Uma Proposta de Arquitectura para Composição Dinâmica de Soundscapes em Videojogos

Durval Pires
Department of Informatics Engineering
University of Coimbra
3030-290 Coimbra, Portugal
durval@student.dei.uc.pt

Valter Alves
Polytechnic Institute of Viseu, Portugal
& Centre for Informatics and Systems
of the University of Coimbra
3030-290 Coimbra, Portugal
valter@estv.ipv.pt

Licinio Roque
Department of Informatics Engineering
University of Coimbra
3030-290 Coimbra, Portugal
lir@dei.uc.pt

Resumo

A imprevisibilidade associada à interactividade presente no meio videojogo, faz com que a dificuldade de fazer sound design para este contexto seja extremamente elevada. Como consequência, é possível encontrar situações em que o contexto de jogo e a soundscape que é ouvida pelo jogador não se complementam.

Neste trabalho é apresentada uma arquitectura, informada pela teoria de Acoustic Ecology, que permite testar uma nova abordagem ao problema da composição dinâmica de soundscapes em videojogos. O principal elemento do sistema apresentado é um módulo de composição dinâmica que executa em run-time. Também como parte da solução proposta, é fornecida uma API que permite que os designers especifiquem elementos e contextos sonoros. Durante a execução do videojogo, o módulo monitoriza a soundscape e aplica algumas técnicas de composição, tendo em conta as especificações dos designers, definidas por via da API.

É apresentado um exercício que consistiu na reimplementação de um jogo usando a arquitetura aqui proposta. Este exercício confirmou a exequibilidade da solução, reforçando a ideia de que as técnicas

desenvolvidas podem realmente ser úteis nas diversas situações de jogo.

Author Keywords

Acoustic ecology, dynamic soundscape composition, game audio, game design, middleware, healthy soundscape, sound design, sound engine, soundscape.

Composição dinâmica de soundscapes, desenho de videojogos, desenho de som, Ecologia Acústica, som em videojogos, middleware, motor de som, soundscape, soundscape saudável.

ACM Classification Keywords

H.5.2 **[Information Interfaces And Presentation (e.g., HCI)**]: User Interfaces – auditory (non-speech) feedback, evaluation/methodology, standardization, styles guides, theory and methods; H.5.1 [**Information Interfaces And Presentation (e.g., HCI)**]: Multimedia Information System – audio input/output, evaluation/methodology; H.5.5 [**Information Interfaces and Presentation (e.g., HCI)**]: Sound and Music Computing – methodologies and techniques; K.8.0 [**Computing Milieux**]: Personal Computing – General – Games.

General Terms

Design, Standardization, Languages, Theory.

Introdução

O sound design aplicado aos videojogos enfrenta o enorme desafio de lidar com a natureza dinâmica do meio [7, 8, 16]. Os jogadores, ao executarem variadas acções sobre diferentes entidades presentes no jogo, tornam-se parte integrante da composição da soundscape. Além disso, como cada uma dessas entidades pode ter variadas expressões acústicas, cada sessão de jogo poder ter uma composição diferente das restantes. Esta imprevisibilidade, se não for devidamente abordada, pode resultar em composições que comprometem as intenções do designer, quer em termos estéticos quer em termos semânticos.

A indústria tem vindo a procurar soluções para este problema, embora maioritariamente recorrendo a ferramentas e técnicas originárias de meios lineares [7]. Ainda assim, existem ferramentas de Audio Middleware que oferecem funcionalidades que tentam alterar essa tendência, [3, 4, 5] e oferecer novas capacidades. Porém, o seu custo, quer monetário quer em termos do *know-how* necessário para as utilizar, faz com que estas ferramentas não sejam viáveis para todos os criadores de jogos [10]. Por outro lado, mesmo com estas ferramentas, é muito difícil para os sound designers conseguirem prever, e preparar, todas as situações passíveis de ocorrer no cenário de jogo.

Um conceito que pode ser valioso na busca de novas abordagens ao problema é o conceito de *soundscape*, originário da teoria de Acoustic Ecology [17] e introduzido por Schaffer [17] e aprofundado por Truax, nomeadamente em termos da sua composição [18]. Este conceito enquadra o som num ambiente que deve ser tratado como um todo, em que todas as relações se influenciam mutuamente. O equilíbrio funcional entre a variedade e a complexidade deste sistema de relações entre diferentes entidades e o ambiente, é fulcral para a manutenção de uma soundscape *saudável.* Adicionalmente, este tipo de soundscapes deve permitir que os sons presentes nela consigam transmitir de forma clara a semântica que a eles está associada.

Layer	Intenção
Ambiance	Sons do ambiente onde o jogador se encontra. Oferece noção de lugar.
Dialogue	Qualquer forma de discurso presente no jogo. Oferece vários tipos de informação.
Music	Qualquer forma de composição musical presente no jogo. Ajuda a definir o tom emocional de cada momento.
Foley	Sons reais que caracterizam uma entidade ou evento.
SFX	Sons "imaginários", que são colocados na cena para ajudar a realçar a expressão sonora de alguma entidade ou evento.

Tabela 1 - Sound Layers propostos por Nick Peck [15]

O conceito de soundscape já foi utilizado em contexto de videojogos, nomeadamente para efeitos de definição de ambientes acústicos [6]. Porém, a definição de soundscape defendida por Schaffer e Truax argumenta que o contexto no qual um som é ouvido é parte capital da sua percepção [18]. O mesmo som, em diferentes contextos, pode assumir diferentes significados. Transferindo este mindset para sound design para videojogos, os diferentes contextos oferecidos durante a jogabilidade, exigem diferentes abordagens de sound design [1,2].

Sistema de Composição Dinâmica de Soundscapes

Embora reconheçamos a dificuldade inerente a esta tarefa, acreditamos ser valioso desenvolver formas de permitir que a composição da soundscape tenha em conta tanto as intenções do designer, como as acções executadas pelo jogador [20]. Também se torna importante que as possíveis soluções encontradas estejam ao alcance de todos os criadores de jogos, mesmo os que à partida detenham menos conhecimentos em termos de sound design. Além disso, em acréscimo às suas potencialidades, o facto de novas abordagens serem dadas a conhecer pode ajudar a sensibilizar os criadores de jogos para o potencial que o som pode ter no processo de game design.

Como tal, nesta secção apresentamos uma proposta de um módulo de composição dinâmica que faz uso de uma abordagem holística, inspirada nos conceitos de Acoustic Ecology. A base desta solução consiste num sistema que modera a soundscape de acordo com eventos que recebe do jogo. Ao receber os eventos, as características dos elementos sonoros presentes na soundscape são analisadas, sendo de seguida aplicadas heurísticas de acordo com essa análise. Este sistema procura oferecer um conjunto de técnicas de composição que mantenham a soundscape saudável.

A caracterização prévia dos elementos sonoros é feita através de uma API também desenvolvida de raiz para esta solução. Esta API procura oferecer acessibilidade e permitir uma abordagem que seja mais próxima, em termos do discurso, de pessoas do domínio do sound design. Nesse sentido, os elementos sonoros, ao serem criados, podem ser classificados de acordo com o seu nível semântico (aqui designado de *Layer*) [15]. Existem 5 Layers diferentes: Ambiance, Dialogue, Music, Foley e Sound Effects (SFX) (ver Tabela 1). De forma semelhante, as fontes também podem ser associadas a personagens do jogo (*Agents*), e também podem ser associadas a tipos de explorações de som (*Patterns*) [1,2].

Na Figura 1 é apresentada a arquitectura geral do sistema proposto. A caixa no canto superior esquerdo representa todo o código que é específico de um projecto de jogo. É aí que está a lógica de jogo, por exemplo. A caixa no canto inferior esquerdo representa a API criada, com as classes e métodos que implementam os recursos oferecidos (por exemplo, o código que cria uma fonte de som ou um contexto). A ideia é que os programadores chamem esse código a partir da lógica de jogo.

O módulo de composição dinâmica é representado pela caixa à direita (DSCM). A comunicação entre o código da API e o módulo de composição é assegurado por mensagens OSC (Open Sound Control). OSC é um protocolo standard para comunicação por mensagens com um formato semelhante a um URL. Este protocolo

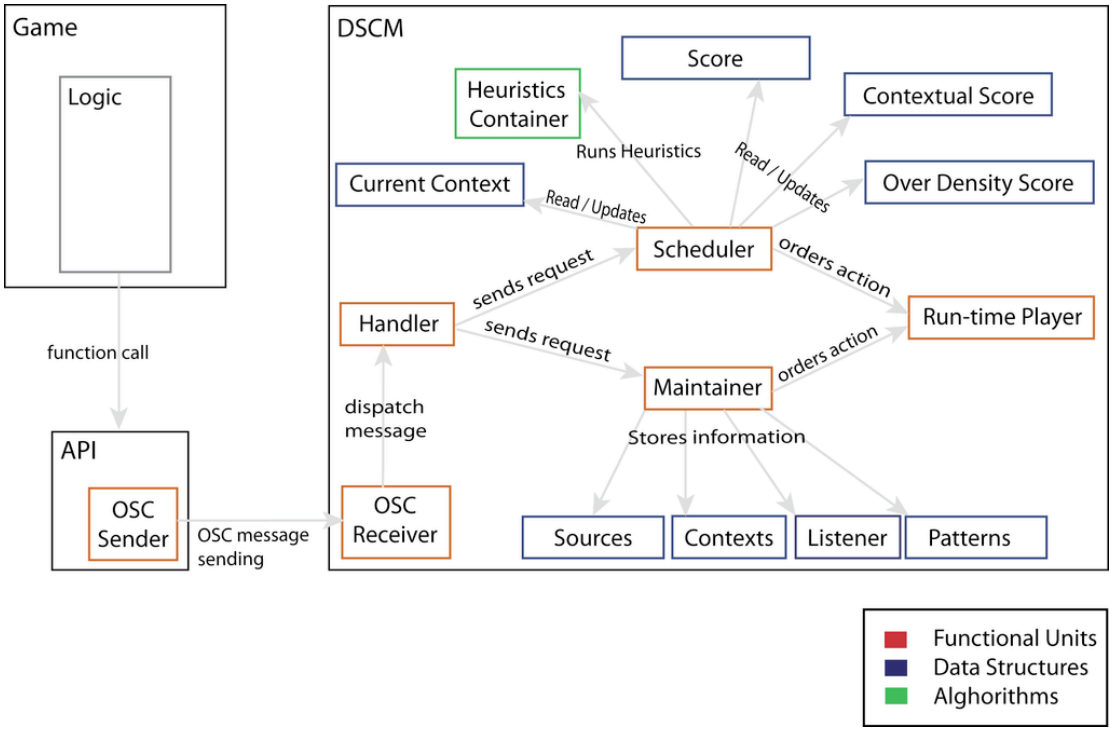

Figura 1 - Arquitectura geral do módulo de composição

é optimizado para ser utilizado por rede, oferecendo um alto nível de interoperabilidade, precisão, e flexibilidade.

Durante a execução do videojogo, o *OSC Receiver* encaminha as mensagens recebidas (enviadas pelo *OSC Sender*) para o *Handler*. O Handler analisa as mensagens recebidas, e encaminha-as, de acordo com o seu propósito, ou para o *Scheduler*, ou para o *Maintainer*. O propósito do Maintainer é para fazer todo o trabalho de background; ou seja, preparar todas os recursos necessários para o processo de composição (criação, eliminação e edição de elementos). Este tipo de tarefas pode exigir cooperação entre o Maintainer e o *Run-Time Player*. Os recursos geridos pelo Maintainer são armazenados nas estruturas que ele controla: *Sources* (contém Sources – representação de cada fonte sonora da soundscape), *Contexts* (ver Heurísticas de Composição), *Listener* (estrutura que mantém informação sobre a posição do ouvinte no mundo de jogo) e *Patterns* (tipos de exploração de som, que podem ser associados a cada Source, que o módulo de composição tem em conta no momento de decidir como actuar sobre a soundscape). Estes objetos são criados quando o designer os inicializa através da API, no código de jogo, sendo mantidos em espera, prontos a serem usados sempre que for exigido.

O Scheduler coordena todo o processo de composição. Oferece resposta a pedidos de start/stop relacionados com Sources e Patterns, bem como com alterações no contexto activo. Sempre que é solicitado que uma fonte seja tocada, o Scheduler responde de acordo com a aplicação de um conjunto de heurísticas, que por sua vez levam em consideração o *Current Context* e as fontes actualmente activas no *Contextual Score*. Esta última é uma estrutura que mantém uma categorização das fontes activas de acordo com a sua semântica.

As heurísticas também podem avaliar outros aspectos das fontes sonoras, tais como o agente e a Pattern a elas associados. Conforme o processo de composição vai decorrendo, o Scheduler pode encaminhar ordens para o Run-Time Player, de forma a este renderizar o som de acordo com as heurísticas. No nosso projecto, o Run-Time Player adoptado foi a biblioteca de som FMOD API.

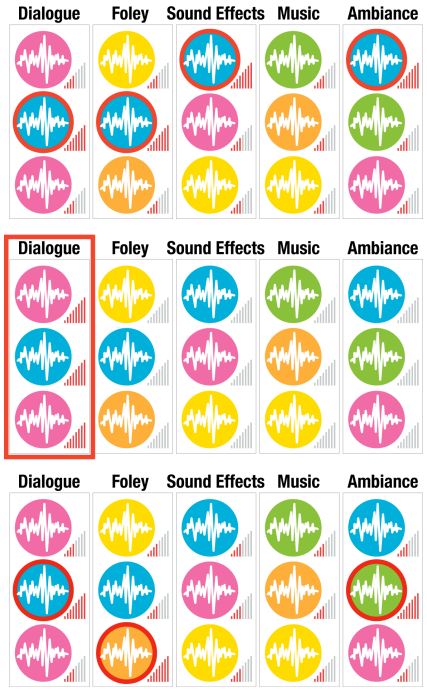

Figura 1 - Exemplo dos diferentes tipos de contexto

É importante referir que, durante o processo de composição, o Scheduler faz uso de três pautas (Scores) diferentes: *Score, Contextual Score, e Over Density Score*. A estrutura chamada *Score* armazena todas as fontes disparadas pela lógica do jogo, e que não terminaram ou foram solicitadas a parar, independentemente do Scheduler ter decidido que deveriam ser audíveis ou não. Esta estrutura permite que o Scheduler tenha, em qualquer momento, uma visão completa da soundscape que foi requisitada pelo jogo. Por seu lado, *Contextual Score* é uma estrutura que armazena apenas as fontes actualmente activas, que fazem parte do Current Context. Na prática, esta é a estrutura que sustenta a soundscape que está realmente a ser ouvida pelo jogador. Finalmente, o *Over Density Score* é uma estrutura que, como o nome sugere, contém as fontes que, apesar de fazerem parte do Current Context, não podem ser ouvidos para não aumentarem demasiado a densidade da soundscape (número de fontes em simultâneo).

Heurísticas de Composição

Com o objectivo de lidar com a natureza dinâmica do meio, foram definidas um conjunto de técnicas para guiar o módulo de composição. Estas técnicas consistem numa lista de práticas comuns em sound design para jogos, que acreditamos serem instrumentais no sentido de alcançar alguns princípios de Acoustic Ecology, nomeadamente, uma soundscape saudável. Este conjunto de técnicas, representado em termos computacionais sob a forma de uma lista de heurísticas, poderá ser modificado ou ampliado. O papel destas heurísticas passa por monitorizar as fontes que a lógica de jogo determina que *deveriam* tocar, e decidir se, e como, essas fontes devem ser tocadas. Estas decisões têm em conta as características dos elementos presentes na soundscape, nomeadamente, a Pattern associada a cada fonte. De uma forma simplificada, podemos dizer que as heurísticas são comportamentos que servem de resposta às Patterns que podem ser associadas às fontes.

As heurísticas que implementámos, nesta fase da investigação, são as apresentadas nos parágrafos seguintes. Naturalmente, este conjunto de heurísticas não permite contemplar a complexidade de todas as situações passíveis de ocorrer num contexto de videojogo, mas serve o propósito de testar a abordagem proposta.

Context – O conceito de contexto serve para distinguir entre sons relevantes num determinado instante e sons não relevantes. Esta heurística serve para dar ênfase aos sons do primeiro grupo, e atenuar ou silenciar os do segundo grupo. Existem 3 tipos de contexto: contexto do tipo *Layer*, onde as fontes dentro de contexto dependem dos layers seleccionados; contexto do tipo *Agent*, onde as fontes dentro de contexto são escolhidas de acordo com o agente a elas associado; e contexto do tipo *Ad-hoc*, onde as fontes dentro de contexto são seleccionadas livremente através do seu nome. Na Figura 2, estão representados os 3 tipos de contexto: *Agent, Layer e Ad-hoc*. Os círculos são fontes que, segundo a lógica de jogo, devem ser tocadas. As cores dos círculos representam agentes, sendo que cada coluna representa um layer. Ao lado de cada círculo está a representação do volume de cada fonte, estando directamente relacionado com o facto dessa mesma fonte estar dentro de contexto, ou não.

Thoughts – Esta heurística oferece uma possível representação do efeito "voz dentro da cabeça",

vastamente usado para representar pensamentos de personagens.

Silence – A noção de silêncio pode ser interpretada e implementada de inúmeras formas. Esta heurística procura oferecer uma possível abordagem, na qual todas as fontes são atenuadas quase na totalidade, exceptuando as fontes de Diálogo e Foley.

Awareness – Em muitos jogos, os game designers recorrem ao som para chamar a atenção dos jogadores para certos elementos relevantes. Esta heurística oferece esse efeito ao dar relevo à fonte sonora que vai ser tocada durante um período de tempo, voltando a soundscape ao seu estado normal uma vez findado esse intervalo.

Dialogue – Esta heurística visa dar sempre relevo a fontes sonoras que representem elementos de diálogo, silenciando as fontes de Foley e SFX, embora deixando presentes as fontes de Ambiance e Music (ainda que bastante atenuadas).

Footsteps – Footsteps são um tipo de Foley bastante importante, pois dá personalidade e distingue as diferentes personagens, além de ser importante em certos aspectos de jogabilidade. Quando a movimentação do jogador não produz feedback sonoro, a sensação de imersão do jogador pode perder-se. Como tal, esta heurística não permite que as fontes que lhe estejam associadas tenham o seu volume atenuado abaixo de um certo valor.

Contextual Music – Música contextual ajuda a caracterizar diferentes contextos, espaços e outros elementos dos jogos. Analogamente, esta heurística, tal como a anterior, não permite que as fontes associadas a ela tenham o seu volume mais atenuado que um certo valor.

Achievement/Failure/No Can Do – *Achievement*, *Failure* e *No Can Do* são tipos de explorações de SFX, vastamente usados em jogos [1,2]. Devido à importância semântica que costumam ter, esta heurística modera a soundscape de forma a que eles sejam sempre ouvidos, com o objectivo de não deixar o jogador "perder" informação importante para a jogabilidade.

Encoded-Embodied – Inspirada pelo modelo psico-acústico apresentado por Walter Murch [13], onde é apresentado um espectro de "cores de som", esta heurística mapeia cada uma das 5 cores do modelo para as 5 Layers que usamos [15]. Seguindo as leis do modelo, e com o objectivo de manter uma densidade de soundscape aceitável, esta heurística não permite mais do que 2 fontes de cada cor (layer) a actuar em simultâneo.

Prova de conceito – Blindfold

Depois de concluído o design e a implementação do módulo de composição dinâmica, foi verificada a exequibilidade do sistema num cenário de jogo. O cenário de jogo escolhido foi o Blindfold, um jogo de aventura *audio-only* (sem componente gráfica) previamente desenvolvido por nós. O jogo consiste numa aventura enigmática que tenta evocar emoções no jogador, sendo a audição o único sentido que pode guiar os jogadores durante a experiência. A Figura 3 apresenta um screenshot com fins de debugging, aqui apresentado apenas para uma facilitar a caraterização do cenário de jogo. O jogo Blindfold foi desenvolvido em XNA [12], tendo a implementação da sua componente sonora sido feita originalmente através de XACT [11].

A experimentação de exequibilidade consistiu então na reimplementação do Blindfold, mas desta vez com toda a implementação e renderização de som feita através do módulo proposto neste trabalho. É importante referir que o objectivo desta experiência não era uma comparação directa de resultados, mas sim testar a exequibilidade da abordagem proposta, assim como verificar os comportamentos de cada uma das heurísticas. O jogo Blindfold, devido à sua variedade de explorações de som, revelou-se um cenário ideal para testar o potencial oferecido pelas diferentes heurísticas implementadas.

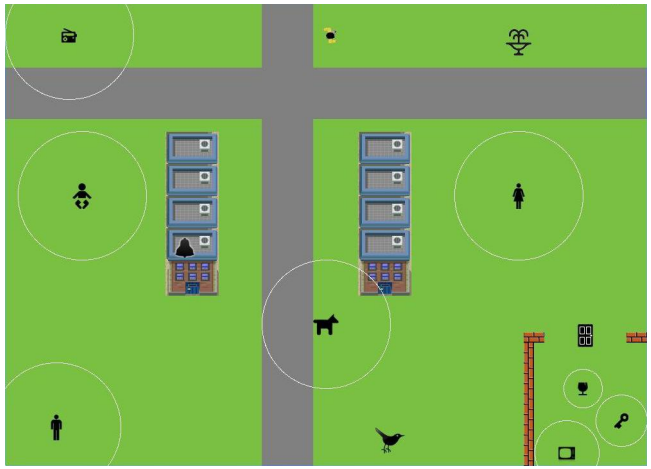

Figura 2 - Screenshot de debug do jogo Blindfold

A título de exemplo, a heurística *Context*, na qual é dado relevo a algumas fontes, e são atenuadas as restantes, foi particularmente útil para ajudar a distinguir diferentes momentos de jogo. Uma delas verificou-se na situação, durante o jogo, em que os jogadores recolhem um bebé que se encontra abandonado no mundo de jogo. Após esse evento, e apenas exigindo uma linha de código para activar o contexto necessário, a soundscape passa a dar relevo aos sons do bebé, e da mãe que chora desesperada à sua procura. Desta forma, o jogador recebe uma pista sobre a tarefa a fazer de seguida, ao mesmo tempo que a composição o ajuda no processo de eco-localização da mãe.

A heurística de *Awareness* também se revelou bastante útil nesta experimentação. Esta heurística permitiu que, por exemplo, sempre que novos elementos entrassem dentro do raio de audição do jogador, estes elementos tivessem maior relevância na soundscape durante alguns segundos. Já as heurísticas de *Dialogue*, e *Thoughts*, permitiram, respectivamente, garantir que as linhas de diálogo presentes no jogo tivesse sempre relevo na soundscape, e permitir que mais tarde pudessem ser recordadas pelo jogador, sob a forma de pensamento. Finalmente, importa referir que a heurística relativa aos SFX (*Achivement/Failure/No Can Do)*, teve também extrema utilidade. Neste jogo, que é *audio-only*, o som tem o papel de dar todo o feedback acerca das acções do jogador, pelo que é fundamental garantir que este tipo de sons com alto valor semântico seja sempre ouvido.

Em função desta nossa experiência de aplicação das heurísticas, argumentamos que se confirmou a exequibilidade da solução proposta neste exercício em particular, assim como saiu reforçada a ideia de que as heurísticas desenvolvidas podem realmente ser úteis em diversas situações de jogo. Defendemos que inúmeras situações de jogo exibiram o proveito retirado das explorações de som já cobertas pelas heurísticas implementadas, e que permitiram realçar aspectos importantes da jogabilidade em diferentes momentos.

Em especial, e na linha do que já foi referido, a heurística de Context permitiu que de forma quase instantânea, a soundscape pudesse ser adaptada a contextos diferentes que se iam moldando de acordo com as acções do jogador.

Além disso, este exercício demonstrou que é possível alcançar um comportamento acústico bastante interessante através de uma implementação significativamente mais leve, usando a API desenvolvida. O facto do módulo de composição poupar o trabalho de implementar vários comportamentos torna o código bastante mais limpo, devido aos comportamentos relacionados com som que "desaparecem" do código da lógica do jogo, por já estarem codificados nas heurísticas do módulo de composição. Por outro lado, esta separação entre lógica de jogo, e a lógica de comportamento do som, leva a uma melhor legibilidade do código do próprio jogo.

Trabalho Futuro

Como referimos, o conjunto de heurísticas aqui apresentadas é ainda reduzido do ponto de vista das situações de jogo que permite de cobrir. Nesta fase, serviu essencialmente para suportar a investigação, incluindo através de experimentação. Como trabalho futuro, temos intenções de actualizar este conjunto de heurísticas, e fazer a respetiva avaliação de comportamento, nomeadamente envolvendo utilizadores.

Também estamos particularmente interessados em estudar o impacto desta proposta como ferramenta de prototipagem rápida, em adição, ou como alternativa, à sua utilização para implementação. A utilização em cenários de prototipagem rápida poderia beneficiar bastante da construção de uma interface gráfica que permitisse a alteração em tempo real de alguns parâmetros relacionados com as heurísticas, assim como o visionamento de alguns dados estatísticos relacionados com a composição em curso.

Porém, devido ao facto do módulo, actualmente, apenas conseguir ter um contexto activo de cada vez, seria extremamente valioso prototipar uma nova abordagem que guardasse contextos em lista de espera, ou inclusive que existisse suporte para mais que um contexto activo simultaneamente. Finalmente, seria também interessante avaliar o impacto da criação de um novo canal de comunicação entre o módulo de composição e o jogo. Desta forma, a lógica de jogo deixaria de ser "surda", e poderíamos testar abordagens nas quais o processo de composição influenciaria o desenrolar da lógica do jogo.

Conclusões

Neste artigo apresentámos uma proposta para o problema composição dinâmica de som em videojogos, começando com a caracterização de uma das maiores dificuldades do sound design para jogos: a natureza dinâmica do meio. Durante um jogo, as acções do jogador podem originar eventos e situações imprevisíveis, dificilmente cobertas na totalidade pelo sound designer no processo de design a priori. Apesar de serem reconhecidas as potencialidades das ferramentas de Middleware para lidar com este problema, elas são tipicamente demasiado caras e complexas, para constituírem uma solução viável para pequenos criadores de jogos.

Defendemos também uma abordagem holística para o sound design para jogos, fundada na Acoustic Ecology,

que propicia uma apreciação global do relacionamentos entre entidades da soundscape, que também se estendendo ao jogador. O conceito de soundscape saudável foi aqui usado para demonstrar um entendimento da diferença entre uma composição que mantém o seu valor comunicacional, e uma mera sobreposição de fontes que simplesmente se tornaram activas num determinado momento. Assim, uma das contribuições do trabalho apresentado neste artigo é a interpretação de conceitos associados à teoria de Acoustic Ecology, para o contexto do sound design para jogos. Esta transcrição de conceitos e princípios reflecte-se não só em aspectos específicos, como a escolha e definição de heurísticas, mas também nos princípios de alto nível que guiam a arquitectura da solução, e a metodologia de composição.

Também valorizámos a procura de soluções acessíveis a todos os tipos de criadores de jogos, incluindo por via da simplificação da implementação de explorações de som em jogos. O desenho e implementação de uma API de especificação de soundscapes é outra das contribuições do trabalho, que vai nesse sentido. É a especificação feita através desta API que permite ao módulo de composição tirar partido das heurísticas, ao permitir uma avaliação das características dos elementos presentes na soundscape. Princípios como acessibilidade e legibilidade foram alguns dos aspectos que guiaram a sua especificação. Apesar de, mesmo assim, exigir algum conhecimento de programação, através desta API pretendemos fazer uso de uma nomenclatura e sintaxe simples, fazendo uso de termos e conceitos próximos da área de sound design. Desta forma, estão criadas condições que permitam que, ao olhar para implementações de som feitas com a API que criámos, possa ser possível esboçar uma ideia das intenções do designer que executou a implementação.

Ainda, resulta deste trabalho uma contribuição em forma de lista de técnicas de composição de soundscapes. O objectivo maior destas técnicas é serem instrumentais para a manutenção da "saúde" da soundscape. A sua especificação foi informada por uma análise a certas prácticas comuns e validadas, usadas actualmente em sound design para jogos, assim como em princípios da teoria de Acoustic Ecology. Estas técnicas, embora ainda careçam de avaliação formal, são uma primeira abordagem a um conjunto de heurísticas que pode ser melhorada e aumentada, procurando dar uma resposta cada vez mais eficaz à imprevisibilidade dos cenários de jogo.

Finalmente, e como contribuição principal deste estudo, apresentámos um módulo que executa a composição dinâmica em run-time. Ao servir-se das técnicas de composição embutidas, o módulo reduz a necessidade de prever e codificar a imensidão de respostas sonoras a todas as situações de jogo que possam surgir. A responsabilidade de moderação da soundscape é, assim, transferida do código de jogo, para o módulo de composição, que o executa autonomamente, e em tempo real.

Argumentamos que o trabalho desenvolvido tem potencial para servir como base para futuras experiências, nomeadamente com o objectivo de melhor avaliar as heurísticas, e o processo de composição. Consideramos que existe potencial para que uma abordagem deste género, e que com o necessário refinamento e trabalho de investigação adicional, poderá vir a constituir realmente uma

alternativa em termos de implementação de som em jogos.

Referências

[1] Alves, V. and Roque, L. 2011. A Deck for Sound Design in Games - Enhancements based on a Design Exercise. In *Proc. of ACE 2011*, Lisboa, Portugal.

[2] Alves, V. & Roque, L. 2012. SoundInGames.com – Sound Design in Games. http://www.soundingames.com

[3] Brandon, A. 2007. Audio Middleware Mix: Professional Audio and Music Production, http://www.mixonline.com/basics/education/audio_audio_middleware/.

[4] Brandon, A. 2007. Audio Middleware, part 2 Mix: Professional Audio and Music Production, http://www.mixonline.com/basics/education/audio_audio_middleware_part/.

[5] Brandon, A. 2007. Audio Middleware, part 3 Mix: Professional Audio and Music Production, http://www.mixonline.com/basics/education/audio_audio_middleware_part_2/.

[6] Chan, S.H., Natkin, S., Tiger, G. and Topol, A. 2012. Extensible Sound Description in COLLADA: A Unique File for a Rich Sound Design. In *Proc. of the Advances in Computer Entertainment,* 2012 Springer.

[7] Collins, K. 2008. Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design. MIT Press.

[8] Ekman, I. 2008. Psychologically Motivated Techniques for Emotional Sound in Computer Games. In *Proc. of the 3rd AudioMostly Conference*, Piteå, Sweden, ACM, 20-26.

[9] Hevner, A. R. and Chatterjee, S. 2010. Design research in information systems. New York; London: Springer.

[10] Kastbauer, D. 2010. Audio Implementation Greats #2: Audio Toolsets [Part2] Designing Sound, http://designingsound.org/2010/01/audio-implementation-greats-2-audio-toolsets-part-2

[11] Microsoft, n.d. XACT Overview, Microsoft Developer Network, http://msdn.microsoft.com/en-us/library/cc308030

[12] Microsoft, XNA Developer Center Microsoft, http://msdn.microsoft.com/en-us/centrum-xna.aspx

[13] Murch, W. 2005. Dense Clarity Clear Density, The Transom Review, transom.org, http://transom.org/?page_id=7006

[14] Open Sound Control, Introduction to OSC Open Sound Control, http://opensoundcontrol.org/introduction-osc

[15] Peck, N. 2001. Beyond the library: Applying filmpostproduction techniques to game sound design - lecture. In *GDC 2001*, San Jose, CA, 20-24 March.

[16] Peerdeman, P. 2010. Sound and Music in Games, Vrije Universiteit, Amsterdam.

[17] Schafer, R.M. 1993. The Soundscape: Our Sonic Environment and the Tuning of the World. Inner Traditions/Bear.

[18] Truax, B. 2001. Acoustic Communication. Ablex.

[19] Vaishnavi, V. and Kuechler, W. 2004. Design Science Research in Information Systems. http://desrist.org/desri.

[20] Went, K., Huiberts, S. and Tol, R. V., 2009. Game Audio Lab - An Architectural Framework for Nonlinear Audio in Games. In *Audio Engineering Society Conference: 35th International Conference: Audio for Games*, London, United Kingdom, AES.