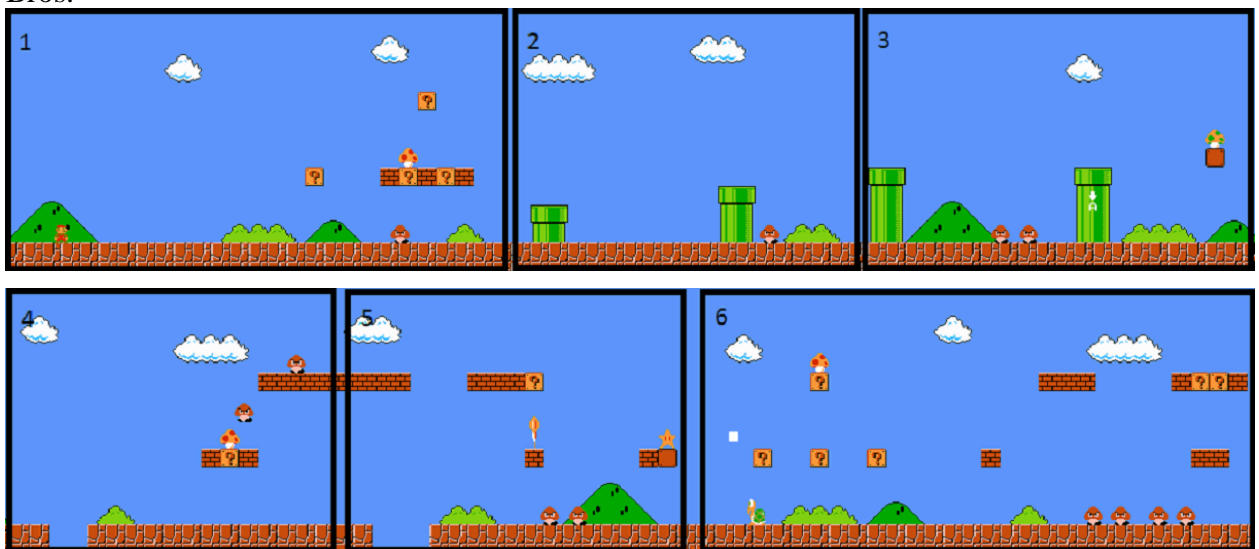
No seu questionamento filosófico a respeito da perda e da falha nos jogos, Jesper Juul, em diversos momentos, deixa transparecer a ideia da repetição, associando-a a uma dimensão ritualística e simbólica (JUUL, 2013, p. 26). Para Walter Benjamin, a repetição constitui a essência da brincadeira infantil (BENJAMIN, 1987, p. 252), uma vontade de recriar a mesma experiência um número indefinido de vezes. Há, no jogo, uma tensão entre passado e presente que, para além de motivações psicológicas, concretiza-se nos videogames por meio dos processos que seus mundos comportam. A repetição enquanto relação do jogo com o jogador, de fato, entrará como alicerce fundamental da experiência lúdica, algo já notado por Johan Huizinga em seu seminal *Homo Ludens*:

Mesmo depois de o jogo ter chegado ao fim, ele permanece como uma criação nova do espírito, um tesouro a ser conservado pela memória. É transmitido, torna-se tradição. Pode ser repetido a qualquer momento, quer seja "jogo infantil" ou jogo de xadrez, ou em períodos determinados, como um mistério. Uma de suas qualidades fundamentais reside nesta capacidade de repetição, que não se aplica apenas ao jogo em geral, mas também à sua estrutura interna. Em quase todas as formas mais elevadas de jogo, os elementos de repetição e de alternância (como no refrain) constituem como que o fio e a tessitura do objeto (HUIZINGA, 2004, p. 11).

Juul, Benjamin e Huizinga procuram raízes e motivos distintos para a repetição, mas o fato é que ela é uma força determinante em quase qualquer jogo e, como tal, não pode ser ignorada. A repetição de *Half-Life* e de *Minecraft* fazem com que suas dinâmicas sejam fundamentalmente diferentes. No primeiro, podemos virar à esquerda em um corredor e nos assustarmos com uma criatura inesperada. A surpresa perderá o efeito assim que morrermos e, ao tentarmos de novo, voltarmos a virar à esquerda e nos depararmos com o mesmo monstro.

A repetição dos processos de um jogo, portanto, é a chave que nos dá entrada a um terreno completamente distinto das dinâmicas de seu mundo. As capacidades e propriedades de um mundo de jogo são arrumadas, muitas vezes, levando em conta as regularidades de experiências repetidas, o que possibilita que os criadores do jogo controlem, até certo ponto, o andamento da experiência.

Figura 15 - A divisão do mundo de jogo segundo as habilidades do jogador em Super Mario Bros.



Embora pareça impossível perceber diretamente estes princípios conectivos, eles são perfeitamente legíveis: no primeiro nível de *Super Mario Bros.*, essa teia conectiva torna-se evidente uma vez que as forças latentes em cada elemento do mundo são identificadas e interrelacionadas. Cada componente se estende por uma determinada medida de espaço mas, sobretudo, impõe condições intensivas para a passagem contínua do jogador e sua perspectiva.

O jogador (figura 15, quadro 1 à esquerda) aparece na margem esquerda da tela, a uma distância razoável de dois focos: um quadrado amarelo com um “?” e uma criatura que se movimenta em sua direção. O contato entre nosso personagem e a criatura causa a morte do primeiro. Em uma eventual segunda tentativa, o jogador se depara com exatamente o mesmo

cenário. Dessa vez, pode tentar fugir no sentido contrário ao que o monstro se movimenta, o que logo se torna impraticável: os limites da tela e o do mundo coincidem. Impossibilitado de fugir, o jogador só conseguirá passar desta etapa se aprender a pular: ou por cima do monstro ou diretamente sobre ele. Uma vez que a função do pulo é dominada, os quadrados amarelos são acessíveis, bem como as plataformas mais altas: além de escapar de monstros, o jogador passa a usar o pulo tanto como ferramenta de locomoção quanto de coleta de utensílios que auxiliam seu progresso.

Adiante, o jogador encontra uma sequência de três canos verdes, posicionados em ordem crescente de altura. Esta ordem não é acidente: sua função é demonstrar ao jogador os níveis alcançáveis pela habilidade de salto que acabou de aprender, e que usará durante todo o jogo. A dificuldade também aumenta junto com o tamanho dos canos: o primeiro é suscedido pelo chão vazio; já o segundo é patrulhado por um monstro. O mundo de jogo cria combinações de propriedades intensivas e extensivas, gerando células de força que se interconectam para formar “frases” lúdicas, assim como letras de um alfabeto se agrupam para criar palavras. Em outros termos, os elementos agem em conjunção, proliferando eventos progressivamente complexos.

Até onde se sabe, jamais essa combinação progressiva e ordenada das regras do jogo havia sido empregada de modo tão cirúrgico quanto no primeiro jogo de Shigeru Miyamoto. A primeira lição que se pode extrair dessa estética é de natureza pedagógica: na meta de ensinar o jogador a pular, a controlar a direção do salto ou a derrotar uma combinação particularmente complicada de inimigos, está pressuposta a provável recorrência de tentativas. Quando morre, o jogador retorna a um ponto anterior, que exhibe a mesma maquinaria de componentes em interação. A conexão do jogador a esta máquina funciona também por meio da insistência no encontro com a mesma composição – mas que admite abordagens distintas. A mesma estética será usada em *Half-Life 2* dezenove anos depois, por exemplo, quando o jogo introduz um novo tipo de inimigo: o jogador sempre avista a criatura em um ambiente controlado para, depois, encontrá-la em cenários cada vez mais ardilosos e desafiadores.

A operação conduzida tanto por *Super Mario Bros.* quanto por *Half-Life 2* consiste em prescrever a progressão da perspectiva do jogador. Continuamente, o jogo pressupõe os ritmos com os quais percorreremos seu mundo, modulando sua dificuldade pouco a pouco, introduzindo novos elementos para, posteriormente, testar nossos conhecimentos de modo cada vez mais sofisticado. O vagueamento da perspectiva é, até certo ponto, previsto pelo funcionamento do jogo. A composição de seu cenário, para lidar com a instabilidade de uma perspectiva que se movimenta com um grau de imprevisibilidade, lapida a atuação de forças no

mundo de jogo. O resultado é uma composição orquestrada em torno das capacidades e propriedades do jogador, um dos modos mais recorrentes de alocá-lo no centro da experiência.

A partir de agora, descreveremos duas propriedades do mundo de jogo que indicam até que ponto o jogador é ou não o centro de sua composição de forças. Uma delas é a instabilidade dos objetos e entidades que ocupam o ambiente: até que ponto essas entidades esperam comandos do jogador para atuarem e se deslocarem? Até que ponto se movem e exercem suas capacidades independentemente de quem joga? Antes disso, porém, nos voltaremos para o segundo parâmetro que aponta o grau de (des)centralização do jogador: a conectividade.

1.5 Conectividade: o jogador implicado no ambiente do jogo

Caso nos propuséssemos a traçar um mapa das conexões possíveis entre os componentes que ocupam o mundo de *Super Mario Bros.*, qual aspecto este mapa teria? Nossa personagem pode pular, colidir, atacar ou ser atacado por todos as entidades que possuem a capacidade de sofrer ou executar estas ações. O avatar que controlamos é, portanto, um nó que atrai ligações de praticamente todas as forças que podem aparecer no jogo: os blocos quebráveis ou não, o solo e os buracos, recompensas e armadilhas, inimigos de todos os tipos: cada um deles pode afetar o jogador e/ou ser afetado por ele de pelo menos uma forma. Mesmo os números na parte superior da tela, que não fazem parte do lócus de manipulação, refletem as ações de ninguém menos que o jogador: quantas vidas possuímos, quantas moedas coletamos e quantos pontos alcançamos.

Selecionemos, portanto, uma outra entidade qualquer do mundo de jogo: o inimigo que se move em nossa direção (figura 15, quadro 1 à direita), por exemplo. Assim como nosso avatar, ele pode se movimentar, pode nos atacar, pode ser sustentado pelo chão e até colidir com superfícies. Portanto, possui algum repertório conectivo em seu arsenal. No entanto, não pode coletar itens, é incapaz de pular para desviar de nossos ataques ou parar de se movimentar. Em nosso mapa imaginário, portanto, o goomba atrai bem menos linhas conectivas que o jogador.

O que chamamos de grau de conectividade, portanto, nada mais é que a capacidade que uma entidade possui de afetar ou ser afetada pelos demais componentes do mundo de jogo. Esse grau, porém, não pode ser totalmente expressado por um número absoluto de conexões: ele só nos diz algo quando o compreendemos em correlação com as outras forças do mundo de jogo. Chegamos, finalmente, à questão das regras: a inevitável discussão pode ser associada, ainda, a um debate acerca de um termo que usamos, até aqui, sem esmiuçar uma definição: *ambiente*.

Tanto no uso de Aarseth et al. (2002) quanto de Galloway (2006), este termo parece ser sinônimo do conjunto de conexões viáveis no mundo de jogo: tanto aquelas que se atualizam em dado momento quanto as adormecidas, em estado potencial. Mais do que qualquer coisa, a definição de Jesper Juul (2005) de que regras especificam “*affordances*” (p. 36) e limitações, nos leva à necessidade de introduzir este termo e o embate filosófico que ele implica.

No livro *The Language of New Media*, Lev Manovich (2001) lista as características essenciais das mídias digitais. Entre elas, está a modularidade, que representa a capacidade dos componentes de um sistema de reorganizarem suas conexões. O alfabeto é um sistema com alto grau de modularidade, já que cada letra pode se conectar com muitas outras para gerar palavras, frases e textos de diferentes estilos. O mesmo vale para as 12 notas musicais. As moléculas de carbono em um pedaço de diamante não possuem grau nulo de modularidade, já que precisam se configurar em um esquema geométrico extremamente específico para gerar a estrutura característica deste mineral. Os blocos de *Minecraft*, em uma comparação preambular com estes outros sistemas, possuiria altíssimo grau de modularidade.

A reorganização de peças modulares pode gerar estruturas mais complexas (uma casa, um túnel ou um circuito de trilhos, por exemplo). Elas são consequência dos potenciais de conexão de cada componente, que pode ser manipulado diretamente pelo jogador²⁴. O potencial de interconexão de peças em um sistema modular não é necessariamente ilimitado: os blocos de madeira em *Minecraft*, conectam-se com os de ferro ou de terra, mas não com os de água ou lava. Não obstante, é possível que nos adaptemos às restrições impostas pelo sistema, de modo a criar uma plataforma que se sustente sobre a água, mesmo sem se conectar a ela. Basta, por exemplo, que a plataforma esteja acoplada a uma outra estrutura fixada no solo.

Figura 16 - Casa construída sobre a água em *Minecraft*.

²⁴ Esta discussão será aprofundada no próximo capítulo, quando retornaremos à questão da emergência, que é exatamente o surgimento de estruturas cujas capacidades e propriedades não estão presentes nos componentes que as formam na escala anterior (neste caso, os blocos individuais que se unem para formar a casa, o túnel e o trilho).



O processo completo envolve uma série de subversões da conectividade do ambiente, descrito aqui: <http://mconstuct.wikia.com/wiki/Waterhouse>

Este exemplo simples de constrangimento circundável oferece um cenário especial para a reflexão a respeito da definição de Juul. Certamente, existem restrições em um sistema como o de *Minecraft*. No entanto, certas restrições são menos rígidas do que se pode pressupor inicialmente, já que podem ser circundadas, por meio do uso do segundo elemento da definição: as *affordances*.

O conceito do psicólogo James J. Gibson (1986) trata justamente do ambiente e do modo como se apresenta ao animal que o habita. Seu ponto de partida, no entanto, trata das condições de possibilidade da percepção visual. O que torna a abordagem de Gibson particular é a relevância que atribui tanto aos atributos perceptivos e físicos de cada animal quanto às propriedades do ambiente capturadas por estes mecanismos. Para Gibson, portanto, o ambiente deixa de ser algo que se constrói previamente ao organismo que o ocupa. Ao mesmo tempo, também não é o inverso disso (algo totalmente construído interiormente por cada indivíduo). É, na realidade, algo no meio do caminho. Um ambiente é a somatória de *relações* possíveis entre entes que compõem um ambiente.

Gibson, divide estes compostos entre meios, superfícies e substâncias. O ar é um meio para a locomoção de seres dotados de mobilidade (animais), bem como para a propagação de luz, som e odor. Uma superfície é a interface de matérias em dois estados diferentes: por exemplo, o chão é uma superfície que existe no encontro entre a matéria em estado sólido (terra, por exemplo) e outra massa de matéria em estado gasoso (o ar). Substâncias são agregados materiais que não permitem a transmissão de luz ou odor, e possuem níveis variáveis de elasticidade, plasticidade, viscosidade, densidade, etc., causando efeitos distintos em um animal – podem ser nutritivas ou tóxicas, por exemplo (GIBSON, 1986, p. 20).

Gibson exhibe um dos exercícios lógicos de que procuram comprovar dinâmica ambígua e variante dos ambientes. Sua trajetória teórica começa diferenciando animais de seres inanimados para, seguidamente, demonstrar uma nova diferenciação, mais específica: entre os animais terrestres e aquáticos. A condição de cada ser muda drasticamente o seu ambiente: para nós a água é substância e o ar meio; para peixes a água é meio, e para anfíbios ambos podem ser meios. Uma *affordance*, na sua definição original é, portanto, nada mais que as capacidades de intervenção oferecidas ao animal, dentro de um determinado cenário de condições perceptivas e ativas:

O que se está discutindo com essas ideias é, de fato, a natureza da relação do animal com seu ambiente. Se quisermos definir essa relação pelo modo de funcionamento de cada componente (ser humano + cidade, por exemplo) é razoável propor que a “natureza” do animal não se encerra em seu próprio corpo. Se o seu entorno redefine suas capacidades de ação, seu funcionamento está diretamente ligado ao que o rodeia. Ou seja, uma ação não surge de um único ator: ela é penetrada pelo nicho que rodeia o animal, por todos os outros agentes que constituem o ambiente.

Não é de se espantar que Jesper Juul nem mesmo tenha incluído em sua definição uma referência a James J. Gibson: a palavra “*affordance*” é usada indiscriminadamente, principalmente no campo dos *game studies*, como sinônimo de “possibilidade de ação”. Por esse motivo, é impossível saber até que ponto o autor subscreve à visão de Gibson e se possui alguma restrição ao seu recorte teórico-epistemológico.

Um dado colateral deste uso, no entanto, é que, uma *affordance* não é algo *do* ambiente ou *do* animal, e sim uma força que "corta através da dicotomia do subjetivo-objetivo e nos ajuda a compreender sua inadequação" (GIBSON, 1986, p. 129). Ora, o mesmo conceito de *affordance*, agora aplicado aos videogames, deveria designar algo pertencente não ao jogo ou ao jogador, mas algo que se desenvolve *entre* eles. Seguir o conceito de *affordance* até o seu limite nos levaria a uma conclusão que provavelmente assustaria Juul: já que regras especificam *affordances*, essa especificação necessariamente deve se concretizar não apenas o mundo de jogo, mas também no jogador – ou ao menos entre os dois. Logo, o jogador está contido nas regras e as regras alcançam e se projetam sobre o jogador.

A afirmação talvez soe evidente se pensarmos que o livro de regras de um jogo de tabuleiro precisa incluir nas suas definições, obviamente, a descrição das capacidades dos jogadores. O que propomos aqui, porém, é levar esta ideia às últimas consequências, afirmando que, na composição do mundo de jogo e da continuidade de seu movimento, podemos vislumbrar a tessitura de um jogador potencial, um jogador *implicado* (AARSETH, 2007).

Como exposto antes, os videogames possuem dimensões autossuficientes (programação, hardware, sistema operacional, dispositivos de input e output, etc.), mas que não se encerram em si mesmos. Para que o mundo de jogo não seja uma célula impenetrável, ele precisa pressupor o jogador por meio de sua composição: a conectividade dos elementos do jogo constrói um jogador-em-potencial:

O jogo abriga expectativas de comportamento dos jogadores, que sustentam-se em uma interface, e que são representadas no jogo por um avatar (este último não em todos os jogos) [...] o jogador implicado possui uma existência concreta e material, já que o jogo não pode se realizar sem que um mecanismo permita inputs por parte do jogador.²⁵ (AARSETH, 2007, p. 132).

Para Espen Aarseth, a estruturação material de um jogo inclui uma construção prévia do comportamento do jogador. No entanto, essa pressuposição ultrapassa a simples determinação de modos de manipulação (ou o lócus de manipulação). Ela se configura, de modo mais elaborado, como uma “limitação da liberdade de movimento e escolha de quem joga”²⁶ (AARSETH, 2007, p. 132). O percurso traçado na primeira fase (e o resto delas) de *Super Mario Bros.*, portanto, implica um jogador que, progressivamente, é moldado pelo jogo. Alguém – na maioria das vezes, um game designer – prescreve uma continuidade de testes, objetivos menores que se aglomeram para gerar objetivos maiores. O jogador implicado é o centro da conectividade: cada bloco, cano, monstro e recompensa é posicionada cuidadosamente para talhar um percurso ideal de um jogador implicado nas regras do jogo.

Um possível argumento contra esta conclusão poderia partir de *Minecraft*: seu ambiente também não pressupõe sempre o jogador? Cada elemento do jogo (excetuando-se alguns poucos, como o sol), pode ser coletado, reposicionado, destruído ou guardado. Se o jogador implicado é uma potência, o jogador em estado virtual, não poderíamos dizer que *Minecraft* também conecta suas peças de forma que este jogador possa executar certas ações? Na verdade, isso não seria válido para qualquer jogo, já que “o jogo não pode se realizar sem que um mecanismo permita inputs por parte do jogador.”?

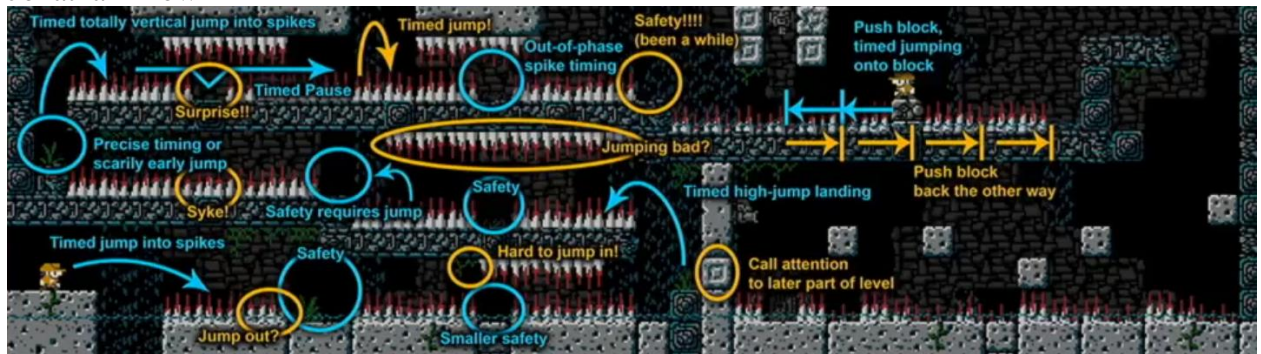
O ponto cego nas indagações acima está no falso pressuposto de que o jogador é a única medida capaz de nos informar a composição de forças do jogo. Certamente, o mecanismo de pressuposição do jogador é necessário para que qualquer videogame funcione em conjunção com um ser humano que joga. No entanto, a análise não pode estacionar neste ponto: se

²⁵ Livre tradução de: The game houses expectations for a player’s behavior, which is supported by an interface, and represented in-game by an avatar (but not the latter in all games). Even more than the implied reader, the implied player has a concrete, material existence, because the game will not be realized unless some mechanism allows player input.

²⁶ Livre tradução de: “a limitation to the playing person’s freedom of movement and choice”.

seguirmos em frente com a investigação, podemos perceber quais jogos limitam-se a trabalhar com a implicação do jogador e quais vão além, implicando outras entidades em sua teia de forças. Adiantando uma conclusão que virá à frente: *Minecraft*, assim como uma série de outros jogos, acrescenta à sua receita uma série de outros ingredientes que, um a um, vão reduzindo a concentração do jogador implicado na mistura de substâncias que dão liga à tecitura do jogo.

Figura 17 - Uma das fases do jogo 1001 Spikes, pontuada com anotações do game designer Jonathan Blow



Fonte: <https://www.youtube.com/watch?v=UwBI7Rnkt78>.

Mas como exatamente essa diferença se materializa na concretude do jogo? Aqui, desviaremos brevemente o foco das teorias dos *game studies* para discutir algumas noções informais de *game design*, tais quais esboçadas pelo programador Jonathan Blow. Em seguida, discutiremos em que sentido podemos conferir rigor epistemológico às suas convicções. Blow, em dois diferentes eventos, apresenta duas noções adversas de *game design*. A figura 17, acima, exemplifica a primeira delas: o jogo *1001 Spikes*, que leva ao extremo a estética de *Super Mario Bros.* já citada neste trabalho.

O jogador começa sua trajetória no canto inferior esquerdo da tela. Sua primeira missão é pular sobre uma superfície povoada por espinhos mortais. Os espinhos, porém, se recolhem sob o solo em períodos regulares, voltando a emergir em poucos segundos. Este é o tempo que o jogador tem para saltar sobre o primeiro buraco. Aterrisando, precisa saltar novamente para escapar dos espinhos que logo o atacam. O primeiro círculo amarelo (“Jump out?”) na figura aponta uma célula de epifania pela qual o jogador precisa passar: o ritmo do jogo abre uma janela de tempo para que o jogador aterrisse na plataforma e siga em frente antes que os espinhos retornem à superfície. Em seguida, o primeiro bloco azul (“Safety”) afrouxa a tensão,

permitindo que o jogador cadencie o próprio ritmo e examine a trajetória aparentemente impossível que vem à frente²⁷.

Seguindo as anotações e descrição de Jonathan Blow, é possível constatar a densidade de “ideias” trazidas pelo mundo de jogo. Cada milímetro de sua extensão bidimensional está comprometido com a montagem de um quebra-cabeça cujas peças não são apenas superfícies, armadilhas e blocos: são as capacidades do jogador. É quase impossível, pelo menos à primeira vista, vislumbrar algum componente da conexão que não esteja posicionado de modo a criar uma zona de tensão ou distensão. O jogador implicado e sua trajetória ideal é o único critério conectivo que dá coerência à construção da fase.

Esta conclusão nos remete à afirmação de Claus Pias tratada anteriormente. Segundo o autor, a interação humano-máquina, a princípio, não posiciona o ser humano em seu centro, mas sim uma interface que humaniza os processos computacionais e interpreta o ser humano como um dispositivo. No entanto, o que acontece quando essa interface – um mundo de jogo, no caso dos videogames – é confeccionada usando como único fio condutor o jogador implicado? O ser humano se reaproxima do centro. O mundo de jogo não é apenas uma concretização do sistema interno, mas um conjunto de conexões variantes improvisando sobre um único tema: o jogador.

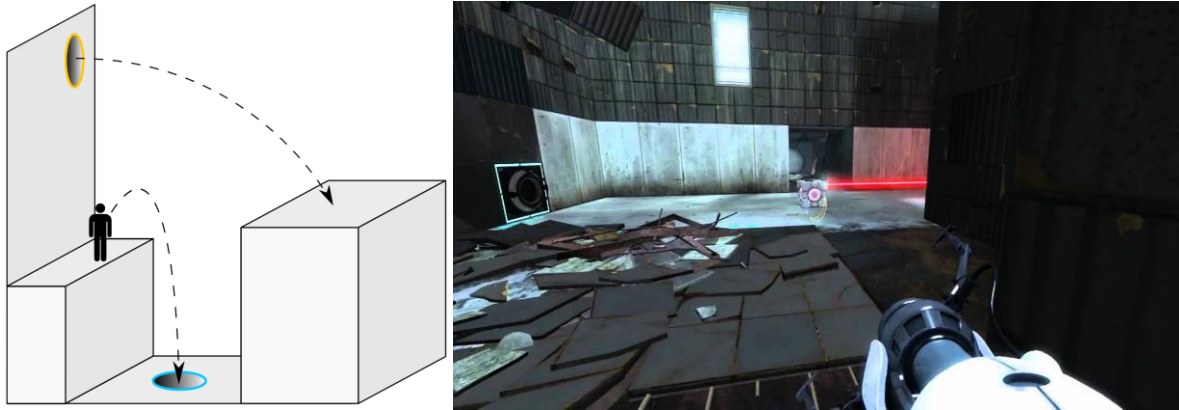
1.5.1 Limitação da conectividade como mecanismo de centralização do jogador

A segunda ideia ensaiada por Jonathan Blow gira em torno da mesma questão: o controle da experiência do jogador por parte dos criadores do jogo. Em uma breve entrevista concedida à revista *PC Gamer*²⁸, o game designer é perguntado sobre o que achou do jogo *Portal 2* (2011). Seu histórico repleto de polêmicas talvez o tenha motivado a responder de modo cuidadoso, embora ainda crítico. Blow enaltece o conceito-chave do jogo: escolher duas superfícies desconectadas (paredes, chão, teto, etc.), posicionar um portal em cada e, com isso, criar uma passagem contínua entre elas. No entanto, para Blow, a aplicação deste mecanismo manifestou-se no jogo de forma discutível: a quantidade de superfícies que admitem essa ação é excessivamente limitada pelos criadores de *Portal 2*.

Figura 18 – Portal - 2

²⁷ O jogo faz uso da conectividade para intensificar o mesmo desafio descrito acima. Primeiro, diminui a altura do teto para impedir saltos mais elevados (“*Hard to jump in!*”). O espaço de segurança que vem a seguir (“*Smaller safety*”) é, portanto, mais difícil de ser alcançado, além de ser menor.

²⁸ Fonte: <http://www.pcgamer.com/jonathan-blow-talks-portal-2-and-player-freedom/>.



À esquerda, conceito-chave de Portal 2: usa-se uma arma para posicionar um buraco laranja e um azul em superfícies brancas, criando uma continuidade espacial entre ambos. Direita: exemplo de limitação imposta pelo jogo – apenas uma pequena porção do espaço permite ser conectada desta maneira.

Essa limitação implica, seguindo esse raciocínio, o estreitamento do campo de possibilidades do jogo. No entanto, Blow não é inocente a ponto de negar que tal estreitamento tenha uma finalidade: direcionar a experiência do jogador. Quanto maior o controle dos criadores sobre as coordenadas espaço-temporais em que certas ações são ou não possíveis, tanto mais centralidade o jogador terá na arquitetura conectiva do jogo. Afinal, o que se está delimitando, de fato, é um campo de ação, nivelado com a precisão necessária para lapidar um ritmo desejável para a performance dos jogadores.

Para Jonathan Blow, o ideal seria tomar o caminho inverso: aproveitar ao máximo o campo de possibilidades do sistema. Tornar a maioria as superfícies do jogo aptas a abrigar portais – quem sabe até a sua totalidade. Novamente, ele entende que as consequências dessa decisão significam um problema de *design* muito mais delicado. Comparando com seus próprios jogos, Blow sintetiza a conversa na seguinte tensão: “eu procuro proporcionar oportunidades para uma experiência, enquanto a Valve [desenvolvedora de Portal 2] busca entregar a você [jogador] uma experiência otimizada”. De modo a evitar conflitos, pontua que são abordagens distintas, mas igualmente válidas.

Tanto no senso comum quanto em setores do mercado, costuma-se afirmar que o funcionamento de um videogame deve ser pensado, condicionado e estudado de acordo com a relação que este estabelece com o jogador. Os *game studies* dificilmente poderiam fugir desse tema. No campo acadêmico, a questão ganha corpo, por exemplo, a partir da ideia de *player-centered game design* (EIMI; MÄYRÄ, 2005). De forma um pouco mais sutil, a ideia de jogo criativo, ou *creative play* (WRIGHT et al., 2002), também prioriza as atividades humanas que fogem do planejado ou projetado pelos criadores do jogo. Desenha-se assim um perfil do jogador como a parte verdadeiramente ativa e responsável pela complexidade do fenômeno do jogo (Cf. SMITH, 2006; p. 30-33). O paradigma da centralização do jogador também persevera

nos *game studies* até hoje com razoável destaque, haja visto o tema do congresso *Foundation of Digital Games em 2017: Celebrating the Player*²⁹.

A resposta a esta perspectiva parece surgir principalmente a partir do trabalho de Ian Bogost, destacando-se o livro *Persuasive Games* (2007). Ao defender o conceito de jogos persuasivos, sustentando-se em videogames com vertentes educacionais, publicitárias e até mesmo políticas, a dimensão ativa parece ser transferida para o lado do sistema. Este abrigaria uma espécie de retórica cujo combustível são as regras das quais os processos se originam. As reações que buscam devolver a centralidade da ação ao jogador (SICART, 2011; VELLA, 2015; STENROS, 2015) voltam a ganhar destaque na contemporaneidade, criticando a desvalorização da criatividade do jogador exercida por visões como a de Bogost.

Se tomarmos como parâmetro a tensão entre o controle da experiência e a abertura do campo de possibilidades, é razoável suspeitar que uma perspectiva de estudo centralizada no jogador realiza um juízo de valor. Ela corta o objeto em pelo menos duas partes, ao mesmo tempo que deposita a maior parte do peso epistemológico em um único prato da balança. Os pressupostos desta perspectiva talvez coincidam com as propriedades de jogos que elaboram experiências, também elas, centradas no jogador. No entanto, perdem sua potência elucidativa uma vez que outros critérios de composição entram, literalmente, em jogo.

Os parâmetros conectivos de *Super Mario Bros.*, *Half-Life 2* e *1001 Spikes* representam uma estética particular, uma série de diretrizes que orientam a teia de relações tramada no mundo de jogo. Essa orientação é, precisamente, o controle da experiência criticado por Jonathan Blow em sua entrevista. Voltando à sua descrição de *1001 Spikes*, é como se o game designer esculpisse uma pedra bruta, composta pela totalidade de ações executáveis em cada ponto do mundo de jogo. A escultura resultante é um mundo que virtualiza uma experiência desejável, mas que atinge seu aspecto graças à remoção impiedosa de pedaços de pedra: cada um uma possibilidade de ação perdida.

Abordar *Minecraft* usando estes mesmos parâmetros como preceitos de julgamento resultaria em uma conclusão inevitável: o jogo não possui qualquer coerência no modo como organiza as conexões que dão aspecto ao seu mundo de jogo. Não há progressão e/ou controle gradativo do entrelaçamento de capacidades e propriedades de cada entidade do jogo. Precisamos, portanto, explorar duas questões relativas a esta aparente falta de coerência: 1) por que ela ocorre?; e 2) qual seria o modelo alternativo que nos permitiria enxergar coerência na conectividade de *Minecraft*?

²⁹ Fonte: <http://fdg2017.org/>.

A questão número 1 pode ser respondida por uma série de propriedades “escondidas” no mundo de jogo. A mais flagrante delas se deve ao método que a programação interna do jogo se utiliza para atualizar o mundo de *Minecraft*. Este método é comumente chamado de geração procedimental de conteúdo (*procedural content generation*), definida como “a criação algorítmica de conteúdo no jogo sob inputs limitados ou indiretos do usuário”³⁰ (TOGELIUS e NELSON, p. 1). Algoritmos, enquanto séries de “descrições exatas de processos que se quer executados” (HOFSTADTER, 1979, p. 298), podem “tercerizar” a criação de “conteúdo”³¹.

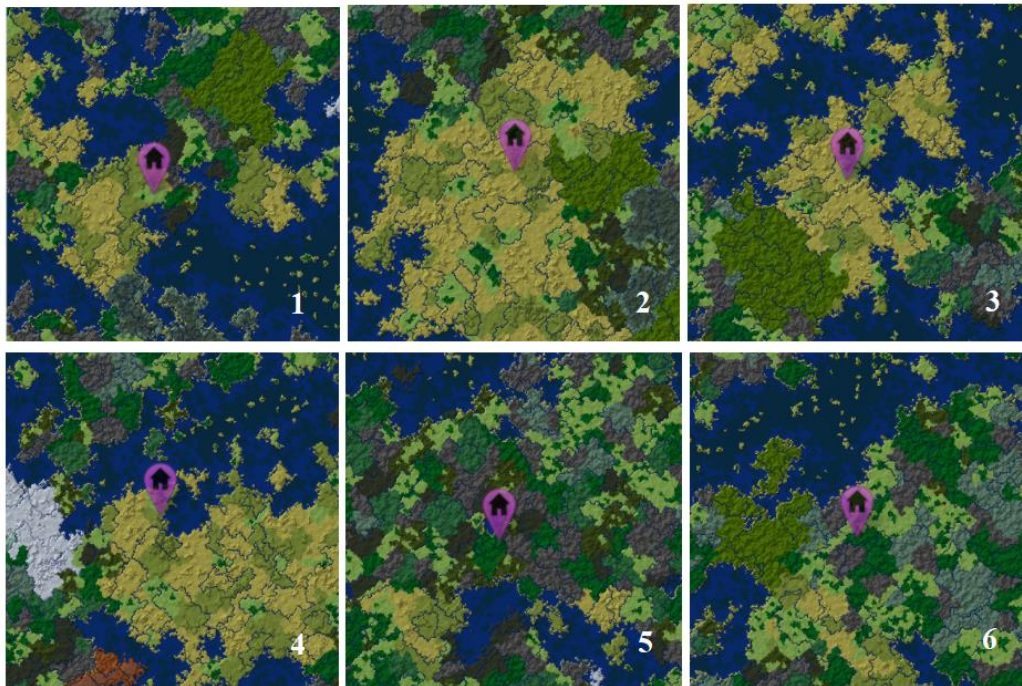
Aqui é importante perceber a diferença de abordagem em relação à que Shigeru Miyamoto empregou em *Super Mario Bros.*, e que vem sendo o padrão da camada superior da indústria desde então. Se na figura 17 há um cuidado cirúrgico com o posicionamento de cada entidade, uma minuciosidade quase artesanal, a geração procedimental é exatamente o oposto: algoritmos se responsabilizam por posicionar blocos, inimigos, paredes, buracos, armadilhas, entre outros componentes do mundo de jogo. No segundo capítulo, descreveremos uma série de jogos que se aproveitam (em diferentes níveis) deste tipo de técnica para gerar mundos onde o jogador se distancia de seu centro. Por enquanto, porém, cabe apenas mostrar as suas consequências em *Minecraft*.

No lugar da abordagem “um lugar para cada coisa e cada coisa em seu lugar”, a geração procedimental do mundo de jogo de *Minecraft* insere uma outra forma de conexão: “lugares do tipo x para coisas do tipo y ; e coisas do tipo z em lugares do tipo w ”. Quando iniciamos *Minecraft*, podemos escolher a opção de gerar um novo mundo: nesse momento, as conexões são traçadas. E, em vez de seguir um encaixe rígido, elas seguem bifurcações: podemos nos ver diante de uma floresta ou de um lago; uma área montanhosa ou uma planície, etc. Isso não quer dizer que as conexões sejam traçadas aleatoriamente – afinal, coisas y não podem ficar em lugares w e lugares x não podem acomodar coisas z . Em termos claros: um lago não pode ser gerado e cima de uma montanha e uma montanha não pode aparecer no meio do oceano. Mas, dentro do campo de possibilidades que resta, qualquer combinação de conexões é possível.

Figura 19 - Mapas de *Minecraft* gerados segundo seus algoritmos, na versão 1.8.

³⁰ Livre tradução de: “algorithmic creation of game content with limited or indirect user input”.

³¹ Aqui, a palavra “conteúdo” é utilizada entre aspas para evitar que, se existe um conteúdo, deve haver uma forma que o veicula. Tal qual concluído por Kristine Jørgensen, no mundo de jogo, forma e conteúdo se misturam de modo indissociável. Podemos entendê-la, aqui, como sinônimo de “dados” que são “processados” pela programação (CRAWFORD, 1987). Nesse sentido, conteúdo é qualquer coisa que se manifeste sensorialmente, enquanto a ideia de processo implica a mudança, movimento ou oscilação desse conteúdo ao longo do tempo.



Os localizadores roxos indicam de onde o jogador começa o jogo. O mapa representa biomas, oceanos, elevações, etc.

Finalmente, podemos abordar a segunda questão elencada anteriormente: como perceber coerência na forma como as conexões são tecidas em *Minecraft*? O terceiro capítulo deste trabalho se concentrará na resposta desta mesma pergunta, focando nos jogos *Dwarf Fortress* e *Ultima Ratio Regum*. Uma descrição breve, porém, poderia projetar um modelo de *Minecraft* da seguinte forma: no nível mais básico, cada um dos mapas acima é composto por diferentes tipos de bloco: areia, terra, água, pedra, etc. Conjuntos de blocos conectados geram entidades de ordem superior: montanhas são uniões de blocos de pedra, carvão, ferro entre outros; cavernas são compostos por estes mesmos elementos, mas situam-se no interior de montanhas ou abaixo do solo (que por sua vez é um conjunto de blocos de terra, pedra ou areia que oferecem sustentação).

Estas entidades de escala superior possuem capacidades emergentes, ou seja, que não estavam presentes nos blocos individuais. Uma montanha oferece a capacidade de escalar, o solo oferece a capacidade de caminhar e correr, uma caverna oferece a capacidade de se abrigar. Biomas podem ser tratados como elementos de terceira escala: são conjuntos de cavernas, montanhas, solos, lagos, vilas, etc. Quando imaginamos todas as formações possíveis dessas escalas, percebemos que o mapa conectivo descentraliza o jogador de duas formas.

Primeiro: cada elemento pode assumir diversas posições, ou seja, não precisam se prender a um percurso ideal centrado no jogador. Uma montanha pode *ou não* esconder veias minerais ricas em diamante. A caverna aberta sob os pés do jogador pode *ou não* estar repleta

de monstros ameaçadores. E, segundo: a determinação do sim ou do não se dá em uma escala que ultrapassa a perspectiva imediata do jogador, consistindo em perguntas referentes a entidades maiores que ele: qual a concentração máxima e mínima de monstros em um bioma? Qual a concentração máxima e mínima de diamantes em um conjunto de x montanhas? Da perspectiva vagueante do jogador, o aparecimento ou não de monstros ou diamantes é quase aleatória. De sua pequenez, é (quase) impossível perceber que a coerência das conexões não está ancorada na sua experiência, mas na edificação de um ambiente instável.

1.6 Instabilidade: as escalas de incerteza dos videogames

O grau de conectividade de um ambiente, desta forma, altera o potencial global de ações do sistema e a instabilidade local percebida pelo jogador. Essa instabilidade é inversamente proporcional à amplitude da perspectiva: quanto mais a visão e as capacidades de ação do jogador se estendem, menos imprevisíveis serão as propriedades do mundo de jogo. A concentração de diamantes em montanhas e de zumbis em cavernas, por exemplo, ficaria mais evidente se um mapa nos apontasse informações que nos indicassem o comportamento global do sistema. Gradativamente, nos aproximaríamos da onipresença e voltaríamos, portanto, ao centro do mundo do jogo. No entanto, por mais imensurável que seja o potencial de conexões, todas elas nos seriam acessíveis quase imediatamente.

Figura 20 - Mundo de SimCity: perspectiva onipresente do ambiente dinâmico



Uma contextualização dessas ideias na história dos videogames talvez seja, por ora, mais útil que conceitos e discussões teóricas – estes virão logo a seguir, uma vez que possamos nutrirlos com exemplos concretos. Jogos como *SimCity*, por exemplo, possuem alto grau de conectividade: cada uma de suas entidades pode se encaixar em múltiplas posições, anexando-se a uma variável imensa de vizinhos possíveis, como peças de lego. Esse alto nível de conectividade, porém, contrasta com uma perspectiva com alto teor de onipresença: possuímos não só uma visão panorâmica da cidade que controlamos, como também acesso quase irrestrito a dados (gráficos, números, relatórios, etc.) que nos dão ainda mais informações a respeito das dinâmicas complexas que fluem na simulação.

A era de *SimCity* é reinada em grande parte por jogos que nos colocavam na posição de deuses. O gênero *god game*, por exemplo, floresce principalmente do trabalho de Peter Molyneux. Atualmente conhecido por suas frases exageradas e por superestimar o potencial de seus próprios jogos, em 1989 Molyneux inicia seu legado com *Populous*. O jogador desempenha o papel de uma divindade que interfere na vida de habitantes de uma vila. Munido de bênçãos e maldições, precisa equilibrar o funcionamento da sociedade abaixo de si, brincando com as vidas artificiais simuladas no jogo. Essa também é a era de Sid Meyer, criador de *Civilization* (1991), no qual o jogador escolhe uma civilização (egípcia, romana, etc.), para gerenciar todos os aspectos de seu funcionamento, em busca de domínio total. O período entre final da década de 1980 e início dos anos 1990, aparentemente, é território dos jogos de ambientes dinâmicos com perspectiva onipresente. No entanto, um outro reino começa a ganhar espaço e a erguer seus alicerces.

Em 1987, *Maniac Mansion* dá início à era dourada dos jogos eletrônicos chamados de *graphic adventures* – gênero que quinze anos depois enfrentaria ostracismo quase absoluto. Nas décadas de 1980 e 1990, porém, gozavam do clamor do público e da crítica. Um duelo ligeiramente anacrônico era travado por duas empresas, que disputavam a liderança desse mercado. O casal Ken e Roberta Williams (que mais tarde fudariam a empresa Sierra On-Line), finca sua bandeira em território inexplorado ao lançar o primeiro jogo adventure que apresenta seu mundo por meio de imagens, e não apenas texto (*Mystery Mansion*, 1980). Mas é Ron Gilbert, ao lado de Gary Winnick, que planta a semente para a eventual hegemonia da LucasArts (então Lucasfilm Games) sedimentada na década de 1990.

A essa altura, Gilbert já havia cunhado um termo sem o qual a grande indústria dos videogames não conseguiria sobreviver pelos próximos vinte anos: *cutscene*. Desde *Maniac Mansion*, os quebra-cabeças e enigmas característicos do gênero seriam intercalados por cenas narrativas e expositivas. Sua função, além de canalizar boa parte do alívio cômico das pérolas da LucasArts, era impulsionar o jogador em direção ao próximo evento do jogo. Durante o próximo desafio, boa parte dos cenários e itens seriam reutilizados, mas de forma bem demarcada e explicada, como um palco de teatro sendo reconfigurado para a próxima cena. O jogador, portanto, era presenteado com um avanço perceptível e irreversível. Um caminho reto intercalado por problemas estáticos, que só se moviam quando eram resolvidos ou quando o jogo fosse desligado.

Em 1989, já com o prenúncio de sua decadência à espreita, a Sierra On-Line ignora os sinais dos tempos e lança *The Colonel's Bequest*, jogo que funciona de maneira oposta a *Maniac Mansion*. Tinha semelhanças com o rival: passava-se em uma mansão e também contava com uma trama dividida entre o suspense e a comédia (embora certamente mais sutis que a do jogo da LucasArts). As semelhanças, porém, limitam-se à temática. Quando o jogador assume o controle da protagonista Laura Bow, o tempo do jogo começa a urgir: dadas interações entre os personagens computam mais quinze minutos no tempo ficcional do jogo. Um relógio aparece no alto da tela a cada quarto de hora “gasto” explorando os aposentos, perambulando pelos corredores ou papeando com os vários membros da família Dijon.

A marcha dos ponteiros, embora esteja atada ao progresso do jogador³², não é ancorada ao seu entendimento e/ou solução dos enigmas escondidos pelo jogo. O mote da narrativa é a investigação do assassinato do patriarca da família Dijon, morto em sua própria casa. O tempo passa, eventos-chave acontecem, novos personagens são assassinados e o criminoso continua a

³² A cada ação executada pelo jogador – seja conversar com um personagem ou entrar em outro cômodo – o jogo computa mais 15 minutos no tempo interno do mundo do jogo.

solta. No final do jogo, o jogador precisa ter conectado os pontos certos e descoberto os segredos guardados pela mansão (passagens secretas, jóias escondidas, um tesouro ancestral) ou pelos personagens (é possível abordá-los diretamente ou espioná-los para descobrir informações sigilosas). Caso não consiga, no entanto, o jogo oferece um final “incompleto”, sugerindo uma possível segunda tentativa de se aprofundar nos meandros do mistério.

The Colonel's Bequest é uma máquina. O jogador dá a partida, movimenta a manivela que faz o motor girar. Dentro da máquina, porém, existem conexões que produzem resultados fugidios, difíceis de captar à primeira vista. O jogo se rebela da sua suposta força motriz: o jogador. Embora dependa dela para funcionar, reserva detalhes e minúcias em lugares improváveis, por meio de movimentos dissimulados.

Figura 21 - Maniac Mansion



Um cofre trancado fica escondido no sótão. Essa parte do mundo de jogo só mudará de estado no momento em que o jogador descobrir a combinação que revela o conteúdo oculto. A mesma estabilidade atada ao jogador vale para todos os outros puzzles.

Em *Maniac Mansion*, o mundo só se movimenta quando a solução para cada puzzle³³ é encontrada, geralmente em condições estáveis. O foco está na estabilidade. Em *The Colonel's Bequest*, o movimento do mundo é o próprio puzzle: as informações disponíveis em um dado momento estático da narrativa (documentos, diários, um corpo caído atrás da porta) são apenas parciais. Os desafios centrais – Quem é o serial killer? Quais são suas motivações? – vão se elucidando apenas à medida que o relógio anda e as engrenagens se movem. O foco está na instabilidade.

A instabilidade, portanto, pode ser definida como o grau de independência entre o percurso ativo do jogador e a atividade das demais entidades do mundo do jogo. Há de se

³³ Os *puzzles*, ou quebra-cabeças, geralmente consistem em achar certos itens – uma chave, por exemplo – e descobrir como combiná-los e ou utilizá-los no contexto correto (a chave pode servir para uma única

notar que *The Colonel's Bequest*, embora possua um grau elevado de instabilidade se comparado a *Maniac Mansion*, ainda possui focos de estabilidade. Primeiro porque o relógio só se move quando executamos ações: em *Minecraft*, pelo contrário, o sol se movimenta e o dia vira noite mesmo que permaneçamos estacionados no mesmo lugar sem pausar o jogo. Em *Dark Souls*, mesmo com o jogo pausado, certos personagens ainda podem se movimentar e nos atacar. Novamente, o grau não pode ser descrito em números absolutos, mas relativos: é necessário que observemos cada jogo dentro de seu contexto geral e, se possível, comparando-os a outros.

Uma segunda nuance digna de destaque é que o grau de independência pode funcionar em uma série de escalas. *Minecraft* possui uma série de focos de instabilidade na escala microscópica: um creeper – monstro verde que explode ao se aproximar do jogador – pode aparecer sem aviso prévio e destruir construções e/ou matar o jogador, fazendo-o perder parte de seus itens. Nessa escala, há ainda outros inimigos e eventos que instauram incerteza: zumbis, esqueletos, lava, explosivos, entre outras entidades que guardam o potencial de instaurar caos imediato no menor sinal de descuido.

No entanto, existem outras escalas possíveis, que ultrapassam o nível do lócus de ação imediato e o alcance da perspectiva do jogador. Em *Minecraft*, a instabilidade nessa escala limita-se à imprevisibilidade das formas e, em certa medida, das propriedades e capacidades dos biomas. Podemos nos deparar com uma montanha estranhamente alta ou uma rede de cavernas bifurcantes aparentemente sem fundo. A montanha nos oferece a capacidade de ver o mundo de uma perspectiva aérea antes impossível; enquanto a mega-caverna nos oferece a chance de entender, ainda que infimamente, o caráter labiríntico de cavernas reais.

Se compararmos *Minecraft* a *Dwarf Fortress*, é possível perceber ainda que a instabilidade de elementos de escala superior não precisa se limitar às suas formas e propriedades inertes: essas entidades podem se movimentar e agir no mundo. Assim como em *Minecraft*, podemos jogar *Dwarf Fortress* comandando um único personagem, abandonado em meio a selvas, desertos, redes de cavernas subterrâneas, vilas, cidades, acampamentos militares e caravanas mercantis. A variedade de entidades de primeira escala, no entanto, não esgota o mundo de jogo em *Dwarf Fortress*.

Se no meio de nossa jornada encontrarmos um elfo, este personagem (primeira escala) pode pertencer a uma gangue de bruxos fora da lei (segunda escala). Esta gangue, como entidade composta, possui seu próprio programa de ações a curto e longo prazo, interagindo com outras entidades de segunda escala, como vilas, exércitos ou caravanas mercantis. Aqui, podemos perceber que há duas modalidades de instabilidade: a *imprevisibilidade conectiva* e o

movimento autônomo. A montanha estranhamente alta e a caverna aparentemente infinita só podem aparecer porque a conectividade do mundo é de alto grau e seus resultados, portanto, são mais imprevisíveis. Esta imprevisibilidade conectiva pode ser acrescentada de movimento autônomo, como os grupos de bruxos fora-da-lei de *Dwarf Fortress*, que podem migrar de um acampamento e invadir uma cidade, por exemplo.

Minecraft possui imprevisibilidade e movimento na primeira e segunda escala. Na primeira, a conexão surpreende na formação de oceanos, cavernas, etc.; na segunda, nos biomas e na configuração global do mundo de jogo (como mostrado na figura 19). Embora possua imprevisibilidade de formas na segunda escala, possui pouco ou nenhum movimento autônomo: não vemos aglomerações de inimigos ou outros tipos de entidade se formarem e agirem enquanto conjunto de segunda ordem.

Dwarf Fortress, em contrapartida, possui ambas propriedades em ambas escalas. *Maniac Mansion* não possui praticamente nenhuma imprevisibilidade conectiva nem tampouco movimento autônomo. Se o jogarmos e rejogarmos, perceberemos que as entidades permanecem idênticas – tanto visualmente quanto dinamicamente – quanto estáticas. *The Colonel's Bequest* apresenta intensa atividade autônoma dos personagens, mas previsível do ponto de vista conectivo: se estudarmos o comportamento dos habitantes da mansão, nos surpreenderemos inicialmente com a continuidade e intensidade de seu movimento. Mas logo notaremos que o mesmo padrão se repete nas partidas subsequentes. Os personagens têm autonomia para se movimentar e alterar o estado do mundo de jogo, mas não para *reconfigurarem* sua atividade: ela sempre se dará no mesmo ritmo e sob as mesmas condições se religarmos o jogo e recomeçarmos a partida.

Figura 22 - The Colonel's Bequest:



A arma Derringer que fica guardada em uma caixa de vidro no escritório será roubada sempre à primeira badalada após a meia noite.

Por meio destes exemplos, é possível apontar que, ao contrário da tipologia de Aarseth et al., não podemos determinar se um jogo possui ou não possui imprevisibilidade conectiva e/ou movimento autônomo. Não se trata de elaborar caixas de múltipla escolha e marcar um “x” em cada categoria, esperando que, no final, tenhamos uma descrição exata de como cada jogo se comporta dentro de um gênero. Ao contrário: cada jogo apresentará focos de instabilidade, que podem se dar da imprevisibilidade conectiva e/ou no movimento autônomo. Cada uma destas propriedades, por suas vezes, pode ser investigada em profundidade em diferentes jogos: quais as especificidades do movimento e da conectividade em *Minecraft*, se comparadas as de *Dwarf Fortress*? É exatamente o que será feito no segundo capítulo.

Antes de iniciá-lo, no entanto, cabe ainda desenvolver duas últimas explicações. A primeira se refere à relação entre instabilidade e conectividade. Assim como perspectiva, dinâmica e jogador pressupõe uns aos outros, criando uma zona comum, a instabilidade e conectividade de um ambiente de jogo são intimamente associadas. Quando uma conexão se desfaz sem que o jogador tenha provocado tal processo, temos aí um pequeno foco de instabilidade. Em *Minecraft*, isso acontece corriqueiramente, na explosão de um creeper ou no vazamento de um canal de lava, por exemplo. Em outros jogos, como *NetHack* (1987), por exemplo, a instabilidade virá não de conexões desfeitas, mas daquelas criadas: por exemplo, de personagens que perambulam pelas cavernas intermináveis e podem pegar as armas usadas que descartarmos.

No próximo capítulo, veremos que uma vez que o jogo entra em movimento, temos dificuldade de separar estas duas categorias. Mesmo assim, é possível definir que quando se fala em conectividade, estamos apontando o *potencial* de associações entre elementos de um sistema. Além de Lev Manovich, que introduz a ideia de modularidade, podemos recorrer aos expostos anteriormente neste capítulo, como a mediação radical de Richard Grusin e a os níveis de descrição interligados de Douglas Hofstadter. Assim como o primeiro, estes outros dois conceitos procuram apontar a dinâmica das conexões, como o “processo, ação ou evento que apresenta as condições para a emergência de sujeitos e objetos”³⁴ (GRUSIN, 2015, p. 129). Isso quer dizer que, observando um processo em uma dada escala, estamos contemplando, de certa forma, o que as conexões na escala anterior produziram:

Quando um time de jogadores de futebol se agrupa, os jogadores individuais retêm sua separabilidade: eles não se misturam em uma entidade composta, na qual sua individualidade se perde. Ainda assim – e isso é importante – certos processos que operam em seus cérebros são evocados pelo contexto de grupo, e que não

³⁴ “Livres tradução de: “process, action, or event that generates or provides the conditions for the emergence of subjects and objects”.

aconteceriam em outro cenário. Portanto, de certa forma, jogadores mudam de identidade quando se tornam parte de um sistema maior, o time. (HOFSTADTER, 1979, p. 309)

Mesmo substituindo, no trecho acima, “jogadores de futebol” por “blocos de *Minecraft*”, a lógica permanece a mesma: a capacidade global de conexões no sistema é intimamente relacionada aos tipos de instabilidade que seus componentes podem criar. No futebol, falamos de esquemas táticos, estratégias de ataque e defesa, estilos de marcação ou troca de passes. Em *Minecraft*, a organização dos blocos em grande escala gera biomas mais ou menos inóspitos, lugares mais ou menos propícios para se construir uma casa e vias propícias ou não para esquematizar linhas de transporte de recursos. Os tipos de comportamento instável do sistema emergem dos seus esquemas de conectividade³⁵.

1.6.1 Instabilidade como fundamento dos mundos de jogo: *speedrunning* e *spelunking*

A última consequência que essa ideia traz para este trabalho relaciona-se com nossas conclusões a respeito da dimensão autônoma dos mundos de jogo. Estes são padrões de ritmos, movimentos, propriedades e capacidades governadas por tendências. A dinâmica de seu funcionamento implica uma conectividade que proporciona focos de estabilidade perturbados por fluxos de instabilidade. *Todo* jogo, por mais lento e estático que seja, produz movimento, mudança, *processos*. Percebemos que este jogo se dá no nível da perspectiva do jogador ou num nível superior. Nos faltou, porém, observar os níveis mais profundos e fundamentais do videogame.

Voltemos à concepção de mundos de jogo. Quando definimos os definimos como independentes de outras camadas que possam se acoplar a eles – camadas simbólicas e ficcionais, por exemplo – carimbamos entrada em um terreno duvidoso. Qual seria o status de instabilidades que fogem do funcionamento dito normal do sistema? *Bugs, glitches, clipping, hacking, cheating*³⁶: os nomes são muitos, mas todos descrevem situações semelhantes:

³⁵ Em vez de subscrever a um único conceito de um único autor, decidimos reunir as ideias de modularidade, mediação radical e níveis de descrição (entre outras que possuem caráter semelhante) nos termos “conectividade” e “instabilidade”. Certamente, o risco de reduzir a complexidade dos conceitos a um único termo pode soar perigoso; no entanto, acreditamos que o texto ganha em clareza, evitando o uso desnecessário de conceitos amplos demais para situações demasiado restritas. Ainda assim, ao longo dos próximos capítulos, continuaremos a descrever os modos de instabilidade e conectividade nos videogames e, sempre que necessário, destacaremos por meio do uso destes e outros autores, as nuances que os termos “conectividade” e “instabilidade” eventualmente não contemplem.

³⁶ Mía Consalvo (2007) tenta categorizar todos estes termos em uma espécie de tipologia. No entanto, talvez até devido aos motivos elencados nas páginas seguintes, parece haver dificuldade para determinar o que exatamente difere cada um dos fenômenos, já que existem zonas mistas onde eles se confundem (Cf. KRAPP, 2011).

processos de jogo que subvertem o funcionamento esperado do sistema. De certa forma, estes ruídos podem ser pensados como *instabilidades indesejadas*. No entanto, esse pensamento só é aceitável se elegermos, por meio de um juízo de valor, comportamentos *desejáveis* que se oponham a estes ruídos.

Esse julgamento nada mais é do que uma fuga do conceito de mundo de jogo, pelo menos tal qual o montamos ao longo deste capítulo. Há muitos motivos para pressupor um comportamento desejável para eles: *bugs* e *glitches* quebram uma suposta imersão do jogador em um contexto ficcional. Por exemplo, quando um personagem que deveria ser achado em um determinado local não está lá – o jogador “lembra” que não está habitando um mundo fictício, mas sim operando um software que acabou de apresentar um defeito. Nesse caso, o comportamento indesejável é aquele que compromete a sustentação da camada ficcional. Outra possibilidade seria afirmar que a vontade do desenvolvedor é soberana: qualquer fenômeno que ultrapasse os limites imaginados pelo autor do jogo, como no caso de *hacking* ou *cheating*, é condenável e precisa ser coibido.





















A questão é que, ao entender os mundos de jogo como dimensões autônomas e levarmos até o final a essa decisão, não há motivo para traçarmos juízos de valor: todo e qualquer comportamento do circuito de estímulos sensoriais e as sensações de controle que proporcionam são igualmente válidas. E, para marcar ainda mais essa horizontalidade de valor, perceberemos que é impossível diferenciar *bugs* e *glitches* de eventos “normais” do jogo sem recorrer a critérios ficcionais ou intencionais. No nível do mundo de jogo, eles nada mais são que efeitos colaterais das instabilidades microscópicas do sistema, forças invisíveis que escondem propriedades e capacidades secretas.

A prática denominada de *speedrunning* – tentativa de terminar modalidades de jogos no menor tempo possível – talvez seja a que mais alimente o surgimento de modos de jogo que bebem da instabilidade mais fundamental dos videogames. Podemos discernir, dentro desta prática, pelo menos duas etapas onde a instabilidade se desenvolve: a primeira se dá no momento em que o jogador atribui um objetivo auto-imposto à experiência (Cf. PARKER, 2008). Aqui, a tarefa de completar o jogo da forma mais rápida possível é uma intabilização – ou pelo menos um redirecionamento – do(s) objetivo(s) supostamente “oficiais” do jogo. Para auxiliá-lo na tarefa de realizar este novo objetivo, o *speedrunner* adentra a segunda e mais intensa instabilização do jogo: ele diseca o sistema de modo a descobrir técnicas que façam com que o jogador minimize até o limite o tempo de completção do jogo.

Scully-Blaker (2014), identifica duas categorias nas quais essas técnicas se encaixam: primeiro, o jogador precisa *refinar* seu desempenho na execução de procedimentos de jogo já

conhecidos, aperfeiçoando sua destreza e agilidade. Depois, o jogador precisa *desconstruir* o jogo, buscando novos procedimentos que o permitam acessar atalhos e economizar tempo. Esta segunda etapa, muitas vezes, envolve a exploração de falhas do software até que ocorram *glitches* e *bugs* no sistema. Se estudarmos a história da prática de *speedrunning*, veremos ainda que alguns jogadores concentram-se na primeira competência, enquanto outros apresentam mais afinidade com a segunda. Alguns raros jogadores ainda conseguem se sobressair em ambas.

Figura 23 - Lista dos 10 melhores tempos de Super Mario Bros.³⁷ (visualizada em 26/11/2017)

Rank	Player	Time	Platform	Date
1st	 darbian	4m 56s 528ms	 NES	1 month ago
2nd	 Kosmicd12	4m 56s 878ms	 NES	2 months ago
3rd	 somewes	4m 56s 894ms	 NES	6 months ago
4th	 StuckInAPlate	4m 56s 978ms	 NES	3 months ago
5th	 SoloQ	4m 57s 593ms	 NES [Emu]	1 month ago
6th	 mav6771	4m 57s 610ms	 NES	27 days ago
7th	 shroud522	4m 57s 676ms	 NES	5 months ago
8th	 i_o_l	4m 57s 693ms	 NES [Emu]	3 years ago
8th	 andrewg	4m 57s 693ms	 NES	1 year ago
10th	 AD2	4m 57s 710ms	 NES	6 months ago

Talvez o exemplo mais clássico na comunidade de *speedrunners* seja a progressão de quebras de recordes do jogo *Super Mario Bros.* (1985). Até 2006, a partida mais rápida do início ao fim do jogo, havia durado 5 minutos e 5 segundos. O jogador Scott Kessler executou este tempo em uma demonstração impecável de refinamento. Porém, ainda que o tempo tenha permanecido imbatível por um ano, acabou sendo superado pelo jogador conhecido como Andrew G. em 2007, que, por sua vez, ofereceu um perfeito exemplo de desconstrução³⁸.

³⁷ Mesmo tendo sido lançado em 1985, *Super Mario Bros.* ainda proporciona quebra de recordes de velocidade: em 17/03/2016, Andrew G. alcançaria o tempo de 4:57 e, em 20/10/2017, o jogador “darbian” alcançaria o recorde atual, 4:565. Fonte com resultados atualizados: <https://www.speedrun.com/smb1>.

³⁸ Um resumo da história completa da progressão de recordes em *Super Mario Bros.* pode ser encontrada no seguinte vídeo: <https://www.youtube.com/watch?v=RdAkY7RfajY>.

Figura 24 - Cutscene de Super Mario Bros. que pode ser pulada, caso o jogador provoque um glitch no jogo.



Andrew descobriu que, na fase 4-2 do jogo, podia causar um *glitch* que impede o sistema de carregar uma das *cutscenes* da fase: uma cena do personagem Mario emergindo de um buraco, agarrado em um planta. A cena é disparada quando o jogador entra em um dos canos verdes da fase, e não pode ser pulada apertando botões: o *speedrunner* perde alguns preciosos segundos assistindo o pequeno filme, para depois retornar à sua corrida contra o tempo.

Andrew, porém, descobriu que podia “enganar” o software, posicionando-se em um lugar imapeável da tela, o que dava ao sistema uma informação “falsa” de sua posição. Assim, quando entrava no cano de onde saía a planta, o jogo ainda não havia carregado a cena que normalmente exhibe naquele momento, o que não deixava escolha ao sistema: ao invés de exibir a cena, o programa pulava imediatamente para o próximo conjunto de dados da fila. Em suma, caso o jogador execute o truque com êxito, aparecerá na mesma etapa da figura acima, mas sem precisar esperar o fim da *cutscene* para retomar controle do jogo. Aproveitando-se deste glitch, Andrew G. conseguiu diminuir o recorde de 5:05 para incríveis 5:00.

Na incipiência da prática de *speedrunning*, explorações de “defeitos” no sistema, tais quais a descrita acima, eram consideradas “ilegais”, e jogadores que aproveitavam-se deles não tinham seus recordes registrados pela organização amadora Twin Galaxies – então tida como árbitra de referência para questões referentes a jogos competitivos. Logo, porém, a comunidade espontaneamente começou a aceitar esse tipo de desvio, transformando a prática em uma mistura do já presente refinamento com níveis cada vez mais elevados de desconstrução. Se os *game designers* e programadores dominam o funcionamento do sistema de dentro para fora, os *speedrunners*, hoje, são provavelmente os melhores conhecedores do funcionamento do jogo do seu ponto de vista inverso. Ao invés de seguir o caminho dos designers e programadores e manipular o sistema para que exiba gráficos, sons e padrões de controle, estes jogadores exploram essa a visuaudiomotora para instabilizar o sistema que ela reveste.

Deste ponto de vista, a desconstrução operada pelos *speedrunners* é mais do que o ato de se aproveitar de defeitos no sistema: é uma forma de ludibriar os programadores do software, encontrando nele propriedades que nem mesmo seus criadores conheciam. O resultado disso possui, muitas vezes, aspecto agressivo – imagens e pixels se distorcem, sons ininteligíveis são vociferados pelos alto-falantes, etc. No entanto, várias outras vezes, são apenas consequências imprevistas da lógica própria do jogo.

Por exemplo, o jogo *Quake* (1996) utiliza-se da *Quake Engine* – um programa intermediário que facilita a implementação de processos gráficos e sonoros. Engines são capazes de instaurar regras estáveis para um jogo, como simulações de leis físicas e processos mecânicos, como gravidade, colisão, etc. Essa camada de regras de “baixo nível” serve como base para regras mais específicas, possibilitando que uma mesma engine acomode o desenvolvimento de diversos jogos.

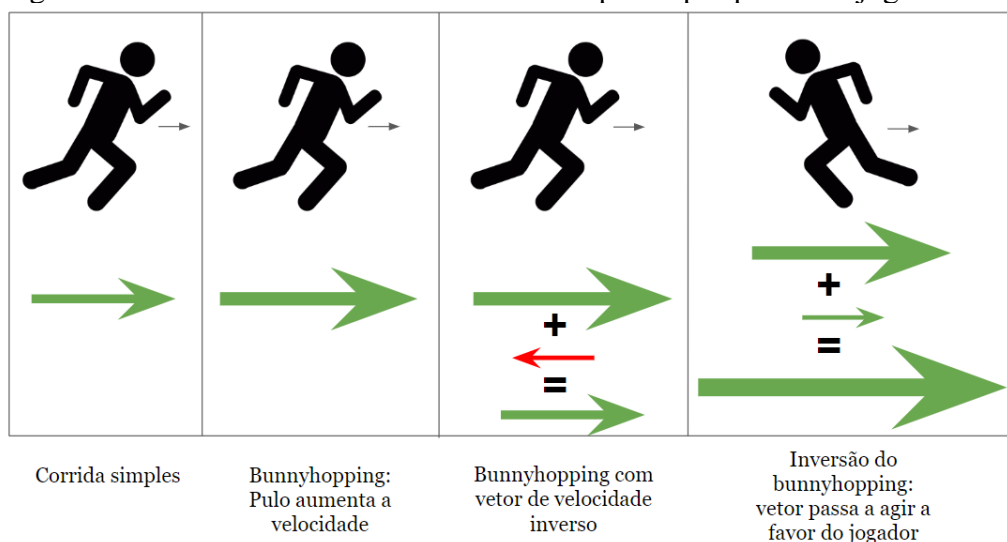
Em *Quake*, logo os jogadores perceberam que se pulassem continuamente enquanto corriam, poderiam aumentar sua velocidade além daquela que alcançavam quando corriam sem pular. Essa estratégia, conhecida como *bunny hopping*, logo tornou-se uma das muitas técnicas de movimento inventadas pela comunidade de *Quake*. A mais famosa delas, o *rocket jumping*, acabou virando sinônimo do que alguns chamam de “emergência” (JUUL, 2002). Dentro da nossa perspectiva, porém, qualquer uma delas é emergente, já que são frutos inéditos do acoplamento de propriedades presentes em uma escala mais simples (DeLANDA, 2009, p. 4).

A mesma *Quake Engine* foi utilizada na produção do jogo *Half-Life* (1998), o que quer dizer que a técnica do *bunny hopping* também migrou em grande parte para este jogo (já que as leis físicas eram produzidas pelo mesmo “motor”). Neste momento, começa uma guerra não-tão-fria entre a Valve (desenvolvedora de *Half Life*) e a comunidade de *speedrunners*. Enquanto os jogadores queriam se utilizar livremente da habilidade, a empresa tentava impedir que a explorassem injustamente – a pesagem da medida entre o justo e injusto, obviamente, ficava a cargo da Valve.

Intencionalmente ou não, os desenvolvedores deixam a disputa ainda mais interessante quando lançam *Half-Life 2* (2004), dessa vez com a sua própria engine, a *Source Engine*. Já que esta possuía leis físicas muito semelhantes às da *Quake Engine*, os *speedrunners* logo aproveitaram-se do *bunny hopping* para aumentar seus recordes. Neste momento, a Valve interfere. Seus programadores criam um mecanismo de coibição do *bunny hopping*, modificando a engine para que, toda vez que um jogador pulasse enquanto corria, uma força contrária empurrasse-os para trás. De certa forma, a Valve mudou a física do universo de *Half-Life 2* para impedir que os jogadores alcançassem velocidades definidas como injustas.

O resultado dessa tentativa, porém, foi o aumento ainda mais intenso dessa velocidade: jogadores perceberam que o jogo supunha que eles necessariamente estariam correndo para frente quando usavam a técnica *bunny hopping*. Pela lógica, o sistema adicionava um vetor contrário para aumentar a velocidade para trás, desacelerando o movimento para frente. Os jogadores, portanto, decidiram tentar correr *de costas*, o que resultou na aceleração ainda maior, já que o efeito do *bunny hopping* foi potencializado pelo vetor de velocidade reversa. A Valve não tentou mais corrigir o erro.

Figura 25 - “Descobrimto” da velocidade superior por parte dos jogadores.



O interessante, se compararmos a estratégia de Andrew G. com a do *bunny hopping*, é que, enquanto uma aparenta ser uma intervenção bruta no funcionamento do sistema, a segunda aproveita-se do funcionamento dito “regular” das engines: nenhuma regra foi quebrada, por assim dizer. A pesquisadora Mia Consalvo identifica o segundo tipo como a prática de *exploiting*, que envolve ganhar vantagem por meio de “ações ou itens ‘achados’, que aceleram ou aprimoram as habilidades de um jogador”³⁹ (CONSALVO, 2007, p. 114). Estas habilidades são “achadas” justamente porque não se trata de interferir no trabalho interno do sistema, mas de descobrir propriedades que os próprios criadores do jogo não descobriram.

Aqui, é oportuno remontar à relação entre o conjunto de regras que rege o jogo e o funcionamento material tanto do sistema quanto do mundo de jogo que este engendra. Gostamos de pensar que as regras de *Super Mario Bros.* são declaráveis por nossa linguagem: “se cairmos no buraco, morremos”; “se alcançarmos o cogumelo, aumentamos de tamanho”. Este é o engano: estas leis existem apenas enquanto traduções convenientes da única dimensão

³⁹ Livre tradução de: “‘found’ actions or items that accelerate or improve a player’s skills, actions, or abilities”.

material assumida pelo videogame: seu sistema, conjunto de estímulos que exhibe, output + input, e finalmente o jogador que a eles se conecta.

No seu nível empírico, o jogo fala a “linguagem” computacional, e o caminho entre esta e a nossa é bem mais sinuoso e ambíguo do que se imagina de antemão. Douglas Hofstadter questiona, nessa mesma linha, o ditado de que “computadores só fazem o que mandamos eles fazerem”. O problema desta ideia é que, muitas vezes, não sabemos exatamente o que estamos ordenando. Muitos níveis de descrição estão implicados no funcionamento do computador que roda um videogame: desde o comportamento dos pixels e amplificadores sonoros até as camadas profundas da engine, do código e do hardware. Quanto maior a distância coberta por este percurso, “menos e menos sabe-se com precisão o que exatamente mandamos o computador fazer! Camadas e mais camadas de tradução podem separar a margem dianteira de um programa complexo da sua contraparte operada em linguagem de máquina”⁴⁰ (HOFSTADTER, 1979, p. 312). Em outras palavras, o que chamamos de “defeito”, “bug” ou “glitch” nada mais é do que resultado muitas vezes inevitável da instabilidade inerente aos computadores.

Caso submetêssemos esses fenômenos a um juízo de valor com base nas intenções dos criadores do jogo e/ou na estabilidade da estrutura ficcional, estaríamos perdendo de vista uma série de nuances da conexão sensorial entre jogador e jogo. Sejam estratégias de desconstrução em *speedrunning* ou mecanismos de movimento refinados como o *bunny hopping* – que é apenas um dentre muitos outros descobertos pela comunidade de jogadores de *Quake* – há uma miríade de técnicas e estratégias a serem investigadas. Estas práticas lúdicas nascem, simultaneamente, da apropriação do jogo pelos jogadores e das instabilidades microscópicas que operam no sistema e reverberam no mundo de jogo. Mais importante: caso as ignorássemos, estaríamos perdendo de vista a relação estreita entre essa espécie de prática e o tipo de conexão possibilitada por jogos que descentralizam o jogador.

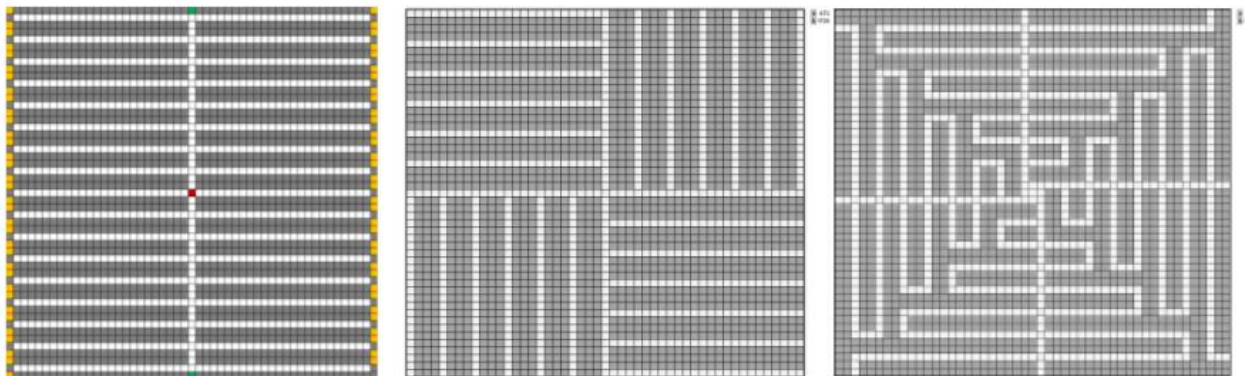
De fato, Shigeru Miyamoto e o resto da equipe que construiu o mundo de *Super Mario Bros.* procuraram domar ao máximo as instabilidades inerentes ao sistema, tentando retê-las em estruturas estáveis. Paredes, buracos e plataformas só possuem propriedades invariáveis graças à essa luta contra a natureza intempestiva dos programas. No entanto, é inevitável que forças instáveis consigam escapar em momentos e lugares específicos. É explorando essas faíscas de instabilidade quase invisíveis que um *speedrunner* descobre que pode causar um incêndio se alimentá-las da maneira certa. Uma vez descobertas, regras como “se cairmos em buraco,

⁴⁰ Livre tradução de: “you know less and less precisely what you've told the computer to do! Layers and layers of translation may separate the "front end" of a complex program from the actual machine language instructions”.

morremos” tornam-se muito menos rígidas e gerais, tornando-se flexíveis, relativas e contingentes.

Como não se prende a desenhar o comportamento do jogo em função do jogador, *Minecraft* liberta os componentes de seu mundo para conversar com muitos outros, cultivando justamente flexibilidade, relatividade e contingência. Alguns *glitches* e *exploits* inevitavelmente nasceram de sua programação. No entanto, se levarmos adiante a ideia de que um *exploit* nada mais é que uma estratégia “achada” pelos jogadores, muitas delas operam dentro de condições de “normalidade”. Um dos exemplos mais corriqueiros é a prática de *spelunking*, desenhada para extrair a maior quantidade possível de recursos em uma caverna.

Figura 26 - Três das técnicas mais simples de otimização de mineração em minecraft.



Fonte: <http://www.minecraftforum.net/forums/minecraft-java-edition/discussion/185947-diamond-mining>

Embora a exploração dessas estruturas seja apenas uma das muitas atividades possíveis em *Minecraft*, existem diversas técnicas e estratégias inventadas por jogadores para otimizar o processo, tornando-o mais rápido e seguro. Assim como na técnica de movimento em *Quake* e *Half-Life 2*, estes métodos são aperfeiçoados com o tempo e precisam mudar de acordo com a evolução de *updates* lançados para *Minecraft*. Na figura acima, a primeira estrutura implica em construir um corredor principal e criar bifurcações laterais perpendiculares. Logo, começam a circular outras técnicas, mais complexas, porém mais eficazes, resultando em maior número de recursos raros (como diamante) em menor tempo.

De certa forma, o que os jogadores fazem em *Minecraft*, é tentar contrapor a instabilidade do mundo de jogo com suas próprias técnicas e recursos, de modo a estabilizar o sistema. Isso não exclui, obviamente, a possibilidade de que jogadores calculem mal suas ações e gerem ainda mais instabilidade – ou que, simplesmente, escolham intensificar a instabilidade do mundo de jogo para disseminar o caos. O ritmo do mundo de jogo, suas propriedades,

capacidades e as tendências que direcionam as suas forças são, no nível mais básico, tensões e relaxamentos, momentos estáveis e instáveis, até que o se encerre a conexão jogador jogo.

Neste capítulo, procuramos, primeiro, apresentar uma definição de mundo de jogo que se sustentasse apenas com propriedades materiais, expressadas em forma de estímulos e que, no contato com um jogador, tornam-se propriedades, capacidades e tendências sensíveis. Em seguida, por meio das noções de conectividade e instabilidade, procuramos introduzir os parâmetros que nos permitem diferenciar os jogos que colocam o jogador no centro de suas teias de forças e aqueles que, em contrapartida, descentralizam o jogador. Este processo envolve a atuação mútua e contínua da dinâmica do mundo de jogo – regida justamente por princípios conectivos e ritmos de instabilidade –, uma perspectiva que processa a dinâmica, apresentando-a no tempo e no espaço e, finalmente, um arranjo de capacidades e propriedades à qual o ser humano se acopla e adapta-se, corporificando o jogador.

Com essa discussão, está apresentada a base teórico-epistemológica que nos permitirá desenvolver o raciocínio em direção aos modos específicos com que a descentralização do jogador ocorre nos videogames. Sustentamo-nos em aspectos sensoriais e materiais da mídia dos videogames, recorrendo a parte das dinâmicas de *Minecraft* e ao contraste deste jogo com outros. Resta, daqui pra frente, utilizarmos dos mecanismos de análise extraídas até aqui para abordar os modos específicos como os mundos de jogo operam o que estamos chamando de descentralização do jogador.

2 MUNDOS ERRANTES: CONECTIVIDADE E INSTABILIDADE NOS SURVIVAL GAMES, *ROGUELIKES* E *ROGUELITES*

Em abril de 2009, cerca de 8 meses antes do lançamento de *Minecraft*, o programador Zach Barth lançava uma das sementes que fez germinar o fenômeno. *Infiniminer* (2009) nasce como um jogo competitivo, no qual dois times precisam escavar um terreno em busca de pedras preciosas: ao final da disputa, o lado que recolhesse maior valor ganharia a partida. No entanto, assim como um experimento que foge do controle, deixou de ser produto da imaginação e das intenções de seu autor para criar vida própria nas mãos anônimas da internet.

Por um descuido de Barth, o código-fonte de seu jogo vaza apenas um mês depois de seu lançamento, alastrando-se como um incêndio em fóruns e plataformas digitais de compartilhamento. Rapidamente, o que foi planejado como um jogo competitivo delineado por metas claras para cada time torna-se um canteiro de obras com blocos. Assim como *Minecraft*, *Infiniminer* gera procedimentalmente seu espaço tridimensional e este, por sua vez, também é composto por blocos de diferentes tipos. Com o código-fonte em mãos, programadores começam a fazer modificações nos algoritmos do jogo, aumentando a proporção deste ou daquele tipo de bloco, remendando sua conectividade e, em alguns casos, engendrando novas propriedades e capacidades para os compostos básicos do mundo do jogo.

Figura 27 - *Infiniminer* (2009): o espaço tridimensional dividido em blocos se assemelha ao de *Minecraft*.



Embora não tenha derivado *Minecraft* diretamente do código-fonte de *Infiniminer*, Markus “Notch” Person chegou a citar o jogo como uma de suas principais influências para criar seu próprio projeto⁴¹. Zach Barth assistiu ao vazamento acidental de seu código dar fama inesperada ao jogo que viria a ser o segundo título mais vendido da história dos jogos

⁴¹ Fonte: <https://www.wired.com/2013/11/minecraft-book/>.

eletrônicos. Mesmo que o impacto de ver sua ideia se transformar em bilhões de dólares possa ter sido inicialmente indigesto para Zach Barth, após *Minecraft*, o programador conseguiu estabelecer uma empresa de sucesso moderado – a Zachtronics –, lançando uma série de outros jogos de nicho⁴².

Poucos jogos, porém, talvez mereçam tanto a alcunha “de nicho” quanto a segunda referência principal citada por Notch: *Dwarf Fortress*. Outro jogo disponibilizado de modo gratuito (embora não com código aberto) na internet por seus criadores, os irmãos Tarn e Zach Adams. Com uma interface de usuário pouco amigável e “gráficos” exibidos em forma de caracteres ASCII, outras muitas características de *Dwarf Fortress* afugentam jogadores novatos. Aprender sequer os primeiros passos para prosperar no jogo requer um investimento de tempo consideravelmente maior que o de jogos de estratégia mais complexos.

Figura 28 - Dwarf Fortress (2006):



À esquerda, representação gráfica de uma das muitas cidades do jogo. A pequena área selecionada corresponde à imagem da direita, o distrito comercial da cidade.

Como dois jogos de nicho, disponibilizados gratuitamente, podem ter sido as principais sementes responsáveis por dar à luz um dos jogos mais populares e lucrativos da história? Embora a palavra de Notch seja uma fonte importante de informação, não basta que a tomemos como lei: é preciso que apontemos onde exatamente em *Infiniminer* e *Dwarf Fortress* conseguimos identificar os aspectos que, mais tarde, apareceriam (mesmo que metamorfoseados) em *Minecraft*.

Infiniminer é um jogo de ampla conectividade: seus blocos são modulares, coletáveis e reposicionáveis; capacidades que, por si só, já geram uma amplitude inumerável de associações. No entanto, o próprio Notch alega – de forma pouco diplomática – que o jogo é limitado,

⁴² Até o momento da escrita deste artigo, Zach continua programando jogos de nicho investindo em mundos de jogo de alta conectividade e que simulam parâmetros de programação e criação de linhas de montagem e processamento – o último deles foi *Opus Magnum*, de 2017.

principalmente por falta de “variabilidade”⁴³. É possível propor que Notch está falando, em outros termos, de um excesso de estabilidade: no momento em que *Infiniminer* deixa de ser um jogo competitivo para se tornar um laboratório de criação, não há mais nada no jogo que pressione os jogadores a criar conexões. Faltaria a *Infiniminer* forças divergentes dos jogadores que operassem no ambiente e movimentassem os estados do mundo de jogo de modo que, como habitantes de seu ambiente, tendêssemos a responder às suas tensões.

Dwarf Fortress é um exemplo exatamente disso. Embora também ofereça uma gama conectiva ampla, há uma instabilidade das entidades do jogo que põe cada conexão em estados de iminente transformação, a cada segundo. Além de seus gráficos ilegíveis para leigos, o funcionamento do mundo de jogo é de uma dinâmica impiedosa. No “modo fortaleza” (*fortress mode*), o jogador comanda uma pequena colônia de anões que deve construir abrigo, sobreviver às intempestividades da natureza e traçar laços comerciais e sociais com outras cidades, vilas e demais grupamentos sociais. Enquanto o jogador constrói e cuida de sua fortaleza, desastres naturais, surtos de violência na colônia e ataques de milícias estrangeiras podem fazer ruir o esforço de infindáveis horas. No “modo aventura” (*adventure mode*), o jogador controla um único personagem que pode percorrer o mundo acompanhado de companheiros ou solitariamente. No entanto, a mais insignificante das distrações pode acarretar na queda em uma armadilha ou no ataque de um animal selvagem. Em ambos os modos, o jogador iniciante sequer tem tempo de experimentar com a conectividade do ambiente antes que sua instabilidade irrompa abruptamente e impeça seu progresso⁴⁴.

Minecraft atravessa o meio termo que separa esses dois pólos. No seu “modo de sobrevivência” (*survival mode*), o jogador, em uma sessão de alguns minutos, costuma dedicar tempo tanto a conectar/desconectar blocos quanto a lidar com as instabilidades do mundo de jogo. Enquanto conecta os blocos de pedra que formarão sua nova casa, precisa se atentar constantemente à aproximação da noite e, junto dela, a instabilidade despertada pelos monstros. Mesmo quando cava um túnel subterrâneo (conectividade) – tarefa aparentemente monótona – precisa se atentar à possibilidade iminente do surgimento de um canal de água ou de lava (instabilidade). Cavar um túnel em *Infiniminer* gera pouca ou nenhuma instabilidade. Enquanto

⁴³ Fonte: <https://notch.tumblr.com/post/227922045/the-origins-of-minecraft>.

⁴⁴ Um caso concreto pode ser encontrado na série de *gameplays* que utilizamos como contraponto à nossa própria experiência com *Dwarf Fortress*. No vídeo a seguir, o personagem controlado pelo jogador e seus companheiros deparam-se com uma hidra (*hydra*), criatura relativamente perigosa e que, no caso em questão, elimina todos os componentes do grupo. A batalha, embora possa ser de difícil compreensão, ilustra alguns pontos básicos da operação da interface de *Dwarf Fortress* – como o fato de que o jogador precisa selecionar quais das muitas cabeças da hidra ele tentará atacar com sua arma. Fonte: https://www.youtube.com/watch?v=05AWsltT8Ns&list=PLdX0AS_VrFWoTP4yD7fAvkEp0__fLBOi6&index=31 (Episódios 7 e 8)

isso, a mesma tarefa desconectar os blocos de *Dwarf Fortress* para criar túneis é a menor das preocupações do jogador que precisa vigiar a variação climática, o temperamento volúvel dos habitantes da fortaleza e as visitas (bem-vindas ou não) que se aproximam de suas portas.

Longe de explicar o sucesso de *Minecraft*, esse equilíbrio é apenas uma das pistas que nos permite investigar a manifestação de instabilidades e conectividades ao longo da história dos videogames. Essa trajetória precisa ser examinada – caso contrário perderíamos de vista o contexto que desenha as condições para que a ruptura de *Minecraft* se efetive. O objetivo deste capítulo é, portanto, abordar esta tarefa sob dois recortes específicos: o da conectividade e o da instabilidade dos mundos de jogo, responsáveis pelo efeito de descentralização.

Enquanto no primeiro capítulo a contraposição de exemplos com o estudo de caso *Minecraft* serviu de instrumento para a elucidação de conceitos teóricos, nesta segunda etapa a abordagem será inversa. Nos apropriaremos dos conceitos e ideias discutidas anteriormente no sentido de investigar a história dos videogames em busca de mundos de jogo cujas forças divergem do jogador. Esse exercício será executado de modo gradual, seguindo as pistas coletadas no primeiro capítulo e tentando abordá-las de modo empírico a partir da análise de diversos videogames. A seleção destes jogos é calcada nos atributos que identificamos em *Minecraft* e que serão encontrados sob novas formas e registros em diversos outros mundos de jogo.

O primeiro deles é o espaço: abordaremos a questão da *exploração* de mundos abertos, ou seja, ambientes jogáveis que se estendem em múltiplas direções, exigindo movimento e navegação constantes por parte do jogador. Neste momento, a matéria prima que nutrirá a discussão serão os jogos que popularizaram os “mundos abertos”: a série *Grand Theft Auto*, que mais tarde influenciaria outras franquias como *Assassin's Creed* e *Far Cry*. Defenderemos a ideia de que estes mundos começam descentralizando o jogador já que, quanto maior a extensão de seus espaços, menor será o jogador em proporção ao tamanho do mundo. No entanto, imediatamente após apresentarem suas extensões desproporcionais, esses mundos *re-centralizam* o jogador de diversos modos: fornecendo mapas, bússolas entre outros modos de orientação detalhada a respeito de onde está e para onde deve ir (para cumprir missões e objetivos, por exemplo). Além disso, reduzirão drasticamente a amplitude de ação do jogador: certas capacidades só poderão ser exercidas em momentos muito específicos, como missões especiais, por exemplo. Em outras palavras, tais jogos centralizam o jogador ao reprimir a instabilidade e conectividade de seus mundos.

Introduziremos a seguir o problema da *adaptação* como contraponto ao processo de exploração. Enquanto os mundos de jogo mais estáveis e pouco modulares exigem do jogador

o constante *reconhecimento* de suas propriedades e capacidades, os mundos de jogo instáveis e altamente modulares desferirão suas forças contra esse processo de reconhecimento. A necessidade de explorar o ambiente para sobreviver às exigências dos mundos logo abrirá caminho para a necessidade de se adaptar, de construir novos ciclos de ação, de modo a inventar formas mais refinadas de habitar o mundo de jogo. Este processo será descrito a partir da análise de três jogos.

O primeiro, *Don't Starve* (2013) deixará claro a passagem do jogador entre dois regimes de atividade: a primeira mecânica e repetitiva de reconhecer itens fundamentais para sobrevivência; e a segunda, instável e conectiva de compor uma base, automatizar coleta de recursos e transformar sua sobrevivência em um processo cada vez mais autossuficiente. *Subnautica* (2014), em seguida, se mostrará como um jogo com traços muito semelhantes. No entanto, concentraremos a análise não na atividade do jogador, mas do seu mundo de jogo, que se divide em biomas – aglomerados de entidades e forças tanto potenciais como ativas –, áreas que se comunicam e traçam fronteiras ambíguas que se influenciam de modo sutil, porém fundamental para despertar as tensões que movimentam seu mundo. Finalmente, chegaremos a *Factorio* (2013), um jogo que esconde uma miríade de ciclos produtivos que podem ser encadeados e, depois, eles próprios podem se encadear com outros ciclos. Assim, o jogo foge da lógica repetitiva e centralizadora do *game loop* (sequências de tarefas pré-estabelecidas requeridas do jogador), demandando o redirecionamento adaptativo que os mundos descentralizadores tendem a demandar.

Finalmente, escavaremos as profundezas da história dos videogames para resgatar uma de suas tradições de criação de jogos mais curiosas. Se *Dwarf Fortres* está por trás do nascimento de *Minecraft*, por trás do nascimento de *Dwarf Fortress* está todo um gênero de jogos cujas raízes alcançam a década de 1970: os *roguelikes*. Começaremos do ponto que dispara o fenômeno: o jogo *Rogue* (1980), que nasce nos terminais da Universidade da Califórnia e espalha-se como um vírus pelos departamentos de computação de instituições americanas e, não muito depois, do mundo. Embora o gênero *roguelike* tenha surgido daí, *Rogue* possui alguns predecessores cujas propriedades nos ajudam a entender sua própria particularidade. Sobretudo *Dragon Maze*, criação de um programador assíduo do computador Apple II, demonstra o poder descentralizador dos labirintos que mudam de forma cada vez que um novo jogo é iniciado. Após checarmos o passado de *Rogue*, deslocamo-nos para seu futuro: selecionamos alguns *roguelikes* que potencializam tanto a conectividade quanto a instabilidade presente no jogo californiano. Alcançamos os dias atuais, com os *roguelites* – versões “remixadas” dos tradicionais *roguelikes* que só fazem se multiplicar desde o sucesso de

Minecraft e do boom da indústria independente de jogos. E, para fechar, analisamos a influência de *NetHack*, o jogo que está sendo desenvolvido desde 1987 de forma descentralizada e colaborativa, dando ao gênero *roguelike* um de seus representantes mais complexos e influentes.

2.1 Exploração extensiva: a abertura dos mundos de jogo

Desde *Grand Theft Auto (GTA) III* (2001), convencionou-se chamar de “*open world*” (mundo aberto) ou “*sandbox*” jogos cujos espaços estendem-se por todas as direções, deixando a cargo do jogador a escolha de qual seguir. Na famosa série da desenvolvedora Rockstar Games, encarnamos um personagem que, nesses moldes, se vê livre para explorar uma cidade – no caso de *GTA III*, a paródia de Nova York, Liberty City.

Esses mundos podem levar dezenas de horas para serem percorridos em sua totalidade, que atingem dimensões muitas vezes continentais. A última instância da franquia, *GTA V* (2013), passa-se em uma cidade digital cuja área equivale a quase 79 km². O valor é calculado por uma aproximação proporcional entre a unidade métrica tradicional e o tamanho das células que conformam a grade tridimensional programada na engine *RAGE (Rockstar Advanced Game Engine)*. A unidade de medida, assim, serve de parâmetro para mensuração de todas as propriedades extensivas do jogo: comprimento, volume e área de entidades como carros, casas e estradas, por exemplo.

Figura 29 - À esquerda, *GTA III*, e em destaque o minimapa. À direita, mini-mapa de *GTA IV* (2006), com ícones de orientação destacados.



A distribuição destas e outras entidades ao longo destes 79 km² configura uma espécie de mapa do mundo de jogo: um conjunto de pontos que simbolizam cada entidade e que se distribuem em coordenadas estáveis. Depois de *GTA III*, uma tendência estabeleceu certas fórmulas para o “gênero” *open world*: essas entidades – sejam elas um personagem ou um lugar

específico –, quando achadas e ativadas pelo jogador, dão acesso a missões (chamadas de *quests*): escapar vivo de um tiroteio, roubar um determinado carro ou arrombar determinada casa. Os eventos são disparados apenas quando o jogador conversa com o personagem certo ou se locomove até o local onde a ação está programada para acontecer. Um mundo aberto repleto de possibilidades; porém cujos disparadores encontram-se quase sempre imóveis e impotentes – até que o jogador, centro daquele universo, os ative. Esses pontos são convenientemente marcados previamente em mapas com ícones comumente chamados de “*waypoints*” (figura 29).

Embora espaços amplos com múltiplos objetivos existam pelo menos desde *Elite* (1980), a criação de jogos de mundo aberto nunca esteve tão em voga quanto a partir dos últimos anos da década de 2000. As evidências disso são numerosas. A Ubisoft, uma das distribuidoras de videogames mais ativas do mercado, lança em 2007, dois anos antes de *Minecraft*, o primeiro jogo da franquia *Assassin’s Creed* (2007). As duas maiores minas de ouro da empresa, antes disso, eram provavelmente as franquias *Rayman* e *Tom Clancy’s Rainbow Six* – um jogo de plataforma e outro de simulação de esquadrões táticos.

A chegada de *Assassin’s Creed* rompe completamente com o cenário interno da empresa. Inspirando-se largamente na estrutura de *GTA III*, leva os mapas expansivos e as missões coordenadas para o cenário das cruzadas do século XII. É a partir deste ponto que a Ubisoft deixa de ser apenas mais uma distribuidora de sucesso para figurar entre as maiores: *Assassin’s Creed* aparece entre os dez jogos mais vendidos de 2007, na companhia de nove outros títulos, todos das bilionárias Electronic Arts, Activision/Blizzard e Nintendo. Desde então, a empresa começa a lucrar na casa dos bilhões de euros por ano.

Figura 30 - Sistema de “*waypoints*” nas três principais franquias de mundo aberto da Ubisoft:



Assassins’s Creed 2 (2009) à esquerda; Far Cry 3 (2012) no centro e Watchdogs 2 (2016) à direita.

Atualmente, a Ubisoft alimenta-se de três franquias de jogos de mundo aberto: *Assassin’s Creed*, *Far Cry* e *Watch Dogs*. *Watch Dogs* já possui duas instâncias (2014, 2016), enquanto *Far Cry* teve quatro jogos lançados (2008, 2012, 2014, 2016), com um quinto

programado para 2018. A franquia *Assassin's Creed* lançou em 2017 o seu décimo título (excetuando-se spin offs, jogos para celular, entre outras versões que somam mais de vinte títulos distintos). Desde 2007, portanto, seus lançamentos são quase anuais.

Além das três franquias recorrentes, a distribuidora de *Assassin's Creed* apostou em dois jogos baseados na obra do autor Tom Clancy: *The Division* (2016) e *Ghost Recon: Wildlands* (2017); dois jogos esportivos: o jogo de corrida *The Crew* (2014), e *Steep* (2017), um jogo de snowboarding; além de *Grow Up* (2016). No momento da escrita deste texto, os únicos dois jogos já anunciados para distribuição em 2018 são de mundo aberto: *Skull & Bones* e *Starlink: Battle for Atlas*.

Este breve exemplo pode aparentar pouco significativo se visto isoladamente, mas é necessário para que tenhamos uma ideia do contexto ao qual ele se associa. Se sondarmos os catálogos da Electronic Arts ou da Activision-Blizzard, encontraremos também uma maior concentração deste estilo de jogo. Coincidência ou não, essa concentração se intensifica ainda mais após 2009, quando *Minecraft* é lançado – embora o jogo não possua a mesma estrutura de missões estáticas sedimentada por *GTA III*.

Outro fato que denota este paradigma são franquias bem estabelecidas que decidiram “migrar” para a estética do mundo aberto. *Mass Effect*, *Dragon Age*, *Mirror's Edge*, *The Witcher*, *Metal Gear Solid* e *Nier* são nomes cujas instâncias originais apresentam espaços bastante restritos se comparados às suas sequências mais recentes: *Nier: Automata* (2017) *Mass Effect: Andromeda* (2017), *Mirror's Edge: Catalyst* (2016), *The Witcher 3* (2015), *Metal Gear Solid V* (2015) e *Dragon Age: Inquisition* (2015).

Outras empresas tradicionais não migraram franquias estabelecidas, mas criaram propriedades intelectuais inéditas seguindo as estruturas de mundo aberto: Insomniac Games com *Sunset Overdrive* (2014), Bungie com *Destiny* e *Destiny 2* (2014, 2017), Guerilla Games com *Horizon: Zero Dawn* (2017), e Square Enix com *Final Fantasy XV* (2017), entre outros. Não é exagero afirmar que a tendência vem se estabelecendo como a mais forte na faixa superior da pirâmide econômica dos videogames, pelo menos aqueles voltados para a experiência solitária.

Além disso, há uma diferença para esta abordagem emergente para outras que se utilizam de espaços abertos e amplos. *Elite*, exemplo dado anteriormente, dá liberdade para uma série de entidades se locomoverem – naves inimigas ou grupos de mercenários saem de suas rotas, fogem do jogador. A temática de ficção científica serve de apoio para que o jogador enfrente piratas e mercenários que não o “esperam” em um ponto específico da galáxia, mas

podem aparecer de repente, sem que um mapa aponte sua localização prévia⁴⁵. Não obstante, a tendência após o sucesso de jogos como *GTA III* e *Assassin's Creed* é pontuar com precisão os disparadores de instabilidade posicionados em seus mundos de jogo, dando ao jogador o máximo de “conforto” para decidir onde ir e quando ir.

Um efeito colateral desta tendência é uma espécie de regra não-escrita de que a área total dos mundos abertos passa a ser a medida da qualidade e sofisticação dos jogos. Por exemplo, desde *GTA III*, a Rockstar se preocupa em aumentar a área jogável de seus lançamentos subsequentes, chegando a se gabar do fato de que seu último título, *GTA V*, ostentar um espaço mais extenso que a soma das áreas dos seus três jogos anteriores⁴⁶. O que pode passar despercebido é que esse maneirismo de marketing é usado pelo menos desde 1996 com o lançamento de *The Elder Scrolls II: Daggerfall*.

Na sua campanha de marketing para *Daggerfall*, a Bethesda Software, desenvolvedora do jogo, afirmava orgulhosamente que sua extensão territorial era equivalente à da Grã-Bretanha⁴⁷. Em uma rápida comparação, poderíamos equiparar este valor aos 79km² de *GTA V*, o que nos levaria à conclusão de que este é mais de 3 mil vezes menor que *Daggerfall*, um jogo lançado 17 anos antes. Isto nos leva imeditamente a questionar como uma empresa, até então relativamente pequena e desconhecida, poderia construir um mundo de jogo cuja área equivale a 3 mil mapas de *GTA V* – um produto cujo orçamento total foi de 265 milhões de dólares (o maior da história dos videogames).

Os números que alimentam essa discussão tornam-se ainda mais esdrúxulos no momento que introduzimos *No Man's Sky* (2016) à conversa. O estúdio Hello Games, integrado por uma dúzia de desenvolvedores, anuncia em 2013 o jogo que simula um espaço povoado por planetas, luas e satélites, onde os jogadores poderiam levantar vôo em naves e, sem transições ou cortes, viajar entre os corpos celestes. O furor causado pelo trailer que acompanhou o anúncio inicial faz com que a Sony acolha o jogo e auxilie em sua campanha de marketing. Logo o chefe da Hello Games, Sean Murray, passa a aparecer em eventos como a Electronic Gaming Expo (E3)⁴⁸, alegando que o espaço jogável de *No Man's Sky* abrigaria aproximadamente 18 quintilhões de planetas, cada qual com o tamanho equivalente a planetas reais.

⁴⁵ Um exemplo de encontro entre o jogador e um grupo de piratas em *Elite*: <https://youtu.be/5f-82ihQ6oQ?t=28>.

⁴⁶ Fonte: <https://kotaku.com/5958949/rockstar-says-grand-theft-auto-v-is-bigger-than-red-dead-redemption-san-andreas-and-gtaiv-combined>.

⁴⁷ Fonte: <https://elderscrolls.bethesda.net/pt/daggerfall>.

⁴⁸ Fonte: <https://www.youtube.com/watch?v=n6JakCSDHPM>.

A chave para desmontar a pretensão destes números está na dimensão material dos mundos de jogo, ou seja, na forma como o processamento do espaço em uma perspectiva se conecta com a nossa própria capacidade de percepção e ação neste espaço. Do lado da máquina, temos o que chamamos de “ambiente” no primeiro capítulo e que, em seu nível expressivo mais abstrato e primordial, é um “espaço matemático” projetado em uma “grade cartesiana”⁴⁹⁵⁰ (BONNER, 2014, p. 4).

Essa projeção é ancorada, em quase todos os casos, no motor do jogo (*engine*), um software que fornece as bases mais primárias da criação do mundo de jogo. No entanto, assim como todo espaço matemático, essa grade não possui início ou fim definidos, apenas uma série de coordenadas abstratas e arbitrárias, convencionalmente identificadas por números. O número 0, novamente por convenção, é uma espécie de centro da grade. No entanto, a posição $x=0$ não possui nenhuma propriedade especial no que se refere ao funcionamento da simulação. Seria fácil para um programador digitar, no lugar do “0”, o valor 100.000 para o eixo x. Ou mesmo para o valor 100.000.000 ou ainda maiores, como 10^{10} . Imediatamente a engine nos “transportaria” para essas “distâncias longínquas”. Obviamente não há movimento espacial nessas operações, apenas cálculos numéricos triviais para qualquer calculadora – tanto mais para um computador contemporâneo.

O espaço matemático só capacita movimento, portanto, quando existem marcadores que possam expressar, por exemplo, uma distância concreta entre duas entidades distintas, e não apenas entre coordenadas desprovidas de aspecto qualitativo. No caso de uma distância que se percorre de um ponto A a outro ponto B, já possuímos um modelo mais próximo do que jogos como *GTA III* ou *No Man`s Sky* fazem: constroem um espaço amplo pelo qual o jogador pode se locomover. É lógico que, mesmo no mundo de jogo mais simples, não se trata apenas de deslocamento: o jogador pode estar à procura de algo ou de algum lugar específicos, o que já implica na habilidade de *navegar*, ou seja, de perceber como as entidades se distribuem no espaço e, possivelmente, traçar rotas de chegada a B ou, caso necessário, um plano de retorno em direção a A.

Locomoção e navegação, nesse sentido, implicam o preenchimento do espaço matemático cartesiano, já que movimentos dependem de referenciais estáveis e navegação debruça-se sobre as relações geométricas traçadas entre esses referenciais. Aproximamo-nos,

⁴⁹ Essa grade pode possuir dois eixos (x e y) em jogos bidimensionais e três eixos em jogos tridimensionais.

Justamente por se tratar de uma construção matemática operada no sistema, poderíamos adicionar até mesmo um quarto eixo w – é a proposta do jogo *Miegakure* (em desenvolvimento), por exemplo.

⁵⁰ Livre tradução de: “mathematical space and therefore on the Cartesian grid”

assim, ainda mais destes jogos – mas não totalmente. A locomoção no jogo depende de mais que distâncias e relações geométricas. O jogador precisa de funções de controle que o permitam “andar”, ou mesmo tomar posse de carros, motocicletas, helicópteros ou quaisquer veículos que possam cobrir essas distâncias. E, quando chegar ao destino, precisará comprar um arma, salvar um personagem ou roubar um banco. Em outras palavras, o jogador precisará *manipular* ativamente certas propriedades e capacidades do jogo. Para engajar na manipulação de modo eficiente, deverá ainda *reconhecer* as diferentes entidades e ter ao menos um modelo prévio de como estas podem intervir no mundo do jogo.

Se o aumento das dimensões extensivas dos mundos de jogo provoca algum efeito, esse efeito é o aumento diretamente proporcional da necessidade de *exploração*. Precisa-se de mais locomoção para percorrer distâncias maiores; maior necessidade de navegação para traçar relações entre lugares e planejar rotas otimizadas; mais manipulação para empregar recursos nessas tarefas e, finalmente, mais reconhecimento para perceber estes processos aparecendo e reaparecendo continuamente no decorrer do jogo.

Os últimos três parágrafos descrevem, com alguma precisão, o que um espaço cartesiano – de três dimensões, no caso de *GTA III* – pode abrigar, enquanto recipiente onde um game designer coloca personagens, veículos, recompensas e armadilhas. É o que procuramos descrever na dissertação de mestrado (MUSSA, 2014), quando falamos de *exploração de ambientes* nos videogames, uma prática que se desenvolve sobre as bases de quatro habilidades cognitivas: locomoção, navegação, reconhecimento e manipulação. O funcionamento coordenado das quatro capacidades é descrito por Hans Paul Moravec (1988, p. 13-50).

A história de Moravec, por si só, merece ser brevemente mencionada. Como professor adjunto da Carnegie Mellon University, Hans Moravec tornou-se uma das maiores referências entre os pesquisadores de robótica e inteligência artificial. Talvez seu destaque se deva à sua preocupação com algo pouco lembrado pelos nomes alinhados à teoria computacional da mente: a necessidade das máquinas de perceberem o mundo à sua volta. Moravec, no livro *Mind Children*, preocupa-se justamente em elaborar as condições sob as quais máquinas poderiam ver, ouvir e tatear o ambiente ao seu redor. Em 2003, o pesquisador aposta em um empreendimento que almeja levar sua própria teoria ao limite: a empresa Seegrid surge com o objetivo de desenvolver robôs capazes de explorar qualquer ambiente físico, sem intervenção ou supervisão humana⁵¹. Uma vez tendo afirmado em 1988 que estaríamos a 50 anos de

⁵¹Fonte: <https://seegrid.com/blog/3d-perception-makes-the-difference-in-vision-guided-autonomous-robots/>.

presenciarmos os primeiros robôs com inteligência equivalente à nossa, Moravec parece ter decidido que ele próprio deveria ser o responsável por concretizar sua própria profecia.

Até o momento, porém, a empresa parece ter optado por investir em um setor um pouco menos radical: veículos autopilotáveis (*self-driving vehicles*). Na sua página digital, percebe-se o conceito-chave que move o projeto – ou pelo menos o modo como escolheram traduzi-lo para o público: “Os veículos da Seegrid vêem o mundo do mesmo modo que seres humanos: seus olhos capturam e constroem um detalhado mapa em 360 graus do mundo a sua volta”⁵².

Subscrever exclusivamente à visão de Hans Moravec implica uma redução da experiência da espacialidade nos mundos de jogo ao mesmo processo realizado pelos veículos automáticos de empresas como a Seegrid, que já povoam as fábricas e galpões industriais. Locomovem-se porque podem se deslocar pelo ambiente, navegam porque sabem ir e voltar a pontos específicos, reconhecem porque distinguem entre diferentes produtos e manipulam porque podem coletá-los e reposicioná-los. Até que ponto circunscrever a experiência da exploração de ambientes a estas quatro habilidades é algo frutífero?

Sem dúvida, em alguma medida, a exploração de mundos abertos demanda um exercício repetitivo. Como vimos no primeiro capítulo, a repetição é algo corrente e essencial a práticas de jogo; uma força-motriz que se manifesta de diversas formas nos vidoegames. A palavra *grinding*⁵³, por exemplo, designa a repetição excessiva de alguma atividade monótona – como matar várias vezes o mesmo tipo de monstro ou executar a mesma missão múltiplas vezes – com o objetivo de se aproveitar de um sistema de recompensas por essas tarefas. Em jogos como *The Witcher 3: Wild Hunt* (2015), por exemplo, o jogador pode se ver em uma situação de desvantagem para derrotar um inimigo demasiado forte, tendo duas opções: a) repensar a estratégia empregada ou b) investir tempo no *grinding*. Por exemplo, pode matar vários inimigos mais fracos para ganhar pontos de experiência e vender os itens que coletar do seu cadáver na vila mais próxima. Com esse dinheiro, pode comprar armas e equipamentos melhores, diminuindo a dificuldade do desafio original sem, necessariamente, aprimorar sua estratégia ou habilidades de combate no jogo.

Um segundo exemplo é a prática de *achievement hunting*, estudada por Mikael Jakobson (2011). O autor concentra-se em jogadores do console Xbox 360, que compram e jogam jogos

⁵² Fonte: <https://seegrid.com/company/>.

⁵³ A prática do *grinding* costuma vir acompanhada de conotação negativa. É comum em jogos de RPG que habitualmente recompensam ações do jogador com pontos de experiência. Derrotar um inimigo poderoso costuma conceder mais pontos que um inimigo fraco. No entanto, o *grinding* permite que um jogador repita a tarefa mais fácil continuamente e acabe alcançando o mesmo número de pontos de experiência que receberia caso desempenhasse a tarefa mais desafiadora.

não por diversão, mas para aumentar sua coleção de *achievements*⁵⁴. A tediosa tarefa destes caçadores envolve a busca de jogos que ofereçam tarefas fáceis de cumprir e que recompensem o jogador com troféus, pontos e outras recompensas pelas suas “façanhas”. Esse tipo de experiência, quando levada ao extremo, resulta em práticas como a de *gold-farming*, exercida por jogadores pagos para “farmor”⁵⁵ moedas virtuais de jogos como *World of Warcraft* (2004) ou *Runescape* (2001). Nesses jogos, a prática de *grinding* resulta em dinheiro, assim como em *The Witcher 3*. A diferença é que, como são jogos para múltiplos jogadores, as moedas ganhas durante tarefas repetitivas são trocadas por dinheiro real, transformando o jogo dos *gold-farmers* em um trabalho digital dos mais tediosos e estafantes.

Talvez seja redundante frisar, mas algo fica claro nestes três exemplos: cada uma dessas tarefas envolve movimento, navegação, reconhecimento e manipulação e, sem exceção, consistem em atividades tão repetitivas quanto aquelas desempenhadas pelos robôs da Seegrid. No seu nível mais básico, ela pode ser subsumida ao que *game designers* chamam de *game loop*, definido nos *game studies*, por exemplo, como “ações cíclicas” (WOLF, 2001, p. 81).

Quase sempre encorajadas por recompensas – e outras vezes desencorajadas por punições – a lógica do *loop* possibilita ciclos de ações repetitivas e mecânicas, que tem em práticas como o *gold-farming* sua face mais extrema. Existe algum nível em que a experiência do jogador, enquanto explora um ambiente, se diferencia desse ciclo mecânico de repetição de ações?

2.2 Adaptação e espaço intensivo: jogos de sobrevivência e a metamorfose do *game loop*

⁵⁴ *Achievements* costumam ser apresentados em menus adicionais ou mesmo externos ao jogo. Nos consoles da linha Xbox e Playstation, por exemplo, o jogador acompanha uma lista de tarefas específica de cada jogo que possui em sua conta. Os criadores do jogo, por suas vezes, designam em ordem de dificuldade cada um dos *achievements*. Essas tarefas costumam ser objetivos divergentes do jogo principal. No jogo *Uncharted 3*, para o Playstation 3, por exemplo, o jogador precisa apenas derrotar inimigos, solucionar puzzles e ultrapassar obstáculos escaláveis para alcançar o final do jogo. No entanto, um *achievement* designa uma tarefa extra: descobrir todos os “tesouros” do jogo, que são nada mais que objetos escondidos em cantos de difícil acesso no cenário. A tarefa é desnecessária para a finalização do jogo, mas adiciona um objetivo extra que, se alcançado, figura como um troféu no perfil pessoal do jogador na Playstation Network.

⁵⁵ *Gold farmers* são pessoas que jogam videogames online, tais como *World of Warcraft*, realizando tarefas repetitivas no sentido de conseguir moedas virtuais do jogo e, em seguida, vende-las por dinheiro “real”. A prática é antiga nos mundos dos videogames. Tendo sido pauta de uma matéria do New York Times em 2007: <http://cult320.onmason.com/files/2012/08/Video-Games-China-Money-Online-Games-New-York-Times.pdf>.

Para investigar essa possibilidade, é necessário que recorramos ao assunto que estamos tratando – os jogos de mundo aberto – e apontemos onde podem esconder forças que rompem com os ciclos repetitivos do *game loop*. Mesmo no mais repetitivo dos jogos, veremos que nenhum ciclo de loops pode se repetir *exatamente* da mesma forma que a anterior. Cada jogo conta com uma série de propriedades que causarão mudanças, mesmo que mínimas, na conjuntura do mundo de jogo tal qual apresentado à perspectiva do jogador.

Em *GTA III*, por exemplo, o trânsito é composto por carros que agem, até certo ponto, de forma autônoma. Cada veículo possui pequenas variações em uma gama de propriedades simuladas: velocidade, aceleração, sensibilidade à mudança de direção, resposta de frenagem, entre outras. O comportamento geral do trânsito compõe, de certa forma, uma unidade autônoma e dotada de propriedades e capacidades próprias. Mas essa unidade é produto da interação contínua entre entidades em uma escala inferior: os veículos, pedestres, sinais de trânsito e o desenho serpenteante das ruas, avenidas e viadutos. Cada um destes elementos básicos, sobretudo os veículos, também possuem autonomia: aceleram e freiam, ultrapassam ou são ultrapassados, enfim: respondem uns aos outros, gerando localmente a conectividade que dá mobilidade ao trânsito global.

Na escala em que a perspectiva do jogador se situa – a dos elementos básicos – o trânsito parece ser estável, mas, sob o menor sinal de perturbação, efeitos em cadeia são capazes de disseminar o caos: batidas, atropelamentos, quem sabe até explosões. Um exemplo semelhante de elementos de ordem inferior gerando instabilidade um nível acima é um dos capítulos do jogo *Uncharted 3* (2011). Joga-se em um navio em alto mar durante uma tempestade. De dentro do navio, inicialmente parece que estamos lidando com um cenário tradicional de jogos de tiro: inimigos revezam-se em ondas de ataque e precisamos achar cobertura e momentos certos para contra-atacar suas investidas. No entanto, o chão logo começa a tremer e se desnivelar, graças à tormenta que cresce do lado de fora. Essa instabilidade possui um traço distinto do resto do jogo: *Uncharted 3* ostentará os mesmos cenários e os mesmos inimigos caso joguemos o mesmo capítulo diversas vezes. O capítulo do navio, porém, sempre será diferente graças à simulação do movimento da água: assim como o trânsito de *GTA III*, o computador dá autonomia a células de movimento que interferem na ação das suas vizinhas, gerando efeitos em cadeia imprevisíveis. O resultado é um mar que balança em todas as direções, sacudindo o navio cada vez de uma forma ligeiramente distinta: nenhuma experiência será idêntica à outra.

Tanto o trânsito de *GTA V* quanto a água que sustenta o navio onde se passa a sequência em alto mar de *Uncharted 3* constituem fontes de instabilidade. No entanto, essas fontes não são fortes o suficiente para potencializar uma reconfiguração total do ciclo de ações em loop

que opera no jogo. Em *GTA V*, uma mesma missão de perseguição ainda passa pelo mesmo encadeamento de etapas que configuram uma estrutura definida de condições de ação (dentro da qual o trânsito apenas introduz certa variabilidade).

Por exemplo, em uma das missões mais importantes do jogo, o jogador precisa performar uma série de ações prescritas para executar um roubo a uma joalheria⁵⁶. A primeira delas é decidir qual de duas versões do roubo será executada: a primeira, mais agressiva envolve roubar rifles pesados possuídos apenas pela polícia. A segunda, mais tática, requer o roubo de uma van de uma empresa de extermínio de pestes e granadas de gás de uma empresa de pesquisas químicas. Nenhum destes itens podem ser roubados ou se quer vistos no mundo de jogo, a não ser quando a missão é iniciada. Embora o jogador possa escolher a versão do roubo (agressivo ou tático) a ser executado, os passos subsequentes deverão obedecer a uma ordem pré-estabilizada.

Essa estabilização das ações passíveis de execução não se resume às missões, mas alastra-se a praticamente toda a composição do mundo de jogo. *GTA V* se passa em uma cidade visualmente dinâmica, mas ludicamente imóvel. As fábricas de Los Santos nada fabricam e os portos nada importam. O trânsito existe e responde à interação, mas os motoristas estão apenas circulando de modo aleatório, bem como os pedestres. É possível entrar nas lojas e em outros lugares específicos, como a casa dos três personagens jogáveis. Fora isso, porém, praticamente todos os outros prédios não podem ser explorados e sua única função é caracterizar o mundo de jogo visualmente. A cidade é uma reunião de propriedades estáveis pontuada com fontes controladas de instabilidade.

Em seu artigo a respeito dos game loops, Miguel Sicart usa *Minecraft* como exemplo. Para o autor, as primeiras ações que aprendemos no jogo – coletar, guardar, construir, etc. – repetem-se continuamente, consistindo em uma fonte de estabilidade para a experiência. No entanto, essa estabilidade contrasta-se com a formação e composição do mundo de jogo, que por si só é uma aglutinação intensa de instabilidades:

Nesse sentido, o loop principal de *Minecraft* é desenhado para encadear as possibilidades aparentemente infinitas de seu mundo expansivo, apresentando ao jogador um horizonte de ações possíveis, um espaço de possibilidades no qual algumas ações podem ser repetidas e estruturarem a atividade. O loop principal encadeia a liberdade do jogador a ações particulares e desejáveis. Desse encadeamento, a atividade do jogo emerge como experiência lúdica⁵⁷. (SICART, 2015, p. 3)

⁵⁶ Gameplay de apoio: sequência de ações necessária para completar o roubo da joalheria em *GTA V*: <https://www.youtube.com/watch?v=K4n6PS17ko4>

⁵⁷ Livre tradução de: “In this sense, *Minecraft*’s core loop is designed to bound the apparently infinite possibilities of the expanding world presenting to the player to a horizon of possible actions, a possibility space

Aqui, vemos que *Minecraft* não está livre de repetição nem de focos de estabilidade. No entanto, acontece uma inversão da lógica de *GTA V*: ao invés da estabilidade do mundo de jogo ser temperada por focos de instabilidade (imprevisibilidade do trânsito, por exemplo), a relação contrária ocorre. É a conectividade do ambiente que provoca uma instabilidade contínua, mas onde se pode encontrar pontos estáveis: um túnel que precisa ser escavado, uma casa que precisa ser construída, um monstro que precisa ser combatido, entre outras constantes. No caso de *GTA V*, essas constantes são as missões, tais como a do roubo à joalheira, que impõem etapas e instruções fixas para sua conclusão.

O mesmo ocorre com os jogos citados anteriormente: *Daggerfall* e *No Man's Sky*. Cada uma das muitas cidades que compõem o mundo de jogo do primeiro é inserem-se em contextos que, cada um desenha quadros ligeiramente diferentes um do outro. No entanto, essa variabilidade compositiva não é forte o suficiente para provocar uma ruptura com a sequência de ações que define a exploração do jogo. O mesmo vale para as múltiplas cores, tamanhos, faunas, floras e topografias dos planetas de *No Man's Sky*: por mais interessante que seja sua variabilidade visual, ela pouco faz para introduzir variabilidade de agência. Estamos presos no mesmo loop de nos deslocar por um planeta, coletar recursos para munição e combustível e, finalmente, alimentar o tanque de nossa nave para alcançar o próximo planeta – e repetir o processo, seguindo os mesmos parâmetros pré-fixados.

Em *Minecraft*, no entanto, a pluralidade de conexões possíveis admite a concretização dessas tarefas sob ritmos e métodos completamente diversos⁵⁸. Um monstro pode ser derrotado à queima roupa com uma espada, à distância com um arco ou indiretamente com a ativação de um explosivo. Túneis podem ser circulares, retangulares mais ou menos profundos, separarem-se em níveis acessíveis por escadas ou mesmo por elevadores – desde que o jogador encontre

in which some actions that can be performed repeatedly structure the activity. The core game loop of *Minecraft* bounds the freedom of the player to the particular actions that are possible, and desirable, in *Minecraft*. From that bounding, gameplay emerges as a ludic experience.”

⁵⁸ No gameplay de apoio para o modo de sobrevivência que selecionamos para *Minecraft*, por exemplo, isso pode ser constatado concretamente. Nos dois primeiros episódios de sua jornada, o jogador executa o loop “coletar – guardar – construir – lutar – coletar” sob condições simples e precárias, já que está apenas iniciando o jogo sem muitos recursos. Mais tarde, no vídeo “*Survival day 40*”, o jogador está executando o mesmo loop, mas as entidades envolvidas e o contexto formado por elas é totalmente distinto: o jogador possui uma casa que quer proteger de um creeper, precisando atrá-lo para um ponto distante da casa antes que exploda e danifique sua construção. Em outras palavras, o jogador ainda está construindo, batalhando e coletando, mas a complexidade das entidades e suas relações mudaram drasticamente. Fonte: https://www.youtube.com/playlist?list=PLEUGTm6wqVrcYeO7glpyyJnHcL3dxF1_A

formas de implementá-los. A quantidade de capacidades admitidas pela conectividade do ambiente possui força o suficiente para desestabilizar a fixidez do *game loop*⁵⁹.

O que de fato estamos apontando com este exemplo é a necessidade de expandir a definição de exploração para além das quatro habilidades elencadas por Moravec. Em um nível mais fundamental, essas habilidades referem-se a um espaço extensivo, abstrato, que age como um recipiente que abriga uma série de outros elementos. Estes, por suas vezes, são quase totalmente fechados em si mesmos, fazendo pouco para se comunicar com outros elementos que compõem o mundo de jogo. Assim como as tavernas de *Daggerfall* e os planetas de *No Man's Sky*, elementos fechados são praticamente impermeáveis às instabilidades que ocorrem (quando ocorrem) à sua volta.

Quanto mais as instabilidades ganham potência de mudar o aspecto do ciclo de ações nuclear do jogo (cavar – minerar – construir, por exemplo) deixa de se resumir à exploração mecânica de Moravec. Isso pode ser afirmado a partir da habilidade de reconhecimento, talvez a peça-chave que sintetize todas as outras. Para se movimentar, o jogador precisa reconhecer as condições corretas: onde há buracos, onde há obstáculos intransponíveis, etc. Para navegar, igualmente: é preciso reconhecer marcos no terreno e rememorar as relações entre eles. Na manipulação de objetos e entidades, o reconhecimento é igualmente necessário: seja para identificar quais entidades se quer manipular quanto para detectar os efeitos dessa manipulação.

É importante salientar de que não se trata de um manifesto contra jogos de exploração, ou mesmo de um juízo de valor que clame pelo banimento da mesma de todos os processos de jogo. Como mostra a pesquisadora Leticia Perani (2008), as interfaces gráficas – setor onde aparato sensorio-motor e a dinâmica do jogo se encontram – funcionam graças às suas apropriações lúdicas da capacidade de explorar, que “suscitam o conhecimento e domínio posterior de determinados ambientes e ações”, cujo efeito depende da “organização espacial das informações” (PERANI, 2008, p. 84). A exploração é um momento necessário e indispensável da aquisição de habilidades a princípio mecânicas mas que, uma vez dominadas, tendem a perder seu aspecto mecânico para se misturarem ao repertório corporificado da cognição do jogador (Cf. RÉGIS e PERANI, 2010).

⁵⁹ Cabe ressaltar que o *game loop* de *Minecraft* descrito por Sicart é constituído de ações contingentes e não necessárias. Isto porque a ação “construir”, por exemplo, é um processo que pode ser desempenhado por uma conexão de entidades diversas: madeira para contruir portas e baús; metal para contruir trilhos e vagões de transporte; *redstone* para contruir interruptores e armadilhas. A ação de construir é estável, mas as entidades que nutrirão o processo podem variar. Em *GTA V* vemos uma conectividade de ordem similar quando o jogador improvisa objetivos. No entanto, quando se trata de missões como o roubo da joalheria, a conexão de entidades e ações é nitidamente mais rígida.

Para que, ao contrário, o reconhecimento fosse a habilidade *primordial* da experiência lúdica, cada componente do mundo de jogo precisaria possuir funções rígidas e pontuais. Caso contrário, seria impossível que o jogador as reconhecesse. Em outras palavras, o jogo precisaria apresentar-se ao jogador apenas por meio de estímulos, símbolos e eventos que possuíssem sentido prévio igual ou ao menos similar àquilo que o jogador já possui em seu repertório cognitivo.

De fato, Virgínia Kastrup (2007), alegará que a psicologia cognitiva historicamente baseia-se na ideia de reconhecimento para desenvolver leis gerais e imutáveis que regem a percepção e afecção humanas. A autora percorre, por exemplo, a relação entre o associacionismo e gestaltismo: o primeiro dirá que só podemos perceber objetos e eventos efetivamente porque possuímos uma experiência prévia que nos informa. O segundo contratará afirmando que a percepção pode até ser informada por nosso repertório passado, mas que possui suas próprias leis – as ditas Leis da Gestalt. Segundo a autora, porém, o gestaltismo postular leis imutáveis, insere a percepção em um invólucro impermeável à ação do tempo e imune ao surgimento da novidade.

Mesmo o construtivismo, apresentado a partir da epistemologia genética de Jean Piaget (KASTRUP, 2007, p. 97-113) incorrerá no mesmo argumento estrutural que, no gestaltismo, a primazia do reconhecimento (ou reconhecimento): a estipulação de um ponto de equilíbrio para o qual a cognição humana se dirige. Cada experiência de ação, percepção ou aprendizagem consistiriam em impulsos que nos direcionam a um sistema cada vez mais equilibrado e, por conseguinte, fechado a interferências externas.

2.2.1 Don't Starve: a dinâmica básica da adaptação

Como já mencionado anteriormente, uma série de jogos posteriores ao lançamento de *Minecraft* seguem alguns de seus parâmetros e montam um grupo de videogames frequentemente referido como jogos de sobrevivência. Embora seja discutível até que ponto estes constituam um gênero bem definido e consistente, há certos traços inconfundíveis em vários de seus “espécimes”. Por enquanto, porém, basta dizer que a instabilidade dos seus mundos, unidas à sua intensa conectividade, serão a munição necessária para provocar a necessidade de que o jogador rompa com os parâmetros de ação que se encerram em um *game loop*.

Em *Don't Starve* (2013), talvez o exemplo mais enxuto desta abordagem, o jogador é submetido às forças nocivas do ambiente logo que o jogo começa. O avatar sofre de fome que

crece com o tempo, a noite traz frio e monstros perigosos (mais um traço inspirado em *Minecraft* e encontrado em outros jogos). É preciso seguir um loop preciso de ações: perambular pelo ambiente – coletar itens quando encontrá-los – construir ferramentas, manter-se nutrido e manter a fogueira do acampamento acesa.

Se aplicássemos as leis do reconhecimento a *Don't Starve*, encontraríamos o personagem-jogador em meio ao desequilíbrio da fome, frio e desabrigo. Diante de tal caos, resta apenas o reconhecimento – de comida, fogo e de uma base para acampar. O jogador novato sonda os botões da interface e as informações do *heads-up display* (HUD)⁶⁰, reconhecendo os símbolos emblemáticos e colocando-os em uso. Coleta, inicialmente, sementes, cenouras, palha, galhos e pedras. Logo, usa a interface para construir um machado, que o leva a derrubar as primeiras árvores e alimentar sua fogueira, esperar a noite cair e o dia raiar, para voltar a coletar comida e lenha longe dos perigos da escuridão⁶¹.

Caso o jogo se limitasse a este ciclo, a exploração cognitiva seria mais do que suficiente para manter o jogador vivo, e o ponto de equilíbrio seria fixo e único. No entanto, algo impulsiona o jogador a romper com o loop, a tentar descobrir novas formas de sobrevivência, que se distanciam progressivamente da exploração mecânica e de seus resultados necessários. Em outras palavras, deixa-se de buscar soluções para problemas pré existentes (Onde há fogo? Onde há comida?) para inventar problemas que divergem das funções repetitivas do *game loop*. Como automatizar a produção de alimentos? Como aumentar a velocidade de aquisição de combustível para o fogo? Como otimizar a caça?

Tais perguntas só podem ser feitas uma vez que se domina o ciclo repetitivo inicial para, a partir dele, criar um ciclo mais complexo e mais potente. Potente no sentido de seus caminhos possíveis: não mais é possível falar em um problema com solução pré-estabelecida. Ou seja, o reconhecimento – por mais útil que seja nos primeiros momentos de jogo, chega a ser até mesmo um obstáculo na hora de instaurar estratégias de sobrevivência mais arrojadas. É preciso esquecer o uso mais direto e óbvio das sementes (comê-las) e usá-las para plantar árvores que virarão lenha e trigo que será convertido em pão. É frutífero desistir de correr inutilmente atrás do primeiro coelho que se vê pela frente e começar a observar os melhores lugares para se

⁶⁰ O termo vem da ergonomia e faz referência à prática de inserir dados relevantes à operação designada (pilotar um veículo, por exemplo) sem demandar que o operador abaixe a cabeça para ler tais dados. Foi apropriado pela indústria de design de games, e costuma se referir à disposição de dados do jogo na tela: quantidade de vidas, munição, pontuação, etc.

⁶¹ No gameplay de apoio que selecionamos para *Don't Starve*, o jogador enfrenta essas condições adversas já no primeiro dia de sua jornada, aos 13 minutos do vídeo 1: https://youtu.be/sdTuaZAKMI4?list=PL_qnVpobSbOL4uKvgxTcrDgMayY-aAB35&t=824 .

instalar armadilhas. Essas atividades são mais complexas exatamente por romperem com o loop e instabilizar aquilo que antes era petrificado pela repetição.

Virgínia Kastrup contrapõe, nesse sentido, uma dimensão cognitiva alternativa à reconhecimento: a *invenção*. Deve-se evitar, aqui, a interpretação da palavra tal qual é lida pelo senso comum, ou seja, como feitos grandiosos: a invenção da máquina a vapor ou do cinema, por exemplo. Na verdade, qualquer fenômeno cognitivo que perturbe os padrões estabelecidos pelo reconhecimento e pelos modelos estáticos que dele surgem podem ser atribuídos, segundo Kastrup, a uma dimensão inventiva da cognição humana⁶². Ao contrário da reconhecimento, a invenção não é um mecanismo de busca de soluções para impasses estáticos, mas uma função de *problematização*, que envolve torcer as funções regulares e fixas de um sistema antes fechado, de modo a instaurar relações entre *processos* e *fluxos*:

... é possível dizer que a primeira [recongnitiva] assegura a aprendizagem como solução de problemas, mas é a segunda modalidade de atenção [inventiva] que assegura a aprendizagem como invenção de problemas. A ressignificação do que seja aprendizagem, cujo indício é a atenção ao presente, é ponto fundamental para um novo pragmatismo, não utilitário, mas inventivo. (KASTRUP, 2007, p. 175)

A partir do momento que a exploração deixa o terreno das habilidades de reconhecimento, abandona também a sua associação ao espaço extensivo. Os pesquisadores Ann Weisler e Robert McCall (1976), por exemplo, identificam na atividade de exploração um conjunto de habilidades semelhantes às do modelo defendido por Moravec. Por outro lado, porém, percebem um lapso divergente que as percorre: a necessidade de extrapolar os limites do que é (re)conhecido. Os cientistas descrevem as definições e o estado da arte das pesquisas a respeito tanto da exploração quanto da brincadeira/jogo (*play*). De início parecem operar em domínios excludentes: ou se explora ou se brinca. No entanto, logo os pesquisadores apontam que existe uma zona de indeterminação onde os comportamentos se confundem. Um dos exemplos dados é o de uma criança que se depara com um ambiente novo, uma casa que não é a dela, no caso. O comportamento, a partir daí, oscila continuamente entre exploração – quando a criança se depara com um brinquedo que não conhece e tenta identificar suas propriedades – e brincadeira – quando pega objetos já conhecidos e joga com eles:

Pode ser bem difícil e não tão útil recortar essa sequência de ações em categorias discretas de exploração e brincadeira/jogo; em contraste, pode ser mais útil analisar essa corrente de comportamentos do ponto de vista de sua temporalidade e

⁶² Virgínia Kastrup se vale de exemplos de experiências relativamente comuns, como a de quem volta para a casa em que morou na infância e é invadido por uma sensação de estranhamento: há uma tensão entre o reconhecimento que se construiu no passado – subsistente na memória – e as inevitáveis mudanças pelas quais o indivíduo passou até o momento do reencontro. Essa tensão perturba a estabilidade dos esquemas de reconhecimento, exigindo a invenção de uma nova cognição (Cf. KASTRUP, 2007 p. 69).

sequenciação de ações qualitativas, sem classificar o episódio inteiro seja como exploração ou como brincadeira/jogo⁶³. (WEISLER; McCALL, 1976, p. 497).

Outro fato interessante notado pelos pesquisadores é que tanto a exploração quanto a brincadeira e o jogo não necessariamente se limitam às suas aplicações mais óbvias. Explorar, por exemplo, é comumente, segundo os autores, entendido como uma função de “redução de incerteza”. O jogo e a brincadeira, por suas vezes, seriam formas de experimentação prazerosa, sem propósito extrínseco e sob condições controladas e com baixo nível de incerteza (WEISLER e McCALL, 1976, p. 493). No entanto, estas concepções demasiadamente fechadas caem por terra quando se observa comportamentos brincalhões comuns aos mamíferos, segundo os autores, tais como subir em árvores, entre outras atividades que *umentam* a incerteza e, não raramente, envolvem riscos tangíveis à integridade física dos animais que as praticam. Também nestas situações, é possível identificar uma corrente contínua de ações de exploração e brincadeira/jogo.

Por mais que se tente, porém, é impossível encaixar esse tipo de comportamento nas quatro categorias cognitivas de Moravec. Caso tudo que um animal inteligente fizesse fosse empregar o quadrante da exploração robótica, este eventualmente cairia em um círculo de repetição do qual jamais conseguiria sair. Quanto mais percebesse recompensas e eficiência na sua atividade cognitiva, menos seria encorajado a alargar essa zona de funcionamento mecânico que conecta o reconhecido ao reconhecido. O mesmo vale para o jogador que foge dos ciclos do *game loop* para adentrar terrenos mais intensos, múltiplos e instáveis. É preciso, portanto, dar conta dos momentos singulares em que a exploração faz nascer novas conexões, bifurcando a espiral e adicionando variabilidade ao processo de jogo.

Não podemos, no entanto, simplesmente chamar este comportamento singular de invenção, já que, como alerta a própria Virgínia Kastrup, a invenção não é, propriamente dizendo, uma faculdade, tal como manipular um objeto ou detectar a direção de onde um determinado som vem. Ela é, na realidade, o processo resultante da *quebra* do “funcionamento concordante das faculdades” (KASTRUP, 2007, p. 73). Essa quebra obriga a cognição a inventar um novo esquema, uma nova teia de relações para substituir a que agora encontra-se despedaçada. Nem sempre bastará recorrer às memórias e às experiências passadas, já que essas dependem da concordância entre as faculdades. Será preciso, pelo contrário, colocar um novo problema com novos termos.

⁶³ Livre tradução de: “It may be quite difficult and not very useful to chop this sequence into discrete categories of exploration and play; in contrast, it may be more useful to analyze this stream of behavior from the standpoint of the timing and sequence of qualitative acts without classifying the entire episode as either exploration or play.”

A invenção é tão fundamental à cognição humana quanto sua contraparte complementar, a inteligência. E, assim como a exploração demanda o poder de análise que todo sistema inteligente possui, a intuição possui suas próprias bifurcações na atividade criativa humana. Escolhemos chamar, aqui, o reflexo da intuição nos jogos de sobrevivência de *adaptação*. Essa palavra implica um processo que, na concepção dos biólogos Francisco Varela e Humberto Maturana, envolve a manutenção de um ciclo, ao mesmo tempo que a constante mudança de seu aspecto:

Para que o domínio de interações de uma unidade de interações sofra uma mudança sem perder sua identidade em relação ao observador, essa unidade precisa sofrer uma mudança interna. Reciprocamente, se uma mudança interna ocorre em uma unidade de interações, seu domínio de interações também muda. Um sistema vivo, por sua vez, sofre uma mudança interna sem perder sua identidade apenas se as previsões ocasionadas por sua mudança interna não interferirem com sua organização circular fundamental. (MATURANA e VARELA, 1970, p. 11-12)

A intrincada teia de relações descrita pelo trecho acima descreve uma unidade de interações (um ser vivo, por exemplo) imbricada em um domínio de interações (o nicho em que o ser vive), um observador externo e o ambiente que enxerga. Esse ambiente total não é a mesma coisa que um nicho e nem mesmo é uma área maior da qual o nicho é apenas uma parte. O ser que habita um ambiente constrói seu próprio nicho e, nele, percebe capacidades e propriedades inacessíveis ao observador (reciprocamente, o observador vê qualidades do ambiente que o ser, concentrado em seu nicho, deixa de perceber).

Podemos nos apropriar deste mesmo modelo para abordar a conexão jogador-jogo em *Don't Starve*, com algumas ressalvas. Primeiro, os autores não usam a palavra “adaptação” para se referir a esse processo. Não obstante, o autor Evan Thompson (2004) utiliza o termo para combater a visão neo-darwinista de que o organismo só se adapta quando uma pressão evolutiva o obriga a mudar de comportamento. Para Thompson, a adaptação é a manutenção da organização circular fundamental citada por Maturana e Varela, consistindo em “uma condição subliminar invariante a todo ser vivo”⁶⁴.

Cabe destacar que a biologia de Maturana e Varela⁶⁵ (e, por extensão, de Thompson) vê como única condição invariante da cognição a própria *mudança das condições* para que o ser vivo perpetue seu ciclo fundamental. A adaptação, portanto, nada mais é que a constante reorganização interna e externa pela qual o ser vivo deve passar caso queira manter a *loop* que

⁶⁴ Livre tradução de: “adaptation is an invariant background condition of all life”.

⁶⁵ Os autores são uma das fontes usadas por Virgínia Kastrup para conceber sua crítica ao gestaltismo, construtivismo e outras vertentes que encaram a psicologia como um território governado por leis rígidas e imutáveis.

o faz existir enquanto indivíduo. O paradoxo fundamental a esta ideia é que para que *loop* se perpetue, ele precisa se modificar incessantemente.

A segunda ressalva necessária diz respeito exatamente à adequação desse modelo a um jogo cuja dinâmica é indiscutivelmente mais limitada que a de um nicho biológico qualquer. A saída para esse impasse é alegar que, enquanto uma plataforma sensorial sustentada em um sistema computacional, *Don't Starve* apresenta uma complexidade que exige adaptação apenas até certo ponto. Em dado momento, o jogador *tende* a esgotar as virtualidades do domínio de interação no qual se insere. “Tende” porque, segundo o conceito de instabilidade traçado no capítulo anterior, jamais saberemos quais novas propriedades ou capacidades surgirão das virtualidades conectivas de cada jogo. No entanto, quanto mais o jogador instaurar mecanismos de estabilização da dinâmica do ambiente de jogo, mais a exploração extensiva predominará sobre a adaptação intensiva.

Concretamente, isso se manifesta nos videogames no que comumente é chamado de *endgame*. O *endgame*, em termos simples, é o momento em que, por mais que o jogo não tenha terminado, o jogador já esgotou os platôs principais que o impulsionariam para níveis de complexidade crescentes⁶⁶. Ao alcançar o último nível, não resta opção a não ser repetir sequencialmente ações cíclicas de modo a manter o que já se conseguiu. Nesse momento, a adaptação encontra-se na sua potência mínima, e a exploração torna-se mais mecânica do que nunca: a estabilidade do jogo está em seu auge.

No jogo de sobrevivência *The Flame in the Flood* (2016), o jogador se vê em meio a uma enchente que transforma quase toda a superfície navegável do mundo de jogo em rios que correm de forma instável. É preciso achar zonas de terra firme, coletar recursos para garantir sua subsistência. A missão é dificultada pela hostilidade do ambiente, sobretudo criaturas perigosas como ursos e lobos. Para chegar ao ponto de conseguir lidar com as bestas, o jogador precisa se adaptar: construir um esquema do terreno, navegar de forma ágil nos rios e construir armadilhas, posicionando-as consistentemente. Esse momento de aprimoração, no entanto, logo é substituído pela sensação de que se descobriu o truque por trás de uma apresentação de

⁶⁶ O termo “endgame” é muito mais comumente aplicado em jogos do tipo MMO (uma sigla que costuma ser usada para abreviar o termo “Massively Multiplayer Online Game”). Neles, o momento do endgame é geralmente mais complexo. Mark Jakobsson (2006) identifica em *Everquest* (1999) que o momento do endgame significava que os jogadores já haviam progredido por todas as etapas do jogo e, portanto, passam a voltarem-se uns contra os outros (p. 216-218). Thiago Falcão (2014), em seu estudo a respeito de *World of Warcraft*, aponta o modo como a Blizzard, criadora do jogo, planeja conteúdos especificamente voltados para jogadores neste nível. No entanto, este conteúdo resume-se em grande parte em eventos chamados *raids* e que, embora necessitem de planejamento e execução minuciosos por parte dos jogadores, “esta progressão se realiza, precisamente, com base na repetição” (FALCÃO, 2014, p. 209).

mágica. Basta repetir a receita repetitiva para transformar a tarefa que antes impunha mil barreiras em uma simples sequência de ações.

Figura 31 - Elementos da HUD de Don't Starve e sua relação com o restante do mundo de jogo.



Para destrinchar esse processo em *Don't Starve*, pode-se elencar os platôs que impõe ao jogador. O jogo conta seus dias em ciclos de 8 minutos e, antes da primeira noite, o jogador precisa impedir três indicadores de alcançarem grau zero: sua fome, saúde e sanidade (figura 31). Enquanto o indicador de fome definha progressivamente com o tempo⁶⁷ – quando chega a zero, o jogador morre – sanidade e saúde são danificados por eventos específicos. Como visto antes, o primeiro momento do jogo envolve a sondagem extensiva do ambiente e o reconhecimento de itens como cenouras e frutas pequenas, que alimentam o jogador e impedem o indicador de fome de alcançar níveis críticos. Após esse primeiro momento, o jogo se abre para caminhos mais plurais: o segundo platô envolve o estabelecimento de uma *base*.

Do ponto de vista da adaptação, a construção da base requer uma recolocação do problema inicial. Não basta mais explorar e coletar recursos, é necessário perceber novas teias de relações entre as propriedades e capacidades das entidades do mundo de jogo. O jogador passa a vaguear cada vez menos, organizando, automatizando e otimizando a coleta, distribuição e confecção de recursos. Pode construir baús para armazenar armamentos, comida entre outros itens numerosos demais para carregar consigo. Outra atividade útil é de selecionar certas espécies de vegetação para iniciar uma plantação, evitando a necessidade de se deslocar à esmo à procura de recursos. Na figura 32 abaixo, é possível ver duas plantações: de árvores

⁶⁷ O indicador de fome, caso o jogador não consuma nenhuma comida neste intervalo, demora 2 dias no jogo (ou 16 minutos reais) para atingir o grau zero e acarretar na morte/perda do jogo.

secas e grama alta, que resultam em combustíveis eficientes, mantendo o fogo principal do acampamento vivo com facilidade.

Figura 32 - Composição possível (mas não única) de uma base.



Os baús ajudam a organizar e armazenar recursos. Painéis aceleram e sofisticam a produção de comida, o que resulta em menos risco de morte por fome e até por ataques de monstros ou outros acidentes (já que comida também recupera o medidor de saúde). A cabana, se usada, permite que o jogador durma, acelerando a passagem do tempo do jogo. Isso permite que o jogador, de certa forma, manipule o tempo – por exemplo, dormindo à noite para acordar de dia, não precisando mais lidar com monstros. As árvores secas e grammas crescem em ritmos mais acelerados que a velocidade de consumo do fogo central que ilumina, aquece e protege a base de monstros.

Nesse cenário a única atividade que demanda um nível relativo de exploração por parte do jogador é a caça e/ou coleta de comida – e, mesmo assim, essas atividades tornam-se relativamente triviais se ancoradas em uma base segura. A exploração em busca de comida, por exemplo, tende a ser quase inexistente se o jogador criar plantações de vegetais comestíveis, como cenoura, abóbora, etc. Domesticar e criar porcos para produção de carne também é possível, assim como abelhas para coletar mel ou outras criaturas específicas do jogo. Cada uma delas introduz desafios específicos – porcos, por exemplo, se estiverem soltos às noites de lua cheia, transformam-se em monstros (*werepigs*), atacando o jogador e qualquer outra criatura que se aproxime. O mundo fora da base também continua instável, e o jogador possui uma série de ferramentas para tentar combatê-la: por exemplo, pode construir cercas em volta de sua propriedade, tentando ao máximo revesti-la contra imprevistos vindos de fora.

Se observarmos este processo exclusivamente sob as lentes extensivas da exploração, notaríamos apenas uma redução da frequência de atividade do jogador. Este parece menos suscetível a entrar em combate ou a viajar grandes distâncias: torna-se quase um administrador demasiado sedentário de sua base, ajustando uma plantação aqui e repondo combustíveis acolá. Faltaria acrescentar a essa análise, no entanto, a ótica da adaptação: nela, abandonamos o domínio do mecanicismo e precisamos, seguindo a indicação de Maturana, perceber a *mudança interna* pela qual o jogador passa para entender a variação correspondente no modo como interage com o mundo de jogo. Isso não significa tentar decifrar o pensamento de quem joga. Na verdade, trata-se apenas de desviar nosso ponto de vista: ao invés de tomar o jogador como um agente autônomo inserido em um espaço tridimensional cartesiano, é necessário percebê-lo como uma unidade de interações inseparável do ambiente ao qual se acopla. O sistema vivo humano conecta-se à máquina mundo de jogo e, assim, cria um vínculo “autocontido e que não pode ser descrito usando dimensões que definem um outro espaço” (MATURANA e VARELA, 1980, p. 89).

A chave para essa mudança é direcionar a atenção para a mudança de qualidade das propriedades postas em movimento pelo jogador: não se trata apenas de coletar mais itens e dominar mais território. É necessário conectar peças, automatizar processos e otimizar aqueles que não podem ser “programados”. Mais do que explorar as propriedades extensivas do mundo, é preciso compreender suas *propriedades intensivas*: entre elas, convivem uma série de velocidades e ritmos: o crescimento de grãos, o consumo de combustível do fogo, os ciclos dia/noite, as fases da lua e as estações do ano – bem como as mudanças que cada um deles põem em movimento. A base é penetrada por esses ritmos e o jogador precisa orquestrá-los, como em um arranjo musical onde as capacidades de cada componente (baús, plantações, cabana, cercas, etc.) conversam entre si de modo a manter os três monitores vitais apontando índices altos.

2.2.2 Subnautica: biomas em tensão dinâmica

Ao contrário de uma linha de montagem, no entanto, encadear peças em *Don't Starve* é uma atividade continuamente ameaçada pelos movimentos bruscos do mundo em torno de sua base. De fato, a estabilidade total pode *quase* ser alcançada, mas nunca totalmente: sempre haverá um escape, por menor que seja, de um movimento imprevisto que sai do mundo e se lança à circunscrição traçada pela perspectiva do jogador. Na análise de *Don't Starve*, nos limitamos a descrever a transição entre dois modos de funcionamento no jogo: da exploração

mecânica e extensiva da busca por recursos à adaptação intensiva e maquínica da composição de uma base funcional e dinâmica. Nessa descrição, porém, demos foco ao ponto de vista do jogador. Como o mundo de jogo, em sua dinâmica, deve funcionar para que a necessidade de adaptação predomine sobre a exploração?

Uma das respostas possíveis – que procuraremos desdobrar nessa etapa do capítulo – é que o mundo se divide em *biomas*. A escolha da palavra é propícia por dois motivos: primeiro porque é usada pela comunidade de jogadores para distinguir as várias áreas que compõem os mundos de jogos de sobrevivência como *Don't Starve*. O segundo motivo é que, pelo menos nos jogos desse tipo, os biomas funcionam em grande parte como uma metáfora para suas contrapartes reais: constituem áreas de qualidades distintas, impondo às entidades que as habitam e/ou visitam a necessidade de se adequar às suas forças.

De maneira mais específica, os biomas de *Don't Starve* não aparecem apenas como diferenças visuais. Áreas desérticas são amareladas e dessaturadas, enquanto florestas densas possuem cores vivas ao longo do dia e escuridão total à noite. Esses biomas, se seguirmos a concepção de mundo de jogo delineada no primeiro capítulo, são mais do que qualidades visuais: são dinâmicas postas em movimento pelo contato entre suas propriedades. A divisão em biomas é, portanto, resultado da instabilidade do jogo: se percorrêssemos toda sua extensão sem encontrar diferenças *ativas* entre um deserto e uma floresta, estaríamos sem dúvida diante de um ambiente muito mais estável.

Em diferentes escalas, esse padrão pode ser observado em uma gama de jogos de sobrevivência, inclusive o próprio *Minecraft*, que possui planícies desérticas, montanhas esbranquiçadas por neve, florestas de diferentes composições florais, entre outros. Certamente, é possível detectar até que ponto esses biomas alteram a dinâmica do jogo, e não apenas sua aparência.

O já citado *No Man's Sky* talvez apresente as diferenças menos intensas entre esses setores. Nele, os planetas fazem o papel de “biomas”, já que um planeta tropical, por exemplo, tipicamente será recoberto de árvores e lagos, bem como de um número relativamente limitado de espécimes de fauna e flora. Para encontrar uma área radicalmente distinta, o jogador deve viajar a outro planeta e se deparar, por exemplo, com um astro revestido por uma atmosfera tóxica. No entanto, a repetição logo esgotará a novidade: poucas diferenças são notáveis entre dois planetas tóxicos ou dois planetas tropicais – o contraste limita-se a cores, formas e aspectos quase que puramente visuais. Um jogo que introduz o mesmo apelo a viagens interplanetárias, *Astroneer* (2016), distribui em um mesmo planeta diversos microbiomas, que consistem em sistemas de entidades em relações distintas. Isso leva, por exemplo, à experiência de descobrir

novos recursos que antes jamais haviam sido avistados pelo jogador em um determinado planeta. Em *No Man`'s Sky*, pelo contrário, por mais que cada astro possua tamanho aproximado ao da terra, é possível reduzir a variedade de recursos minerais, por exemplo, a apenas três ou quatro tipos: por mais que se explore cavernas profundas e oceanos quilométricos, não há esperança de se achar novos tesouros.

Sendo assim, o que estamos procurando nestes exemplos são interações diversas *nos* biomas e *entre* os biomas. Para explorar essa afirmação, enquanto um padrão recorrente em jogos de sobrevivência, optamos por abandonar (temporariamente) *Don`'t Starve* e abordar um segundo jogo: *Subnautica* (2013). Seu mundo ajuda a compreender, primeiro, que essa divisão não como é traço essencial ao “gênero” sobrevivência. Existem jogos, como *Factorio* (2013) ou *Day Z* (2012) que não recortam seus mundos em sub-áreas com suas próprias sub-dinâmicas. São, portanto, e, mesmo assim, apresentam condições para a exploração e adaptação em um mundo onde o jogador é descentralizado. Mais importante que catalogar quantos e quais mundos de jogo dividem-se biomas, é expor o mecanismo que essa divisão desdobra: a tensão dinâmica entre as entidades que compõem o funcionamento do jogo.

Figura 33 - Subnautica: jogador explora o oceano.



É possível avistar à frente cardumes de peixes e plantas que oferecem oportunidade de extração de recursos.

Subnautica imerge o jogador em um oceano, no qual seu único abrigo é um pequeno cômodo ejetado de uma enorme nave. A sua imensa estrutura pode ser vista ao longe, flutuando no oceano e embebida em chamas causadas pela explosão e pelo seu choque contra o planeta desconhecido. De castigo em sua cápsula, resta ao jogador ancorar-se nesse ponto e usá-lo como uma base – mesmo que bem menos dinâmica que a base analisada em *Don`'t Starve*.

Assim como no estudo de caso anterior, veremos em *Subnautica* o aparecimento de alguns padrões dos jogos de sobrevivência: precisamos nos atentar a três níveis que diminuem com o tempo e que, quando zerados, acarretam na morte do jogador: fome, sede e oxigênio

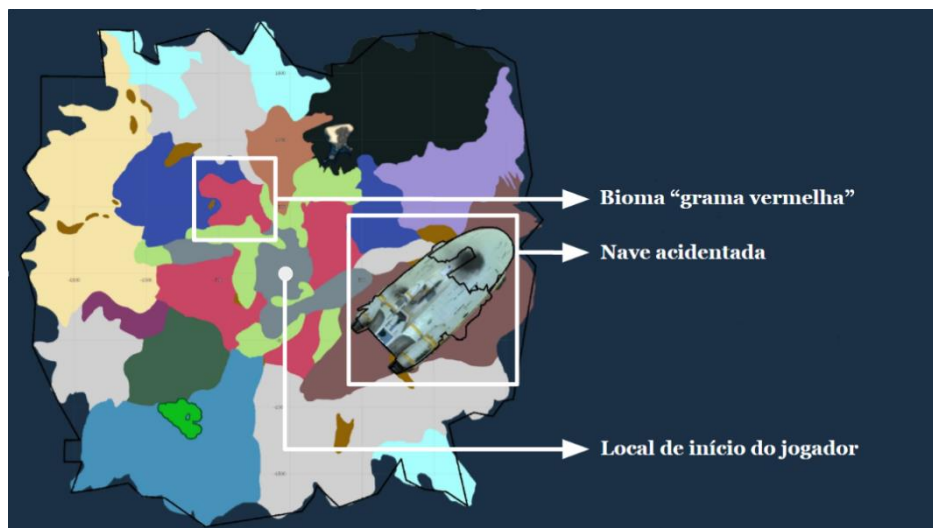
(esse último apenas quando se está debaixo d'água). Um quarto indicador aponta o nível de saúde do jogador, que é danificado por ataques de criaturas hostis e privação de oxigênio, e recuperado pelo uso de kits médicos e ingestão de alimentos.

A pressão de ver seus indicadores de fome e sede definhando logo impulsiona o jogador aos primeiros momentos de exploração. *Subnautica* introduz uma propriedade inédita em relação ao espaço bidimensional de *Don't Starve*. O oceano permite que o jogador nade e se desloque em 360°, em qualquer direção. Enquanto em *Minecraft* utilizamos da habilidade de cavar para alcançar níveis cada vez mais profundos, a simples locomoção em *Subnautica* pressupõe essa habilidade. De início, porém, o jogador conhece pouco do mundo que o rodeia e, caso não tome as atitudes necessárias, pode, por exemplo, se aprofundar demais na água e não conseguir voltar à superfície a tempo de respirar.

A saída está, mais uma vez, em explorar as redondezas imediatas da base em busca de certos itens essenciais para que o jogador atinja uma das fases crítica de adaptação: a ampliação de sua base. Por exemplo, peixes inofensivos circulam por perto: estes, se capturados, podem ser levados à base, onde o dispositivo chamado de *fabricator* processa alimentos. Além de comida, o jogador precisa de líquido para matar sua sede: peixes chamados *bladderfish* também nadam pelas proximidades do bioma inicial onde o jogador “nasce”, e se processados pelo *fabricator* podem resultar em água. Corais e depósitos de sal também se distribuem pelo espaço, e deles o jogador pode coletar itens que, quando combinados fabricam água sanitária. Este item de segunda ordem é fundamental para desinfetar água, que por sua vez é mais eficaz em manter o nível de sede acima da zona de perigo.

Nos primeiros momentos do jogo, portanto, tudo que o jogador precisa para sobreviver está a curto alcance. Porém, essa feliz coincidência se mostra menos piedosa com o cair da noite. O jogador precisará utilizar sua lanterna – e encontrar depósitos de cobre para fabricar baterias caso a sua se esgote – lidar com visibilidade reduzida e criaturas marinhas bem menos amistosas. *Subnautica* à noite se apresenta quase como um jogo de terror, uma impressão que pode ser intensificada, também, pela experiência insólita de penetrar em um novo bioma.

Figura 34 - Mapa de *Subnautica*.



Apesar das áreas estarem destacadas por cor, vale frisar que no jogo não há fronteiras claras entre os biomas.

Se observarmos o mapa do mundo de jogo, essas áreas são claramente demarcadas e definidas. Acontece que, no mundo de jogo, as indicações são menos rígidas: talvez a cor da grama que nasce no solo seja diferente, ou a água seja levemente mais turva. As fronteiras entre biomas, para utilizar o termo proposto por Daniel Vella (2015) em sua análise de *Dark Souls* (2011), são *indistintas*: não são tanto barreiras que separam duas áreas cerradas, mas pontos de contato onde tensões ambíguas se desenvolvem.

Por exemplo, em um dos biomas que faz fronteira com a área inicial do jogo, a água é esverdeada e plantas esguias crescem quase até a superfície, tornando-o um dos setores mais facilmente reconhecíveis do mundo de jogo⁶⁸. Nele, tubarões agressivos chamados de *stalkers* patrulham seus territórios, às vezes sumindo escondidos entre as algas que crescem na área. *Stalkers*, porém, não se limitam a vigiar áreas definidas em patrulhas fixas e cíclicas. Caso um peixe arrisque chegar perto de sua ronda, *stalkers* logo perceberão e perseguirão a presa. O jogador pode ser uma delas: se chegar muito perto, chamar a atenção de um e tentar fugir, *stalkers* não desistirão facilmente da perseguição. Além disso, ainda possuem a capacidade de coletar itens úteis ao jogador, como placas de metal que deslocaram da nave após o acidente. *Stalkers* colecionam itens como esse, buscando-os em pontos longínquos e depositando-os perto de suas próprias “bases”, quase como quem desafia o jogador a tentar roubá-los.

Sendo assim, embora seja tentador reduzir *Subnautica* ao seu mapa, essa atitude acarretaria na perda de boa parte das dinâmicas que ocorrem no jogo. Isto porque os biomas

⁶⁸ No gameplay de apoio a seguir, é possível acompanhar o jogador em seus primeiros momentos de reconhecimento do território inicial de *Subnautica*. Sem perceber, o jogador cruza a fronteira entre o bioma inicial e uma das áreas patrulhadas por *stalkers*, atraindo um por acidente. Fonte: https://youtu.be/7sv5_YwLt6o?list=PLKcc0xOPUy2Gn0B8dTSuf3zJPibn69UT4&t=1562

não são apenas uma reunião de propriedades inseridas em fronteiras sólidas. São, na verdade, áreas com um nível razoável de autonomia, mas que, não obstante, estabelecem contato e influência mútuos com outros biomas. Fica óbvio, a partir dessa constatação, que essa é uma das fontes de instabilidade em *Subnautica*: mesmo depois de explorarmos e nos adaptarmos a um bioma x , ainda precisamos estar atentos às possíveis perturbações que o contato com suas fronteiras com o bioma y indistintas podem causar.

O mecanismo que se esconde por trás desse tipo de organização é a *continuidade* estabelecida entre as diferentes “subdinâmicas” que compõem o dinamismo global do mundo de jogo. Em *No Man’s Sky*, por mais interessante que um planeta seja, ele está distante demais (e, mais importante: desconectado demais) dos outros planetas da galáxia. Os aliens que o habitam não trocam mercadorias com outros sistemas solares; a lua que gira em torno de sua órbita não afeta as marés e o ritmo das estações do ano; as plantas que os ocupam não são comidas por seres herbívoros e estes, por suas vezes, não precisam se preocupar em ser caçados por seres carnívoros. Fica muito difícil descrever as conexões entre componentes sem recorrer a suas propriedades extensivas: os anos-luz que separam galáxias, as áreas vastas das superfícies dos planetas, por exemplo. Os ritmos até existem: dias viram noites, e animais fazem rondas cíclicas. No entanto, é exatamente por esses ritmos não se perturbarem mutuamente que é possível posicioná-los em um mapa e medi-los extensivamente: a exploração basta, a adaptação é ínfima.

É certo que o contato entre biomas em *Subnautica* não alcança complexidade e conectividade tão altas quanto esses exemplos do que falta em *No Man’s Sky*. No entanto, elas apontam, de modo sutil e incipiente, o princípio básico que pode gerar dinâmicas progressivamente mais intensas, sobretudo em jogos como *Dwarf Fortress* e *Ultima Ratio Regum* – que serão estudados no terceiro capítulo. Por enquanto, basta notar que o resultado da conectividade de *Subnautica* implica uma mudança de atitude. É preciso refazer internamente a impressão que se tem do mundo para dar conta das instabilidades que dela surgem.

2.2.3 A topologia de Factorio: ciclos progressivos de adaptação

No artigo “In the Frame of the Magic Cycle: The Circles of Gameplay” (2009), Bernard Perron e Dominic Arsenault (constroem um modelo que se assemelha com as análises que desenvolvemos até aqui. Os autores buscam estabelecer um entendimento da ideia de círculo mágico que desmistifique certas interpretações questionáveis do conceito de Johan Huizinga, mas sem, necessariamente, descartar o conceito por completo. Arsenault e Perron parecem

concentrarem-se em denotar que o círculo mágico é menos uma redoma alienada do cotidiano e mais uma mudança de atitude, a instauração de círculos (ou ciclos) de jogo. Portanto, “nesse sentido, a imagem do círculo mágico externaliza o processo cognitivo implicado no ato de entrar em um jogo” (p. 114)⁶⁹.

Os autores admitem que a proposta parece desprezar algo de místico fundamental ao jogo; algo de transcendente que se esvairia caso reduzíssemos sua força a uma análise mental, corporal, enfim, cognitiva. O contraponto que oferecem é que o efeito é justamente oposto: ao observarmos o estabelecimento de contato entre jogo em termos práticos e empíricos, temos acesso a novos significados para a ideia de círculo mágico. Outro artigo, de Thiago Falcão e Emmanoel Ferreira, investe nessa mesma aposta, identificando o círculo mágico como uma combinação da estrutura do jogo enquanto objeto e do “jogo enquanto atividade; no momento em que ele adquire jogadores e, assim, estes iniciam os processos de apropriação e de adaptação comuns à relação entre homens e objetos” (FALCÃO; FERREIRA, 2016, p. 79).

Não seria essa atividade, portanto, vista sob seu aspecto cíclico, a própria dinâmica contínua de exploração e adaptação? Esses ciclos, quando redirecionados pela adição de variabilidade proveniente da adaptação inventiva, transformam-se em espirais (Cf. ARSENAULT; PERRON, 2009 p. 116)., levando o jogador a experimentar novas dinâmicas de instabilidade e conectividade. Aqui, trata-se de saber quem é responsável por definir as condições sob as quais este espiral se movimenta. Enquanto jogos que centralizam seu funcionamento no progresso do jogador são arquitetados de acordo com o progresso de quem joga, em jogos como *Minecraft*, *Don't Starve* e *Subnautica* esse tende a se descentralizar do jogador.

Factorio (2013) talvez seja o caso mais adequado para tratar deste tema. Nele é possível enxergar a manifestação material do “círculo mágico” de Arsenault e Perron, bem como as estruturas de funcionamento que motivam os ciclos de atenção identificados por Falcão e Ferreira. Nesse sentido, é possível demonstrar como jogo e jogador se conectam na atividade lúdica: ambos desempenham papéis diferentes, porém complementares, para o desenvolvimento do espiral de adaptação.

Figura 35 - Perspectiva aérea do mapa de Factorio.

⁶⁹ Livre tradução de: “In that sense, the image of the magic circle externalizes the cognitive processes implied by the act of entering a game.”



Destaque para três dos recursos básicos do jogo, que precisam ser localizados e extraídos continuamente pelo jogador.

O ciclo que inicia o jogo é o de extração de quatro recursos básicos: madeira, carvão, minério de cobre e minério de ferro. O primeiro é encontrado em florestas: diferentes espécies de árvore proverão madeiras com propriedades distintas. Pedras são facilmente achadas, enquanto depósitos de cobre e ferro possuem concentração um pouco mais diluída. Portanto, o primeiro ciclo de atividade do jogador é explorar o ambiente em busca destes recursos. O mundo gerado proceduralmente espalha-os segundo seus próprios algoritmos, o que quer dizer que a tarefa é quase puramente mecânica: não há como intuir onde haverá cobre ou ferro, é preciso perambular cegamente até se deparar com uma mina de tamanho satisfatório.

Em seguida, o segundo passo do primeiro ciclo se apresenta: o jogador precisa extrair os recursos. Faz isso, de início, por um processo também mecânico: clica em árvores ou em um depósito de ferro, por exemplo, e comanda o avatar a iniciar a extração. Trata-se de um processo simples: um clique e a seleção de uma habilidade. Uma vez munido do resultado de seu trabalho, o jogador pode armazená-los no inventário. Logo, porém, este ficará cheio e o jogador precisará iniciar o segundo ciclo: a construção de novos objetos, erguidos a partir do material que acabou de extrair.

De certa forma, parte do que guia o progresso dos ciclos de adaptação em *Factorio* são as “receitas” que ensinam o jogador a criar objetos com novas propriedades e capacidades. Para armazenar minérios de ferro e cobre que extraiu com sua picareta, por exemplo, o jogador pode construir uma caixa com a madeira que coletou das árvores. Essa caixa armazena um número maior de objetos que o limite carregável manualmente pelo avatar. Sendo assim, o jogador delega ao mundo de jogo uma função que, antes era executada pelo pequeno personagem que controla: armazenar itens.

O segundo ciclo, portanto, começa quando o jogador constrói objetos e delega a estes certas funções repetitivas. Ao fazê-lo, abre espaço em seu regime de atenção para outras

atividades de conexão. É nesse ponto que entram as “receitas” de construção: elas imporão ao jogador a quantidade e ritmo de extração de cada recurso. Madeira, por exemplo, é um item que logo se torna supérfluo. Como combustível, o carvão rende mais; como material de construção, ferro e pedra são mais duradouros e resistentes. Portanto, deverão ser extraídos em maior velocidade. Recorremos novamente às nossas receitas: há, entre elas, um objeto feito de ferro que permite que automatizemos a extração de pedra e minérios metálicos: a *mining drill* pode ser instalada sobre depósitos e extrair peças individuais de pedra e metais continuamente. Precisa apenas de um combustível que a faça funcionar: inicialmente, a opção mais eficiente é carvão. Delegamos, novamente, uma tarefa que nos pertencia a um objeto automatizado. Agora deixamos de nos preocupar em minerar e passamos a criar formas de automatizar e manter uma linha de produção.

Fornalhas para processar minério de ferro/cobre em ferro/cobre puros também são úteis, e nos dão acesso a estes recursos mais avançados, protagonistas de uma série de receitas. Todas elas resultam em objetos com novas propriedades e capacidades que precisam ser organizadas *espacialmente*. Isso porque a disposição dessas entidades mecânicas no espaço consiste no terceiro ciclo de jogo em *Factorio*. É preciso otimizar a localização das minas de extração de minérios, construir esteiras automáticas para leva-los até suas fornalhas, posicionar braços mecânicos que retire o metal processado da fornalha e o coloque em um outro trilho, dessa vez destinado a uma unidade de construção automática.

Figura 36 - *Factorio*



Passo a passo de um dos muitos ciclos que o jogador precisa engendrar. O passo 1 consiste em extrair minério de ferro com um minerador; o passo 2. Usa fornalhas para processar minérios em placas de ferro. O passo 3, finalmente, usa uma montadora para transformar as placas em engrenagens. Esse ciclo precisa ainda se conectar a outros ciclos, como o de produção de outras peças, geração de energia (a óleo, a vapor, solar. Etc.), entre outros.

Unidades como essa permitem que o jogador delegue aos objetos que cria não apenas a construção de objetos simples, mas a *execução de receitas*. Esteiras automáticas podem levar a uma mesma unidade de construção, por exemplo, 1 peça de ferro, 5 de enxofre e 100 de água para construir 50 unidades de ácido sulfúrico (usado em produção de baterias, que por sua vez representam uma fonte de energia eficiente). O mesmo pode ser feito com objetos mais complexos, usados com frequência pelo jogador, como braços mecânicos. As peças ajudam a produzir as próprias peças que produzirão novas peças.

Este ciclo, portanto, envolve a conexão de objetos em circuitos especializados de produção. Logo, o quarto e último ciclo de jogo em *Factorio*, envolve conectar *circuitos com outros circuitos*⁷⁰. A produção nuclear deverá se agrupar à de energia elétrica; a extração de petróleo deverá ser coordenada à produção de metal e de plástico. As receitas continuam a existir e demandam materiais de produção cada vez mais complexos (ou seja, que demandam mais etapas de produção) para gerar resultados com propriedades e capacidades cada vez mais intensas. A receita final do jogo é uma lista de “ingredientes” para a construção de um foguete que, no background ficcional, possibilitará que o avatar fuja do planeta onde o jogo se passa e volte à sua terra natal. Em termos práticos, construir o foguete significa que o jogador terminou a campanha principal de *Factorio*.

É necessário frisar alguns pontos. Primeiro, cada vez que um novo ciclo se abre em *Factorio*, a conectividade do mundo se amplia. É possível, assim como em *Don't Starve*, conectar mais coisas a mais coisas, o que torna a exploração menos mecânica e mais adaptativa. As receitas se apresentam como fórmulas fechadas, que fazem convergir certas forças do mundo de jogo em combinações necessárias e inflexíveis. O posicionamento dos objetos construídos, no entanto, é fluido, flexível e alcança níveis progressivos de modularidade. Os circuitos de jogo funcionam muito mais como frases compostas por letras do alfabeto: embora funcionem sob uma certa gramática, podem ser reconstituídas com razoável flexibilidade.

Essa dimensão do jogo se manifesta com mais evidências no momento em que o jogador precisa conectar ciclos autossuficientes (figura 36) a outros ciclos também autossuficientes. O espaço é fundamental nessa tarefa: enquanto para fabricar manualmente uma engrenagem basta que possuamos os seus ingredientes em mãos (duas placas de metal processado), automatizar esta produção é mais complexo. Deveremos fazer uso de esteiras, pistões e máquinas,

⁷⁰ No gameplay de apoio, podemos perceber como já na primeira hora de jogo, o jogador começa a automatizar processos antes manuais, justamente ao conectar um circuito ou ciclo de ações a outro, por meio de itens que realizam a ligação. Fonte: <https://youtu.be/fBY-5toItTE?list=PL-7ICWbZUDjmBkUdFGh1CxxN0gn9oL0aY&t=34>

posicionando-os em locais estratégicos do espaço gradeado do mundo de jogo. O espaço, assim torna-se progressivamente menos extensivo e mais intensivo: não estamos lidando mais com casas discretas justapostas de modo homogêneo. Cada casa do espaço de *Factorio* torna-se heterogeneamente única: uma delas será a posição ideal para um pistão, e outra possibilitará a composição mais eficiente de pontos de distribuição de energia elétrica para alimentar um fabricante.

Por último, cabe notar a diferença entre a progressão em *Factorio* e um jogo como *Super Mario Bros.* ou mesmo nos exemplos usados por Arsenault e Perron:

Mesmo em jogos de emergência, como *The Sims* (Maxis, 2000) or *Civilization* (MicroProse, 1991), o jogador inicialmente possui alguns poucos parâmetros e recursos para gerenciar, mas eventualmente vai além das simples tarefas cotidianas e do controle de apenas uma cidade até relações românticas e um reino ou império inteiros para governar⁷¹ (ARSENAULT; PERRON, 2009, p. 116).

Os autores tratam aqui da progressão da espiral do ciclo mágico nos jogos, contrapondo jogos de progressão tradicionais, em que as fases vão se dificultando com o tempo e jogos de emergência, que seriam menos estruturados, mas ainda assim acabariam resultando em uma progressão semelhante.

De fato, mesmo nos jogos citados acima, e até em *Factorio*, é possível dizer que há uma estrutura de progressão que coincide com o desenvolvimento do jogador. Isso não quer dizer que esses jogos usam os mesmos métodos para encorajar e intensificar esse desenvolvimento. Como dito anteriormente, um jogo estruturado como *Super Mario Bros.* molda cada ponto de sua geometria no sentido de encorajar comportamentos progressivamente complexos por parte do jogador. *Civilization* e *The Sims* são mais modulares: possuem alto grau de conectividade, permitindo que o jogador construa sua casa ou império conectando peças e reconstituindo o mundo de jogo para que este se adeque à sua progressão. Em outras palavras, não é a geometria do mundo de jogo que guia o jogador em direção ao progresso, mas sim a topologia da conectividade que precisa ser desmembrada para dar origens a entidades mais conectadas e cada vez mais complexas.

O processo que descrevemos em *Factorio* é exatamente esse: por mais que o jogador esteja explorando uma determinada extensão de mundo, está, muito mais, se adaptando a uma teia de forças cerzida pela conectividade das entidades. Nesse sentido, o jogador se perde em uma miríade de possíveis circuitos onde está longe de ser o centro. Mesmo em *The Sims* ou

⁷¹ Livre tradução de “Even in games of emergence, such as *The Sims* (Maxis, 2000) or *Civilization* (MicroProse, 1991), the gamer at first only has a few parameters and resources to manage, but eventually goes beyond the simple menial tasks and single-city micromanagement to romantic relationships and a whole kingdom or empire to run.”

Civilization, encontramos modos de nos colocar em perspectiva onipresente e voltar a uma posição aérea onde conseguimos ver cada entidade e suas propriedades em atuação. Em *Factorio*, isso pouco nos ajuda: mesmo que possamos ver o cenário de cima, nosso poder de ação não alcança distâncias muito grandes. Somos um único personagem, preso ao seu tamanho ínfimo perto da estrutura imensa que pode construir. A única forma de possuir onipresença e onipotência sobre essa estrutura é, paradoxalmente, delegar a ela própria os meios para se manter segura, robusta e potente.

2.3 Origens errantes: conectividade e instabilidade na camada subalterna dos videogames

Os irmãos Tarn e Zach Adams já sabiam programar antes de aprenderem a ler. A cultura dos computadores estava começando a aflorar quando seu pai, também programador, percebeu que familiarizar os filhos com a técnica de escrever em algoritmos poderia influenciar positivamente o futuro de ambos. A infância e adolescência da dupla deu origem a inúmeros programas – pequenos jogos, em sua maioria. Tarn desenvolveu maior familiaridade com os códigos ao longo da fase adulta e, em 2002, com a assistência do irmão, começou a tecer as bases de um projeto especial.

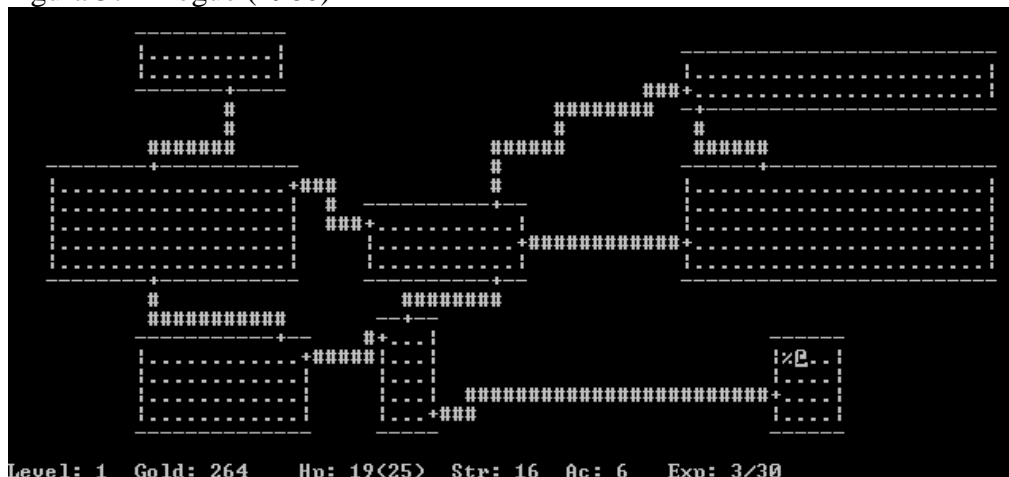
Dwarf Fortress traduz a empolgação de Tarn e Zach com jogos que se transformavam quando os jogadores não estavam olhando. Desde os primórdios dos sistemas de *bulletin board*, já baixavam versões dos *dungeon crawlers*, que simulavam cavernas labirínticas povoadas por criaturas agressivas. Algumas dessas cavernas – as preferidas dos irmãos – eram brutalmente impiedosas com o jogador: seus monstros aniquilavam os aventureiros em poucos golpes, caso não se preparassem adequadamente. Controlava-se um único personagem cuja morte era incontornável: não havia possibilidade de salvar o jogo e retornar de um ponto anterior. Era necessário recomeçar do zero, criando um aventureiro novo. Essa mecânica ficaria conhecida como “*permanent death*” (morte permanente), apelidada comumente de *permadeath*.

Nas partidas subsequentes, outra propriedade impiedosa dos jogos se manifestava: as cavernas não mais manifestavam a mesma configuração. A cada partida, corredores mudavam de direção, salas mudavam de tamanho e monstros trocavam de posição, impedindo que o caminho fosse decorado ou desenhado em mapas. Não havia limite para as combinações possíveis, já que elas não eram organizadas peça por peça por um game designer. A programação continha definições de parâmetros de construção interpretáveis pelo processador – por exemplo, a área máxima e mínima de cada sala, ou o conjunto de monstros que podem ou não aparecer em cada andar da caverna, entre outras variáveis. A unidade de processamento

central passava então a abrigar virtualmente, simultaneamente, todas as variações possíveis das cavernas. Nos *roguelikes*, o game designer não constrói o jogo em sua forma final. Seu papel é criar a semente que faz seus mundos nascerem. Esse método é o mesmo utilizado por *Minecraft*, a já mencionada geração procedimental de conteúdo.

Rogue (1980) foi o jogo que popularizou essa mistura de atributos, dando origem ao gênero batizado de “*roguelike*” (literalmente, “parecido com *Rogue*”). É a criação de um grupo de estudantes que reunia-se nos subsolos da Universidade da Califórnia; primeiro em Santa Cruz, onde Michael Toy e Glenn Wickman se conheceram e começaram em 1978 e começaram a programar as bases do jogo; e depois em Berkeley, quando Ken Arnold se juntou à colaboração desenvolvendo códigos que permitiam transferir o jogo para diferentes espécies de terminais⁷². *Rogue* viria a representar a cultura underground da indústria dos videogames como nenhum outro, por uma série de razões. Primeiro pelo seu completo desprezo por qualidade visual, depois pela preocupação muito maior com intensidade de processos computacionais e finalmente pela tradição posterior de liberar o download e o código original dos jogos de forma gratuita, possibilitando um processo de desenvolvimento colaborativo e descentralizado.

Figura 37 - *Rogue* (1980)



Ao se deparar com uma tela de jogo de um *roguelike* tradicional, o jogador leigo provavelmente pensará que está tendo acesso a um código de programação ou a um *mainframe* das primeiras décadas da história da computação. Não há imagens, apenas caracteres. Estes, porém, não formam palavras inteligíveis. Apenas após uma familiarização maior com o jogo é possível perceber que eles são a expressão visual das entidades de seu mundo. Em *Rogue*, por exemplo, uma sequência de “#####” desenha um corredor que liga uma sala a outra. Em uma

⁷² Fonte: <https://www.youtube.com/watch?v=4IkrZkUV01I>.

sala, cada “.” significa uma unidade espacial ocupável – como as casas de um tabuleiro de xadrez. Logo percebemos que estamos diante de um espaço visto “de cima”, como a planta de uma casa (figura 37). O jogador é representado por um “@” que se desloca pelo espaço seguindo nossos comandos. Também não é incomum avistar letras maiúsculas se movendo pelas salas: um “B” é um morcego, enquanto um “D” sinaliza um dragão e um “Z” um zumbi.

Além da ausência de gráficos e do refinado código que gera seus espaços procedimentalmente, os desenvolvedores de *roguelikes* clássicos tomam uma terceira atitude para se distanciar da camada superior da indústria dos videogames: muitos deles não cobram por suas criações. É uma espécie de código de ética da comunidade de desenvolvedores e jogadores que um *roguelike* legítimo não só é disponibilizado de modo gratuito para download, mas também divulga seu código fonte livremente. Qualquer um com conhecimento de programação, portanto, pode simplesmente copiar o código e modifica-lo com nível microscópico de detalhamento.

Principalmente em sua incipiência, isso possibilitou com que o gênero *roguelike* simulasse uma espécie de árvore genealógica que traçava, literalmente, uma evolução de espécies. Após o ancestral-comum *Rogue*, dois herdeiros primordiais: *Moria* e *Hack*. De cada um deles, emergiram escolas de desenvolvimento de *roguelikes* que exploram estéticas de instabilidade e conectividade segundo princípios distintos. A partir dos anos 1990, o sucesso nas universidades foi substituído por uma cultura de nicho quase imperceptível – mas ainda viva. Finalmente, com o boom da indústria independente de videogames e com o sucesso de *Minecraft*, tanto desenvolvedores quanto jogadores parecem ter lembrado da história dos *roguelikes*. Com isso, pa O objetivo dessa sessão é rastrear a descentralização dos jogadores operada por este fenômeno lúdico. Não almejamos, com isso, catalogar ou desenvolver tipologias dos *roguelikes*. Comunidades como a *RogueBasin*⁷³ já fazem um trabalho excepcionalmente extenso e completo nesse sentido. Queremos, na verdade, apontar os modos de descentralização que os *roguelikes* trouxeram já na incipiência da indústria, bem como diagnosticar sua influência sensível nos videogames descentralizadores de hoje, inclusive *Minecraft*.

Iniciaremos este processo por meio de uma analogia entre jogos como *Rogue* e indivíduos físicos. Veremos que computadores *simulam* processos de formação que mantêm propriedades e capacidades semelhantes a de processos do mundo natural. Depois, procuraremos nas raízes dos *roguelikes* os modos mais simples de criação de ambientes:

⁷³ Fonte: http://www.roguebasin.com/index.php?title=Main_Page.

primeiro lidaremos apenas com labirintos, onde o jogador pode apenas escolher a direção de seu movimento. Depois disso, abordaremos jogos que, como o próprio *Rogue*, acrescentam à necessidade de movimento preciso, a percepção e aprendizagem do comportamento de diferentes entidades, como monstros e armadilhas. Progressivamente, alcançaremos mundos de jogo onde nuances nestes comportamentos são conectadas de modo a simular mundos com composições de forças qualitativamente distintas: cavernas onde tesouros se escondem em meio a ameaças, mas também cidades onde há segurança e oportunidade de repouso e negócio; cenários pós-apocalípticos onde se precisa coletar sucata para construir armas improvisadas; territórios disputados por fações de diferentes orientações éticas, entre muitas outras – tudo isso sem sair da esfera de influência dos *roguelikes*.

2.3.1 Cristalização: intensificação da conectividade na gênese roguelike

Figura 38 - Fotografias de flocos de neve – Wilson Bentley



O fotógrafo americano Wilson Bentley começou a capturar imagens de flocos de neve em 1885. Atraído pela beleza única de cada exemplar – é quase impossível encontrar dois idênticos –, criou um método para registrá-las antes que os flocos sublimassem e os seus padrões intrincados fossem perdidos para sempre. Em 1910, havia conseguido fotografar com nitidez 1600 flocos de neve, cada um deles inconfundível com os outros. Neste mesmo ano, escreveu o texto *Snow Beauties*, produto do estudo que realizou a respeito dos processos físicos que geram as formas dos flocos de neve que fotografava:

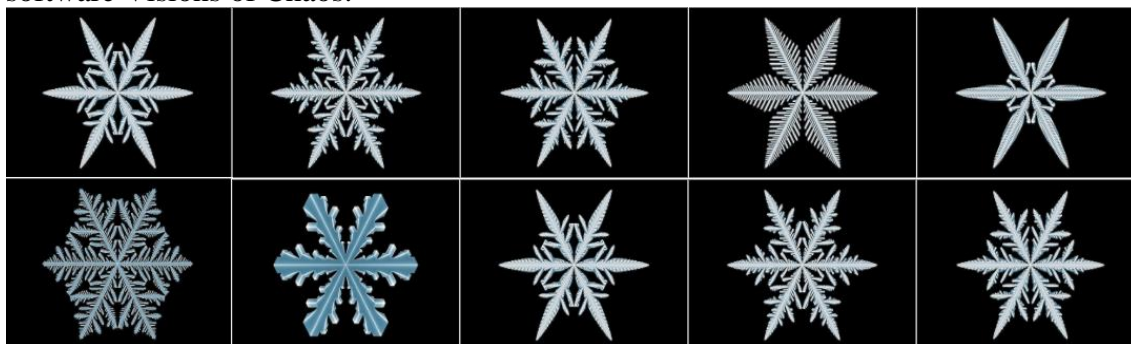
[...] estas fotografias servem para representar com alguma justiça quase todos os tipos e variedades de neve que ocorrem na natureza, no entanto mostram uma fração infinitesimal das variações individuais de forma e desenho interior dentre as incontáveis miríades de cristais que compreendem cada tipo. (BENTLEY, 1910)

Por mais distintos que os flocos de neve sejam, existem padrões em seus formatos que controlam o caos que guia a composição de suas formas. Por exemplo, todos possuem simetria hexagonal. Outras propriedades surgem como atributos comuns em subconjuntos, gerando

“espécies” com mais traços parecidos, mas ainda assim exibindo desenhos característicos. Em suma, existem constantes que atraem a repetição, mas ao mesmo tempo forma-se um embate de forças cujas variáveis atuam com liberdade o suficiente para produzir diferença dentro do mesmo processo.

É possível identificar na formação de flocos de neve, portanto, um conjunto de variáveis que interferem nos caminhos e transformações consumadas na interação entre moléculas de água. Essa identificação possibilita a simulação computacional do nascimento dos flocos de neve, que designam aos pixels do monitor formas igualmente complexas. Um programa relativamente acessível, como o software *Visions of Chaos*⁷⁴, consegue a partir de certos parâmetros, originar formas tão variadas quanto as fotografias de Bentley.

Figura 39 - Simulação computacional da formação de flocos de neve, executadas no software *Visions of Chaos*.



Baseado em uma simplificação do modelo matemático de Janko Gravner e David Griffeath (2009).

Há de se levantar o paradoxo, no entanto, de que não se programa em uma simulação computacional *todas* as variáveis que atuam na formação de um floco de neve. Tal missão seria excessivamente árdua, necessitando da inserção de parâmetros relativos tanto ao movimento climático global quanto de vetores de nível quântico. A simulação acima sequer mede a massa ou densidade de cada floco - valores que também são variáveis na natureza. Sendo assim, como é possível que, simulando apenas fração das forças que originam um floco de neve, um computador consiga gerar formas tão variadas quanto o fenômeno físico “original”?

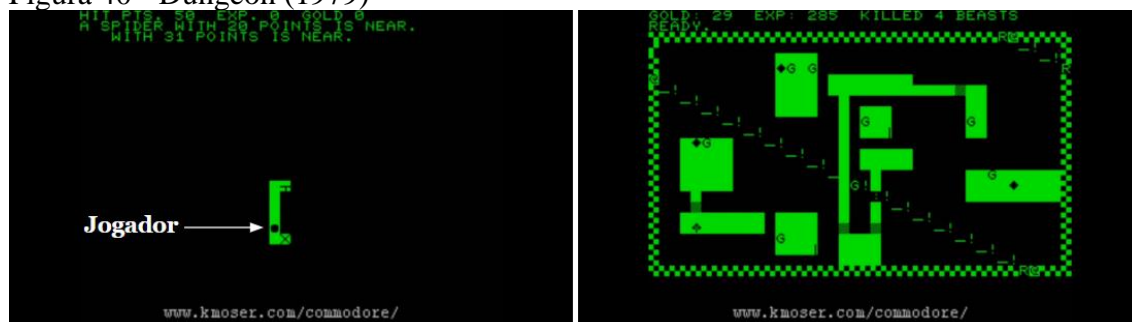
Essa pergunta poderia ser reescrita da seguinte forma: como um sistema pode funcionar de modo complexo se sua modelagem artificial inscreve um conjunto de parâmetros relativamente simples? A resposta está, mais uma vez, na relação entre estabilidade e instabilidade. Certas propriedades dos flocos de neve são estáveis: a simetria hexagonal, por exemplo. Por mais imprevisíveis que sejam suas formações, elas sempre serão “espelhadas” e

⁷⁴ Fonte: <http://softology.com.au/voc.htm>.

em seis lados. Atributos como esse constituem blocos robustos, que não se dissipam apesar da instabilidade das outras conexões que direcionam os traços da figura. Isso quer dizer que “quando as propriedades emergentes de um todo são estáveis, elas podem sobreviver a mudanças em detalhes das interações entre suas partes”⁷⁵ (DeLANDA, 2011, p. 14).

Uma simulação computacional, portanto, pode gerar padrões de repetição e diferença, a partir da identificação de blocos estáveis que se conectam para gerar efeitos mais complexos. Este princípio, que aqui foi ilustrado com o exemplo dos flocos de neve, pode ser relacionado com a geração de fases que se observa em *Rogue*. Cada nível da caverna se cristaliza de acordo com parâmetros que designam uma série de propriedades estáveis. O número de salas em cada andar é fixo: são nove, que se distribuem em três fileiras de três salas cada. O tamanho das salas varia, mas dentro de uma margem limitada. O mesmo vale para os tipos de monstros e itens, bem como posições que ocupam nas salas: estas não são propriedades inertes, mas também não têm liberdade total: existe uma variedade fixa de monstros e itens e um conjunto fechado de posições que estes podem ocupar nas salas. Sendo assim, *Rogue* é um dos exemplos mais icônicos incipientes da *geração procedimental de conteúdo* (Cf. TOGELIUS et al., 2015) nos videogames.

Figura 40 - Dungeon (1979)



No entanto, não foi o primeiro a conectar espaços procedimentais. Se retrocedermos um ano antes, em 1979, um jogo bastante parecido como *Rogue* já havia sido lançado: *Dungeon* (figura 40) apresentava conceitos semelhantes: salas de tamanho variável conectadas por corredores. Assim como em *Rogue*, estes corredores são invisíveis até que o jogador chegue perto deles. É como se o espaço só se mostrasse totalmente uma vez que o avatar já tivesse “visto” e “memorizado” aquele setor⁷⁶.

⁷⁵ Tradução livre de “When the emergent properties of a whole are stable they can survive changes in the details of the interactions between its parts.”

⁷⁶ Aqui vemos um exemplo de como um jogo em terceira pessoa (que vemos nosso avatar ao invés de incorporar sua visão na câmera) pode simular um processo de jogo que costumeiramente se associa a jogos de primeira

Retrocedendo ainda mais um ano, 1978 viu o lançamento de mais dois jogos que também geravam seus mundos de jogo de forma parecida, dando caráter a labirintos de corredores e salas abertas. *Dungeon Campaign*, na verdade, pode ser descrito como um labirinto povoado por criaturas de diferentes tipos. Já em *Beneath Apple Manor (BAM)*, a geração do espaço continua procedimental, e os caminhos se distribuem em bifurcações que abrigam a necessidade de escolher o sentido do movimento. Esse padrão, no entanto, sofre uma reconstrução significativa: não é mais necessário escolher a cada um ou dois passos, como *Dragon Maze*. Os corredores em *BAM* são mais amplos e extensos, diminuindo a proporção de desvios necessários para prosseguir no jogo.

Figura 41 - Beneath Apple Manor (1978)



A maior incrementação feita por *BAM*, no entanto, é na estrutura da “perseguição” entre monstros e jogadores. Na realidade, a dinâmica deixa de ser de perseguição para se tornar uma mistura entre esconde-esconde e pique-pega, na qual tanto o jogador quanto o(s) monstro(s) pode desempenhar o papel de perseguidor ou de vítima – papéis estes que podem se inverter em poucas jogadas. Esse padrão lúdico, assim como *Dragon Maze*, não surge pronto, mas emerge das forças na interface, inclusive da sua conformação espacial. O labirinto de *BAM* também possui uma saída, cujo posicionamento é consideravelmente mais fácil de ser encontrado. Não é essa, portanto, a fonte primária de desafio no jogo. Achar a saída de cada nível não significa muita coisa caso o jogador não se prepare para o que virá adiante.

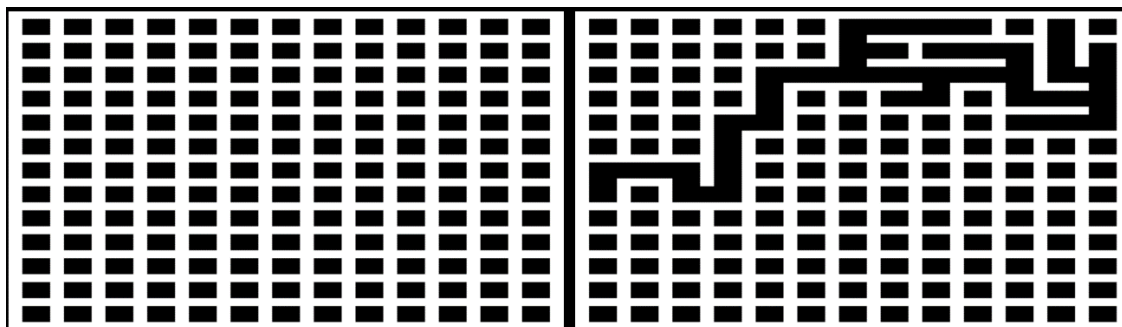
peessoa. O mundo de jogo de *Rogue*, *Dungeon* e boa parte dos *roguelikes* só se mostra ao jogador quando este explora o ambiente e percorre sua extensão. A perspectiva destes jogos, portanto, processa o espaço de modo vagoante. Mesmo que possamos ver uma fase inteira na mesma tela, só possuímos essa perspectiva onipresente porque, antes, vagueamos por sua superfície.

Cada andar da mansão de *BAM* representa um desafio mais complicado: inimigos mais fortes e resistentes, mais armadilhas e menos tesouros, armas e ferramentas. Em *BAM*, o jogador precisa coletar ouro para comprar equipamentos melhores e derrotar monstros para adquirir pontos de experiência, que podem avançar atributos. Essas ações aumentam a resistência do jogador ao dano causado por monstros e sua capacidade de incidir dano aos mesmos – ou seja, permite que o jogador seja menos perseguido e torne-se o agente ativo da perseguição. Essa dinâmica se deve à conectividade das salas ao longo dos corredores, que são menos labirínticas que as de *Dungeon Campaign*. Enquanto nele o movimento do jogador é uma busca incessante pela localização da saída, *BAM* exige outra abordagem. Frequentemente o movimento reto precisa se reorientar à procura de baús escondidos, chaves para abrir portas e vendedores que forneçam armas e curativos.

Em suma, *BAM* é uma implementação antecipada do que seria visto em *Rogue* dois anos depois: corredores interconectando salas, formando um espaço repleto de oportunidades tanto para bons quanto maus encontros. A saída de cada nível exerce uma força determinante, mas que é permeada por uma quantidade de vetores menores e multidirecionais: os itens que podem aumentar a capacidade de sobrevivência do jogador e as intervenções agressivas dos diferentes monstros que povoam seus níveis. É um campo de forças com múltiplos atratores, que tornam o caminho de cada jogador mais sinuoso e menos determinista – mesmo se excetuarmos o fato de que cada partida produz um espaço novo.

Talvez a forma mais simples de explicitar os fundamentos deste funcionamento seja com um jogo lançado ainda antes dos elencados acima. *Dragon Maze* foi desenvolvido por Gary J. Shannon, um programador prolixo autor de uma série de jogos para o computador *Apple II*. Sua intenção com *Dragon Maze* era criar uma máquina de gerar labirintos. O desafio era chegar à saída antes que o “dragão” (representado por um quadrado vermelho) achasse o jogador (representado por um quadrado marrom). Quando ligamos o jogo, vemos uma grade formada por quadrados pequenos posicionados um ao lado do outro. Essa sobreposição de casas não forma um labirinto: é um espaço extensivo e matemático, dono apenas de coordenadas estéreis. O que o algoritmo de Shannon faz é “recortar” as arestas dos quadrados, conectando-os em linhas maiores, que logo dobram para baixo, para cima e para os lados, desenhando caminhos sem saída, paredes protuberantes, enfim: um labirinto.

Figura 42 - Dragon Maze



Momentos do labirinto em formação. À esquerda, o estado inicial; à direita, etapa intermediária do processamento de corredores

O caminho de formação do labirinto não é nem aleatório nem determinista. Ele segue parâmetros fixos (é necessário existir um local de início e um caminho que o leve até a saída) e variáveis (os corredores podem ocupar de 1 a 24 casas). Os corredores são formados quando lados dos quadrados são rompidos, conectando duas casas vizinhas. O software seleciona, a cada passo da formação do labirinto, um lado de um quadrado para ser descartado. Esta operação é feita de modo que sempre existirá um caminho sinuoso que ligue a posição inicial do jogador à saída. Para tanto, a seleção é limitada por regras, que designam quais lados podem ser descartados em uma dada posição do quadrante e do estado geral no qual este mesmo quadrante se encontra naquele momento de sua formação. O computador calcula as opções e seleciona uma, a partir de uma função algorítmica que simula aleatoriedade⁷⁷.

Os labirintos procedimentais de *Dragon Maze* são ocupados por duas entidades – jogador e dragão – que podem realizar apenas um tipo de ação: o movimento. Este pode ser executado em apenas quatro direções, e é sempre restringido pelas paredes dos labirintos⁷⁸. Por *Dragon Maze* ser um ambiente com poucos componentes, podemos facilmente destacar quais eles são e como exibem variabilidade a partir do encontro uns com os outros. As formações procedimentais de labirintos geram caminhos distintos e problemas da mesma ordem, mas com variações de detalhes (de qual lado o dragão está vindo e para qual lado devo/posso escapar?). O encontro repetitivo com a pluralidade de processos é uma experiência distinta do encontro

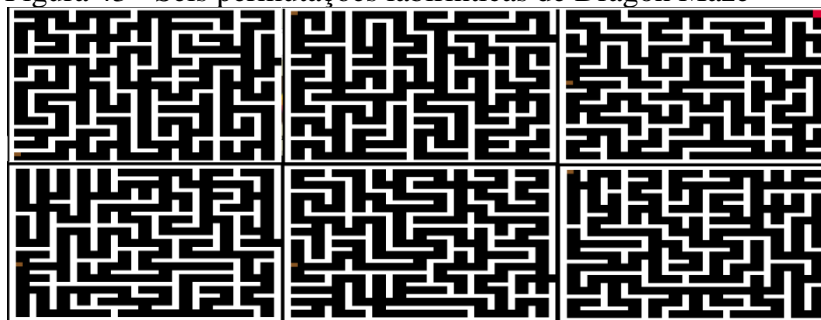
⁷⁷ Rigorosamente falando, computadores não são diretamente capazes de gerar números aleatórios. A aleatoriedade usada em aplicações computacionais geralmente vem de números pseudo-aleatórios, que podem ser obtidos sob diversos métodos. Em *Minecraft*, *Rogue* e a maioria dos jogos que usam geração procedimental de conteúdo se valem de algoritmos que, embora produzam resultados imprevisíveis, podem ser repetidos a partir de uma “semente”, uma sequência de números que representa os caminhos pseudo-aleatórios tomados pelo código do jogo em uma dada instância (Cf. BACKUS, 2017, p. 99).

⁷⁸ Se observarmos as permutações de *Dragon Maze* com atenção, perceberemos que os espaços pretos são os caminhos pelos quais o jogador pode andar, enquanto os brancos são as paredes que restringem o movimento. Também perceberemos que é impossível achar um espaço preto que não faça vizinhança com pelo menos um espaço branco. Isso quer dizer que sempre haverá uma parede impedindo que escolhamos uma das quatro direções de movimento possível.

com um labirinto fixo, desenhado por mãos humanas. O jogo permite a conexão não apenas a um labirinto individual, mas ao campo estruturado de possibilidades virtualizado no sistema.

O modo de formação de labirintos em *Dragon Maze* é análogo à cristalização de flocos de neve. Existe um conjunto de diretrizes fixas que devem se repetir, mas que são adornadas por variações que se atualizam a partir de um campo expansivo de possibilidades. Enquanto os flocos de neve chamam a atenção pela beleza simétrica, a propriedade rara dos labirintos de Shannon é que todos eles são jogáveis. Seu programa é, verdadeiramente, uma máquina de geração de mundos de jogo, que parte de um espaço matemático extensivo até chegar a um emaranhado de caminhos bifurcantes com propriedades intensivas e topológicas.

Figura 43 - Seis permutações labirínticas de Dragon Maze



Os labirintos procedimentais de *Dragon Maze* se encontram com dois agentes – jogador e dragão – que podem realizar apenas um tipo de ação: o movimento em apenas quatro direções, que é sempre restringido pelas paredes dos labirintos. É desse encontro que surge a conexão lúdica do mundo do jogo com o jogador. A conexão é qualitativamente distinta de um labirinto fixo, devido à repetição da experiência: só se tem acesso à multiplicidade de formação de espaços do sistema se a experiência for repetida.

Por *Dragon Maze* ser um ambiente com poucos componentes, podemos facilmente destacar quais eles são e como exibem variabilidade a partir do encontro uns com os outros. As formações procedimentais de labirintos geram caminhos distintos e problemas da mesma ordem, mas com variações de detalhes (de qual lado o dragão está vindo e para qual lado devo/posso escapar?). O encontro repetitivo com a pluralidade de processos é uma experiência distinta do encontro com um labirinto fixo, desenhado por mãos humanas. O jogo permite a conexão não apenas a um labirinto individual, mas a uma máquina de geração de labirintos.

Por mais que o jogador possa apenas escolher a direção de seu movimento, existem outras forças em jogo, capazes de influenciar sua decisão. A primeira delas é um dragão que possui uma capacidade especial: a de fazer o jogador perder. Há ainda uma única casa do labirinto com uma propriedade intensiva igualmente forte: a de fazer o jogador ganhar. Se

quiséssemos analisar *Dragon Maze* como uma união de corredores e paredes, estaríamos perdendo de vista a gravitação exercida por essas duas entidades. Em cada momento do jogo, as escolhas de movimento do jogador são permeadas pela atração positiva da saída e pela repulsão negativa vinda do posicionamento do dragão. O desafio consolida-se em uma propriedade ainda mais reveladora do mundo de jogo: a saída é fixa, enquanto o dragão se locomove.

É impossível compreender o que surge da união de propriedades em *Dragon Maze* analisando-as separadamente. A locomoção do dragão só é variável porque o espaço se forma procedimentalmente. O desafio de achar a saída só é efetivo porque o dragão se move, e vice-versa. As propriedades intensivas e as capacidades das entidades “dragão” e “saída” são protótipos do que veríamos alguns anos depois em *Rogue*. Cada nível da caverna possui seus inimigos e suas saídas. Mas há também portas trancadas, armadilhas de diferentes tipos, escadas que sobem e descem para os níveis inferiores e superiores (quanto mais inferior, mais desafiador). Além disso, ao contrário de *Dragon Maze*, *Rogue* possui áreas abertas, onde é possível se locomover nas quatro direções de seu espaço gradeado. Assim como em *BAM*, isso permite que salas amplas abriguem uma exploração multidirecional. Também possibilita estratégias mais complexas, como a de tentar atrair um monstro até uma armadilha, derrotando inimigos relativamente fortes sem se expor a seus ataques.

Figura 44 - Um dos níveis de *Rogue*, ainda parcialmente inexplorado. O jogador (@) enfrenta uma das criaturas do jogo, um “kobold” (K).



A simplicidade gráfica de *Rogue* foi um de seus maiores aliados. O jogo rapidamente se popularizou nas universidades, que podiam ter os jogos instalados em seus computadores de modo relativamente simples. O padrão de caracteres ASCII usado pelo jogo era largamente adotado por essas universidades, o que significou também que não era necessária nenhuma

adaptação por parte das máquinas: bastava instalar e rodar o programa. O código não era acessível diretamente (os criadores só disponibilizariam o código original em 1986) era possível observar o comportamento das entidades e do mundo de jogo para que outros estudantes em outras universidades (re)criassem suas próprias versões.

Figura 45 - À esquerda, *Moria* (1983) e à direita *Hack* (1985), descendentes diretos de *Rogue*.



Um dos rebentos mais disseminados de *Rogue* nasceu quatro anos depois, em 1983. *Moria* foi inspirado na mitologia literária de J. R. R. Tolkien, nutrindo-se de livros como *O Senhor dos Anéis* e *O Silmarillion* para dar corpo ao mundo de jogo. Foi desenvolvido na Universidade de Oklahoma, sob a liderança de Robert Koeneke. Vários aspectos de seu funcionamento foram baseados em *Rogue*: a composição do cenário por meio de caracteres ASCII, a divisão do mundo de jogo em salas abertas e corredores fechados (figura 45, esquerda) e a movimentação por turnos - o jogador se move e o mundo do jogo move suas “peças” (monstros, armadilhas, etc.), como em um jogo de xadrez.

No entanto, *Moria* não deixou de tomar certas liberdades em relação a seu predecessor. Se em *Rogue* o jogador começava sua campanha já imerso em uma caverna, repleta de monstros e salas escuras, *Moria* ambientava o jogador, primeiro em uma pequena vila. Em termos práticos, essa vila consistia em uma área aberta sem monstro ou armadilha alguma, um oásis de segurança que antecedia o mergulho incerto na caverna. Nesse mesmo espaço, blocos sólidos numerados eram acessíveis, cada um deles uma loja especializada em um tipo de item. Bastava que o jogador movesse o avatar até um dos números, dando acesso a um menu com os itens a venda. As lojas vendiam armaduras (diminuem o dano físico infringido ao jogador pelos monstros e armadilhas), armas (amplificam a capacidade de ataque do jogador), pergaminhos mágicos (cada um dá acesso a tipos diversos de magia de ataque ou defesa, e uma vez usados, desaparecem), comida e poções de cura (recuperam os pontos de vida perdidos pelo jogador), entre outros. O jogador, além de comprar os itens, pode ainda negociar com o vendedor, oferecendo valores menores que os indicados pela loja. A barganha pode resultar em sucesso,

mas pode também ofender o vendedor, que pode banir o herói temporariamente de seu estabelecimento.

Michael Toy, responsável pela geração procedimental de *Rogue*, chegou a admitir que a programação que conectava as peças do espaço jogável de seu projeto era demasiado simplória. Embora essa afirmação possa ser creditada a seu excesso de modéstia, um dos méritos de *Moria* (e outras adaptações de *Rogue*) foi adicionar variabilidade às redes de corredores e salas. Essa variabilidade só se intensificou quando o próprio *Moria* começou a ser modificado e adaptado. Cada um de seus descendentes adicionavam suas próprias contribuições e bifurcações em relação ao mundo de jogo original. *Imoria* (1987), desenvolvido na Universidade de Washington, cria novos tipos de terreno que mudam o modo como o jogador se locomove (áreas inundadas, por exemplo). Enquanto isso, *Umoria* (1987) era encubado em Berkeley, também na Universidade da Califórnia, fazendo poucas alterações em relação ao original, mas transferindo sua programação de para a família de sistemas operacionais UNIX. Entre outros fatores, essa ação possibilitou que o jogo fosse adotado em diversos outros terminais e plataformas (inclusive os populares Atari ST, Commodore Amiga e IBM-PC).

Figura 46 - Angband (1990)



Em 1990, a epidemia dos *roguelikes* já havia cruzado o Atlântico: na Universidade de Warwick, na Inglaterra, Alex Cutler e Andy Astrand desenvolveram o mais famoso herdeiro de *Moria*. *Angband* adicionava um nível de complexidade nunca visto à cristalização dos mundos de jogo dos *roguelikes*. A cidade de *Moria*, que antes contava com apenas seis lojas, agora se

ampliava por uma área equivalente a várias “telas”. Nada comparado ao emaranhado de corredores e salas irregulares que delineavam a fisionomia de suas cavernas (figura 46). Isso incrementava ainda mais a intensidade dos encontros com monstros e inimigos. Era possível, por exemplo, se esconder de um monstro com um campo de visão limitado e alvejá-lo com magias ou flechas sem ser detectado. A pluralidade de monstros e os comportamentos variáveis de cada um adicionava uma camada lúdica agora mais acentuada que nunca: é necessário compreender quais monstros são mais ou menos fortes e quais aparecem nos andares iniciais ou finais. O mesmo vale para itens como armaduras, armas e pergaminhos.

A complexidade do desafio também é algo que começa a fugir do controle em *Angband*. Ao contrário de *Rogue*, que possuía por volta de 26 níveis em sua caverna, e de *Moria* que chegava aos 50, *Angband* gerava nada menos que 100 camadas para sua caverna. Não só isso, a extensão destes níveis era muito maior, o que significava maior presença de itens, armadilhas e monstros; mais espaço para fugas, artimanhas táticas, etc. Já na década de 1990, *Angband* gozava não só de mais poder de processamento para virtualizar sua geração do mundo de jogo, mas também maior capacidade de compartilhamento e intercâmbio entre espécies de computadores e linguagens de programação. Assim, conseguiu gerar um número imenso de adaptações, netos de *Moria* e bisnetos de *Rogue*.

Figura 47 - Zangband (1994):



Porção de mata selvagem (verde) permeada por uma estrada que leva a uma cidade (marrom/cinza), margeada por um terreno coberto de neve (branco)

Zangband é um exemplo de amplificação ainda mais intensa das capacidades conectivas dos *roguelikes*. Baseia-se diretamente em *Angband*, mas conecta à sua vila inicial uma série de estradas que levam a outras vilas, com novos itens a venda e cavernas a serem exploradas. Oferece a capacidade do jogador de coletar itens em uma caverna, vendê-los e usar o dinheiro para se preparar para a perigosa viagem necessária para alcançar uma nova cidade, já que será

necessário passar por áreas selvagens e abandonadas. Por sua vez, *Tales of Middle Earth* (2000) descentraliza ainda mais as áreas principais do jogo. Há uma caverna que precisa ser explorada até o último nível. Mas há também cidades com diferentes inimigos e possíveis aliados, espécies de itens e criaturas, entre outras.

Figura 48 - Tales of Maj'Eyal (2013)



Jogador na cidade de Last Hope (esquerda) e a localização da mesma cidade no mapa-mundi do jogo (direita)

Tales of Middle Earth ganha em 2010 uma reforma total. Abandona o mundo de Tolkien e vira *Tales of Maj'Eyal* (*ToME*). Constitui, assim, um mundo com cidades, oceanos, ilhas, desertos, áreas congeladas, entre vários outros pontos com suas próprias propriedades, capacidades e entidades. Em sua composição espacial, *ToME* poderia lembrar um jogo como *Grand Theft Auto III*, que descrevemos no início do capítulo: possui missões fixas cuja ordem de execução cabe ao jogador decidir. No entanto, é a mobilidade e a instabilidade de suas conexões que incrementa a descentralização dos seus jogadores.

2.3.2 Transformações topológicas: a emergência dos roguelites

A epidemia de influências conceituais *roguelike* logo escaparia do nicho dos programadores universitários e alcançaria estratos mais populares. Em 1992, o Sega Mega Drive recebe o lançamento de *ToeJam & Earl*, um jogo em 16 bits que somava visual *cartoon* às suas fortes inspirações na estética lúdica dos *roguelikes*. Em 1996, a empresa Blizzard extrapola os limites ao criar *Diablo*, um jogo que denuncia sua influências à primeira vista, mesmo que seu diretor não houvesse confessado a paixão pela dinastia *Rogue*. Entre outros jogos que alcançaram certo sucesso no mercado não-underground, destacam-se *Azure Dreams* (1997) e a série *Mystery Dungeon* (1993).

Mesmo estes jogos, porém, acabaram representando soluções de popularidade de um gênero que cada vez mais representava um nicho subalterno da camada mais lucrativa da indústria dos jogos eletrônicos. Enquanto o mercado dos videogames supostamente sofria com

a crise de 1983⁷⁹, *Moria* era lançado e dava início ao período mais produtivo do gênero. No entanto, quando a indústria se recupera e ultrapassa todos os seus recordes nos anos 1990, o contraste com a impopularidade e aparente estagnação dos *roguelikes* parece dar ao gênero um aspecto ainda mais obscuro.

Principalmente a partir de 2007, o mercado de jogos de computadores e consoles experimenta a ascensão explosiva dos jogos independentes. Essa parcela da indústria é, muitas vezes, formada por game designers solitários como Markus Persson, criador de *Minecraft*, ou times pequenos com menos de uma dúzia de integrantes. É nesse cenário que Derek Yu, já renomado por seu trabalho com *Aquaria* (2007) – jogo feito a quatro mãos – decide começar a trabalhar em um jogo que, de forma simplista, pode ser descrito como uma mistura de *Rogue* com *Super Mario Bros*.

O mundo de jogo de *Spelunky*, ao contrário dos *roguelikes* tradicionais, não se movimenta por turnos e não nos é apresentado sob uma perspectiva “aérea”. Vemos nosso avatar assim como em *Super Mario Bros*: projetado em duas dimensões contra um cenário neutro, e nosso movimento é contínuo: apertamos um botão para caminhar e o personagem só interrompe sua marcha quando soltamos a tecla. Logo aprendemos a pular, correr e calcular acrobacias e o uso de nossas ferramentas: uma bomba que explode paredes e uma corda que nos permite erguer-nos a alturas antes inalcançáveis. Reflexos e tempo de reação são fundamentais, assim como em quase qualquer jogo de plataforma. Isso pode nos levar à pergunta: se tudo isso é onde estaria, em *Spelunky*, a influência de *Rogue*?

Roguelikes tradicionalmente encorajam o pensamento cuidadoso e tático. Parar e pensar em cada jogada admite ao jogador o luxo de projetar mentalmente as consequências de cada um de seus atos. Assim como no xadrez, outro jogo conhecido pela sua riqueza estratégica, isso é enaltecido pela divisão em turnos: cada lado do tabuleiro joga de forma *assíncrona* ao lado adversário, ou seja, enquanto um se movimenta, o outro espera. *Roguelikes* fazem o mesmo: o jogador joga e o mundo espera sua vez.

No entanto, seria inocência cogitar que a riqueza tática – em um *roguelike* ou no xadrez – origina-se exclusivamente da divisão em turnos. Para nos livrarmos dessa hipótese basta lembrar que por séculos teorias táticas foram desenvolvidas, executadas e estudadas tendo como laboratório de testes os campos de batalhas travadas em múltiplas guerras. Mesmo nos videogames, já apresentamos a prática de *speedrunning* que envolve um planejamento a desconstrução de um campo estruturado de possibilidades que pode, neste contexto, ser lido

⁷⁹ Conferir: <https://www.independent.co.uk/life-style/gadgets-and-tech/features/why-did-atari-bury-its-et-video-game-9299190.html>.

como um comportamento tático. *Spelunky*, pelo modo como estrutura seu mundo de jogo, possibilita essa mesma abordagem, somando movimento contínuo e riqueza tática.

Figura 49 - *Spelunky* (2009):



À esquerda, uma parte da primeira fase de *Spelunky*. À direita, um quadro ilustrando a geração do labirinto. A letra “A” marca o quadrante onde o jogador começa o jogo. A letra B marca o quadrante onde se acha a saída (as setas em azul indicam o caminho ideal).

O que há de *Rogue* em *Spelunky*, portanto, é a metamorfose contínua de seu mundo de jogo e a preparação tática que essa dinâmica demanda. Ao contrário de *Super Mario Bros.*, cuja composição necessária e imóvel ajuda a colocar o jogador no centro dos processos que desencadeia, *Spelunky* dispõe sua teia de forças sob outras condições. O jogador é designado a começar em um ponto alto da caverna. A saída à qual deve chegar é sempre localizada no ponto mais baixo da fase. Há necessariamente um caminho livre de paredes (figura 49, direita, setas azuis) que leva o jogador do início (quadrante A) à saída (quadrante B). Excetuando-se tais condições invariáveis, no resto de sua composição o jogo tem liberdade para “improvisar” em sua metamorfose.

Na figura 49, à esquerda, vemos a tela inicial da primeira fase do jogo. O campo delimitado pela linha branca é uma das células que compõe a caverna. À esquerda, vemos que uma fase do jogo é composta por 16 células. O algoritmo do jogo seleciona de forma procedimental as conexões entre as células, de modo a gerar uma dentre as várias saídas possíveis. Aqui, o processo é semelhante ao de *Dragon Maze*. Ainda mais variação é adicionada se percebermos que, mesmo dentro de uma célula, há subcomponentes que podem ser sorteados para gerar células distintas. Fora isso, há o posicionamento dos inimigos que também é randomizado (obedecendo certas condições). E, finalmente, a conectividade do jogo permite que inimigos, itens, personagens hostis ou não afetem uns aos outros, gerando processos instáveis e descentralizadores da mesma estirpe dos *roguelikes* tradicionais.

Spelunky, deste modo, dá a partida para a ressurreição dos *roguelikes* (Cf. GARDA, 2013). No entanto, game designers independentes continuaram a tomar liberdades cada vez maiores com a escola na qual se inspiravam – o que nem sempre agradava os fãs mais tradicionalistas⁸⁰. Jogos como *Rogue Legacy* (2013) e *Risk of Rain* (2013) abertamente se colocavam como sucessores da mesma linha “evolutiva” de *Rogue*. No entanto, estes jogos focavam em duas propriedades que se destacam nos *roguelikes*: a morte permanente (permanent death) e a geração procedimental de conteúdo. Não eram baseados em turnos e acabavam deixando de lado a necessidade de cuidado tático. Assim, a ressurreição *roguelike* a partir de *Spelunky* acabou gerando o grupo de jogos comumente chamados de *roguelites* – *roguelikes* menos rigorosos e associados ao que muitos chamam de jogos “casuais”⁸¹.

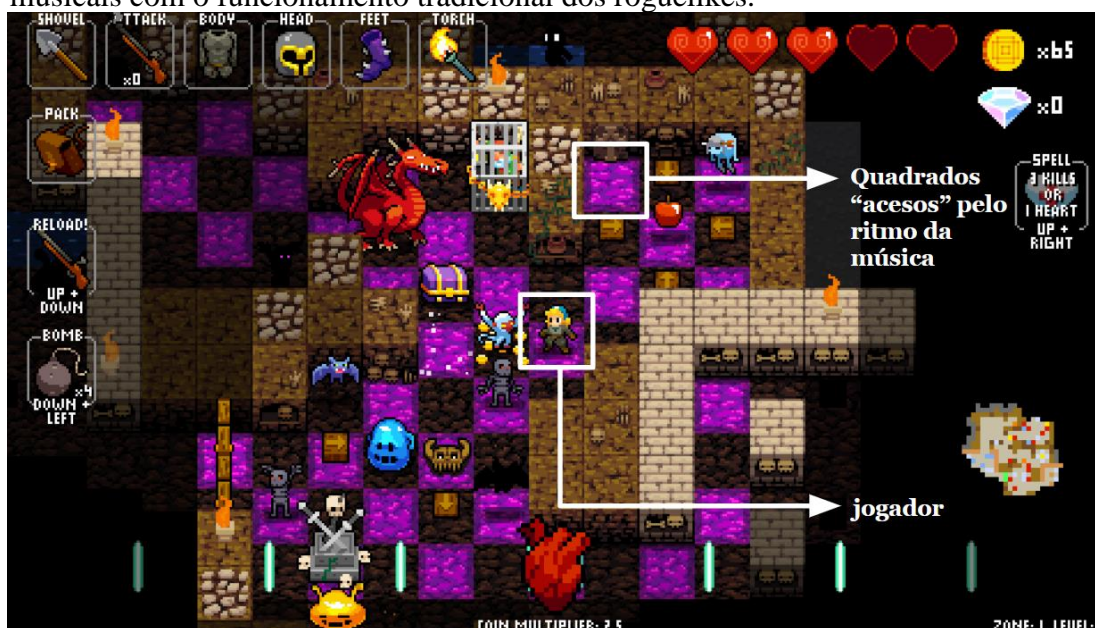
Sejam *roguelikes* “legítimos” ou não, os *roguelites* injetaram ainda mais variabilidade em um gênero que já a possuía de berço. As misturas com outros gêneros tornaram-se inevitáveis: após o sucesso de *Spelunky*, *Rogue Legacy* e outros, muitos mais surgiram, frutos de cruzamentos entre as tradições *roguelike* e outras estéticas lúdicas. Roguelites com perspectiva em primeira pessoa como *Eldritch* (2015) e *City of Brass* (2017) tornaram-se comuns. *Abyss Odyssey* (2014) dá ao jogador – e aos seus inimigos – golpes e movimentos parecidos com os de jogos de luta como a série *Street Fighter*, misturando as sequências de socos e chutes com exploração de cavernas labirínticas.

A mistura mais improvável, no entanto, talvez seja a de *Crypt of the Necrodancer*. À primeira vista, um *roguelike* quase tradicional. No entanto, ao invés de andarmos em turnos, o movimento do avatar deve obedecer ao ritmo da trilha musical do jogo. *Crypt of the Necrodancer* mistura elementos de jogos musicais como *Audiosurf*, *Guitar Hero* e *Dance Dance Revolution*. Nestes jogos, o input dos comandos deve ser sincronizado ao andamento da trilha musical que embala o jogo.

⁸⁰ Em 2008, apenas um ano antes do lançamento de *Spelunky*, a comunidade de desenvolvedores de *roguelikes* se reuniria na Roguelike Delevopment Convention, reunião da qual surgiria a infame “Berlin interpretation”. O documento consiste em uma série de propriedades de valor alto e outras de valor baixo que poderiam ser identificadas em um jogo, de modo a definir se este é ou não um *roguelike*. Originalmente, tratava-se de uma espécie de guia para a catalogação e reconhecimento de videogames do gênero a partir da enumeração de seus atributos. No entanto, com a explosão de *roguelites* que vieram nos anos subsequentes, a definição passou a servir como fonte de disputa para os fãs mais tradicionalistas. Fonte: <http://www.gamesofgrey.com/blog/?p=403>.

⁸¹ Parte dessa fama entre os fãs mais tradicionais pode ser fruto da “exigência” de uma curva de aprendizado profunda e dificuldades altas dos *roguelikes* “verdadeiros” (Fonte: <https://web.archive.org/web/20160919020229/http://insight.ieeeusa.org/insight/content/views/371703>). No entanto, essa concepção parece perder de vista o fato de que o próprio *Rogue*, como lembra Maria B. Garda (2013), foi concebido como um jogo “casual” – mesmo que o termo não existisse tal como é empregado hoje – para ser jogado em partidas relativamente rápidas e muitas vezes em grupos.

Figura 50 - *Crypt of the Necrodancer* (2013): o jogo mistura conceitos de jogos musicais com o funcionamento tradicional dos roguelikes.



Em *Crypt of the Necrodancer*, o chão das cavernas é dividido em grade, e, assim como no xadrez, elas se dividem em duas cores: as casas “acesas” e as “apagadas”. De acordo com o ritmo da música, as casas oscilaram continuamente – as acesas se apagam e as apagadas ascendem – alternando a coloração do chão. O jogador deve sempre estar ocupando as casas acesas e se mover antes que o andamento da música faça a casa se apagar. Caso não consiga acompanhar o ritmo da música e acabe ocupando um quadrante apagado, o jogador é punido: perde poder de ataque e bônus na pontuação do jogo.

Já lidamos antes com a questão do ritmo em jogos como *Minecraft* e *Don't Starve*. Se resgatarmos nossas conclusões para essa breve análise de *Crypt of the Necrodancer*, poderíamos perceber que o andamento da música se acopla ao jogo, dando cadência para as ações dos jogadores. No entanto, a música não é a única fonte de ritmo: ela encontra-se com um conjunto de impulsos que vêm do lado oposto: do mundo de jogo. Mesmo que *Crypt of the Necrodancer* fosse um jogo de turnos contra o xadrez, suas jogadas ainda estariam impregnadas de ritmo. Um jogador de xadrez pode ser mais agressivo ou mais cauteloso, assim como o jogador de um *roguelike*. Em ambos os casos, são as propriedades intensivas das entidades do jogo – e as capacidades que as conectam que determinam as condições para que estes ritmos sejam performados.

De fato, é possível experimentar *Crypt of the Necrodancer* no modo de turnos tradicional: basta escolher jogar com o personagem “Bard”, que faz com que as criaturas só se movam quando o jogador se move. Voltamos para a mesma divisão temporal do xadrez: nosso

adversário nos espera para jogar. No entanto, por mais que a experiência seja qualitativamente distinta, ainda há traços da “dança” original. O dragão vermelho, por exemplo (figuras 50 e 51) ainda se move com o mesmo padrão (1- movimento em direção ao jogador; 2- turno de preparação de ataque; 3- ataque: fogo que atinge todas as casas à sua frente). Esse padrão pode ser lido e contra-atacado com uma sequência de movimentos que os jogadores apelidaram de “*Red Dragon Dance*”.

Figura 51 - Crypt of the Necrodancer: parte da sequência de ações que compõe a dança Red Dragon Dance.

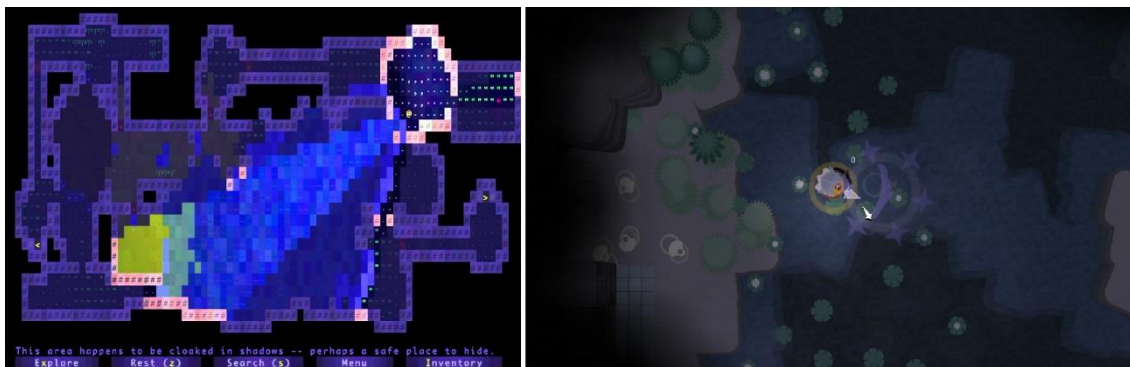


Com a personagem padrão, a dança⁸² deve ser exercida no ritmo da música (quanto mais rápido seu andamento, maior a velocidade que deveremos executar os seus passos). Com o *Bard*, porém, podemos “treinar” os passos sem a pressão descentralizadora da música. O mundo do jogo nos esperará para mover suas peças, como um professor de dança ensinando o aluno a executar uma coreografia. Embora a dança e o ensaio sejam desafios distintos, os passos permanecem os mesmos e, portanto, algo do desafio se mantem em ambos.

Da mesma forma, *Spelunky* adiciona a necessidade de reflexos para saltar, atirar bombas e desviar de projéteis inimigos. Mantém, no entanto, um rastro original de instabilidade e conectividade que o liga tanto a jogos de plataforma quanto aos *roguelikes* mais táticos. Um exemplo semelhante é o do *roguelike Brogue* (2009) que ganhou um remake, *Unexplored* (2017). A diferença do jogo mais novo é que o tempo é contínuo: inimigos se movem sem nos esperar. Além disso, as casas que compõem seu espaço gradeado são também contínuas, assim como as de *Spelunky*: podemos ocupar espaços intermediários entre elas.

Figura 52 - À esquerda, *Brogue* (2009); à direita, *Unexplored* (2017)

⁸² No tutorial do jogo, o jogador passa por uma dessas danças, ao adquirir a boradsword, uma arma que ataca até três monstros simultâneos – desde que estejam alinhados no “tabuleiro”. O jogador precisa aguardar o alinhamento dos monstros que andam em sincronia, para atacar no momento certo, assim que cheguem ao alcance da sua espada. Segue o gameplay de apoio: https://youtu.be/X_F-vZZqIK8?list=PL_rQc0PIUa8OexJwNx9QklWIZUcgva4KC&t=205



Apesar disso, o jogo mantém boa parte das propriedades de sua matéria prima: armas, explosivos, inimigos e entidades em geral possuem dinâmicas muito semelhantes. A diferença está no seu *modo de atualização*, na forma como a relação entre jogador e jogo se desenvolve. Quando esta muda, a experiência sofre uma guinada e se transforma; mas pode transportar seus antigos ritmos para este novo modo de processamento. É o que ocorre de *Brogue* a *Unexplored*: uma mediação, uma tradução das dinâmicas de um modo de processamento a outro, que mantém suas propriedades apesar de transformá-las.

Trata-se de uma demonstração do estudo *topológico* do espaço e do tempo nos videogames. A topologia, afinal, é o estudo matemático das propriedades de uma figura que se mantém estáveis mesmo quando a figura é transfigurada (Cf. DELANDA, 2002, p. 26). Tomando emprestada a ideia e transportando-a para o estudo dos jogos (de modo análogo ao que Manuel Delanda da matemática à filosofia) podemos afirmar que cada ponto da nossa “figura” topológica é uma ação possível dentro do mundo de *Brogue*. Essas ações são condicionadas pelas entidades, suas propriedades individuais e suas capacidades de interação em grupos. Caso quiséssemos mudar qualquer aspecto do jogo – sua aparência visual, por exemplo – *mantendo* as relações recíprocas entre os pontos que tomamos como referência (as ações possíveis no jogo), estaríamos diante de uma transformação topológica de *Brogue*.

Dentre todas transformações topológicas possíveis, *Unexplored* é apenas uma delas. Uma que optou por mudar tanto seu aspecto visual quanto o processamento do movimento espacial e dos turnos, mantendo, porém, parte da teia de relações que estrutura o campo de possibilidades de ação do jogo. Não se trata de falar que são o mesmo videogame: muito pelo contrário. Topologicamente, uma faca pode ser uma transformação de um objeto de metal inofensivo, como uma esfera, desde que (matematicamente falando) não tenhamos adicionado, removido ou sobreposto nenhum de seus pontos originais. A topologia permite justamente que encontremos todas as potências escondidas em um mesmo material, suas direções de transformação e suas propriedades virtuais, que estavam ali o tempo todo, esperando para serem

conjuradas a partir de um mesmo campo de forças – e não é essa atividade – essa busca – que, afinal de contas, chamamos de jogo?

2.3.3 Regimes de instabilidade em *Hack*, *NetHack* e seus herdeiros

De *Rogue* a *ToME*, os *roguelikes* incrementam pouco a pouco suas potencialidades conectivas, que resultam em mundos cujas propriedades se combinam para gerar capacidades (do mundo e do jogador) cada vez mais potentes. No item anterior, demos prioridade à descrição da cristalização de cada mundo de jogo, ou seja, de como suas conectividades são arranjadas de modo a produzir um campo de forças jogável. Neste item, redirecionaremos o foco para o movimento que dá vida a essa conectividade.

Assim como em um caleidoscópio, é a disposição de entidades em associação que gera as “imagens” de um jogo. Giramos e movimentamos o brinquedo para provocar mudanças nesse arranjo, o que não simplesmente faz as formas antigas mudarem de posição. Na verdade, as mesmas entidades individuais dão origens a desenhos inteiramente novos, por meio da disseminação de seus reflexos nos vários espelhos. A primeira vista, poderíamos pensar que jogos como *ToME* se randomizam e geram apenas versões diferentes das mesmas conexões. No entanto, a conectividade de seus mundos de jogo possui intensidade o bastante para gerar movimentos que dão formas a contextos qualitativamente distintos dos anteriores.

Em um caleidoscópio, essa mudança de qualidade é visível diretamente: avistamos padrões simétricos de cores e linhas que nunca se repetirão, por mais que tentemos reproduzir suas condições iniciais originais. Nos mundos de jogo, o desafio de enxergar tais mudanças é agravado pelo fato de que elas são invisíveis, não aparecem como uma imagem fixa já pronta. Talvez por esse motivo, os labirintos de *Dragon Maze* (figura 43) pareçam ser idênticos. No entanto, cada conexão individual de corredores e paredes dá ritmos específicos às ações que neles se desencadearão, cada um com suas próprias nuances e formas.

Como vimos, as únicas capacidades que entram em ação em *Dragon Maze* são as de movimento do jogador e do dragão, somadas à propriedade atrativa da saída e repelente também do dragão. Quando se incrementa, não só a quantidade, mas a qualidade de conexões que jogamos no caleidoscópio dos mundos de jogo, estes desenvolverão processos proporcionalmente mais “coloridos” à medida que entrarem em movimento. Se antes nos concentramos em uma das linhas evolutivas principais de *Rogue* – de *Moria* a *ToME* – podemos, agora abordar uma segunda ramificação: a que leva a *Hack* e *NetHack* (1987).

Hack (figura 53, esquerda). começa a ser gestado em 1982, na escola Lincoln-Sudbury Regional High School, como um projeto educacional⁸³. Brian Harvey ajudou a desenvolver, em 1979, a divisão de computação da instituição, que dava a oportunidade a alunos de experimentarem com terminais simples. Em uma das atividades da turma supervisionada por Harvey, estudantes de ensino médio visitaram a Universidade da Califórnia em Berkeley: o local de nascimento de *Rogue*⁸⁴, que a essa altura já era a grande estrela do departamento local de ciências da computação. Dois estudantes em especial, Michael Thome e Jay Fenlanson encantaram-se a tal ponto com o jogo que, com o apoio de Brian Harvey, decidiram transformá-lo em dever de casa.

Hack viria a público apenas em 1984, quando Andries Brouwer, um matemático holandês que hoje leciona na Universidade de Eindhoven, aproveitou-se dos terminais da instituição de pesquisa CWI (*Centrum Wiskunde & Informatica*) para divulgar sua adaptação do jogo criado pelos estudantes⁸⁵. Isso só foi possível, no entanto, porque durante a produção de seu dever de casa, os amigos decidiram que divulgariam o jogo livremente, sem cobrar qualquer valor. Mesmo os autores de *Rogue* não tinham essa intenção, já que consideravam a possibilidade de tentar ganhar dinheiro com sua criação (Cf. CRADDOCK, 2016). Quase de forma acidental, portanto, os *roguelikes* tornaram-se uma espécie de símbolo da causa do software livre, graças a uma ideia até então praticamente inexistente de alguns estudantes que decidiram lançar seu jogo de graça.

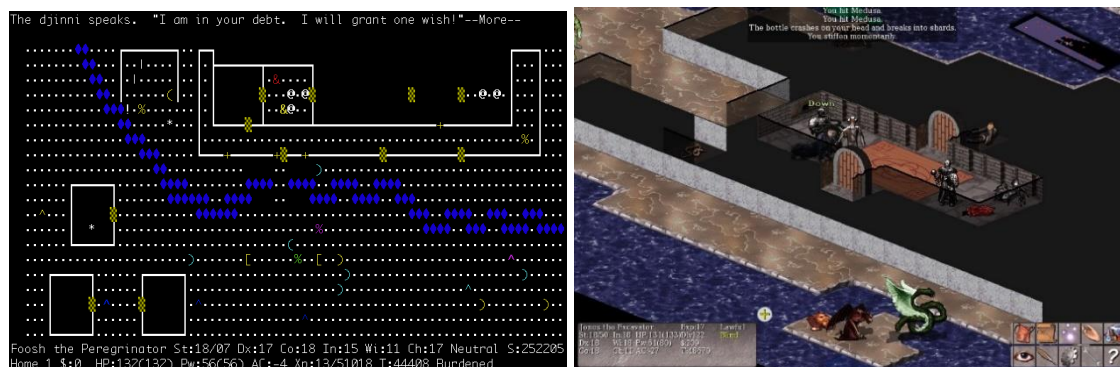
Além dessa ideia, *Hack* traz aos *roguelikes* uma adição incremental de instabilidade: são 52 tipos de monstros com comportamentos e atributos diferentes, o dobro dos 26 que circulavam no original. Quando Brouwer assume o projeto, chega a triplicar essa quantidade, fazendo também alterações na geração de níveis e nas fases finais do jogo, intensificando o desafio de alcançar o Amuleto de Yendor, a chave para a vitória. *Hack*, portanto, planta a semente de um projeto inerentemente colaborativo e cujo produto é uma experiência lúdica complexa e instável. *NetHack*, o seu principal descendente, é o disparador que faz essa semente espalhar e crescer como uma erva daninha.

Figura 53 - NetHack (1987)

⁸³ No sexto capítulo de seu livro “Dungeon Hacks”, David Craddock (2016) relata com entrevistas e pesquisa histórica o surgimento de *Hack*, bem como de outros marcos do gênero, desde *Rogue*, até *Moria* e *Angband*.

⁸⁴ Fonte: <https://www.linux.com/news/train-life-nethacks-papa?theme=print>.

⁸⁵ Fonte: <https://homepages.cwi.nl/~aeb/games/hack/hack.html>.



À esquerda, versão original usando caracteres ASCII. À direita, modificação gráfica “Vulture’s Eye” que substitui os caracteres por representações visuais mais detalhadas.

NetHack é um importante marco na história tanto dos videogames quanto do software livre. Não possui um único autor, mas dezenas. Está sendo desenvolvido ativamente – com hiatos e ressurgimentos – até a data da escrita deste trabalho, em 2018. Para fins da exploração do nosso tema, no entanto, *NetHack* possui uma característica ainda mais flagrante: os personagens que circulam por suas cavernas deixam de ser apenas empecilhos para o progresso do jogador. Eles possuem, também, todas as capacidades que o jogador possui, e podem alterar o mundo de jogo e suas entidades sob o mesmo registro que nós, humanos, podemos.

Se voltarmos aos jogos de plataforma dos anos 1990, e mesmo aos fliperamas dos anos 1980, teremos dificuldade imensa de encontrar um único jogo do qual se possa afirmar a mesma constatação acima. Em *Space Invaders*, as naves só podem se mover em um padrão específico. Em *Pac-Man*, os fantasmas não podem nos atordoar como fazemos com eles. Em *Super Mario Bros.*, cada inimigo possui propriedades únicas, mas nenhum deles possui uma gama de capacidades tão numerosa e potente quanto o arsenal de habilidades do personagem principal: o jogador. A centralização do jogador e o seu alto grau de conectividade relativo ao resto do mundo de jogo pressupõe que ele é uma entidade especial, um agente com potencial de ação superior ao dos outros seres que circulam no ambiente.

É fato que certos jogos apresentam desafios especiais que muitas vezes surpreendem o jogador com sua complexidade e dificuldade. *Megaman X* (1993), por exemplo, emprega um padrão comum em diversos estilos de jogo: os “chefes” (*bosses*) de estágios e/ou fases, posicionados em pontos específicos do percurso do jogo. *Megaman X* possui 8 estágios iniciais, cada um com um chefe final; e a ordem de acesso a cada um dos estágios é escolhida pelo jogador. Quando os chefes são alcançados, estes revelam-se dinâmicos e desafiadores, chegando a intimidar o jogador menos experiente. Ao derrotar o primeiro deles, porém, um prêmio nos é concedido: uma nova arma/poder ofensivo. Cada chefe vencido aumenta nosso arsenal, tornando os desafios subsequentes menos desafiadores. O percurso desenhado para o

jogador, por mais que seja de difícil navegação, ainda é centralizado na sua “evolução”, no seu acúmulo de capacidades.

NetHack é um dos primeiros exemplos de jogo onde essa relação é invertida. De fato, muitos personagens que circulam por suas cavernas subterrâneas são capazes de interferir no mundo de jogo de modos aos quais jogadores iniciantes jamais conseguirão se equiparar. Isto porque esses monstros possuem o mesmo conjunto-base de propriedades que nosso avatar: podem não só se movimentar, atacar e se defenderem, mas também fugir de batalhas, se esconderem e nos impedir de nos movimentarmos. Possuem equipamentos como armaduras, armas e escudos e, se deixarmos nossa espada ou escudo caírem no chão, certos monstros podem catá-las e usarem para seus próprios objetivos. Alguns podem ser subornados para interromperem seus ataques e outros possuem a capacidade de “conversar” com o jogador, transformando uma batalha subterrânea e sangrenta em uma discussão pacífica entre personagens.

Em certas bifurcações de *NetHack*, como *NetHack 4*⁸⁶, a instabilidade da agência dos monstros é ainda mais intensa, gerando conexões não só em direção ao jogador, mas entre os próprios monstros. Por exemplo, ao encontrar um “orc”, uma criatura potencialmente hostil, o jogador pode se surpreender em descobrir que este não está interessado em atacá-lo, mas sim em destruir um elfo, outra espécie de criatura que circula nos corredores do jogo. Nesse caso, o conflito é disparado graças ao comportamento atribuído às duas “classes” de monstro: elfos e orcs se odeiam naturalmente, então, caso o jogador tenha escolhido jogar como um humano, os monstros possuem menos inclinação a agredi-lo, preferindo travarem batalhas entre si.

NetHack e suas variadas versões são laboratórios de teste para detectar lapsos de instabilidade. O jogo mostra que os processos surpreendentes e descentralizantes emergem da conectividade potencial do mundo de jogo. Desde 1987, os muitos programadores que se revezaram no aprimoramento de *NetHack* preocuparam-se em sondar e esmiuçar o jogo em busca de possíveis interações entre itens e monstros, poderes mágicos e tipos de terreno, objetos inanimados e reações químicas: uma miríade de combinações que gera efeitos em cadeia ao mesmo tempo instáveis e lógicos. Uma porta pode ser aberta por chaves, mas pode ser queimada por fogo se for de madeira ou derretida por temperaturas muito altas se for metálica. O jogador

⁸⁶ O projeto de *NetHack 4* nasce em 2015, durante um hiato relativamente prolongado de desenvolvimento da versão principal de *NetHack* (que voltaria a lançar atualizações em 2016). Um grupo de programadores, seguindo o espírito das primeiras apropriações de *Rogue*, resolve bigurcar o projeto e, batiza-o de *NetHack 4* – uma continuação direta da versão 3.4.3 do *NetHack* original. Assim, os programadores tentam “se apossar” do original, adicionado atributos e novas funções para um jogo que, segundo eles, estava estagnado e merecia continuar a evoluir tal como nos últimos 30 anos. Fonte: <http://nethack4.org/faq.html>.

pode equipar em sua cabeça um capacete ou elmo para diminuir o dano causado por ataques. No entanto, caso seu inimigo for uma medusa que o paralisa com o olhar, o jogador pode colocar o item “toalha” no lugar do capacete. A toalha impedirá a visão do jogador, tornando-o imune ao poder da medusa. Ou o jogador pode largar a espada e equipar na mão direita um espelho, refletindo o poder paralisante da medusa contra o próprio monstro.

Certas interações, como a possível entre espelho e medusa, são acrescentadas artesanalmente, ou seja, são codificadas diretamente pelos programadores. No entanto, uma série de interações espontâneas surgem da interação entre processos pré-moldados. Por exemplo, o jogador pode empurrar pedregulhos na água e a pedra passará a servir como bloco de sustentação, permitindo ao jogador que ande sobre ela. Usando essa capacidade, o jogador pode catar pedras pelo mundo de jogo e usá-las para atravessar até um ponto antes inalcançado. Essa é apenas uma das maneiras que *NetHack* oferece ao jogador para percorrer caminhos “impossíveis”.

A influência de *NetHack* no mundo *roguelike* pode ser resumida à riqueza de processos que sua conectividade impulsiona, impondo um desafio aos programadores da comunidade, que passaram a tentar superar a intensidade caleidoscópica de suas cavernas. Alguns jogos conseguem se destacar nessa competição. *UnReal World* (1992) gaba-se de simular o terreno habitado pelos povos nortenhos durante a idade do ferro. O jogador precisa compreender as intrincadas relações entre múltiplas espécies de animais e seus habitats, bem como processos de extração de água potável, alimentos, materiais para confecção de roupas que forneçam aquecimento e estruturas que protejam o jogador da nevasca e da tempestade.

Figura 54 - *UnReal World* (1992):

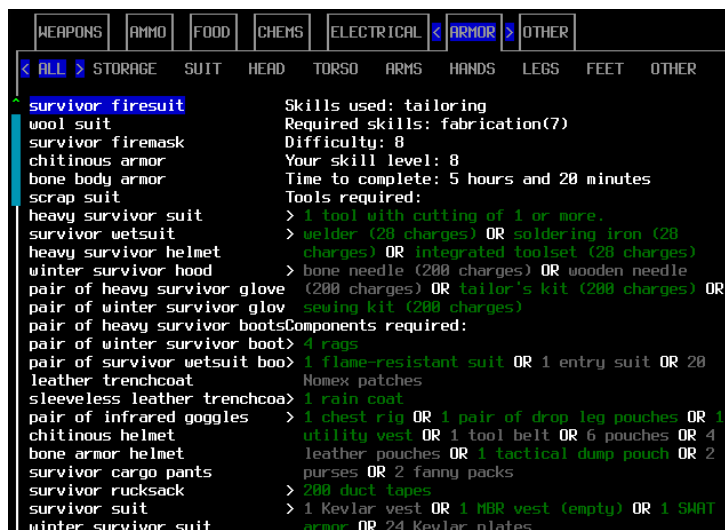


Jogador em uma cabana contruída “a mão”, no processo de fabricação de uma lareira.

Outro candidato, *Cataclysm: Dark Days Ahead* (2013) insere o jogador em um mundo pós-apocalíptico repleto de zumbis e outros monstros. No entanto, o combate deve ser a última alternativa: o jogador terá muito mais sucesso caso concentre seus esforços em coletar desde baterias a sucata e lixo esquecidos na evacuação da cidade. Destes itens isolados, é possível gerar construções e artefatos indispensáveis para a sobrevivência no mundo do jogo. Na figura abaixo (55), selecionamos um único item – uma indumentária à prova de fogo – exemplificando a miríade de potencialidades do mundo de jogo que podem ser extraídas e redirecionadas para a confecção de um equipamento.

É preciso ter desenvolvido a habilidade de “fabricação” e, depois se especializar na sub-habilidade “costura” (as habilidades são desenvolvidas à medida que o jogador desempenha as tarefas no decorrer do jogo). Depois, é preciso ter em mãos as ferramentas necessárias: no caso, qualquer uma que possua a propriedade “cortante” com 1 ponto ou mais. Além disso, serão necessários instrumentos com a capacidade de soldar e tipos determinados de agulha. Finalmente, o jogo elenca os componentes necessários: tanto objetos feitos de pano quanto armaduras e/ou equipamentos mais robustos (como kevlar, material com alta durabilidade) podem ser “reciclados”. Como se trata de um item de proteção avançado, muitos itens de difícil acesso no jogo são necessários para que se construa. No entanto, vale notar que há diversas vias para a confecção: pode-se, por exemplo, usar pedaços de pano velho *ou* vestimentas de pano usadas como componentes.

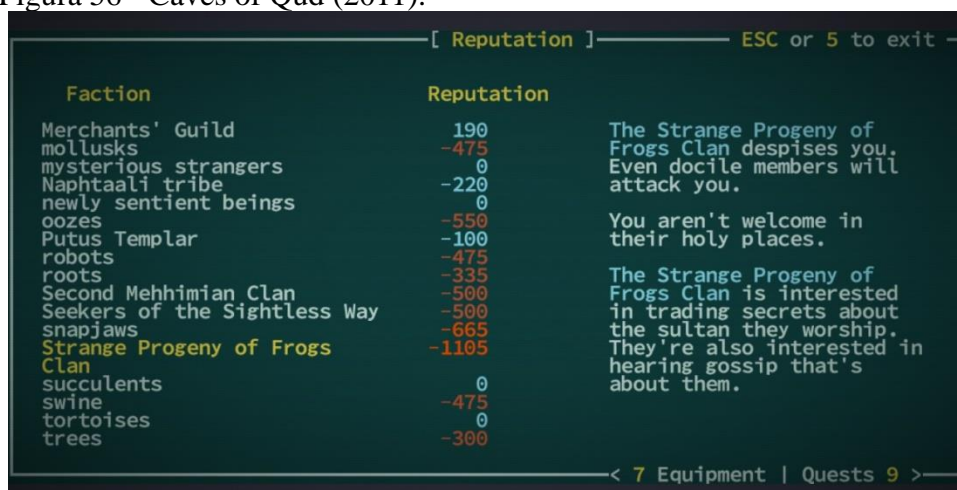
Figura 55 - Cataclysm:



Dark Days Ahead (2013). Menu do jogo onde se executa a confecção de objetos. Aqui, o jogador selecionou a aba “armaduras” (armor), selecionou a indumentária “survivor jumpsuit” (em azul à esquerda). À direita, o jogo exhibe as habilidades e ferramentas necessárias para confeccionar o item.

Como Cataclysm – pelo menos no exemplo acima – concentra-se em exibir sua conectividade, há de se ressaltar que outros jogos investem na criação de grupos de entidades que, de modo alheio ao comportamento do jogador, geram processos instáveis. *Caves of Qud* (2011) é um exemplo: o jogo se passa em um mundo pós-apocalíptico, milhares de anos após do evento desastroso que encerrou com a vida na terra tal qual conhecemos. O jogador explora, portanto, uma sociedade mutante e semi-estável cuja cultura é completamente alienígena. Água é usada tanto como fonte de hidratação como moeda e item ritualístico. Animais de várias espécies e até plantas possuem inteligência elevada e podem se comunicar com o jogador.

Figura 56 - Caves of Qud (2011):



tela onde o jogador pode checar sua reputação com cada uma das facções do mundo de jogo. À esquerda, o nome das facções; no centro o valor positivo ou negativo da reputação e à direita a descrição da facção selecionada.

Aí entra o ponto que nos interessa no jogo: seu sistema de facções. Cada uma das espécies do jogo constitui uma facção. Outras facções são preenchidas por entidades de várias

espécies – de plantas carnívoras a caranguejos gigantes. Algumas aparecerão para qualquer jogador, outras são geradas procedimentalmente pelo código do jogo. No entanto, o que as une é o sistema que controla as relações entre as facções. Na figura 56, acima, vemos o menu que exhibe a reputação (negativa ou positiva) do jogador com cada um dos grupos. No entanto, não é apenas o jogador que é amado ou odiado: cada uma das facções também possui suas inimizades e alianças. Isso possibilita situações instáveis, como o jogador chegar a um campo de batalha durante o enfrentamento entre dois sectos rivais. Tal evento pode ser muito proveitoso, já que se uma ou ambas tribos fossem inimigas do jogador, este sofreria ataques instantâneos. No entanto, o sistema de facções é instável para o bem e para o mal: caso o jogador demonstre sinais de aliança com a facção *x*, a facção *y* (que odeia *x*) reconsiderará sua reputação – o que quer dizer que o jogador poderá ser muito bem-vindo à próxima vila que visitar ou causar uma guerra imediata⁸⁷.

Existe, portanto, uma teia invisível de associações que entremeia as entidades de um *roguelike* como *UnReal World*, *Cataclysm* ou *Caves of Qud*. Cada entidade parece simples: é manifestada visualmente como pequenos caracteres, símbolos ou desenhos – cada um ocupando uma casa na grade geométrica do jogo. À medida, porém, que se acresce uma determinada gama de propriedades a essas entidades, temos a chance de pará-las de modo a gerar capacidades inéditas: começamos a vislumbrar que o mundo do jogo é mais do que a soma de suas partes. *Rogue* é mais do que a união de caracteres ASCII do sistema Unicode e *NetHack* é mais que um mosaico de *sprites*.

É o que Mark Johnson (2015), pesquisador e desenvolvedor do *roguelike Ultima Ratio Regum*, argumenta ao afirmar que as convenções codificadas dos *roguelikes* não necessariamente ajudam o jogador a compreender o que está acontecendo. O uso recorrente de caracteres e cores em jogos diferentes (como um “D” vermelho para representar um dragão) pode ter o efeito inverso e *atrapalhar* o jogador, segundo Johnson. A explicação é que, principalmente nos *roguelikes*, um mesmo significante (o “D” vermelho) pode tomar parte em complexos sistemas de códigos. Ou seja, as capacidades e propriedades do dragão vermelho

⁸⁷ O sistema de facções torna-se ainda mais instável quando se coloca na equação a atuação de outros sistemas do jogo, como a geração procedimental de entidades históricas e sua importância para cada facção. Os game designers documentam nuances desse sistema, mostrando que podem simular a emergência de um líder importante para uma facção, que deixa rastros de sua história no mundo de jogo. Pode-se encontrar objetos pintados com imagens de seus feitos heroicos, ou mesmo a espada ou escudo usados pelo personagem lendário em uma batalha importante. Caso faça uso aberto destes itens, o jogador terá sua reputação melhorada com as facções que veneram a figura histórica – e, naturalmente, será odiado pelas facções que a desprezam (Cf. GRINBLAT; BUCKLEW, 2017).

mudam de jogo para jogo, o que implica que o próprio signo se torna diferente (mesmo que ainda vejamos o mesmo “D” vermelho na tela).

A compreensão da teia de afetações que se desenvolve durante o jogo só pode ser compreendida ativamente, de modo que o jogador derive da experiência as forças que compõem cada entidade do mundo de jogo. Este capítulo procurou demonstrar parte do comportamento das teias que operam em jogos de sobrevivência, em *Rogue*, seus antepassados e descendentes, bem como nos *roguelites*. Com isso objetivamos criar um retrato das propriedades descentralizadoras que emergiram na indústria dos videogames ao longo de sua história: o movimento da exploração à adaptação; os ciclos de jogo que forçam o jogador a romper com a repetição e os regimes de conectividade e instabilidade que põe forças divergentes em confronto com o jogador. No próximo capítulo, selecionaremos jogos específicos para investigar como uma série de fluxos de descentralização se unem em um único ambiente de jogo.

3 ULTRADINÂMICAS: UM MERGULHO EM *ULTIMA RATIO REGUM* E *DWARF FORTRESS*

Os dois capítulos anteriores consistiram em aprofundamentos teórico e histórico, respectivamente, na ideia de descentralização como modo de comunicação jogo-jogador. Resta, por último, um mergulho empírico que consista em um olhar descentralizado. Afinal, tanto na teoria do primeiro capítulo quanto na sondagem histórica do segundo, lançamos um olhar amplo sobre o objeto, o que nos coloca em uma posição privilegiada: centralizada. É necessário complementar este estudo com uma perspectiva que reflita a posição do jogador diante dos fluxos de instabilidade e conectividade que o descentralizam. Em outras palavras, cabe a este trabalho mergulhar, ele mesmo, em mundos de jogo que descentralizem o ato de pesquisa.

Este último capítulo tem como objetivo exatamente isto: selecionar dois jogos que incorporem as ideias discutidas até aqui – um sob o ponto de vista da conectividade e o outro da instabilidade – e que sirvam como campo para nosso estudo. Embora a perspectiva tenha mudado, a metodologia permanece a mesma: estamos em busca de propriedades e capacidades que engajem a sensorialidade do jogador de forma descentralizadora.

Iniciamos esta missão com uma última reflexão teórica acerca da natureza da descentralização no que tange a atividade cognitiva do jogador. Estamos falando, afinal de contas, de dinâmicas intensivas e complexas de sistemas computacionais, bem como de sua projeção em estímulos, que precisam ser capturadas e manipuladas pelo aparelho sensório-motor do jogador. O que surge do emparelhamento dessa dinâmica com nossos sentidos?

Como base para essa investigação, lançaremos mão de três referências principais: a ideia de *meio de interioridade* e *meio de exterioridade* na filosofia do jogo de Gilles Deleuze e Felix Guattari nos mostrará que existem pelo menos dois tipos de regras: uma que encoraja a estabilização, voltando-se para o interior de um objeto ou entidade; e outra que dissipa processos de instabilidade, prolongando-se para o exterior e agitando outras forças do mundo do jogo. Em seguida, associaremos essa constatação à uma outra, dessa vez do filósofo Henri Bergson: perceberemos que mesmo atividades lógicas, como a de jogar de xadrez, debruçam-se sobre uma densa camada de *esquemas dinâmicos* que aglutinam sensações e impressões sensoriais, condicionando decisões e operações racionais.

Munidos destes conceitos, executaremos nossa ida a campo – em um sentido que não se limita a uma metáfora. Encararemos os mundos de *Ultima Ratio Regum* e *Dwarf Fortress* como terrenos concretos a serem desbravados, assim como é feito por qualquer jogador que queira se ambientar nos seus ecossistemas. Também como um jogador faria, tentaremos contextualizar

as forças aparentemente caóticas e incompreensíveis destes jogos em algo estável e passível de interação.

3.1 Esquema dinâmico: dois tipos de regras e suas reverberações sensoriais

O fim dos anos 1970 trouxeram consigo um presente escondido: nascia a era dos *easter eggs*. Durante décadas, game designers enterravam em seus mundos de jogo mensagens ocultas, passagens secretas e às vezes, atrás destas, áreas inteiras aguardando serem encontradas pelos exploradores mais rigorosos. Em uma de suas famosas palestras, geralmente concedidas na Game Developers Conference (GDC)⁸⁸, o programador Brian Moriarty – um dos nomes por trás da influente Lucasfilm Games, medita a respeito desta tradição. *The Secret of Psalm 46* é uma intrincada reunião de ideias a respeito de segredos codificados, a busca desenfreada que costumam provocar e sentimento que motiva a busca: “awe” – uma palavra cujas muitas traduções para o português falham em transmitir sua multiplicidade significativa⁸⁹.

Se os *easter eggs* ainda costumam ser utilizados na indústria, talvez seja muito mais por um aceno cortês ao passado do que por um real interesse do público em caçá-los. Talvez a rápida disseminação destes segredos na internet tenha tornado a busca menos intrigante e, por vezes, inócua. Ou talvez parte dessa mudança se deva a uma diferença estrutural: pelo menos nos jogos de sobrevivência, esconder mensagens ou tesouros faria pouco sentido. No espaço amplo e na instabilidade alta, eles se perderiam como um grão de areia no deserto.

O que não quer dizer que esse tipo de jogo não tenha a oferecer a sua própria espécie de mistérios. Como apresentado brevemente no primeiro capítulo, as estratégias por vezes chamadas de *spelunking* ou simplesmente *mining*, em *Minecraft*, consistem em parâmetros de ação “achados” pelos jogadores. Não se trata, porém, de um achado com finalidade pré-definida, como a de um código secreto cuja mensagem precisa ser (de)cifrada: esse tipo de achado floresce da atividade espontânea das entidades do jogo. As estratégias inventadas pelos jogadores, assim como no xadrez, por exemplo - são fruto da riqueza lúdica das propriedades e capacidades exercidas pelas entidades do jogo.

Essa pluralidade de combinações, embora se origine de interações complexas estudadas inclusive em modelos matemáticos, pode ser rastreada até um princípio relativamente simples. Se essas entidades de um mundo de jogo são construídas como as paredes e buracos de *Super*

⁸⁸ Fonte: <https://www.youtube.com/watch?v=u0OY1RDe8Yg>.

⁸⁹ As traduções costumam flutuar entre o sentido de “sublime”, “maravilhoso” e uma conotação de “assombro”, “espanto”.

Mario Bros., suas propriedades voltam-se primariamente para seu *interior*. Tentam fechar-se ao máximo contra a instabilidade do ambiente “de fora”, de modo a desempenhar sua função na arquitetura do mundo de jogo (colisão, sustentação, etc.) Quando achamos uma fresta entre elas – tal qual os speedrunners que desconstroem as regras – é porque, geralmente, essa estrutura está comprometida.

Porém, quando as propriedades das entidades se voltam primariamente para o seu exterior – para as conexões que podem realizar –, a instabilidade é mais uma aliada que uma inimiga. Os comportamentos “achados” passam a ser um resultado natural dessa exterioridade⁹⁰. Sendo assim, abordaremos este fenômeno de modo a dar concretude ao modelo de como funcionam nos videogames e, por conseguinte, de como descentralizam o jogador. Começaremos a empreitada a partir de uma sistematização filosófica: um ensaio teórico sobre jogos encontrado em *Mil Platôs*, de Deleuze e Guattari (1980). Escolhemos começar pela teoria para que, após sua exposição, possamos usar os casos práticos para dar concretude ao pensamento descrito por ela.

Especificamente, nos preocuparemos com as definições de “meio de interioridade” e “meio de exterioridade”, conceitos empregados para distinguir dois jogos milenares: xadrez e Go (DELEUZE e GUATARI, 1980, p. 352). Enquanto o xadrez tornou-se ubíquo o bastante na cultura ocidental para que tenhamos nos acostumado a escrever seu nome com letra minúscula, Go é um caso distinto. Trata-se de um jogo também de origens antigas e desconhecidas, jogado em um tabuleiro dividido por linhas que formam quadrados.

Figura 57 - Go (esquerda) e xadrez (direita).

⁹⁰ Aqui, chegamos de uma vez por todas à questão da emergência – ou pelo menos do modo como é abordada nos game studies –, já que os resultados da conectividade entre entidades produzem virtualidades mais variáveis, complexas e dinâmicas que a simples lista de regras que as torna viável.



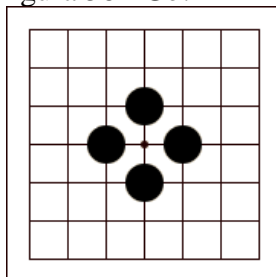
As semelhanças terminam e começam as diferenças: enquanto posicionamos as peças do xadrez no centro dos quadrados, as “casas” onde as peças descansam em Go são definidas pelo encontro das retas verticais e horizontais. Outra diferença sutil é sua adaptabilidade: o tabuleiro profissional possui 19x19 casas (figura 57, esquerda), o que eleva a intensidade de conexões possíveis, tornando o jogo mais complexo. Crianças e iniciantes, porém, costumam começar jogando em tabuleiros menores, de 9x9 e 13x13 casas, o que permite conhecer as regras em um território simplificado. Por mais que se tenha visto versões diferentes do xadrez no passado e no presente, essas adaptações exigem uma mudança brusca nas regras e no comportamento das peças. O jogo é muito mais rígido, assim como uma fase de *Super Mario Bros.* é mais rígida que um mundo de *Minecraft*. O comportamento das peças é, precisamente, o que servirá de âncora para que Deleuze e Guattari percebam a diferença entre os dois tipos de jogo:

As peças do xadrez são codificadas; possuem uma natureza interna e propriedades intrínsecas das quais seus movimentos, situações e confrontações são derivadas. Possuem qualidades; o cavaleiro permanece um cavaleiro, o peão um peão, o bispo um bispo. Cada uma é como o sujeito de um enunciado, dotado de poder relativo, e esses poderes relativos se combinam em um sujeito de enunciação, ou seja, o jogador de xadrez ou a forma de interioridade do jogo. Peças de Go, em contraste, são pastilhas, discos, unidades aritméticas simples, e possuem função apenas anônima, coletiva ou de terceira pessoa [...] sem propriedades intrínsecas, apenas circunstanciais. (DELEUZE e GUATARI, 1980, p. 352-353).

O trecho acima, embora breve, evoca uma série de questões relativas não só aos jogos que põe em tensão, mas ao esforço filosófico desenvolvido no livro *Mil Platôs*. A comparação entre o “jogo de estado” xadrez e o “jogo nômade” Go remete a uma tensão política entre máquina de guerra e Estado, entre forças de desterritorialização e reterritorialização. Limitemo-nos a demonstrar essa lógica apenas em como se demonstra nos dois jogos. Os autores encontram no *meio de interioridade* de cada peça a força que torna xadrez um jogo de relações intersubjetivas, ou seja, entre entidades com propriedades e capacidades atribuídas a priori.

Enquanto isso em Go, é o *meio de exterioridade* de cada peça que as coloca em um campo alheio à da subjetivação, ou seja, um campo onde as forças se constituem e desconstituem de modo dinâmico, a cada turno.

Figura 58 - Go:



Quatro peças negras cercam um território, criando um olho. Posicionar uma peça branca nesse território torna-se ilegal ou inviável, já que peças cercadas são capturadas. O território em questão passa a ser das peças negras, a não ser que as brancas cerquem o olho.

Isso se demonstra evidentemente na dinâmica do jogo: no xadrez, é necessário mover peças e ocupar territórios maiores, com mais casas. Em Go, isso significa pouco ou quase nada: é preciso conectar peças em casas adjacentes, gerando entidades maiores chamadas de “grupos”. “Olhos”, por exemplo, são grupos de peças que cercam uma única casa vazia por todos os lados, impedindo esta porção do tabuleiro de ser ocupada pelo adversário⁹¹. Olhos podem se espalhar pelo tabuleiro e ser conectados a outros olhos ou grupos de peças, gerando por exemplo as estruturas “duplo-olho”, “falso olho”, “dragão”, “dragão de duas cabeças”, entre muitas outras. Essas estruturas não são apenas padrões geométricos desenhados pelas posições das peças: são forças topológicas capazes de exercer influência, balancear tensões e aumentar a “espessura” das peças brancas ou pretas.

Em outras palavras, as propriedades das peças de Go, além de circunstanciais, não são pré-estabelecidas – ou, para usar o termo de Deleuze e Guattari, “codificadas”. Elas são produto da interação de nível inferior que se desdobra e prolonga-se à camada superior. No xadrez, as

⁹¹ As regras de Go são ainda mais simples que as do xadrez. Resumidamente, jogadores alternam turnos, posicionando peças individuais nas casas do tabuleiro. Jogadores acumulam território ocupando casas e cercando territórios com seus grupos. Se um jogador cercar peças individuais ou grupos adversários, essas peças serão capturadas. O jogador com mais território vence. Portanto, Go também incentiva a acumulação extensiva de territórios, a diferença é que o método para se chegar a tal acumulação precisa envolver o entendimento e manipulação de zonas de instabilidade.

propriedades de cada peça são pré-estabelecidas e é compreendendo a composição de relações entre estes códigos que se decifra a mensagem em cada um dos estados do tabuleiro. Exatamente por isso, pede que o jogador *ocupe* espaço, enquanto Go pede que o jogador *influencie* o espaço, sem necessariamente dominá-lo. Um jogador inexperiente pode achar que está ganhando uma partida por possuir mais extensão de território, apenas para descobrir no próximo turno que um de seus grupos mais numerosos possuía um ponto fraco que podia ser obliterado com uma única peça do adversário.

É verdade que há, no xadrez, um nível superior de estratégias, usos inventivos das propriedades codificadas das peças, e é aqui que é possível perceber que não se trata de um juízo de valor, mas sim de tendências. Enquanto o xadrez tende a gerar relações estruturais geometricamente rígidas, Go tende ao inverso: a transformar o tabuleiro em uma zona de instabilidades, onde agrupamentos de peças precisam ser recompostos e movimentados para se nutrir de potências voláteis, que podem ser desfeitas em poucos turnos.

Isso não quer dizer que não possa germinar invenção entre as brechas da estrutura rígida do xadrez, ou que Go não incentive certas estratégias calcadas na estabilidade e na ocupação de territórios. Assim como é possível quebrar as regras e atravessar uma parede em *Super Mario Bros.*, grandes enxadristas fizeram ruir estratégias antes dominantes, tornando-as obsoletas graças às suas inovações⁹². É compreensível, porém, que devido à sua tendência rígida, o xadrez tenha passado a ser visto como um desafio fechado. É o que a própria Virginia Kastrup acaba propondo ao criticar o modelo clássico de aprendizagem e cognição (representado pelo xadrez): “As ações cognitivas prototípicas são jogar uma partida de xadrez ou resolver um problema de cálculo matemático, ambas desempenhadas pelo exercício de regras que delimitam *a priori* um conjunto fechado de soluções possíveis.” (KASTRUP, 2007, p. 166).

O local de argumentação de Kastrup é justamente o de confrontação com as teorias clássicas da cognição, em especial a teoria computacional da mente, que imagina a cognição humana como uma pirâmide, no topo da qual estariam habilidades intelectuais (cf. OLIVEIRA, TIMPONI e MAIA, 2012). Por associação, nomes proeminentes desse campo acabam por associar o xadrez a esse tipo de tarefa, como se o ser humano que joga xadrez agisse exatamente do mesmo modo que um computador que realiza a mesma tarefa. Falamos aqui, no entanto, do

⁹² Mesmo em um jogo com séculos de idade, as inovações continuam acontecendo. Uma das mais recentes é a ressurreição de uma tática chamada de Abertura do Bispo, ou posteriormente de Defesa Berlinesse (*Berlin Defense*). A abertura data do século XVI, e era considerada inofensiva até Vladimir Kramnik desferi-la contra ninguém menos que Garry Kasparov, um dos maiores enxadristas da história. O resultado foi o título de campeão mundial de 2002 para Kramnik, e a disseminação do uso da tática até os dias atuais (2018, ano de conclusão deste trabalho).

pontod e vista dos game studies, em especial do modo como cada jogo mobiliza sensorialmente seu jogador. E, deste ponto de vista, nada poderia estar mais equivocada que associar o processo de jogar xadrez à tarefa de computar dados. E a primeira fonte para questionarmos essa ideia é justamente uma das matérias-primas do pensamento de Kastrup: Henri Bergson.

Em seu ensaio “O Esforço Intelectual”, Bergson confronta a questão dos jogadores de xadrez “às cegas”: enxadristas capazes de desafiar dois ou três jogadores ao mesmo tempo, sem que se percam do andamento de cada partida individual. A pergunta de Bergson é, portanto, que tipo de esforço é esse feito por esses jogadores? O autor recorre a pesquisas da psicologia para chegar à seguinte conclusão:

O que está presente no espírito do jogador é uma composição de forças, ou melhor, uma relação entre potências aliadas ou hostis. O jogador refaz mentalmente a história da partida desde o início. Ele reconstitui os acontecimentos sucessivos que conduziram à situação atual. Ele obtém, assim, uma representação do todo que lhe permite, em um momento qualquer, visualizar os elementos. Esta representação abstrata é, aliás, una. Ela implica uma penetração recíproca de todos os elementos uns nos outros. O que o prova é que cada partida aparece ao jogador com uma fisionomia própria. (BERGSON, 2006, s.p.)

A conclusão curiosa à qual chegam os psicólogos citados por Bergson é que o jogador não precisa ter em mente um “espelho interno” de cada partida, jogando-a mentalmente. Em outros termos, não se trata de um esforço de imaginar cada posição de cada peça individualmente, mas sim de captar a fisionomia própria de cada uma delas. Bergson chama a essa fisionomia de *esquema dinâmico*.

O esquema dinâmico é como um ponto sobre o qual convergem as dobras de tudo que se precisa saber sobre um determinado processo. Embora talvez nem todos sejamos enxadristas talentosos, todos nós decoramos passagens de livros ou poemas, entre outras sequências de palavras. O que Bergson alega é que essas tarefas podem envolver a decoração exaustiva de cada palavra individual, como se cada uma tivesse o mesmo valor dentro da geometria das frases. No entanto, há uma estratégia alternativa: percebemos que cada frase possui substantivos que, se decorados, facilitam a lembrança dos pronomes e preposições que os acompanham. É como se esses substantivos puxassem consigo outros “dados” sobre a frase.

O mesmo acontece com um jogo de xadrez: se o jogador quisesse decorar cada posição de cada peça sua e do adversário, estaria depositando demasiada energia em uma tarefa que de pouco adiantaria – a não ser que fossemos computadores. No entanto, a posição de uma única peça importante – como um bispo ou torre – pode trazer consigo toda uma história da partida que imediatamente “ressoa” no aparato cognitivo humano, como quando ouvimos uma melodia familiar ou imaginamos a fisionomia de uma pessoa. Não precisamos percorrer mentalmente

cada parte da fisionomia de alguém se quisermos lembrar de sua face: a imagem vem como uma impressão só, qualitativa e indivisível. Para o jogador de xadrez experiente, acontece o mesmo.

3.2 Ultraconectividade em *Ultima Ratio Regum*

O esquema dinâmico, portanto, é aquilo que o jogador apreende quando age em comunhão com as forças intensivas do jogo, aquelas que não se manifestam de forma quantitativa e espacialmente, mas sob aspectos qualitativos e temporais. Cada uma dessas forças intensivas, é verdade, se estende no tabuleiro e desenha um padrão geométrico de casas vazias e ocupadas que possui extensão definida. Porém, é na topologia de forças que cada peça potencializa que se esconde a real conexão do jogador de xadrez com a multiplicidade de caminhos aberta pelo jogo.

É possível imaginarmos o mesmo dos videogames? Percebemos cada jogo como uma sequência mais ou menos rápida de estímulos que se sucedem, instaurando ritmos e ciclos mais ou menos longos. Quanto mais esses ciclos contornarem o jogador, maior nossa influência sobre o seu desenvolvimento; quanto mais nos distanciarmos de seus centros, porém, maior será nossa necessidade, como jogadores, de procurar forças de estabilização e conexão que nos permita tornar a influenciar de maneira significativa o mundo de jogo.

Sob esse ponto de vista, é possível que a mesma premissa do esquema dinâmico bergsoniano sirva de parâmetro experiência nos jogos dos quais estamos tratando. Se nos jogos onde o jogador tende a se aproximar do centro da dinâmica a estabilidade das forças dificulta a incorporação de uma agência divergente, os jogos descentralizantes tendem ao contrário. Se muito, a própria magnitude dos graus de instabilidade e conectividade é que servirão de empecilho para a ambientação do jogador: não haverá a mão cuidadosa do game designer guiando sua trajetória ao longo do percurso de adaptação à dinâmica. E, quanto mais intensa esta for, mais traumático será o processo adaptativo.

O melhor laboratório para testar essa hipótese é exatamente um jogo cujas conectividade e instabilidade se mostrem as mais intensas possíveis. Começaremos pela conectividade, com o jogo *Ultima Ratio Regum* (2011)⁹³, ou *URR*, desenvolvido solitariamente pelo programador

⁹³ Parte das informações obtidas sobre este jogo foram obtidas por meio da leitura do blog de seu desenvolvedor (www.ultimaratioregum.com) e de conversas via e-mail. No entanto, a maior parte do só foi incluído no texto final foi extraída da versão atual do jogo até o momento da escrita do trabalho: 0.7. Uma pequena quantidade de dados foi obtida de uma versão de testes de *URR* 0.8, da qual participei como avaliador voluntário entre

e pesquisador Mark Johnson. Mark começa seu projeto com o objetivo de gerar procedimentalmente nada menos que um sistema solar, cuja estrela é orbitada por planetas e satélites rochosos, congelados ou vulcânicos⁹⁴. Dentre os astros, encontra-se uma variação da Terra: um planeta povoado por vegetações e oceanos extensos – mas também por seres humanos. Continentes são recortados por fronteiras: os humanos agrupam-se em feudos, assentamentos tribais ou fortalezas nomâdes.

Figura 59 - Mapa territorial de um dos mundos gerados por Ultima Ratio Regum



As áreas mais extensas representam regiões feudais, cada uma com uma cidade capital. Certas civilizações dominam mais de um território, tal qual é possível observar na legenda.

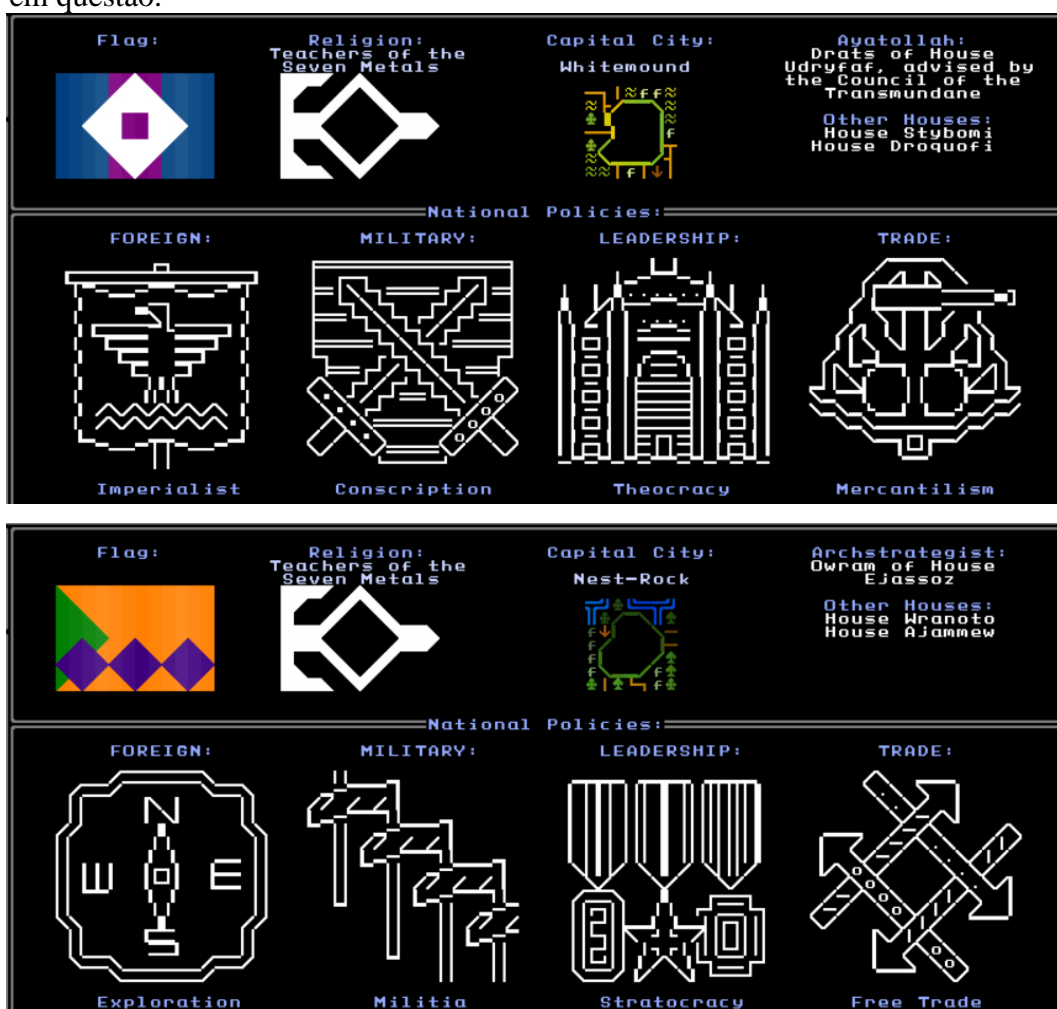
Rios, montanhas e demais formações geológicas (como desertos e áreas polares) caracterizam diferentes biomas, através dos quais os grupos humanos movem relações comerciais, diplomáticas e bélicas. Essas conexões dão aspectos específicos a cada nação e a cada grupo. Em um nível macroscópico, observamos essas qualidades apenas como territórios

março e maio de 2016. O desenvolvedor Mark Johnson permitiu a inclusão dos dados nesta pesquisa e, como a versão 0.8 ainda não foi lançada no momento que este texto é escrito, está sujeita a mudanças.

⁹⁴ Não é possível explorar os planetas e satélites com a mesma liberdade que se pode navegar pela “Terra”. No entanto, estes servem como referência para outras propriedades do jogo, como o calendário, por exemplo. Logo veremos que, mesmo que o sistema solar seja obviamente simulado para ser maior do que continentes, cidades e distritos, é possível que em um único prédio sejam encontrados mais fios conectivos do que em um planeta inteiro.

dominados por diferentes donos, e que possuem tais e tais aliados e/ou desafetos. Se o jogo se resumisse a isso, estaríamos lidando com um mundo semelhante aos jogos de estratégia como *Europa Universalis IV* e *Crusader Kings 2*, cujas relações entre diferentes territórios são os princípios ativos de seus esquemas dinâmicos. Em *URR*, no entanto, é possível se aprofundar além do nível macro, observando novas camadas que denunciam a intensidade conectiva do jogo, como um fractal que revela novos detalhes de suas formas à medida que aumentamos o *zoom*.

Figura 60 - Dois quadros exibindo as características de duas civilizações do mundo em questão.



Acima, "Drotsafi Priestdom" e, abaixo, "Empire of Wrinommu". O sistema do jogo gera bandeiras e nomes procedimentais para cada uma, atribuindo arquétipos relacionados às práticas religiosas, militares, diplomáticas e comerciais de cada uma delas.

Como cada mundo gerado por cada jogador é diferente, geramos um mundo especificamente para esta pesquisa, de modo a demonstrar as conexões traçadas ao longo dos níveis de funcionamento do jogo. Escolhemos uma das civilizações como a protagonista: seu nome, assim como uma série de outros atributos, é gerado pelo próprio sistema: Drotsafi

Priestdom (figura 60, acima). O primeiro termo, “Drotsafi”, é fruto de técnicas de geração de palavras que associam sílabas e letras de modo a imitar nomes próprios “reais”. Já o segundo é pré-definido pelo sistema, podendo ser alocado em diferentes títulos de diferentes civilizações. Uma série dessas pré-definições podem ser encontradas no mundo que geramos: uma delas encontra-se acima (figura 60, abaixo), a civilização “Empire of Wrinommu”.

Além de “*Priestdom*” e “*Empire*”, há também títulos tais como “*Children of Tosorre*”, “*Kindred of Ojypir*” e “*Nation of Zekirre*”. Apesar de pré-definidos, há traços de conectividade na atribuição destes nomes às suas respectivas civilizações. Nomes como “kingdom” e “empire” são associados a grupos feudais, enquanto “kindred” e “clan” remetem aos costumes de tribos de caçadores. Duas civilizações nômades de nosso mundo recebem nomes que fazem jus à suas práticas vageantes: “*Roamers of Jadimnë*” e “*Riders of Motunê*”. Obviamente, esses termos existem em muito menor número que os nomes gerados procedimentalmente, cujo número de possibilidades alcança a casa dos milhões.

A mesma estratégia de composição será empregada não só em nomes, mas em outros atributos mais desenvolvidos do mundo de jogo – os quais abordaremos a seguir. Vale ressaltar de antemão, porém, que são os critérios de composição tais como esses que revelam o caráter qualitativo do mundo de jogo, assim como a composição de um tabuleiro de xadrez em um determinado momento da partida revela atributos de sua dinâmica.

Nesse sentido, podemos voltar às ideias de meios de exterioridade e interioridade. Percebemos que nos nomes das civilizações possuímos dois tipos de entidades. Os nomes gerados procedimentalmente representam o primeiro tipo: suas peças possuem muito mais força exterior que interior, já que as sílabas só alcançam sua potência máxima quando se unem a outras, externas a elas. Nomes como “Kingdom” e “Roamers”, com suas propriedades fechadas e pré-moldadas, possuem mais força interna: por mais que possam se conectar aos nomes procedimentais, percebemos muito mais suas qualidades internas – assim como uma peça de xadrez.

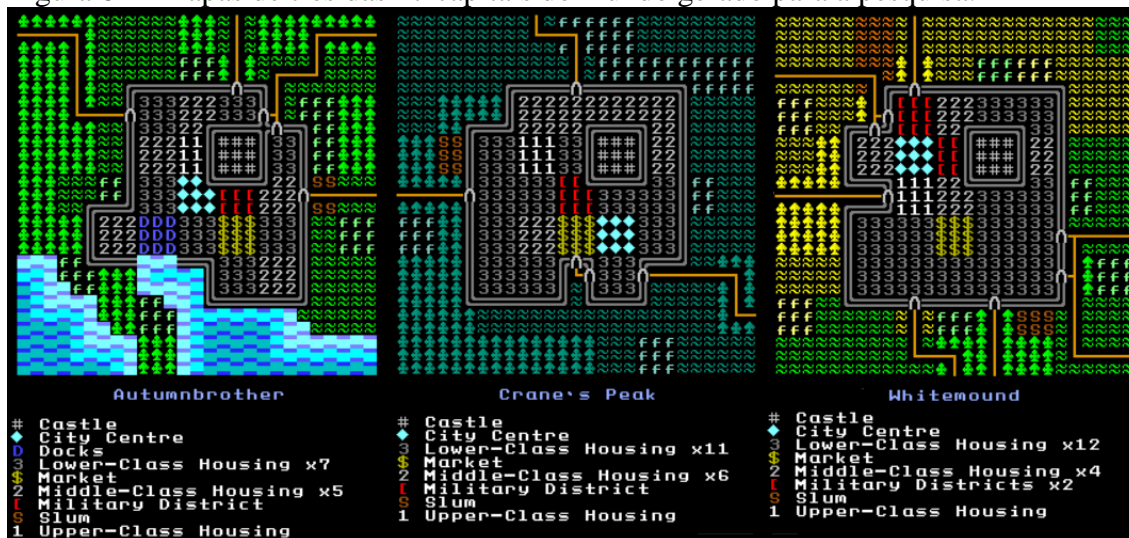
O mesmo processo acontece nos atributos mais desenvolvidos de cada civilização. Nosso estudo de caso, a Drotsafi Priestdom, possui certos arquétipos pré-definidos que designam uma série de propriedades (figura 60). Entre elas, selecionamos quatro: diplomática (*foreign*), militar (*military*), liderança (*leadership*) e comercial (*trade*). A cada um desses atributos, é designado um arquétipo pré-definido⁹⁵. A diplomacia Drotsafi, por exemplo, é

⁹⁵ Na mesma figura 60, é possível observar que acima dos quatro atributos selecionados, é possível ver outros que não são pré-definidos, mas gerados procedimentalmente. O nome da cidade, sua capital, o nome de sua religião e os principais líderes da civilização.

imperialista. No restante do mundo, há outras nações que seguem o mesmo paradigma diplomático, como a Kingdom of Ekyrim, por exemplo. Ao contrário dela, que possui uma liderança monárquica, a Drotsafi Priestdom obedece um regime de liderança *teocrático* (compartilhado por apenas mais uma das 35 civilizações que compõem nosso mundo de jogo). Na figura 60, podemos ver ainda o exemplo da Empire of Wrinommu possui arquétipos distintos da Drotsafi Priestdom. Sua política externa é de exploração e sua liderança é estratocrática (liderada por militares).

A Drotsafi Priestdom possui ainda um modelo militar de *recrutamento* (*conscription*) e práticas comerciais *mercantilistas*. Já em Wrinommu vemos os arquétipos *miliciano* (*militia*) e *livre comércio* (*free trade*) para os atributos militar e comercial, respectivamente. No entanto, o que exatamente significam essas designações na dinâmica prática do jogo, sob tudo no modo como ela se apresenta ao jogador? Para descobrir, é necessário descer mais um nível na escala. Já percebemos que cada civilização é um mosaico feito de peças pré-fabricadas (arquétipos) e outras compostas dinamicamente (nomes de religião, líderes, capital, etc.). Devemos agora adentrar essas peças e avalia-las microscopicamente.

Figura 61 - Mapas de três das 17 capitais do mundo gerado para a pesquisa.

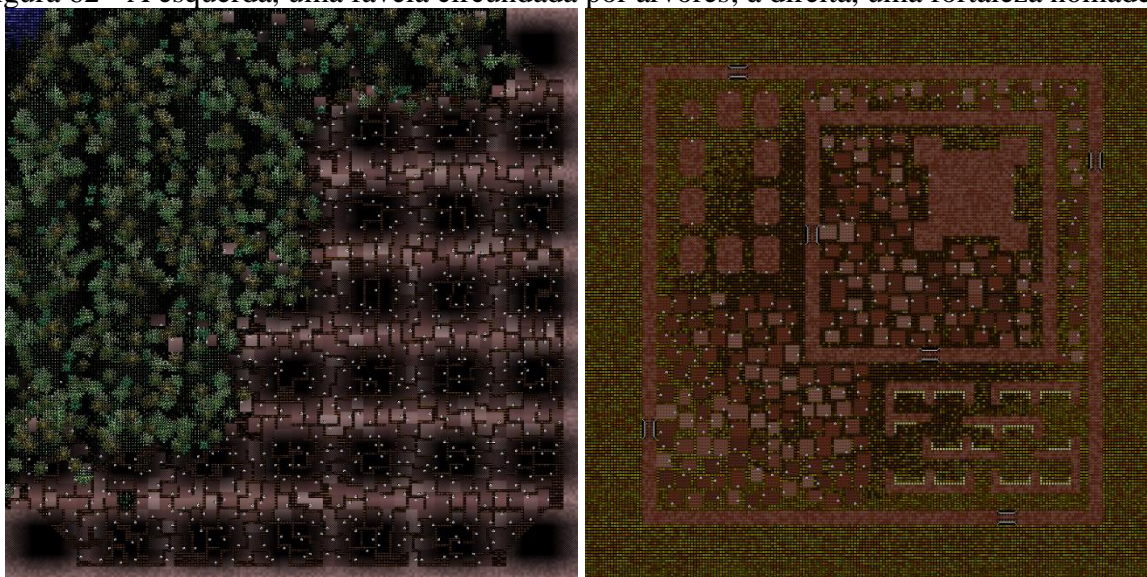


Cada cidade é composta por distritos (legendados no mapa com diferentes caracteres). Há distritos comerciais, residenciais (de alta, média e baixa classe), militares, portuários, centrais, etc.)

Acima (figura 61), temos três exemplos de uma das peças que compõem o mosaico de conexões de cada civilização: suas respectivas capitais. São cidades com nomes gerados procedimentalmente, porém com outros atributos pré-definidos que podem ou não aparecer em suas composições. Cada cidade é composta por distritos e circundada por fazendas, favelas, estradas, vegetação, entre outras entidades. Começemos pelos distritos: há algumas variações dentre estes: há três níveis econômicos (representados pelos números 1, 2 e 3 na figura 61).

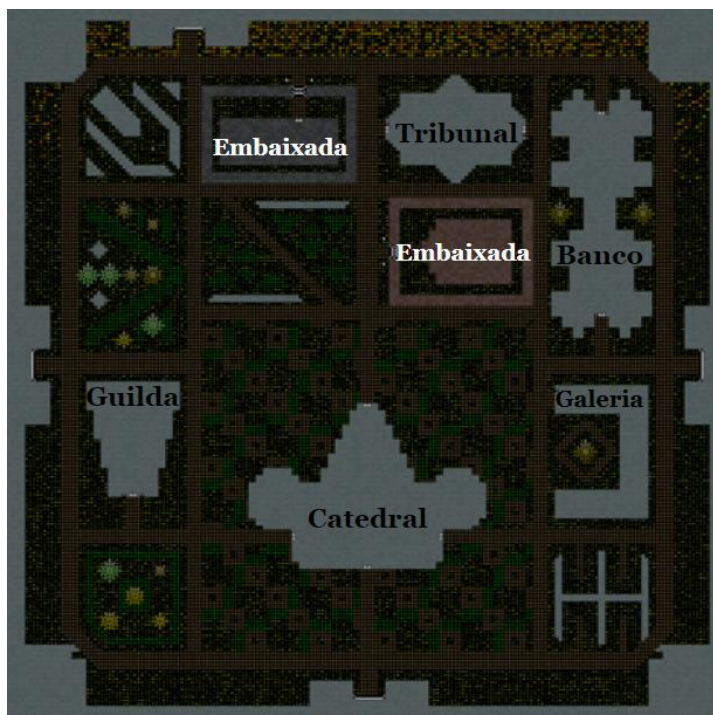
Além deles, há outras espécies de distritos que serão encontrados em cidades cujos arquétipos de suas civilizações condizerem. Por exemplo, distritos militares aparecem nas três capitais acima (e na maioria das capitais do jogo). Em Whitemound (figura 62, direita), capital da Drotsafi Priestdom, vemos não um, mas dois distritos militares (como vimos anteriormente, possui um regime militar de recrutamento e um arquétipo diplomático imperialista). Civilizações com arquétipos militares pacifistas, por exemplo, não possuem sequer um distrito militar em suas capitais.

Figura 62 - À esquerda, uma favela circundada por árvores; à direita, uma fortaleza nômade.



Dentre os 23 distritos de Whitemound, 13 são zonas residenciais pobres, 4 zonas de classe média e uma de alta classe. Dois distritos militares, um comercial e um distrito inteiro ocupado pela fortaleza central do governo da Drotsafi Priestdom. No entanto, apenas diferenciá-los por nome ainda não nos dá a verdadeira dimensão da intensidade conectiva de *URR*. Precisamos avançar mais um passo e mergulhar no único distrito que não mencionamos: o distrito central de Whitemound.

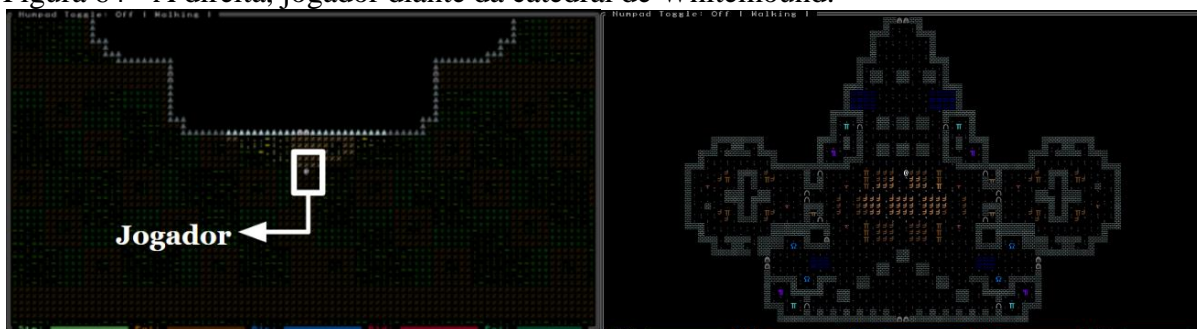
Figura 63 - Distrito central de Whitemound.



Tratando-se da capital de uma civilização teocrática, era de se esperar que a maior construção fosse uma catedral. Além disso, há duas embaixadas de nações feudais próximas, um tribunal, um banco e uma galeria com estátuas e obras de arte.

Aqui, começamos a ter uma dimensão de como este mundo se apresenta à perspectiva do jogador. No centro de Whitemound, temos acesso a uma série de estruturas e personagens. De um ponto de vista extensivo, trata-se de um terreno razoavelmente grande, que demanda pelo menos uma hora para ser percorrido e explorado com cuidado. Do ponto de vista de sua conectividade, porém, as dimensões espaciais e temporais se perdem: neste distrito, temos acesso a um mundo de conexões que, paradoxalmente, torna-se mais vasto que o mapa mundi que inaugurou esta seção do capítulo.

Figura 64 - À direita, jogador diante da catedral de Whitemound.

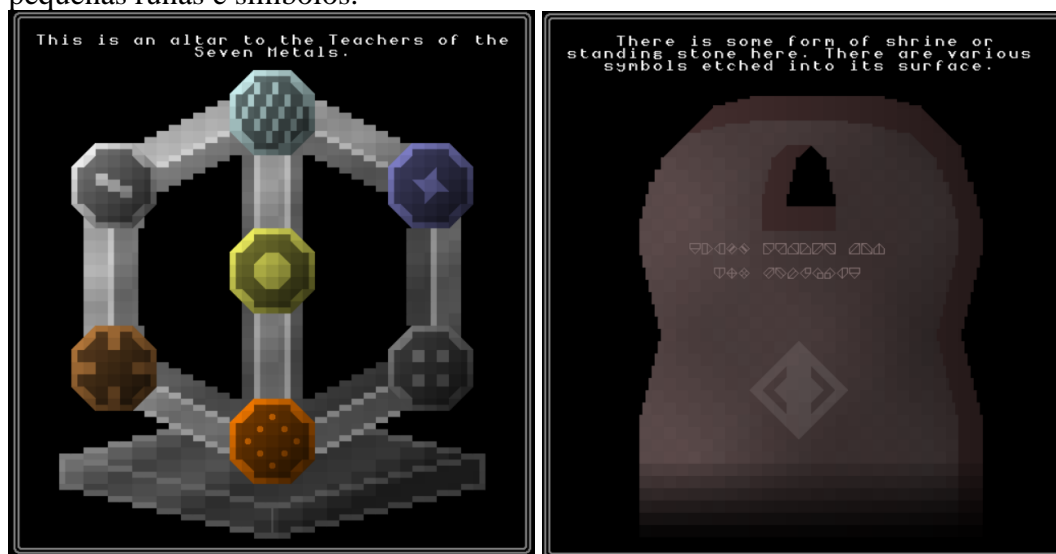


À esquerda, o interior da construção. Cada prédio de cada distrito pode ser adentrado e explorado.

Tomemos como exemplo a catedral de Whitemound. Ela pertence à religião que governa a nação, denominada “Teachers of the Seven Metals”. Ao adentrarmos o templo, nos deparamos

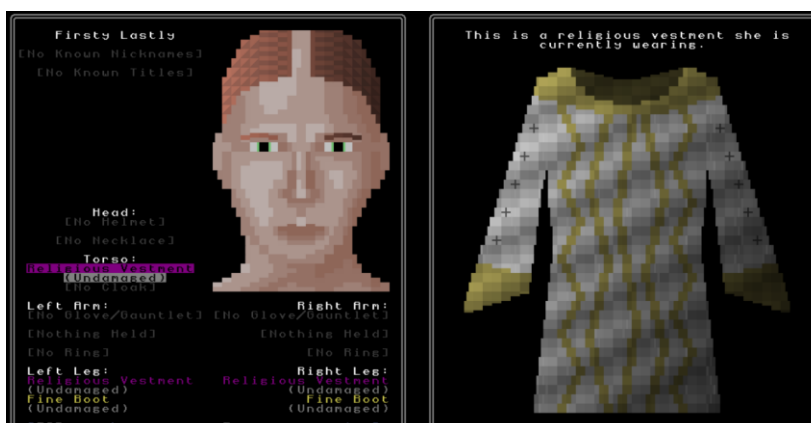
com móveis, pisos adornados, porta-incensos e ídolos religiosos (figura 65). Tais objetos refletem a cultura desenhada pelas peças que compõe a cidade e, no nível superior, a civilização. Na versão final do jogo, o criador Mark Johnson planeja que esses símbolos e sinais permitam que o jogador siga pistas e chegue a lugares e personagens que o permitirão desatar nós de complôs e segredos tecidos pelas próprias conexões do jogo.

Figura 65 - Ídolos religiosos da seita Teachers of the Seven Metals, compostos por pequenas runas e símbolos.



Feito este estudo de caso, conseguimos vislumbrar com mais detalhamento a integração entre os níveis de conectividade de *URR*. O sistema solar é gerado, contendo um planeta habitado por populações feudais, nômades e tribais. Estas dividem-se em territórios e traçam relações de aliança, enquanto os atributos da civilização influenciam as espécies e distribuição dos distritos, a composição dos distritos reflete os arquétipos da civilização. No entanto, graças às conexões de uma civilização com as outras, de um distrito com os outros e de um personagem com os demais, nos vemos diante de um mosaico tão vasto quanto nossa vontade de explorá-lo permitir. E não se trata apenas de vastidão extensiva, mas *intensiva*. Podemos intuir as políticas bélicas ou comerciais de uma civilização a partir das suas reflexões no nível microscópico: quem são seus soldados? O que vendem seus mercadores? Onde se instalam seus governantes e que relações traçam com outras nações?

Figura 66 - Caso encontremos um personagem de *URR*, podemos examinar sua vestimenta.



Acima, vemos uma sacerdotisa da religião Teachers of the Seven Metals. Sua roupa indica sua ocupação religiosa, assim como as roupas de um nobre e um plebeu indicarão sua posição social.

Em última instância, portanto, poderíamos dividir 7 escalas espaciais em *URR*: sistemas solares, planetas, civilizações, cidades, distritos, construções, e as entidades mínimas: personagens e objetos. Essa divisão é em grande parte arbitrária, já que poderíamos descer até o nível das partículas que compõem um personagem, como a sua face (que é gerada pelo sistema do jogo, alcançando mais de dez milhões de possibilidades⁹⁶), peças de roupa, etc. Não é isso que nos interessa, mas sim estipular o lugar do jogador dentro dessas escalas.

Percebemos que o jogador ocupa a escala do que chamamos de entidades mínimas: objetos e personagens. Suas ações só alcançam este nível: não é possível manipular uma cidade inteira, muito menos uma civilização, como em *Europa Universalis IV* ou *Cruzaider Kings 2*. Só se pode conversar com personagens individuais, observar objetos que se apresentem à nossa perspectiva e caminhar por distritos que nos permitam entrada. Além disso, os outros personagens são capazes de exercer ações na mesma escala: interagem uns com os outros, obtêm informações e executam tarefas referentes a seus trabalhos. Por meio destas ações, deixam transparecer motivações que denunciam o background de suas respectivas culturas – seja na dimensão religiosa, militar ou qualquer outra.

⁹⁶ Fonte: <http://www.ultimaratioregum.co.uk/game/2015/05/02/the-big-reveal/>.

Figura 67 - Tela de conversação entre o jogador (direita) e uma sacerdotisa encontrada na catedral (esquerda).



Abaixo de ambos, vemos os assuntos sobre os quais podemos conversar com cada personagem – embora nem todos possuirão informações sobre cada um deles.

Por mais que a perspectiva do jogador só alcance esta escala, é possível romper essa barreira justamente por meio da adaptação à teia conectiva do jogo, obtendo informações e influência sobre ordens superiores. Já na versão 0.8 do jogo, será possível tentar conversar com personagens (figura 67), escolhendo assuntos específicos (desde diferentes cidades e culturas até jóias, artefatos e eventos específicos). Conversando com o sacerdote acima, por exemplo, teríamos a chance de perguntar a respeito de sua religião e descobrir que seus líderes governam a cidade de Whitemound. Um sacerdote, no entanto, certamente saberá menos que um bispo mais poderoso. Um nobre que aceite se comunicar conosco terá informações valiosas relativas à corte, mas um morador de lugares mais pobres não é desprovido de informações: pode nos apontar a mercados paralelos que vendam objetos raros e ofereçam preços atrativos; ou mesmo nos confiar o local secreto de uma seita religiosa herética.

Cada conexão acima forma, também, partículas que desenham a teia invisível – porém sensível – do mundo de jogo. Como demonstrado, embora possamos dividir a extensão do mundo em escalas segundo nosso gosto, essa teia é bem mais maleável e pouco suscetível à definição de fronteiras estanques. De um objeto simples, é possível retirar uma informação que nos leve a um personagem ou um pequeno local que podem abrir um campo de possibilidades antes completamente impensado – assim como uma única peça de Go pode desmontar um grupo adversário inteiro. Novamente, isso se deve ao fato de que quanto mais nos aprofundamos nas propriedades e capacidades dos personagens e objetos nessa escala, mais notamos a relativa restrição de seu meio de interioridade. Muito mais abrangente são seus meios de interioridade,

que traçam conexões intensivas capazes de concatenar todas as escalas do mundo de jogo em uma só.

3.3 Ultraintabilidade em *Dwarf Fortress*

Figura 68 - Ilasil, “The Absolute Planes”, no ano 250 de sua história.



Na conclusão do livro *Cybertext* (1997), Espen Aarseth aponta para uma “escala ergódica” (p. 179), um espectro contínuo cujas extremidades seriam: a) anamorfose, problemas fixos solucionáveis por uma única estratégia (estáveis) e b) metamorfose, textos em mutação constante (instáveis). Para ilustrar o primeiro deles, Aarseth utiliza-se de um diálogo. Na anedota, um desenhista mostra um desenho de um tigre a um repórter, mas este enxerga apenas linhas em forma de labirinto. O artista pergunta se ele não consegue identificar o animal nas linhas, quando finalmente o repórter descobre o tigre escondido no desenho. O artista, satisfeito, diz finalmente que sua intenção era que o observador fosse “co-autor” da obra.

Aarseth obviamente critica o uso da expressão, dando a entender que o desenho é anamórfico, um problema dotado de solução prévia. Não se trata de uma obra co-criada por observador e artista, mas sim um produto cujos componentes são pré-conectados, exigindo que

o esforço de decifração da conexão. Não há, segundo Aarseth, criação, mas a descoberta de algo que exige o ajuste do nosso modo de olhar. A metamorfose, do outro extremo, revela “tigres” contingentes, imprevistos e imprevisíveis:

Desta perspectiva, derivamos, portanto, três categorias; novelas [...], literatura anamórfica (enigmas solucionáveis) e literatura metamórfica (textos de mudança e imprevisibilidade). Os tigres observáveis na última não são planejados, nem fixados ou domados. Estranhamente, porém, nossa influência como agentes literários nestes labirintos é muito mais real que nos dois tipos anteriores.⁹⁷

Aarseth lamenta ainda o fato de que, quanto mais próximos da extremidade metamórfica de sua escala ergódica, mais raros tornam-se os cibertextos. De fato, não há muitos jogos que implementem a mesma intensidade conectiva de *Ultima Ratio Regum*. O jogo foi planejado para ser concluído em um prazo de 10 anos, sob um cronograma cuidadoso de implementações de sistemas graduais, até atingir sua versão final⁹⁸. Na fase atual, o jogo ainda está tecendo ainda mais fios de sua teia conectiva: virão nas próximas versões sistemas de geração de armas e armaduras características de cada civilização (0.9) e geração procedimental de fauna e flora (1.0). Em 2020, quando finalmente estiver pronto, o jogo entrará na sua fase de instabilidade – atualmente, trata-se de um mar de conexões que oferece pouco em termos de movimento e metamorfose que não partam do próprio jogador.

Há, no entanto, um viodeogame já mencionado algumas vezes ao longo deste trabalho que provavelmente figuraria no texto de Aarseth como exemplo de cibertexto metamórfico (caso já existisse à época do lançamento do livro. Embora esta seja uma afirmação difícil de se verificar, não parece haver sinal de que exista no mundo um jogo mais complexo que *Dwarf Fortress*. E se Mark Johnson projetou modestos 10 anos para a finalização de seu projeto, Tarn Adams e seu irmão Zach Adams já trabalham há 12 em *Dwarf Fortress*. O cronograma dos irmãos mais parece um túnel sem final à vista, mas em algumas entrevistas a dupla estima pelo menos mais uma década de trabalho pela frente.

Nesse meio tempo, o jogo – que é disponibilizado gratuitamente pelos irmãos – ganhou notoriedade pela sua interface incomparavelmente complexa, pela profundidade de sua simulação e pelas histórias anedóticas que surgem de seu sistema – os tigres indomados premeditados por Aarseth. Dentre elas, a mais famosa é a da fortaleza de Boatmurdered. *Dwarf Fortress* possui alguns modos de jogo, dentre os quais o mais tradicional é o “modo fortaleza”

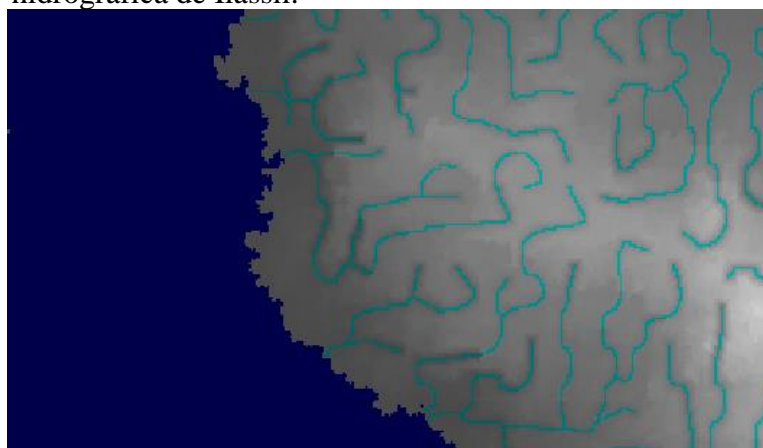
⁹⁷ Livre tradução de: “From this perspective we then derive three categories; novels [...], anamorphic literature (solvable enigmas), and metamorphic literature (the texts of change and unpredictability). The tigers that can be observed in the latter are unplanned, unbound, and untamed. But strangely, in these labyrinths our influence as literary agents is much more real than in the two previous ones”.

⁹⁸ Fonte: <http://www.ultimarioregum.co.uk/game/development-plan-2/>.

(*fortress mode*), no qual jogadores comandam um grupo de anões, cada um com um arquétipo que orienta suas funções e habilidades. Construtores, por exemplo, erguem estruturas e mineiros exploram os interiores das veias de minerais. A fortaleza cresce, podendo chegar às centenas de habitantes.

A fortaleza de Boatmurdered vem de um jogo colaborativo: o primeiro jogador inicia o processo e, em dado momento, passa o bastão ao próximo, que dará continuidade à mesma instância do jogo. A dinâmica ficou conhecida como “jogo de sucessão” (*succession game*), tendo se tornado tradição na comunidade de jogadores. A linhagem de Boatmurdered ficou famosa por começar de modo tradicional, mas evoluir em um festival de incidentes improváveis, que assustaram até mesmo os jogadores mais veteranos. Parte do apelo da história vem da imprevisibilidade do sistema: uma guerra foi travada entre anões e elefantes selvagens. Os anões, ao tentar defender a fortaleza e pressionados pela instabilidade de seu ambiente, acabaram se transformando em monstros furiosos que iniciaram a intrigante prática de devorar os elefantes que matavam. Muitas rodadas depois, quando um outro jogador já controlava Boatmurdered, um segundo surto foi iniciado por uma anã construtora que tinha o hábito de talhar em pedras cenas de anões matando elefantes. Prontamente, a personagem entrou em uma espiral de comportamentos violentos que se alastrou pela fortaleza como epidemia, causando sua ruína graças a um incêndio provocado por rios de lava.

Figura 69 - Porção limitada do mapa exibindo a configuração hidrográfica de Ilasil.

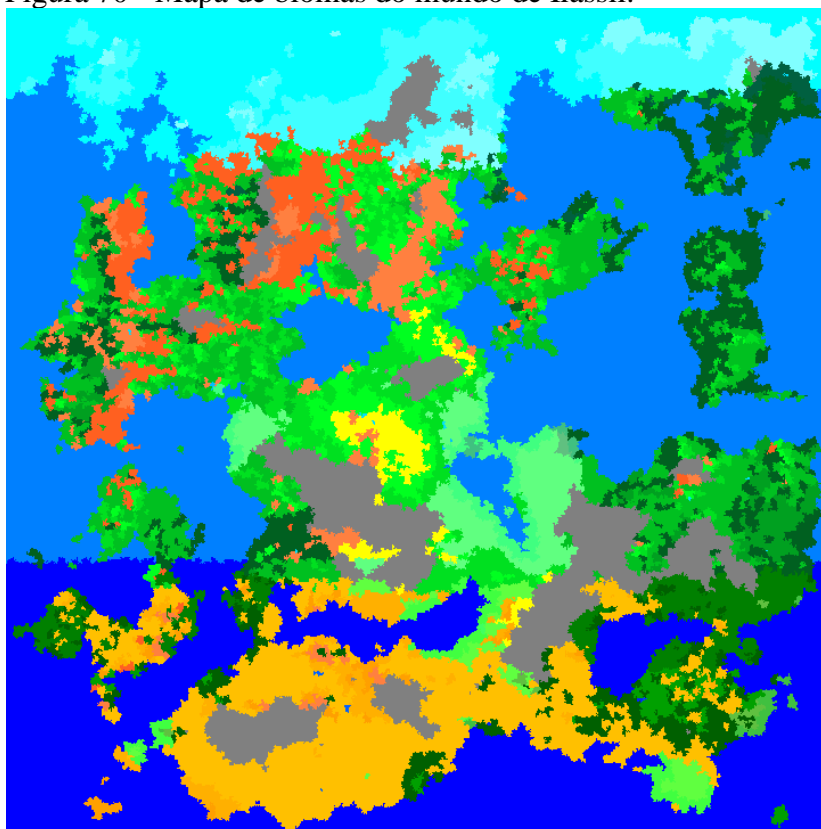


Em azul escuro, vê-se o oceano; em cinza, as terras e em azul claro os rios. Quanto mais escuro o tom de cinza, mais próximo do nível do mar. Portanto, nos setores mais elevados, posicionam-se as nascentes e, nos mais baixos, os rios desaguam no oceano.

Como Boatmurdered demonstra, se *URR* está preparando o terreno para uma metamorfose que virá nas próximas versões, *Dwarf Fortress* já possui alto grau de instabilidade

correndo em seus sistemas. A culminância deste processo chegou em 2014, quando o jogo foi atualizado com o que Tarn Adams chamou de “*world activation update*”⁹⁹, ou atualização da ativação do mundo”. No modo fortaleza descrito acima, o jogador escolhe em que parte do mundo (figura 70), sua civilização de anões será sediada. Aqui encontramos um processo semelhante ao de *URR*, quando o jogo gera proceduralmente uma série de características geológicas, tais como elevação de montanhas, planícies e vales; distribuição de minérios e redes de cavernas; fluxo de rios, surgimento de lagos e lençóis freáticos; clima, vegetação, temperatura e fauna, entre outras. Tais condições constituem o primeiro nível de influência do ambiente sobre a fortaleza do jogador: uma construção próxima a desertos ou vulcões terá a seus dispor recursos e instabilidades bem diferentes de uma assentada no alto de um pico recoberto por neve.

Figura 70 - Mapa de biomas do mundo de *Ilasil*:



Em verde, vegetações e florestas mais ou menos densas; em amarelo, desertos e áreas áridas; em cinza, áreas montanhosas. O azul claro do oceano delimita a zona temperada, enquanto o tom escuro do oceano inferior marca a zona tropical.

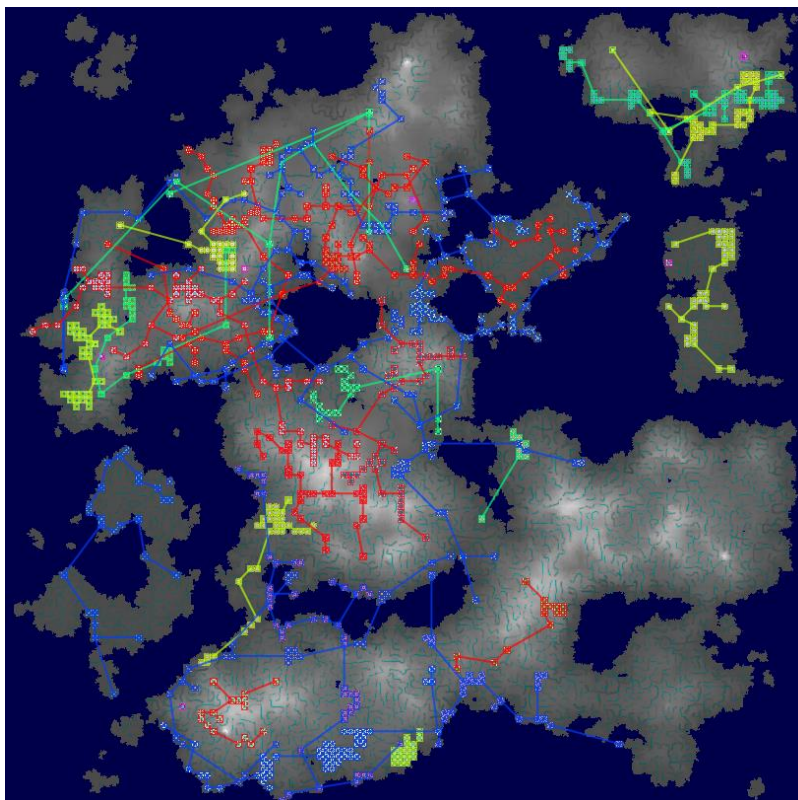
O que a atualização de 2014 mudou no jogo foi o modo como o restante do mundo se comporta enquanto o jogador interage com o ambiente imediato que rodeia sua fortaleza. Antes

⁹⁹ Fonte: <http://www.bay12forums.com/smf/index.php?topic=140013.0>.

de 2014, os atributos tais como os descritos acima eram gerados e podiam ser manipulados pelo jogador. Por exemplo, é possível aposentar uma fortaleza e iniciar uma nova no mesmo mundo, finalmente tendo acesso a um terreno longínquo, acessível antes apenas por meio do mapa. No entanto, apesar destes terrenos existirem em potencial, permaneciam imóveis e sem atividade. A partir de 2014, porém, o mundo seria ativado.

Embora os atributos geológicos não ofereçam muita instabilidade, as civilizações e grupos que os ocupam são bem diferentes. Há, espalhados pelo mundo de Ilassil, centenas de outras fortalezas ocupadas por dezenas de milhares de personagens. Com a atualização, essas civilizações passam a possuir seus próprios objetivos e metas, planejando e executando ações no sentido de alcançá-los. De certa forma, o jogo apenas põe em continuidade um processo iniciado no momento de sua geração. Um dos traços mais únicos de *Dwarf Fortress* é sua geração procedimental de eventos históricos. De modo prático, esses eventos distribuem-se em criações de fortes, vilas e acampamentos; a evolução de vilas para cidades e de fortes para fortalezas; batalhas entre duas civilizações distintas (ou entre grupos distintos dentro de uma mesma civilização); declarações de guerra entre civilizações; vitórias e derrotas de batalhas e guerras, cidades e vilas pilhadas, cercos a castelos e fortalezas; conquistas e demarcações de novos territórios, entre muitos outros.

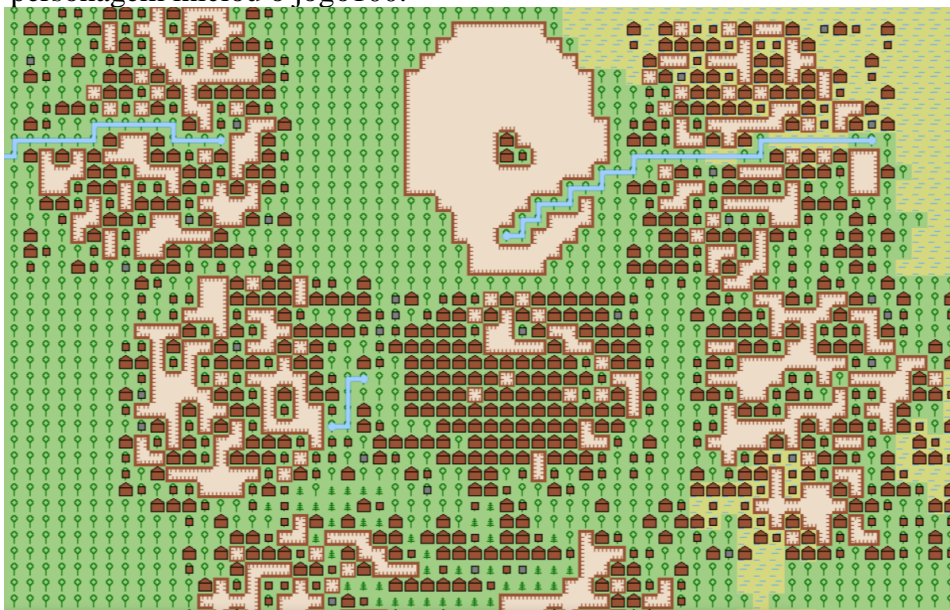
Este processo acontece, obviamente, de modo gradual. Nos primeiros anos de Ilassil, surgem civilizações que clamam territórios e situam operações de extração de recursos e construção e manutenção de suas sedes. Logo, porém, estas começam a disputar domínios sobre rios ou minas, por exemplo, com outros grupos. Tais disputas levam a interações bélicas. Nem tudo se resume a violência, no entanto: vilas próximas estabelecem relações comerciais para suprir demandas que o ambiente imediato de suas instalações não supre; e mesmo quando um conflito se desenrola, muitas vezes se resolverá de modo diplomático. Pouco a pouco, o mundo toma forma, e por volta de 15 minutos se passam antes que o software termine de simular 250 anos de história.



Depois de 250 anos de história. Em azul, civilizações humanas; em amarelo, civilizações de elfos; em verde, de goblins; em vermelho, de anões.

Torna-se mais fácil imaginar a situação dos jogadores nos mundos de *Dwarf Fortress* pós-atualização. Em vez de lidar com civilizações e personagens estacionários, passa a ser necessário perceber as dinâmicas disparadas pelo mundo. De guerras a relações comerciais, o jogador perceberá no canto ínfimo que ocupa o reflexo das instabilidades que se chocam e reverberam por toda a extensão do mundo de jogo. Uma fortaleza já ocupa um lugar relativamente humilde diante das 39 civilizações do mundo de Ilasil – cada uma delas composta por múltiplas fortalezas, cidades e vilas. No entanto, o segundo modo de jogo de *Dwarf Fortress* dá ao jogador o comando não de uma fortaleza, mas de um único personagem.

Figura 72 - Parte da cidade de Tempestboot, cidade élfica onde nosso personagem iniciou o jogo¹⁰⁰.



Como experimento prático, criamos um personagem e começamos nossa jornada na cidade de Tempestboot. O objetivo é testar até que ponto conseguimos notar os traços de instabilidade do mundo de jogo no nível de interação mínimo do jogo – entre nosso personagem e outros personagens, de modo semelhante à escala mínima de *URR*. Escolhemos Tempestboot por meio do software Legends Viewer¹⁰¹, uma aplicação que extrai dados de *Dwarf Fortress*, de modo semelhante com que um programa extrai dados de redes sociais ou de bancos de dados em geral. A riqueza dos dados do Legends Viewer permite procurar civilizações grupos, personagens específicos, averigura sua história e nos mover, por meio de hiperlinks, pelas conexões e reconexões disseminadas pela instabilidade do jogo.

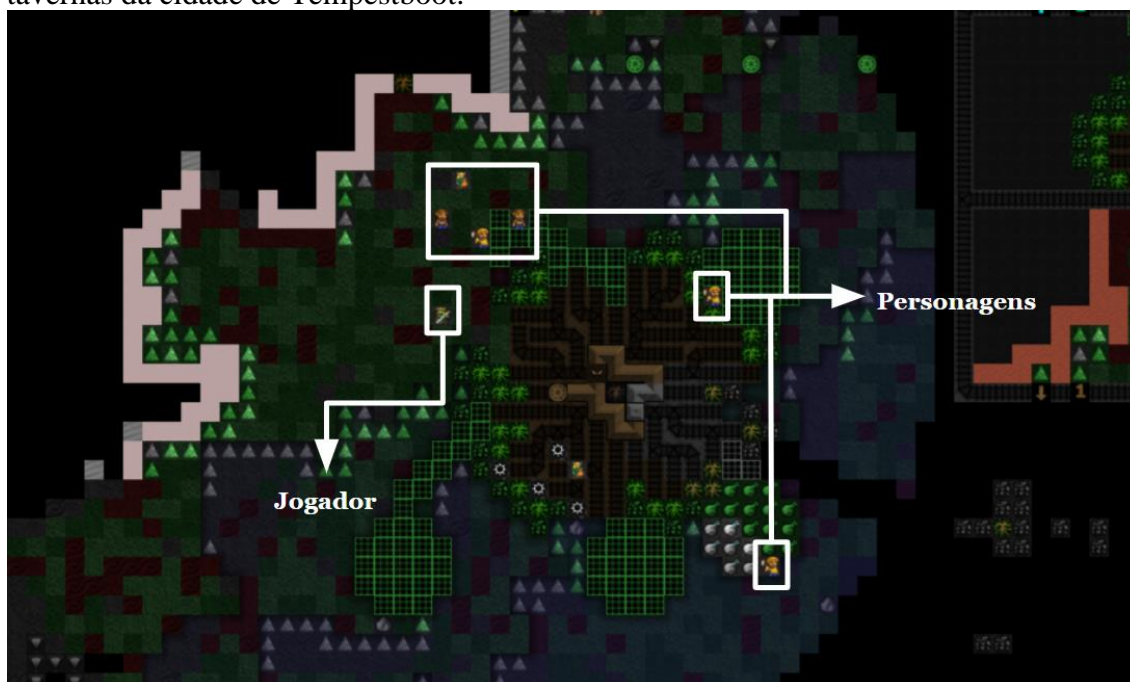
Tempestboot possui uma série de traços únicos. O primeiro deles é a sua população é lar de por volta de 5100 habitantes, um número bastante alto. É uma cidade de origem élfica, comandada pelo grupo The Mellow Wax, membro da civilização The Wisp of Loving. A cidade não começou com essa quantidade de habitantes: outro de seus traços interessantes é que foi inaugurada no primeiro ano da história de Ilassil, por um pequeno grupo de elfos que se expandiu pouco a pouco até a casa dos milhares de habitantes. Para dar conta de tantos cidadãos,

¹⁰⁰ A imagem acima foi extraída do jogo por meio do software Uristmaps, uma aplicação desenvolvida para transformar os dados de *Dwarf Fortress* em um mapa semelhante aos do Google Maps. A aplicação não é interativa, mas é útil para visualizar mapas de cidades, vilas e outros pontos do mundo de jogo. Fonte: <http://www.uristmaps.org/>.

¹⁰¹ O software Legends Viewer extrai dados de um dos modos de jogo de *Dwarf Fortress*, o “Legends Mode”. Nele, é possível ver a história dos milhares de personagens do jogo. O Legends Viewer se alimenta dessa base de dados, oferecendo uma série de outras funções, como ciração de árvores genealógicas, mapas de guerras, batalhas, alianças comerciais, entre outras.

novas casas, dormitórios, centros comerciais e até mesmo tavernas precisaram ser construídas e mobiliadas por trabalhadores com habilidades específicas para criar e manter os espaços. O toque cultural único dos elfos é que quase tudo é feito de madeira, e boa parte das casas de Tempestboot é cravada diretamente em árvores muitas vezes maiores que o nosso jogador (figura 73).

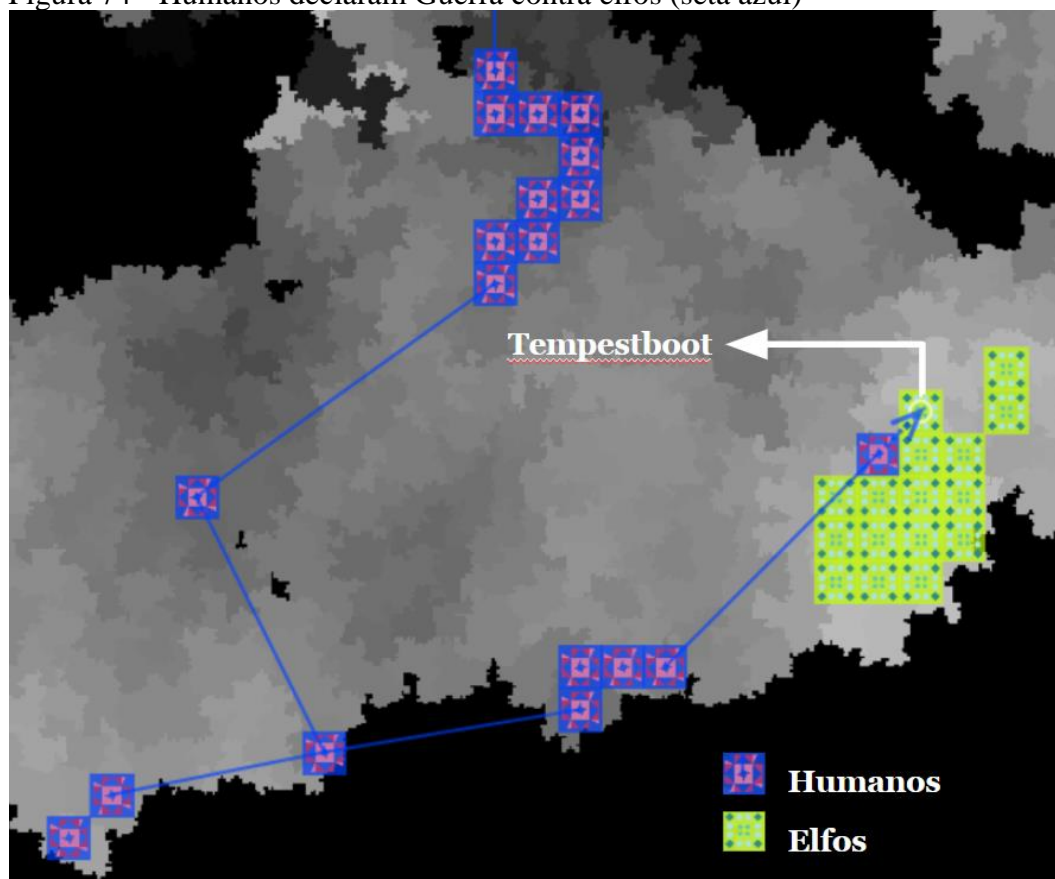
Figura 73 - Personagens e nosso avatar circundam uma árvore que serve como uma das tavernas da cidade de Tempestboot.



A árvore se estende por vários níveis verticais, podendo ser escalada, o que nos dá acesso a novos cômodos – nesse caso, quartos que podemos alugar para descansar da viagem.

Os personagens apontados na imagem acima (figura 73) não são elfos como nosso personagem. São, na realidade, humanos. *Dwarf Fortress* possui uma fauna tão ou mais variada que sua diversidade de biomas. Parte dessa fauna se organiza em grupos que, por suas vezes, aglomeram-se em civilizações – novamente, um traço semelhante à distribuição das nações em *URR*. Cinco raças demonstram essa habilidade: Elfos, anões, humanos, goblins e kobolds – estes últimos com bem menos desenvoltura que os quatro primeiros. Como vimos, civilizações entram em guerra e, como esperado, raças diferentes tendem a guerrear muito mais entre si do que a atacar seus iguais. O que explica, então, que humanos tenham passado a construir, habitar e manter casas arbóreas e a praticar comércio, diplomacia, medicina, performances artísticas em uma das maiores cidades élficas de Ilasil?

Figura 74 - Humanos declaram Guerra contra elfos (seta azul)



A civilização élfica The Wisps of Loving, apesar de ter fundado Tempestboot no ano 1 da existência de Ilasil – e ter mantido a cidade por mais 249 –, se envolveu em uma série de enfrentamentos bélicos com grupos e civilizações humanas. Na figura acima (74), vemos que a evolução histórica do mundo deixou a capital da civilização em uma situação desconfortável. Cada bloco amarelo representa uma das habitações élficas da civilização. É possível perceber que, adjacente a estes blocos, há um azul: trata-se da vila élfica de Famousfin, colorida com a legenda que representa os rivais humanos. Não se trata de um bug: no software Legends Viewer, podemos inspecionar a história de Famousfin e ler a seguinte entrada:

Em 243, no início do inverno, a civilização The Kingdoms of Wandering derrotou o grupo The Tornados of Joining, da civilização The Wisps of Loving, colocando a humana Itke Shakenlines no comando de Famousfin. O novo governo foi batizado de The Wild League, durante o evento A Conquista de Famousfin, como parte da guerra The Dangerous Conflict. (dados extraídos de *Dwarf Fortress*, por meio do software Legends Viewer).

Como resultado, a vila de Famousfin é, no momento (ano 250) habitada por apenas um ser humano (a líder militar Itken Shakenlines) e 53 elfos. Famousfin, não obstante, passou a se tornar parte da civilização que passou a comandar ataques constantes a Tempestboot, The Kingdoms of Wandering. Vimos em *URR* que cada civilização possui uma capital, à qual se

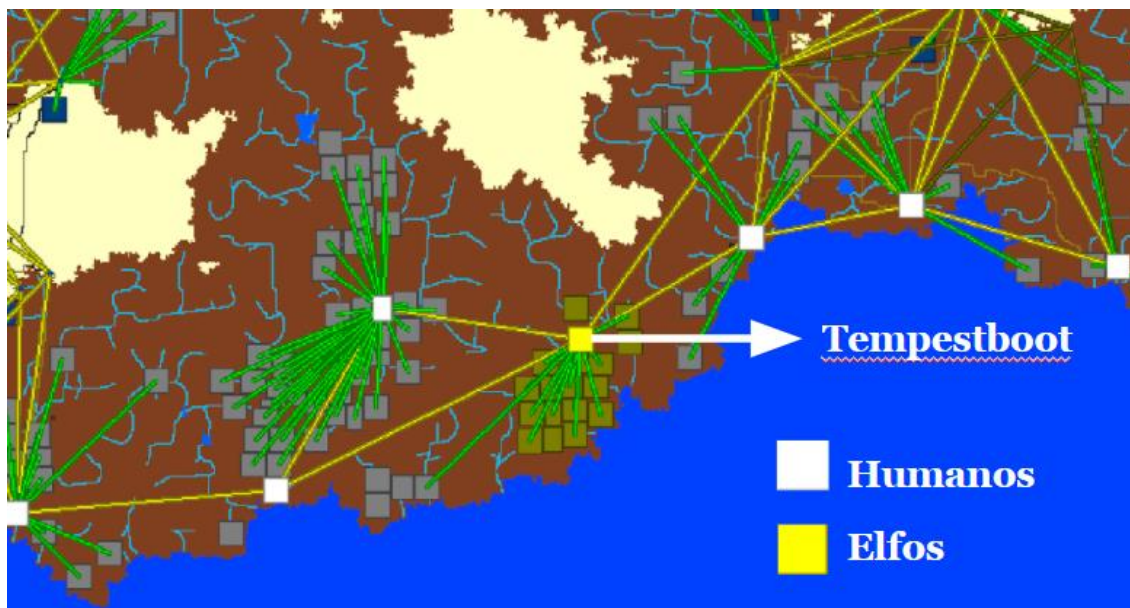
justapõem assentamentos menores, como favelas ou fazendas – um processo de conexão direta feita pelo jogo, a partir de partículas mais ou menos pré-moldadas. Em *URR*, uma favela ou fazenda sempre será diferente da outra, mas uma vez geradas, cada um serve como um bloco discreto que pode ser justaposto a uma capital.

Em *Dwarf Fortress*, o processo é mais instável: uma capital emerge de processos contínuos, nos quais personagens estabelecem laços financeiros, afetivos, militares, etc. Esses processos direcionam, muitas vezes, uma quantidade grande de personagens para uma vila específica, que, graças à sua maior quantidade de trabalhadores e recursos, poderá se expandir e alcançar o tamanho de uma capital. Esse status não é apenas um título: a capital centraliza operações comerciais (figura 75) e militares. Por esse motivo, conquistar uma capital em uma guerra leva mais facilmente a um domínio da civilização – e potencialmente à sua ruína, o que não é impossível.

É o que os humanos da civilização *The Kingdoms of Wandering* parece almejar. Antes de conquistar *Famousfin* em 243, já haviam atacado *Tempestboot* diretamente no ano de 123, durante a guerra *The Conflict of Fists*. O evento durou quase dois anos, tendo sido finalizado no outono de 125, depois de três batalhas: duas vencidas pelos elfos e uma que resultou na pilhagem de *Tempestboot* – embora não em sua conquista. Mais de cem anos depois, a cidade manteria-se de pé, mas cercada de todos os lados por cidades e vilas lideradas por civilizações humanas. Ao longo da passagem desse século, no entanto, *Tempestboot* não permaneceu imóvel: sua população de elfos diminuiu consideravelmente, principalmente em 247, quando a segunda guerra entre a cidade e os humanos se iniciou.

Batizada de “*The Striteful Conflict*”, a guerra durou menos de meio ano, com apenas uma batalha, também vencida pelos elfos. A vitória de uma batalha ou mesmo de uma guerra em *Dwarf Fortress*, porém, não necessariamente é algo positivo. A batalha ostentou pouco mais de 400 óbitos para os humanos, mas mais de 1300 baixas para a civilização *The Wisp of Loving*. *Tempestboot*, o principal alvo da avançada (figura 74), foi deteriorada pela violência da batalha: e é esse o cenário ao qual nosso pequeno personagem é apresentado. Os personagens que vemos acima são construtores humanos reerguendo estruturas perdidas na batalha da qual mais de 3 mil entidades – entre elfos, humanos e animais domados para batalha – participaram.

Figura 75 - Relações comerciais de *Tempestboot*.



Apenas 7 dos 1383 combatentes do lado élfico sobreviveram à absurda instabilidade causada pela guerra The Striteful Conflict. Muitos destes, no entanto, não eram elfos: a maior parte eram humanos. Isso é explicado pela interação contínua entre as civilizações ao longo dos 250 anos de história de Ilasil. Não apenas interação bélica: mas diplomáticas e comerciais – já que Tempestboot, mesmo sob ameaça, ainda era obrigada a vender e comprar das vilas humanas próximas (e vice-versa). Caravanas comerciais saíam de uma civilização a outra e, por vezes, um ou mais membros decidem tricar de residência. Com as baixas na primeira guerra entre Tempestboot e The Kingdoms of Wanding, um vácuo de população se criou, que logo foi preenchido por humanos. Daí percebermos, nos registros históricos do Legends Viewer, elfos lutando pela civilização humana e humanos defendendo a civilização élfica que sua própria nação decidiu ameaçar.

O que podemos absorver de toda essa instabilidade? Há algumas ideias discutidas anteriormente que podem nos ajudar a mostrar o modo como a multiplicidade de processos instáveis em *Dwarf Fortress* descentraliza o jogador – e porque isso é significativo para o estudo dos videogames. A primeira ideia que imediatamente salta aos olhos é a de emergência. O termo possui uma série de repercussões, especialmente relativas ao campo da complexidade, que estuda o que pode se chamar de sistemas adaptativos complexos (FROMM, 2004), ou simplesmente sistemas complexos: um conjunto de partículas em contato mútuo que geram entidades de nível superior cujas propriedades e capacidades não podem ser reduzidas àquelas presentes no nível inferior (OLIVEIRA, 2003; DeLANDA, 2009).

O pesquisador Gustavo Audi (2014) concentra-se, por sua vez, na questão da emergência nos videogames, e de que forma se relaciona à produção de incerteza no jogador.

O problema principal de seu artigo é relativo à questão da liberdade do jogador que, condicionado pelas regras do jogo, estaria circunscrito a um universo fechado de possibilidades. O motivo seria a natureza cerceante das regras, que fecham o *campo de possibilidades* dando a ele um caminho necessário e, portanto, desprovido da capacidade de criação e invenção:

As regras controlam o sistema do jogo criando limitações na gama de possibilidades de eventos a fim de tornar o mundo do jogo uma realidade possível e viabilizar a experiência lúdica. Também diminuem a autonomia do sistema, valorizando a participação do jogador, e direcionam-no a eventos, como superar um obstáculo, progredir na narrativa, passar de fase, etc. Por exemplo, para o jogador abrir uma porta e passar para a aventura seguinte, ele precisa de uma chave mágica (regra 1). A chave só pode ser obtida derrotando um monstro (regra 2) e a magia só pode ser adquirida conversando com um mago (regra 3). Assim, o jogador é obrigado a explorar o ambiente em dois locais (monstro e mago) e ainda saber mais sobre a história ao conversar com o mago. (AUDI, 2014, p. 72)

Aqui, podemos ver que estamos diante de um conjunto de regras que definem meios de interioridade para as entidades do jogo. Apenas uma chave mágica pode abrir a porta – as regras, portanto, interiorizam na chave (e apenas nela) a capacidade de abrir, assim como na porta a capacidade de ser aberta. Cria-se uma situação análoga à alegoria de Virgínia Kastrup sobre o quebra-cabeça (2007, p. 27): se cada peça possui um lugar específico e “vizinhas” pré-determinadas, isso a transforma em um indivíduo fechado, destinado a um lugar pré-definido para exercer sua função interior. O mesmo pode ser dito do monstro e do mago.

Voltemo-nos, porém, para o famoso caso de *Life*, o jogo criado pelo matemático John Conway a partir das teorias de autômatos celulares dos pioneiros John von Neumann e Arthur Burks (1966). Trata-se de uma grade cartesiana bidimensional onde cada célula pode estar viva ou morta. O estado de cada célula é definido a cada turno, segundo apenas quatro regras. 1) Uma célula com menos de duas vizinhas vivas morrerá no próximo turno. 2) Uma célula com 2 ou 3 vizinhas vivas permanece viva no próximo turno. 3) Uma célula com mais de 3 vizinhas vivas morrerá no próximo turno. 4) Caso uma célula morta possua exatamente 3 vizinhas vivas, também estará viva no próximo turno.

Qual a diferença entre as quatro regras de Conway para as 3 regras descritas no cenário da chave e do mago? Obviamente, a diferença está no fato de que não são definidas propriedades interiores a cada célula, mas sim capacidades de agir *exteriormente*. As regras do primeiro tipo pré-definem um estado fixo para pareamentos pré-determinados de peças, como em um quebra-cabeça. As do segundo tipo também pré-definem certas exigências para que cada célula esteja viva ou morta. No entanto, cada peça influencia o estado das outras que a rodeiam: o resultado do primeiro turno de nascimentos e mortes de células é a matéria-prima para a metamorfose que o sistema sofrerá no próximo turno. Em outras palavras, as regras do primeiro tipo disparam

suas propriedades de dentro para fora, enquanto as da segunda espécie limitam-se a atuar dentro das bordas das entidades a que se referem.

Achar regras desse tipo não se trata de um trabalho fácil: o matemático John Conway dependeu de anos de trabalho para alcançar um sistema que produzisse resultados interessantes o bastante sem que a simplicidade das regras fosse comprometida. Uma de suas inspirações foi o jogo Go (figura 57). Anos depois da criação de *Life*, uma série de padrões interessantes em seu comportamento foi notado: era possível criar distribuições de células que causavam o aparecimento desaparecimento caótico de “peças” vivas e mortas no “tabuleiro”. Outros já demonstravam maior estabilidade: eram capazes de se formar blocos que permaneciam parados, outros que “cuspiam” outros blocos, como entidades que se reproduzem. Em outros termos as regras definiam comportamentos para células que se influenciavam mutuamente e geravam indivíduos de ordem superior com novas propriedades e capacidades. Depois de muita experimentação, foram descobertos padrões capazes de simular portas lógicas – partículas básicas que se comunicam para criar computadores – e, logo em seguida, máquinas de turing universais¹⁰², apenas com as 4 regras descritas acima (DeLANDA, 2009, p. 29)¹⁰³.

Independentemente de seu status de sistema complexo ou emergente, o que nos interessa do ponto de vista da conexão jogador-jogo é que, uma vez que regras definem a exteriorização da força de ação das entidades do jogo, o jogador passa a conviver com agências que não a sua própria. Cada uma delas serve de ignição para que ações externas aconteçam, gerando sistemas coerentes (como a organização de cidades, fortalezas e vilas) que se sustentam e *emergem* de eventos imprevisíveis e caóticos (como o comportamento individual de cada personagem que habita estes locais). Como observado por Audi, as limitações dessa imprevisibilidade diminuem a autonomia do sistema em relação ao jogador – embora seja questionável até que ponto isso “valoriza” o papel do jogador. Esse será o ponto de partida para a conclusão deste trabalho.

¹⁰² Máquinas de Turing são construtos matemáticos capazes de computar qualquer operação expressável matematicamente, enquanto máquinas de Turing universais são máquinas de Turing capazes de simular a construção de outras máquinas de Turing. Um exemplo de relativa facilidade de compreensão é um computador ou smartphone: o sistema operacional destas máquinas funciona computacionalmente (máquina de Turing), mas dentro deles é possível simular um computador menos poderoso, como um emulador de um computador antigo (algo que só uma máquina de Turing universal pode fazer).

¹⁰³ DeLanda mostra como novos usos de autômatos celulares são capazes de gerar “químicas artificiais”, gerando indivíduos emergentes capazes de simular equações diferenciais usadas no estudo de comportamento de fluidos, por exemplo. O autor chama essa relação de isomorfismo: não se trata de “imitar” um processo físico no computador, mas de criar um sistema cujo comportamento, turno a turno, possui propriedades semelhantes às de processos físicos. É exatamente essa abordagem utilizada por Tarn Adams para acrescentar potencial d emergência aos mundos de *Dwarf Fortress*: os rios seguem modelos matemáticos hidrográficos reais, assim como a simulação do crescimento de montanhas e vulcões e mesmo de distribuição de populações por terras como Ilasil (Cf. ADAMS, 2015)

CONSIDERAÇÕES FINAIS

Podemos, por fim, amarrar as pontas dos raciocínios desenvolvidos durante este trabalho. Primeiro, temos o cerne do primeiro capítulo: a ideia de que mundos de jogo aparecem ao jogador como uma mistura de forças que se transformam em estímulos visuaudiomotores, prolongando-se até o aparato sensorio-motor humano – e, por mais que tenhamos usado *Minecraft* como alvo de análise, tratamos em grande parte dessa ideia em abstrato. No segundo capítulo, tentamos dar corpo a essa abstração ao tratar de jogos específicos: os *survival games* e os *roguelikes*, nos quais percebemos a importância dos ritmos multidirecionais, da conectividade e da instabilidade – e da adaptação que exigem do jogador, que necessita reinventar os *loops* e ciclos de ação para se manter ativo no jogo. Finalmente, chegamos à escala microscópica dos mundos de jogo: a posição do jogador diante da conectividade de *URR* e da instabilidade de *Dwarf Frotress*. Nesses três momentos, nosso objetivo foi o mesmo: compreender de que modo as forças de um jogo podem atravessar o jogador, descentralizando-o.

Não por coincidência, o filósofo da técnica Gilbert Simondon pode nos ajudar uma última vez a concatenar todos estes raciocínios em um só, através de uma das ideias fundamentais de seu livro a respeito dos modos de existência dos objetos técnicos (2007). Cabe frisar que, embora Simondon demonstre preocupação detalhista com os meandros e funcionamentos internos das máquinas propriamente ditas, seu apreço por tais minúcias serve a um propósito igualmente indispensável em seu pensamento: a de investigar como máquina e humano podem trabalhar conjuntamente.

Simondon introduz a ideia ao comparar o trabalho do artesão ao trabalho de um operário industrial. O artesão é o centro de sua oficina: cada objeto só se refere ao artesão, enquanto o artesão tem acesso a todos. Objetos não conversam uns com os outros. Enquanto isso, o operário é o instrumento das máquinas: estas se conectam a todos os operários, mesmo que os operários trabalhem alienados uns dos outros. Sim, estão orquestrados em torno de uma mesma meta. Porém, a meta é conduzida a partir do ritmo das máquinas: não há oportunidade para que o operário imponha seus próprios ritmos e transforme o produto que fabrica – que dirá de estabelecer comunhão com outros trabalhadores de forma a inventar novos construtos.

Não que Simondon pense que a “culpa” seria das máquinas: ele vislumbra a força humana que orchestra a operação industrial: o capitalista. Este age como um jogador de *Civilization*, lançando sua perspectiva onipresente sobre cada detalhe da produção e do seu encadeamento global em uma linha de montagem. Para Simondon, certamente está alienado do

processo de produção. No entanto, o filósofo se distancia do que, para ele, representaria a teoria de Karl Marx: a raiz da alienação não é a detenção do capitalista sobre os meios de produção, mas a incapacidade do operário de alcançar uma atitude técnica em seu trabalho, ou seja, da falta de uma *autorregulação* entre seu modo de existência e o da máquina (SIMONDON, 2007, p. 136).

Aqui, Simondon mostra a particularidade de seu pensamento: mesmo que o operário esteja em uma posição desprivilegiada, tanto o capitalista como também o artesão – que gozam de uma suposta superioridade em relação a suas ferramentas – não alcançam o verdadeiro potencial do que o autor chama de atitude técnica, ou *tecnicidade*:

O agrupamento industrial não é o único realizável com os objetos técnicos: pode-se também realizar agrupamentos não-produtivos, que têm por fim ligar o homem à natureza, por meio de um encadeamento regrado de mediações organizadas, criar um acoplamento entre o pensamento humano e a natureza. O mundo técnico intervém aqui como sistema de convertibilidade¹⁰⁴ (SIMONDON, 2007, p. 262).

Essa ligação entre a agência humana e a máquina é perdida de vista quando o progresso e a finalidade *externa* da operação técnica são postos acima da causalidade *interna* da operação do indivíduo técnico. Em outros termos, quando a operação do ser humano harmoniza-se com a atividade do indivíduo técnico, os dois tornam-se uma força única, um ciclo em que “toda causalidade tem um sentido de finalidade, e toda finalidade um sentido de causalidade” (SIMONDON, 2007, p. 137).

Para alcançar esse ponto de tecnicidade, Simondon vislumbra a necessidade de uma atitude que se situa entre o domínio onipresente sobre o sistema e a alienação impotente e automatizada. Acreditamos que – agora já nos voltando totalmente ao contexto dos videogames – a busca por esse estágio é uma tendência premente em qualquer jogo (pelo menos na totalidade daqueles analisados neste trabalho), e a sua realização é essencialmente um fenômeno de comunicação.

O jogador começa alienado, desprovido das habilidades que o permitirão navegar na correnteza movimentada pelo funcionamento do jogo, até que seu aparato sensorial conecta-se à proliferação de estímulos advindos do mundo de jogo, como procuramos demonstrar no primeiro capítulo. Aqui, a organização das forças de cada mundo pode ser uma aliada ou um

¹⁰⁴ Tradução livre de: “El agrupamiento industrial no es el único realizable con los objetos técnicos: se puede también realizar agrupamientos no productivos que tienen por fin ligar al hombre a la naturaleza por medio de un encadenamiento reglado de mediaciones organizadas, crear un acoplamiento entre el pensamiento humano y la naturaleza. El mundo técnico interviene aquí como sistema de convertibilidad”.

empecilho: quanto mais o jogo usa o progresso do jogador como critério para sua construção, mais difícil será para o jogador estabelecer a autorregulação da qual fala Simondon¹⁰⁵.

Em seguida, os ritmos do jogo impõem seus padrões, aos quais o jogador tenta se adequar, de modo a reinventar seus próprios ciclos de ação, como a variedade de jogos abordados no segundo capítulo podem ilustrar. Novamente, percebemos os níveis crescentes que a instabilidade e a conectividade podem alcançar, tornando ainda mais improvável que a tendência da autorregulação seja alcançada. Por vezes, parece que a tendência pode se inverter: são tantas propriedades e capacidades diferentes, que o jogador poderia talvez estar sendo excessivamente repellido pelo jogo.

Por último, o jogador precisa alcançar novos níveis de conjunção e associação que o permitam agir produtivamente em mundos de jogo de alto grau de descentralização, como os analisados no terceiro capítulo. Aqui, sem dúvida, o jogo recobre o jogador de surpresas e empecilhos quase impossíveis de serem rastreados. Na realidade, jogos como *Dwarf Fortress* exigem investigação conjunta de vários jogadores em comunidade para que tutoriais, guias e truques aliviem a transição entre o completo desentendimento entre jogo e jogador até uma futura autorregulação entre ambos. Estariam esses jogos descentralizando o jogador de modo a serem impenetráveis, como se a dificuldade de adaptação fosse uma espécie de troféu a ser erguido, já que só os “fortes” seriam capazes de entrar em harmonia com esses jogos? Em suma, jogos como *Dwarf Fortress* de fato “desvalorizam” o jogador?

Antes de responder à pergunta, cabe apenas ratificar o objetivo desse estudo: investigar uma fagulha que se alastrou e incendiou o mundo dos videogames: *Minecraft*. Percebemos uma inconsistência entre o que o discurso comum dizia dos videogames e o que esse jogo trouxe à tona. Pouca consideração com o progresso e percurso do jogador e muito mais preocupação em erguer um mundo com fluxos de influência entre outras entidades: animais e monstros, montanhas e lagos, oceanos e ilhas, veias de minerais e rios de lava. Puxamos o fio deste problema de pesquisa, a partir do qual alcançamos a ideia de mundo de jogo, a genealogia dos *roguelikes* e a profundidade sem fundo de *Dwarf Fortress* e *URR*.

Não esgotamos o assunto. Ao escolher *Minecraft* como fio a ser puxado, deixamos uma pletora de tramas desprovidas de exame mais detalhado. Se falamos, por exemplo, em jogos de

¹⁰⁵ Não se trata aqui da dificuldade de realizar por falta de reflexos ou de informações. Um jogo de plataforma pode exigir que o jogador realize um salto que exija um nível de destreza alto demais. Assim como um game designer pode criar um jogo propositalmente difícil, no qual os inimigos possuem rapidez extrema e sejam invencíveis mesmo que o jogador os golpeie várias vezes. Trata-se, na realidade, de um tipo de dificuldade específico: o de compreender como cada entidade de mundo de jogo e se comporta e quais são suas capacidades e propriedades. *Subnautica*, por exemplo, não é um jogo difícil no primeiro sentido, mas sim no segundo.

estratégia como a série *Civilization* de modo a exemplificar a perspectiva onipresente, poderíamos ter abordado toda uma gama de jogos que reinventam essa relação. Casos como o a série *Mount & Blade*; os jogos da empresa Soldak Games como *Depths of Peril* (2008) e *Zombasite* (2016): todos jogos inspirados no gênero estratégia, mas que reduzem a perspectiva do jogador, exigindo que, com seu personagem solitário, lide com facções, exércitos, etc. Suas origens nos levariam a falar de jogos influentes como *Freelancer*, *Starflight 1 e 2* e a série *Space Rangers*.

Outro fio que poderia ser seguido são os jogos comumente chamados de *immersive sims* (simuladores imersivos), gênero inspirado em *Ultima Underworld*, que influenciou *Deus Ex*, *Thief*, *Hitman: Bloodmoney*, entre outros jogos, sobretudo da empresa Looking Glass Studios. Seus mundos investem em personagens com rotinas próprias, objetivos com múltiplas formas de serem completos e conexões imprevistas entre objetos e entidades. Se seguíssemos seus rastros, provavelmente chegaríamos a RPGs que investem nos mesmos atributos, como *Fallout* e *Ultima VII*.

Por último, um outro caminho possível seria ressaltar o sucesso de jogos como *Stardew Valley* (2015), que simula os ritmos da vida rural: ciclos de dia/noite, tempo de plantação e colheitas, personagens que saem de suas casas para trabalhar durante o dia e voltam à noite para dormir. Voltaríamos no tempo para falar de jogos semelhantes, como as séries *Rune Factory*, *Harvest Moon* e *Animal Crossing*. Em suma, há muitas vias inexploradas que poderiam ser abordadas para falar do modo como os jogos de videogame descentralizam seus jogadores – várias delas, inclusive, frutíferas o bastante para protagonizar pesquisas inteiras.

Do fio que escolhemos seguir, no entanto, chegamos a algumas conclusões – ou melhor, a problemas refinados e melhor colocados. Desenvolvemos uma abordagem epistemológica tanto no que diz respeito ao jogador quanto aos mundos de jogo – e, mais ainda, à conexão entre ambos. Fizemos um resgate histórico dos jogos que influenciaram a descentralização em *Minecraft* e uma análise de seus efeitos – ainda muito recentes – em jogos de sobrevivência como *Don't Starve*, *Subnautica* e *Factorio*. Procuramos nutrir essa história não apenas com registros e datas, mas buscando demonstrar como o objeto de estudo – a descentralização – apareceu e se transformou ao longo das últimas décadas.

Sendo assim, podemos reformular a pergunta feita acima: esses e outros tantos videogames estariam desprezando o papel do jogador, desvalorizando sua posição diante do jogo? O juízo de *valor* (literalmente) nos parece improdutivo. Na introdução, criamos a analogia que compara uma floresta (ambiente onde o sujeito se vê descentralizado) a uma casa cuidadosamente arquitetada (ambiente centralizado em volta de quem o ocupa). Com as

discussões do terceiro capítulo, acreditamos ter transformado o que era uma analogia em uma descrição concreta da experiência descentralizada, sintetizada pela ideia de tecnicidade em Simondon.

O jogador constrói técnicas de autorregulação entre si e o mundo de jogo, percebe estímulos inconsistentes e tenta associá-los entre si, construir um esquema dinâmico. Assim como o jogador de xadrez, o jogador de *Dwarf Fortress* “fareja” as melhores estratégias, como um cozinheiro que procura o ponto certo do tempero. Não se trata de um processo puramente lógico-racional – embora o seja, em parte. É, muito mais, uma sedimentação de hábitos aparentemente repetitivos, mas que serão reorganizados e recombinaados para recriar formas, gerar novas máquinas, conceber conexões com o mundo de jogo.

Para que esse fluxo comunicacional se estabeleça de modo eficiente, o jogador precisa responder às forças que movem o mundo de jogo, incorporá-las, tornando-as parte de si. Não se trata de uma metáfora: quando se joga um jogo como *Dwarf Fortress*, a lógica do jogo se impregna em nossas subjetividades de forma sensível. Em sua palestra a respeito deste fenômeno, o professor Frank Lantz resume o processo:

Algo extraordinário acontece: incrementalmente, pouco a pouco, sua mente cria novas passarelas entre seus olhos e seus dedos. Seus olhos cavam os ritmos da pista cada vez mais fundo, nas profundezas dos seus bancos de memória. Tal qual uma música que se inicia como notas em uma pauta, arrancando uma a uma deliberadamente, instrumentalmente. E então, torna-se um hábito. Segunda natureza, automático. Então, torna-se algo mais. Uma parte de você. Aquilo que estava do lado de fora, no mundo, é agora uma parte sua, com a qual você pode fazer coisas. E você pode senti-la. Aquela coisa que acontece o tempo todo está passando na frente dos seus olhos. E você pode senti-la tomando forma em seus dedos¹⁰⁶.

A descrição enfática e quase poética de Lantz não será de nada estranha para qualquer pessoa que tenha passado horas treinando, se aperfeiçoando em um videogame. Estamos acostumados a associar jogos ao aprendizado e, no momento em que ele toma forma, é por meio da dinâmica entre nossos sentidos e os ritmos que dançam na tela. É compreensível que ao longo da história dos videogames tenham surgido técnicas para facilitar esse processo de autorregulação: técnicas de centralização do jogador. Mas é igualmente razoável supor que certas tradições de jogos queiram explorar o extremo oposto desse espectro, vasculhando

¹⁰⁶Livre tradução de: “Something amazing happens: incrementally, bit by bit, your mind creates new passageways between your eyes and your thumbs. Your eyes dig the grooves of the track deeper and deeper into your memory banks. Like a song that starts out as notes on a graph, plucking them out, deliberately, instrumentally. And then becomes a habit. Second nature, automatic. and then becomes something more. A part of you. This thing that was out there in the world is now just a part of you, that you can do things with. And you can feel it. You can feel it. this thing that happens all the time is happening right in front of your eyes. And you can feel it taking shape in your hands.” Palestra ministrada por Frank Lantz na Game Developers Conference, em 2014. Fonte: <https://www.youtube.com/watch?v=CrO5IGoPgcw>.

esquemas de jogo que apresentem novos desafios para a apreensão humana. O nascimento de *Minecraft* denuncia uma dentre muitas dessas tradições. De jogos irregulares, inquietos, incompreensíveis e idiossincráticos. Por baixo dessa crosta, um núcleo fervente de passarelas virtuais entre nossos olhos e dedos.

REFERÊNCIAS

AARSETH, Espen. **Cybertext: perspectives on ergodic literature**. Baltimore/London: The John Hopkins University Press, 1997.

AARSETH, Espen. **I Fought the Law: Transgressive Play and The Implied Player**. 2007. Disponível em: <http://www.digra.org/wp-content/uploads/digital-library/07313.03489.pdf>. 2007. Acesso em 9/08/2016.

AARSETH, Espen et al. **A multi-dimensional typology of games**. 2003. Disponível em: <http://www.digra.org/wp-content/uploads/digital-library/05163.52481.pdf>. Acesso em:

ADAMS, Tarn. **Simulation Principles from Dwarf Fortress**. 2015. Disponível em: http://www.gameapro.com/GameAPro2/GameAPro2_Chapter41_Simulation_Principles_from_Dwarf_Fortress.pdf. Acesso em: 29/01/2018.

ARSENAULT, Dominic; PERRON, Bernard. **In the Frame of the Magic Cycle: The Circle(s) of Gameplay**. In: PERRON, Bernard and WOLF, Mark J. P. *The video game theory reader 2*. New York: Routledge, 2009.

ANDERSON, Joseph; ANDERSON, Barbara. *Motion Perception in Motion Pictures*. In: DE LAURETIS, Teresa; HEATH, Sean. **The Cinematic Apparatus**. Londres: McMillian Press, 2008.

AUDI, Gustavo. **Emergência e Incerteza em Jogos de Videogame**. 2014. Disponível em: <http://dx.doi.org/10.11606/issn.2238-7714.no.2014.83586>. Acesso em: 29/01/2018.

BACKUS, Kenny. *Managing OutputBoredom versus Chaos*. In: SHORT, Tanya e ADAMS, Tarn. **Procedural Generation in Game Design**. Boca Raton: CRC Press, 2017.

BENJAMIN, Walter. **Magia e técnica, arte e política: Obras escolhidas**. São Paulo: Brasiliense, 1987.

BENTLEY, Wilson. **Snow Beauties**. Disponível em: <http://snowflakebentley.com/WBsnow.htm>. Acesso em: 29/01/2018.

BERGSON, Henri. **O Esforço Intelectual**. 2006. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-31732006000100008. Acesso em: 29/01/2018.

BERGSON, Henri. **O Pensamento e o Movente**. São Paulo: Martins Fontes, 2006.

BONNER, Mark. **Analyzing the Correlation of Game Worlds and Built Reality: Depiction, Function and Mediality of Architecture and Urban Landscapes**. 2014 Disponível em: https://library.med.utah.edu/e-channel/wp-content/uploads/2016/04/digra2014_submission_15.pdf. Acesso em: 29/01/2018.

BOOT, Walter et al. **The effects of video game playing on attention, memory, and executive control**. 2008. Disponível em: <https://doi.org/10.1016/j.actpsy.2008.09.005>. Acesso em: 29/01/2018.

CALLEJA, Gordon. **In-Game: From Immersion to Incorporation**. Cambridge, Massachusetts: MIT Press, 2011.

CLARK, Andy. **Supersizing the Mind: Embodiment, Action, and Cognitive Extension**. Oxford: Oxford University Press, 2008.

CONSALVO, Mia. **Cheating: Gaining Advantage in Videogames**. Cambridge: The MIT Press, 2007.

COSTELLO, Brigid Mary. **The Rhythm of Game Interactions: Player Experience and Rhythm in Minecraft and Don't Starve**. 2016. Disponível em: <https://doi.org/10.1177/1555412016646668>. Acesso em: 29/01/2018.

CRADDOCK, David. **Dungeon Hacks: How Nethack, Angband, and Other Roguelikes Changed the Course of Video Games**. Press Start Press, 2016.

CRAWFORD, Chris. **Process Intensity**. 1989. Disponível em: <http://www.erasmatazz.com/library/the-journal-of-computer/jcjd-volume-1/process-intensity.html>. Acesso em: 29/01/2018.

DELANDA, Manuel. **Intensive Science and Virtual Philosophy**. London: Continuum, 2005.

DELANDA, Manuel. **Philosophy and Simulation: The Emergence of Synthetic Reason**. London: Continuum, 2011.

DELEUZE, Gilles; GUATTARI, Felix. **Mil Platôs: capitalismo e esquizofrenia, vol.4** São Paulo: ed. 34 2008. 4ª reimpressão.

DOMINJON, Lionel et al. **Influence of control/display ratio on the perception of mass of manipulated objects in virtual environments**. 2005. Disponível em: <https://doi.org/10.1109/VR.2005.1492749>. Acesso em: 29/01/2018.

DONALD, Merlin. **Origins of the modern mind**. Cambridge: Harvard University Press, 1991.

DORVAL, Michel; PÉPIN, Michel. **Effect of Playing a Video Game on a Measure of Spatial Visualization**. 1986. Disponível em: <http://journals.sagepub.com/doi/abs/10.2466/pms.1986.62.1.159> Acesso em: 29/01/2018.

ERMI, Laura; MÄYRÄ, Frans. **Player-Centred Game Design: Experiences in Using Scenario Study to Inform Mobile Game Design**. 2005. Disponível em: http://www.gamestudies.org/0501/ermi_mayra/ Acesso em: 29/01/2018.

FALCÃO, Thiago. **Não humanos em Jogo: Agência e Prescrição em World of Warcraft**. 2014. Disponível em: www.poscom.ufba.br/publicacoes/teses-2/. Acesso em: 29/01/2018.

FALCÃO, Thiago; FERREIRA, Emmanoel. **Through the Looking Glass: Weavings between the Magic Circle and Immersive Processes in Video Games.** 2016. Acesso em: <https://pdfs.semanticscholar.org/1056/d4a278200737aa1a1ca04c6bc22b4eef8645.pdf>. Acesso em: 29/01/2018.

FASTERHOLDT, Martin. **You say jump, I say how high?: A study of jumping in 2D platform games.** Disponível em: http://martinf.dk/jump/Jumping_Thesis_Fasterholdt_09.pdf. Acesso em: 29/01/2018.

FROMM, Jochen. **The Emergence of Complexity.** 2004. Disponível em: <http://www.uni-kassel.de/upress/online/frei/978-3-89958-069-3.volltext.frei.pdf>. Acesso em: 29/01/2018.

GALLOWAY, Alexander. **Gaming: Essays on Algorithmic Culture.** Minneapolis e Londres: University of Minnesota Press, 2006.

GARDA, Maria B. **Neo-rogue and the essence of roguelikeness.** 2013. Disponível em: <http://ptbg.org.pl/dl/134/Maria%20B.%20GARDA%20-%20Neo-rogue%20and%20the%20essence%20of%20roguelikeness.pdf>. Acesso em: 29/01/2018.

GIBSON, James J. **The Ecological Approach to Visual Perception.** Londres: Psychology Press, 1986.

GRAVNER, Janko; GRIFFEATH, David. **Modeling snow-crystal growth: A three-dimensional mesoscopic approach.** 2009. Disponível em: <https://doi.org/10.1103/PhysRevE.79.011601>. Acesso em: 29/01/2018.

GREEN, C. Shawn; BAVELIER, Daphne. **Action video game modifies visual selective attention.** 2004. Disponível em <https://doi.org/10.1038/nature01647>. Acesso em: 29/01/2018.

GRIFFTH, Jerry et al. **Differences in Eye-Hand Motor Coordination of Video-Game Users and Non-Users.** 1983. Disponível em: <http://journals.sagepub.com/doi/abs/10.2466/pms.1983.57.1.155?journalCode=pmsb>. Acesso em: 29/01/2018.

GRINBLAT, Jason; BUCKLEW, Brian. **Subverting Historical Cause & Effect: Generation of Mythic Biographies in Caves of Qud.** Disponível em: <https://doi.org/10.1145/3102071.3110574>. Acesso em: 29/01/2018.

GRUSIN, Richard. **Radical Mediation.** 2015. Disponível em: <https://uwm.edu/english/wp-content/uploads/sites/109/2015/09/Grusin-Radical-Mediation.pdf>. Acesso em: 29/01/2018.

HOFSTADTER, Douglas. **Gödel, Escher, Bach: An Eternal Golden Braid.** Basic Books, Inc., New York 1979.

HUIZINGA, Johan. **Homo Ludens.** São Paulo: Perspectiva, 5 ed., 2004.

ILIADIS, Andre. **Informational Ontology: The Meaning of Gilbert Simondon's Concept of Individuation.** 2013. Disponível em: <https://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1015&context=cpo>. Acesso em: 29/01/2018.

JAKOBSSON, Mikael. **The Achievement Machine**: Understanding Xbox 360 Achievements in Gaming Practices. 2011. Disponível em: <http://gamestudies.org/1101/articles/jakobsson?report=reader>. Acesso em: 29/01/2018.

JØRGENSEN, Kristine. **Gameworld Interfaces**. Cambridge/London: MIT Press, 2013.

JOHNSON, Mark. **Modelling Cultural, Religious and Political Affiliation in Artificial Intelligence Decision**. 2015. Disponível em: https://www.cs.kent.ac.uk/events/2015/AISB2015/proceedings/aiAndGames/AI-games-15_submission_01--MarkJohnson-modelling.pdf. Acesso em: 29/01/2018.

JUUL, Jesper. **Half-Real**: Video Games between Real Rules and Fictional Worlds. Cambridge, Massachusetts: MIT Press, 2005.

JUUL, Jesper. **A casual revolution**: reinventing video games and their players. Cambridge, Massachusetts: MIT Press, 2009.

JUUL, Jesper. **The Art of Failure**: An Essay on the Pain of Playing Video. Cambridge: MIT Press, 2013.

KASTRUP, Virgínia. **A invenção de si e do mundo**: Uma introdução do tempo e do coletivo no estudo da cognição. Belo Horizonte: Autêntica, 2007.

KIRSCH, David; MAGLIO, Paul. **On Distinguishing Epistemic from Pragmatic Action**. 1992. Disponível em: https://doi.org/10.1207/s15516709cog1804_1. Acesso em: 29/01/2018.

KRAPP, Peter. **Noise Channels**: Glitch and Error in Digital Culture. University of Minnesota Press 2011.

LAUREL, Brenda. **The Art of Human-Computer Interface Design**. Reading MA: Addison Wesley, 1990.

LUNDGREN, Sus e BJÖRK, Staffan. **Game Mechanics**: Describing Computer-Augmented Games in Terms of Interaction. 2003. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.5147>. Acesso em: 29/01/2018.

MANOVICH, Lev. **The language of new media**. Cambridge, Mass: Mit Press, 2001.

MATURANA, Humberto e VARELA, Francisco. **Autopoiesis and Cognition**: The Realization of the Living. Dordrecht: D. Reidel Publishing Co., 1980.

MCCAULEY, Michael. **Cybersickness**: Perception of Self-Motion in Virtual Environment. 1992. Disponível em: <https://doi.org/10.1162/pres.1992.1.3.311>. Acesso em: 29/01/2018.

MESSIAS, José. **“Saudações Do Terceiro Mundo”**: games customizados, gambiarra e habilidades cognitivas na cultura hacker. Disponível em: www.pos.eco.ufrj.br/site/download.php?arquivo=upload/tese_jfranco_2016.pdf. Acesso em: 29/01/2018.

MONFORT, Nick; BOGOST, Ian. **Racing the Beam: the Atari Video Computer System**. Cambridge: The MIT Press, 2009.

MORAVEC, Hans. **Mind Children: The future of robot and human intelligence**. Cambridge, Massachusetts: Harvard University Press, 1988.

MUSSA, Ivan. **Exploração de ambientes em jogos eletrônicos**. 2014. Disponível em: <http://www.ppgcom.uerj.br/wp-content/uploads/Disserta%C3%A7%C3%A3o-Ivan-Mussa.pdf>. Acesso em: 29/01/2018.

NITSCHKE, Michael. **Video Game Spaces: Image, Play, and Structure in 3D Worlds**. Cambridge: The MIT Press, 2009.

NØRGÅRD, Rikke. **Just Like Driving: Computer Games as Actual Practice and Objects of Presentation**. 2009. Disponível em: https://pure.au.dk/ws/files/19526642/JUST_LIKE_DRIVING.pdf. Acesso em: 29/01/2018.

OLIVEIRA, Luiz Alberto. Biontes, Bioides e Borgues. In: NOVAES, Adauto. **O Homem máquina: A ciência manipula o corpo**. São Paulo: Companhia das Letras, 2003.

OROSY-FILDES, Cindy; ALLAN, Robert. **Videogame Play: Human Reaction Time to Visual Stimuli**. 1989. Disponível em: <https://doi.org/10.2466/pms.1989.69.1.243>. Acesso em: 29/01/2018.

PARKER, Felan. **The Significance of Jeep Tag: On Player-Imposed Rules in Video Games**. 2008. Disponível em: <http://journals.sfu.ca/loading/index.php/loading/article/view/44>. Acesso em: 29/01/2018.

PERANI, Letícia. **Interfaces gráficas e os seus elementos lúdicos: aproximações para um estudo comunicacional**. (Dissertação de Mestrado). RJ: PPGC/UERJ, 2008.

PEREIRA, Vinicius. **G.A.M.E.S. 2.0 – Gêneros e Gramáticas de Arranjos e Ambientes Midiáticos Mediadores de Experiências de Entretenimento, Sociabilidades e Sensorialidades**. 2008. Disponível em: http://www.compos.org.br/data/biblioteca_294.pdf. Acesso em: 29/01/2018.

PIAS, Claus. The Game Player's Duty: The User as the Gestalt of the Ports. In: HUHTAMO, Erkki e PARIKKA, Jussi. **Media Archaeology: Approaches, Applications, and Implications**. University of California Press: 2011.

REGIS, Fátima; PERANI, Letícia. Comunicação e entretenimento na Cibercultura: repensando as articulações entre lúdico, cognição e tecnologia. **E-Compós**, v.13, n.2, maio/ago. 2010.

RÉGIS, Fátima; TIMPONI, Raquel; MAIA, Alessandra. **Cognição integrada, encadeada e distribuída: breve discussão dos modelos cognitivos na cibercultura**. 2012. Disponível em: <http://revistacmc.espm.br/index.php/revistacmc/article/view/346/pdf>. Acesso em: 29/01/2018.

SCULLY-BLAKER, Rainforest. **A Practiced Practice: Speedrunning Through Space With de Certeau and Virilio**. 2014. Disponível em: <http://gamestudies.org/1401/articles/scullyblaker>. Acesso em: 29/01/2018.

- SICART, Miguel. **Against Procedurality**. Disponível em: http://gamestudies.org/1103/articles/sicart_ap. Acesso em: 29/01/2018.
- SICART, Miguel. **Loops and Metagames: Understanding Game Design Structures**. 2015. Disponível em: <https://pdfs.semanticscholar.org/af20/58cc881a0c26228a356a519485a8d2d662d5.pdf>. Acesso em: 29/01/2018.
- SIMONDON, Gilbert. **El modo de existencia de los objetos técnicos**. Argentina: Prometeo Libros, 2007.
- SMITH, Jonas Heide. **Plans And Purposes: How Videogame Goals Shape Player Behaviour**. 2006. Disponível em: <http://jonassmith.dk/weblog/wpcontent/dissertation1-0.pdf>. Acesso em: 29/01/2018..
- SODRÉ, Muniz. **A ciência do comum: Notas para o método comunicacional**. Petrópolis, RJ, Vozes: 2014
- SPERLING, George. **Movement perception in computer-driven visual displays**. 1976 Disponível em: <https://link.springer.com/content/pdf/10.3758/BF03201762.pdf>. Acesso em: 29/01/2018.
- STENROS, Jakko. **Playfulness, Play and Games: A Constructionist Ludology Approach**. Tampere University Press: Tampere, 2015.
- SWINK, Steve. **Game Feel: A Game Designer's Guide to Virtual Sensation**. Amsterdam: Morgan Kaufmann/Elsevier, 2009.
- THOMPSON, Evan. **Life and mind: From autopoiesis to neurophenomenology**. 2004. Disponível em: <https://pdfs.semanticscholar.org/c731/32b168018e61c2f6a5202437fbe1e00c8d04.pdf>. Acesso em: 29/01/2018.
- TOGELIUS, Julian; SHAKER, Noor and NELSON, Mark J. **Procedural Content Generation in Games: A Textbook and an Overview of Current Research**. 2015. Disponível em: <http://pcgbook.com/>. Acesso em: 29/01/2018.
- VELLA, Daniel. **No Mastery Without Mystery: Dark Souls and the Ludic Sublime**. 2015. Disponível em: <http://gamestudies.org/1501/articles/vella>. Acesso em: 29/01/2018.
- VON NEUMANN, John; BURKS, Arthur. **Theory of Self-Reproducing Automata**. 1966. Disponível em: <http://cba.mit.edu/events/03.11.ASE/docs/VonNeumann.pdf>. Acesso em: 29/01/2018.
- WEISLER, Ann; McCALL, Robert. **Exploration and play: Resume and redirection**. American Psychologist, v. 31, p. 492-508, 1976.
- WOLF, Mark. **The Medium of the Video Game**. Austin: University of Texas Press, 2001.

ZAGAL, J.P. et al. **Towards an ontological language for game analysis**. 2005. Disponível em: <http://homes.lmc.gatech.edu/~cfernandez/papers/OntologyDIGRA2005.pdf>. Acesso em: 29/01/2018.

APÊNDICE - Ludografia comentada

1001 Spikes (2013) / Designer: Samu Wosada e Tyrone Rodriguez / Plataformas: Linux, Microsoft Windows, Nintendo 3DS, OS X, PlayStation 4, PlayStation Vita, Wii U, Xbox One, Nintendo Switch

Aquaria (2007) / Designer: Alec Holowka e Derek Yu / Plataformas: Android, iOS, Linux, OS X e Microsoft Windows

Assassin's Creed (franquia – 2007-2017) / Designers originais: Patrice Désilets, Jade Raymond e Corey May / Plataformas: PlayStation 3, PlayStation 4, Xbox 360, Xbox One, Wii U, Microsoft Windows, OS X, Nintendo DS, PlayStation Portable, PlayStation Vita, iOS, HP webOS, Android, Nokia Symbian e Windows Phone

Assassins's Creed II (2009) / Designer: Patrice Desilets / Plataformas: iPhone OS, Microsoft Windows, PlayStation 3, Xbox 360, PlayStation 4, Xbox One

Audiosurf (2008) / Designer: Dylan Fitterer / Plataformas: Microsoft Windows e Zune HD

Call of Duty: Modern Warfare 3 (2011) / Designer: Steve Ackrich / Plataformas: Microsoft Windows, OS X, PlayStation 3, Xbox 360, Wii, Nintendo DS.

Civilization (1991) / Designer: Sid Meier / Plataformas: MS-DOS, Microsoft Windows, Amiga, Macintosh, Atari ST, Sega Saturn, Super NES, PlayStation, N-Gage

Dance Dance Revolution (franquia – 1998-2016) / Designers originais: Katsunori Okita e Kazuya Takahashi / Plataformas originais: Arcade, Playstation

Dark Souls (2011) / Designer: Hidetaka Miyazaki / Plataformas: PlayStation 3, Xbox 360, Microsoft Windows, Nintendo Switch, PlayStation 4, Xbox One

Destiny (franquia – 2014-2017) / Designer: James McQuillan e Luke Smith / Plataformas: Microsoft Windows, Playstation 3, PlayStation 4, Xbox 360, Xbox One

Dragon Age: Inquisition (2015) / Designer: Mike Laidlaw / Plataformas: Microsoft Windows, Playstation 3, PlayStation 4, Xbox 360, Xbox One

Elite (1984) / Designers: David Braben e Ian Bell / Plataformas: BBC Micro, Acorn Electron, Apple II, Amstrad CPC, Commodore 64, ZX Spectrum, MSX, Tatung Einstein, IBM PC compatible, Acorn Archimedes, Commodore Amiga, Atari ST e NES

Far Cry (franquia – 2004-2018) / Designer original: Cevat Yerli / Plataformas: Microsoft Windows, Arcade, PlayStation 3, PlayStation 4, Wii, Xbox, Xbox 360 e Xbox One

Far Cry 3 (2012) / Designer: Patrick Plourde / Plataformas: Microsoft Windows, PlayStation 3, Xbox 360

Final Fantasy XV (2017) / Designers: Hajime Tabata e Takatsugu Nakazawa / Plataformas: Playstation 4, Xbox One, Microsoft Windows

Ghost Recon: Wildlands (2017) / Designer: Dominic Butler / Plataformas: Microsoft Windows, PlayStation 4, Xbox One

Gran Turismo 6 (2013) / Designer: Kazunori Yamauchi / Plataforma: Playstation 3
Tetris (1984) / Designer: Alexey Pajitnov / Plataforma original: Electronika 60.

Grand Theft Auto III (2001) / Designers: Adam Fowler e Obbe Vermeij / Plataformas: Playstation 2, Microsoft Windows, Xbox, Playstation 3 (PSN), OS X, iOS, Android
Assassin's Creed (2007)

Grand Theft Auto V (2013) / Designers: Leslie Benzies e Imran Sarwar / Plataformas: PlayStation 3, Xbox 360, PlayStation 4, Xbox One, Microsoft Windows

Grow Up (2016) / Designer: Dale Scullion / Plataformas: Microsoft Windows, PlayStation 4, Xbox One

Guitar Hero (franquia – 2005-2017) / Designer original: Greg LoPiccolo / Plataforma original: Playstation 2.

Half-Life 2 (2004) / Design colaborativo (empresa: Valve) / Plataformas: Microsoft Windows, Xbox, Xbox 360, PlayStation 3, OS X, Linux, Android

Horizon: Zero Dawn (2017) / Designer: Mathijs de Jonge / Plataforma: Playstation 4

Infiniminer (2009) / Designer: Zach Barth / Plataforma: Microsoft Windows

Maniac Mansion (1987) / Designer: Ron Gilbert e Garry Winnick / Plataformas: Commodore 64, Apple II, IBM PC, Amiga, Atari ST, NES

Mass Effect: Andromeda (2017) / Designers: Mac Walters e Ian Frazier / Plataformas: Microsoft Windows, PlayStation 4, Xbox One

Metal Gear Solid V (2015) / Designer: Hideo Kojima / Plataformas: Microsoft Windows, Playstation 3, PlayStation 4, Xbox 360, Xbox One

Mirror's Edge: Catalyst (2016) / Designers: Per Juhlén e Erik Odeldahl / Plataformas: Microsoft Windows, PlayStation 4, Xbox One

Mystery House (1980) / Designer: Roberta Williams / Plataforma: Apple II

Nier: Automata (2017) / Designers: Yoko Taro, Takahisa Taura, Isao Negishi e Takada Shohei. Plataformas: Microsoft Windows, PlayStation 4, Xbox One

Pong (1972) / Designers: Allan Alcorn e Nolan Bushnell / Plataforma: Arcade

Populous (1989) / Designer: Peter Molyneux / Plataformas: Acorn Archimedes, Amiga, Atari ST, MS-DOS, PC Engine, Game Boy, NEC PC-9801, SNES, Mega Drive, Master System, Sharp X68000, Mac OS e FM Towns

Portal (2007) / Designer: Kim Swift / Plataformas: Microsoft Windows, PlayStation 3, Xbox 360, OS X, Linux, Android

Portal 2 (2011) / Designer: Joshua Weier / Plataformas: Microsoft Windows, OS X, Linux, PlayStation 3, Xbox 360

Pro Evolution Soccer (franquia – 2001-2017) / Designer original: Shingo “Seabass” Takatsuka / Plataforma original: Playstation.

Quake (1996) / Designer: John Romero, American McGee, Sandy Petersen e Tim Willits / Plataformas: MS-DOS, Mac OS, Sega Saturn, Nintendo 64, Amiga, Linux

Rayman (franquia – 2005-2015) / Designer original: Michael Ancel / Plataforma original: Atari Jaguar

RuneScape (2001) / Designers: Andrew Gower e Paul Gower / Plataformas: Microsoft Windows / OS X / Linux

Shenmue (1999) / Designer: Yu Suzuki / Plataformas: Dreamcast, Xbox, Android, Microsoft Windows, PlayStation 4

Skull & Bones (em desenvolvimento) / Designer: Bill Money / Plataformas: Microsoft Windows, PlayStation 4, Xbox One

SimCity (1989) / Designer: Will Wright / Plataformas: Acorn Archimedes, Acorn Electron, Amiga, Amstrad CPC, Atari ST, BBC Micro, Browser, C64, CDTV, DESQview, DOS, EPOC32, FM Towns, GBA, iOS, Linux, Mac OS, Mobile phone, NeWS, OLPC XO-1, OS/2, PC-98, SNES, Tk, Unix, Windows, X11 TCL, X68000, ZX Spectrum

Starlink: Battle for Atlas (em desenvolvimento) / Designer: Laurent Malville / Plataformas: Microsoft Windows, PlayStation 4, Xbox One

Steep (2017) / Designers: Igor Monceau e Arnaud Ragot / Plataformas: Microsoft Windows, PlayStation 4, Xbox One e Nintendo Switch

Sunset Overdrive (2014) / Designers: Marcus Smith, Drew Murray e Cameron Christian / Plataforma: Xbox One

Super Mario Bros. (1985) / Designers: Shigeru Myiamoto e Takashi Tezuka / Plataforma: NES

The Colonel’s Bequest (1989) / Designers: Roberta Williams e Chris Iden / Plataformas: Commodore Amiga, Atari ST, MS-DOS

The Crew (2014) / Designers: Stephane Beley e Julian Gerighty / Plataformas: Microsoft Windows, PlayStation 4, Xbox 360

The Division (2016) / Designers: Magnus Jansen e Ryan Barnard / Plataformas: Microsoft Windows, PlayStation 3, Xbox 360

The Elder Scrolls II: Daggerfall (1996) / Designer: Julian Lefay, Bruce Nesmith e Ted Peterson / Plataforma: MS-DOS

The Sims (2000) / Designer: Will Wright / Plataformas: Microsoft Windows, Max OS, Playstation 2, Xbox, Gamecube

The Witcher 3 (2015) / Designers: Konrad Tomaszkiewicz, Mateusz Kanik, Sebastian Stępień, Grzegorz MocarSKI e Matthew Steinke / Plataformas: Microsoft Windows, PlayStation 4, Xbox One

The Witness (2016) / Designer: Jonathan Blow / Plataformas: Microsoft Windows, iOS, Playstation 4

Tom Clancy's Rainbow Six (franquia – 1998-2015) / Designer original: Brian Upton / Plataformas: Microsoft Windows, Nintendo 64, PlayStation, Mac OS, Game Boy Color, Dreamcast, PlayStation Network

Uncharted 3 (2011) / Designers: Amy Hennig, Justin Richmond, Richard Lemarchand e Jacob Minkoff / Plataformas: Playstation 3 e Playstation 4

Watchdogs 2 (2016) / Designers: Jonathan Morin e Danny Bélanger / Plataformas: Microsoft Windows, PlayStation 4, Xbox One

World of Warcraft (2004) / Designers: Rob Pardo, Jeff Kaplan e Tom Chilton / Plataformas: Microsoft Windows e OS X

Roguelikes

- **Rogue** (1980) / Designers: Michael Toy, Ken Arnold e Glenn Wichman / Plataformas selecionadas: Amiga, Amstrad CPC, Atari 8-bit, Atari ST, Commodore 64, DOS, Macintosh, TOPS-20, TRS-80 CoCo, Unix, ZX Spectrum

Gameplay de apoio: <https://www.youtube.com/watch?v=I2Q9iZupsqA>

Rogue apresenta ao jogador uma série de cavernas – conjuntos de salas interligados por corredores. Cada sala pode abrigar uma gama de itens, armadilhas e/ou monstros. A posição, tamanho e interligações de corredores de cada sala é “sorteada” ao início de cada partida pelo sistema de geração procedimental do jogo. O mesmo vale para a posição inicial de cada item, das armadilhas e inimigos. O jogador precisa achar a passagem para a camada seguinte da caverna, aprofundando-se nos seus andares até chegar ao último, recuperar o Amuleto de Yendor e retornar a salvo à superfície. Embora a disseminação de *Rogue* esteja longe de poder ser explicada em algumas linhas, alguns fatores demonstraram-se essenciais para sua pervasividade: sua facilidade de reprodução (já que trabalhava com caracteres emecanismos de programação presentes na maioria das universidades ocidentais), sua agilidade e facilidade de abordagem (pelo menos em comparação aos jogos textuais que tematizavam as mesmas aventuras) e, finalmente, o elemento de maior complexidade: a experiência de lidar com um sistema em constante metamorfose.

- ii. **Hack** (1984) / Designers originais: Brian Hardy, Michael Thome, Jay Felanson e Andrew Brouwer / Plataforma: Usenet

Gameplay de apoio: https://www.youtube.com/watch?v=DWSKf_rsorM

Hack, criação conjunta de alunos de ensino médio com seu professor de computação, é um incremento de *Rogue*: sedimenta-se nas mesmas bases de cavernas niveladas povoadas por monstros, tesouros e armadilhas. Adiciona, porém, ingredientes de complexidade ao multiplicar a quantidade de níveis na caverna, a variabilidade da topologia das salas e corredores, os tipos de armas, habilidades mágicas, tipos de inimigos, entre outros fatores de instabilidade. O resultado é um jogo que inaugura a tendência colaborativa dos *roguelikes* além de germinar a semente de complexidade que se atualizaria em jogos como *Spelunky*, *Cataclysm* e *UnReal World*.

- iii. **NetHack** (1987) / Design colaborativo (<https://tung.github.io/nethack-www/how-to-contribute.html>) / Plataformas selecionadas: UNIX, Atari 520/1040ST, Commodore Amiga, IBM PC, OpenVMS, MS-DOS, Microsoft Windows, Linux, OS X, Android, iOS

Gameplay de apoio:

<https://www.youtube.com/playlist?list=PL5A7CB3958FD23935>

NetHack é um jogo sem desenvolvedor oficial, levado a cabo por um comunidade que se reconstitui e se mantém ativa desde 1987. Baseado em *Hack*, o jogo leva às últimas conseqüências a tendência estabelecida por seu antecessor: intensificar o potencial de conexões entre entidades do sistema e, conseqüentemente, disparar efeitos imprevisíveis de instabilidade. Também ficou conhecido pela sua dificuldade, provocada exatamente pela gama de opções de jogo. O jogador deve escolher uma classe (desde ladrão, guerreiro e mago até turista ou samurai), cada qual com suas capacidades. Outro atributo influente é a presença de personagens não-jogáveis e não-hostis, como vendedores – ou personagens que ajudam o jogador e/ou o atacam sob certas condições.

- ii. **Moria** (1983) / Designer original: Robert Koeneke / Plataforma original: VAX 11/780

Gameplay de apoio: <https://www.youtube.com/watch?v=opy0OCkBOg4>

Moria é a âncora de uma das duas ramificações de *Rogue*, tendo se sedimentado como uma influência tanto sobre a camada ficcional quanto sobre a confecção do mundo de jogo. Instaura tanto a influência direta da obra de J. R. R. Tolkien (até então apenas indireta, por meio do jogo RPG de mesa *Dungeons & Dragons*) quanto a tendência de construção de um cenário além das cavernas. *Moria* apresenta uma vila com vendedores de armas, armaduras e poções, por exemplo. Seus descendentes incrementariam essa fórmula que une forte background ficcional com extensão e variabilidade do mundo de jogo, manifestada em jogos como *Caves of Qud* e *Ultima Ratio Regum*. *Moria*, assim como *NetHack*, também contribui para tradição da concessão de comportamentos distintos para personagens, como pedintes que podem implorar por dinheiro nas cidades e até roubar moedas do jogador.

- iii. **Umoria** (1987) / Designers: Robert Koeneke, Jimmey Wayne Todd Jr., James E. Wilson / Plataformas: Microsoft Windows, Linux, MS-DOS, Mac Classic, Atari ST, Amiga, Unix

Gameplay de apoio:

https://www.youtube.com/playlist?list=PLSX2U_ZE4HukONDyOtHxf893Y9WBSgH-q

Umoria é uma versão de *Moria* “portada” (reescrita) para a linguagem de programação “C” e lançada na plataforma UNIX. Além de corrigir bugs e aprimorar o processamento do código, adiciona algumas funcionalidades, como a capacidade de customizar o personagem na tela de início do jogo.

- iii. **Imoria** (1987) / Designers: Kenneth Case, Mary Conner, Robert DeLoura, Dan Flye, Todd Gardiner, Dave Jungck, Andy Walker e Dean Yasuda / Plataforma: Linux

Gameplay de apoio: não encontrado

Imoria foi uma bifurcação de *Moria* desenvolvida na Universidade de Washington. Seu traço diferencial foi adicionar variabilidade à dinâmica do jogo: certos locais do seu mundo eram inundados por água, por exemplo. A cidade também ganhou em instabilidade: diferentemente de seu antecessor, *Imoria* gerava a cidade procedimentalmente, mudando a conformação, distribuição e itens vendidos em lojas.

- iii. **Angband** (1990) / Designers originais: Alex Cutler e Andy Astrand / Plataforma: Unix

Gameplay de apoio: <https://www.youtube.com/playlist?list=PL120B20F201414550>

Angband prolonga a pretensão de seus “irmãos”, adicionando ainda mais variedade e imprevisibilidade à geração da cidade, caverna, monstros e seus comportamentos, itens, classes de personagens, habilidades mágicas, etc. *Angband* torna-se, graças ao seu código aberto e elegantemente escrito e polido, um novo nó de onde saem muitos “netos” de *Moria*.

- iv. **Zangband** (1994) / Designer: Topi Ylinen / plataformas: Unix, MS-DOS, Microsoft Windows, OS X

Gameplay de apoio: <https://www.youtube.com/watch?v=XFjveoF29K4>

Zangband toma um dos primeiros passos para uma ampliação considerável do campo de jogo e da variedade de ambientes e biomas nos roguelikes. Adiciona múltiplas cidades, cavernas secundárias, missões alternativas, entre outros fatores que contribuem para estender a dimensão do mundo de jogo – sem deixar de usufruir e incrementar em certos pontos a variedade de inimigos, personagens e itens e seus respectivos comportamentos.

- iv. The Troubles of Middle Earth** (2000) / Designer: DarkGod / Plataformas: Unix, MS-DOS, Windows, Mac OS X

Gameplay de apoio: https://youtu.be/IbIH5LGFH_s?list=PLYJarq38-WXGvRcJ-HaaJw-Im_pR3bSP0&t=1097

The Troubles of Middle Earth (posteriormente renomeado como *Tales of Middle Earth*) expande o conteúdo ficcional baseado na obra de J. R. R. Tolkien, bem como segue a tradição estabelecida por Zangband de (re)criar cidades, florestas, fortalezas, entre outros ambientes e as respectivas entidades que os habitam.

- iv. Tales of Maj'Eyal** (2013) / Designer: DarkGod / Plataformas: Windows, Unix, Mac OS X

Gameplay de apoio:

https://www.youtube.com/playlist?list=PLB_LlwcQ_MdX2Za8eLeFkMiMRW6_Juyep

O jogo que ficou conhecido como *ToME* desvia da norma quando decide abandonar a inspiração tolkiniana e investir em uma base ficcional original. Também leva às últimas consequências o projeto de construir um mundo diverso e amplo, com múltiplas cidades, separadas por oceanos, vegetações e vilas – cada um destes pontos ocupados por personagens, itens e missões. A essa altura, *ToME* já incorpora algo que viria a ser encontrado em muitos outros roguelikes: por mais que haja uma missão “principal”, o jogo oferece tantos outros caminhos que se pode passar centenas de horas sem sequer dar atenção a esse caminho.

- ii. Brogue** (2009) / Designer: Brian Walker / Plataformas: Microsoft Windows e Linux

Gameplay de apoio:

<https://www.youtube.com/playlist?list=PLpY5O7biJ2h7zYEQQTEOcCh6QMYGCRhT8>

Brogue é uma adaptação tardia de rogue, porém investe em um potencial inexplorado pela maioria de seus descendentes. Ao invés de se preocupar em expandir a quantidade de itens, monstros e ambientes, *Brogue* mantém a pouca quantidade destes vetores. No entanto, cria métodos para enriquecer as possíveis conexões que estes podem estabelecer. Nesse sentido, uma sala pode ser ocupada por uma armadilha aparentemente igual a outras tantas, mas ao acioná-la, o golpe pode ser uma rajada de vento, uma labareda de fogo ou um gás venenoso. O mesmo vale para armas, que podem ser abençoadas por diferentes divindades, aumentando seu potencial – divindades que também podem conceder benefícios caso o jogador cumpra certas exigências (não consumir este ou aquele alimento) ou oferecer certos sacrifícios (o cadáver de certo monstro ou um escudo valioso).

- iii. Unexplored** (2017) / Designer: Joris Dormans / Plataforma: Microsoft Windows

Gameplay de apoio:

https://www.youtube.com/playlist?list=PLcsBnp6zp_Jd6eTEP0Z6WSeRtRZyTo4TS

Unexplored aposta na mesma linha seguida por *Brogue*. Embora não seja uma adaptação direta, tomamos a liberdade de incluí-lo como um descendente, devido as inúmeras semelhanças. No entanto, também possui sua cota de inovações. O movimento contínuo (sem turnos e sem casas com separação rígida) é um deles. Outro digno de menção é a configuração das salas: ao invés de expandi-las ou aumentar seu número, *Unexplored* cria conexões imprevistas entre os níveis da caverna. Pode-se descobrir no 5º nível um item que serve de chave para abrir um baú no 3º, por exemplo. Assim, o jogo ganha uma maior conectividade entre os níveis, fugindo da usual progressão incessante em direção às camadas mais profundas do mundo de jogo.

- **Caves of Qud** (2011) / Designers: Brian Bucklew e Jason Grinblat. Plataforma: Microsoft Windows

Gameplay de apoio:

<https://www.youtube.com/playlist?list=PLt8DIPRtpvB48Yt0ivSCLqgmDIneKfutK>

Caves of Qud possui um forte viés ficcional, ao mesmo tempo que desenvolve sistemas instáveis com sua geração procedimental. O jogador começa sua campanha na mesma cidade de Joppa todas às vezes, adquirindo um número de missões iniciais que o mandam para , mais ou menos os mesmos lugares (cavernas, ruínas, vilas, etc.). No entanto, com o avançar do jogo, a repetição dá lugar à variação. O jogador pode descobrir detalhes sobre a vida de “sultões” que governaram o mundo pós-apocalíptico antes de seu decaimento. Na prática, isso significa achar descrições textuais de pinturas ou registros gravados em armas, armaduras e estátuas. Além de explorar ruínas ocupadas por facções que adoram certos sultões, pode procurar caravanas itinerantes, esconderijos lendários de ferreiros e cozinheiros que vendem suas criações, além de rastros do mundo pós-apocalíptico. Grande parte desses itens, personagens e lugares tem suas propriedades geradas de modo automático pelo programa do jogo, até mesmo seus nomes.

- **Cataclysm: Dark Days Ahead** (2013) / Designer: Kevin Grenade / Plataforma: Microsoft Windows

Gameplay de apoio:

https://www.youtube.com/playlist?list=PLgGm2Hh4HVu1jBnwZZ7_QFIRArJcsSVU7

Cataclysm gera um mundo com cidades grandes e pequenas, estradas e desertos, povoados por monstros e humanos disputando o posto de quem oferece mais perigo. Grande parte da missão do jogador é encontrar itens aparentemente inofensivos – como pedaços de plástico ou metal – para confeccionar as armas, ferramentas e demais utensílios (violentos ou não) que possibilitarão a fuga de ataques e a exploração segura do ambiente repleto de ameaças. Se relativizarmos a importância do revestimento ficcional, o jogo lembra jogos de sobrevivência como *UnReal World* ou *Subnautica*: a sua conectividade imprevisível e ampla precisa ser navegada com cuidado pelo jogador, possibilitando a adaptação às instabilidades do mundo de jogo.

- **UnReal World** (1992) / Designer: Sami Maaranen / Plataformas: Microsoft Windows, Linux, OS X

Gameplay de apoio:

https://www.youtube.com/playlist?list=PL_P5ZR8QXRoWva_9jVKYZwkoIwV_xZgZ

O mundo de jogo de *UnReal World* é composto por uma série de forças escondidas: seu ambiente é composto por florestas, montanhas, lagos, oceanos – biomas ocupados por uma farta fauna selvagem e uma flora que coleciona propriedades e capacidades. O jogo se vende, portanto, como um verdadeiro simulador de sobrevivência, no qual o jogador precisa saber os modos adequados de extrair e armazenar a madeira caso queira construir uma cabana; ou se preocupe com quais plantas podem ou não ser consumidas de forma segura. Desta forma, *UnReal World* se assemelha aos jogos de sobrevivência que ganharam notoriedade após o lançamento de *Minecraft* – mesmo que esses jogos possuam apenas uma fração da dificuldade e complexidade que o *roguelike* em questão almeja simular.

- **Dwarf Fortress** (2006) / Designers: Tarn Adams e Zach Adams / Plataforma: Microsoft Windows, Mac OS X e Linux

Gameplay de apoio:

https://www.youtube.com/watch?v=0qKsioMM7U8&list=PL3NGFJJxD5oSvbxQNI9Cw_8B4W41D1Xzj (Fortress mode)

https://www.youtube.com/playlist?list=PLdX0AS_VrFWoTP4yD7fAvkEp0_fLBOi6 (Adventure mode)

Dwarf Fortress possui o objetivo de ser um simulador de mundos fantásticos. O jogo começou como um *roguelike* tradicional: cavernas, cidades e outros ambientes bidimensionais nos quais o jogador explora e conecta objetos para criar novas entidades. Logo, ganhou um “eixo z”: o jogo deixa de ser bidimensional para simular uma grade cartesiana tridimensional: passou a ser possível escalar montanhas, cavar túneis subterrâneos, etc. Foram acrescentadas diversas outras funções, como a simulação de propriedades diferentes para metais, elevação de terrenos afetando caminhos de rios, distribuição de fauna e flora, simulação de agrupamentos sociais que vão desde pequenas vilas até cidades com milhares de habitantes. Guerras, história, civilizações que se erguem e definham – tudo isso manifestado em uma interface incompreensível para leigos. Neste momento (2018), *Dwarf Fortress* está ganhando sua próxima série de atualizações, que implementará um método de geração procedimental de mitos para cada civilização. Os mitos, além de influenciar as personalidades e comportamentos dos personagens, fundarão as bases de funcionamento dos seus sistemas mágicos, que permitirão a criação de rituais e feitiços com diferentes efeitos. Os irmãos Tarn e Zach Adams estimam que o jogo continuará em desenvolvimento por pelo menos mais 15 anos.

- **Ultima Ratio Regum** (2011) / Designer: Mark Johnson / Plataformas: Microsoft Windows

Gameplay de apoio: não encontrado

Mark Johnson, criador de *URR*, começou seu trabalho em 2011 e estima que estará completo em 2022. O mundo de *URR* já conta com uma teia conectiva capaz de gerar continentes, nações e cidades-capitais. Cada cidade possui distritos pobres e ricos, setores militares e zonas portuárias – cada um destes com suas respectivas casas e prédios – mansões de famílias nobres, catedrais de religiões geradas procedimentalmente, quarteis, bancos e monastérios. No entanto, o que diferencia cada uma dessas construções uma das outras é seu interior: barracos em favelas possuem pouquíssimo espaço e, quando muito, uma mesa e poucas cadeiras. Catedrais de religiões estatais ocupam quase metade de um distrito, e são decoradas com candelabros, ídolos

religiosos (gerados procedimentalmente) e dezenas de cadeiras para acomodar fiéis. Por mais que isso tudo seja manifestado de forma simplória pela interface do jogo, é inegável que a composição conectiva possui coerência interna – do elemento mais ínfimo até a formação dos continentes. A próxima etapa do trabalho de Mark é adicionar a instabilidade ao mundo: famílias lutando por poder e influência, exércitos preparando-se para batalhas, dinâmicas econômicas e diplomáticas entre nações, etc.

Pré-roguelikes

- **Dragon Maze** (1977) / Designer: Gary J. Shannon / Plataforma: Apple II

Gameplay de apoio: não encontrado

Gary J. Shannon talvez tenha sido um dos primeiros programadores a empregar a técnica de geração procedimental com o objetivo de provocar variedade nos desafios de um jogo. *Dragon Maze*, obra do programador americano, consiste em poucas regras: o jogador deve achar o caminho que o leva de sua posição inicial até a única saída do labirinto. Enquanto isso, a cada turno que o jogador usa para se mover pelas casas do “tabuleiro”, um dragão também se move – uma unidade de espaço por vez. Cabe ao jogador achar a saída antes que o dragão o ache primeiro. A simplicidade dessa premissa é temperada pelo fato de que, a cada jogo, o labirinto se transforma: o sistema do jogo calcula os parâmetros necessários para que nunca uma mesma topografia de bifurcações se repita. Assim, o labirinto possui seu próprio comportamento, e o jogador precisa entender sua metamorfose para se preparar adequadamente para o desafio.

- **Beneath Apple Manor** (1978) / Designer: Don Worth / Plataforma: Apple II

Gameplay de apoio: <https://www.youtube.com/watch?v=6kM19sw-jZM&t=>

BAM é o roguelike nascido antes de *Rogue*. O anacronismo talvez seja facilmente explicado: seu desenvolvedor apropria-se da ideia de Gary J. Shannon, implementada em *Dragon Maze* e associa-a a um cenário mais variado: mais inimigos, corredores maiores e salas mais distribuídas, itens que podem ser usados para facilitar as missões, etc. *BAM*, portanto, é uma concatenação de influências: traz os cenários estáticos como o de *Colossal Cave Adventure*, adiciona a instabilidade dos labirintos de *Dragon Maze* e reveste o resultado com um mundo de inspiração em *Dungeons & Dragons*.

- **Dungeon** (1979) / Designer: Brian Sawyer / Plataforma: Commodore PET

Gameplay de apoio: não encontrado

Dungeon é outro jogo que se inspira em *Dragon Maze*, embora possuísse uma distribuição de salas e corredores menos sofisticada que a de *BAM*: o posicionamento das salas e dos itens não possui tanta variabilidade, repetindo-se com mais frequência de jogo para jogo. Não obstante, sua conformação mais simples parece coincidir com a premissa de *Rogue*, segundo um de seus criadores (Glenn Wichman): não se tratava de criar um jogo complexo, mas o contrário: uma aventura a la *Dungeons & Dragons* de fácil acesso e com poucos elementos que distraiam o jogador das batalhas e dos monstros.

- **Dungeon Campaign** (1978) / Designer: Robert Clardy / Plataforma: Apple II

Gameplay de apoio: <https://www.youtube.com/watch?v=1xY7bZJ1LOQ>

Dungeon Campaign se parece com uma versão mais colorida e aprimorada de *Dragon Maze*. Robert Clardy admitidamente se inspirou no jogo, mas não deixou de acrescentar uma dose de apreço pelos RPGs da série *Dungeons & Dragons*. O jogador, assim como é de praxe nas aventuras do jogo de mesa, pode percorrer as cavernas com seus amigos – cada um tomando conta de um personagem com seus pontos de vida e habilidades específicas. *Dungeon Campaign* é mais um exemplo de que, embora *Rogue* tenha ganhado a fama de “ancestral comum”, o “espírito” do gênero já estava presente de alguma forma em outros jogos menos lembrados.

Roguelites

- **Abyss Odyssey** (2014) / Designers: Andres, Carlos e Edmundo Bordeu. Plataforma: Microsoft Windows, Playstation 3, Playstation 4 e Xbox 360

Gameplay de apoio: https://www.youtube.com/watch?v=QI6vL0j_wXA

O jogador controla um personagem capaz de desferir golpes de diferentes níveis e alcances, de modo semelhante ao de um jogo de luta ou *beat'em'ups*. A diferença é que a conformação labiríntica do espaço bidimensional do jogo lembra uma caverna de um *dungeon crawler* ou *roguelike*. Assim como nesses gêneros, as cavernas são geradas proceduralmente, e os inimigos que as habitam também possuem seus próprios conjuntos de golpes e poderes que podem disparar contra o jogador.

- **Azure Dreams** (1997) / Designer: Hiroshi Tamawari / Plataformas: Playstation e Game Boy Color

Gameplay de apoio: <https://www.youtube.com/playlist?list=PL9F5ACC0DEBEEDACC>

Azure Dreams é um efeito indireto dos roguelikes no Japão, que se alastrou em grande parte graças ao sucesso da série *Mystery Dungeon*. Além de cavernas com monstros, armadilhas e itens valiosos, o jogador pode navegar com seu personagem por uma cidade, na qual pode investir os resultados de suas expedições, alimentando sua economia e provocando o surgimento de novos estabelecimentos (teatro, hospital, academia de luta, etc.). Dessa forma, o jogo se mostra como uma mistura de roguelike com jogos de gerenciamento como *SimCity* (ainda que com sistemas significativamente mais simples).

- **City of Brass** (2017) / Designer: Ed Orman / plataforma: Microsoft Windows, Playstation 4 e Xbox One

Gameplay de apoio:

<https://www.youtube.com/playlist?list=PLIfRmGRRImvsV1cpwFsOirrwmX-4QconF>

Roguelite em primeira pessoa com um revestimento visual remetente a lendas e clichês do mundo árabe. Os mundos de jogo são gerados proceduralmente, consistindo em salas com armadilhas e inimigos que devem ser abordados separadamente. O jogador pode adquirir e fazer

uso de poções e armas que concedem diferentes vantagens, bem como se utilizar de aspectos do ambiente ou mesmo das armadilhas e reverte-las contra os próprios monstros.

- **Crypt of the Necrodancer** (2013) / Designer: Ryan Clark / Plataformas: iOS, Microsoft Windows, OS X, Linux, PlayStation 4, PlayStation Vita, Xbox One e Nintendo Switch

Gameplay de apoio:

https://www.youtube.com/playlist?list=PL_rQc0PIUa8OexJwNx9QklWIZUcgva4KC

O jogador controla uma personagem que se move uma casa por vez em um mundo de jogo bidimensional e gradeado. A trilha musical, no entanto, dita o ritmo em que o movimento deve ser executado: caso o jogador se mova fora do andamento da música, perderá o bônus que confere maior poder de fogo para enfrentar os monstros que povoam os níveis da caverna. Estes, por sua vez, também dançam conforme a música, desenhando padrões que devem ser absorvidos pelo jogador para que evitar golpes e achar brechas para golpear.

- **Diablo** (1996) / Designers: David Brevik, Erich Schaefer, Max Schaefer, Eric Sexton e Kenneth Williams / Plataformas: Microsoft Windows, OS X, PlayStation

Gameplay de apoio: <https://www.youtube.com/playlist?list=PLA0E7478EAF1C7804>

O jogo desenvolvido pelo que viria a se tornar a gigante Blizzard talvez tenha sido a primeira instância de sucesso dos roguelikes na camada superior da indústria dos videogames. O jogo se aproveita da mecânica de geração procedimental de cavernas e monstros, criando um cenário de estilo semelhante aos RPGs da época – porém investindo mais em batalhas e exploração do que em diálogos e tramas narrativas. *Diablo*, principalmente em suas continuações, viria a exigir do jogador não apenas estratégia e gerenciamento de poções, armas e pergaminhos, mas destreza e ritmo para lidar com os muitos inimigos presentes em cada caverna do jogo.

- **Eldritch** (2013) / Designer: David Pittman / Plataformas: Microsoft Windows, OS X, Linux

Gameplay de apoio:

<https://www.youtube.com/watch?v=mHhgROZQKoY&list=PLhv63wSI9Alpn9S2tXwMdd6BhymYfZ2m6>

Roguelite em primeira pessoa, no qual o jogador controla um único personagem em um mundo de jogo tridimensional. Há ênfase em mecânicas de *stealth* (furtividade), ou seja, em que se deve evitar confrontos abertos e preferir atacar inimigos desprevenidos um a um, usando o elemento-surpresa. Assim, como muitos outros roguelites, Eldritch limita-se a empregar as mecânicas de morte permanente e geração procedimental de conteúdo, no que tange sua herança roguelike. O revestimento ficcional do jogo é baseado na mitologia de H.P. Lovecraft.

- **Mystery Dungeon** (franquia – 1993-2015) / Designer original: Koichi Nakamura / Plataforma original: Super NES

Gameplay de apoio: <https://www.youtube.com/watch?v=qr9jube8F-0>

Uma das reverberações do fenômeno roguelike no Japão, a série *Mystery Dungeon* demonstra a sinergia histórica poucas vezes lembrada entre os jogos de RPG orientais e ocidentais. Assim como as séries *Dragon Quest* e *Final Fantasy*, *Mystery Dungeon* põe o jogador no papel de um único herói que explora cavernas e batalha monstros. A franquia, porém, substitui os arcos narrativos e cenários pré-moldados por ambientes gerados proceduralmente e por um foco maior em criação e conectividade. O jogador deve adquirir itens raros nas cavernas e usá-los para customizar e aprimorar o estoque de sua loja, como em *Torneko no Daibōken: Fushigi no Dungeon*. Eventualmente a série tornaria-se ponto de referência para outras criações, prolongando ainda mais o alcance de Rogue – sobretudo com instaurações como a subfranquia *Pokemon: Mystery Dungeon*.

- **Risk of Rain** (2013) / Designers: Matthew Griffin, Duncan Drummon e Paul Morse / Plataformas: Microsoft Windows, OS X, Linux, PlayStation 4, PlayStation Vita

Gameplay de apoio:

<https://www.youtube.com/playlist?list=PLVmM0UVcquYJtxhqdsIJlyK0fjVmzDnVY>

O jogo se apropria das duas principais mecânicas características dos roguelites (morte permanente e geração procedural de conteúdo) e pede ao jogador que salte de plataforma a plataforma, coletando armas e utensílios para derrotar monstros de diferentes níveis de força. É um jogo em duas dimensões, no qual progressivamente, os níveis ficam mais difíceis, pedindo do jogador destreza e velocidade, bem como planejamento adequado do uso de itens.

- **Rogue Legacy** (2013) / Designer: Teddy Lee / Plataformas: Microsoft Windows, Linux, OS X, PlayStation 3, PlayStation 4, PlayStation Vita, Xbox One

Gameplay de apoio:

https://www.youtube.com/playlist?list=PLx5CbzE2kDZ2Q_zPtTbB0E16BgMTkck11

Um dos roguelites independentes pós-*Minecraft* de maior sucesso, principalmente com sua versão para PC. Inspira-se em *Rogue* e cia. Na geração procedural de masmorras e salas do castelo que serve de arena de batalha e exploração. Além disso, utiliza-se da mecânica de morte permanente com um adendo único: a cada personagem que o jogador “perde” quando morre, o jogo registra uma espécie de árvore genealógica. Cada morte do jogador grava um elemento da tapeçaria familiar, cada qual com seus traços, propriedades físicas, vantagens e desvantagens.

- **Spelunky** (2009) / Designer: Derek Yu / Plataformas: Microsoft Windows, Xbox 360, PlayStation 3, PlayStation 4, PlayStation Vita, Chrome OS

Gameplay de apoio:

<https://www.youtube.com/playlist?list=PLHYAYMSbpcvuQqGRxH1UIzmYAPPQuBpNj>

Spelunky é o roguelite disparador da “ressurreição” do gênero, que coincidiu com a incipiência de *Minecraft*. O jogador comanda um aventureiro munido de apenas um chicote, algumas bombas (capazes de quebrar paredes e ferir inimigos) e cordas (acoplam-se ao teto e permitem que o jogador as escale). As cavernas escondem outros itens, armadilhas e personagens (hostis ou não), que podem interagir uns com os outros de maneira inesperada. Por exemplo, bombas afetam paredes, chão, criaturas, itens e o próprio jogador, que podem ser disparados ao ar e colidir com outras criaturas e itens, incitando novas reações em cadeia, e assim por diante.

- **ToeJam & Earl** (1991) / Designer: Greg Johnson / Plataforma: Sega Mega Drive.

Gameplay de apoio:

https://www.youtube.com/playlist?list=PLq1ck3Vj7Ns5_CZdONuvRKjTicqhmQJSW

Possivelmente o primeiro roguelite a alcançar notoriedade (décadas antes do termo ser inventado), *ToeJam & Earl* muda uma série de parâmetros do gênero. Os inimigos e personagens caminham nos seus próprios ritmos (não há divisão por turnos), o cenário ficcional é muito distante dos roguelikes (uma mistura de ficção científica em estética cartoon e inspirações na cultura do funk e hip-hop americanos). Mas, assim como os roguelites futuros, mantem a dificuldade brutal do gênero e a geração de níveis que se bifurcam em corredores. Uma funcionalidade notável é a adaptação do estilo para dois jogadores – que embora já existisse no mundo roguelike, estava longe de ser um atributo comum.

Survival games (Minecraft + pós-Minecraft)

- **DayZ** (2012) / Designer: Dean “Rocket” Hall / Plataforma: Microsoft Windows

Gameplay de apoio: <https://www.youtube.com/playlist?list=PLbOJKCIrq-bdrLCppSY3Ew7fO1P8HY1v>

DayZ surge como uma modificação (*mod*) de Arma 3, um jogo de guerra com forte carga simulacional e realista. A modificação se utiliza das armas, avatares e objetos como casas e veículos, distribuindo-os por um terreno que imita zonas rurais e urbanas, em um mapa extenso. Servidores na internet recebiam por volta de 100 jogadores simultâneos cada um. Era necessário buscar recursos como armas e mantimentos, de forma semelhante aos jogos de sobrevivência pós-*Minecraft*, bem como escapar de monstros (zumbis) que rondam o mundo do jogo. No entanto, durante a disputa por território e recursos, jogadores tendiam a formar alianças ou, mais comumente, a temerem uns aos outros mais que aos monstros.

- **Don't Starve** (2013) / Designer: Jamie Cheng / Plataformas: Microsoft Windows, OS X, Linux, Playstation 4

Gameplay de apoio:

<https://www.youtube.com/playlist?list=PL8cHO25HeS8i9an5SMmpfJD1MdxN2F7ov>

O título do jogo resume a premissa principal: evitar que o cronômetro regressivo da fome do jogador chegue ao zero, causando sua morte. Os empecilhos que dificultam a missão resumem-se a monstros e ao ritmo incessante que guia o ciclo dia-noite, tornando *Don't Starve* um exemplar ideal das premissas básicas do jogo de sobrevivência pós-*Minecraft*. Somado a esses, o jogo ainda ostenta um último atributo comum a esses jogos: a necessidade de construir novos itens, por meio da conexão e combinação de objetos menos raros. Em outras palavras, o jogador precisa explorar o mundo para coletar itens e combiná-los para usar suas propriedades de forma adaptativa.

- **Factorio** (2013) / Designer: Michael Kovarik / Plataformas: Microsoft Windows, OS X, Linux

Gameplay de apoio:

<https://www.youtube.com/playlist?list=PL-7ICWbZUDjmBkUdFGh1CxxN0gn9oL0aY>

Jogo de sobrevivência com premissas semelhantes às de *Minecraft* ou *Don't Starve*, mas que expande as capacidades conectivas das entidades e objetos do mundo de jogo. Deste modo, *Factorio* torna-se um laboratório de experimentações de construção: trilhos devem guiar recursos aos seus devidos destinos, máquinas devem automatizar a construção e refinamento de materiais e, posteriormente, encaminhá-los para as aplicações devidas. Quanto maior se torna a fábrica montada pelo jogador, mais o jogo deixa de lembrar *Minecraft* e transparece sua verdadeira vocação: simular linhas de montagem, das quais o jogador possui uma perspectiva limitada, mas constantemente ampliada pela automatização de sua própria agência.

- **No Man's Sky** (2016) / Designer: Sean Murray / Plataforma: Microsoft Windows

Gameplay de apoio:

<https://www.youtube.com/playlist?list=PLxVCT8htDB0d8GqsxczSXUnDcdZND8esy>

No Man's Sky aparece como uma promessa comercial sem apoio de grandes produtoras ou distribuidoras. Logo a Sony se apropria do potencial publicitário do jogo, focando a campanha nos aspectos técnicos do jogo: em suma, um espaço jogável que se utiliza de técnicas de geração procedimental para simular uma área equivalente a extensão do nosso universo. A imensidão dessa afirmação escondia o verdadeiro método do jogo: a repetição incessante de padrões dos sistemas solares, planetas com biomas estáticos e pouco variáveis, bem como a suposta fauna “procedimental” que pouquíssimo exibia de novidade seja em sua aparência ou (muito menos) em seu comportamento – mesmo que o jogador viajasse de uma borda à outra da galáxia.

- **Subnautica** (2014) / Designer: Charlie Cleveland / Plataformas: Microsoft Windows, MacOS, Xbox One

Gameplay de apoio:

<https://www.youtube.com/playlist?list=PLKcc0xOPUy2Gn0B8dTSuf3zJPibn69UT4>

Subnautica incorpora cronômetros biológicos ao jogador de maneira semelhante ao padrão de outros jogos de sobrevivência. Embora uma de suas novidades seja a ambientação em um cenário subaquático, o jogo também incrementa certos padrões empregados pelo gênero, como o aspecto e o comportamento variante das criaturas e dos biomas que estas ocupam. Torna-se, assim, um jogo de investigação das propriedades e capacidades destes ecossistemas: o desafio tende a concentrar na decifração das forças adormecidas (ou não) em cada área do mapa, muito mais do que na necessidade de combater criaturas hostis.

- **The Flame in the Flood** (2016) / Designers: Bryn Bennect, Chad LaClair, Damian Isla, Forrest Dowling, Gwen Frey e Scott Sinclair / Plataformas: Microsoft Windows, macOS, Xbox One, PlayStation 4 e Nintendo Switch

Gameplay de apoio:

<https://www.youtube.com/playlist?list=PLIhNUXJ4tecfDEUFv4KBIARf8edGQCbN->

O jogo é uma combinação de roguelite com survival, algo também comum na era pós-*Minecraft*. O jogador controla um único personagem, acompanhado de um cão, devendo explorar o espaço em busca de recursos e proteger-se de criaturas que saem à caça ao cair da noite. A geração procedimental do espaço não é suficiente para esconder a pouca potencialidade conectiva dos itens e demais entidades do mundo de jogo, algo que contrasta com a vasta gama de combinações e conexões de jogos como *Subnautica* ou *Factorio*.

