

Geração procedural de coreografias para jogos de dança

Alice Rocha Abreu^{1*}João Ricardo Bittencourt¹Rossana Baptista Queiroz¹Vinícius Jurinic Cassol²¹Universidade do Vale do Rio dos Sinos, Escola Politécnica, Brasil²Baby Owl Studios, Brasil

RESUMO

Este artigo apresenta as etapas de desenvolvimento de uma proposta de geração procedural de coreografias para jogos de dança. Unindo movimentos criados previamente por um coreógrafo com redes neurais artificiais, busca-se oferecer uma nova alternativa para o desenvolvimento de jogos desse gênero, tornando o processo mais barato e mais acessível para desenvolvedores independentes que desejem iniciar uma produção semelhante às já existentes no mercado.

Palavras-chave: Ritmo, dança, coreografia, animação procedural.

1 INTRODUÇÃO

O gênero musical de jogos é consagrado no mercado, com títulos como *Just Dance*¹, que vem crescendo com o tempo e atualmente possui versões anuais de seus jogos, e também títulos da franquia *Dance Central*², exclusiva da *Microsoft*. São jogos que atraem jogadores de todas as idades, desde jovens adolescentes até adultos, e com isso, investem em um repertório diverso, com diferentes artistas, bandas e gêneros musicais presentes em cada nova versão lançada no mercado. Além disso, depois do lançamento, os jogos continuam recebendo atualizações com novas músicas, o que mantém os usuários interessados por um bom tempo.

Porém, uma limitação que esse tipo de jogo encontra ainda é o fato da coreografia apresentada para cada música se manter sempre igual. Considerando que o jogador pode gostar do fato de querer repetir a mesma música diversas vezes para conseguir melhorar o seu desempenho, por exemplo, isso pode ser visto como algo positivo. Porém, de certa forma, também pode diminuir o fator *replay* do jogo, visto que o jogador que já conseguiu atingir o seu máximo em determinada coreografia provavelmente não voltará para a mesma música, e dessa maneira, aos poucos pode ir deixando o jogo de lado.

A proposta deste trabalho é a criação de um agente inteligente para geração de coreografia procedural para jogos de dança. Dada uma determinada música, o agente deverá gerar diferentes coreografias para ela, de forma que, se fosse utilizado posteriormente em um jogo de dança, poderia agregar a experiência dos usuários aumentando o fator *replay* não só com a adição de novas músicas no jogo através de atualizações como é feito nos jogos já existentes no mercado, mas também, na inovação de termos diferentes experiências com uma mesma faixa.

2 TRABALHOS RELACIONADOS

Nesta seção são abordadas referências teóricas utilizadas como base durante o desenvolvimento deste projeto. Estas referências são em sua maioria técnicas e estudos já consolidados nos eixos de dança e geração de conteúdo procedural, assuntos pesquisados de forma ampla para determinar quais alternativas seriam mais adequadas para o desenvolvimento do projeto.

*e-mail: alliceabreu@gmail.com

¹<http://www.justdancenow.com/>

²<http://www.harmonixmusic.com/games/dance-central/>

2.1 Análise Laban de movimento

Um ponto importante sobre geração procedural é que nem tudo envolvido no processo deve ser procedural. Partindo desse princípio, é necessário ter alguma forma de categorizar os movimentos. Optou-se pelo estudo da análise de movimento Laban como opção de abordagem eficaz para categorizar os movimentos.

A análise é uma teoria proposta por Rudolph von Laban, coreógrafo, bailarino e considerado um dos maiores teóricos da área de dança do século XX, sendo inclusive referenciado em algumas obras como o "pai da dança"[1]. A análise Laban de movimento, ou Labanálise, como também é chamada, é uma das duas ramificações sugeridas para melhor organização no estudo do Sistema Laban. Na outra ramificação, temos a labanotação, que busca fazer o registro exato da coreografia, de certa forma, criando uma partitura dos movimentos realizados.

Com a labanálise, há um foco maior na qualidade do movimento, registrando elementos específicos utilizados de cada um[1]. Laban define quatro conceitos que podem influenciar qualquer tipo de movimento: espaço, tempo, peso e fluxo[2]. O espaço pode ser caracterizado de duas formas: direto ou indireto. O tempo pode ser classificado como acelerado ou desacelerado. O peso pode ser analisado como leve ou forte. Esses três conceitos representam os aspectos sensoriais de caráter mecânico do movimento. O fluxo é o único conceito que não se origina em relação a gravidade e sistemas mecânicos, é analisado por seu aspecto livre ou controlado.

Através da combinação dos três conceitos descritos, com exceção do fluxo, Laban define o que chama de verbos de ação. Se o movimento tiver uma modificação de peso, será chamado de pontuar ou torcer. Modificado pelo eixo de espaço, tem-se tatear ou deslizar[2]. Caso seja modificado pelo tempo, tem-se os verbos de ação pressionar ou sacudir. O verbo de ação socar pode ser definido pelas seguintes características: peso forte, espaço direto e tempo acelerado. Ao alterar os valores dos três conceitos para o oposto, ou seja, peso leve, espaço indireto e tempo desacelerado, tem-se movimentos que se encaixam na definição de flutuar. Com as diferentes combinações possíveis com esses três fatores, existem ao todo oito verbos de ação que podem representar visualmente e categorizar todos os movimentos que se encaixam nesses requisitos.

Os conceitos definidos por Laban são utilizados na criação da ferramenta para edição e categorização de movimentos e foram escolhidos por sua ampla influência e importância no cenário de estudo da área de dança. Fazendo parte da aprendizagem de profissionais da área desde a graduação e com um amplo número de laboratórios espalhados pelo mundo dedicados a estudar e amplificar a divulgação das teorias propostas por Laban, esta se mostra uma abordagem sólida por sua importância e também amplitude no cenário profissional.

2.2 Geração procedural de coreografias

A geração procedural de coreografias, também abordada em alguns artigos como geração procedural de dança é um tópico que vem crescendo e sendo mais abordado recentemente em artigos da área de computação gráfica. Originadas da geração de movimentos de forma procedural, surgem novas abordagens tentando misturar técnicas já existentes e aplicadas de geração de movimentos com o ritmo, e a partir disso surgem os trabalhos de geração de dança.

Em sua maioria, os trabalhos que buscam implementar esse tipo de algoritmo utilizam da cinemática inversa como pilar principal. O conceito de cinemática inversa baseia no princípio de que sabendo os valores das coordenadas de uma articulação é possível calcular as coordenadas do extremo do membro[3]. Ou seja, se sabe-se as coordenadas atuais da posição do cotovelo direito de um modelo 3D, pode-se calcular quais as coordenadas da mão direita desse mesmo modelo. Essa é uma técnica que está começando a ser utilizada por alguns jogos de grandes estúdios ou até mesmo projetos independentes para dar maior fluidez às animações. *Uncharted 4*³ pode ser citado como um exemplo de jogo que investiu na técnica para trazer um novo sistema de animações em seu último lançamento[4].

Usando do princípio da cinemática inversa e suas aplicações em diferentes ritmos de música, como salsa e rumba, deixando fixos apenas a posição do pé e o tempo de mudança de peso do movimento, já é possível iniciar uma geração procedural de dança que seja aplicável a esses dois ritmos, como mostrado por Elizabeth e Geoffrey Matthews em uma implementação de 2011[5]. Já outras abordagens, como esta, proposta em 2006[6], traz opções de ritmo mais amplas, não focando apenas na dança. Nesse caso, temos uma proposta de uma ferramenta em que o artista pode ajustar parâmetros como *timing* e "exagero" de um determinado movimento, dando ao diretor maior identidade ao projeto de animação que ele está conduzindo.

É interessante notar a evolução que temos neste tipo de trabalho desde 2005, quando o jogo *Spore*⁴ apresentou criaturas que dançavam proceduralmente na *GDC (Game Developers Conference)* daquele ano. Em 2006 trabalhos na mesma área já começaram a surgir e desde então diversas técnicas diferentes e propostas com diferentes eixos foram apresentadas. Desde algumas mais focadas em ritmos específicos como salsa e rumba, até propostas mais recentes em que com base em movimentos reais capturados através de *Kinects*, usando *deep learning*, uma rede neural que após 48 horas de treinamento e ajustes, consegue fazer uma dança totalmente funcional[7].

3 VISÃO GERAL DO PROJETO

A geração procedural de coreografias é dividida em cinco etapas principais.

A primeira, ilustrada pelos dois primeiros retângulos da Figura 1 é uma **ferramenta para criação, edição e categorização de movimentos** que é utilizada por um coreógrafo. A ideia é que com a ajuda de um profissional capacitado na área de dança, seja possível obter um banco de movimentos mais realistas e devidamente analisados, fazendo com que a coreografia que será gerada utilizando esses movimentos tenha um melhor resultado técnico, visto que não foi analisado apenas por desenvolvedores e *game designers*, mas por alguém ligado a área de dança.

Na segunda etapa, ilustrada pelo terceiro retângulo da Figura 1, tem-se a necessidade de desenvolver o **tratamento e análise de áudio**. Isso é necessário, visto que um dos objetivos principais do projeto é que seja possível gerar coreografias para diferentes músicas. Então, é necessário que o áudio dessas diferentes músicas seja analisado a fim de que tenhamos a coreografia sempre acompanhando o ritmo da música escolhida. Para isso, após o usuário informar qual a música deseja utilizar no momento, a análise do áudio dessa música é realizada pela biblioteca *Essentia*⁵, com o objetivo de identificar quais pontos da música uma batida ocorre, permitindo que a coreografia gerada posteriormente acompanhe sempre o ritmo dela.

Depois do usuário escolher determinada música e a análise da biblioteca *Essentia* já realizada, temos a terceira etapa, que também é representada por uma tarefa a ser realizada pelo profissional da

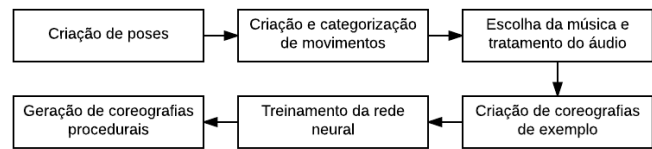


Figura 1: Arquitetura do projeto. Fonte: da autora.

área de dança que está utilizando a ferramenta: **a criação de coreografias de exemplo**. Representada pelo quarto retângulo da Figura 1, nesta etapa, usuário cria uma coreografia exemplo para determinada música que ele escolher. Essa coreografia, criada com a utilização de movimentos já cadastrados anteriormente, é utilizada como exemplo para o treinamento da rede neural do projeto.

Com as coreografias de exemplo já criadas pelo profissional, é possível realizar quarta etapa, o **treinamento da rede neural**. Com o objetivo de deixar a taxa de erros a menor possível, ou em um nível aceitável, o treinamento é realizado com a análise dos movimentos utilizados nas coreografias de exemplo e suas características juntamente com as informações de áudio das músicas utilizadas nessas coreografias. O conjunto de exemplos passados para o treinamento é um arquivo único com as informações de todas as coreografias base utilizadas, com cada linha contendo o instante em que uma batida ocorre na música e que movimento é executado naquele momento.

Depois de treinada, a rede neural finalmente pode cumprir seu objetivo principal, que é **gerar a coreografia** para a música escolhida pelo usuário. Tendo como entradas informações como o tempo atual da música, se ocorreu uma batida naquele instante e movimentos anteriores realizados pelo modelo 3D, as saídas da rede são equivalentes aos parâmetros definidos por Laban, espaço, tempo e peso. As informações obtidas na saída da rede são utilizadas para a busca de movimentos que tenham essas características e que portanto, serão os próximos a serem utilizados na coreografia.

4 TECNOLOGIAS E DESENVOLVIMENTO

A *engine Unity*⁶ foi definida como a que seria utilizada para o desenvolvimento projeto. Foi escolhida principalmente por sua ampla documentação e assets disponíveis gratuitamente, fatores fundamentais para o desenvolvimento *indie*, principal foco desse projeto. Os modelos utilizados para a visualização de movimentos e poses na ferramenta para o coreógrafo foram obtidos através da plataforma *Mixamo*⁷, da *Adobe*⁸ e a biblioteca que foi utilizada para tratamento de áudio foi a *Essentia*⁹, biblioteca desenvolvida em C++ amplamente utilizada no âmbito científico para tratamento e manipulação de áudio.

4.1 Ferramenta para criação, edição e categorização de poses e movimentos para coreografias

A ferramenta foi desenvolvida para seguir o seguinte princípio: se tratando de geração procedural, nem tudo pode ser procedural. Tinha-se a necessidade de categorizar os movimentos que o dançarino realiza para que assim seja possível criar uma lógica sobre isso, que defina em que ordem e momentos eles são executados.

Por se tratar de uma ferramenta destinada a um tipo de usuário que não utiliza ferramentas complexas em seu computador pessoal, buscou-se trazer a simplicidade como fator principal no desenvolvimento da interface para o usuário que fará uso da ferramenta. Utilizando em sua maioria ícones intuitivos, como uma lixeira para

³<https://www.unchartedthegame.com/pt-br/>

⁴<http://www.spore.com>

⁵<http://essentia.upf.edu/documentation/>

⁶<https://unity3d.com/pt>

⁷<https://www.mixamo.com/>

⁸<http://www.adobe.com/br/>

⁹<http://essentia.upf.edu/documentation/>

indicar o botão de exclusão e um disquete para indicar o botão de salvar, ícones já tradicionais em suas funções, a interface torna-se simples e de fácil entendimento para o usuário. Assim, ele pode focar na sua tarefa principal, que é o cadastro e categorização de movimentos e não perde tempo se dedicando a aprender como usar o software.

4.1.1 Poses

Uma pose é uma determinada posição em que um dançarino fica quando se mantém parado. Também pode-se entender como um *frame* de uma animação. É com a interpolação entre duas poses diferentes que tem-se um movimento. Dentro da ferramenta, pode-se definir que uma pose seria composta por uma lista de *bones*, o que seria equivalente a uma representação de um osso humano em um modelo 3D. Cada *bone* contém um identificador e um valor que indica sua posição naquela determinada pose.

Como identificador, optou-se por utilizar valores que podem ser obtidos através da própria *Unity 3D*. Utilizando-se da enumeração *HumanBodyBones*¹⁰, é possível obter um valor inteiro correspondente ao *bone*. Chamando *HumanBodyBones.Hips*, tem-se o identificador em valor inteiro do *bone* do quadril, *HumanBodyBones.LeftUpperLeg* do *bone* superior da perna esquerda e assim por diante. A vantagem dessa abordagem é que o identificador se torna útil para qualquer modelo que tenha a mesma topologia humanóide e que esteja sendo utilizado dentro da *Unity 3D*.

4.1.2 Criação de poses

Processo ainda em desenvolvimento, a ideia é que tendo a visualização de um modelo 3D na tela, o usuário possa selecionar determinado membro e arrastá-lo para outra posição, ou rotacioná-lo. Dessa forma, ele tem total liberdade artística para criar as poses que julgar mais apropriadas e/ou criativas e salvá-las quando julgar que terminou o processo de criação. A ferramenta ainda fornece um *feedback* para o usuário caso algum membro esteja com uma posição e/ou rotação inválida em relação aos outros, para que poses que não poderiam ser executadas por dançarinos reais não possam ser salvas.

Por enquanto, as poses são criadas através de *Animation Events*¹¹ da engine *Unity 3D*, em que, utilizando de animações obtidas na plataforma *Mixamo*, alguns *frames* são setados com um evento que chama uma função para salvar as posições dos *bones* do modelo em arquivos texto no momento em que a animação que esses *frames* fazem parte for executada.

4.1.3 Edição de poses

Na edição de poses, o usuário pode navegar por todas as poses já salvas na ferramenta, visualizando-as. Caso deseje mudar alguma posição de algum *bone*, ou fazer uma pequena alteração, basta clicar em um botão e uma tela semelhante a tela de criação de poses é exibida.

4.1.4 Exclusão de poses

Processo realizado na mesma tela de edição de poses, conforme pode visto na Figura 2, a exclusão de poses é uma tarefa que necessita de certa atenção do usuário.

O usuário pode percorrer todas as poses que possui salvas, visualizando cada uma na tela. Caso deseje excluir determinada pose, basta clicar em um botão específico para isso, representado graficamente por uma lixeira. Porém, a ferramenta faz uma verificação se essa pose é utilizada em algum movimento também já salvo. Caso seja, informa ao usuário em quais movimentos essa pose é utilizada e o avisa que a exclusão da mesma acarreta na exclusão dos movimentos em que ela é referenciada. Esta verificação é realizado para evitar comprometimento das informações salvas.

¹⁰<https://docs.unity3d.com/ScriptReference/HumanBodyBones.html>

¹¹<https://docs.unity3d.com/Manual/AnimationEventsOnImportedClips.html>

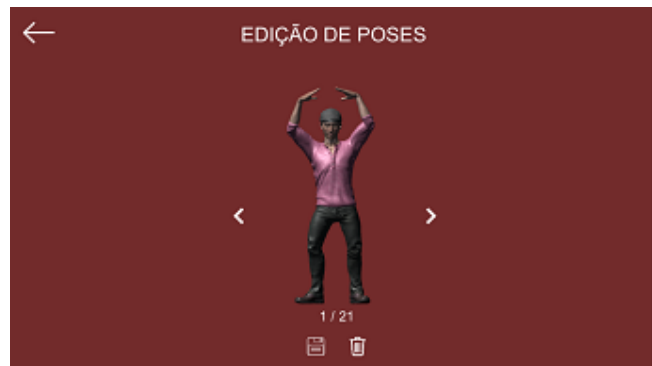


Figura 2: Tela de edição de poses da ferramenta. Fonte: da autora.

4.1.5 Movimentos

Um movimento é composto por duas poses. Além disso, possui quatro atributos que podem ser alterados, que são: tempo, espaço, peso e dificuldade. Os três primeiros são utilizados seguidamente no meio da dança para categorizar movimentos de acordo com os princípios da Labanálise. Já o último, pode ser definido também pelo profissional que está cadastrando os movimentos a fim de oferecer a possibilidade posterior de criar coreografias com diferentes dificuldades.

4.1.6 Criação de movimentos

A criação de movimentos, ilustrada na Figura 3, é realizada através da combinação de duas poses distintas. O usuário pode escolher, entre as poses já salvas, duas poses diferentes e assim criar um movimento. Além da pose inicial e final, algumas configurações do movimento devem ser informadas, para melhor categorização deste. Desta forma, é mais simples para os desenvolvedores que utilizarem as informações geradas tratá-las posteriormente. Caso um movimento de igual aparência já exista, o usuário é informado que não é possível cadastrar um movimento duplicado.



Figura 3: Tela de criação de movimentos da ferramenta. Fonte: da autora.

4.1.7 Edição de movimentos

Na edição de movimentos, o usuário pode visualizar todos os movimentos que já possui salvos e alterar suas características. Caso seja um movimento que possui categoria x, pode alterar para a categoria y e assim por diante.

4.1.8 Exclusão de movimentos

A exclusão de movimentos é um processo realizado de forma muito simples. Em uma tela específica, o usuário pode visualizar todos os

movimentos que já salvou e excluí-los caso deseje. Como nenhum outro dado depende das informações salvas referentes aos movimentos, apenas uma janela de confirmação é exibida para o usuário e após ele confirmar a ação, o movimento é excluído.

4.2 Tratamento de áudio

Para o tratamento de áudio, foi escolhida a biblioteca gratuita *Essentia*[8]. Desenvolvida em C++ e disponível sob a licença *Affero GPLv3* ou sob licença proprietária caso solicitado, *Essentia* é uma biblioteca muito utilizada no meio acadêmico devido a sua grande robustez e a variedade de funcionalidades disponibilizadas. Além disso, já vem sendo utilizada também na indústria, como por exemplo, no jogo *Crypt of the NecroDancer*¹². Escolhida para esse projeto principalmente por ser uma opção gratuita e muito completa, a biblioteca *Essentia* possui diversos algoritmos que podem ser utilizados para diferentes fins, desde *audio loaders* até algoritmos que detectam o quão "dançável" é determinada música [8].

Um dos algoritmos disponibilizados pela biblioteca é o *BeatTrackerMultifeature*. Este algoritmo combina cinco diferentes formas de detecção de batidas na música e leva em consideração a maior concordância entre eles para então considerar que uma batida realmente aconteceu [8]. Esse algoritmo pode ser utilizado através de um arquivo executável disponibilizado no próprio site da biblioteca. Informando o caminho para o arquivo de música a ser analisado e o caminho desejado para um arquivo txt de saída, o algoritmo, ao ser executado, analisa toda a música e coloca no arquivo txt os tempos em segundos, em que ele aponta que há uma batida na música. Tratando os dados do arquivo txt, temos como definir ações para cada vez que uma batida ocorrer na música.

Seguindo os princípios de funcionamento desse algoritmo da biblioteca, foi desenvolvido um *asset* para a *engine Unity* para tratamento de áudio. Nesse *asset*, o usuário, após utilizar o *prompt* de comando do seu sistema operacional para executar o arquivo executável da biblioteca, informa no *inspector* da interface da *Unity* qual o caminho do arquivo txt gerado e qual o arquivo de música que esse arquivo corresponde, sendo que este último deve estar na pasta de projeto. Com essas informações setadas, o usuário pode chamar uma função de um *script* do *asset* para definir que quando uma batida acontecer, o que ele determinar deve acontecer.

Esse *asset* é utilizado na etapa de análise de áudio do projeto. Depois que o usuário informa a música escolhida, o algoritmo é acionado para analisar o áudio da música e gera um arquivo txt contendo os instantes em segundos em que temos batidas na música. Essas informações serão utilizadas posteriormente como entradas na rede neural que analisando esses e outros parâmetros, construirá uma coreografia para a música informada.

5 RESULTADOS PRELIMINARES

A ferramenta de criação, edição e categorização de movimentos encontra-se em desenvolvimento atualmente, com a opção do usuário criar movimentos (como pode ser visto na Figura 3 na sessão 4), editar suas características e excluí-los. Quanto às poses, o usuário pode visualizar e deletar poses já existentes na ferramenta, como mostrado na Figura 2 na sessão 4. Futuramente, irá ser desenvolvido a opção do coreógrafo poder criar as poses que desejar rotacionando os membros do modelo 3D, dessa forma, o coreógrafo poderá criar as poses e movimentos de acordo com a sua preferência para o projeto em que está trabalhando.

Com o tratamento de áudio já totalmente concluído, foram realizados testes iniciais com a geração de coreografias simples.¹³ Acessando os movimentos já criados pelo coreógrafo utilizando a versão disponível da ferramenta, a cada vez que uma batida ocorre em determinada música, um movimento é sorteado para ser o próximo a ser executado pelo modelo 3D exibido na tela. Dessa forma,

já é possível observar o funcionamento básico da interpolação entre um movimento e outro e também observar como o modelo se comporta em relação ao ritmo da música, etapa fundamental para validar o desenvolvimento.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Este artigo apresenta uma proposta de geração procedural de coreografias para jogos de dança. Com o objetivo principal de oferecer a possibilidade de gerar coreografias para jogos desse gênero de forma barata financeiramente, sem a necessidade de utilização de processos caros, como a captura de movimentos, por exemplo. O projeto ainda está em desenvolvimento, sendo que até o momento as etapas de criação de movimentos e poses foram desenvolvidas e juntamente com a análise de áudio, também já implementada, testes de geração de coreografias simples, ainda sem o uso de inteligência artificial já foram realizados.

As próximas etapas de desenvolvimento incluem uma forma mais sofisticada de criação de poses pelo usuário, processo que no momento ele ainda não participa, pois como mencionado anteriormente na sessão 4, as poses atualmente são criadas através de *frames* selecionados em determinadas animações obtidas através da plataforma *Mixamo* e salvas através de uma função chamada por um *AnimationEvent* da *Unity 3D*. Além disso, com a criação de poses implementada, esse processo também irá tornar o processo de edição de poses mais prático para o usuário, sendo que este, ao acessar uma pose já existente, poderá fazer pequenas mudanças nas posições dos membros de forma prática e rápida, o que não é possível atualmente.

Após essa evolução no processo de criação e edição de poses, a próxima etapa é a possibilidade do usuário criar coreografias de exemplo, tarefa fundamental para posterior treino da rede neural. Após o treinamento da rede e testes de seu funcionamento, as coreografias geradas pela inteligência artificial serão expostas para coreógrafos para que possam ser analisadas e validadas por eles. Ajustes na rede neural serão realizados conforme o *feedback* dos coreógrafos, para que as novas coreografias geradas atendam aos ajustes sugeridos por eles e tenham uma maior validade técnica.

REFERÊNCIAS

- [1] J. Franken, "Reflexões e reflexos do sistema laban/bartenieff na prática docente em dança," *Revista Cena em Movimento*, vol. 1, no. 3, pp. 1–3, 2013.
- [2] H. M. Kopke, "Anima scribens: possibilidades para o desenvolvimento de partituras do movimento animado," Master's thesis, Universidade Federal de Juiz de Fora, 2015.
- [3] R. N. Jazar, *Inverse Kinematics*, pp. 263–296. Boston, MA: Springer US, 2007.
- [4] F. Gugelmin, "Uncharted 4 usa técnicas de cinemática inversa em suas animações." http://games.tecmundo.com.br/noticias/uncharted-4-usa-tecnicas-cinemática-inversa-animacoes_800973.htm. Acessado em: 22-08-2017.
- [5] E. A. Matthews and G. B. Matthews, "Procedural generation of cuban dance motion," *Proceedings of CGAMES'2013 USA*, vol. 00, pp. 293–297, 2011.
- [6] T. Li and J. Chen, "Procedural rhythmic character animation: an interactive chinese lion dance," *Journal of Visualization and Computer Animation*, vol. 17, no. 5, pp. 551–564, 2006.
- [7] L. Crnkovic-Friis and L. Crnkovic-Friis, "Generative choreography using deep learning," in *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016)*, (Paris, France), Sony CSL, Sony CSL, 2016.
- [8] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. R. Zapata, and X. Serra, "Essentia: an audio analysis library for music information retrieval," in *International Society for Music Information Retrieval Conference (ISMIR'13)*, (Curitiba, Brazil), pp. 493–498, 04/11/2013 2013.

¹²<http://necrodancer.com>

¹³<https://youtu.be/f1RPQWVG8ZM>