

Discutindo o mundo através de algoritmos: filosofia política em simulações de computador¹

Pierre Depaz

Professor de Emerging Media na NYU Berlin, Berlin, Alemanha .
pierre.depaz@nyu.edu

Tradução: Luiza Santos

Universidade Federal do Rio Grande do Sul.
luizacdsantos@gmail.com

Revisão da Tradução: Fabiana Freitas

Universidade Federal do Rio Grande do Sul.
frrfreitas@gmail.com

Resumo

Esse artigo irá examinar em quais aspectos as simulações de computador são políticas. Para tanto, inicia buscando compreender de que forma os jogadores interagem com aquelas simulações, para posteriormente destacar e comparar os diferentes papéis e responsabilidades do jogador e do designer sobre os modos como o pensamento político pode ser cristalizado durante o jogo. A partir deste exame, irei delimitar de quais formas as simulações de computador podem contribuir para o pensamento filosófico e político.

Palavras-chave

Simulações de Computador. Política. Programação. Design.

1 Introdução

Todo software gera um impacto na forma como percebemos e agimos no mundo, de transações bancárias até mídias sociais, passando por registros hospitalares e urnas eletrônicas. Como pontos em comum, todos estes fazem referência direta a nossa realidade,

¹ Esta tradução corresponde ao manuscrito do autor. Uma versão definitiva foi publicada com o título "Computer Simulations as Political Manifestos" na obra Games and Politics, Goethe-Institut e.V. (2016). Esta publicação da tradução foi permitida pelo Goethe Institut e.V. como parte dos direitos de uso pessoal e não permite redistribuição.

agindo sobre esta para modificá-la, com ou sem input² humano direto (KITCHIN; DODGE, 2011). Em contrapartida, softwares interativos e, mais especificamente, jogos de computador, possibilitam input e feedback de e para os jogadores em tempo real, permitindo exploração maior das características intrínsecas desse sistema auto-contido. Como sistemas auto-contidos, referem-se a nossa realidade apenas através de metáforas e representações, apresentando “uma versão de” ao invés de “serem baseados em”. Na fronteira entre realidade e ficção, nós começamos a ver pistas da presença simultânea da *praxis* e da *poiesis*, entre a atualização da interação e a abstração criativa de um mundo ficcional.

Simulações de computador, então, nos permitem interagir com uma representação responsiva do nosso mundo. Simulações científicas computadorizadas já estão em uso em múltiplos campos das ciências sociais (EPSTEIN, 2002) e relações internacionais (CEDERMAN, 1997), para organizar e monitorar a evolução de uma grande quantidade de dados baseada em um determinado conjunto de regras definido pelo pesquisador. Na busca pela verdade científica, o modo de funcionamento interno destas simulações são detalhadamente documentados para marcar o viés inerente do próprio pesquisador. Por outro lado, entretanto, a atualidade das simulações de jogos durante seu período de execução também contém uma certa visão de mundo sobre o ambiente que envolve os jogadores, sem revelar seu código fonte ou reconhecer seu impacto político.

Esse artigo irá examinar em quais aspectos as simulações computacionais são políticas e de que forma os jogadores interagem com elas, para posteriormente destacar e comparar os papéis e responsabilidades distintos, tanto do jogador quanto do designer, em como o pensamento político pode ser cristalizado durante o jogo. A partir deste exame, irei delimitar de quais formas as simulações de computador podem contribuir para o pensamento filosófico e político. Ao longo deste artigo, os termos designer e desenvolvedor serão usados de forma intercalada, uma vez que defino ambos como autores do sistema e do software, mesmo que estes não estejam limitados a uma única pessoa.

² Nota da tradutora: o autor explica, neste mesmo texto, que *input* do usuário é o conjunto de dados inicial necessário para que a simulação ocorra. O pesquisador também explicará, mais adiante, a diferença entre *input* e *output*: enquanto no *input*, a ação vem do próprio jogador, o *output* é sempre alterado pelo algoritmo pelo qual a ação é processada.

2 As características políticas da simulação

No centro de qualquer sistema de jogo existe a ideia de interação baseada em regras (SALEN; ZIMMERMAN, 2012). Interação como questão/resposta, como um diálogo, se torna automático em sistemas digitais onde as regras não são mais atualizadas pelos jogadores, mas sim pelo código, que age como barreiras imutáveis que modulam a agência do jogador. Simulações de computador são loops de feedback regulados pelo código, assim como os nossos sistemas social e político são regulados pelas leis (WIENER, 1993). Como representações de um outro fenômeno, simulações de computador conferem direitos à pessoa que está jogando, não apenas determinando como o mundo virtual reage, mas também como o jogador pode agir. Como uma representação formal que utiliza signos escritos, visuais e sonoros para representar o mundo, podemos partir do trabalho de Jacques Rancière (2013) para traçar um paralelo entre a apreciação estética e a apreciação política. Assim, se torna aparente que simulações de computador são peças auto-contidas da filosofia política.

Mesmo que os jogos sejam definidos pelo fato de serem interativos, a introdução da automação e da computação nos jogos digitais possui uma consequência bastante específica. Enquanto o ato de jogar historicamente implicava em uma constante renovação do acordo implícito entre jogadores de seguir as regras (HUIZINGA, 2004), a automação dessas regras por computadores redefine a agência do jogador. Ao invés de atualizar estas regras ao segui-las, é a própria ação do jogador que está sendo atualizada por um sistema de jogo que é monitorado constante e automaticamente, sessenta vezes por segundo. Ao remover o acordo entre jogadores, os jogos digitais melhoram a sua *affordance* de jogo, que age de forma unilateral. Existe uma forma de jogar e é assim que os jogadores devem fazer. Ainda que seja possível argumentar que a criatividade do jogador ainda está viva e forte nos jogos de computador modernos, eu gostaria de pontuar que é apenas o escopo do enquadramento que foi ampliado, ou seja, o espaço, e não o enquadramento em si. Tomando como exemplo o jogo Grand Theft Auto V (GTA V), no qual os designers e engenheiros afirmam existir praticamente um número ilimitado de atividades, existe uma diferença entre os objetos e atores nos quais se pode aplicar a interação permitida (o jogador pode matar qualquer um, dirigir para qualquer lugar) e o que é a interação de fato (é impossível ter uma conversa com qualquer personagem não-jogável). Se o sistema não permite, é como se não existisse

no mundo virtual. Enquanto no *input* a ação vem apenas do próprio jogador, o *output* é sempre alterado pelo algoritmo no qual é processada. Qualquer ação supõe então a reação, a resposta, que é essencialmente a soma da ação de input e da forma como o sistema decide processá-la, de como o sistema decidiu qualificá-la.

O jogador possui um papel duplo no sistema no qual está jogando. Por um lado, ainda é o agente, a fonte de input para o sistema, mas, por outro lado, o jogador é também objeto de seu próprio output, uma vez que processado pelo algoritmo do jogo. Essa dualidade é necessária para permitir uma experiência de jogo, para permitir a metáfora do push/pull (empurre/puxe) tão comumente utilizada para caracterizar os jogos.

Aqui nós encontramos a primeira correlação entre filosofia política e política em geral. Um dos principais tópicos desse campo é a relação entre o indivíduo e a comunidade, e as relações de poder entre um e outro. Quanto um indivíduo pode sacrificar para ganhar pertencimento a um grupo? Quanta agência pode ser removida de um jogador sem remover o senso de agência do mesmo. A agência do jogador em um jogo de computador pode ser vista como a liberdade do cidadão, que é capaz de agir diferentemente em diferentes regimes, dependendo do conjunto de regras existente. O questionamento filosófico destas regras está no centro do braço político do Iluminismo desde seu princípio (LA BOETIE, 1987).

O gênero específico da simulação, como uma categoria estética dentro do campo mais largo dos jogos de computador, exhibe características particulares. Enquanto outros jogos podem ser tão abstratos quanto desejarem - variando de Tetris (PAJNITOV, 1984) ao Super Hexágono (CAVANAGH, 2012) -, simulações precisam ter suas raízes em fenômenos e organizações conhecidas. Eles fazem isso ao combinar estruturas formais existentes de certos aspectos de nossa sociedade e a forma como nós olhamos para eles em nosso imaginário comum. Os temas das simulações computacionais variam amplamente, desde a reencenação de fenômenos físicos em Flower (CHEN, 2009) até times ficcionais compostos de jogadores reais no jogo em série NBA 2K (ELECTRONIC ARTS, 1999-2018).

Essa relação entre um fenômeno preexistente e a sua representação em jogos de computador pode ser vista como uma remediação (BOLTER, 1998). Enquanto esse conceito foi amplamente estudado em mídias não-interativas, a ideia de representar um mesmo conceito com *affordances* diferentes gera resultados interessantes em simulações de computador. O conceito não é apenas transposto para um programa de computador, mas se torna manipulável. Os seus outputs, a sua linearidade, não são mais definidos e o jogador, se

limitado nas ações que pode executar, não é limitado em termos de objetivos que podem ser alcançados dentro destes limites. O produto remediado e digitalizado pode ser acessado de um novo ângulo.

Por estas simulações serem construídas tendo como base conhecimentos prévios (como uma cidade funciona, como um humano se comporta, como um atleta dribla...), existe um constante vai e vem que acontece entre o conhecimento preexistente do modelo e a exploração dos parâmetros da simulação. Para que essa relação possa aparecer e se sustentar, é preciso que exista aparência de credibilidade. É preciso que exista uma conexão formal entre o modelo e a representação que acontece na mente do jogador. O modelo mental do fenômeno representado é então projetado em uma simulação na forma de input do usuário. Input do usuário, de fato, é o conjunto de dados inicial necessário para que a simulação ocorra, e esse conjunto de dados é geralmente baseado em experiências prévias ou suposições. Portanto, possuir um mundo no qual se pode acreditar e com o qual é possível se relacionar permite que o *input* do usuário seja mais verdadeiro e o output resultante seria mais interessante de se investigar a partir de um ponto de vista filosófico e mais coerente com o ponto de vista do jogador. Se, então, pegarmos o fato deste input ser mais baseado nas crenças e conhecimentos do jogador, nos distanciamos de um ajuste tradicional de ação e reação que acontece com outros tipos de jogos nos quais a ação é mais intensa.

Quando o conjunto de dados iniciais da simulação é processado e o que o output mostra é uma situação que não está em conformidade, a reação usual do jogador não é de entender o que houve de errado na execução da ação, mas na suposição por trás daquela ação. A credibilidade do mundo aumenta o sentimento de estranheza, de discrepância entre a visão de mundo assumida e a parte programada. O Vale da Estranheza (FREUD, 1919), o sentimento indistinguível de falta de realidade nos força ao questionamento. A partir de um instinto é possível que se forme uma exploração lógica de porque a representação de um mundo no qual o jogador está interagindo não está alinhada com as suas expectativas.

Assumindo que o homem é um animal político e social (DREFCINSKY, 2009), então, cada um de nós possui uma opinião sobre as duas maiores questões formuladas no Iluminismo: “De que forma nós inventamos um dado sistema para vivermos juntos?” e “Esse sistema é o melhor possível?”.

A palavra “sistema” aqui pode referir tanto às nossas estruturas políticas do passado e do presente, mas também a menor, auto-contida estrutura da simulação digital. Uma vez

que o *game design* por trás da simulação é muito diferente de outras práticas de game design - especialmente em termos de como apresentar aos jogadores o objetivo e as ferramentas para atingir aquele objetivo -, é legítimo para os jogadores tentar alcançar suas próprias metas dado um determinado conjunto de ferramentas. O que acontece aí é a aparição de um conflito entre as visões de mundo do jogador e do designer. Dado um ponto de referência similar fora do sistema do jogo, existe a realização que a apreciação daquele ponto de referência não é o mesmo. Como Jacques Rancière (2013) coloca, esse reconhecimento dos elementos formais no nosso mundo é equivalente a uma instância política, ligando a estética e a política. Simulações de computador, então, dão um passo a mais ao permitir confrontar, em tempo real, duas apreciações de mundo e deixar que estas possam interagir uma com a outra para explorar os espaços onde a discrepância e, portanto, a dialética podem acontecer.

A forma como um sistema pode confirmar ou confrontar suposições do jogador não é objetiva. Como todo outro software, este é escrito por humanos e suas qualidades e funcionalidades são baseadas no que o desenvolvedor julga necessário. Explorar os limites da simulação, então, é explorar os limites das crenças de outra pessoa. Na seção seguinte iremos analisar como estas crenças são implementadas do lado técnico, do ponto de vista do desenvolvedor, como estas são recebidas pelo jogador e como esses diálogo possui uma nova perspectiva de representar e entender nossas convicções.

3 O código de computador como um novo manifesto

Historicamente, computadores foram inventados para facilitar e automatizar processos que não podem ser facilmente executados por humanos - nomeadamente, organizar e rearranjar uma grande quantidade de dados. O resultado deste propósito de simplificar tarefas é a bifurcação da forma como os processos são implementados, dando ao usuário final apenas o output que o desenvolvedor quer que ele veja. Entretanto, como iremos ver, o processo de implementar algoritmos permitindo que a simulação funcione é extremamente relevante para uma nova forma de prática filosófica. Esta prática acontece em três etapas distintas. A primeira delas é reconhecer as convicções do desenvolvedor, que segue para a implementação destas convicções no software - ou seja, no ato de programar

em si. O último passo é então a apresentação para o usuário quais escolhas específicas são feitas que contribuem para o reconhecimento - ou falta de - do homem por trás da máquina.

Não limitado ao desenvolvimento de jogos, a arte e prática da engenharia de software depende mais de uma organização de conceitos amplos do que da implementação direta. Se a engenharia de software clássica, ou seja, fora do jogos, foca mais na necessidade de otimizar o funcionamento de uma tarefa, no desenvolvimento de jogos essa otimização é centrada em torno do prazer do jogador ou da construção de mundo, aspectos que muitas vezes se cruzam.

Trabalhos recentes nos estudos de ciência e tecnologia lançaram uma luz em como designers e objetos desenhados incorporam valores (NISSEMBAUM; FLANAGAN, 2015). Desde o planejamento urbano de Robert Moses, feito para evitar que certos segmentos da população acessassem determinadas áreas geográficas (KROES; VERBEEK, 2014), até submissões de formulários online que apenas especificam uma certa perspectiva de gênero ou raça, designers colocam sua visão de mundo em seus produtos, quer façam isso conscientemente ou não. Jogos digitais não são uma exceção à regra e, portanto, não apenas possuem valores e percepções, mas também permitem que jogadores interajam com elas. Além disso, pertencendo a um ramo mais emotivo dos produtos de software, os desenvolvedores atingem conexões subjetivas com seus jogadores através da escrita do código. Se a intenção do designer é criar um ambiente baseado na sua percepção do que pode trazer alegria, medo, excitação ou surpresa, o problema permanece na implementação dessa intenção.

O conceito de *sampling* no processamento de sinal é a redução de um sinal contínuo para um sinal discreto. *Sampling* ocorre quando um som é convertido da modulação por pressão de ar ou informação luminosa de uma foto em um conjunto discreto de dados em bytes. Um dos desafios de *sampling*, então, é representar tão acuradamente quanto possível o produto original, quer seja um som, uma imagem ou uma ideia. De fato, o processo no qual o desenvolvedor se envolve ao converter ideias abstratas é próximo tanto do *sampling* quanto do método cartesiano. O método cartesiano, nesse contexto, pode ser visto de uma perspectiva de subdivisão de um problema ou dificuldade em sistemas menores, que são assumidos como menos complicados (DESCARTES, 2009). A consequência da influência desse paradigma pode ser visto como o componente filosófico da emergência, ou gestalt digital, no qual o todo se torna mais do que apenas a soma de suas partes.

Estes dois conceitos podem nos ajudar a examinar mais de perto a prática de implementação de simulações de computador, que eu defino como a representação do mundo usando uma estrutura de dados. Por um lado, existe a necessidade de quebrar o que consideramos conceitos humanos amplamente utilizados em jogos de computador - como amizade, antagonismo, exploração e etc - em instruções que possam ser lidas por uma máquina; por outro lado, essa prática permite uma leitura mais focada do que estes conceitos realmente são. Enquanto um designer implementa uma mecânica na qual, por exemplo, um recurso é esgotado, o processo real de esgotar esse recurso precisa ser cuidadosamente escrito para que um computador possa agir a partir dessas instruções. As atitudes podem variar desde simplesmente diminuir a contagem em um determinado período de tempo (como na maior parte dos jogos de estratégia em tempo real) até a implementação de um algoritmo que permite que os recursos prosperem tanto quando são cuidados pelos jogadores quanto quando não são (PERSSON, 2009). Escrever conceitos com os quais os jogadores consigam se identificar na mídia computador o código força uma auto-reflexão, que é tanto expressiva (devido ao fato de que o código funciona, executa) mas que também pode ser referenciada, uma vez que esta escrita.

Esta prática de programar conceitos também permite uma discussão filosófica de uma forma nova. Enquanto o diálogo da filosofia se desenvolveu basicamente a partir de interpretações distintas de trabalhos anteriores e de leituras detalhadas de teorias anteriores, o código é interpretado de uma forma única. O intérprete, permitindo construir um executável de um código fonte, é outra parte de um software que valida o código que está lendo e aprova este como um software em ação. Para um dado código, existe apenas um resultado possível, passível de interpretação. Portanto, quando estamos discutindo visões de mundo, se expressar através de um código, a discussão se foca não mais no que o código significa mas como o código age. Simulações de computador, então possibilitam um envolvimento mais ativo do pensamento filosófico, uma vez que estamos vendo este em ação, vendo este ser realizado e atualizado, não apenas fazendo conjecturas. Como ambos falam sobre a intenção do designer, o código fonte e o executável coexistem lado a lado, eles possibilitam uma lente dupla através da qual podemos visualizar a posição do designer.

Se o designer está escrevendo leis imutáveis para a simulação que é apresentada ao jogador, então podemos ver uma correlação com os filósofos políticos do Iluminismo. Quando pensadores do século XVII e XVIII na Europa Central começaram a conceitualizar estruturas alternativas de poder que se distanciavam dos modelos monárquicos e

episcopais existentes, havia uma necessidade de novas regras sobre as quais os humanos pudessem viver. Os manifestos políticos que seguiram, como “O Contrato Social” (ROUSSEAU, 2007), “Leviatã” (LOCKE, 2005) ou “Dois tratados sobre o Governo” (HOBBS, 2001) foram tentativas no pensamento filosófico onde os autores tentaram criar o melhor conjunto de regras sobre os quais uma sociedade poderia operar, dado um determinado conjunto de suposições sobre o comportamento e desejo humanos. Se nós olharmos para o código fonte, nós estamos olhando para o equivalente algorítmico da filosofia política da perspectiva do designer. A principal diferença, então, é a habilidade de testemunhar as consequências destas regras baseadas em suposições e como estas impactam uma dada população de agentes.

O principal problema que surge da análise é o ofuscamento deliberado da mecânica da simulação quando em um ambiente de jogo. Pelo bem da interface gráfica, da experiência do usuário e do prazer de jogar, as simulações de computador possibilitam apenas a visão de uma das perspectivas de um fenômeno duplo. De fato, existe uma tendência por parte dos designers de ou esconder a formulação do comportamento, ou exibi-la como tal, e não como aquilo que ela representa. Essa abordagem pode ser encontrada em simulações como o jogo Spore, onde a evolução das espécies é representada de forma estritamente formal e visual, partindo da ideia comum de história linear para deixar o jogador preencher os espaços vazios sobre a forma como o sistema realmente funciona. De forma oposta, jogos como Civilization são baseado em um entendimento acurado dos números apresentados ao jogador. Deste ponto de vista, eles existem pelo seu próprio benefício, como ferramentas para o jogador manipular a fim de alcançar uma das condições de vitória. Ainda assim, em nenhum lugar desta representação é confirmado que estes sejam os números reais, as estruturas de dados reais usadas para representar o mundo. Por causa do desejo de mediar até mesmo o sistema do próprio jogo, não existe forma imediata para que jogador possa fazer uma leitura detalhada das suposições e opiniões políticas do designer na sua representação de mundo. Essa camada adicional reintroduz um nível de interpretação que estava ausente e aproxima a concepção do jogador para uma visão de mundo pré-estabelecida (no caso de SimCity, sustentabilidade urbana (KOLSON, 1996) e, no caso de Spore, um entendimento supérfluo das teorias Darwinistas).

Nessa seção examinamos o papel do desenvolvedor tanto a partir do modo cartesiano quando através do sampling, no qual este possui ideias e opiniões políticas e que, para serem comunicadas ao jogador, precisam ser quebradas em pedaços menores e

escritas em código, para posteriormente serem interpretadas por outro software. Entretanto, uma vez que o pensamento dialético é um dos pilares da filosofia ocidental clássica, seria possível elaborar uma nova forma de dialética ao estabelecer uma relação entre o código do jogo e a simulação que está rodando, mas, desta vez, do ponto de vista do jogador.

4 Simulação como uma dialética atemporal

Uma vez que estabelecemos a possibilidade de um diálogo mais direto entre jogador e desenvolvedor, iremos agora olhar para alguns exemplos particulares de como esse relacionamento pode produzir pensamento filosófico no caso de interação com simulações de computador.

A dissonância ludonarrativa, um tópico bem discutido em *game studies*, é o fenômeno de perceber a discrepância entre ação e conceito (UPTON, 2015) - ou seja, o que está sendo possibilitado pelo designer em contraponto com o que é apresentado pelo mesmo designer. Entretanto, se olharmos para isso a partir da perspectiva do jogador, nós temos essa discrepância entre a visão de mundo permitida pelo designer e a subjetividade da visão de mundo do jogador. Se a suspensão da descrença acontece com a realidade como ponto referencial, então a suspensão não é mais válida quando falamos de representações idealistas de mundo. Se a primeira é perceber a dissonância, então a segunda seria imaginar por que essa dissonância existe, ou seja, porque dois conceitos de mundo estão em conflito e o que pode ser aprendido a partir deste conflito.

Podemos tomar como exemplo o aumento de recursos em Minecraft e pensar como eles representam a agência e necessidade humanas em relação aos recursos naturais. Uma forma é que recursos crescem devagar sem receber água diretamente, enquanto o input humano pode modificar a topografia e possibilitar que estes tenham uma fonte de água constante e uma colheita maior. Se alguns aspectos da simulação são deixados de lado, como bactérias e estações, é para sublinhar a possibilidade da ação humana como fator principal. Mesmo que o pensamento fenomenológico seja prevacente na ideologia dos jogos digitais - uma vez que são softwares centrados no usuários -, olhando para o comportamento baseado em regras da simulação em relação com o comportamento ideal do

jogador pode nos fornecer novas perspectivas de um dado fenômeno, como crescimento biológico e reprodução.

A parte do designer nessa troca, entretanto, permanece estática. Ainda que possa aparecer durante a simulação comportamentos que não foram escritos pelo designer, existem poucos jogos que modificam de fato o seu código enquanto estão em funcionamento - um destes é *Lose/Lose* de Zach Gage, eventualmente levando a uma pane do computador. Portanto, a única discussão que acontece enquanto a simulação está correndo é interna, ou seja, restrita ao próprio jogador. O jogador pode refletir sobre suas ações, mas como suas ações levarão a um mesmo resultado (pois depende que aquela ação seja possibilitada pelo designer ou não), chega um momento para a reflexão da ontologia dessas ações. Nós saímos do “porque eu não consigo fazer isso?” para o “porque eu não tenho permissão para fazer isso?” e finalmente chegamos em “porque eu estou tentando fazer isso?”. A consciência do jogador de suas próprias ações apenas pode existir passando por um entendimento do desejo do designer de canalizar estas ações para uma direção particular.

Essa interrogação, então, esse desejo de desvendar como um sistema funciona, lembra não apenas os questionamentos políticos do Iluminismo, mas também o paradigma geral do Iluminismo em termos de aquisição de conhecimento (KANT, online). Se a revolução newtoniana levou a um entendimento mais amplo e profundo da física, então é possível que a curiosidade daquele jogador e a dissonância ludonarrativa possa levar a um interesse mais generalizado aos detalhes técnicos das simulações de computador e mais genericamente a uma alfabetização em linguagem de programação. O fenômeno *Mooding*, no qual jogadores decidem modificar o código de um dado jogo para incluir seus próprios estilos de jogar (removendo a viagem instantânea na série *Elder Scrolls*) ou suas visões de mundo (integrando unidades jogáveis do Estado Islâmico no *Arma III* para fornecer uma representação distinta das forças geopolíticas³), é essencialmente uma intervenção do próprio jogador baseado na não concordância com a simulação apresentada.

Essa possibilidade ainda precisa de um pré-requisito do software: que ele seja de código aberto. Se os jogos de computador devem agir como manifestos, então o seu código deveria ser tão mutável quanto a prensa móvel era, quando transformou o ambiente midiático europeu. Se existem exemplos de bases abertas de código fonte, isso vem sendo

³ HALL, J., The Videogame That Allows You to Play As An ISIS Fighter, disponível em: <http://www.dailymail.co.uk/news/article-2937641/ISIS-fighters-distributing-video-game-allows-players-play-role-Islamist-kill-Westerners.html>

motivado por avanços tecnológicos. O lançamento de *Doom* foi centrado em torno dos gráficos em 3D em tempo real, enquanto o lançamento do código de *Limbo* foi mais focado na implementação do design de áudio. Mesmo quando *SimCity* passou a ser de domínio Público, uma das simulações mais proeminentes e evidentemente política, foi recebida com a possibilidade de hackear no sentido original - isto é, mexer com a tecnologia apenas pelo prazer de mexer com a tecnologia.

Isso também implica uma consciência do ponto de vista do designer e do desenvolvedor em termos do seu impacto no mundo. Diversos autores no campo dos Estudos de Software buscaram compreender as consequências do comportamento codificado no nosso mundo físico e semântico. Enquanto a maior parte dos trabalhos não se referem diretamente aos jogos digitais, a retórica é a mesma: códigos intangíveis afetam o mundo tangível (COX, 2014). Particularmente em termos de discurso, designers de jogos precisam perceber que executar um código é uma ação e que esta ação possui consequências que são, senão materiais, pelo menos morais e políticas.

Essas duas perspectivas opostas, então, estão dormentes dentro das simulações. Apenas através do nosso desejo mútuo de reconhecer uma a outra - como designers reconhecem seus impactos na visão de mundo do jogador e os jogadores reconhecem a presença de uma consciência subjetiva atrás dos scripts - pode ser possível alcançar uma discussão filosófica que seja similar àquela do Iluminismo. Examinar o código através das instruções passo a passo funciona como um manifesto de leitura, enquanto reconhecer seu papel em um impacto cultural e idealmente público forçaria os designers a formular sua lógica, seu sistema em uma formato que é tanto legível quanto executável.

5 Conclusão

Em conclusão, nós vimos que as simulações de computador são especiais pois representam um fenômeno vivo e com o qual podemos nos relacionar. Elas estão remediando e representando esse fenômeno para que o jogador possa experimentá-lo no jogo, tendo como base conhecimentos anteriores. Como todos os jogos de computador, simulações são baseadas em *input* e *output*. É a natureza do *input* e o processo do *output* que diferenciam as simulações do restante. Simulações representam um ponto de encontro

entre duas perspectivas de uma mesma situação percebida, aquela do designer e a do jogador, vindo de direções separadas com diferentes subjetividades.

Esse processo político de representação de mundo como simulações acontece, para o designer, através da escrita do código. Esse ato pode ser compreendido como uma lente dupla de *sampling* e método cartesiano. A quebra de conceitos humanos em códigos não ambíguos permite que uma comunicação de ideias e pontos de vista sem precedentes aconteça. Ainda, esse processo é escondido do jogador, uma vez que o que é representado proceduralmente nem sempre é o que é representado visualmente ou semanticamente. O jogador é então deixado em seu lado desse diálogo em potencial, onde a única possibilidade de pensamento filosófico é através da auto-reflexão.

Essa perspectiva ontológica das ações do jogador pode então levar ao pensamento filosófico, através de um movimento conjunto tanto do lado do jogador quanto do lado do designer. Enquanto o jogador segue um paradigma similar que chegou a seu auge durante o Iluminismo, aquele do exame em detalhes de suposições anteriores tanto quanto o desejo de descobrir o modo de funcionamento interno de sistemas existentes. Em paralelo, para o discurso político ser completo, os desenvolvedores devem tornar o seu trabalho acessível, para que uma leitura detalhada pelo público seja possível e os jogadores possam colocar em perspectiva suas preferências políticas particulares no mundo representado, tanto com o código escrito quanto com o código executável de outro indivíduo.

Referências

BOLTER, J; GRUSIN, R. **Remediation**: Understanding New Media. Cambridge: MIT Press, 1998.

CEDERMAN, L. **Emergent actors in world politics**: how states and nations develop and dissolve. Princeton: Princeton University Press, 1997.

COX, G. **Speaking Code**. Cambridge: MIT Press, 2014.

DESCARTES, R. **Discurso do Método**. São Paulo: Escala, 2009

DREFCINSKY, S. **Plato and the Social Contract**. [s.l.]: University of Winsconsin, 2009.

EPSTEIN, J. Modeling Civil Violence: An Agent-Based Computational Approach. In: PROCEEDINGS OF THE NATIONAL ACADEMY OF SCIENCES, 2002, [s.l]. **Proceedings...** [s.l]: 2002.

FREUD, Sigmund. **The “Uncanny”**. 1919. Disponível em: <http://web.mit.edu/allanmc/www/freud1.pdf>. Acesso em: 09 abr. 2019.

HOBBS, T. **Leviatã**. São Paulo: Companhia das Letras, 2001

HUIZINGA, J. **Homo Ludens**: o jogo como elemento de cultura. São Paulo: Perspectiva, 2004

KANT, I. **What Is Enlightenment?** Disponível em: <http://www.columbia.edu/acis/ets/CCREAD/etscc/kant.html>. Acesso em: 09 abr. 2019.

KITCHIN, R; DODGE, M. **Code/Space**: Software and the everyday life. Cambridge: MIT Press, 2011.

KOLSON, K. **The politics of SimCity**. [s.l]: American Political Science Association, 1996.

KROES, P; VERBEEK, P. **The Moral Status of Technical Artifacts**. Berlim: Springer, 2014.

LA BOETIE, E. **Discurso da servidão voluntária**. São Paulo: Brasiliense, 1987.

LOCKE, J. **Dois Tratados sobre o Governo**. São Paulo: Martins Fontes, 2005.

NISSEBAUM, H; FLANAGAN, M. **Values at Play in Digital Games**. Cambridge: MIT Press, 2015.

ROUSSEAU, J. **Contrato Social**. Porto Alegre: L&PM, 2007.

SALEN, K, & ZIMMERMANN, E. **Regras do Jogo - Fundamentos do Design de Jogos**. São Paulo: Blucher, 2012.

UPTON, B. **The Aesthetics of Play**. Cambridge: MIT Press, 2015.

WIENER, N. **O uso humano de seres humanos**. São Paulo: Cultrix, 1993.

Referências Ludográficas

CAVANAGH, T. **Super Hexagon**, 2012.

CHEN, J. **Flower**, 2009.

ELECTRONIC ARTS. **NBA 2k** (series), 1999-current.

PAJNITOV, A. **Tetris**, 1984;

PERSSON, M. **Minecraft**, Mojang, 2009.

Discussing the world through algorithms: political philosophy in computer simulations

Abstract

This paper will examine in which aspects computer simulations are political. To this end, it begins by seeking to understand how players interact with those simulations and then highlights and compares the different roles and responsibilities of the player and the designer in how political thinking can be crystallized and enforced during play. Drawing on this examination, I outline in what ways computer simulations can contribute to philosophical and political thought.

Keywords

Computer Simulation. Politics. Programming. Design.

Copyright (c) 2019 Pierre Depaz. Creative Commons License. Este trabalho está licenciado sob uma licença Creative Commons Attribution-NonCommercial 4.0 International License. Os Direitos Autorais dos artigos publicados neste periódico pertencem aos autores, e os direitos da primeira publicação são garantidos à revista. Por serem publicados em uma revista de acesso livre, os artigos são de uso gratuito, com atribuições próprias, em atividades educacionais e não-comerciais.

