

**Fernando Emboaba de
Camargo**
UNICAMP
Campinas, SP, Brasil

José Fornari
UNICAMP
Campinas, SP, Brasil

PROBLEMÁTICAS ACERCA DA PAISAGEM SONORA NOS GAMES

ISSUES AROUND SOUNDSCAPE IN GAMES

RESUMO

Desenvolver uma paisagem sonora nos games é contribuir para a apresentação de um mundo virtual passível de interação, provendo a inteligibilidade da narrativa ao interagente. Sob essa premissa, classificaremos a paisagem sonora em duas perspectivas: paisagem sonora tópica e atópica. A primeira é resultado direto das ações sugeridas na narrativa, normalmente visualizadas ao interagente; a segunda, resultado indireto da narrativa, com ações não visualizadas ao interagente. Paralelo a isso, analisaremos games em que se perceberam equívocos cujas soluções partiram de experimentos desenvolvidos no software Pure-Data-extended.

Palavras-chave: paisagem sonora; games; trilha sonora.

ABSTRACT/ RESUMEN

Developing a soundscape in games contribute to the presentation of a virtual world capable of interaction, providing intelligibility of the narrative to the player. In this article, we will classify the soundscape in two perspectives: topical and atopic soundscapes. The first is a direct result of actions suggested in the narrative, normally visible to the player; the second, indirect result of the narrative, with actions invisible by the player. Parallel to this, we will analyze games in which mistakes were realized whose solutions where solved by experiments developed in the Pd-extended software.

Keywords / Palabras Clave: soundscape; games; soundtrack.

Recebido: 21/05/2020 / Aprovado: 18/10/2020

Como citar: CAMARGO, F. E.; FORNARI, J. Problemáticas acerca da Paisagem Sonora nos Games. Revista GEMInIS, v. 11, n. 3, pp. 185-210, set./dez. 2020.

Direito autoral: Este artigo está licenciado sob os termos da Licença Creative Commons-Atribuição 3.0 Internacional.

1 Introdução

Com uma indústria que cresce exponencialmente desde a década de 70 (DONOVAN, 2010), os *games*, também conhecidos como jogos eletrônicos, são mídias que disponibilizam a um interagente¹ (*player* ou jogador) acesso interativo a um mundo virtual gerado por algoritmos (CORREA, 2014). Essas mídias, em geral, são compostas por três grandes áreas: a programação - geradora dos algoritmos que regem o *game*; as artes visuais - responsáveis pelos aspectos gráficos na exposição do mundo virtual; e a trilha sonora - responsável pelos aspectos sonoros na significação do mundo virtual.

A trilha sonora, definida como todo som de uma mídia (CARRASCO, 1993), desenvolveu-se com o avanço da tecnologia, por vezes transcendendo as limitações técnicas com soluções únicas para cada *game*. No ano de 2020, dispõe-se de muitos recursos para o desenvolvimento de uma trilha sonora, cujas pesquisas científicas ainda são incipientes para compreendê-los (COOLINS, 2008b).

A qualidade de uma trilha sonora para *games* está relacionada ao seu grau de coesão na invenção do mundo virtual disponível pelo *game* ao interagente (JUUL, 1999). Seu papel, nessa composição, é atuar sobre todos os outros componentes de um *game*, ou seja, em relação ao gráfico, à programação e à narrativa. Nesta pesquisa, voltamo-nos para o grupo de sons da trilha sonora que pretende significar os atributos dos objetos e seres virtuais dispostos no mundo virtual de um *game*, nominado paisagem sonora². Para tanto, nos pautaremos em uma nomenclatura que subdivide a paisagem sonora em dois tipos: **paisagem sonora tópica**³ e **paisagem sonora atópica**⁴ – terminologia desenvolvida na tese de doutorado *Interatividade e narratividade sonora nos games* (CAMARGO, 2018). Cada um dos tipos exerce funções distintas: a paisagem sonora tópica, para sons com fonte sonora visível ao interagente, e a paisagem sonora atópica, para sons com fonte sonora invisível ao interagente, ou seja, ausente na visão do ecrã. Após exposta esta terminologia, traçaremos quatro fases que integram todo processo do desenvolvimento deste trabalho, enumeradas a seguir.

¹ Termo elaborado por Elisa Corre referindo-se a um "sujeito social e cognitivo que busca informação com vistas a solucionar questões de ordem pessoal, profissional ou acadêmica..." (CORREA, 2014, p.37).

² *A soundscape is defined as the overall sonic environment of an area (PORTEOUS; MASTIN, 1985) or as an environment of sound with emphasis on the way the sound is perceived and understood by individual or by a society (TRUAX, 2001), which can be a sound or a combination of sounds that creates an environment or an atmosphere (CHANDRASEKERA, 2015, p. 1004).*

³ A palavra **tópico** baseia-se no termo grego *tópos* que significa **lugar** (ILLUSIE, 2004).

⁴ A palavra **atópico** baseia-se no termo grego *tópos*, porém com o prefixo de negação, significando assim **sem lugar** (ILLUSIE, 2004).

1ª Fase) Levantaremos cinco *games* (*Hearthstone: Heroes of Warcraft*⁵, *This War of Mine*⁶, *Clash of Clans*⁷, *Don't Starve*⁸) com intuito de analisar o modelo em que a paisagem sonora foi aplicada, tanto naqueles sons com fonte sonora atribuída a um elemento gráfico (paisagem sonora tópica), quanto naqueles cuja fonte sonora seja imprecisa (paisagem sonora atópica).

2ª Fase) Após o levantamento das análises da 1ª Fase, apontaremos algumas fragilidades acerca da paisagem sonora atópica que precisam ser pensadas, com cautela, tanto pela equipe de programação quanto pelo *sound designer*. A falta desse olhar para a *paisagem sonora atópica* pode transparecer o programa (algoritmos que regem as variáveis de um *game*) ao interagente, e então comprometer a verossimilhança do mundo virtual.

3ª Fase) Considerando os dados expressos na 2ª Fase, desenvolveremos experimentos no *software Pd-extended*⁹ versão 0.43.4¹⁰ (interface visual para programação multimídia) para validar nossos argumentos e apresentar soluções empíricas acerca da paisagem sonora atópica (FARNELL, 2010).

4ª Fase) Por fim, exibiremos três *softwares* (*middlewares*) que concedem ferramentas auxiliares para composição e aplicação da paisagem sonora nos *games*, a saber: *Wwise*¹¹, *FMOD*¹², *Pd-extended*. Além desses *softwares*, é possível desenvolver algoritmos descritos em um documento de texto, imagem e vídeo, para o programador incorporá-los na *engine* (*software* responsável em unir e executar todas as mídias de um *game*).

⁵ O *game* *Hearthstone: Heroes of Warcraft*, desenvolvido e publicado pela *Blizzard Entertainment*, foi lançado em 2014 e suporta as plataformas *Windows*, *MAC*, *Android*, *IOS*. Disponível em <<https://www.gamespot.com/hearthstone-heroes-of-warcraft/>>, acessado no dia 01/08/2017.

⁶ O *game* *This War of Mine*, desenvolvido pela *11 bit studios*, foi lançado em 2014 e suporta as plataformas *Windows/ OS X/ Linux/PS4/Xbox One/ Android/ IOS*. Disponível em <<https://www.gamespot.com/this-war-of-mine/>>, acessado no dia 01/08/2017.

⁷ O *game* *Clash of Clans*, desenvolvido e publicado pela *Supercell*, foi lançado em 2012 e suporta as plataformas *Android/ IOS*. Disponível em <<https://www.gamespot.com/sponsored-clash-of-clans/>>, acessado no dia 01/08/2017.

⁸ O *game* *Don't Starve*, desenvolvido pelas empresas *Klei Entertainment*, *505 Games*, *BlitWorks*, *Capybara Games*, foi lançado em 2013 e suporta as plataformas *Windows/ OS X/ Linux/ PSVita/ PS3/ PS4/ Xbox One/ Wii U/ IOS*. Disponível em <<https://www.gamespot.com/dont-starve/>>, acessado no dia 05/05/2017.

⁹ O *software* *Pure-Data extended* (atualmente descontinuado) foi lançado em 2013 e suporta as plataformas *Windows/ Linux/ OS X*. Disponível em <<https://puredata.info/downloads/pd-extended>>, acessado no dia 01/08/2017.

¹⁰ Disponível em <<https://puredata.info/downloads/pd-extended/releases/0.43.4>>, acessado no dia 01/08/2017.

¹¹ O *software* *Wwise* (*Wave Works Interactive Sound Engine*) foi lançado em 2016 pela *Audiokinetic* e suporta as plataformas *Android/ AndroidTv/ Linux/ IOS/ MAC/ TvOS/ Windows/ PSVita/ Ps3/ Ps4/ PsVR/ WiiU/ Nintendo Switch/ Xbox 360/ Xbox One/ Oculus Rift/ Vive/ Google Cardboard/ Daydream/ GearVr/ Microsoft HoloLens*. Disponível em <<https://www.audiokinetic.com/products/wwise/>>, acessado no dia 01/08/2017.

¹² O *software* *FMOD* foi desenvolvido pela *Firelight Technologies* e suporta as plataformas *Windows/MAC/ IOS/ Linux/ Android/ BlackBerry/ Wii/ Wii U/ 3DS/ Xbox/ Xbox360/ PS2/ PS3/ PSVita/ Google Native Client*.

2 A imersão na paisagem sonora virtual

A trilha sonora - entendida como todos os sons inseridos em uma mídia - é central para garantir a imersão em um mundo virtual. Parte-se, por conseguinte, de que a imersão seja a propriedade referente à experiência de um indivíduo em engajar-se cognitivamente e sensorialmente no ambiente virtual controlado pelo interagente. No caso dos *games*, elaboramos a atuação da trilha sonora em três segmentos: a trilha musical; sons responsoriais à interação; a paisagem sonora.

Por um lado, no âmbito da trilha musical, enfatiza-se o afeto¹³ do interagente perante as diversas situações decorrentes do mundo ficcional. Os sons de interação, por outro lado, respondendo ao comando efetuado pelo interagente, traçam um caminho didático e instantâneo entre as mecânicas do *game* e o desempenho de quem o manipula. Visto isso, resta estabelecer a paisagem sonora como, tão somente, a parte responsável por materializar o som do mundo virtual a fim de torná-lo consistente. Assim, afirma Gadamer:

A atração do jogo, o fascínio que exerce, consiste precisamente no fato de que o jogo domina os jogadores. [...] O assunto real do jogo não é o jogador, mas sim o próprio jogo. O que mantém o jogador envolto em sua magia, o que o motiva a jogar e o que o mantém ali é o próprio jogo eletrônico (GADAMER, 1989, p. 106 *apud* AARSETH, p.130, 2007).

Um *game* que atinge um grau satisfatório da paisagem sonora contribui para a intensificação da imersão do interagente, pois ele percebe tanto cognitivamente quanto sensorialmente que seu personagem pertence àquele mundo virtual, que existe, por si, pronto para ser explorado. Todavia, atingir esse grau de sofisticação é laborioso, pois os erros cometidos na invenção e aplicação de uma paisagem sonora são recorrentes. Quando uma paisagem sonora não é tratada cautelosamente, pode transparecer os algoritmos do *game* (descrito no exemplo de *Clash of Clans*). Ocorre, dessa forma, uma incoerência na apresentação do mundo virtual, que pode acarretar uma quebra de imersão no interagente, já que ele percebe os limites do algoritmo que gerencia o disparo dos sons, desmistificando aquele mundo virtual – exceções nas quais os desenvolvedores buscam intencionalmente o contato direto do interagente ao programa do *game*.

¹³ Afeto é aqui entendido como sentimento motivador.

3 Os dois tipos de paisagem sonora em games

Nossa classificação acerca da paisagem sonora é categorizada em dois tipos: a paisagem sonora tópica e a paisagem sonora atópica, ambos os termos elaborados a partir dos conceitos de “visualizado” e “acusmático”, de Michael Chion (CHION, 2008, p. 61)¹⁴.

A paisagem sonora tópica é o grupo de sons associados aos elementos gráficos materializados no ecrã, visíveis para o interagente com sua fonte sonora definida, tornando natural o reconhecimento de suas características físicas, bem como a causa de sua existência. Por exemplo, em um mundo virtual, isso ocorre quando o interagente observa um cão que late. O som do latido está diretamente relacionado aos elementos gráficos que descrevem esse cão no ambiente virtual do *game*. Além disso, o cão é acessado visualmente e auditivamente pelo interagente.

Paralelamente às ações dos personagens em um mundo virtual, ocorrem diversas outras ações que não são claramente identificadas e muito menos controladas pelo personagem do interagente. No mundo real, quando estamos em uma floresta tropical, por exemplo, escutamos o som de animais, movimentação nos galhos e árvores, pássaros, vento entre as árvores, dentre diversos outros sons. Não obstante, não identificamos precisamente a fonte sonora, mas, ao mesmo tempo, não nos assustamos, pois são movimentações de elementos pertencentes à flora e fauna daquela localidade. A inserção desse grupo de sons em um *game* é extremamente custosa nos quesitos de tamanho e processamento, comprometendo sua otimização¹⁵. Por conta disso, cada *game* apresenta formas de suprir esse grupo de sons, caso seja relevante mantê-los, que é aqui, nominado como paisagem sonora atópica.

3.1 A paisagem sonora tópica na construção de um *game*

A paisagem sonora tópica descreve os elementos gráficos de um *game* concedendo-lhes atributos por meio do som. Para exemplificar esse grupo de sons, utilizamos alguns exemplos retirados do *game Don't Starve*, porque é conduzido tão somente por esses sons instalados em elementos gráficos passíveis de interação no mundo virtual que ele mesmo compõe.

Em *Don't Starve* - lançado em 2013 e atualmente disponível para as plataformas: *Windows/ OS X/ Linux/ PSVita/ PS3/ PS4/ Xbox One/ Wii U/ IOS* - o interagente extrai os recursos (madeira, pedra, grama, animais etc.) dispostos no mundo virtual para sobreviver. Esses recursos pertencem a

¹⁴ Visualizado significa “associar imediatamente o som a uma imagem precisa”, e acusmático significa “o segredo da causa e do seu aspecto, antes de a revelar”, ou seja, o som sem a associação com uma imagem que gera um suspense (CHION, 2008, p. 61).

¹⁵ A otimização é a ação de retirar o excesso de informação de um *software*, seja na programação dos códigos, seja nos gráficos, seja no som. Isso para que exija menos armazenamento e processamento instalação e executar do *software*.

ecossistemas específicos como: bosque, pântano, floresta, rochedo, pasto, caverna. Em cada um desses ecossistemas, predomina uma cor de piso. No bosque, essa cor é verde; no pântano, roxo; na floresta, marrom; no rochedo, bege; no pasto, amarelo; e na caverna, cinza. Para aprimorar essa descrição, quando o personagem caminha sobre um ecossistema, os recursos, que ali estão, emitem sons. Nesse caso, eles são atribuídos precisamente aos seus devidos elementos gráficos, logo, visíveis e passíveis de interação pelo interagente. Em outras palavras, por meio do som pertencente à paisagem sonora tópica, o interagente identifica com precisão a direção e o posicionamento dos recursos, bem como os seus atributos, servindo de guia, mesmo sem esse recurso visível na tela. Esta referência direcional, por meio do som, garante altos níveis de desempenho no *game*, já que ultrapassa os limites da visibilidade proporcionada pela câmera, concedendo soluções para sobrevivência no *game*, principalmente, no período da noite, do mundo virtual em questão, quando a visão é turva e ainda mais limitada.

No bosque, a paisagem sonora tópica é constituída de sons característicos de abelhas, coelhos, pavões, sapos, que são elementos interativos de que o personagem consegue extrair diversos alimentos. No *game*, geram-se algoritmos que regem os cálculos na relação entre proximidade do personagem e recursos. Por consequência, alteram-se a intensidade do som sobrevivendo deles e sua posição quanto ao personagem.

Além de descrever os elementos gráficos de um *game*, a paisagem sonora tópica arbitra sobre esses elementos determinando suas características. Por exemplo, ainda no *game Don't Starve*, encontram-se árvores de madeira e árvores de pedra, apenas com modificação na cor. Porém, a grande diferença está no som da animação que se emite quando o personagem extrai esse recurso. Para árvore de madeira, usa-se um machado cujo som representa uma colisão contra uma tábua de madeira; já na árvore de pedra, usa-se uma picareta cujo som representa uma colisão contra uma pedra. Ou seja, mesmo a árvore de pedra exibindo uma aparência de uma árvore padrão - por vezes não diferenciamos de prontidão os dois tipos de árvores durante o *gameplay*¹⁶- o som determina para o interagente que aquele objeto é de pedra e não de madeira.

A partir desse exemplo, levantamos as individualidades do som no mundo virtual com fonte sonora determinada, por nós classificada como paisagem sonora tópica. Essa identificação é importante, já que agrupa aqueles sons que necessitam de ser espacializados e caracterizados, quer

¹⁶ *Gameplay* são as possibilidades sensitivas e mecânicas com que o interagente atinge o mundo fictício de um *game*. Vale esclarecer que *gameplay* consiste no fluxo do jogo e ocorre por meio da interação em função de mecânicas e regras do jogo, enquanto a jogabilidade corresponde à facilidade de jogar o jogo, à qualidade e à experiência do interagente (AGUIAR; BATTAIOLA, 2016).

por existirem enquanto elemento agente no mundo virtual, quer por significarem os atributos específicos daquele elemento gráfico.

3.2 A paisagem sonora atópica na construção de um *game*

A paisagem sonora atópica é o grupo de sons invisíveis para o interagente e rarefeitos na interação, já que esse grupo de sons não está acoplado a nenhum elemento gráfico. Sua importância em um *game* é enriquecer a significação de um ambiente sônico, sempre em contraponto com os demais sons, e otimizar a ocupação do espaço e o processamento.

Para desenvolvermos uma paisagem sonora atópica, apoiaremos em três requisitos básicos:

- 1) Enriquecer a significação do ambiente sônico no mundo virtual de um *game*.
- 2) Ser coerente à *diégesis*¹⁷ - narrativa, exposição do mundo ficcional, no caso o mundo virtual dos *games*.
- 3) Dissimular o programa computacional gerador do mundo virtual, caso não seja a intenção.

Em relação ao primeiro requisito, a paisagem sonora atópica deve suprir o silêncio com sons que contribuam para a descrição do ambiente sônico em questão. Por exemplo, no contexto de um *game* cuja narrativa ocorre em uma cidade grande, a paisagem sonora atópica seria formada por sons de buzinas, motores, celulares, vozes, entre outros. Ao escutar esses sons, o interagente não distinguirá precisamente sua fonte, visto que estão mesclados. Além disso, ela inexistente no próprio algoritmo enquanto fonte sonora pareada com um elemento gráfico. No entanto, o interagente percebe esta fonte como verossímil ao mundo real, uma vez que o grupo de sons da paisagem sonora atópica sugere múltiplas ações paralelas ao seu personagem no mundo virtual, caracterizando a descrição do *game*.

O segundo requisito aponta para a coerência na descrição da paisagem sonora atópica no mundo virtual em todas as transformações no decorrer da narrativa. Consoante o mesmo exemplo - a ideia dos sons de uma cidade grande - caso insermos o som de um rugido de leão no mundo virtual, teríamos que o contextualizar, visto que esse rugido não pertence à gama de sons que normalmente se escutaria em uma cidade. Ocorrido isso, são previstos um estranhamento do interagente e uma readaptação dos outros sons em função deste distúrbio na paisagem sonora atópica. Por essa razão, é primordial que compreendamos por completo a narrativa (*diégesis*) do mundo virtual em toda sua dimensão temporal.

¹⁷ “Universo espaço-temporal onde ações ocorrem” [tradução nossa] (COSTE; PIER, 2011).

O terceiro requisito, que atenta para os cuidados na transparência do programa na narrativa, é o mais agravante e recorrente em relação à paisagem sonora atópica, porque compromete a verossimilhança do mundo virtual de um *game*. É nessa categoria que pautamos nossa pesquisa a fim de buscar caminhos para o desenvolvimento otimizado de uma paisagem sonora atópica. Para tanto, iniciaremos nosso discurso apontando, primeiro, a problemática em torno da apropriação de um som no ambiente sônico do mundo real e, segundo, a inserção deste som no mundo virtual. À guisa de exemplo, elegemos o grupo de sons emitidos pelas ondas do mar.

A princípio, as ondas do mar produzem um som ininterrupto que, se ouvido desatentamente, parece autossimilar. Caso o indivíduo escute com atenção, perceberá nuances: variações de intensidade, altura, timbre e duração do som. Este tipo de som é predominantemente aleatório, no entanto, as variações dos seus parâmetros sonoros retêm nossa percepção do significado do que é o mar, dada a riqueza da resultante sonora, mesmo que o indivíduo esteja desatento. Esse tipo de paisagem sonora atópica em um *game* normalmente é apresentado por meio de uma faixa sonora executada em *loop* - repetição contínua do arquivo de som – e ativada quando o personagem do interagente entra em contato com esse ambiente do mundo virtual. Nessa relação, deparamo-nos com uma discrepância entre o mundo real e o mundo virtual. Uma paisagem sonora atópica de um mar percebida por um interagente desatento ao estímulo sonoro parece igual; todavia quando se prontifica a escutá-lo, não somente se certifica de que o som é realmente sempre igual, como percebe o momento exato do *loop* da faixa sonora. Esse fator enfraquece a imersão do interagente no *game*, uma vez que permite o acesso à programação que gerou o mundo virtual. Essa falha, quando irrefletida, compromete a veracidade de toda a narrativa desenvolvida.

Dentre os *games* analisados, sob os três requisitos apresentados no início desta seção, percebemos que a paisagem sonora atópica, corresponsável pela elaboração da consistência do mundo virtual do *game*, é substituída pela trilha musical. Por exemplo, o *game Hearthstone: Heroes of Warcraft*, lançado em 2014 pela *Blizzard Entertainment*, apresenta um mundo ficcional com diversas criaturas e raças fantásticas. Sua narrativa concentra-se no embate entre dois personagens em um jogo de cartas situados em uma taverna. A presença da paisagem sonora atópica é reduzida em sons de conversas ao fundo de sala, por diferentes timbres de vozes incessáveis. Paralelo a esta paisagem sonora atópica, o *game* disponibiliza uma trilha musical diversa e constante com intensidade superior à das conversas. Como consequência disso, a paisagem sonora atópica é percebida apenas nos momentos de silêncio entre uma música e outra. Para ilustrar a explicação, verifique o vídeo de nossa autoria entre dois minutos e quatorze segundos até três minutos e vinte segundos: <https://youtu.be/su_UyxPyMRs>, acessado no dia 15/05/2020.

Em suma, retiramos desse exemplo que a resultante sonora elege um número diminuto de possibilidades na descrição de um ambiente sônico de uma taverna. No entanto, não transparece o *loop* da faixa, pois o som das conversas, ou seja, a paisagem sonora atópica, efetua as transições entre as músicas e demais cenas do *game*, maquiando o momento de transição entre a execução do arquivo de som. Outro ponto para realçar é que a trilha musical contempla equalização e reverberação incoerentes com aquelas dos sons das conversas. Por conseguinte, ela não alcança um resultado verossímil com o mundo real.

Outro exemplo no uso de uma paisagem sonora está no *game This War of Mine*, lançado pela *11 bit Studios* em 2014. A narrativa do mundo virtual demonstra uma cidade destruída pelo caos urbano de uma guerra. O interagente controla civis reclusos em uma casa, onde tentam sobreviver à fome, ao frio e ao caos social. A trilha musical e a paisagem sonora atópica unem-se em uma única faixa sonora. Ou seja, os sons de tiros e bombas são tratados como instrumentos musicais no discurso poético dos outros instrumentos. No total, o arquivo de som tem duração de três minutos, sendo executado em média duas vezes por dia do mundo virtual do *game*, que equivale, por sua vez, a sete minutos no mundo real. O *loop* da trilha musical em conjunto com a paisagem sonora atópica é efetuado após a explosão de uma grande bomba, seguidas de silêncios, assim, tornando as transições entre o final e o início da faixa sonora imperceptíveis durante o *gameplay*. Ambos os casos levantados - *Hearthstone: Heroes of Warcraft* e *This War of Mine* - demonstram soluções que cumprem com os requisitos necessários, coerentes ao mundo virtual exposto, mesmo enfatizando a trilha musical para solucionar os problemas. Assim, é possível suprir as necessidades básicas de uma paisagem sonora atópica. Para complementar a explicação, verifique, entre três minutos e trinta segundos até quatro minutos e cinquenta segundos, no vídeo de nossa autoria, disponível em: <https://youtu.be/su_UyxPyMRs>, acessado no dia 08/09/2020.

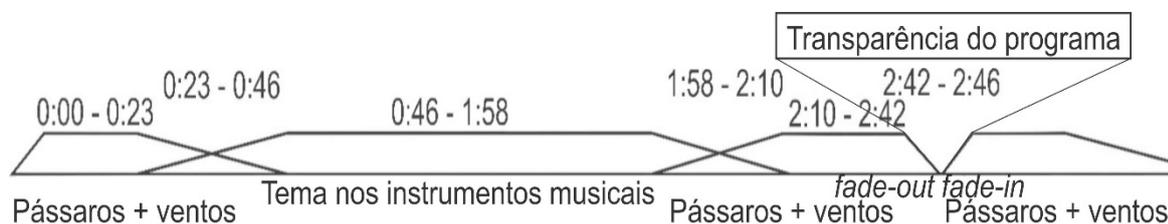
Ambos os exemplos de *games* proporcionaram uma experiência coerente ao seu mundo virtual, mesmo enfatizando a trilha musical ao invés da paisagem sonora atópica. No entanto, levantamos um exemplo que comete equívocos na elaboração da paisagem sonora atópica gerando contradições perante os requisitos em que nos baseamos. Esse *game* é o *Clash of Clans*, lançado pela empresa *Supercell* em 2010, que segue a mesma premissa dos *games* anteriormente apresentados (*Hearthstone: Heroes of Warcraft* e *This War of Mine*), ou seja, mistura a trilha musical e a paisagem sonora atópica para otimizar o *game*.

Clash of Clans contempla um mundo virtual com diversos elementos e estereótipos inspirados na mitologia nórdica, como valquírias, gigantes, guerreiros, entre diversos outros. O

jogador é o senhor de uma vila, seu objetivo é fomentar um exército para combater contra outras tribos, bem como melhorar seu sistema de defesa e gerenciar recursos.

Na cena em que o interagente organiza sua vila, a trilha musical e a paisagem sonora atópica, unidas em uma só faixa sonora, são compostas por sons de pássaros, ventos e instrumentos musicais (cordas, madeiras, metais e percussão). A faixa sonora tem duração de dois minutos e quarenta e dois segundos com adicional do *loop* de quatro segundos, graficamente disposto na linha de tempo demonstrada a seguir:

Figura 1 - *Timeline*, em segundos, da trilha musical do game *Clash of Clans*.



Fonte: Ilustração do autor

Nos quatro segundos em que ocorre o *loop* da faixa sonora (O *fade out*¹⁸, inserido aos 2 minutos e 42 segundos, e o *fade in*¹⁹, inserido aos 2 minutos e 46 segundos), transparece o programa por trás do mundo ficcional, pois evidencia-se o ponto exato do *loop* enquanto se executa a faixa sonora, com apenas sons de pássaro e de vento. O resulta sonoro desse efeito adicionado a essa faixa resulta em uma sensação inexistente no mundo real, já que todos, os pássaros e o vento, diminuem a intensidade até atingir a ausência de som, e logo após, partem do silêncio e aumentam a intensidade voltando para seu estado inicial. Essa maneira de lidar com a paisagem sonora atópica é incompatível com um ambiente sonoro no mundo real, seria apenas possível em um ambiente musical. A consequência desse tipo de efeito – *fade out* e *fade in* – em paisagem sonora atópica, prejudica a veracidade desse mundo virtual e, por consequência, enfraquece seu poder de imersão.

Para complementar a explicação, verifique, entre cinco minutos e dezoito segundos até sete minutos e trinta e sete segundos (final do vídeo), no vídeo de nossa autoria, disponível em: <https://youtu.be/su_UyxPyMRs>, acessado no dia 15/05/2020.

A utilização da trilha musical em conjunto com a paisagem sonora atópica é uma maneira de suprir com a descrição de um mundo virtual, recorrente nos *games*, até mesmo naqueles com alto

¹⁸ Efeito de *diminuendo* de todos os sons de um som ou faixa sonora até alcançar o silêncio, ou seja, desaparecimento do som.

¹⁹ Efeito de *crescendo* de todos os sons de um som ou faixa sonora que parte do silêncio e alcança a intensidade desejada na gravação.

orçamento e equipe de desenvolvimento. Ou melhor, abusam da trilha musical para conduzir a narrativa, mitigando o potencial da paisagem sonora atópica, essencial para o engajamento de estímulos e a experiência no mundo virtual (SINGER, 1998 *apud* CHANDRASEKERA, 2015, p. 1005).

A partir das observações acerca da paisagem sonora atópica, desenvolvemos uma solução que emancipa a paisagem sonora atópica da trilha musical. Com isso, é possível aprimorar o leque de opções nas formas de descrição e significação de um mundo virtual.

4 Criando paisagens sonoras atópicas

Para validar nossa proposta, bem como possibilitar os testes no processo de elaboração da paisagem sonora atópica, utilizamos os softwares livres *Pd-extended* (*Pure Data - extended*)²⁰, uma plataforma livre para a programação de dados multimodais com processamento em tempo real, e *Audacity*²¹, um editor de áudio. Ambos foram adequados para desenvolver e reproduzir uma paisagem sonora atópica. No *Pd-extended*, montamos os algoritmos para execução sonora, denominados *patches*²². Já no *Audacity*, tratamos de cada som que compõe a paisagem sonora atópica em questão (FARNELL, 2010).

Tal como o exemplo anteriormente apresentado a respeito do som do mar (Tópico 3.2), nosso modelo é baseado em aleatoriedade temporal de segmentos sonoros que pretendem descrever uma paisagem sonora atópica com fluxo contínuo de sons. Lembrando que esse tipo de paisagem sonora atópica não permite executar o recurso *fade out* e *fade in* na faixa sonora, apresentada como equívoco no exemplo sobre o *game Clash of Clans*. O ambiente sonoro selecionado foi o de uma caverna úmida – sons de gotas d’água espacialmente distribuídas com alta reverberação. Para tanto, em suma, selecionamos um arquivo sonoro e retiramos dele alguns sons de gota d’água, a fim de montarmos um algoritmo que gerencie a ativação e a manipulação dos parâmetros sonoros: altura, duração e intensidade. Desse modo, elaboramos uma paisagem sonora atópica, que evita os problemas de *loop* anteriormente levantados, pois ela é gerada a todo momento pelo montante de fragmentos sonoros de maneira mimética ao nosso mundo real.

²⁰ Disponível em <<https://puredata.info>>, acessado em 22 de março de 2017.

²¹ O *software Audacity*, desenvolvido na *Carnegie Mellon University*, foi lançado em 2000. Ele suporta as plataformas *Windows/ OS X/ Linux*. Disponível em <<http://www.audacityteam.org>>, acessado em 22 de março de 2017.

²² Denomina-se *patch* um diagrama que realiza um algoritmo no *software Pure-data* ou no *Pd-extended*.

5 Descrição e apreciação dos fluxogramas que descrevem a paisagem sonora atópica de uma caverna úmida

Para iniciarmos nosso experimento, selecionamos uma faixa sonora que descreve, acusticamente, uma caverna úmida, por meio do som contínuo de gotas d'água. O arquivo contém duração de 51 segundos, no formato *.WAV* (16 bits e *Sample Rate* = 44,1 KHz e de resolução), ocupando espaço de 8.863 KB, retirado do site: <www.freesound.org>, acessado no dia 10/05/2016. A partir disso, recortamos dessa faixa sonora 12 sons de gotas d'água com duração entre meio a dois segundos. Eles, que se configuram também no formato *.WAV* (16 bits e *Sample Rate* = 44,1 KHz e de resolução), foram nominados com a letra "G", sob a seguinte numeração: G01 (227 KB); G02 (229 KB); G03 (260 KB); G04 (345 KB); G05 (369 KB); G06 (296 KB); G07 (142 KB); G08 (314 KB); G09 (193 KB); G10 (317 KB); G11 (348 KB); G12 (330 KB). Todos esses fragmentos sonoros totalizaram 3.364 KB, ou seja, mais ou menos 2,6 vezes menor que o original.

Visto que esses fragmentos sonoros foram retirados de um contexto sonoro, em que os sons de gotas d'água são executados continuamente, utilizamos o *software Audacity* para modificarmos o som de cada gota d'água, retirando ruídos, aplicando *fade in* e *fade out* no final de cada faixa sonora, bem como nivelando a intensidade com os valores entre -14 dB a -9 dB de pressão sonora.

Após editados os 12 áudios de gotas d'água, montamos dois fluxogramas no *Pd-extended*: o primeiro com randomização da intensidade (volume do som) e espacialidade (*pan*) na execução dos sons de cada gota d'água e com modificação estrutural de *reverbs*; já o segundo com randomização semelhante ao experimento anterior, porém com acréscimo do parâmetro duração, executando o arquivo mais lento ou mais rápido que o original. Por consequência disso, ocorreu uma alteração na altura. Assim, executar os arquivos mais lentamente do que o original modifica a altura, tornando-a mais grave; executar os arquivos mais velozmente já modifica a altura de sons mais agudos.

Para compreendermos os *patches* desenvolvidos, primeiro colocaremos sucintamente todos os objetos que nos auxiliaram na elaboração da paisagem sonora com os fragmentos sonoros de gotas d'água, nas seções 5.1 até 5.4, para que, na seção 5.5 e 5.6, possamos apresentar o *patch* em si, bem como nossas considerações sobre os resultados obtidos. Todos os aspectos apresentados a seguir foram norteadores para a conclusão de nossos *patches* que estão executados e exemplificados nos *links* a seguir: <https://www.youtube.com/watch?v=rfVsji4TVEM&feature=youtu.be>, para o primeiro fluxograma (seção 5.5.1), acessado no dia 21/05/2020, e <https://www.youtube.com/watch?v=j5rT9GXJgpM&feature=youtu.be>, para o segundo fluxograma (seção 5.5.2), acessado no dia 21/05/2020. Ambos os vídeos incorporam legendas em toda a visita

aos espaços virtuais dos objetos construtores do algoritmo que executam nossa paisagem sonora atópica de caverna úmida.

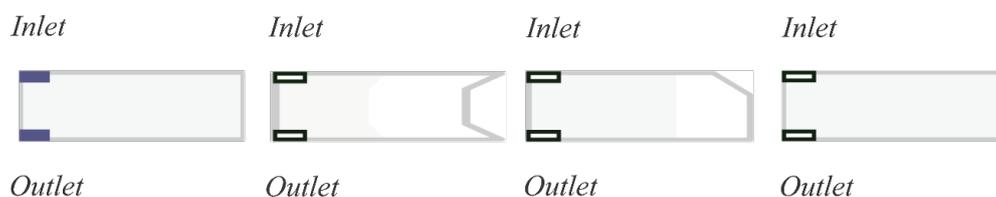
5.1 Pure Data

Pure-Data é um *software* desenvolvido por Miller Puckette nos anos 90. No nosso caso, usamos uma versão derivada desse original denominada *Pure-Data-Extended*, apresentada no texto como *Pd-extended*. Esse *software* oferece uma linguagem de programação visual com interatividade baseada no *dataflow* (fluxos de dados) que são apresentadas em *patches* – grupo de objetos que executam uma ou mais ações (FARNELL, 2010). A descrição dos objetos do *software* no texto terá como padrão o uso de colchetes para tornar, visualmente, mais parecido com o próprio *Pd-extended* - Exemplo: [readsf~]; [phasor~]; [trigger]. Nosso intuito, dessa forma, é elaborar uma paisagem sonora atópica descrevendo gotas d’água por meio de 12 fragmentos sonoros regidos com aleatoriedade em altura, intensidade, tempo de execução e reverberação.

5.2. Caixas e *subpatches*

Cada objeto, demonstrado por meio de uma caixa, carrega um ou mais *inlets* (entradas) para receber uma informação, disposto visualmente como um pequeno retângulo acima da caixa. Assim como ocorre com um ou mais *outlets* (saídas), para dar continuidade ao fluxo de dados, descrito como o pequeno retângulo abaixo da caixa.

Figura 2 - Divisão de caixas no *software Pd-extended*.



Fonte: Ilustração do Autor

Como se nota na *figura 2*, o *Pd-Extended* utiliza-se de quatro tipos de caixa. A caixa da esquerda, a primeira caixa, é aquela do tipo que processa apenas sinais sonoros; a segunda, denominada como *message box* (caixa mensagem), é utilizada para adicionar informação a outra caixa; a terceira, denominada *number box and symbol box* (caixa de número e caixa de símbolo), serve para explicitar resultados obtidos; a quarta é a caixa que processa operações matemáticas.

Além disso, é possível criar sua própria caixa utilizando-se de um objeto com prefixo *pd*. Essa função é aplicada também para organizar um *patch*, agrupando outros *patches* no seu interior.

Para tanto, criamos um arquivo com as letras *pd* antes do nome [pd nome], para que, com isso, consigamos abrir outra janela com outro *patch*, denominado, no nosso caso, *sub-patch*. Assim, é possível incluir os valores [inlet] e [outlet] para determinar os valores de entrada e saída do objeto [pd nome], a fim de tornar o *patch* organizado, separado em seções e subseções.

5.3 Objetos não sonoros

O *software Pd-extended* disponibiliza diversos objetos que são responsáveis por operações matemáticas. Abaixo, descrevemos os usados em nossos fluxogramas.

5.3.1 Aleatoriedade

Para gerarmos eventos aleatórios, contamos com dois objetos, respectivamente [randomF] e [random]. O objeto [randomF] gera números aleatórios decimais e [random] números aleatórios inteiros. A mensagem [seed] é um recurso válido para controlar um pouco as variações da aleatoriedade.

5.3.2 Cronômetro e pulsação

O objeto [Metro] gera um pulso a cada período (a medida de tempo padrão do *software* está em milissegundos: 1000 = 1 segundo), determinado pelo valor inserido à direita do objeto [Metro]. Por exemplo, [Metro 2000] emite um pulso a cada dois segundos, e, se trocarmos [Metro 2000] para [Metro 10000], emite-se um pulso a cada dez segundos.

5.3.3 Trigger, bang, toggle

O objeto [trigger], também representado como [t], recebe qualquer tipo de sinal. Por conta disso, abrangem-se diversas funções: ativar mensagens, converter tipos diferentes de objetos, entre outras.

O objeto [bang], representado graficamente como um círculo dentro de um quadrado, é responsável por enviar um pulso, ou confirmar que um pulso foi recebido. Em ambas as situações, o círculo pisca na cor preta, significando que um sinal passou pelo caminho ligado ao objeto [bang].

O objeto [toggle], representado por um quadrado, ativa ou desativa um objeto ou estrutura. Quando clicado, um "X" ocupa o espaço do quadrado demonstrando que o objeto [toggle] está ativado; se clicado novamente, o "X" é apagado e o objeto volta a ser um quadrado vazio.

5.3.4 Seleção

O objeto [sel] é responsável por ampliar e dividir os caminhos de um sinal. Por exemplo: [sel 0 1] dispomos de dois caminhos, um a partir do número 0 e outro do número 1; já se tivermos [sel 0 1 2 3 4 5 6] serão sete caminhos, respectivamente: 0; 1; 2; 3; 4; 5; 6.

O fator [moses] comporta possibilidades semelhantes ao fator [sel]. Sua diferença é que ele divide o caminho em apenas duas saídas e não em diversas saídas como o [sel]. Por exemplo: [moses 4] significa que até o valor 3 temos um caminho e a partir do valor 4, outro caminho.

5.3.5 Atraso

O fator [pipe] permite atrasar a duração de um sinal. O número inserido será referente ao valor, em segundos, do atraso do envio do sinal. Por exemplo, se inserirmos o fator [pipe 5000] entre duas reproduções sonoras, o programa reproduzirá o primeiro som para depois de cinco segundos executar o segundo som. Se modificarmos o fator [pipe 5000] para [pipe 20000], a diferença do atraso do som passa de cinco segundos para vinte segundos.

5.3.6 Início automático

O objeto [loadbang] inicia a execução sonora quando abrimos um arquivo do *Pd-extended*.

5.3.7 Send e Receive

Trata-se de objeto que recebe e enviar valores sem a necessidade de ligação (como se estivessem conectado por *wifi* ou *bluetooth*). Para tanto, criamos uma caixa [s nome] na saída do sinal, e [r nome] para entrada do sinal.

5.4 Objetos sonoros

Os objetos sonoros carregam um *design* gráfico diferenciado dos outros objetos. Os pontos de intersecção são caracterizados pela cor cinza, conforme mostrado na primeira caixa da Figura 2 e mediante o símbolo “~” no final do nome da caixa.

5.4.1 Reprodução básica do som [readsf~]

Desenvolvemos um *patch* com o objeto [readsf~] para reprodução do som. Ele se limita nos arquivos cujos formatos são *.WAV*, *.AIFF*, *.AIF*.

5.4.2 Reprodução básica do som [soundfiler], [phasor~] e [tabread4~]

Desenvolvemos outro patch com objeto [soundfiler] e [tabread4~] para armazenamento do arquivo sonoro, e com [phasor~] para a reprodução do som. Sua diferença perante o objeto [readsf~] é que controlamos a velocidade de execução da faixa sonora, alterando, assim, sua altura (frequência do som).

5.4.3 Reverbs

A partir desse *patch* incluiremos todas as possibilidades de variante timbrística com o objeto [freeverb~], que disponibiliza a manipulação de *reverbs*. Por conseguinte, os *reverbs* contribuem para simulação espacial do som na paisagem sonora atópica, oferecendo uma maior veracidade para o ambiente sônico do mundo virtual, já que, no mundo real, por exemplo, a resultante acústica do som de um pássaro em uma floresta difere-se quando o mesmo som de um mesmo pássaro é reproduzido em uma caverna.

5.4.4 Otimização

Os *reverbs* trazem uma riqueza timbrística para a paisagem sonora atópica. No entanto, se o utilizarmos em cada fragmento sonoro, o programa pode não suportar o extenso número de processamentos, levando a um *crash* (interrupção do funcionamento do *software*). Para resolver esse problema, utilizamos de um *reverb* geral com os objetos [throw~] – que envia o sinal sonoro – e [catch~] – que recebe o arquivo sonoro.

5.4.5 Espacialização do som

O recurso responsável pelo balanço das saídas sonoras é o objeto [pan~], que nos permite simular espaços em um mundo ficcional.

5.4.6 Medidor de amplitude

O medidor do nível de decibel é essencial para visualizarmos as nuances da intensidade sonora, para assim prevenir distorções indesejáveis no som.

5.5 Os patches

Por meio dos elementos descritos nas seções anteriores, efetuamos diversos testes no *software Pd-extended*, a fim de desenvolvermos uma paisagem sonora atópica coerente com os nossos

requisitos de análise, principalmente em relação à transparência do programa computacional gerador do mundo virtual.

A estrutura, que abrange ambos os objetos de reprodução, [readsf~] e [phasor~], dispõe-se de 3 *sub-patch* nominados: [pd saida_som], [pd gotas_esquerda], [pd gotas_direita]. Inserimos o medidor de decibel para observarmos as variantes da intensidade quando executarmos o algoritmo que gerencia todos os sons da paisagem sonora atópica. Indicam-se, no caso, a expressão para ativar o som no *Pd-extended*, [; pd dsp \$1], e os objetos, [loadbang] e [s loadbang], para efetuar o primeiro pulso, assim que o arquivo principal do *patch* for aberto. [Tentei arrumar, mas não sei se ficou bom. Revise, por favor]

No *sub-patch* [pd saida_som], adicionamos os objetos de [catch~ L1] e [catch~ R1] seguidos do objeto [freeverb~], obtendo, assim, a possibilidade de *reverbs* de todos os pulsos sonoros executados em todo *patch*.

Os *sub-patches* [pd gotas_esquerda] e [pd gotas_direita] executam os arquivos de som que compõe a paisagem sonora, sendo o primeiro referente à saída sonora da esquerda e o segundo da direita. Com esse *patch* finalizado, dedicamo-nos aos valores que descrevem a gama da aleatoriedade da paisagem sonora. Asseguramos que a amplitude varia entre 0 a 2 descrita no objeto [randomF 2]. Este valor foi selecionado para que possamos dimensionar os números decimais em quatro níveis de audição, a saber:

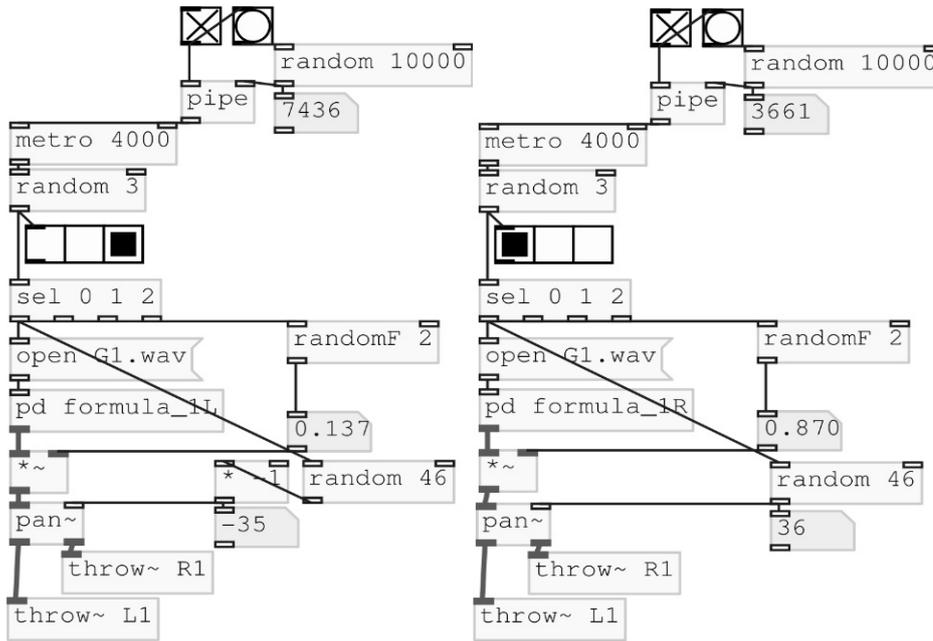
- a) 0.00 a 0.5 sem som até o *pianíssimo*
- b) 0.5 a 1.0 com os níveis de piano a *mezzo piano*
- c) 1.0 e 1.5 entre *mezzo piano* a *mezzo forte*
- d) 1.5 a 2.0 com níveis de *mezzo forte* a *forte*.

Adicionamos espacialização através do objeto [pan~], dividindo-o em 5 níveis entre os números -45 (esquerda) a +45 (direita):

- 1º) Extrema esquerda: valores entre -45 a -30.
- 2º) Esquerda/Centro: valores entre -29 a -10.
- 3º) Centro: valores entre -10 a +10.
- 4º) Direita/Centro: valores entre +10 a +29.
- 5º) Extrema direita: calores entre +30 a +45.

Para alcançarmos esses valores, adicionamos aleatoriedade na espacialização por meio dos objetos [random 46], que resulta de 0 a +45, e [random 46] [* -1], entre -45 até 0.

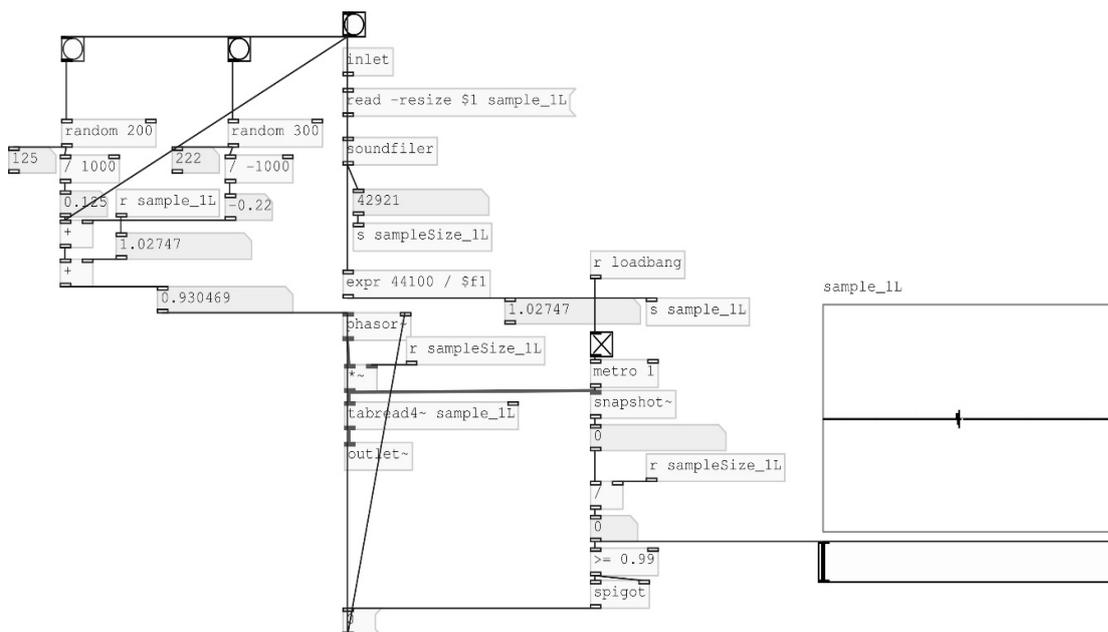
Figura 4 - sub-patch da execução do som por meio do objeto [phasor~].



Fonte: *Printscreen do patch no Pd-extended*

Esse *sub-patch* [formula_nL] pauta-se em outros objetos matemáticos e musicais para ler o arquivo, pois precisa calcular o tamanho da amostra sonora (*Sample Size*) e inseri-lo em uma tabela para, então, alterar a velocidade de execução. A estrutura, apresentada a seguir, refere-se ao som da gota d’água G01 executada espacialmente entre o centro e o lado esquerdo:

Figura 5 - Estrutura interior do sub-patch [formula_1L].



Fonte: *Printscreen do patch no Pd-extended*

5.6 Resultados

Com base nos requisitos que qualificam uma paisagem sonora atópica, apresentamos conclusões positivas e negativas acerca dos nossos fluxogramas desenvolvidos no *software Pd-extended*, baseando-se em nossa percepção da resultante sonora.

Os fluxogramas proporcionaram positivamente: (1) na liberdade da composição geral de uma paisagem sonora atópica, já que possibilita controle de qualquer fragmento sonoro. Além disso, amplia as possibilidades nas modificações estruturais da paisagem sonora atópica, tornando-a mais densa, suave, intensa, rarefeita, entre diversas outras modificações estruturais; (2) na verossimilhança em mimetizar, em um mundo virtual, o som de uma caverna úmida, assemelhando-se, também, da faixa sonora original, de que retiramos os fragmentos sonoros; (3) na otimização de espaço no *hard disk*, visto que a faixa sonora original, da qual retiramos o material sonoro, ocupa 8.863 KB de espaço no *hard disk* e que todos os sons de gotas d'água retirados dessa faixa sonora totalizam 3.364 KB (os projetos do *Pd-extended* possuem 35 KB para o fluxograma com base no objeto [readsf~] e 98 KB para o fluxograma que usa o [phasor~]), cerca de duas vezes e meia menor).

Em suma, a fragmentação dos sons concedeu-nos uma enorme gama de recursos na invenção de uma paisagem sonora atópica em diversos aspectos. As possibilidades de composição ampliam o desenvolvimento de variados arcos narrativos que, juntos, formam um ambiente ficcional complexo e independente do interagente. Isso confirmou que é possível, por meio de algoritmos, gerar, virtualmente, um ambiente sônico que corrobore a *diégesis* de um *game*, intensificando sua verossimilhança com o mundo real, já que contempla os requisitos propostos no início deste artigo, que foram norteadores para análise da paisagem sonora e para busca de soluções de problemas sonoros, haja vista possíveis incoerências entre mundo virtual e real.

Para os aspectos negativos, levantamos os seguintes itens: (1) dificuldade na seleção e fragmentação do som da faixa original. Deparamo-nos com ruídos indesejáveis que limitaram a escolha do material de cada fragmento. Esse ruídos indesejáveis foram reduzidos por meio de edição de áudio no *software Audacity*, já que, no processo de fragmentação e seleção do som das gotas d'água, o fragmento perde sua funcionalidade descritiva, antes contextualizado no material original; (2) por conta do problema descrito no primeiro item deste parágrafo, além da redução do ruído, fez-se necessária a aplicação de outras modificações sonoras, tais como: a compressão da intensidade e dos picos de cada fragmento; a ampliação da duração para inserirmos os efeitos de *fade in* e *fade out*; o nivelamento do grau de intensidade entre todos os fragmentos; (3) por fim, dependendo da quantidade de elementos inseridos no fluxograma, pode ocorrer um aumento de processamento da

máquina durante a execução do *patch*, nesse caso, cabe ao músico e programador adequar o nível de processamento do arquivo *Pd-extended* para as demandas essenciais do *game*.

Os aspectos negativos levantados anteriormente podem ser minimizados se buscarmos sons já separados, sem a necessidade de tirá-lo de algum contexto. Com isso, a única questão que realmente poderá prejudicar o processo é o terceiro item, que leva em conta o nível de processamento para gerenciar os fragmentos sonoros no *Pd-extended*.

6 Middlewares para desenvolvimento de paisagem sonora

Em nosso exemplo, utilizamos o *software Pd-extended* para desenvolvermos uma paisagem sonora atópica. Quando se usa um *software* para gerar processamento paralelo ao *software* central de um *game*, a *engine* é nomeada *middlewares*. Resumidamente, os *middlewares* nos *games* são *softwares* compatíveis com uma *engine* e que, no nosso caso, suportam o desenvolvimento do som nos *games*. No entanto, esses *middlewares* são dispensáveis, já que, normalmente, as *engines*, como as *Unity 3D*, *Unreal*, *Construct*, *CryEngine*, disponibilizam seu próprio sistema de gerenciamento sonoro. Todavia, a interface das *engines* são diversas e, geralmente, distante dos padrões de *software* de música. Por conta disso, o músico, ao invés de aprender o funcionamento da *engines* e suas interfaces, pode aprimorar-se em um *middleware* para elaborar sua paisagem sonora atópica e também a trilha sonora, tendo assim maior controle de sua aplicação no produto final de um *game*. Tendo isso em mente, apontaremos algumas opções para desenvolvimento da paisagem sonora atópica, alternativas ao *Pd-extended*.

6.1 FMOD

FMOD, desenvolvido pela *Firelight Technologies*, é um *software* voltado para o desenvolvimento de trilha sonora interativa. Ele pode ser usado para gerar paisagem sonora atópica. Há uma versão gratuita, que suporta os sistemas operacionais *Windows*, *OS X*, *Linux*, *Android*, *BlackBerry*. Além disso, é compatível com a programação de diversas *engines*, dentre elas a *Unity 3D*, *Unreal*, *CryEngine*.

Esse *software* apresenta uma interface gráfica semelhante a um *software* de áudio multipista, conhecido como *DAW (Digital Audio Workstation)*, comum no ofício de gravação, edição, mixagem e composição musical. Dessa forma, o *FMOD*, dentre os *softwares* aqui expostos, costuma ser o mais familiar para um músico. Para elaborar uma paisagem sonora neste *software*, é necessário adaptar os eventos na *Timeline* (linha do tempo que executa os sons) utilizando-se de sub eventos, que são eventos gerados através de outro evento.

6.2 *Wave Works Interactive Sound Engine (Wwise)*

O *Wwise* é um *software* desenvolvido pela *Audiokinect* voltado para elaboração de trilha sonora interativa em *games* com ferramentas específicas para gerar paisagem sonora atópica. Possui versão gratuita e, semelhante ao *FMOD*, suporta os sistemas operacionais *Windows*, *OS X*, *Linux*, *Android*, *BlackBerry*. Além disso, é compatível com outras *engines*, dentre elas a *Unity 3D*, *Unreal*, *CryEngine*.

Sua interface difere-se dos *softwares* convencionais de áudio, sendo mais próximo das *engines*. Com isso, é mais difícil a um músico adaptar-se, mesmo aquele acostumado aos *softwares* de gravação. No entanto, suas ferramentas abarcam todas as propriedades competentes para gerar uma trilha sonora de alta complexidade. É um *software* com diversas ferramentas para desenvolver uma paisagem sonora atópica.

6.3 Algoritmo no documento de texto

Uma das opções para o músico, responsável pela trilha sonora de um *game*, é elaborar algoritmos apresentados em um documento de texto ou imagem com nome dos arquivos sonoros e seus comportamentos. A partir disso, o programador transcreve esse documento para linguagem de programação da *engine*.

Por um lado, essa maneira de compor uma paisagem sonora atópica concede liberdade para o músico desenvolver suas ideias sem a necessidade de dominar a operação de qualquer *software middleware* de áudio e a *engine* do *game*. Por outro lado, em função da imprevisibilidade de todas as variantes, é bem possível que ocorram diversos ajustes após a primeira transcrição do programador.

7 Considerações finais

O desenvolvimento da paisagem sonora atópica atinge uma infinidade de possibilidades, desde uma simples reprodução de gotas d'água até uma simulação complexa do funcionamento de uma cidade inteira. Este artigo tratou de problemas acerca da paisagem sonora nos *games*. Inicialmente, transportamos o significado do termo com objetivo de comportar nosso recorte, para depois subdividi-lo em duas categorias: paisagem sonora tópica – com uma fonte sonora visível ao interagente e acoplada a algum elemento gráfico do mundo virtual – e paisagem sonora atópica – com a fonte sonora invisível ao interagente, desvinculado a um objeto do mundo virtual.

A partir dessa dicotomia, analisamos as relações transversais entre essas duas categorias de paisagem sonora, mediante três requisitos: (1) enriquecer a descrição do mundo virtual, (2) ser coerente com a *diégesis* do *game* e (3) dissimular o programa que rege o mundo virtual do *game*. Por

meio dessas análises, considerou-se que a paisagem sonora atópica, quando condensada em uma faixa sonora, é passível de incoerência, se comparada a um ambiente sônico do mundo real. Em contrapartida, parte dos *games* analisados optou por mesclar a paisagem sonora atópica com a trilha musical, mitigando seu impacto na descrição sonora do mundo virtual. Em detrimento disso, propomos a elaboração da paisagem sonora atópica através de algoritmos, com ou sem *middlewares*.

Para efetivar essa paisagem sonora atópica, desenvolvemos algoritmos no *software Pd-extended* que resultaram em dois fluxogramas, descritos no texto e exemplificados por vídeos compartilhados na plataforma *Youtube*. Nosso desfecho, ainda em âmbito exploratório, apontou resultados otimistas em nossa proposta, já ela que nos garantiu a manipulação libertina dos fragmentos sonoros, gerando um ambiente sônico otimizado, no *hard disk*. Ademais, mostrou-se verossímil com o mundo real. Esperamos que nossa exposição da problemática acerca da paisagem sonora nos *games* ofereça suporte para futuros estudos na área, bem como aproxime os ofícios de música e programação, para que, juntas, tragam soluções eficazes na descrição de um mundo virtual.

Referências

- AARSETH, Espen J. **Cybertext: perspectives on Ergodic Literature**. Baltimore and London: Johns Hopkins University Press, 1997.
- AGUIAR, M.; BATTAIOLA, A. L. Gameplay: uma definição consensual a luz da literatura. In SBGames, **XV Simpósio Brasileiro de Games e Entretenimento Digital**. São Paulo, 2016.
- CAMARGO, Fernando Emboaba de. **Interatividade e narratividade sonora nos games**. 2018, 188f. Tese (Doutorado pelo Instituto de Artes) – Universidade Estadual de Campinas, 2018.
- CARRASCO, Claudiney Rodrigues. **Trilha Musical: Música e Articulação Fílmica**. 1993. Dissertação (Escola de Comunicações e Artes). Universidade de São Paulo, São Paulo, 1993. Disponível em <<https://repositorio.usp.br/item/000728336>>, acessado no dia 08/09/2020.
- CHANDRASEKERA, Tilanka. Virtual environments with soundscapes: a study on immersion and effects of spatial abilities. **Environment and Planning B: Planning and Design**. Volume 42, p.1003 - 1019, 2015.
- CHION, Michael. **A Audiovisão: som e imagem no cinema**. [tradução Pedro Elói Duarte]. Portugal: Lisboa, Edições Texto & Grafia, 2008.
- COOLINS, Karen. An Introduction to the participatory and Non-linear Aspects of Video Games Audio. **Essay on Sound and Vision**. Edição Stan Hawkins e John Richardson. Helsinki: pp.263 – 298, Helsinki University Press, 2007.
- COOLINS, Karen. **From Pac-Man to Pop Music: Interactive Audio in Games and New Media**. Canada: Universidade de Waterloo, 2008a.

COOLINS, Karen. **Game Sound: an introduction to history, theory, and practice of vídeo game music and sound design**. Estados Unidos: MIT Press, 2008b.

CORREA, Elisa C. D. Usuário, não! Interagente: proposta de um novo termo para um novo tempo. **Encontros Bibli**: revista eletrônica e ciência da informação, v.19 n°41, p. 23049, set/dez, 2014.

COSTE; PIER, Didier e John. **Narrative Levels**. Alemanha: University of Hamburg, Interdisciplinary Center of Narratology, 2011. Disponível em: <http://wikis.sub.uni-hamburg.de/lhn/index.php/Narrative_Levels>, acessado no dia 05/05/2017.

DONOVAN, Tristan. **Replay: The History of Video Games**. United Kingdom: Yellow Ant, 2010.

GORBMAN, Claudia, **Unheard Melodies: narrative film music**. Londres: The British Film Institute, 1987.

FARNELL, Andy. **Designing Sound**. Estados Unidos: The MIT Press, 2010.

PORTER, Alastrar et al. **Freesound**. 2005. Disponível em <<http://freesound.org/>>. Acesso no dia 21/05/2020.

ILLUSIE, Luc. What is... a Topos?. **AMS**, vol.51, n°9, p.1060-1061. Outubro, 2004.

JØRGENSEN, Kristine. **‘What are those Grunts and Growls Over There?’ Computer game áudio and player action**. 2007, Tese (Doutorado em Cognição e Comunicação) - Departamento de Mídia, Universidade de Copenhagem, Copenhagem 2007.

JUUL, Jesper. **A clash between game and narrative**. 1999. (Dissertação Mestrado em Linguagem e Literatura) - Instituto de língua Nórdica e literatura, Universidade de Copenhague, Copenhague, 1999.

MIRA, David Moedas. **Das sombras de Platão ao realismo de Iracema: a representação do real no cinema**. 2012, Dissertação (Mestrado em Ciências da Comunicação) - Universidade Nova de Lisboa, Faculdade de ciências sociais e humanas, 2012. Disponível em <<http://hdl.handle.net/10362/7882>>. Acesso no dia 14/09/2020.

PUCKETTE, Miller S. **PD Documentation**. 1996. Disponível em <<https://puredata.info/docs/manuals/pd>>. Acesso no dia 21/05/2020.

Referências multimídias

11 bit.studios (2014). **This war of mine** [Microsoft Windows, Mac OS, Android, IOS]: 11 bit.studios, Deep Silver. Disponível em <<https://www.gamespot.com/this-war-of-mine>>. Acesso no dia 21/04/2020.

Audiokinetic (2011). **Wwise**. [PC, MAC, LNX]. Audiokinetic. Disponível em <<https://www.audiokinetic.com>>. Acesso no dia 13/09/2020.

Blizzard Entertainment. (2014). **Hearthstone: Heroes of Warcraft** [Android, IOS, Microsoft Windows, Mac OS]: Blizzard Entertainment. Disponível em <<https://www.gamespot.com/hearthstone-heroes-of-warcraft>>. Acesso no dia 21/04/2020.

Crytek (2002). **Cryengine**. [PC, MAC, LNX]. Crytek. Disponível em <<https://www.cryengine.com>>. Acesso no dia 13/09/2020.

Epic Games (1998). **Unreal Engine**. [PC, MAC, LNX]. Epic Games. Disponível em <<https://www.unrealengine.com>>. Acesso no dia 13/09/2020.

Firelight Technologies (2011). **FMOD**. [PC, MAC, LNX]. Firelight Technologies. Disponível em <<https://www.fmod.com>>. Acesso no dia 13/09/2020.

Klei Entertainment, BlitWorks, Capybara Games, 505 Games (2013). **Don't Starve**. [PC, MAC, LNX, PS4]. Klei Entertainment, 505 Games. Disponível em <<https://www.gamespot.com/dont-starve>>. Acesso no dia 21/04/2020.

Supercell. (2012). **Clash of Clans** [Android, IOS]: Supercell. Disponível em <<https://www.gamespot.com/sponsored-clash-of-clans>>. Acesso no dia 21/04/2020.

Scirra (2011). **Construct**. [PC, MAC, LNX]. Scirra. Disponível em <<https://www.construct.net>>. Acesso no dia 13/09/2020.

Unity Technologies (2005). **UnityEngine**. [PC, MAC, LNX]. Unity Technologies. Disponível em <unity.com>. Acesso no dia 13/09/2020.

Informações sobre o Artigo

Resultado de projeto de pesquisa, de dissertação, tese: Tese UNICAMP “Interatividade e narrativa sonora nos games”.

Fontes de financiamento: CAPES

Apresentação anterior: Na Tese: mesma ideia, mas texto diferente

Agradecimentos/Contribuições adicionais: não se aplica.

Fernando Emboaba de Camargo

Doutor em música pelo Instituto de Artes (IA) na Universidade Estadual de Campinas. Mestre em musicologia na Escola de Comunicação e Artes (ECA), na Universidade de São Paulo. Atua como professor de Artes (Efetivo) na Prefeitura Municipal de Ribeirão Preto. Professor Doutor na Universidade de Ribeirão Preto (UNAERP) ministrando disciplinas de Harmonia, Análise e Estruturação, Atividades Práticas, Orientação de Monografia.

E-mail: fernandoemboabadecamargo@gmail.com

ORCID: <https://orcid.org/0000-0003-4491-3197>

José Fornari

Pesquisador carreira Pq do Núcleo Interdisciplinar de Comunicação Sonora da UNICAMP. Possui 2 PósDocs; um em Síntese Sonora Evolutiva (2004, no NICS) e outro em Modelos computacionais de cognição musical (2007, na Finlândia). Tem Doutorado (2003) e Mestrado (1994) em Processamento de áudio musical (FEEC, UNICAMP). É formado em Música Popular (1994) e Engenharia Elétrica (1990), ambos pela UNICAMP.

E-mail: fornari@unicamp.br

ORCID: <https://orcid.org/0000-0003-1049-7105>