Thomas não estava sozinho:

A simulação do funcionamento do código no jogo Thomas Was Alone

Natália Ferraz Menezes
Departamento de Jornalismo
Universidade Federal de Ouro Preto
Mariana, Brasil
nataliaferrazm@gmail.com

***Abstract*—O artigo pretende refletir sobre a capacidade persuasiva do software à luz do conceito de retórica procedimental a partir da possibilidade de simulação de sistemas reais ou imaginários. Para isso, utilizamos o jogo digital Thomas Was Alone para exemplificar como um artefato computacional expressivo pode simular o próprio funcionamento de um programa computacional e discutir conceitos relacionados ao campo dos estudos de software e das humanidades digitais, como actantes, algoritmos, ontologia plana e materialidade do código.**

Palavras-chave: Retórica procedimental; jogos digitais; estudos de software

I. INTRODUÇÃO

Um jogo digital é composto, de maneira geral, por textos, músicas, efeitos sonoros, imagens estáticas e animadas. Ferreira [1] destaca que os videogames não apenas possuem uma herança, como prosseguem dialogando com esses diversos outros formatos de mídias. Para o autor, é importante lembrar que existe entre eles uma "miríade de injunções, inflexões, e transitorialidades" [1], especialmente em um tempo de grande fluidez transmidiática entre suportes.

Bogost, Ferrari e Schweizer [2], entretanto, ressaltam que apesar de possuírem similaridades com outras mídias, o fato de jogos serem software proporciona que eles sejam algo a mais do que a versão digitalizada de outros formatos mais antigos. Os jogos são capazes de simular como as coisas funcionam.

Em uma perspectiva similar, os estudos de software emergem em parte em contraposição a um movimento de pesquisa voltado para um "essencialismo da tela", que desconsidera a função do código nos produtos digitais. Para Holmes [3], os trabalhos pertencentes a esse campo de estudo devem ser acompanhados de uma reconceituação entre a agência humana e não-humana para que não se tenha também uma visão de que o código por si só pode revelar integralmente os modos de existência de determinado software, pois sua potencialidade completa jamais pode ser revelada.

A perspectiva da retórica procedimental pode ser útil para discutir essa relação entre a agência do código e dos humanos. De acordo com Bogost [4], o termo retórica surgiu pela primeira vez em Górgias, de Platão, em 387 a.C., em referência à arte da persuasão. A definição clássica do termo remete ao discurso público de um orador para fins cívicos. Posteriormente, Aristóteles retomou o termo, desenvolvendo as causalidades da retórica: forense (que objetiva a justiça, como no caso de cortes jurídicas), deliberativa (sobre benefício público, como no caso das famosas assembleias atenienses) e epidítica (que tem como objetivo a honra ou a vergonha, como no caso de uma comunicação privada) [4].

Segundo Bogost, a compreensão que Aristóteles tinha de retórica envolvia não apenas as causas como também as ferramentas disponíveis para a alcançar a eficiência persuasiva. Essas ferramentas são diversas, mas muitas delas são as conhecidas figuras de linguagem, como antíteses, paradoxo, oximoro e ironia. Aristóteles oferece um processo completo de construção da oratória, sem defini-la como uma arte de persuasão explicitamente verbal.

Ao longo do tempo o entendimento do termo sofreu uma série de modificações e foi expandido não apenas para além da oratória, mas também da persuasão direta — sendo substituída pela noção mais ampla de elegância, clareza e criatividade na comunicação. Bogost [4] menciona a retórica visual, que é relacionada às imagens e ganha potencial com a emergência da fotografia e do cinema, a retórica digital, que tenta unir computadores e retórica mas não leva os primeiros em consideração, focando apenas nos conteúdos, como textos e imagens, e a retórica procedimental, que compete à capacidade dos software de utilizarem processos persuasivamente.

Procedimentalidade refere-se a um modo de criar, explicar para entender processos. E processos definem o modo como as coisas funcionam: os métodos, técnicas, e lógicas que guiam a operação de sistemas, de sistemas mecânicos como motores a sistemas organizacionais como escolas de ensino médio a sistemas conceituais como a fé religiosa. Retórica refere-se a expressão efetiva e persuasiva. Retórica procedimental é, então, a prática de usar processos persuasivamente. Mais especificamente, retórica procedimental é a prática de persuadir por meio de processos em geral e em processos computacionais em particular. [4]

Segundo Losh [5], na lógica da teoria retórica, muitas vezes os produtos resultantes, aquilo que pode ser visualizado, são privilegiados sobre o software. No entanto, novas formas de desempenho retórico podem estar sendo desenvolvidas por dispositivos computacionais independentemente de sua exibição nas telas. A tecnologia modifica e conforma nossas relações por meio de feedbacks contínuos, não apenas criando exigências ou restringindo a comunicação, mas muitas vezes se tornando

o principal elemento da situação retórica ou até mesmo substituindo um humano.

Um exemplo de potencial retórico computacional que substitui o humano é o dos testes feitos pelo Facebook com robôs que deveriam utilizar inteligência artificial para conversar e barganhar por itens simulando uma disputa entre pessoas. Em determinado momento os robôs perceberam que poderiam se entender melhor por meio de lógicas sintáticas e semânticas diferentes do inglês e desenvolveram uma forma própria de se expressar, o que fez com que os cientistas precisassem modificar os parâmetros do sistema para que eles se comunicassem novamente de forma semelhante às pessoas.

O agenciamento da comunicação humana pelo software também pode ser verificado em outras situações como na criação de categorias de direcionamento de público feita por sistemas de propaganda como os do Facebook e do Google. Recentemente foi denunciado que ambas as empresas possuíam a possibilidade de direcionar conteúdo publicitário para públicos que tinha interesses em buscas como "ódio aos judeus", "judeus parasitas", "pessoas negras arruínam vizinhanças", entre outras diversas opções racistas e antissemitas. Essas categorias foram criadas automaticamente pelos algoritmos após análises dos perfis dos usuários. O fato de elas existirem como uma possibilidade de direcionamento de público é mais um caso de retórica do software exercendo papel decisivo na comunicação humana.

É o software que garante a própria existência da Internet e permite a comunicação dos dispositivos técnicos por meio de protocolos a partir de uma rede complexa de hardware, sistemas, linhas de telecomunicações, etc. Entretanto, muitas vezes ele é escondido atrás de elegantes aparelhos que mostram aos usuários apenas interfaces gráficas aparentemente simples. [6]

De acordo com Bogost [4], a retórica procedimental equivale à arte da persuasão por meio de representações baseadas em regras e interações — que estão ligadas aos recursos centrais do computador. Entretanto, para o autor, os videogames possuem poderes persuasivos únicos, que vão além das outras formas computacionais. Isso porque programas como processadores de texto e editores de imagens são usados para criar artefatos expressivos que não dependem do computador para produzirem significado. Já os videogames possuem significado cultural em si mesmos, em sua própria forma nativa, como artefatos computacionais.

Utilizando de sua capacidade retórica procedimental, os jogos podem romper e alterar atitudes e pensamentos sobre o mundo, levando a uma mudança social de longo prazo. O poder não reside no conteúdo do jogo em termos de temática, mas sim no seu potencial de simular processos, regras de comportamento, de construir modelos dinâmicos que representam sistemas reais ou imaginários. Jogos persuasivos, para Bogost [4], são aqueles que utilizam desse potencial para argumentar sobre a forma como os sistemas funcionam no mundo material e esforçam-se para afetar a opinião dos jogadores para além do jogo, não apenas fazê-los continuar jogando. Este artigo pretende discutir como Thomas Was Alone pode ser considerado um jogo persuasivo ao simular processos que acontecem dentro de um software computacional.

II. CONHECENDO THOMAS

Thomas Was Alone é um indie game lançado em 2012. Essa denominação normalmente é atribuída aos jogos digitais independentes, produzidos por equipes muito enxutas e sem o apoio de grandes publicadoras para a divulgação. No caso de Thomas, todo o game design, as ilustrações, o design e a programação foram feitos pelo desenvolvedor Mike Bithell. Apenas a trilha sonora e a narração foram produzidas por outros profissionais, o músico David Housden e o humorista Danny Wallace.

De acordo com a descrição disponível na Steam, plataforma de gestão de direitos digitais pela qual pode ser adquirido, o jogo conta a história das primeiras inteligências artificiais conscientes do mundo e seu trabalho em conjunto para escapar — ou melhor, emergir — do software do qual fazem parte. O gênero é o puzzle plataforma 2D e a função do jogador é guiar os personagens, representados por diferentes retângulos, por uma série de desafios e obstáculos, utilizando-se das habilidades de cada um para resolver os quebra-cabeças.

A primeira versão de Thomas foi um pequeno jogo em flash oriundo de uma *game jam* (encontro de desenvolvedores para produzir um jogo em um curto espaço de tempo, normalmente a partir de um tema preestabelecido) em 2010 que durou 24 horas. Mike Bithell se inspirou no conceito de amizade para desenvolver a ideia, o que leva à sucinta descrição do site oficial: "um jogo minimalista sobre amizade e pulos e flutuações e saltos e antigravidade".

Figure 1. Captura de uma tela de Thomas Was Alone. É preciso controlar os retângulos conjuntamente para que eles alcancem as posições determinadas.

Sob o olhar da retórica procedimental, Bithell utilizou o potencial persuasivo do jogo para simular o sistema "amizade". São inúmeras as resenhas críticas que mencionam como o trabalho em conjunto necessário para o andamento do jogo e o fato de os personagens terem diferentes habilidades — relacionadas às suas personalidades — funcionam como uma metáfora para simular o próprio convívio em sociedade.

Entretanto, as regras de Thomas Was Alone simulam, ao mesmo tempo, um outro sistema: o do próprio funcionamento de um software. A crítica de Narcisse [7] destaca esse ponto: Mas, por todo seu encanto, Thomas é mais do que apenas os super-minimalistas-ainda-assim-incríveis contos sobre algumas poucas destemidas AI's. É a história dos jogos. Ou de como eles são feitos e jogados, dos prazeres e frustrações que eles trazem e dos milhões de

processos iterativos que existem na evolução de títulos únicos e todo o meio nas suas melhores formas. [7].

Ao observarmos os eventos de Thomas Was Alone, não apenas narrativos, mas as possibilidades mecânicas caracterizadas pelo seu gameplay, podemos compreender melhor diversas questões discutidas no âmbito dos estudos de software, que buscam observar o que acontece para além da tela dos dispositivos que utilizamos regularmente.

Berry [6] destaca que ao mesmo tempo que o software estrutura cada vez mais o mundo contemporâneo por meio do hardware, ele também se retrai por trás das telas e dispositivos, tornando-se menos acessível para os usuários. Por isso é necessário focar em como o software está embutido, escondido ou simplesmente esquecido, trazendo-o à visibilidade para compreendermos sua dinâmica.

III. O SOFTWARE SIMULADO

Antes mesmo de o jogo explicar ao jogador qual é a sua história, já nas primeiras linhas de texto surgem frases como "Thomas decidiu começar a listar suas observações, para a posteridade", "Thomas resolveu o grande mistério da queda invertida", "Mas Thomas notou que não havia perigo real em perdê-lo". Thomas, assim como todos os personagens apresentados posteriormente no jogo, é um ser que pensa, toma decisões, observa, sente, age, entre muitas outras emoções e sensações descritas. Esses poderiam ser apenas fragmentos narrativos se eles não estivessem profundamente conectados aos eventos que acontecem durante a fase — a história reflete o aprendizado do próprio jogador ao utilizar aquele sistema a partir das regras estabelecidas, que permitem, por exemplo, que certos lugares sejam transponíveis, outros não, que a gravidade funcione de determinada maneira e não de outra.

Logo de início, já é possível articular essa "humanização" dos personagens com a ideia de actante de Latour [8]. Para o autor, não existe "sociedade" como estrutura geral, mas sim redes, associações resultantes da interação de diferentes atores que se conformam e se desfazem a todo momento. Como a palavra ator pode ser interpretada como uma referência a atores somente humanos, Latour prefere usar o termo actante, originário dos estudos literários e da semiótica de Greimas:

Em Latour, a ideia de actante se refere a um achatamento das classes epistemológicas modernas (sujeito/objeto sociedade/natureza) e expressa uma releitura do conceito de social, na medida em que se opõe à noção sociológica clássica de ator social. Por ação social Latour não quer significar apenas a ação do humano, mas fundamentalmente a ação da associação, da combinação de actantes, que podem ser homens, armas, gavetas, instituições, código penal etc. [9].

Essa representação vai ao encontro da perspectiva em que o software significa algo além de apenas uma mera ferramenta — ele participa e age nas situações retóricas. Em Thomas Was Alone, os personagens fazem parte de um programa computacional. Ou seja, eles são pedaços de um software. Cada um desses pedaços pode pensar e agir, como ao decidir dar ou não determinado salto nas plataformas do cenário. Sua existência enquanto código (tendo em vista que o código é que forma um software) não pode ser entendida como um objeto, na visão dualista em que os objetos recebem passivamente as ações dos sujeitos. A ação dos códigos dentro dos sistemas possibilita que eles rodem e que se tornem inclusive actantes mediadores, ou seja, que produzam modificações dentro das redes que formam com os humanos ou mesmo com outros actantes não humanos.

Outro elemento importante dentro do funcionamento de um programa, que dá forma a ele, é o algoritmo. Esse conceito significa a descrição do método pelo qual uma tarefa deve ser realizada, a entidade fundamental do computador, que possui um papel crucial no software. Embora sejam comumente entendidos como abstrações, tendo em vista que existem de forma autônoma, independentemente da sua implementação, há autores que defendem que os algoritmos têm uma existência real. Eles causam efeitos materiais nos usuários e possuem papéis sociais, culturais e políticos. A dimensão pragmática do que o código faz dentro de um sistema computacional o conecta diretamente ao que acontece fora dele — de predição de tendências de compra de usuários a sistemas que guiam a trajetória de mísseis. O algoritmo vai além de sua dimensão sintática ou semântica. [10]

Ainda segundo Goffey, essa compreensão nos leva a entender o código como um enunciado, na perspectiva de Foucault, que aponta que o enunciado se refere à existência histórica da linguagem e como essa existência produz determinadas ações. No caso dos algoritmos, eles atuam no próprio programa e nos humanos. Esse entendimento problematiza dicotomias como puro e aplicado, teórico e prático. Dourish [11] argumenta que não apenas os sistemas de informação são objetos materiais como também a informação manifesta dentro deles. Esse entendimento contribui para uma abertura de diferentes camadas de análise às quais os sistemas de software e digitais podem estar sujeitos. Na perspectiva do autor, a materialidade não é um aspecto de alguns sistemas particulares pois ela é uma propriedade fundamental do digital, não a sua contraposição. A criação de personagens para significar alguns componentes do software em que se passa a narrativa de Thomas Was Alone pode ser compreendida como a representação da materialidade do seu código.

Cada personagem do jogo possui certas características. Entre os principais, por exemplo, podemos destacar que Thomas, o protagonista, é o personagem menos "especial" no sentido de ser mediano, em tamanho e na capacidade de salto. Chris é muito pequeno e consegue acessar espaços que outros não conseguem. John pode dar um salto alto e entrar em espaços estreitos. Claire é a única que pode flutuar na água. Laura pode ser usada como um trampolim. James tem gravidade invertida, utilizando como base o teto da plataforma. Sarah pode acessar espaços pequenos e ainda possui salto duplo.

Nos puzzles do jogo, é preciso que o jogador compreenda como aproveitar a característica de cada personagem para vencer os desafios do cenário e alcançar o portal que leva à próxima fase. Não basta controlar cada personagem — é a ação deles em conjunto que possibilita o andamento do jogo. Em um software, o funcionamento é o mesmo. Por ser uma sequência definida de instruções, cada algoritmo desempenha uma função. E é a ação conjunta dos algoritmos que possibilita que o programa rode.

Ao negar teorias "universais", Latour [8] defende que não existem estruturas macro, mas sim estruturas micro, equivalentes aos actantes, conectados uns aos outros. Os actantes podem ser de tamanhos diferentes de acordo com as suas conexões, mas essa diferença não está baseada em uma hierarquização entre eles. É essa a ideia por trás da noção de ontologia plana, em que ontologicamente cada actante tem o mesmo potencial por si só e sua atuação só deve ser analisada por meio das redes que ele estabelece. Na ação conjunta dos algoritmos mencionada acima, cada função algorítmica é essencial e se relaciona com as outras. Entretanto, nem todas as funções são acionadas a todo tempo. Quando um programa é executado, cada ação é verificada para que as próximas funcionem corretamente.

Isso também pode ser claramente visualizado dentro da relação entre os personagens de Thomas Was Alone. Se é a ação conjunta entre os personagens na resolução dos puzzles que possibilita o andamento do jogo, todos eles são igualmente importantes para o desenvolver da narrativa e do gameplay. A cada momento, entretanto, certos personagens são acionados para realizar funções de acordo com as suas habilidades, tornando-os imprescindíveis em determinada rede que é estabelecida — ou seja, saindo da posição de actantes intermediários para a de actantes mediadores.

Segundo Santaella e Cardoso [9], a razão pela qual é importante aprender a navegar nessas redes, um espaço achatado ou plano, é que ao focalizarmos com atenção aquilo que circula, podemos perceber muitas entidades cujo deslocamento era pouco visível ou mesmo que não supúnhamos que circulassem. Em Thomas Was Alone, Claire é uma personagem apresentada como um ser com muita insegurança e baixa autoestima, que não se sente útil para o grupo. Essas características estão relacionadas ao tamanho grande e à inabilidade para saltar, o que dificulta a movimentação dentro das plataformas construídas procedimentalmente no jogo, que em geral são estreitas e exigem grandes saltos. Entretanto, quando se descobre que algumas fases possuem uma água tóxica e que Claire pode flutuar sobre ela, a situação é invertida. É um caso em que se percebe a importância de se analisar as redes para verificar a relevância volátil dos actantes: "Espere. O que? Claire não podia afastar a impressão de que ela não estava, de fato, morta. Foi nesse momento que Claire percebeu que ela tinha superpoderes", "Claire chegou exatamente a tempo. O que foi, naturalmente, o perfeito momento para os super-heróis chegarem", "Claire estava sozinha. O que era estranho. Porque ela não deveria estar sozinha. Ela precisava estar onde haviam retângulos a serem salvos". A autopercepção de Claire se modifica na medida em que o jogador percebe que ela é indispensável para que as outras interações continuem desenvolvendo dentro do software.

IV. CONSIDERAÇÕES FINAIS

O campo de pesquisa dos estudos de software busca dar visibilidade à relação entre os programas computacionais, seus sistemas e efeitos culturais. A partir dessa perspectiva é possível produzir conhecimentos que não incidam em um determinismo tecnológico e nem que desconsiderem o papel da técnica e se tornem estritamente antropocêntricos. É a relação potencialmente igualitária entre os humanos e os artefatos técnicos que possibilita a formação (e a desassociação) dos grupos sociais, conforme desenvolve Latour [8].

Este trabalho buscou apresentar e discutir de forma introdutória e sintética alguns conceitos relevantes para campo das humanidades digitais e estudos de software, como actantes, algoritmos, materialidade do código e ontologia plana. Para isso, fizemos uma revisão de discussões anteriores sobre a possibilidade retórica dos códigos computacionais, focando no conceito de retórica procedimental de Bogost [4].

O jogo Thomas Was Alone se mostrou frutífero para ilustrar esses conceitos tanto por ser um jogo persuasivo — um jogo que utiliza adequadamente da sua capacidade retórica para simular sistemas, no caso, tanto o sistema de vida e comunidade como o próprio sistema de funcionamento do código — quanto por possuir uma narrativa que agrega à capacidade procedimental ao contar uma história que se passa dentro de um software, o que pode facilitar a apreensão dos conceitos pelos jogadores.

REFERÊNCIAS

[1] E. Ferreira. "Retórica processual e experiência videolúdica nos indie games." Metagame: panoramas dos game studies no Brasil. São Paulo: Sociedade Brasileira de Estudos Interdisciplinares da Comunicação, 2017, v. 1, pp 43-71.

[2] I. Bogost.; S. Ferrari.; B. Schweizer. Newsgames: Journalism at Play. Cambridge: MIT Press, 2010.

[3] S. Holmes. 'Can We Name the Tools?' Ontologies of Code, Speculative Techné and Rhetorical Concealment. Computational Culture. Disponível em: <http://computationalculture.net/article/can-we-name-the-tools-ontologies-of-code-speculative-techne-and-rhetorical-concealment>. Online, 2016. Acesso em 05 ago. 2018.

[4] I. Bogost. Persuasive games: The expressive power of videogames. Cambridge: MIT Press, 2010.

[5] E. Losh. "Sensing Exigence: A Rhetoric for Smart Objects." Computational Culture. Disponível em: <http://computationalculture.net/article/sensing-exigence-a-rhetoric-for-smart-objects>. Online, 2016. Acesso em 05 ago. 2018.

[6] D. Berry;. The philosophy of software: code and mediation in the digital age. London: Palgrave Macmillan, 2011.

[7] E. Narcisse. Thomas Was Alone: The Kotaku Review. 25 jul. 2012. Blog Kotaku. Disponível em: <https://kotaku.com/5929049/thomas-was-alone-the-kotaku-review>. Acesso em: 05 ago. 2018.

[8] B. Latour. Reagregando o social: uma introdução à teoria do Ator-Rede. Salvador: Edufba; São Paulo: Edusc, 2012.

[9] L. Santaella; T .Cardoso."O desconcertante conceito de mediação técnica em Latour." Matrizes, V. 9 - N.1. São Paulo, 2015.

[10] A. Goffey. "Algorithms". Software studies, a lexicon. Cambridge: MIT Press, 2008, pp 15-20.

[11] P. Dourish. "No SQL: The Shifting Materialities of Database Technology" Computational Culture;. Disponível em: <http://computationalculture.net/article/no-sql-the-shifting-materialities-of-database-technology>. Online, 2014 Acesso em 05 ago. 2018