



sessões do

# MAGNÁRIO

VOL. 22 | N. 38 | 2017 | <http://dx.doi.org/10.15448/1980-3710.2017.2>



CURTA NOSSA  
PÁGINA

**Dossiê  
Game Studies**

Crédito: Sessões do Imaginário - n. 1 - 1996.

**A estética do imaginário  
no cinema**

Demian Garcia e André A. Medeiros

**P. 94**

**Materialidades fílmicas, magia e  
montagem**

Ednei de Genaro

**P. 112**

**Videogames, transgressão e  
criatividade no Brasil**

Emmanoel Ferreira

**P. 72**

# Code breakers e bunny hoppers: transgressão inventiva nas camadas intra e suprajogo

*Code breakers and bunny  
hoppers: inventive transgression in  
the intra and supra game layers*

José Carlos Messias Santos Franco<sup>1</sup> 

Ivan Mussa Tavares Gomes<sup>2</sup> 



**DOSSIÊ**

## Resumo

Ao longo da história dos videogames é possível detectar a atuação de comunidades e jogadores que subvertem as expectativas estruturais dos sistemas para inventar novos modos de jogo. Essa abordagem transgressiva se manifesta em diversas escalas, que possuem suas especificidades, mas que, de certo ponto de vista, apresentam lógicas análogas. Este artigo tem como objetivo evidenciar essas lógicas por meio de três estudos de caso. Na primeira escala, observaremos a lógica transgressiva suprajogo, que será ilustrada pela modificação amadora brasileira de *Guitar Hero III*, o jogo *Guitar Hero Brazucas*. Na segunda, mudaremos para a escala intrajogo, analisando a desconstrução criativa de regras por parte dos *speedrunners* de *Super Mario Bros.* e *Half-Life 2*. Finalmente, utilizaremos a dinâmica do metajogo em *GunZ: The Duel* como demonstração concreta da cognição inventiva demandada por essas transgressões.

## Palavras-chave

Videogames; transgressão; invenção; cultura hacker.

## Abstract

Throughout the history of video-games it is possible to detect the action of communities and players that subvert the structural expectations of the systems to invent new modes of game. This transgressive approach manifests itself in several scales, which have their specificities, but which, from one point of view, present analogous logics. This article aims to highlight these logics by means of three case studies. In the first scale, we will observe the transgressive supraplay logic, which will be illustrated by the Brazilian amateur modification of *Guitar Hero III*, the game *Guitar Hero Brazucas*. In the second, we will move to the intra-gamer scale, analyzing the creative deconstruction of rules by the *Super Mario Bros. speedrunners*. and *Half-Life 2*. Finally, we will use the dynamics of metajogo in *GunZ: The Duel* as a concrete demonstration of the inventive cognition demanded by these transgressions.

## Keywords

Video games; transgression; invention; hacker culture



## Introdução

O BBC Micro, plataforma que pela primeira vez rodou o jogo *Elite* (Acornsoft. Firebird. Imagineer, 1984), talvez seja, atualmente, menos lembrado que o jogo que lhe concedeu notoriedade. A década de 1980 e 1990 foi palco de uma corrida tecnológica concomitante de hardwares e softwares<sup>3</sup>. Porém, por mais que os computadores e elementos tecnológicos que acomodavam jogos como *Elite* fossem fruto do trabalho industrial e serializado de megaempresas, as técnicas de programação e design que geram novas capacidades estéticas pertencem a outra ordem: são fugas e dissidências do *modus operandi* empresarial. A proposta deste artigo é abordar essa subversão como uma mistura técnica-estética própria, que se manifesta de diversas formas na(s) cultura(s) dos videogames.

Vinte e dois *kilobytes* de memória foram necessários para que David Braben e Ian Bell, dois universitários ingleses, programassem *Elite*. O jogo tornou-se o primeiro título a exibir gráficos tridimensionais contínuos, ou seja, seu espaço não apenas dava a impressão de profundidade, como também podia ser navegado em três coordenadas direcionais. Mesmo esse salto astronômico em relação ao que havia sido produzido até então, porém, não esgota as inovações do jogo: a extensão espacial e os processos lúdicos que acomodava parecem inconcebíveis para a época e condições materiais vigentes. 256 planetas orbitavam 32 estrelas distribuídas em 8 galáxias, cada um dos quais mantinha seu próprio esquema comercial e era patrulhado por policiais, piratas e mercenários. Por meio da tecnologia de geração procedimental, o jogo cria um universo grande tanto em extensão quanto em oportunidades.

Por mais que a indústria dos jogos eletrônicos receba frequentemente o adjetivo de “bilionária”, as invenções mais produtivas de sua história foram cunhadas em porções por desenvolvedores independentes e com pouquíssimos recursos. Também em 1984, *Tetris* (Pajitnov, 1984) foi programado por um trabalhador russo entediado com sua rotina: o resultado foi um dos jogos mais influentes de todos os tempos. Ao longo dos anos 1990, quatro jovens americanos – criadores de *Commander Keen* (Apogee, 1990) *Wolfenstein 3D* (Apogee Software. Form Gen, 1992) e *Doom* (Id Software, 1993) – definiram padrões de processamento gráfico que continuariam a ser adotados duas décadas à frente. E, em 2009, *Minecraft* (Mojang, 2011), o hobby de um programador sueco, tornou-se o jogo mais vendido e mais influente da atualidade. Também não é de se admirar que, em todas estas ocasiões, inovações estéticas e lúdicas nasçam da negação da lógica produtivista e serializada instaurada em empresas tradicionais<sup>4</sup>.

As sementes que dão origem a novas técnicas/estéticas nos videogames, nesse sentido, tendem a vir de zonas alternativas às práticas sedimentadas pela grande indústria. O que não quer dizer, porém, que a segunda não se aproprie das primeiras. O caso recente mais emblemático talvez seja o desenvolvimento do jogo *Sonic Mania* (Pagodawest Games. Headcannon, 2017), fruto da conturbada relação entre a produtora Sega e um grupo de desenvolvedores-fãs cujo passatempo era criar versões não-oficiais da franquia mais famosa da empresa. A Sega, após anos dedicando-se à missão de exterminar juridicamente grupos como esse, resolve finalmente contratá-los<sup>5</sup>. As semelhanças entre os parâmetros de design de *Sonic Mania* e os títulos originais da década de 1990 alcançou um sucesso

incomparável aos retumbantes fracassos da Sega em tentar ressuscitar seu mascote ao longo das últimas duas décadas.

Embora os exemplos sejam inúmeros, a transgressão observada na etapa de produção dos jogos não esgota as manifestações deste fenômeno. Além de *game designers*, jogadores também se apropriam de técnicas subversivas para modificar, refazer e/ou ressignificar as práticas de jogo que emergem de sua interação com os videogames. Por exemplo, quando o jogo *Dark Souls* (From Software, 2011) foi lançado para a plataforma PC/Windows, diversos *bugs* e falhas de desempenho transformaram a versão em um produto praticamente inviável de ser jogado. Uma destas falhas – um teto de resolução relativamente baixo de 1024x700 pixels – foi “corrigida” por um fã 23 minutos após o jogo ser posto à venda<sup>6</sup>. Menos de um mês depois, a comunidade de jogadores desenvolveu um *patch* que não só resolvia as graves quedas da taxa de quadros por segundo, como elevava o valor máximo desta mesma taxa de 30 para 60 quadros por segundo<sup>7</sup>.

Por último, ainda temos uma camada ainda mais microscópica da transgressão nos jogos: a subversão por parte do jogador da atuação regular do sistema. Aqui, as ocorrências mais comuns são *glitches* e *bugs*, defeitos operacionais que, normalmente, produzem comportamentos aberrantes no software. A comunidade de *speedrunners*, por exemplo, costuma catalogar *bugs* e *glitches* no intuito de inventar formas mais rápidas de chegar ao final dos jogos. No entanto, é possível dizer que essas subversões não se dão apenas quando o sistema parece ser corrompido, mas apresentam-se até mesmo quando este aparenta estar funcionando segundo as regras: é o caso de dois jogos, *Super Mario*



*Bros.* (Nintendo R&D2, 2003) e *Half-Life 2* (Valve Corporation, 2004) e de certas estratégias de jogo que nasceram da manipulação inventiva de suas regras.

Este artigo, portanto, tem como objetivo abordar estas duas escalas – supra e intrajogo – sob um ponto de vista cognitivo. Em suma, procuramos destacar que, em todas elas, o que observamos é um comportamento inventivo e subversivo que não só está presente em toda história dos videogames, mas que, na nossa hipótese, é componente essencial da experiência lúdica. Primeiro, descreveremos a escala suprajogo na modificação brasileira de *Guitar Hero III* (Neversoft, 2007), o jogo *Guitar Hero Brazucas* (Independente, s/a). Depois, apresentaremos a desconstrução criativa das regras em processos de *speedrunning* em *Super Mario Bros.* e *Half-Life 2*. Finalmente, investigaremos estes fenômenos sob a concepção cognitiva de invenção, a partir da dinâmica do metagame em *GunZ: The Duel* (Maiet Entertainment, 2003).

### Transgressão suprajogo: modding, patching e gambiarra

Há uma fortuita confluência, pelo menos na língua inglesa, entre as denominações de *hacking* (hackear), *gaming/playing* (jogar), *configuring/modding* e até *cheating* (trapacear) (Consalvo, 2007; Zimmerman, 2004), estando todos de alguma forma conectados historicamente. À primeira vista, poderíamos supor que estes modos de subversão – que aqui estamos chamando de subversões “suprajogo” são uma ocorrência recente na história dos videogames. No entanto, sob uma certa perspectiva, ela inaugura sua influência nos primeiros momentos do nascimento da indústria:

*Spacewar*, tal qual foi desenvolvido por Steve Ruszel em 1962, foi o produto de um dado território, o MIT, desenvolvido por estudantes de engenharia que possuíam certas habilidades lúdicas requeridas para se apropriar dele, compartilhavam certos pontos de referência ficcionais (como a fascinação por ficção científica) e alguns dos valores da atividade lúdica. Isso logicamente combinava com a prática de hackear que entre outras coisas defendia a circulação livre de informação e a abertura para que objetos fossem modificações por seus usuários<sup>8</sup> (Genvo, 2014, p. 128, grifos no original, tradução nossa).

Essa influência estende-se também à ideia de gambiarra, que além de já possuir uma “aura” clandestina, também é associada às culturas *hacker* e *gamer*, e que, por esse motivo, serve de ponto de confluência das diversas práticas de intervenção nos jogos. Principalmente pelo *ethos* lúdico apontando por Sebastián Genvo, o desenvolvimento de jogos já na sua incipiência nutriu-se da mesma potencialidade técnica e estética que associamos ao termo “gambiarra”. Embora a palavra tenha suas raízes na língua portuguesa e no Brasil, ela não precisa ser uma prática relegada aos países emergentes ou subdesenvolvidos.

Consalvo (2007), Gee e Hayes (2010) e Newman (2008) fazem menções a práticas igualmente precárias, transgressoras e até clandestinas/ilegais, que fazem ou pelo menos fizeram parte da cultura *gamer*, inclusive nos EUA ou na Europa. Em seu livro sobre trapaça (*cheating*), Consalvo (2007, p. 65-73) conta que o desbloqueio de consoles (Playstation 2, Xbox e GameCu-

be) para que eles executassem mídias que não as estabelecidas pelo fabricante era uma prática comum. Ela lista CDs de instalação e/ou decodificação (GameShark, GameGenies, Code Breaker, Free Loader seriam exemplos), placas de modificação (*mod chips*) e até a prática de soldar componentes dentro do hardware do console, como forma de alterar suas especificações. Algumas dessas intervenções eram feitas através do software, outras requeriam ações no hardware<sup>9</sup>. O amplo espectro dessas intervenções vai desde a alteração da região onde os jogos podem ser rodados até regras dos jogos, como número de vidas e a possibilidade de pausar os que normalmente não possuem essa opção.

Na franquia *Guitar Hero* (Harmonix, 2005)<sup>10</sup>, as modificações são chamadas de *hacks* por seus criadores, isso porque elas empregam o uso de software que “invade” o jogo original, para que os usuários possam alterar suas funcionalidades, aspecto e conteúdo. Na rede social Orkut, comunidades dedicadas a estas experimentações podiam ser encontradas desde 2007, como é o caso de *Guitar Hero Hacks* e *Guitar Hero Creators*. Devido à complexidade técnica dos chamados consoles da sétima geração (Playstation 3, Xbox 360, Nintendo Wii), a maior parte das iniciativas encontradas contempla os games para Playstation 2 e Windows.

Criar seu próprio *hack* envolve uma série de etapas, todas minuciosamente explicadas através de incontáveis tutoriais escritos nas redes sociais e até videoaulas no Youtube. Por meio da captura de tela, esses videotutoriais conseguem mostrar passo a passo como realizar uma modificação, o que, acreditamos, ajudou a difundir a prática dos *hacks* de *Guitar Hero* para além de usuários com qualquer familiaridade com customização.



**Figuras 1 e 2:** *Guitar Hero Brazucas*, mod de *Guitar Hero III: Legends of Rock*. À esquerda, é possível ver um exemplo de modificação visual do jogo, enquanto à esquerda vemos a intervenção direta nas regras: os jogadores adicionaram músicas, por exemplo, da banda Mamonas Assassinas, alterando as notas (botões coloridos) e o ritmo em que devem ser executadas. **Fonte:** Maroto (2015).

Em um tutorial, compartilhado pela comunidade desde 2009, os usuários que quiserem fazer seu próprio *hack* são instruídos a, em primeiro lugar, passar os arquivos do DVD original do jogo para o computador (embora também seja possível baixá-lo da Internet). Em seguida, usando um programa chamado GH Explorer (GHex), é possível navegar pelo conteúdo e fazer substituições das músicas originais por outras a sua escolha (cada jogo da franquia possui um limite de músicas, geralmente 70 no caso dos *hacks* feitos em cima de *Guitar Hero II*). Para criar seu próprio *setlist*, o jogador pode selecionar de músicas pré-traduzidas para o jogo, por meio de manipulação de arquivos .midi e .chart. Muitos usuários, porém, optam por criar suas próprias composições. Para isso são necessários a música em MP3 que se quer inserir e outro programa experimental chamado Feedback, que justamente cria as notas musicais do jogo. Essa sincronização é feita manualmente pelo usuário. Ele precisa ouvir a música e escolher onde ficará a nota e a qual botão ela corresponderá. Além disso, ele pode escolher onde ficam os eventos bônus da música (*Star Power*) – notas em forma de estrela, que ao serem executadas corretamente dobram a pontuação –, e animações do avatar (guitarrista) e do palco durante a performance.

Esse e outros tutoriais foram criados pelo usuário Dark Shadow, uma figura central na comunidade *Creators* e nos *hacks* encontrados, pois seu apelido (*nickname*) aparece nos créditos de muitos deles. Ele atualizava constantemente esses tutoriais, separando-os por cada edição do jogo. Seus tutoriais são ilustrados, além de possuírem separação por capítulo (sem paginação), links para download e informação de contato. O tutorial de *Guitar Hero II* (Harmonix, 2006) tem 21 páginas,



já o de *Guitar Hero III*, em 2010, já em seu quinto volume, possui 171 páginas<sup>11</sup>.

Mia Consalvo (2007) e especialmente Newman (2008) dedicam um espaço considerável à discussão de atividades extrajogo ou, neste caso, jogabilidades suprajogo (*superplay*), tais quais as descritas acima. Para os autores, dependendo da modalidade e da exigência material e cognitiva, a ideia de intervenções suprajogo abarca desde jogabilidades extremas (nível de dificuldade), domínio total de todas as possibilidades do jogo (pontuação, passagens secretas, *easter eggs*, bônus), cenários peculiares (condições específicas na qual vencer o jogo, objetivos autoimpostos), façanhas ou acrobacias radicais etc; mas passam também por trapaças e subversões mais intensas. Formas de brincar, jogar com a própria estrutura do jogo de maneira independente de seu progresso regular ou missões principais. Enfim, reconfigurar a conexão as próprias regras do videogame enquanto ele está sendo jogado.

Para Newman (2008), o suprajogo revela algo sobre a natureza dos videogames enquanto algoritmos que os diferenciam de jogos condicionados apenas pelas regras e não por operações binárias. O que está em jogo é a capacidade de superar ou não as limitações do software como suporte:

[...] está claro que aquilo que é costumeiramente chamado de “jogabilidade emergente” (ver Smith, 2001; Salen e Zimmerman, 2004) está no centro da prática de speedrunning [terminar o jogo o mais rápido possível], mesmo que boa parte desta emergência esteja centrada na exploração de falhas [glitches] ao invés do uso de um conjunto de

ferramentas ou recursos abertos. Assim, é difícil definir exatamente onde e se a emergência foi pretendida pelo desenvolvedor ou é uma consequência de seus erros de programação e anomalias inevitáveis e limitações da sua modelagem e design<sup>12</sup> (Newman, 2008, p. 135, tradução nossa).

A visão de Newman (2008) parece ser perturbada por uma definição de emergência – e, por extensão, de *glitch*/falha. Sobretudo na primeira referência citada, o texto de Harvey Smith, de 2001, as propriedades emergentes do sistema são aquelas que não foram previstas originalmente por uma figura soberana: o criador do jogo. Enquanto isso, a segunda referência, Salen e Zimmerman, definem emergência como a complexidade que se origina de poucas regras simples (2004, p. 152). A primeira, além de apelar à falácia da intencionalidade (Juul, 2002), contradiz a segunda no sentido de que um *game designer* pode deixar de perceber um comportamento do sistema que não seja efeito da complexidade da atualização das regras.

Certa percepção dessas condições materiais e a contingência fazem parte dessas propriedades configurativas e transgressoras dos jogos, sendo um forte argumento contra as hipóteses da imersão e de uma interpretação simplista da ideia de círculo mágico. Afinal, como enfatiza Moulthrop:

Ao concebermos a configuração como uma forma de envolver não apenas os elementos imediatos, mas também as condições sociais e materiais do videogame – e por extensão, as condições de outros sistemas regrados como trabalho e cidadania – pode ser muito importante insistir na diferença

entre jogar e interpretar, a melhor para resistir à imersão. Qualquer análogo do letramento para mídias interativas precisaria incluir tal resistência<sup>13</sup> (Moulthrop, 2004, p. 66, tradução nossa).

Além disso, num nível menos elementar do que o das operações lógicas dos *bits*, os videogames possuem outras estruturas suprajogo, que também admitem intervenções. A principal delas seria o motor do jogo (*game engine*) que “consiste em diversos componentes como o renderizador [*renderer*] que visualiza o espaço do jogo, um motor de física, código de rede [*networking code*], programa da inteligência artificial, sistema de som e outras partes”<sup>14</sup> (Nieborg, Graaf, 2008, p. 182, tradução nossa).

Há ainda outros programas como editores dentro do jogo, kits de desenvolvimento de software (*Software Development Kits – SDK*), programa para modelagem 3D e outros tipos de editores gráficos, além de bibliotecas diversas – um conjunto de programas escritos para uma determinada função (um princípio físico específico ou uma característica visual como a reflexão luminosa) que influenciam nas práticas de customização. Sem acesso a muitos desses softwares de ponta, muitas vezes os jogadores-hackers encontram soluções caseiras como programas de “menor porte” como Photoshop/Paint, editores de áudio gratuito ou livres, e mesmo versões pirateadas dos mesmos.

Daí surge o problema levantado por Newman (2008) sobre a interação entre esses diversos programas e rotinas e como eles interagem (não sem falhas/*glitches*) na construção de um game. Por isso, e para evitar que o game simplesmente não seja executado



(não funcione), as *engines* têm uma função centralizadora. Segundo Nieborg e Graaf:

Engines são projetadas propositalmente modulares para que elas possibilitem o melhoramento/aprimoramento de partes específicas do motor sem ‘quebrar o código’, ou de plug-ins [inserções] de terceiros (‘sub-engines’) que descarregam complexas rotinas de software. Essas subpartes são também conhecidas como ‘bibliotecas’ e estão sendo constantemente trabalhadas pelos desenvolvedores<sup>15</sup> (Nieborg; Graaf, 2008, p. 183, tradução nossa).

Por isso, diz-se que a criação de *mods*, contando com aplicações em interface gráfica, também contribui para o fenômeno denominado de letramento procedimental por Mateas (2008). Mesmo sem acesso a todos os componentes ou sem poder vislumbrar o todo a partir das partes disponíveis e acessíveis, essa interação dentro da arquitetura lógica dos games se faz visível. E, assim, as interconexões e instruções algorítmicas podem ser, senão compreendidas formalmente, ao menos intuídas na sua função dinâmica.

### Transgressão intrajogo: speedrunning, bunny hopping e exploiting

A prática denominada de *speedrunning* – tentativa de terminar modalidades de jogos no menor tempo possível – talvez seja a que mais alimente o surgimento de estratégias subversivas internas ao processo de jogo, que aqui chamaremos de “inrajogo”. No fundo, por mais que ocorram em uma escala mais restrita, elas

possuem as mesmas propriedades das atividades de *modding* e reconfigurações suprajogo em geral: são formas de torcer o sistema até que este trabalhe contra seu direcionamento original, resultando em experiências qualitativamente distintas.

Apenas no que tange essa prática, podemos discernir pelo menos duas etapas onde a transgressão se desenvolve: primeiro, no momento em que o jogador atribui um objetivo auto-imposto à experiência (Parker, 2008). Aqui, a tarefa de completar o jogo da forma mais rápida possível é uma subversão – ou pelo menos um redirecionamento – do(s) objetivo(s) do jogo. Para auxiliá-lo na tarefa de realizar este novo objetivo, o *speedrunner* adentra a segunda e mais intensa camada de subversão: a de dissecar o sistema de modo a descobrir técnicas que façam com que o jogador minimize 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 se concentram na primeira competência, enquanto outros apresentam mais afinidade com a segunda. Alguns raros jogadores ainda conseguem se sobressair em ambas.

Talvez o exemplo mais clássico na comunidade de *speedrunners* seja a progressão de quebras de recordes

do jogo *Super Mario Bros.* (Nintendo, 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 desconstrução<sup>16</sup>.

Andrew descobriu que, na fase 4-2 do jogo, podia causar um *glitch* que provoca o sistema a não carregar uma das *cutscenes* da fase: uma cena do personagem Mario emergindo de um buraco, agarrado em uma 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 exibe naquele momento, restando apenas passar para a cena seguinte: o jogador aparece na mesma etapa da figura acima, mas não precisa esperar o fim da *cutscene* para retornar ao jogo. Aproveitando-se deste *glitch*, Andrew G. conseguiu diminuir o recorde de 5:05 para 5:00.

Na incipiência da prática de *speedrunning*, explorações de defeitos no sistema tais quais a descrita acima eram consideradas “ilegais”, e jogadores que se aproveitavam delas não tinham seus recordes contabilizados pela organização amadora Twin Galaxies. Logo, porém, a comunidade espontaneamente começou



Figuras 3 e 4: Cenas de Super Mario Bros. que pode ser pulada, caso o jogador provoque um glitch no jogo.

Fonte: Andrewg1990 (2011).

a aceitar esse tipo de desvio, transformando a prática em uma mistura do já presente refinamento com níveis cada vez mais elevados de desconstrução. Os jogadores *speedrunners*, hoje, são provavelmente os melhores conhecedores do funcionamento do sistema do jogo do seu ponto de vista inverso. Ao invés de projetar o sistema para que exiba gráficos, sons e padrões de controle, exploram essa camada visuaudiomotora (Pereira, 2008) para interferir no sistema que ela esconde.

Deste ponto de vista, a desconstrução operada pelos *speedrunners* é mais do que se aproveitar de defeitos no sistema: é uma forma de ludibriar os programadores do software, encontrando nele propriedades que nem mesmo seus criadores conheciam. Essas propriedades podem se apresentar como defeitos mas, muitas vezes, são apenas consequências imprevistas da lógica própria do jogo.

Por exemplo, o jogo *Quake* (Id Software, 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 “emer-



gência” (Juul, 2002). A mesma *Quake Engine* foi utilizada na produção do jogo *Half-Life* (Valve, 1998), o que quer dizer que a técnica do *bunny hopping* também migrou para este jogo (já que as leis físicas eram produzidas pelo mesmo “motor”). Neste momento, começa uma guerra fria entre a Valve (desenvolvedora de *Half Life*) e a comunidade de *speedrunners*. Enquanto os jogadores queriam se utilizar da habilidade, a empresa tentava impedir que a explorassem injustamente – a 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* (Valve Corporation, 2004), dessa vez com a sua própria *engine*, a *Source Engine*. Já que esta possuía leis físicas semelhantes às da *Quake Engine*, os *speedrunners* logo aproveitaram-se do *bunny hopping* para aumentar seus recordes. Até que a Valve interfere novamente. Os programadores criaram um sistema para tentar coibir o *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 as leis físicas do mundo de *Half-Life 2* para impedir que os jogadores alcançassem velocidades que julgavam 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 o *bunny hopping*. Pela lógica, o sistema adicionava um vetor inverso 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 rever-

sa. A Valve não tentou mais corrigir o erro.

O interessante, se compararmos a estratégia de Andrew G. com a do *bunny hopping*, é que, enquanto uma aparenta ser uma intervenção no funcionamento do sistema, a segunda aproveita-se do funcionamento 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”<sup>17</sup> (Consalvo, 2007, p. 114, tradução nossa). 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.

O ponto é que subverter o funcionamento intrajogo é um processo que independe do acontecimento ou não de falhas no sistema. E, mais importante, o que consideramos ou não uma falha é muito mais fruto do julgamento pragmático humano do que um aspecto inerente dos computadores. Se as regras do computador (ou do jogo) controlam processos e procedimentos que dão estabilidade ao jogo, o que jogadores como *speedrunners* buscam são usos improváveis e imprevistos de regras. O que procuram fazer é gerar instabilidade nessas formações e redirecioná-las para construir outras formas de agir, novas estabilidades.

No caso de *Super Mario Bros.*, Andrew G. se utilizou de um parâmetro computacional que ordena as sequências de telas que serão ou não acessadas pelo jogador, deslocando seu funcionamento para sua própria intenção (pular uma etapa do jogo para zerá-lo mais rápido). Este parâmetro funcionou exatamente como deveria: percorreu seus códigos e diretrizes e realizou

uma adaptação condizente. O estímulo instável – e não necessariamente uma falha – veio do jogador que deu um uso a estes parâmetros que não havia sido achado pelo seu criador. No *bunny hopping*, jogadores de *Quake* e *Half-Life 2* descobriram funções dos vetores de velocidade que controlavam seu movimento e usaram sua potência para criar novas formas de se locomover no espaço do jogo.

Não há, portanto, mal funcionamento, e sim um desvio consciente por parte dos jogadores de uma função até então estável do sistema. Ao contrário do que os termos *glitch*, *bug* e *exploit* possam fazer parecer, sua atuação nada mais é do que um sintoma do próprio funcionamento computacional, e, mesmo quando impedem totalmente o processo de jogo (por exemplo, quando um software faz o computador “travar”), talvez escondam uma “rachadura que pode dar acesso a erros melhores”<sup>18</sup> (Krapp, 2011, p. 91). De certa forma, o exercício criativo de programadores de jogos consiste em achar modos de instruir o computador a criar padrões visuais, auditivos e cinéticos que abriguem a experiência lúdica. Isso por si só, já consiste em um desvio inventivo da funcionalidade da computação: não é surpreendente, deste ponto de vista, que essa subversão continue acontecendo quando os videogames passam a pertencer aos jogadores.

## Invenção vs. Repetição: a dimensão subversiva das práticas lúdicas

Em todas essas escalas, a prática lúdica pode ser pensada sob um prisma cognitivo: seja para piratear ou para explorar falhas computacionais, é necessário que busquemos por fissuras naquilo que aparentemente é absolutamente sólido. Virgínia Kastrup (2007) mostrará



que esse tipo de atitude – chamada por ela de invenção<sup>19</sup> – é mais do que uma função cognitiva, tal qual reconhecer padrões ou elaborar modelos abstratos. A invenção, na verdade, é tomada como a dimensão temporal da cognição, aquilo que possibilita que fuja exatamente do reconhecimento e de modelos mentais estáticos:

Tal exercício é divergente em dois sentidos: como os processos são dissociados entre si, que não en-

tram em concordância, e como funcionamento divergente em relação ao funcionamento cognitivo, que depende da concórdia das faculdades. Identificam-se então duas tendências ou, na linguagem de Deleuze, dois níveis de funcionamento cognitivo: um nível repetitivo ou recognitivo e um nível inventivo (Kastrup, 2007, p. 73).

Um processo comum nos jogos *multiplayer*, o *metagaming* (Carter et al., 2015), demonstra de forma clara

o enfrentamento entre reconhecimento e invenção nas práticas lúdicas. Um determinado campo de atuação em um jogo se estabelece: formam-se times, partidas, competições, campeonatos, etc. Estratégias e formas de jogar começam a se estabelecer, gerando uma camada extra de jogo: não basta apenas saber executar as ações durante a partida, mas entender o padrão geral de estratégias na comunidade como um todo: quais são as mais eficientes nesta ou naquela situação?

Conhecer o *metagame*, portanto, implica em saber quais estratégias se *repetem* em quais situações e em construir uma espécie de modelo ou algoritmo de quais delas utilizar em quais situações. No entanto, o que costuma ocorrer é o surgimento de estratégias divergentes dos padrões sedimentados, mas que acabam mostrando-se mais eficientes. Um exemplo especialmente interessante se deu no jogo *GunZ: The Duel* (Maiet Entertainment, 2003). O videogame consistia em arenas de batalha onde vários jogadores brigavam para ser o último competidor a sobreviver. Diversas armas eram utilizáveis, mas, inicialmente, a maioria usava armas de fogo, que gozavam de maior poder de alcance e de dano.

Em dado momento, alguns jogadores perceberam que as espadas (que só atingiam a curto alcance e com menos eficiência) permitiam que seus avatares pulassem e permanecessem por breves momentos no ar. Ao desferir novos golpes, podiam movimentar-se continuamente, “voando” por vários segundos consecutivos. Essa estratégia de jogo, apelidada de *K-Style*, logo alastrou-se até dominar o metagame do jogo, transformando o jogo de tiro em um mar de espadachins voadores<sup>20</sup>.



Figura 5: Três jogadores se enfrentam em uma das arenas de batalha em *GunZ: The Duel*.  
Fonte: Jeong (2010).



Para chegar a este nível, a comunidade precisou romper com a repetição de estratégias e fazer emergir algo novo.

É possível, em exemplos como esse, visualizar os modos possíveis de subversão/invenção em ambientes midiáticos digitais. Sua própria fruição demanda uma ação ou performance, um esforço não trivial (Arrseth, 1997) vindo do jogador e, em contrapartida, seu modo de funcionamento, como afirma Alexander Galloway (2006), possui também autonomia vinda do componente computacional. A subversão, neste contexto, seria resultado da ação inventiva que se aproveita e redireciona as forças ativas máquicas por meio da adaptação cognitiva, corporificada, da performance do jogador.

Esta tomada do corpo não envolve apenas os sentidos, mas a personalidade, gosto, afetos e outros tipos de laços, que vão formar esse vínculo (*attachment* – Hennion, 2010) sociotécnico com os videogames e possíveis outras mídias mobilizadas no processo. A construção desse vínculo está ancorada nas propriedades materiais dos jogos (agora num sentido amplo não só os digitais), e por sua vez essa materialidade engloba um modo de existência ou de funcionamento ou de comportamento que “escorre” para fora dos limites espaços-temporais do jogo, o que instaura a configuração como potente vetor da experiência sociotécnica.

A complexidade material dos videogames desenha os contornos das possibilidades de intervenção e invenção de uma maneira que não necessariamente ocorre em outras expressões midiáticas. A despeito disso, existem concepções do processo de jogo que resultam em experiências engessadas, ilustradas principalmente por

apropriações das dinâmicas de jogo tais quais a gamificação. Em oposição a esta, Sebastián Genvo propõe a ideia de ludicização (*ludicisation*), que, assim como a primeira, tentaria enxergar dinâmicas lúdicas em processos extrajogo. No entanto, essas se dariam num nível estrutural e/ou formal que, ao contrário da primeira, incluiria o movimento e imprevisibilidade dos processos de jogo:

[...] chamar os processos pelos quais situações são transformadas em jogos de processos de ludicização, deste modo escolhendo fazer uma distinção entre essa ideia e outras noções como a de gamificação que não considera os aspectos transitório e processual de fenômenos lúdicos, preferindo em vez de isso atribuir características fixas a esses fenômenos<sup>21</sup> (Genvo, 2014, p. 120, tradução nossa).

Para o autor, essa ludicização não se expressa da mesma maneira num mesmo contexto ou numa mesma época. Isso porque ele acredita que o jogo (na sua dimensão transgressiva) opera um agenciamento de elementos que atravessa maneiras de jogar de um território para outro. E é assim que a contingência – o grau de indeterminação do resultado de um jogo (Malaby, 2007) –, pode ser atingida, uma noção que ele credita ao também francês Jacques Henriot. Nesse sentido, o território para qual essas maneiras de jogar, ou propriedades lúdicas seriam passadas, significa o condicionamento de modos de existência outros do que os imediatamente associados pelos parâmetros regulares do jogo. A dimensão cognitiva dos atos de subversão e transgressão nos jogos – seja na sua camada supra ou

inrajogo – pode ser vista como uma interrelação entre a invenção (processo de abalo das faculdades de reconhecimento) e ludicização (emergência de dinâmicas lúdicas em processos anteriormente “engessados”).

## Considerações finais

Este artigo procurou apontar a dinâmica transgressiva que opera em duas escalas da experiência lúdica: primeiro, em processos de apropriação suprajogo, como a criação de *mods* e *patches*, e aquelas internas ao jogo, como as *speedruns*. O objetivo é perceber que, apesar das diferentes condições materiais em que cada um destes (e outros) processos se inserem, seu modo de funcionamento e o tipo de prática cognitiva que põem em jogo são muito semelhantes. Por fim, propomos uma breve investigação dessa constatação sob o prisma dos conceitos de invenção – tal qual elucidado por Virginia Kastrup – e de ludicização, pensado por Sebastián Genvo.

Embora seja possível mergulhar em uma destas escalas e encontrar nuances e aspectos próprios a elas, acreditamos ser necessário, também, compará-las e perceber o que possuem de similar. No limite, essa comparação ajuda a notar de forma concreta que, no âmbito do jogo, encontramos o enfrentamento de duas forças: uma de enrijecimento de processos lógicos, racionais e repetitivos e outra de ruptura com essa solidificação, por meio de inovações, brechas entre as regras e desvios de função. Enquanto a primeira é essencial pela própria essência dos jogos como sistemas baseados em regras, a segunda é o que permite que jogos abriguem o inesperado e que ganhem vida nas mãos criativas dos jogadores.



## Referências

AARSETH, Espen J. **Cybertext: perspectives on ergodic literature**. Baltimore: John Hopkins University Press, 1997.

ANDREWG1990. **Super Mario Bros. Speed Run – 4:58.89**. 2011. (5min33secs). Disponível em: <<https://www.youtube.com/watch?v=kcmKPmj9yeE&t=>>>. Acesso em 25 jan. 2018.

BUNNYHOP, Super. **Taxman and Stealth: The Story behind Sonic Mania's Fans Turned Developers**. 2016. (09min15secs). Disponível em: <<https://www.youtube.com/watch?v=omKGDC4JaMU>>. Acesso em 25 jan. 2018.

CARTER, Marcus *et al.* The Demarcation Problem in Multiplayer Games: Boundary-Work in EVE Online's eSport. In: **Game Studies**, v. 15, n. 1, jul. 2015.

CONSALVO, Mia. **Cheating: Gaining Advantage in Videogames**. Cambridge: The MIT Press, 2007.

DOCSLIDE. **Super Tutorial Customizando Guitar Hero III VOL. 5.0**. 2015. Disponível em: <<http://docslide.com.br/documents/super-tutorial-customizando-guitar-hero-iii-vol-50.html>>. Acesso em: 25 jan. 2018.

GALLOWAY, Alexander R. **Gaming: Essays on Algorithmic Culture**. Minneapolis: University of Minnesota Press, 2006.

\_\_\_\_\_. **The Interface Effect**. Cambridge, UK: Polity Press, 2012.

GEE, James Paul; HAYES, Elisabeth R. **Women and Gaming: The Sims and 21st Century Learning**. New York: Palgrave Macmillan, 2010.

GENVO, Sébastien. Looking at the history of video games through the prism of ludicisation processes. In: **Kinephanos**, History of Games International Conference Proceedings, jan. 2014.

HENNION, Antoine. Loving Music: from a Sociology of Mediation to a Pragmatics of Taste. In: **Comunicar**, n. 34, v. XVII, 2010.

JEONG, Jaekang. **[K-Gunz] Pro Players Movie**. 2010. (5min12secs). Disponível em: <<https://www.youtube.com/watch?v=ppSU5xeEMdU>>. Acesso em 25 jan. 2018.

JUUL, Jesper. **The Open and the Closed: Games of Emergence and Games of Progression**. In: Computer Games and Digital Cultures Conference Proceedings. Tampere University Press, June 2002.

KASTRUP, Virgínia. **A invenção de si e do mundo**. Uma introdução do tempo e do coletivo no estudo da cognição. Belo Horizonte: Autêntica, 2007.

KITTLER, Friedrich. **There is no software**. 1995. Disponível em: <<http://www.ctheory.net/articles.aspx?id=74>> Acesso em: 25 jan. 2018.

KRAPP, Peter. **Noise Channels: Glitch and Error in Digital Culture**. University of Minnesota Press 2011.

MALABY, Thomas M. **Beyond Play: A New Approach to Games**. In: Games and Culture, v. 2, n. 2, abr. 2007.

MAROTO, Garoto. **Mamonas Assassinas - Débil Metal - Expert 98%**. 2015. (3min40seg). Disponível em: <<https://www.youtube.com/watch?v=4IW6qg2hXWw>>. Acesso: em 25 jan. 2018.

MATEAS, Michael. Procedural Literacy: Educating the New Media Practitioner. In: DAVIDSON, Drew (Org). **Beyond Fun: Serious Games and Media**. Halifax: ETC Press, 2008

MOULTHROUP, S. From work to play: molecular culture in the time of deadly games. In: WARDRIP-FRUIIN, N.;

HARRIGAN, P. (Org.). **First Person: new media as story, performance and game**. Cambridge, Massachussets: MIT Press, 2004.

NELSON, Mike. Modder Purportedly Fixes Dark Souls PC Resolution Limitations in 23 Minutes [Update] Tested! **Gamespy**, ago. 2012. Disponível em: <<http://pc.gamespy.com/pc/dark-souls/1225853p1.html>>. Acesso em: 25 jan. 2018.

NEWMAN, James. **Playing with Videogames**. New York: Routledge, 2008.

NIEBORG, David B.; GRAAF, Shenja van der. The mod industries? The industrial logic of non-market game production. In: **European Journal of Cultural Studies**, v. 11, n. 2, 2008.

PARKER, Felan. The Significance of Jeep Tag: On Player-



Imposed Rules in VideoGames. In: **Loading....** v. 2, n. 3, 2008. Disponível em: <<http://journals.sfu.ca/loading/index.php/loading/article/view/44>>. Acesso em: 25 jan. 18.

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. In: **Anais do XVII Encontro da Compós**. São Paulo: UNIP, 2008.

SALEN, Katie e ZIMMERMAN, Eric. Rules of Play. Cambridge, MA: MIT Press, 2004.

SALT, Sumoning. **World Record Progression: Super Mario Bros.** 2017. (23min38secs). Disponível em: <<https://www.youtube.com/watch?v=RdAkY7RfajY>>. Acesso em: 25 jan. 2018.

SCULLY-BLAKER, Rainforest. A Practiced Practice: Speedrunning Through Space With de Certeau and Virilio. In: **Game Studies**: v. 14, n. 1, ago. 2014.

SMITH, Harvey. **The Future of Game Design: Moving Beyond Deus Ex and Other Dated Paradigms.** 2001. Disponível em: <<http://www.witchboy.net/articles/the-future-of-game-design-moving-beyond-deus-ex-and-other-dated-paradigms/>> Acesso em: 25 jan. 2018.

STRATEGYWIKI. **GunZ The Duel/Korean-Style and Advanced Techniques.** S/A. Disponível em: <[https://strategywiki.org/wiki/GunZ\\_The\\_Duel/Korean-Style\\_and\\_Advanced\\_Techniques](https://strategywiki.org/wiki/GunZ_The_Duel/Korean-Style_and_Advanced_Techniques)>. Acesso em: 25 jan. 18.

WALTER, Maximilian. Dark Souls: Prepare to Die Edition - Fans bringen das Spiel auf 60FPS. **GameStar**, 2012. Disponível em: <[http://www.gamestar.de/artikel/dark\\_](http://www.gamestar.de/artikel/dark_)

[souls\\_prepare\\_to\\_die\\_edition,3005061.html](http://www.gamestar.de/artikel/dark_souls_prepare_to_die_edition,3005061.html)>. Acesso em: 25 jan. 2018.

ZIMMERMAN, Eric. Narrative, Interactivity, Play and Games: Four Naughty Concepts in Need of Discipline. In: **First Person: new media as story, performance and game.** Cambridge, Massachussets: MIT Press, 2004.

### Referências Audiovisuais

ACORNSOFT. FIREBIRD. IMAGINEER. **Elite.** 1984. Formato Digital. BBC Micro, Acorn Electron, Apple II, Amstrad CPC, Commodore 64, ZX Spectrum, MSX, Tatung Einstein, IBM PC compatible, Acorn Archimedes, Amiga, Atari ST, Nintendo Entertainment System

APOGEE SOFTWARE. **Commander Keen.** 1990. Formato Digital. MS-DOS, Windows.

APOGEE SOFTWARE. FORM GEN. **Wolfenstein 3D.** 1992. Formato Digital. MS-DOS.

FROM SOFTWARE. **Dark Souls.** 2011. Formato Digital. PlayStation 3, Xbox 360, Microsoft Windows, Nintendo Switch, PlayStation 4, Xbox One.

HARMONIX. **Guitar Hero.** 2005. Formato Digital. Play Station 2.

\_\_\_\_\_. **Guitar Hero II.** 2006. Formato Digital. PlayStation 2, Xbox 360.

ID SOFTWARE. **Doom.** 1993. Formato Digital. MS-DOS.

\_\_\_\_\_. **Quake.** 1996. Formato Digital. MS-DOS,

AmigaOS, Classic Mac OS, Sega Saturn, Nintendo 64, Linux, Microsoft Windows.

INDEPENDENTE. **Guitar Hero Brazuca.** S/A. Formato Digital. Play Station 2. Windows.

MAIET ENTERTAINMENT. **GunZ: The Duel.** 2003. Formato Digital. Microsoft Windows.

MOJANG. **Minecraft.** 2011. Formato Digital. Microsoft Windows, Mac OS X, Linux.

NEVERSOFT. **Guitar Hero III: Legends of Rock.** 2007. Formato Digital. PlayStation 2, PlayStation 3, Wii, Xbox 360, Microsoft Windows, Mac OS X.

NINTENDO. **Super Mario Bros.** 1985. Cartucho. Nintendo Entertainment System.

NINTENDO R&D2. **Super Mario Advance 4: Super Mario Bros 3.** 2003. Formato Digital. Game Boy Advance.

PAGODAWEST GAMES. HEADCANNON. **Sonic Mania.** 2017. Formato Digital. Nintendo Switch, PlayStation 4, Xbox One, Windows.

PAJITNOV, Alexey. **Tetris.** 1984. Cartucho. Commodore 64.

VALVE. **Half-Life.** 1998. Formato Digital. Microsoft Windows, PlayStation 2, OS X, Linux.

VALVE CORPORATION. **Half-Life 2.** 2004. Formato Digital. Microsoft Windows, Xbox, Xbox 360, PlayStation 3, OS X, Linux, Android.



## Notas

- 1 Doutor em Comunicação (Escola de Comunicação UFRJ). Pós-doutorando em Ciências Sociais Aplicadas (PPGCOM UFF – Rua Tiradentes, 148, Niterói/RJ, CEP:24210-510). E-mail: jmessias.santos@gmail.com.
- 2 Doutorando em Comunicação (PPGCOM UERJ – Rua São Francisco Xavier, 524, 10º andar, sala 10.129, bloco F, Pavilhão João Lyra Filho Maracanã, Rio de Janeiro/RJ, CEP: 20550-900). E-mail: ivanmussa@gmail.com.
- 3 Conferir, por exemplo, o trabalho de Arsenault et al. (2013), disponível em: <<https://www.gamejournal.it/graphical-technologies-innovation-and-aesthetics-in-the-video-game-industry-a-case-study-of-the-shift-from-2d-to-3d-graphics-in-the-1990s/>>.
- 4 Tanto Alexey Pajitnov quanto Markus Persson eram programadores que usavam seu tempo “livre” no trabalho para programar projetos pessoais, paralelamente aos que desenvolviam oficialmente. A id Software, empresa que produziu Doom, surgiu da dissidência interna de uma empresa maior, a Softdisk, quando John Carmack apresentou uma inovação de processamento gráfico (inicialmente bidimensional) a Tom Hall e John Romero.
- 5 Uma explicação detalhada do caso pode ser encontrada em Bunnyhop (2016).
- 6 Disponível em Nelson (2012).
- 7 Disponível em Walter (2012).
- 8 No original: “Spacewar, as it was developed by Steve

Russel in 1962, was the product of a given territory, the MIT, developed as it was by engineering students who possessed certain of the ludic skills required to appropriate it, certain shared fictional reference points (such as a fascination for science fiction) and certain of the ludic activity’s values. This logically fitted into a practice of hacking which amongst other things advocated the free circulation of information and openness to objects being modified by their users”.

- 9 Como foi mostrado com Galloway (2012) e Kittler (1995), a BIOS (Basic Input/Output System) compõe a memória permanente de qualquer objeto computacional – consoles possuem a mesma arquitetura de um computador. Ela é um programa “gravado diretamente no silício”, por isso para ter controle total do aparelho seria necessário acessá-la.
- 10 Embora a franquia tenha sido oficialmente descontinuada por sua distribuidora, Activision, em 2010, durante o auge de sua popularidade, entre 2005 e 2009, e, em certa medida, ainda os dias atuais, desperta o interesse de jogadores e fãs de música, geralmente rock.
- 11 Disponível em Docslide (2015).
- 12 No original: “[...] it is clear that what is often termed ‘emergent gameplay’ in the game development community (see also Smith 2001; Salen and Zimmerman 2004) is at the very heart of speedrunning, even though much of this emergence is centred on the exploitation of glitches rather than the use of an open set of tools or resources. As such, it is difficult to define exactly where and whether the emergence is intended by the developers or is a consequence of their

inevitable programming errors and anomalies and limitations of their modelling and design”.

- 13 No original: “If we conceive of configuration as a way of engaging not just immediate game elements, but also the game’s social and material conditions - and by extension, the conditions of other rule-systems such as work and citizenship -- then it may be very important to insist upon the difference between play and interpretation, the better to resist immersion. Any analogue of literacy for interactive media would probably need to encompass such resistance”.
- 14 No original: “The engine is the core piece of game software, consisting of several components such as the renderer that visualizes the game space, a physics engine, networking code, artificial intelligence code, a sound system and other parts”.
- 15 No original: “Engines are purposefully modular in design so that they can enable the upgrading of particular engine parts without ‘breaking the code’, or third parties to develop plug-ins (‘sub-engines’) to offload complex software routines. Such sub-parts are also known as ‘libraries’ and are constantly worked on by developers”.
- 16 Um resumo da história completa da progressão de recordes em Super Mario Bros. pode ser encontrada em Salt (2017).
- 17 No original: “[...]‘found’ actions or items that accelerate or improve a player’s skills, actions, or abilities”.
- 18 Aqui, Peter Krapp faz uma referência a uma das epí-



grafes do quarto capítulo de seu livro *Noise Channels: Glitch and error in digital culture*, uma frase de Samuel Becket: “Ever tried, ever failed. No matter. Try again, fail again, fail better.”

19 É necessário chamar atenção para o fato de que a palavra *invenção*, aqui, não se refere necessariamente a atos grandiosos, como a invenção da máquina a vapor ou do cinema. 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 humana. Isto vale inclusive para experiências comuns, como a de quem volta para a casa que morou na infância e é invadido por uma sensação de estranhamento: há uma perturbação na estabilidade do reconhecimento que exige a invenção de uma nova cognição (Kastrup, 2007).

20 Disponível em Strategywiki (s/a).

21 No original: “We propose to call the processes by which situations are transformed into games ‘ludicisation’ processes, thereby choosing to make a distinction between this idea and other notions such as that of gamification which does not take into account the shifting, process-linked aspects of ludic phenomena, preferring instead to attribute fixed characteristics to these phenomena”.