

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Desenvolvimento de Avatares Personalizáveis para Aplicações Musicais

Marta Raquel Soares Maio

Mestrado Integrado em Engenharia Informática e Computação

Orientador: António Coelho (PhD)

Co-orientador: Fabien Gouyon (PhD)

26 de Julho de 2012

Desenvolvimento de Avatares Personalizáveis para Aplicações Musicais

Marta Raquel Soares Maio

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Doutor António Augusto de Sousa

Arguente: Doutor Luís Gonzaga Mendes Magalhães

Vogal: Doutor António Fernando Vasconcelos Cunha Castro Coelho

26 de Julho de 2012

Resumo

Nos últimos anos, os meios televisivos têm perdido terreno para a Internet. Sendo a Internet um meio bastante mais dinâmico e com o qual o utilizador pode interagir mais diretamente, é normal que isto ocorra. A solução tem sido adaptar a televisão à Internet, começando a existir serviços em pacotes televisivos bastante semelhantes a alguns que podem ser encontrados *online*, nomeadamente serviços de reprodução de música. Neste tipo de aplicações, o canal de comunicação com o utilizador é, maioritariamente, o som, ficando por explorar a vertente da imagem.

Surge, então, a ideia de associar ao som a sua maior forma de expressão: a dança. Combinando estes dois meios de comunicação é possível proporcionar ao utilizador uma experiência mais ativa e intensa com a aplicação.

Desta forma, a presente dissertação incide sobre a criação de uma aplicação em que quem a usa pode escolher uma música e selecionar um modelo humano 3D - avatar - que dança de forma síncrona com o trecho musical, consoante o género musical do trecho escolhido. A aplicação permite a personalização do avatar escolhido, como as suas roupas e cor de cabelo. As potencialidades desta integração de som e imagem são diversas, não só para a televisão, mas também para qualquer reprodutor de música que permita a visualização de imagem. Adicionalmente, meios *online* poderiam permitir a partilha de músicas entre utilizadores e visualização dos avatares envolvidos.

De forma a obter este resultado final, foi desenvolvido um algoritmo que recebe informações de tempos de batidas de um trecho musical e que sincroniza uma animação de dança do avatar com a música.

Abstract

In recent years, television media have lost ground to the Internet. Since the Internet is a much more dynamic environment with which the user can interact more directly, it is normal for this to happen. The solution has been adapting the television to the Internet, beginning to exist services in television bundles very similar to some that can be found online, such as music playback services. In such applications, the communication channel with the user is mostly the sound, leaving the image channel unexplored.

Then, it comes the idea of associating the sound to its highest form of expression: dance. Combining these two media, it is possible to provide the user a more active and intense experience with the application.

Thus, this dissertation focuses on the development of an application that enables users to pick a song and select a 3D human model - avatar - that dances synchronously with the piece of music, depending on the song's genre. The application shall, similarly, allow the customization of the avatar chosen, as their clothes and hair colour. The potential of this integration of sound and image is vast, not only for television, but for any music player that allows image viewing. Additionally, online media could allow the sharing of music between users and the visualization of the avatars involved.

In order to obtain this final result, it was developed an algorithm that receives information of beat time from a piece of music, and that synchronizes a dance animation of the avatar with the chosen song.

Agradecimentos

Gostaria de agradecer a todos os que direta ou indiretamente contribuíram para a concretização desta dissertação.

Um especial obrigada ao meu orientador, Professor António Coelho, por toda a ajuda e tempo dispensados. Ainda gostava de agradecer ao Engenheiro Fabien Guyon e ao Investigador João Oliveira pela disponibilidade e conselhos sobre a plataforma Marsyas.

Queria também agradecer aos meus pais por terem sempre acreditado em mim e aos meus irmãos por todo o incentivo e motivação que nunca deixaram de dar.

Ainda ao Rui..."eu mais".

Por fim, gostava de agradecer aos amigos da FEUP e aos que fiz neste primeiro semestre em Madrid. Foi um prazer passar os últimos cinco anos com estas pessoas incríveis.

Marta Maio

*"Yesterday is history. Tomorrow is a mystery.
But today is a gift, and that is why it's called the present."*

Kung Fu Panda

Conteúdo

1	Introdução	1
1.1	Enquadramento e Motivação	1
1.2	Descrição do Problema	2
1.3	Objetivos	3
1.4	Resultados Esperados	3
1.5	Trabalho Relacionado	3
1.6	Metodologia de Desenvolvimento	4
1.7	Estrutura do Documento	4
2	Estado da Arte	5
2.1	Avatares	5
2.1.1	Designação e Origem	5
2.1.2	Criação	6
2.1.3	Importância e Aplicações	10
2.2	Música	13
2.2.1	Análise Musical - Rastreamento de Batidas	13
2.3	Avatares Dançantes	16
3	Metodologia	25
3.1	Visão Geral	25
3.2	Avatar	26
3.3	Deteção de Batidas	28
3.4	Algoritmo de Movimento com Música	30
3.5	Conclusões	31
4	Implementação do Sistema	35
4.1	Ferramentas	35
4.1.1	Modelação 3D	36
4.1.2	Motor de Jogo	37
4.1.3	Reconhecimento de Informação Musical	37
4.2	Visão Geral da Implementação	38
4.3	Arquitetura	39
4.4	Aplicação Principal	40
4.4.1	Visão Geral da UI	41
4.4.2	Avatar	41
4.4.3	Menus e Cenários	45
4.4.4	Som e Movimento com a Música	48
4.5	Aplicação de envio de batidas	49

CONTEÚDO

4.5.1	Comunicação com a Aplicação Principal	50
4.5.2	Integração com Last.fm	51
5	Conclusões e Trabalho Futuro	53
	Referências	55

Lista de Figuras

2.1	Modelo articulado e correspondente ligação entre as suas articulações [GAO07]	9
2.2	Tomada de perspetiva em ambientes colaborativos [Bai06]	12
2.3	Conceitos de ataque, transitório e início numa nota [BDA ⁺ 05]	14
2.4	Método tradicional de deteção de batidas [BDA ⁺ 05]	15
2.5	Identificador de batidas [GEKO09]	16
2.6	Algoritmo de análise de movimento [NNIY02]	19
2.7	Algoritmo de geração de movimento [NNIY02]	20
2.8	Visão geral da arquitetura do sistema [KWS07]	21
2.9	Visão geral do sistema [ODY ⁺ 08]	22
2.10	Aplicação da análise de gestos topológica [ONG10]	23
2.11	Workflow da metodologia usada na representação e síntese de dança [NL10]	24
3.1	Figuras obtidas através da técnica <i>claymation</i>	27
3.2	Malha e esqueleto que vão compôr o avatar	28
3.3	Transformações geométricas de certos membros do bípede de forma a criar uma animação	29
3.4	Arquitetura do Algoritmo de Deteção de Batidas, [OGMR10]	32
3.5	Solução para a sincronização da animação com a música	33
3.6	Esquema explicativo da sincronização da animação com a música	33
4.1	Exemplo de interface do <i>software</i> Unity	38
4.2	Diagrama explicativo do sistema [OGMR10]	39
4.3	Arquitetura concetual do sistema	40
4.4	Avatares criados para a aplicação	43
4.5	Malha e esqueleto constituintes do bípede	44
4.6	Diagrama de fluxo entre os diversos componentes da aplicação	45
4.7	Interface inicial da aplicação	46
4.8	Cenário do vestíbulo onde ocorre a personalização do avatar	47
4.9	Último cenário da aplicação	48
4.10	Resultado visual da análise musical	49
4.11	Aplicação de envio de batidas	50
4.12	Comunicação entre as diferentes aplicações	51

LISTA DE FIGURAS

Lista de Tabelas

4.1	Resumo das principais funcionalidades das ferramentas Maya e 3ds Max [Mac]	. 36
-----	--	------

LISTA DE TABELAS

Abreviaturas e Símbolos

2D	Two Dimensions
3D	Three Dimensions
ADTG	Annotated Dynamic texture Graph
API	Application Programming Interface
BPM	Beats Per Minute
CVE	Collaborative Virtual Environment
ECA	Embodied Conversational Agent
FP	Feature Point
HMM	Hidden Markov Model
IBT	INESC-Porto Beat Tracker
IK	Inverse Kinematics
LDA	Linear Discriminant Analysis
LDS	Liner Dynamic System
MFCC	Mel-frequency Cepstrum
MIR	Music Information Retrieval
RMPC	Remote Music Control Protocol
RMPCtss	Remote Music Control Protocol Time Synchronization Server
TCP	Transmission Control Protocol
TGA	Topological Gesture Analysis
UDP	User Datagram Protocol
VW	Virtual World

Capítulo 1

Introdução

Esta primeira parte do relatório descreve o contexto desta dissertação e a motivação para esta dissertação na secção 1.1. Na secção 1.2 é sintetizado o problema de que é alvo o tema desta dissertação. De igual forma, são descritos os objetivos a atingir no final do desenvolvimento do trabalho proposto e os resultados obtidos, nas secções 1.3 e 1.4, respetivamente. Na secção 1.5 desta parte são mencionados os principais trabalhos realizados relacionados com o tema tratado. Já na secção 1.6, é descrita a metodologia adotada para o desenvolvimento do sistema. Finalmente, na secção 1.7, é feita uma pequena descrição da estrutura da dissertação.

1.1 Enquadramento e Motivação

Embora nos últimos anos a área de televisão tenha evoluído bastante, existindo já, inclusive, televisões que permitem visualizar conteúdos em 3D ou que possibilitam o acesso a páginas *Web*, estão ainda a ser exploradas formas de melhor interatividade com o utilizador. É facto que, cada vez mais, a televisão perde terreno para a Internet. Isto é, praticamente tudo o que é possível ver numa televisão pode ser também visto no computador. Por outro lado, o contrário não acontece, o que leva a indústria televisiva a apostar em novos conteúdos que cativem o utilizador.

Surge, por isso, a necessidade de preencher a falha que distancia uma pessoa da televisão e a leva a procurar informação ou entretenimento num computador. Desta forma, começam a surgir aplicações com as quais o utilizador pode interagir, estando mais diretamente relacionado com o conteúdo do mesmo. Exemplo disto são serviços incluídos em pacotes televisivos pagos, como aluguer de filmes e reprodução de música. Relativamente a este último ponto, destacam-se, no nosso país, duas principais aplicações: a Music Box da Meo e a My Way da Zon. Ambas as plataformas são inspiradas e baseadas noutras já existentes na Internet, como o Grooves shark ¹ ou a LastFM ². Perante a forte concorrência da Internet, uma utilização tão útil como popular, a

¹<http://grooves shark.com/>

²<http://www.last.fm>

semelhança entre os programas televisivos e interativos não valida que essa exata popularidade se verifique nas duas aplicações.

De forma a reestabelecer a vantagem perdida, os meios televisivos podem aproveitar um canal de transmissão que não é explorado, ainda, no que diz respeito à reprodução de música: a imagem. Então, como tirar partido desse mesmo meio de comunicação para voltar a cativar a audiência? Por que não solucionar este problema com a utilização de avatares dançarinos, bailando ao ritmo da música que o utilizador escolhe? E se, para além disso, estes pudessem ser personalizados, como forma de entreter ainda mais eficazmente o utilizador?

Desta forma, esta dissertação surge num contexto em que existe a referida possibilidade de maior interação em diversas áreas. Sendo a música um fator onnipresente nos dias de hoje, em qualquer aparelho que possua possibilidade de reprodução de som e visualização de imagem, uma aplicação que suportasse estes dois canais de comunicação só ficaria a ganhar. As oportunidades de integração de uma aplicação do género são amplas: tanto em dispositivos móveis e reprodutores de música do computador como em jogos, onde uma personagem poderia reagir automaticamente a sons do ambiente. Ainda, com a chegada da Internet na televisão, a possibilidade de partilha entre os utilizadores é vasta, uma vez que tornaria mais interessante o simples ato de ouvir música no televisor. A melhor forma de associar imagem a música será através da sua principal forma de expressão: a dança. A dança tem um papel importante, visto que, como referido por [ONG10],

"Os movimentos da dança formam uma classe complexa de movimentos humanos que oferecem infinitas formas de expressividade, modos de comunicação não-verbal, diversos vocabulários culturais e um uso rico de interações multimodais com música e outras modalidades."

É então realizado o desenvolvimento de avatares dançarinos, que se movimentam ao ritmo da música e podem, inclusivamente, ser personalizados pelo próprio utilizador, como forma de tornar a sua experiência com a televisão mais interessante e cativante. Para além disso, a dança do avatar varia consoante o estilo musical do trecho escolhido, com o objetivo de tornar a dança mais dinâmica e realista.

1.2 Descrição do Problema

Como descrito na secção 1.1, o propósito desta dissertação é o desenvolvimento de uma aplicação em que avatares escolhidos e personalizados pelo utilizador dancem ao ritmo de uma música. É um tema complexo que abrange problemas provenientes de diversas áreas como a interpretação de ritmo e a transformação em movimentos do avatar segundo um determinado passo de dança.

As aplicações semelhantes desenvolvidas até ao momento possuem avatares bastante básicos (*sticky figures* e segmentos unidos por pontos) e, por isso, não se deparam com certos problemas inerentes à modelação 3D, como é exemplo o espaço que existe ao adicionar roupa a um avatar - esta não se ajusta ao seu corpo - ou como tratar as deformações que acontecem quando o tamanho

de um avatar é alterado. Para além disso, do conhecimento da autora, não existe nenhuma aplicação que integre a possibilidade de personalização de um avatar que, posteriormente, se mova ao ritmo da música, em conformidade com as características escolhidas, ou seja, contemplando um estilo de dança adequado à fisiologia do avatar.

1.3 Objetivos

Esta dissertação tem como principal objetivo o desenvolvimento de uma aplicação, orientada a um contexto de multimédia na televisão, interativa com o utilizador. Foi criada uma aplicação que realiza a animação de um avatar de forma síncrona com a reprodução de um trecho musical. Desta forma, a dissertação tem os seguintes objetivos:

- Modelação 3D de um modelo humanoide;
- Adaptação e integração de algoritmos de interpretação musical já existentes em robots para um avatar;
- Desenvolvimento de uma aplicação em que o utilizador pode escolher um avatar e personalizá-lo;
- Classificação do género musical do trecho selecionado;
- Dança do avatar de forma síncrona com a música, conforme o estilo musical do trecho;
- Análise dos resultados obtidos;
- Elaboração de um protótipo funcional final.

1.4 Resultados Esperados

No âmbito desta dissertação pretendeu-se que fosse desenvolvida uma aplicação de uso fácil e intuitivo, que permitisse ao utilizador, no início, escolher o seu avatar, dando-lhe, caso o desejasse, hipótese de personalizar algumas das suas características físicas e vestuário. De seguida, o utilizador deveria escolher uma música a seu gosto e o avatar escolhido deveria dançar de forma síncrona com essa mesma música, de acordo com o género musical do trecho selecionado.

1.5 Trabalho Relacionado

O estado da arte pode dividir-se em dois temas principais: avatares e música. No que diz respeito aos avatares e à sua criação, é importante ter em conta alguns trabalhos da área da medicina, quanto aos graus de liberdade de cada articulação, entre outros, como o desenvolvido por [TWR⁺05]. Aquando da modelação de avatares, surge o problema de falhas entre o corpo e a

roupa aplicada, cuja solução é apresentada por [BPW93], através de um algoritmo de preenchimento de falhas. Destacam-se algumas aplicações interessantes, entre as quais um apresentador virtual [NZB00] e uma *framework* para geração visual de comportamentos num avatar [KB01].

Já na área da música, foi importante o estudo efetuado sobre diversos trabalhos relacionados com o reconhecimento de batidas e processamento de som. Os sistemas desenvolvidos por [GM96] e [YNT⁺07] seguem uma abordagem de recurso a sistemas multiagente, em que os agentes cooperam de forma a conseguirem calcular a posição da batida seguinte e avaliar as suas hipóteses. Ainda relacionado com o campo da música, podem destacar-se trabalhos relativos à sincronização de ritmo com os movimentos de robots e avatares, dando a sensação de dança. Em [GEKO09], um robot efetua um movimento a cada batida que recebe, resultante de um algoritmo de reconhecimento de batidas. Trabalho semelhante a este é o realizado por [OCFT⁺08], desta vez recorrendo a um avatar.

1.6 Metodologia de Desenvolvimento

A metodologia utilizada para a abordagem do problema contempla diversas fases. Primeiramente, foi efetuada uma pesquisa do estado da arte das áreas a serem tratadas na dissertação. Esta fase ocorreu durante o primeiro semestre e estendeu-se, embora com menor intensidade, no decorrer do desenvolvimento do sistema. Foi selecionada a plataforma a adotar e entrou-se na fase de desenvolvimento propriamente dita. As funcionalidades foram implementadas de forma incremental e iterativa, bem como o teste das mesmas, em ambiente controlado. O resultado final consiste num protótipo funcional, ao qual foi efetuada uma série de testes finais e elaborada a presente documentação referente ao trabalho realizado.

1.7 Estrutura do Documento

O presente documento encontra-se dividido em cinco partes principais: Introdução, Estado da Arte, Metodologia, Implementação do Sistema e Conclusões. O capítulo introdutório fornece uma visão geral da dissertação proposta. Estão incluídos tópicos como a motivação e contexto em que surgiu o tema da dissertação, bem como os objetivos a atingir no final da mesma, de forma geral. No Estado da Arte, é descrito o estudo bibliográfico efetuado em cada uma das áreas que este tema abrange. Diversos artigos e pesquisas relevantes estão incluídos nesta secção. De seguida, no terceiro capítulo deste relatório, é explicada a abordagem que deverá ser adotada por qualquer pessoa que tencione desenvolver um sistema semelhante. Assim sendo, este capítulo refere-se à Metodologia, e explica o procedimento teórico para a criação de avatares personalizáveis para aplicações musicais. No quarto capítulo - Implementação do Sistema - são referidos os procedimentos aplicados na prática, consoante o definido na Metodologia e resultados adquiridos. Por fim, no quinto e último capítulo, é feito um resumo do contexto e resultados finais da dissertação e são apresentadas algumas conclusões finais, bem como perspetivas de trabalho a ser realizado no futuro.

Capítulo 2

Estado da Arte

Serve o presente capítulo para fornecer uma visão geral sobre o estudo realizado até ao momento nas áreas de desenvolvimento de avatares, reconhecimento de ritmo e modelação 3D.

Na Secção 2.1, respeitante ao desenvolvimento de avatares, é mencionada a sua origem e estudos sobre o comportamento dos utilizadores ao personalizar o seu avatar. Ainda são referidas aplicações que recorrem a avatares para a sua interface e conceitos na criação de um modelo humano 3D. A Secção 2.2 trata de esclarecer conceitos no que diz respeito a batidas e notas musicais e apresenta abordagens para resolver o problema do reconhecimento de tempo de início de uma nota. Já na Secção 2.3, é abordado o problema da interpretação musical com o objetivo de animar um avatar ou robot de forma síncrona com um trecho musical.

2.1 Avatares

Na presente Secção é fornecida uma visão geral dos objetos em estudo nesta dissertação: os avatares. É explicada a sua origem e significado, bem como a sua influência na comunicação e expressividade individual no mundo atual. Do mesmo modo, são abordados neste capítulo os métodos necessários e técnicas existentes para a criação de um modelo humano 3D.

2.1.1 Designação e Origem

O termo *Avatar* descrevia, originalmente, a encarnação terrestre do deus hindu Vishnu, significando "descida de uma divindade à Terra". É também utilizado por outras religiões para se referirem à encarnação. Mais tarde, este termo ganhou uma nova vertente quando foi utilizado para representar seres humanos virtuais no videojogo *Habitat*, de 1986. Neste jogo, o utilizador era representado por um avatar, tendo uma perspetiva de terceira pessoa sobre si mesmo. Contudo, só mais tarde, no ano de 1992 é que foi popularizada em maior escala esta denominação, através da novela de ficção científica de Neal Stephenson, *Snow Crash* [Ste00]. Nesta história, o termo *avatar* era utilizado para se referir à simulação virtual do ser humano num outro mundo virtual

denominado Metaverse. No estudo sobre o livro de Neal Stephenson efetuado por [AB98], este analisa as características dos avatares descritas na história e faz uma reflexão quanto ao seu estudo no mundo atual. Os aspetos analisados foram:

- Aparência: são tão diversos como as pessoas que representam. Existem três tipos, nomeadamente, avatares realistas, avatares a preto e branco e avatares de várias gradações;
- Vestuário e acessórios: as roupas dos avatares traduzem aquilo que o humano está a usar na realidade;
- Etnicidade: o herói da história é parte asiático e parte afro-americano. Estão presentes diversas etnias no livro;
- Locomoção: os avatares não caminham simplesmente. São também capazes de correr, passear, entre outros;
- Ação Corporal: são descritas várias ações como lutar com espadas, sentar-se, gesticular, retirar objetos do bolso, etc.;
- Formas de Comunicação: os avatares, assim como os seres humanos, comunicam através de expressões, linguagem verbal e corporal;
- Personalidade: as personalidades são tão diversas como as pessoas que representam. Apenas as personalidades dos *daemons* (atores autónomos) são limitadas.

Nos dias de hoje, o termo avatar é ainda utilizado para descrever objetos semelhantes aos introduzidos por *Snow Crash*: um modelo humano 3D (ou, em algumas situações, apenas uma imagem ou gráfico) que mostra uma representação virtual do seu utilizador.

2.1.2 Criação

A modelação 3D é o processo de desenvolvimento de uma representação matemática de qualquer superfície tridimensional de um objeto, utilizando software especializado. Como é explicado em [Net09], os métodos para a criação de uma figura humana tridimensional são os seguintes:

1. Modelação: construção de um objeto tridimensional através de um conjunto de pontos ou vértices ligados por polígonos
 - Construção por Geometria Sólida: utilização de poliedros básicos e aplicação de operações sobre os mesmos (interseção, subtração, etc.);
 - Nurbs: descrição matemática de superfícies, através de pontos de controlo;
 - Modelagem Poligonal: superfícies representadas através de uma malha de polígonos. Cada objeto possui uma ou mais malhas (*meshes*), em que cada uma representa uma parte desse mesmo objeto;

2. Animação: aplicar movimento a um objeto 3D

- Keyframing: cada *frame* contém uma posição chave do modelo. Cada *frame* pode ser alterado e é então efetuada a interpolação de movimento entre cada uma, dando a sensação de ação;
- Rotoscopia: a animação desejada é extraída de um vídeo real;
- Motioncapture: técnica relacionada com o cinema, que capta movimentos reais através do uso de aparelhos eletrônicos e os converte em animação;

3. Texturização: aplicação de uma textura sobre um polígono pertencente ao objeto 3D

- Color Mapping: uma imagem é replicada sobre a superfície do objeto;
- Bump Mapping: modifica a direção dos vetores normais dos vértices do objeto;
- Normal Mapping: igual ao bump mapping, mas a textura utilizada não é cinza, mas sim RGB;
- Displacement Mapping: utiliza a textura para modificar a estrutura da geometria e, por consequência, para dar uma aparência diferenciada sem nenhum artifício;
- Lightmapping: estrutura de dados que contém o brilho de uma superfície.

A geração de uma personagem 3D pode ser feita recorrendo a:

- Geração por *presets*: geração por modelos pré-fabricados (*presets*). O utilizador pode escolher as características da sua personagem como olho, nariz, boca, roupa, etc.;
- Extração automática por fotografia: são extraídos pontos-chave de fotografias e adaptados a uma cabeça de um modelo humanoide 3D;
- Manipulação de *bones*: existem no interior de um modelo 3D estruturas não-renderizáveis: *bones*. São responsáveis pelas articulações dos modelos, tendo influência sobre os vértices das mesmas;
- Interpolação por parametrização: utilização de um modelo base e de modelos adicionais pré-fabricados que irão sofrer operações de interpolação pelo utilizador com o modelo base.

2.1.2.1 Articulações e Graus de Liberdade

Uma vez que os avatares são criados à imagem e semelhança do ser humano, são igualmente constituídos por membros que se ligam entre si. Possuem também articulações, que resultam da junção de dois segmentos e que, com animação aplicada, podem conferir movimento ao avatar. Segundo [TWR⁺05], podem distinguir-se três tipos de articulações no corpo humano, sendo estas: articulações fibrosas (unidas por tecido fibroso), cartilaginosas (os ossos são unidos por cartilagem hialina e fibrosa) ou sinoviais. Estas últimas são as mais importantes no que diz respeito à locomoção humana e, por isso, as que serão alvo de estudo. Possuem um líquido chamado sinovial

que funciona como lubrificante, sendo revestidas por uma membrana sinovial ou cápsula. Estas articulações são categorizadas conforme a forma da sua superfície. Existem, por isso, sete subtipos derivados destas articulações, que foram estudados tanto por [TWR⁺05], como por [Mac08], que chegaram aos seguintes resultados:

1. Articulações monoaxiais (1 Grau de Liberdade)

- Articulação Dobradiça ou Gínglimo: acontece quando uma superfície côncava de um osso encaixa na superfície côncava de outro osso. Apenas permite movimentos de flexão/extensão, e.g. articulações do cotovelo, joelho e tornozelo;
- Articulação Pivô: permite a realização de rotação. Uma superfície arredondada de um osso é articulada com uma depressão pouco profunda de outro osso, e.g. articulação radiocubital e atlas (no cimo da espinha).

2. Articulações biaxiais (2 Graus de Liberdade)

- Articulação Condilar: permite movimento em duas direções, sendo possível realizar movimentos do tipo flexão/extensão e adução/abdução. Exemplo deste tipo de articulações é a articulação rádio-cárpica (punho);
- Articulação Elipsoidal: a parte final de um osso arredondada encaixa com a cavidade elíptica de outro. Efetua movimentos lado-a-lado e frente-trás nos eixos principais da elipse, como por exemplo, articulações do pulso.

3. Articulações triaxiais (3 Graus de Liberdade)

- Articulação Sela: o seu nome deriva do facto de as superfícies oponentes dos ossos que se encaixam se encontrarem modeladas como uma sela - são côncavas e convexas, articulando-se reciprocamente. Realizam movimentos lado-a-lado, frente-trás, bem como rotações, e.g. articulações do metacarpo do polegar;
- Articulação Esferoide: uma superfície arredondada encaixa numa depressão profunda igualmente arredondada de outro osso. Permite movimentos do tipo flexão/extensão, adução/abdução e rotação, acontecendo, por exemplo, na bacia e articulações de ombros;
- Articulação Plana: os ossos deste tipo de articulação possuem superfícies quase planas, sendo que o movimento ocorre em quase todos estes planos (lado-a-lado e frente-trás são os movimentos mais comuns), e.g. entre a omoplata e a clavícula.

É ainda referido que cada articulação do corpo humano possui um alcance de movimento. Este limite é determinado por fatores como o tipo de articulação, tamanho do músculo na articulação, entre outros. Em termos de ângulos, este alcance é medido em graus para cada grau de liberdade, ou seja, para cada plano em que o movimento é possível numa articulação. Quando uma articulação possui mais do que um grau de liberdade, o alcance de movimento na articulação para cada

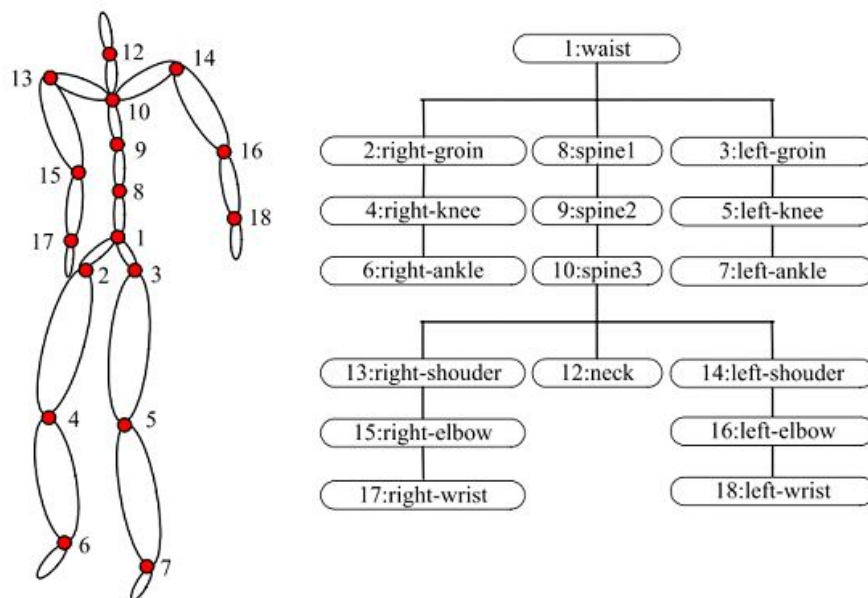


Figura 2.1: Modelo articulado e correspondente ligação entre as suas articulações [GAO07]

grau de liberdade pode variar pois um grau de liberdade pode influenciar outros. As articulações são dependentes. É necessário estabelecer uma hierarquia de dependências ao construir um modelo humano 3D, como é caso o modelo utilizado por [GAO07] e como se pode ver na Figura 2.1, que mostra a correspondência entre as diferentes articulações.

Existem, porém, estudos biomédicos referidos por [SM05], feitos com uma estrutura cinemática com catorze graus de liberdade, que indicam que os ângulos de elevação sagital (corte anatómico que divide o corpo humano em metades esquerda e direita) são mais reutilizáveis do que os ângulos das articulações em diferentes situações de um modelo andante.

2.1.2.2 Problemas de Ajuste Corporal

Para resolver o problema da modificação dos membros do corpo e da postura, [KM95] introduz uma nova representação de formas baseada na composição paramétrica de primitivas. O objetivo é automaticamente segmentar os contornos do corpo sem assumir um modelo de corpo humano à partida. São combinadas diversas informações 2D, de forma a estimar a forma 3D das partes do corpo de um sujeito.

Ao serem aplicados alguns acessórios a um avatar podem existir alguns problemas. Aquele que talvez seja mais importante e frequente é o que acontece quando se aplica roupa no modelo virtual. Como explicado em [BPW93], o que acontece quando se aplica roupa a um avatar (mesmo em malhas poliédricas), é que a roupa não é deformável como na vida real e, por isso, aparecem brechas entre os segmentos. Isto ocorre uma vez que, à medida que a geometria é feita no segmento, é herdada a transformação geométrica sem qualquer tipo de compensação para a interação de material, pele ou roupa na articulação. De modo a resolver este problema, foi desenvolvido um

algoritmo de preenchimento de brechas, que compensa estes espaços vazios quando a figura está a ser animada. O que este algoritmo faz é ligar os limites de dois segmentos adjacentes, gerando superfícies *spline*, usando a informação da tangente dos dois segmentos nos limites.

2.1.3 Importância e Aplicações

Quando a comunicação via internet se começou a popularizar, nomeadamente através de aplicações do tipo *instant messaging*, pouco havia que os utilizadores pudessem personalizar. Era-lhes possível, no início, alterar o seu nome ou email conforme a ideia que queriam transmitir. Mesmo no mundo dos videojogos, o utilizador possuía reduzida possibilidade de individualização da sua personagem. Esta situação veio, mais tarde, a revelar-se como um aspeto importante a ter em conta na forma como o utilizador interagia com os outros através do computador.

Como referido num conhecido cartoon publicado em 1993 no jornal The New York Times pelo artista Peter Steiner, "*On the Internet, nobody knows you're a dog*"¹, ou seja, num meio de interação em que o utilizador não é diretamente visto, este pode, de certa forma, escolher quem quer ser. Assim sendo, percebeu-se que é importante para os utilizadores fazerem-se representar por algo mais do que palavras, mesmo que essa representação não correspondesse à realidade. Abriu-se, assim, um caminho para as pessoas criarem uma extensão de si mesmas ou, então, uma outra representação daquilo que são. Mais recentemente, derivado do crescimento rápido de inúmeras redes sociais e de minijogos que estas incluem, gerou-se um negócio de milhões em personalização de avatares e respetivos adereços. Antes dessa explosão, já existiam dois tipos de mundos virtuais (Virtual Worlds - VW) populados por avatares: mundos focados em combate, como o *World of Warcraft* ou *League of Legends*; e mundos sociais, em que a interação é a chave e em que é possível, não só personalizar o avatar, mas também o ambiente onde este se insere. Neste último caso, as plataformas mais conhecidas serão o *Second Life* e *Sims online*. Cada um destes meios, assim como outros equivalentes, possui milhões de utilizadores, sendo que muitos destes investem dinheiro real para equipar o seu avatar com uma característica física ou acessório mais exclusivo. Estima-se que todos os anos sejam gastos cerca de 100 milhões de dólares em vendas *online* de acessórios e habilidades para avatares.

Com um impacto tão grande que os avatares parecem causar nos mundos virtuais, é importante perceber como estes afetam os utilizadores reais que os criam. Em algumas conclusões tiradas por [Hem06], refere-se que a vivência na pele de um avatar - ver pelos seus olhos, interagir sob a sua forma com outros avatares - pode ser uma experiência bastante intensa. E, de facto, como refere [KFC07], os avatares criados podem oferecer um meio promissor para perceber o que os seus criadores pensam sobre a sua representação na vida virtual. Desta forma, se um avatar serve para representar o utilizador que o criou, seria de esperar que este se assemelhasse ao seu criador. Contudo, um estudo realizado por [DWYW09], chegou a algumas conclusões interessantes:

- Idade: as pessoas mais novas criam, por norma, avatares com idade semelhante à sua; enquanto pessoas de maior idade criam avatares mais novos;

¹<http://www.unc.edu/courses/jomc050/idog.jpg>

- Gênero: o chamado *gender swapping* acontece maioritariamente nos homens, i.e., os seus avatares são do sexo feminino. As mulheres têm maior tendência para criar avatares idealizados do que os homens. Os homens têm tendência para criar avatares que se destaquem. Como referenciado em [HW10], este fenómeno é o caso mais drástico derivado da expressão de liberdade que a criação de um avatar dá aos utilizadores de se criarem a si mesmos de forma diferente da realidade;
- Peso: pessoas com excesso de peso criam muitas vezes avatares idealizados;
- Principal e secundários: os utilizadores questionados têm, em média, 8 avatares cada. Contudo, a maioria tem um avatar principal, ao qual estão mais ligados, e que utilizam mais frequentemente;
- Característica física: a característica física mais importante para os utilizadores é o cabelo e a sua cor. Isto acontece porque o cabelo é uma parte bastante manipulável do corpo e afeta, normalmente, toda a aparência da pessoa. Ainda há a ter em conta que é a parte mais visível nos meios virtuais;
- Ligação afetiva: os utilizadores que criaram um avatar idealizado têm uma maior ligação emocional com o mesmo.

A tendência é, então, a de criar um avatar que represente a pessoa ideal e não a pessoa em si. Este facto, contudo, não é entendido pelos criadores, uma vez que grande parte afirma que o seu avatar é parecido consigo quando, na verdade, isso não acontece.

Foi, inclusive, estabelecido um fenómeno que se refere às alterações que as representações virtuais podem causar aos seus criadores: efeito Proteu. Como é explicado em [YB07], o termo deriva do deus Proteus, que era conhecido por poder assumir diferentes representações de si mesmo. É explicado como o efeito que ocorre quando o comportamento de um indivíduo entra em conformidade com o da sua personagem virtual. Contudo, esta mudança apenas acontece quando a pessoa está a agir através do avatar.

Este facto pode ser igualmente usado como meio de estudar o que acontece quando um utilizador possui um avatar contra os seus padrões. No artigo [YB06], apresentaram-se os dados de uma experiência em ambiente virtual imersivo, em que foi demonstrado que a tomada de perspetiva de avatares que diferem do utilizador reduz alguns estereótipos sociais negativos. Porém, segundo, [HW10], a discriminação permanece nos mundos virtuais. É também curioso, que nos mundos virtuais, as pessoas se comportem de acordo com o papel que desempenham na realidade, formando assim distribuições sociais semelhantes às existentes no mundo real.

Mas nem só em jogos e meios de conversação *online* estão presentes os avatares. Diversas empresas e companhias qualificadas começam a tirar partido da utilização de avatares em meios virtuais imersivos, de forma a realizar, por exemplo, reuniões de trabalho *online* ou então, numa outra vertente, os avatares podem ser utilizados para a via do ensino. O estudo da eficácia destes meios foi feito por [Bai06]. Este começa por introduzir o conceito de *Ambiente Virtual Colaborativo* (CVE), onde são utilizados avatares como representação digital dos participantes da interação.

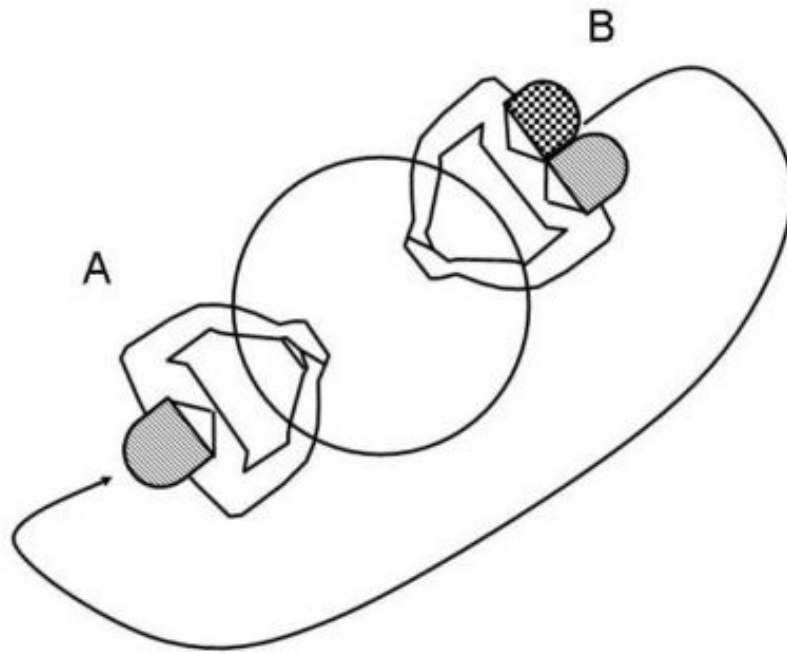


Figura 2.2: Tomada de perspectiva em ambientes colaborativos [Bai06]

Este tipo de ambientes ajuda a resolver o problema de um utilizador não conseguir olhar para a câmara do computador e para a pessoa com quem fala ao mesmo tempo, numa videoconferência, uma vez que o seu avatar tem o olhar dirigido para o recetor, independentemente para onde o emissor esteja a olhar. Outro problema resolvido é o do atraso inevitável da imagem em videoconferências, mesmo em ligações de banda larga. Num CVE, estes atrasos não existem, visto que os modelos 3D são gerados e posteriormente é preciso apenas processar as pequenas modificações de movimento efetuadas. Um CVE é renderizado 60 vezes por segundo para cada participante.

A ideia de proximidade pode também ser adquirida, uma vez que é possível individualizar o contacto social, mesmo numa interação com diversas pessoas. É assim possível para estudantes, por exemplo, estarem numa aula virtual e cada um deles poder ver a explicação do professor no quadro como se estivesse mesmo à sua frente. Podem também fazer pausa ou voltar atrás. Outra vantagem é o facto de o avatar que representa o professor poder ser alterado, isto porque, visto tratar-se de uma representação virtual, não tem de corresponder exatamente à realidade. Desta forma, pode tornar-se a aula virtual mais agradável e apelativa se o avatar que representa o professor estiver a sorrir, mesmo que na verdade isso não aconteça. Ainda é possível ver a interação social de diferentes perspetivas, ou seja, o utilizador pode ver pelos olhos do seu avatar ou pelos olhos da pessoa com quem está a falar, como exemplificado na Figura 2.2:

Aplicação semelhante foi desenvolvida por [NZB00], que criou um apresentador humano virtual, que recebe excertos de textos com comandos embecidos e é animado em tempo real em sincronia com o texto recebido. O agente é capaz de realizar diversos gestos que denotam capacidades de exposição, baseados em gestos convencionados para apresentações públicas.

Um complemento interessante a este último trabalho seria o estudo efetuado por [BA00], que procura desenvolver uma *framework* de criação de avatares com consistência, tanto em termos de ações, como em termos de comunicação em geral. O trabalho incide, então, sobre o estudo de aspetos cognitivos, focados na comunicação não-verbal, de forma a poderem criar avatares cujo comportamento se adeque ao seu "estado de espírito".

Ainda relacionado com a tentativa de tornar os avatares mais humanos, tanto fisicamente como no campo das emoções, destaca-se a *framework* desenvolvida por [KB01] para a geração visual de comportamento numa figura humana 3D. Esses comportamentos estão relacionados com movimento de cabeça e olhos. São associadas atividades motoras, como andar ou levantar e ações cognitivas, como pesquisa visual ou monitorização, com padrões previamente definidos de olhares. Ou seja, é gerada a atenção visual de um agente para um dado conjunto de primitivas.

2.2 Música

[Got07] começa por distinguir dois tipos de experiências perante a audição de música: passiva e ativa. A abordagem passiva é, normalmente, efetuada por pessoas que se limitam a apreciar a música, escutando-a casualmente na televisão ou rádio, não tendo contudo conhecimento técnico específico. Por outro lado, a abordagem ativa é usualmente atribuída a cantores ou compositores que interagem de forma mais direta com a música em si.

"A música é, em grande medida um fenómeno baseado em eventos para o intérprete e ouvinte. Nós acenamos com a cabeça ou batemos com os nossos pés ao ritmo de um excerto; a atenção do artista é focada em cada nota sucessiva." [BDA⁺05]

Desta forma, tanto no âmbito dos artigos referidos nesta pesquisa bibliográfica como do próprio tema da presente dissertação, o objetivo é desenvolver aplicações que proporcionem ao utilizador uma experiência cada vez mais ativa com a música e, portanto, mais imersiva.

2.2.1 Análise Musical - Rastreamento de Batidas

A tarefa de rastreamento de batidas é relativamente simples, se for efetuada do ponto de vista da cognição humana. Mesmo pessoas com pouca experiência musical são capazes de identificar ritmos e padrões de batidas, com uma precisão mais ou menos alta. Contudo, quando se trata da mesma tarefa no caso de um programa de computador, a facilidade de identificação não é tão linear e é ainda hoje objeto de estudo.

O acompanhamento computacional de batidas de áudio, como um processo fundamental na análise rítmica automática, é um pré-requisito para a perceção do ritmo e um fundamento para diversas aplicações MIR (Music Information Retrieval). O seu objetivo é o de analisar eventos temporais na música e transformá-los em abstrações na forma de estrutura métrica. Por sua vez, cada estrutura métrica é caracterizada pelo seu período (distância entre duas batidas consecutivas) e pela sua fase (posição da batida em relação a um ponto de referência).

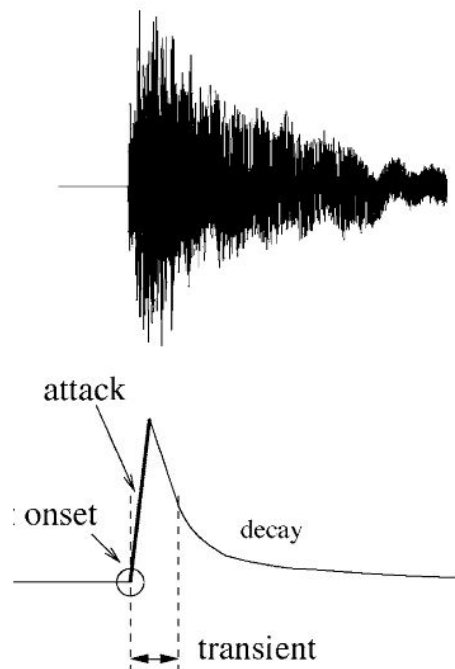


Figura 2.3: Conceitos de ataque, transitório e início numa nota [BDA⁺05]

A detecção/rastreamento de batidas procura, então, padrões na ocorrência das notas musicais, acabando por ser uma análise de alto nível que se apoia no método de detecção dos tempos de início/ataque de cada nota musical. Este tempo pode ser entendido como o tempo de início de cada nota musical.

Este processo é explicado por [BDA⁺05], que esclarece três conceitos: ataques, transitórios e inícios, que estão presentes na Figura 2.3.

O ataque de uma nota consiste no intervalo de tempo em que a variação da amplitude aumenta. Os transitórios consistem em pequenos intervalos durante os quais o sinal evolui de uma maneira inesperada. Por fim, o início é o instante escolhido para marcar a extensão de um transitório. Na maior parte dos casos, o início corresponde ao começo de um transitório.

Quanto a algoritmos de detecção de batida, o método mais comum é o utilizado na Figura 2.4.

Começa-se pelo pré-processamento do sinal de áudio de forma a tornar mais eficazes os passos seguintes. Este tratamento prévio de sinal é opcional e consiste em transformar o sinal original para acentuar ou atenuar algumas características do mesmo. Entre os tratamentos que podem ser aplicados nesta fase destacam-se a separação do sinal em múltiplas bandas de frequência e na separação de transições e estados parados. A fase de redução refere-se ao processo de transformação do sinal de áudio numa função de detecção que manifesta a ocorrência de transitórios/transições do sinal original. Assim que a redução é efetuada, deverão surgir na função de detecção máximos locais (i.e. picos). Assim sendo, é utilizado um algoritmo de *peak picking* para estimar os tempos de início dos eventos dentro da análise de sinal, recorrendo aos picos encontrados.

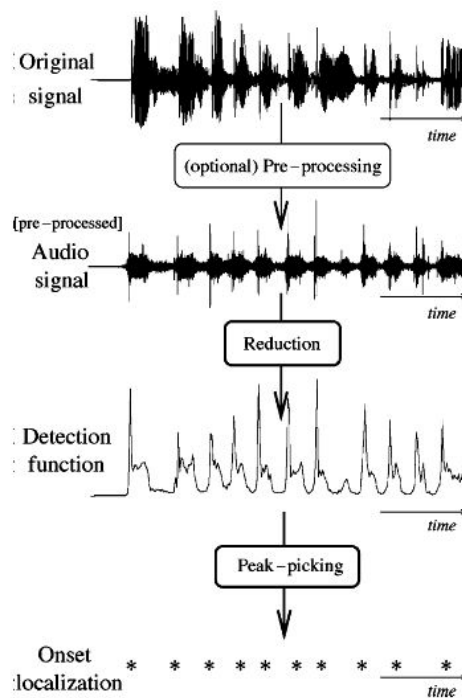


Figura 2.4: Método tradicional de detecção de batidas [BDA⁺05]

O sistema desenvolvido por [GM96] é capaz de reconhecer, em tempo real, a posição temporal de batidas de músicas de diferentes instrumentos. Isto é possível devido a uma arquitetura multiagente que lida com situações ambíguas simultaneamente, o que lhe permite formular várias hipóteses de posições de batidas em paralelo. O cálculo da posição da batida é feito da seguinte forma: cada batida possui duas propriedades - período e fase. No início, cada agente calcula o período. Para isso, o agente recebe uma sequência de vetores de tempos de início de notas e calcula a sua autocorrelação.

Para determinar a fase da batida, o agente calcula a relação cruzada entre a sequência de vetores de tempos de início de notas e a sequência de tempos de batidas cujo intervalo é aquele entre batidas calculado (avalia como coincidem). Estes agentes atuam de forma cooperativa e é prevenida, assim, a hipótese de um agente perder a sua batida. Cada agente é capaz de avaliar a fiabilidade da sua hipótese com base na informação atual, sendo que a hipótese mais fiável é a escolhida como resultado final.

Já em [GEKO09], o algoritmo de reconhecimento de batidas está apresentado na imagem 2.5 e é efetuado da seguinte forma:

- O clip de áudio é dividido em *frames* à medida que é transferido para o sistema. A divisão é feita sob a forma de sub-bandas de frequência que são semelhantes à percepção do ouvido humano;
- A resolução das sub-bandas é diminuída e suavizada;

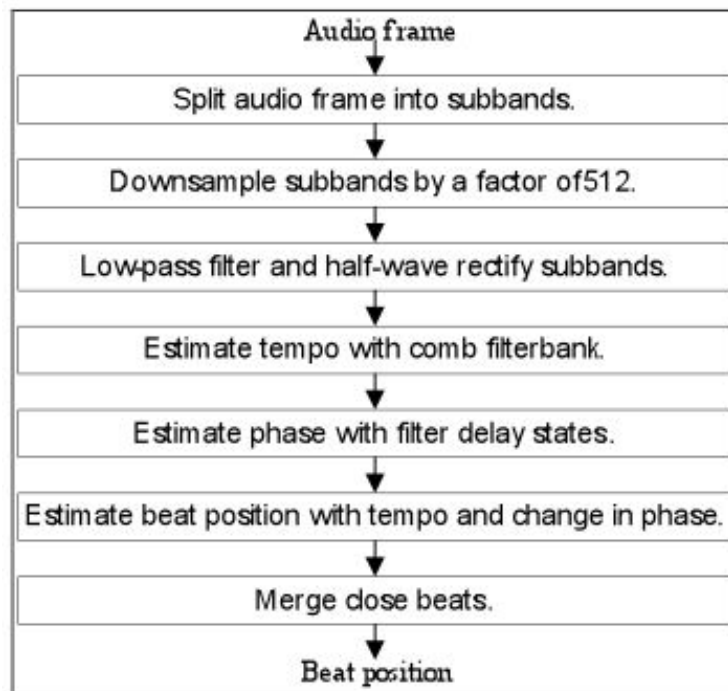


Figura 2.5: Identificador de batidas [GEKO09]

- A música passa por filtros de forma a estimar o seu andamento;
- A fase da batida seguinte é determinada pelo atraso dos estados que passaram no filtro anterior;
- A posição da batida é determinada através do andamento e de mudanças na fase;
- As batidas próximas são juntas.

Os autores de [GNM97] desenvolveram um protocolo chamado RMCP (Remote Music Control Protocol), com o intuito de partilhar informação musical através de redes. Este protocolo é baseado num protocolo UDP/IP e, por isso, difunde informação sem o atraso que acontece quando se efetuam múltiplas transmissões. A necessidade de desenvolver este protocolo surgiu do facto de os protocolos de rede baseados em MIDI terem algumas desvantagens, nomeadamente: têm uma largura de banda limitada, apenas permitem comunicações locais e são ineficazes na partilha de informação entre vários dispositivos.

2.3 Avatares Dançantes

O primeiro dançarino virtual a processar batidas em tempo real foi desenvolvido no ano de 1994. A tentativa de sincronização de um objeto com a música abrange diversas áreas; não só a área da computação gráfica, mas também a área da robótica.

"A dança como uma forma complexa expressiva do movimento é capaz de transmitir significado, emoção e idiossincrasias sociais que abrem canais de comunicação não-verbal, e promove interações intermodais ricas com a música e o ambiente." [ONG10]

A aplicação mais semelhante à do âmbito desta dissertação que fez uso do protocolo RMCP é o programa "Cindy", referido em [Got07]. Este sistema mostra dançarinos virtuais que se movimentam ao ritmo de uma música. As batidas são reconhecidas por um método de rastreamento de batidas e podem também ser escolhidas diferentes sequências de dança. O protocolo RMCP é baseado no típico modelo servidor-cliente, em que diversos servidores recebem pedidos de vários clientes. Outra vantagem deste protocolo é que permite a calendarização da informação, usando *timestamps* para o processamento de pacotes em tempo real. Cada pacote deste protocolo RMCP possui um corpo e um cabeçalho com o tipo de mensagem, *timestamp*, identificador do servidor-alvo, identificador do servidor e tamanho da mensagem. O método de reconhecimento de batida utilizado reconhece a estrutura hierárquica das batidas, tendo conhecimento dos tempos de início, mudança de acorde e padrões de baterias.

Entre outros trabalhos desenvolvidos, destacam-se alguns exemplos como o efetuado por [ARCN09], que consiste num protótipo desenvolvido para um ambiente colaborativo 3D. Neste sistema, as emoções são introduzidas sob a forma de caracteres em emails ou *chats* e mapeadas para um conjunto de gestos previamente definidos que são aplicados a avatares em tempo real. Sendo que um avatar representa um participante num mundo virtual, é uma característica positiva que este expresse também o seu estado emocional. Este fator é igualmente importante, tal como no mundo real, para interações sociais. Esta plataforma foi criada com o auxílio do motor *jMonkey* para a renderização dos gráficos. Um módulo de tratamento de emoções do avatar mapeia os ícones inseridos e desencadeia a animação e pose no avatar.

A área de adaptação de dança para objetos não-humanos é especialmente popular em robots. Também nestes é possível sincronizar os movimentos das articulações com um ritmo de uma música. Têm ainda a vantagem de possuírem articulações mais flexíveis, isto é, com um maior número de graus de liberdade do que um avatar, que deverá permanecer fiel aos movimentos humanos.

É exemplo o projeto desenvolvido por [YNT⁺07] que tem o objetivo de associar uma função que conceda inteligência ao robot para prever a batida seguinte com um movimento físico: passo. É utilizada uma função de reconhecimento de batidas e um método de controlo de feedback para reduzir a diferença temporal entre a batida e o passo. Os comandos de controlo são enviados via *sockets* TCP. O método de reconhecimento de batida implementado devolve o tempo da próxima batida e a batida atual em tempo real. São usados doze agentes para calcular o intervalo entre batidas e prever o tempo da seguinte. Cada agente avalia a sua própria hipótese e é devolvida aquela que tem maior grau de certeza. Este método é semelhante ao explicado na secção anterior, por [GM96]. O robot possui um microfone e capta os sons, que vai então analisar e dar um passo consoante os *inputs* que recebe.

Ainda na área da robótica, foi desenvolvido por [GEKO09] um robot que dança de forma síncrona com uma música. O robot considerado possui 16 graus de liberdade. O processo de

reconhecimento de batida efetuado foi explicado anteriormente, na Figura 2.5. Cada movimento é efetuado logo após receber informação da batida, estando esse movimento armazenado numa base de dados de movimentos criada para o efeito. Cada movimento consiste na deslocação de um membro de um sítio para outro. Os braços podem efetuar movimentos contínuos, mas as pernas têm de regressar à posição inicial, no fim de cada movimento, de forma a não destabilizar o robot. De forma a tornar os movimentos mais parecidos com os dos humanos, as trajetórias foram determinadas com funções cicloides. Estas funções tratam de cada articulação em termos de tempo. O parâmetro C controla quão linear o deslocamento é, sendo que quão mais baixo é o valor de C, mais linear é o deslocamento (2.1).

$$\theta_i = \frac{2\pi t - C \sin(2\pi t)}{2\pi} \quad (2.1)$$

Outro projeto que foi capaz de atingir movimentos de dança num robot é descrito em [NNIY02] e consiste num sistema de captura de dança real e consequente adaptação para robots, usando cinemáticas inversas (IK) - para o ângulo das articulações - e equilíbrio dinâmico. O método de cinemática inversa é explicado em [DAT08] e consiste em determinar os parâmetros de uma articulação de forma a atingir a posição pretendida, isto é, determina os valores das coordenadas das articulações se for conhecida a localização do extremo do membro. De forma a detetar as primitivas de movimento, tomou-se atenção a velocidade local dos *frames* e a mãos e pés do modelo, pois representam os pontos de início e fim do segmento do movimento. O algoritmo de análise de movimento, representado na Figura 2.6 é efetuado da seguinte forma:

1. Definição do centro do corpo;
2. Conversão para coordenadas das mãos e pés (*target points*);
3. Segmentação preliminar – calcula-se as velocidades dos *target points* e deteta-se o mínimo local. Aplica-se também um filtro gaussiano;
4. Avalia-se a correlação entre a trajetória dos segmentos;
5. Os segmentos com trajetória semelhante são juntos.

Os movimentos de dança consistem, assim, em movimentos iterativos de primitivas. Se existirem duas primitivas de movimento que se querem concatenar, a sua transição é gerada da seguinte forma (2.7):

1. Estabelecer uma perna de apoio durante a transição, cuja posição será aquela para a qual a perna da primitiva se deslocará;
2. Calcular a posição do pé que não está apoiado, cintura, corpo e pescoço durante as transições. São usadas polinomiais de 2ª ordem para continuar com a posição e velocidade;

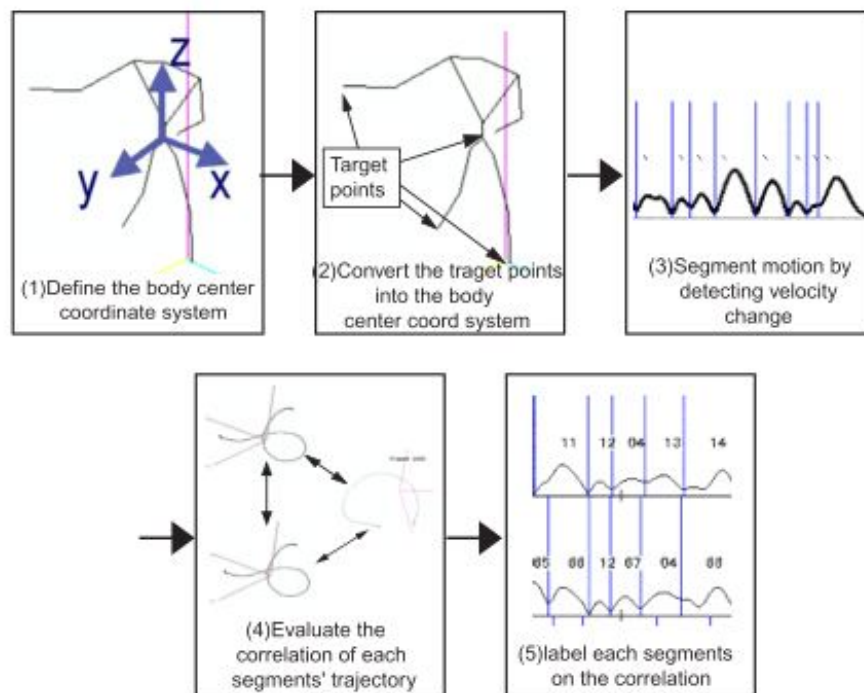


Figura 2.6: Algoritmo de análise de movimento [NNIY02]

3. Interpolação linear para a cintura e pescoço;

4. Interpolação de pernas e braços.

Já no trabalho desenvolvido por [GAO07], efetua-se a aprendizagem do modelo estatístico de um conjunto de sequências de captura de movimento. A abordagem é semelhante à anterior: incorporar na mesma *cluster* de movimentos com trajetória semelhante. O método baseia-se em agrupar os dados de movimento para primitivas que capturam características dinâmicas locais. Textura dinâmica, modelagem da dinâmica de cada *cluster* usando um sistema dinâmico linear (LDS), anotar os LDS que têm significado claro, o cálculo cruzado de entropia entre os quadros de LDS para construir um grafo direcionado chamado grafo textura dinâmico anotado (ADTG), que tem dois níveis estrutura. O nível mais baixo mantém o detalhe do movimento vivo, enquanto o nível mais elevado generaliza o movimento e encapsula as conexões entre LDS.

O artigo escrito por [KWS07] descreve o processamento de uma personagem animada ao ritmo da música, inserido num meio virtual. Para tal, utiliza a seguinte ordem de ações:

1. Base de dados com movimentos de dança, cada melodia tem associada informação;
2. Um analisador de batida detecta os tempos de batidas;
3. Um cliente de movimento seleciona clips de movimentos da base de dados de acordo com a informação da melodia;

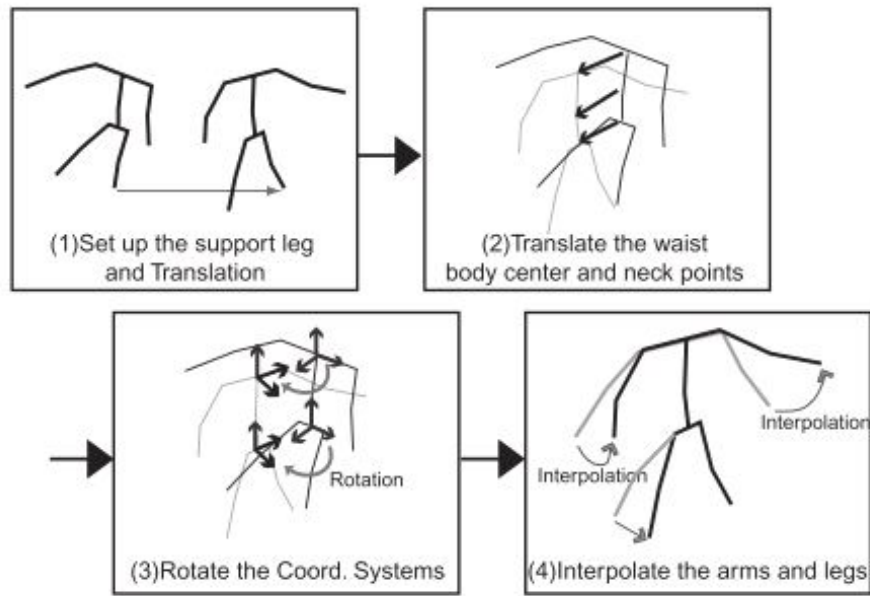


Figura 2.7: Algoritmo de geração de movimento [NNIY02]

4. *Time-wrap*: sincroniza-se o tempo de início do clip com o tempo de batida da música.

Para a construção da base de dados utilizou-se o *Microsoft Access*. Cada movimento era representado por três atributos: número do clip, nome do ficheiro e número de batidas por minuto (bpm). Uma sequência de dança pode ser considerada como um conjunto de movimentos básicos. Esses movimentos básicos são adquiridos cortando as sequências de movimento original em cada batida grande. São também removidas partes transitórias dos movimentos através do processo de Análise do Componente Principal (PCA), que é uma ferramenta matemática usada para reduzir o número de variáveis, mantendo a variabilidade original dos dados. Uma boa síntese do sistema pode ser visualizada na Figura 2.8.

Em suma, utilizando uma arquitetura baseada em TCP/IP, o "Servidor de Música" analisa a música que foi selecionada pelo utilizador e gera mensagem contendo informação extraída dessa mesma música. Essas mensagens são então enviadas para o "Gestor de Movimento" que escolhe clips de movimento que correspondem à informação tirada da música. De seguida, o utilizador escolhe de entre os clips de movimentos apresentados; escolhe essa que é transmitida para o "Reprodutor de Movimento". De forma a mostrar o movimento de forma sincronizada com a música, o "Reprodutor de Movimento" começa por comparar os andamentos da música (recebidos na mensagem enviada pelo "Servidor de Música") e os clips de movimento (a partir da base de dados) e processa um novo andamento para o movimento.

No artigo feito por [ODY⁺08] é descrito o processo de conceção e desenvolvimento de uma *framework* de treino e criação de um avatar que se movimenta ao ritmo da música. Uma visão geral do sistema pode ser encontrada na Figura 2.9.

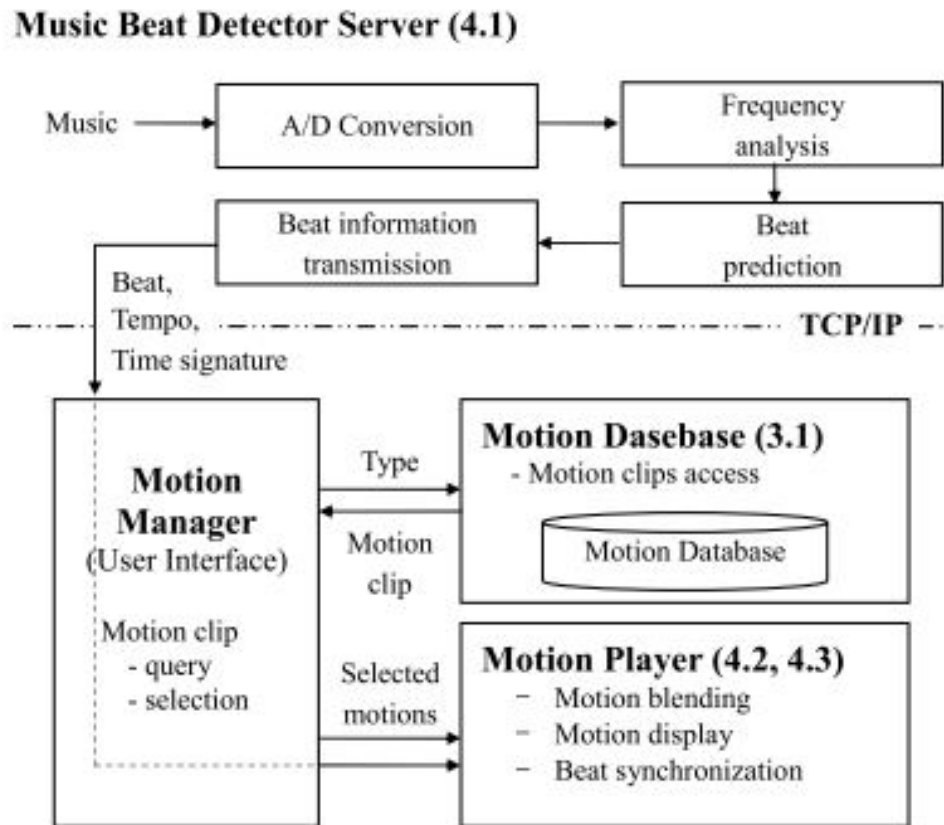


Figura 2.8: Visão geral da arquitetura do sistema [KWS07]

Resumidamente, grava-se um vídeo de uma pessoa a dançar, do vídeo extraem-se as posições 3D das articulações e os seus respetivos ângulos. Do áudio extraem-se as batidas e respetiva localização. Classifica-se o género de áudio a partir de MFCC e, a partir daqui, treinam-se HMM's (síntese dos dados) e reconhecem-se padrões de áudio (coreografia). Isto traduz-se na animação de uma *sticky figure*, através da posição das articulações e/ou num modelo humano 3D, resultado do processamento dos ângulos das articulações. Nesta *framework*, o avatar está concebido para dançar dois géneros musicais: salsa e *belly*.

Já a abordagem efetuada por [BBET97] utiliza um modelo humano 3D que contém entre 40 a 120 graus de liberdade. Estes modelos são baseados numa estrutura em árvore de articulações. Existe um módulo “Agente”, que trata das ações a serem efetuadas pelo avatar. Possui ações ativas e inativas em listas dinâmicas que contêm a sucessão de ações. Na ação há dois aspetos a ter em conta: o conjunto de articulações que a ação controla e a prioridade das ações.

No que diz respeito à análise do movimento do corpo, realizam-se diferentes análises conforme se tratam dos ângulos das junções ou das posições 3D das junções. No primeiro caso, são empregues três HMM's (torso, membros inferiores, membros superiores). Já no segundo caso, são empregues dois HMM's (membros inferiores, membros superiores). De referir que os membros inferiores dizem respeito ao movimento das pernas e os membros superiores ao movimento dos

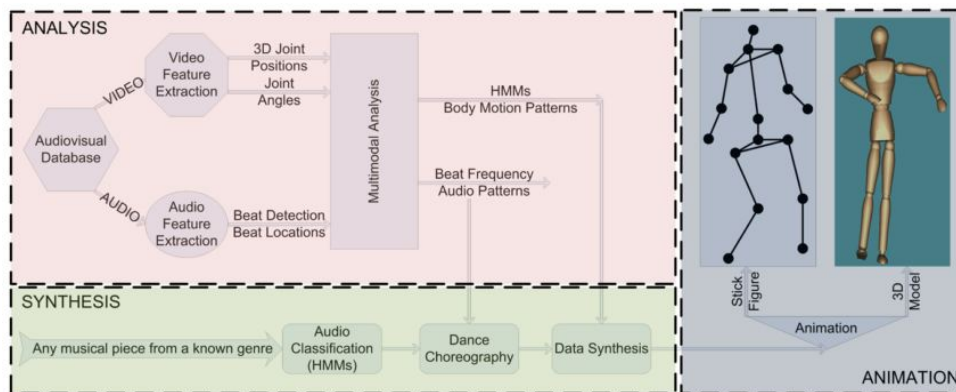


Figura 2.9: Visão geral do sistema [ODY⁺08]

braços. Já para a análise do áudio, procede-se a 3 tarefas básicas: detecção de início, estimação de periodicidade e estimação de localização de batida. Na fase de síntese, realizam-se os seguintes passos:

1. Classificação de género musical;
2. Extração de batida e andamento;
3. Para cada figura agregam-se os diferentes HMM's num único HMM;
4. Parâmetros da posição do corpo são gerados;
5. Suavização do movimento graças a um filtro Gaussiano *low-pass*, para remover a modificação abrupta na transição de um estado para o seguinte.

De sublinhar que cada figura está associada a um género musical. Para a animação desta mesma figura, foi utilizada a ferramenta *OpenGL* para a *sticky figure*, capaz de animar um dado conjunto de pontos em coordenadas 3D, e desenvolvida uma ferramenta chamada *Xbody* para o modelo humano, que utiliza um modelo 3D e mostra os segmentos da animação ordenados de forma hierárquica. Foi utilizado, para modelação do modelo humano 3D, a documentação da ferramenta *open source Xface*. Esta ferramenta, como explicado em [BNZP07], é utilizada para a criação de agentes de conversação incorporada (ECA). A *toolkit* da *Xface* contém quatro partes de software: biblioteca *Xface*, para programadores; editor *XfaceEd*, que gera malhas MPEG4 a partir de modelos 3D estáticos; *Xface Player*, que mostra a ferramenta em ação e *XfaceClient*, que é usado como um editor de script e controlo através de rede com um jogador. A *Xface* aceita ficheiros MPEG4 FAP e scripts SMIL-Agent e anima com movimentos de cabeça e músculos da cara um modelo 3D humano. O utilizador pode escolher os FP's que pretende deformar.

No caso de [ONG10], é proposto um novo método de geração de dança síncrona com a música baseado no modelo topológico de um estilo de dança, previamente capturado por um sistema de captura de movimento. Este modelo foi baseado na Análise de Gestos Topológica (TGA) que,

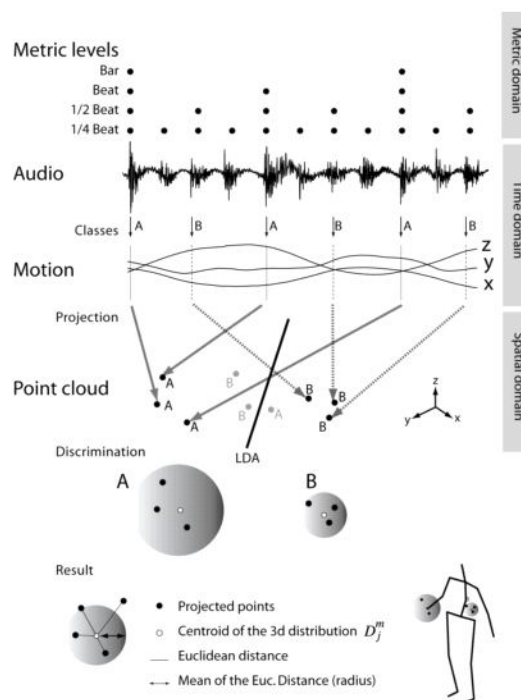


Figura 2.10: Aplicação da análise de gestos topológica [ONG10]

como é explicado em [NL10], consiste na projeção de deixas musicais trajetórias de gestos, que geram nuvens de pontos num espaço tridimensional. Este método está dividido em duas partes. Na primeira parte, as deixas musicais são projetadas no espaço dos gestos de dança. A repetição dos movimentos no espaço resulta nas referidas nuvens de pontos das deixas musicais efetuadas. Na segunda parte, de acordo com proximidade, relação qualitativa, entre outros, as nuvens de pontos são estudadas em relação ao corpo do dançarino.

Neste trabalho, este método foi aplicado como se pode ver na Figura 2.10. Em primeiro lugar, a estrutura métrica da música é sincronizada com as gravações de movimento e anotada. De seguida, essas deixas são projetadas para vetores de movimento. Os pontos métricos são projetados como diferentes classes e, finalmente, as nuvens de pontos são discriminadas, usando análise LDA (Linear Discriminant Analysis). A metodologia usada, representada na Figura 2.11, inclui quatro fases:

1. Aquisição de dados, através da captura de movimento;
2. Representação do estilo de dança, utilizando o método TGA;
3. Síntese de sequências de dança;
4. Visualização das sequências de dança;
5. Avaliação das sequências de dança sintetizadas.

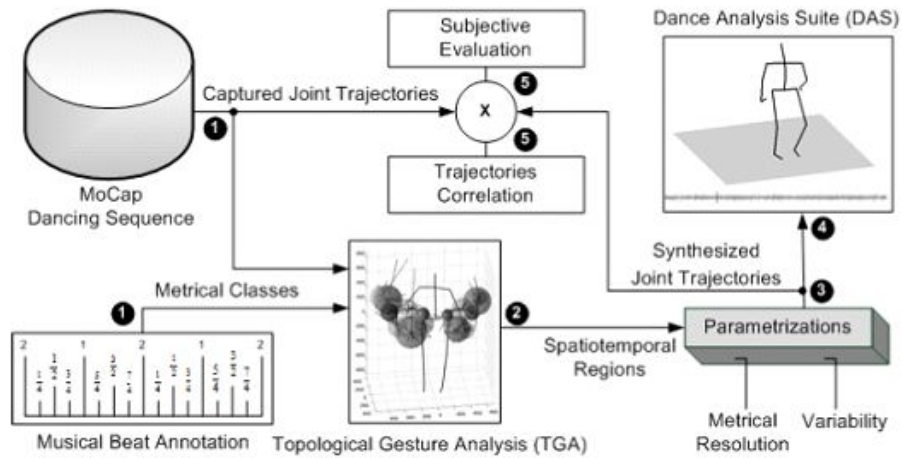


Figura 2.11: *Workflow* da metodologia usada na representação e síntese de dança [NL10]

Ainda, de mencionar uma abordagem efetuada por [RZS10], baseada na síntese de movimento através da otimização baseada em grafos. Foi usada otimização restritiva de forma a encontrar os segmentos de movimento curtos entre *frames* numa base de dados de movimento. Estes movimentos funcionam como transições no grafo. Uma vez que a otimização baseada em grafo está completa, a síntese de movimento em si consiste na otimização discreta para o movimento que satisfaz as restrições.

Capítulo 3

Metodologia

O presente capítulo diz respeito à metodologia utilizada na concepção do projeto presente nesta dissertação. Serve, por isso, para fornecer uma visão global e teórica sobre os métodos adotados e o processo seguido.

Começa por fazer uma visão geral do tema - Secção 3.1 - , recorrendo a conceitos provenientes do capítulo anterior. Na Secção 3.2 é explicado o procedimento de criação de avatares, e também como pode ser realizada a sua animação, de forma a constituir uma dança. Já a terceira parte, respeitante à Secção 3.3, refere-se ao funcionamento do algoritmo de deteção de batidas utilizado de forma a obter os instantes de tempo de cada batida de um trecho musical. Já na Secção 3.4 é explicado o algoritmo desenvolvido para a sincronização da dança do avatar com a música escolhida.

Por fim, na Secção 3.5, são apresentadas conclusões relativamente às decisões de metodologia e abordagem de implementação tomadas para a concretização do trabalho da presente dissertação.

3.1 Visão Geral

Como foi visto no capítulo anterior, este é um tema que envolve áreas tão variadas como o processamento de som, modelação 3D, reconhecimento de batidas e protocolos de controlo de música. Desde o primeiro avatar dançarino, em 1994, bastante evoluiu e diferentes métodos surgiram, mesmo possuindo uma base comum. Diversas soluções foram estudadas, desde a captura de movimentos reais e correspondência com uma simulação do corpo humano, até arquiteturas multi-agente de estimação de posições de batidas e avaliação de hipóteses, na área do processamento de som. Esta é uma área que está em crescimento e cada vez mais se vão aperfeiçoando técnicas de modo a tornar o movimento de um simples avatar semelhante ao movimento complexo de dança real.

Nenhuma das abordagens estudadas se destaca das restantes em todos os campos que este tema abrange, uma vez que cada método possui características próprias indicadas para tipos de

problemas específicos. As abordagens escolhidas mais semelhantes à tomada relacionam-se com a dança em robots. O objetivo desta tese é, então, o de integrar um algoritmo de reconhecimento de batidas, um leitor de música e um sistema de renderização de um avatar 3D animado, numa aplicação consistente e adequada aos utilizadores.

3.2 Avatar

O algoritmo de interpretação de batidas que tem, no final, o objetivo de sincronizar a dança com a música, pode ser aplicado a qualquer modelo 3D que consiga exercitar um movimento rítmico de acordo com som. Modelos humanos são a opção mais natural para este efeito. Os modelos 3D humanos podem ser comparados às personagens utilizadas na técnica de animação *claymation*. Nesta técnica, explicada em [Tay96], cada figura é deformável, contendo uma estrutura base sólida e sendo revestida por plasticina de argila, como mostrado na Figura 3.1¹. Do mesmo modo, o avatar a utilizar deve conter uma estrutura de esqueleto agregada com uma malha exterior, que lhe dá forma. Este avatar pode ser construído recorrendo a ferramentas de modelação ou obtido através de lojas *online* ou modelos grátis. Para soluções que passem por um desenvolvimento manual do avatar, começam a existir soluções para o ajustamento automático de agrupamento de uma malha triangular a uma estrutura de esqueleto, como referido por [RB12].

O propósito do avatar é o de receber e interpretar os dados provenientes do programa de reconhecimento de batidas musicais e o de se mover de acordo com estes dados. De modo a obter estes resultados, é necessário que o modelo 3D se movimente como um esqueleto humano, visto que, apenas assim, é possível obter uma representação fiel dos traços e caracteres de um determinado indivíduo, incluindo os movimentos dos seus membros, relativamente à dança.

O método utilizado para a criação do avatar escolhido passa pela agregação de uma malha de uma personagem a um esqueleto pré-definido - estes dois elementos estão representados na Figura 3.2. Mais especificamente, com a malha e bípede em cena, alinham-se estes dois elementos, de forma a que o esqueleto se encontre no interior da malha. Em suma, o avatar é concebido a partir da noção de um ser humano: um esqueleto envolto por uma camada externa (pele). Porém, apesar de estes estarem corretamente posicionados, o avatar apenas funciona como um todo depois de ter o esqueleto associado a si. Isto consegue-se, na maioria dos casos, aplicando a componente de física ao centro de massa do bípede, que faz com que qualquer movimento feito pelo mesmo seja, igualmente, acompanhado pela malha.

Por fim, apesar de a física ser aplicada ao modelo e de a malha se mover com o esqueleto, algumas partes da malha poderão sofrer um efeito de "arrastamento" ao movimentarem-se. Isto acontece, uma vez que cada parte do bípede possui uma cobertura que nem sempre pode estar a abranger todas as partes da malha, principalmente as mais exteriores. Este problema é resolvido através do ajustamento do tamanho de abrangência das coberturas em causa.

A personalização do avatar pode ser feita alterando diversos parâmetros, sejam as suas texturas e materiais sejam as suas dimensões. Podem também ser modificados aspetos físicos do avatar,

¹<http://www.aardman.com/features/released-features/chicken-run/>



Figura 3.1: Figuras obtidas através da técnica *claymation*

como peso e altura, a partir de escalamentos da personagem. Esta abordagem pode ser interessante na medida em que um avatar poderia dançar/movimentar-se de forma mais rápida ou mais lenta consoante a sua idade, por exemplo.

Contudo, o avatar por si só não é capaz de realizar movimentos que, seguindo uma sequência lógica, constituam uma dança. Assim sendo, é necessário animar o objeto em questão. Para construir uma dança, existe a solução de definir o movimento para cada membro do corpo do avatar a que se quer conferir locomoção. Todavia, esta abordagem torna mais complicado obter uma dança coerente num todo, que se pareça com uma dança real. Outra abordagem possível é a de criar animações pré-definidas, em que cada movimento chave é definido em certos *frames*, sendo o movimento total complementado pela interpolação das posições final e inicial.

Avançando no tempo de animação e definindo novas posições em determinados tempos, cria-se um movimento-chave entre cada instante. Ou seja, por exemplo, tendo o primeiro *frame* definido no instante 1 e o seguinte distinto no instante 10, o *software* para a animação deve criar o restante movimento que prevê para esse mesmo intervalo. Fazendo a interpolação entre o movimento final e inicial, é capaz de completar os *frames* que faltam, criando uma locomoção contínua e suave. Exemplos de movimentos-chave podem ser visto na Figura 3.3.

De referir que, de forma a animação da dança ser consistente, cada animação singular criada deverá começar e acabar exatamente na mesma posição. Uma vez que a animação deve ser repetida em ciclo, é importante que não se note a transição quando a animação volta ao início ou quando é intercalada por outra diferente.



Figura 3.2: Malha e esqueleto que vão compôr o avatar

3.3 Detecção de Batidas

Na presente secção, é explicado o algoritmo utilizado pelo sistema de deteção de batidas de uma música. A escolha do programa a utilizar prendeu-se com a sua contribuição passada na área de robots dançarinos e denomina-se de IBT, desenvolvida por [OGMR10]. Está integrado na plataforma Marsyas e os seus detalhes técnicos são analisados em maior pormenor na Secção 4.1.3. No presente capítulo é explicado o seu algoritmo e funcionamento. É utilizado com o objetivo de definir a duração dos intervalos nos quais a animação deve acontecer. Ou seja, devolvendo informação dos tempos de batidas, é possível que a sincronização mais ou menos exata de uma animação com um trecho musical aconteça. Este sistema percorre, como exemplificado na Figura 3.4, os seguintes procedimentos:

1. Extração de áudio;
2. Indução;
3. Rastreamento da batida.

No primeiro ponto, extração de áudio, é selecionado um fluxo espectral (medida de quão rápido o poder de espectro de um sinal muda) e é-lhe aplicado um filtro de *Butterworth* de forma a suavizar o fluxo escolhido e a reduzir falsas deteções. Na fase de indução, começa-se por calcular as hipóteses de período. A função de periodicidade é processada por um algoritmo de *peak-picking*, que considera o conjunto de dados da função de periodicidade e encontra os picos significantes.

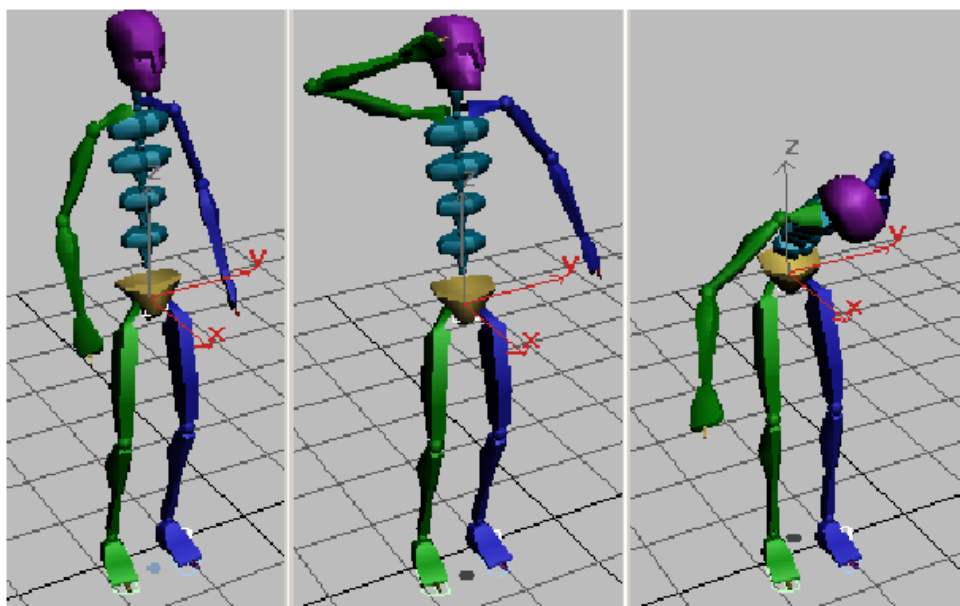


Figura 3.3: Transformações geométricas de certos membros do bípede de forma a criar uma animação

De seguida, tendo o conjunto de hipóteses de período, calculam-se as hipóteses de fases, ou seja, para cada hipótese de período descoberta, um número de hipótese de fases é considerado. Para cada hipótese de período é gerado um *template* de sequência de batidas (*beat train template*), de período constante, do mesmo tamanho da janela de indução. Seleciona-se, então, o *template* que melhor encaixa e devolve a fase correspondente. Neste ponto, estão já calculadas as hipóteses de períodos e de fases. Por fim, processa-se cada conjunto de hipóteses e pontua-se cada uma. Nesta fase de configuração de agentes existem, então, hipótese de períodos, fases e pontuações, que são usadas pelos agentes para iniciarem o seu processo de rastreamento de batidas. Por último, na fase de rastreamento de batidas, supervisionam-se os valores que vão sendo processados em tempo real do fluxo espectral, considerando as variações de andamento e tempos.

De forma a compensar erros, considera-se uma região de tolerância interna, para tratar de períodos pequenos e derivações de fases, que são compensados por uma fração desse erro detetado na região de tolerância interna. Por outro lado, existe também uma região de tolerância externa, que é o contrário da anterior e que para lidar com variações de tempos gera três hipóteses para cada aspeto: fase (tempo), período (andamento) e período e fase. Uma avaliação dos agentes vai sendo realizada em cada momento, de forma a determinar o melhor agente.

Existem duas versões deste sistema: versão casual (*online*) e uma versão não-casual (*offline*). Na versão casual de processamento é retornado o melhor agente atual, enquanto na versão não-casual, apenas o último melhor agente é considerado. Embora a versão *online* seja vantajosa na medida em que envia os tempos de batidas a cada instante, este método não é tão fiável quanto o *offline*. Para além disso, começa a enviar dados apenas passados, aproximadamente, cinco segundos depois do início da música. Isto acontece já que, nesse tempo, o sistema formula hipóteses de

padrões de batidas e analisa os primeiros dados sem os retornar. Depois de ter o padrão definido, não se adapta a mudanças de ritmo. Já a versão *offline* realiza a análise da música por completo (originando o referido tempo de espera até ao final da análise) e, só depois, é que retorna o resultado obtido. Isto resulta de um processo de análise de hipóteses e estimativa mais adequado. Ou seja, as batidas são analisadas até ao final e, só aí, é feito um *backtracking* e tomada uma decisão.

3.4 Algoritmo de Movimento com Música

Uma vez que o avatar está criado e possui animações de dança, o passo seguinte é o de sincronizar esses mesmos movimentos com um trecho musical. Contudo, há que ter em atenção que a animação do avatar deve estar sincronizada com a reprodução da música escolhida e não o contrário. Isto acontece, uma vez que mudar o progresso normal da música comporta resultados piores do que se essa alteração acontecer sobre a música. O ser humano é capaz de identificar alterações num trecho musical, se este for adaptado à medida que a animação se desenrola. Todavia, se a animação se desenrolar e for adaptada à medida que a música toca, o humano não é capaz de acompanhar essas ligeiras alterações de movimento, se a frequência dos *frames* da animação for suficientemente alta.

De modo a obter a dita sincronização do movimento com a música, deve ser aplicado um algoritmo de reconhecimento de batidas à música, quer esta seja escolhida pelo utilizador, quer seja pré-selecionada anteriormente. Este algoritmo deverá retornar instantes de tempo em que ocorrem batidas, detetando padrões de ritmos num trecho musical. Com essa informação, a dança adapta-se ao ritmo da música, sincronizando a animação do avatar com o trecho musical. De forma a isto acontecer, o ciclo de animação do modelo 3D deve coincidir com o intervalo entre batidas. Para isso, foi definido o conceito de "unidades de animação". Como a Figura 3.5 indica, uma unidade de animação consiste numa animação curta, e.g. entre 0.5 a 1 segundo, que é aplicada entre batidas consecutivas. Isto significa que a animação é executada na totalidade entre cada intervalo de batidas.

De forma a criar um movimento global fluido e natural, cada unidade de animação deve começar e acabar na mesma posição, e.g. se a posição inicial é a do avatar de pé com os braços para cima, a posição final deverá ser exatamente igual, de forma a não se notar um desfasamento de movimento quando este repete. Em adição, se várias unidades de animação diferentes partilharem a mesma posição de início e fim, estas podem ser reproduzidas intercaladamente, com o objetivo de tornar a animação mais variada. À medida que a música é reproduzida, sucessivas unidades de animação são executadas em ciclo, até ao final do trecho.

Tendo estes conceitos como base, o funcionamento do algoritmo desenvolvido ocorre da seguinte forma: o algoritmo que vai adaptar a animação à música, utiliza a lista de tempos de batidas supracitada, que consiste num ficheiro ".txt" com estes dados. A partir dos mesmos, o objetivo é o de realizar ciclos de pequenas animações completas, começando numa batida e acabando na seguinte. Isto implica que a animação dure o tempo entre duas batidas (denominado de *tempo musical*), e o seu ciclo se complete neste mesmo espaço de tempo. Para isso, como se pode ver

na Figura 3.6, considera-se o tempo atual da música em cada *frame* (T). Para cada T , é possível aceder à batida que ocorreu imediatamente antes, que é denominada de T_a , e à batida seguinte: T_{a+1} . Assim sendo, para o primeiro instante de batida, a sua batida anterior considera-se 0.

Sabendo que T_a deverá corresponder a 0% do estado da animação e que T_{a+1} deverá corresponder a 100%, pode ver-se a percentagem de tempo decorrido entre T_a e T_{a+1} , e é possível definir o estado em que a animação se deve encontrar a cada instante. Ou seja, para cada *frame*, efetua-se uma interpolação linear, calculando

$$\frac{(T - T_a)}{(T_{a+1} - T_a)} \quad (3.1)$$

Isto significa que em cada *frame* é feita a interpolação dos estados da animação, usando como peso os tempos das batidas e o tempo atual da música que está a decorrer.

Para finalizar, a personalização e variedade podem não se restringir ao avatar. O mesmo pode ser feito com as unidades de animação utilizadas. Assim sendo, é possível usar diferentes unidades de animação para diferentes géneros musicais, artistas ou ritmos. Ainda, não só as animações podem variar, mas também o avatar - pode ser escolhido um género de avatar que coincida com o género do artista, por exemplo.

3.5 Conclusões

No final deste capítulo podem destacar-se alguns aspectos relevantes tomados no que diz respeito à pesquisa e estudo de opções de abordagem, bem como a sua avaliação a nível de aplicabilidade e eficácia no contexto do trabalho desenvolvido.

As decisões tomadas passaram então, primeiramente, pela escolha e definição do método de criação do modelo 3D de forma a que este não fosse apenas uma malha simples, mas que fosse possível movimentar os seus membros. Com esta característica adicionada ao avatar, a animação pode ser criada, sendo construída de forma a que o modelo humano 3D pareça realizar uma dança contínua de um movimento simples e curto. No que diz respeito à componente de personalização do avatar, existem diversas possibilidades que podem tornar a aplicação mais interessante para o utilizador.

Tendo o avatar construído como o objecto sobre o qual se vai proceder a dança de acordo com uma música, sentiu-se a necessidade de recorrer a um *software* que faça o reconhecimento de informação musical. Este *software* é baseado num sistema multiagente em que cada agente envolvido determina uma série de hipóteses de tempos de batidas. O resultado final consiste numa lista de tempos de batidas correspondentes a padrões de som reconhecidos pelo programa. Com esta informação, baseou-se o desenvolvimento de um algoritmo de sincronização de animação do avatar com a música.



Figura 3.4: Arquitetura do Algoritmo de Detecção de Batidas, [OGMR10]

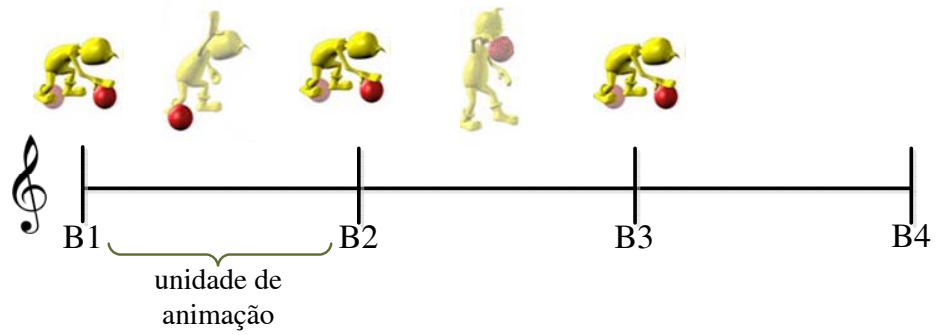


Figura 3.5: Solução para a sincronização da animação com a música

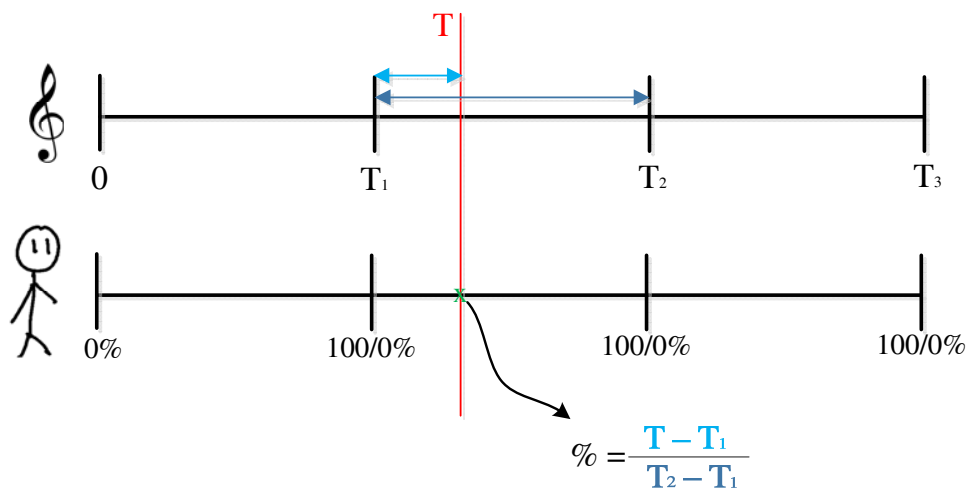


Figura 3.6: Esquema explicativo da sincronização da animação com a música

Metodologia

Capítulo 4

Implementação do Sistema

Neste quarto capítulo da dissertação é explicada em detalhe a implementação efetuada de forma a criar a aplicação final. É, por isso, explicitado o modo como os conceitos teóricos de implementação, explicados no Capítulo 3, são adaptados e integrados na prática, ou seja, os passos efetuados e resultados obtidos. A estrutura do presente capítulo segue um fluxo lógico de implementação, começando por fornecer fazer uma análise das ferramentas utilizadas para o desenvolvimento do trabalho, na Secção 4.1. São explicados os aspetos relevantes de cada programa, bem como a razão da sua escolha, neste contexto.

Na Secção seguinte - 4.2 - é fornecida uma visão geral do funcionamento da aplicação. Já a Secção 4.3 é referente à arquitetura que foi implementada no sistema, demonstrando as diversas partes constituintes da aplicação e a ligação entre si.

De seguida, na Secção 4.4, são explicadas as diversas partes constituintes da aplicação principal e a sua respetiva conceção e funcionamento. Nesta Secção são abordados temas como a criação e animação de um avatar, menus e cenários do programa e algoritmo de sincronização da animação com a música. Por fim, o último capítulo - 4.5 - trata da aplicação secundária ou aplicação de envio de batidas. É explicitado como esta parte do sistema comunica com a aplicação principal e como é efetuada a categorização do género musical de um trecho.

4.1 Ferramentas

De forma a efetuar o desenvolvimento da aplicação proposta pela presente dissertação, é necessário recorrer a um *software* de reconhecimento musical e retorno de informação sobre o trecho de música, um *software* orientado à modelação 3D para realizar a criação de um avatar e um motor de jogo para o desenvolvimento da aplicação em si. Nesta secção, são então referidas e justificadas as ferramentas escolhidas em cada uma destas áreas, em que consiste cada uma delas e como se adaptam para a resolução do problema proposto.

Tabela 4.1: Resumo das principais funcionalidades das ferramentas Maya e 3ds Max [Mac]

	3ds Max	Maya
Foco	Jogos	Filmes
Curva de Aprendizagem	<2 meses	<3 meses
Interface	Limpa, mais intuitiva	Flexível, menos intuitiva
Qualidade de Renderização	Excelent3	Excelente
Animação	Bastante Bom	Excelente
Modelação	Excelente	Bastante Bom
NURBS	Baixo	Bastante Bom

4.1.1 Modelação 3D

De forma a efetuar a modelação 3D de um modelo humano, foi efetuado um estudo do *software* existente, de forma a determinar o mais apropriado para o efeito. Para isso, analisaram-se os seguintes programas: Blender, Maya e 3ds Max.

Em primeiro lugar, considerou-se a hipótese de utilizar o Blender, tanto como *software* de modelação 3D bem como motor de jogo. O Blender, desenvolvido pelo estúdio alemão NeoGeo, consiste num *software open source* utilizado para criar videojogos, aplicações 3D, filmes animados, entre outros. Foi escrito em C, C# e Python e possui diversas funcionalidades como o controlo em tempo real durante simulações da área da física ou durante uma renderização, ferramentas de animação que incluem cinemáticas inversas, armadura, entre outros e integração com diversos motores de renderização externos, através de *plugins*.

Apesar disso, o Blender possui a desvantagem de ter uma interface com o utilizador bastante complicada, sendo, por isso, difícil de aprender. Como é de resto mencionado por [CB05], é pouco evidente a forma de chegar às funcionalidades e mesmo os atalhos de teclado variam consoante o modo em que o utilizador se encontra, isto é, a mesma tecla acede a diferentes funcionalidades dependendo do contexto. Ainda, o modo como interage com a cena atual é pouco óbvio. Por esta razão, e visto que o Blender serviria os propósitos de motor de jogo e de ferramenta de modelação, optou-se por pôr de parte esta opção, mesmo sendo *open source* e possuir inúmeras funcionalidades.

Deste modo, a escolha cingiu-se aos *softwares* Maya e 3ds Max. Ambos pertencem à Autodesk, sendo que o 3ds Max sempre foi um produto da Autodesk e o Maya foi adquirido recentemente à Silicon Graphics. O 3ds Max é mais orientado para a modelação 3D, enquanto o Maya é mais focado na animação e, por isso, é mais utilizado na indústria cinematográfica. Ambos suportam ações de modelação, animação e renderização.

Pode ser vista uma tabela comparativa em 4.1, adaptada de [Mac]. Pode concluir-se que as duas aplicações possuem uma interface bastante poderosa; a animação é de maior qualidade no Maya, enquanto a modelação é mais forte no 3ds Max. A curva de aprendizagem é semelhante para os dois, sendo menos acentuada no 3ds Max. Na modelação por NURBS, o Maya possui clara vantagem, pois permite criar uma forma apenas com uma face. Contudo, a escolha foi determinada pelo foco de cada uma: o Maya está mais centrado na animação e o 3ds Max na modelação e, sendo a modelação o propósito de uma plataforma do género para esta dissertação,

foi escolhido o 3ds Max para modelação de avatares.

4.1.2 Motor de Jogo

Para o desenvolvimento da aplicação, é necessário o uso de um motor de jogo. O programa escolhido para o efeito foi o Unity ¹. O Unity consiste numa ferramenta para a criação de videojogos 3D e de outros conteúdos interativos, criada pela Unity Technologies. No ano de 2010, ganhou um prémio de inovação tecnológica na categoria de *software*, atribuído pelo *Wall Street Journal*.

O programa foi concebido em C# e C++ e trabalha com Javascript, C# e Boo (Python). É capaz de importar diversos tipos de arquivos, tanto em 2D como em 3D, suportando arquivos do tipo ".blend", ".max" e ".fbx". Possui uma interface bastante intuitiva (exemplo na Figura 4.1), com possibilidade de edição visual, inspetores de propriedade detalhados, pré-visualização de jogo e um ambiente de desenvolvimento integrado com estrutura hierárquica. É bastante orientado por uma lógica de *drag & drop*, o que facilita a utilização. De entre vários conceitos-chave para a interação com Unity, destacam-se três explicados de seguida:

- **Objetos de Jogo** - corresponde a cada objeto do jogo. Determinam aparência e comportamento, já que contêm Componentes.
- **Componentes** - determinam a aparência e comportamento do Objeto de Jogo ao qual estão agregados. São partes integrantes desenvolvidas pelo utilizador. Resumidamente, são a classe base de tudo o que é agregado a um Objeto de Jogo.
- **Pré-fabricados** - são um tipo de recurso. Funcionam como Objetos de Jogo reutilizáveis podendo ser inseridos numa cena diversas vezes. Cada Pré-fabricado corresponde a uma instância do Pré-fabricado original. Desta forma, qualquer alteração ao Pré-fabricado original reflete-se nas suas instâncias.

Adicionalmente, tem compatibilidade com navegadores *web*, como Internet Explorer, Firefox, Google Chrome, Safari, Opera e Camino.

Este programa trata de agregar toda a informação recolhida da análise de música, cenários e dança, constituindo a aplicação principal.

4.1.3 Reconhecimento de Informação Musical

A ferramenta utilizada para a extração de informação de uma música será o Marsyas (Music Analysis, Retrieval and Synthesis for Audio Signals)². Foi desenvolvido por George Tzanetakis e tem sido usado um pouco por todo o mundo em alguns projetos tanto académicos como comerciais, sendo a melhor solução para este efeito. É uma *framework open source*, cujo principal objetivo é a experimentação de áudio, a sua síntese e análise do sinal.

¹<http://unity3d.com/>

²<http://marsyas.info>

Implementação do Sistema

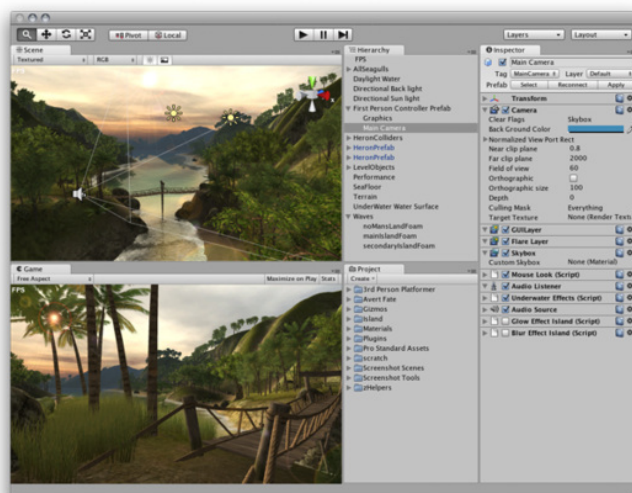


Figura 4.1: Exemplo de interface do *software* Unity

O sistema, utilizado em Marsyas, que trata da análise efetiva de padrões de ritmos denomina-se de IBT. Trata-se de um sistema baseado numa arquitetura multiagente para indução de andamento e detecção de batida, explicado em mais detalhe na Secção 3.3 e desenvolvido por [OGMR10]. Um esquema simples explicativo pode ser encontrado na Figura 4.2. . O IBT começa por receber um ficheiro de áudio e efetua os seguintes procedimentos: análise de andamento (tempo), e de fase. Ao mesmo tempo, inicializa os agentes do sistema.

Assim sendo, em linhas gerais, a detecção de batidas é efetuada por um sistema multiagente. Vão existir agentes de avaliação e um agente "árbitro" (*Agent Referee*). Este último é responsável por, de tempos a tempos, avaliar a votação dos agentes ativos em cada predição de tempo de batida. Vai selecionar o melhor agente em cada instante (*frame*, recorrendo a uma determinada heurística, neste caso, uma função de classificação. Por fim, o sistema é capaz de retornar a estimacção de batidas.

A aplicação recorre a este sistema, uma vez que este devolverá a informação das batidas de uma música para, mais tarde, serem traduzidas em movimentos no avatar.

4.2 Visão Geral da Implementação

Fazendo uma perspetiva geral da aplicação como um todo e das suas partes constituintes, destacam-se quatro entidades: aplicação principal, aplicação secundária, integração com Marsyas e Last.fm.

O fluxo lógico da aplicação desenrola-se, em primeiro lugar, com a aplicação principal. Esta aplicação foi desenvolvida no motor de jogo Unity usando as linguagens C# e Javascript, e consiste numa série de menus e cenários onde é possível efetuar a personalização e dança do avatar. De forma a movimentar o avatar sob a forma de dança, a aplicação comunica com a aplicação secundária, também denominada de *Beat Sender*. Este programa regista a música escolhida pelo

Implementação do Sistema

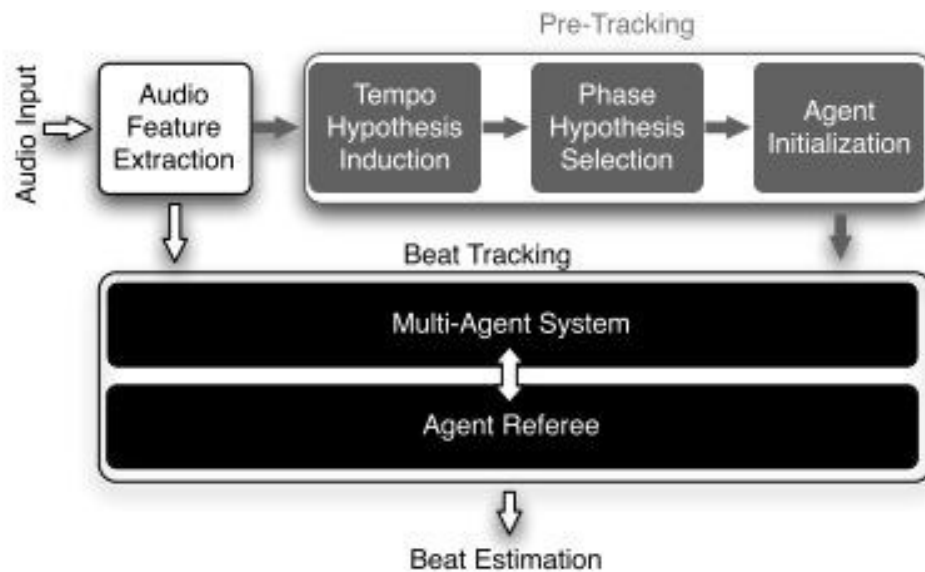


Figura 4.2: Diagrama explicativo do sistema [OGMR10]

utilizador e, opcionalmente, comunica, por sua vez, com uma outra entidade externa: Last.fm. Ao utilizar é possível, através dos serviços deste sítio *web*, obter a classificação do género musical do trecho escolhido.

De seguida, o caminho do ficheiro é enviado como parâmetro do comando "ibt -nc", que corre usando o sistema IBT, explicado na Secção 4.1. Este vai proceder à análise das batidas da música e devolve um ficheiro de texto com os instantes de tempo de cada uma. Após terminar esta análise, o *Beat Sender* envia os dados resultantes à aplicação principal. Com esta informação, a aplicação principal é capaz de, não só escolher o tipo de dança que deve ser executada pelo avatar, mas também de reproduzir a música escolhida, ao mesmo tempo que vai sincronizando a animação de dança com esta.

4.3 Arquitetura

Na Figura 4.3 é apresentada a arquitetura concetual do sistema desenvolvido. Este é dividido em três camadas: interface com o utilizador, lógica de negócio e camada de dados. É apresentada, de seguida, uma breve descrição de cada uma destas camadas e dos seus constituintes:

- Interface 3D: esta camada de topo consiste no espaço em que o utilizador e a máquina interagem. Neste caso, essa interação é feita através de uma aplicação de aspeto simples e intuitivo, constituída, maioritariamente, por elementos 3D.
- Scripts e Controlo de Fluxo: esta camada corresponde a todo o processamento que acontece em segundo plano (por detrás da aplicação visível ao utilizador). Neste caso, e como se

Implementação do Sistema

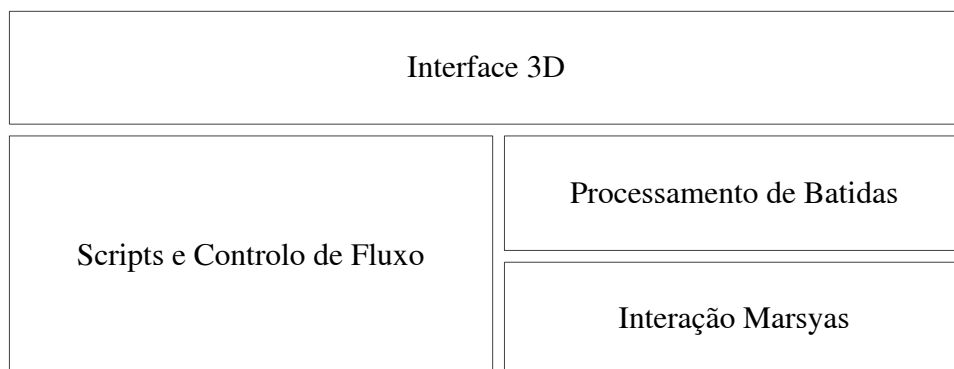


Figura 4.3: Arquitetura concetual do sistema

pode ver na Figura 4.3, esta camada encontra-se do lado esquerdo do esquema e refere-se aos scripts desenvolvidos e aos mecanismos de controlo de fluxo (menus).

- **Processamento de Batidas:** refere-se ao tratamento dos dados dos instantes de batidas que é feito, de forma a controlar o movimento de um modelo 3D.
- **Interação Marsyas:** refere-se à utilização e à comunicação deste programa com a aplicação desenvolvida. São requisitados ao sistema do Marsyas - IBT - os dados relativos à música escolhida pelo utilizador. Este trata, por isso, da análise de padrões de ritmo do trecho musical.

4.4 Aplicação Principal

A aplicação principal foi desenvolvida, recorrendo ao motor de jogo Unity, utilizando as linguagens C# e Javascript. Para a comunicação com a aplicação secundária, de forma a processar a troca de informação, foi utilizada a linguagem C#. Já a linguagem Javascript foi utilizada, como o nome indica, para a elaboração de diversos scripts, não só respeitantes à criação de menus, como à interação de objetos, personalização do avatar e a sua animação.

Esta aplicação é composta por três cenários distintos: menu principal, vestiário e discoteca. No primeiro cenário, o utilizador pode escolher dar início às cenas seguintes, consultar as instruções de navegação da aplicação ou fechar a mesma. Já o segundo cenário permite a escolha de um entre dois modelos 3D disponíveis e a personalização de alguns dos seus elementos constituintes. Finalmente, na terceira e última cena, correspondente à discoteca, desenrolam-se os movimentos de dança protagonizados pelo avatar. Para isso, o utilizador deve escolher o botão "Dance!", que desencadeia o início da aplicação secundária. A partir deste momento, a aplicação em Unity fica a aguardar receber a informação acerca da música que o utilizador escolhe. A música é selecionada através da interface com o utilizador do programa secundário, podendo ser verificado o seu estilo musical. Quando a posição temporal das batidas é concluída, essa informação é transferida à

aplicação principal que, em primeiro lugar, inicia a reprodução da música. Ao mesmo tempo, sincroniza os movimentos do avatar com o trecho musical, de acordo com a informação das batidas recebidas, criando assim um movimento fluido e realista, que acompanha o ritmo da música.

4.4.1 Visão Geral da UI

A interface com o utilizador é composta por diversos componentes nas diversas cenas. Na primeira cena, o menu, a interface é constituída por um menu de opções no centro e por um avatar animado do lado direito. A seguinte cena é bastante semelhante à primeira, já que é composta por um menu no lado esquerdo e um avatar animado no lado direito do cenário. Este segundo menu contém diversas opções de personalização do avatar presente na cena, bem como uma opção para a troca do avatar por outro diferente. Já a cena seguinte não possui nenhum menu à partida, apenas um avatar com animação de espera. Quando o utilizador escolhe a tecla "Dança", abre-se, então, uma aplicação, na qual o utilizador terá de escolher uma música do tipo ".wav". Poderá, se assim pretender, procurar o género da música introduzida ou então introduzi-lo manualmente.

4.4.2 Avatar

De forma a exteriorizar as danças criadas consoante os dados de batidas recebidos, procedeu-se à criação de um avatar. O modelo humano 3D supracitado foi criado através da junção de uma malha de um humano a um esqueleto. Este processo foi efetuado, usando o *software* 3ds Max, já que este está orientado para a modelação 3D e possui uma componente denominada de "Bípede", que fornece a estrutura esquelética para juntar ao modelo.

De modo a atribuir movimento ao avatar, recorreu-se à animação por *frames*, também em 3ds Max. Em determinados instantes de tempo, definiu-se uma posição para os membros do corpo do avatar e foi feita a interpolação linear entre estes vários movimentos, conferindo-lhe uma fluidez na locomoção. A partir do momento em que o avatar se encontra completo e com a animação definida, é necessário que este seja exportado para Unity de forma a ser integrado no cenário da aplicação, no qual irá desempenhar a sua função principal: dançar.

Finalmente, já em Unity, o papel do avatar é o de servir como meio de exteriorização dos dados recebidos da análise das características musicais da canção escolhida pelo utilizador. A animação pré-definida do avatar, é, por isso, moldada de acordo com a informação que este vai recebendo a cada instante.

Todo este processo é explicado em maior detalhe nas secções que se seguem, nomeadamente: a criação do avatar na Secção 4.4.2.1, a sua animação na Secção 4.4.2.2 e, por fim, a exportação do modelo 3D na Secção 4.4.2.3.

4.4.2.1 Criação

Um dos elementos mais relevantes do trabalho desenvolvido é o objeto sobre o qual se aplicam os movimentos que constituem uma dança, ao som de uma música. De forma a criar esse objeto - o avatar - foi considerada a hipótese de o fazer na ferramenta sobre a qual todo o restante trabalho

estava a ser produzido – Unity. Contudo, esta hipótese foi descartada, uma vez que o Unity, como motor de jogo, não está orientado para modelação e, por isso, não possui as funcionalidades necessárias para o efeito. Assim sendo, e como referido na Secção 4.1, utilizou-se o *software* 3ds Max para a criação do avatar.

Entre algumas questões a considerar aquando do planeamento da criação do avatar, destaca-se o facto de existirem diversas formas de o fazer. Ou seja, o avatar tanto poderia ser criado modelando uma figura de raiz e adicionando-lhe, posteriormente, um esqueleto para poder realizar movimentos, como utilizando uma malha pré-existente e associando-lhe um esqueleto. Visto que a primeira opção é consideravelmente mais trabalhosa e o resultado final seria bastante semelhante ao resultado da segunda, optou-se, então, por esta última. Com este método, criaram-se duas personagens: um cientista chamado Dr. X e um engenheiro chamado Wilson, que podem ser vistas na Figura 4.4. Ambas as malhas destas personagens foram adquiridas em tutoriais disponibilizados pelo sítio *web* oficial do produto 3ds Max³. É explicada, de seguida, em detalhe, a técnica seleccionada para o desenvolvimento destes dois avatares.

Primeiramente, junta-se à cena a malha da personagem, que é, de certa forma, como a sua pele. Funciona como o revestimento do avatar, dando-lhe forma. Contudo, esta malha não funciona por si só. De forma a esta poder mover-se de forma semelhante ao ser humano, é preciso atribuir-lhe algo similar ao que o próprio ser humano usa para se movimentar: um esqueleto.

Em 3ds Max, é possível criar um esqueleto, utilizando funcionalidades como "Ossos" ou "Bípede". Estas duas ferramentas são bastantes semelhantes. Contudo, a ferramenta "Ossos" surgiu primeiro que a "Bípede" e a grande diferença reside no facto de a funcionalidade "Ossos" permitir criar membros um a um. Assim sendo, é mais apropriada para os casos em que o avatar não é o de um humano. Desta forma, a funcionalidade usada foi a "Bípede", uma vez que já possui modelos de bípedes integrais pré-definidos, que podem corresponder a um corpo preenchido de homem, mulher, ou um modelo clássico, apenas constituído pelo esqueleto, que não diferencia o género do bípede. É ainda possível adicionar ou retirar ligações do esqueleto, de forma a ajustarem-se ao tamanho da malha. A referida estrutura comporta-se como um esqueleto real, sem ser necessário qualquer tipo de configuração adicional. Um exemplo dessa estrutura desenvolvida composta por malha e esqueleto pode ser vista na Figura 4.5.

4.4.2.2 Animação

De forma a criar um avatar dançarino, foi necessário conferir-lhe movimento, de maneira a originar uma sequência de posições que se assemelhassem a uma dança. Este tipo de tarefa envolve lidar com articulações do esqueleto de forma individual. Deste modo, optou-se por utilizar o *software* 3ds Max, tal como foi feito para a criação do avatar, já que permite a manipulação direta do esqueleto.

Em 3ds Max, é possível realizar dois tipos de animações em bípedes: animação de passos e de forma livre. A primeira, animação de passos, consiste - como o nome indica - em criar uma série

³<http://usa.autodesk.com/adsk/servlet/item?id=18995103&siteID=123112&linkID=9241175>



Figura 4.4: Avatares criados para a aplicação

de posições pré-definidas para os pés do bípede. O utilizador pode, entre outras coisas, alterar o número de espaços e o espaçamento entre os mesmos. Já o segundo tipo de animação, forma livre, não utiliza passos, mas necessita, antes, da definição de cada chave da animação, manualmente. Visto que o propósito desta animação é assemelhar-se a uma dança, foi utilizada a animação do tipo forma livre. De referir que existe, ainda, a possibilidade de misturar estes dois tipos de animação com o modo "mixer", combinando passos e movimentos livres.

No tipo de animação de forma livre, e já com o corpo do avatar preparado, utiliza-se a barra de espaço temporal, com instantes de tempo correspondentes a *frames*. Cada movimento definido num *frame* designa-se por chave. Com o modo "Auto Key" ativo, cada transformação geométrica de membros feita no avatar, passa a ser a nova chave no momento temporal em que o avatar se encontra. O programa 3ds Max é, então, capaz de realizar uma interpolação linear entre cada instante de animação, calculando a locomoção intermédia entre cada movimento-chave definido.

Assim sendo, foram criadas animações, não só de dança, mas também de espera, saudação, entre outros. Cada animação criada deu origem a um avatar diferente, com uma animação fixa, criada em 3ds Max. Isto foi realizado, então, rodando ou mexendo diferentes partes do corpo do avatar em diferentes instantes de tempo (*frames*). As animações foram desenvolvidas, tendo por base vídeos de amostras de danças de diferentes estilos de música e, no caso da animação do vestiário, um vídeo do popular jogo "The Sims"⁴, onde o vestiário é um conceito que lhe está associado. No que diz respeito às animações de dança, foram criadas quatro animações de dois

⁴<http://thesims.com>

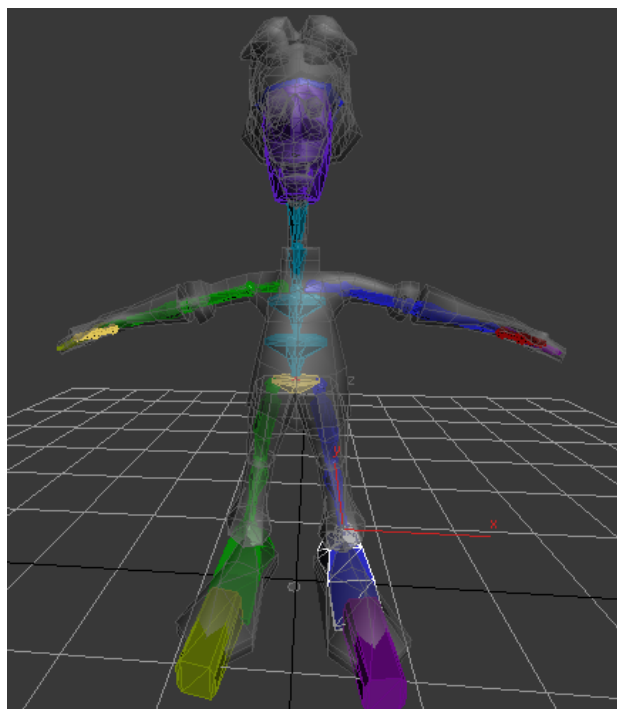


Figura 4.5: Malha e esqueleto constituintes do bípede

estilos de dança. Ou seja, de forma a tornar a dança mais interessante e menos repetitiva, para cada estilo de dança existem duas animações diferentes de pequena duração, com o fito de alternarem entre elas. Os estilos de dança escolhidos para serem reproduzidos pelo avatar foram o *hip-hop* e *pop/rock*.

De notar que, uma vez que os movimentos em Unity devem ser usados em ciclo, teve-se em atenção o facto de a posição inicial da animação e a posição final terem de corresponder. Isto foi possível já que o 3ds Max permite copiar e colar poses de diferentes *frames*, gravando a posição do esqueleto do avatar e guardando-a, de maneira a poder voltar a ser usada.

4.4.2.3 Exportação

Após o modelo ser construído e preparado com animações, está pronto para ser integrado com os restantes elementos da aplicação. Uma vez que o trabalho em 3ds Max está concluído, o avatar tem de ser transferido para o projeto de Unity. Porém, esta transferência não se realiza de forma direta, sendo preciso exportar o objeto em questão num formato comum aos dois programas. O formato apropriado para este efeito é do tipo ".fbx", sendo, geralmente, necessário instalar um exportador de *fbx* ao 3ds Max.

Para modelos simples, criados de raiz no programa, a exportação deve ocorrer sem contrapartidas. Todavia, tratando-se de modelos mais complexos, com texturas, materiais e diversas partes (esqueleto e malha para este caso), poderão ocorrer problemas de incompatibilidade. Caso isso aconteça, o modelo pode não ser exportado corretamente. No desenvolvimento desta fase da

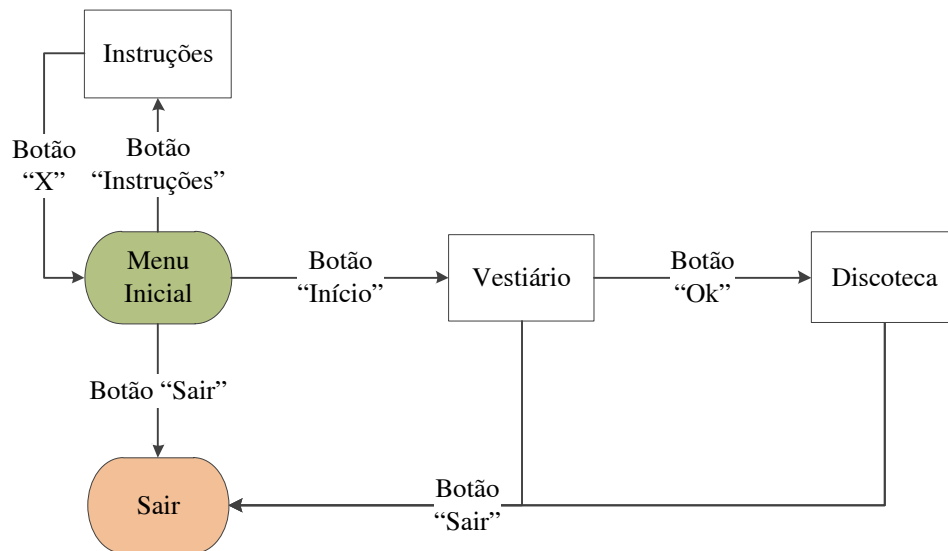


Figura 4.6: Diagrama de fluxo entre os diversos componentes da aplicação

dissertação, o avatar sofreu alterações na malha ao ser exportado para Unity, tornando-se num modelo disforme irreconhecível. Foi, por isso, necessário mudar certas configurações no exportador de *fbx* e no próprio modelo, como garantir que ele se encontrava como "editable poly". Ainda para a animação realizada poder ser reproduzida em Unity, a opção "bake animations" deve ser selecionada no exportador. Aí, escolhe-se o intervalo de tempo pretendido, normalmente a duração da animação.

Já depois de corretamente exportado para Unity, as suas animações, texturas, materiais, entre outros, são automaticamente extraídas e guardadas como componentes do objeto avatar. Assim sendo, o avatar encontra-se pronto para lhe serem aplicados scripts que definam o seu comportamento na lógica da aplicação.

4.4.3 Menus e Cenários

Tendo já o avatar preparado, é necessário integrá-lo num cenário e criar toda a lógica de menus, de forma a intercalar diferentes cenas que confirmam lógica à aplicação e a dividam em partes relevantes. Assim sendo, a aplicação, inicialmente, apresenta um menu composto por um botão de "Início", outro de "Instruções" e um de "Sair". A cena seguinte consiste num vestiário virtual, onde o avatar pode ser trocado por outro e as suas roupas alteradas. Na cena final, o avatar encontra-se num cenário alusivo a uma discoteca, onde a música é escolhida pelo utilizador e o avatar dança ao som da mesma. Para melhor compreensão, pode ver-se um diagrama de fluxo de menus que reflete a explicação, na Figura 4.6.

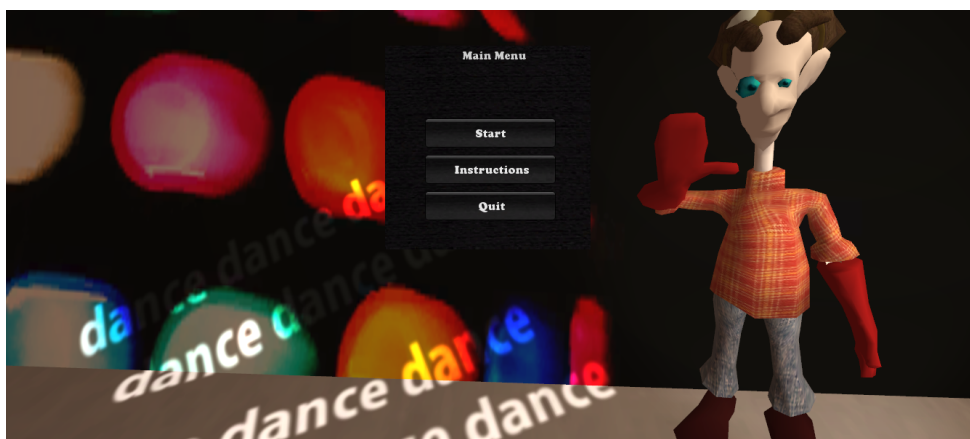


Figura 4.7: Interface inicial da aplicação

4.4.3.1 Menu Inicial

Ao abrir a aplicação, o utilizador depara-se com um cenário que contém um menu e um avatar com uma animação, que simula uma saudação. O menu possui as seguintes opções:

- Start - liga a cena corrente à cena seguinte da aplicação - vestiário - que é explicada na secção seguinte - 4.4.3.2.
- Instructions - explica como utilizar as principais funcionalidades da aplicação e de como nagevar nas cenas.
- Quit - encerra a aplicação.

Esta primeira cena pode ser vista na Figura 4.7.

4.4.3.2 Vestiário

O menu seguinte - vestiário - foi inspirado no conceito, tantas vezes utilizado por diversos jogos e aplicações com personagens personalizáveis, onde as mesmas eram personalizadas como se de um provador se tratasse. A cena envolvente, que pode ser vista na Figura 4.8, foi adaptada da disponibilizada pela página de Recursos do sítio *web* do Unity⁵ e consiste numa sala composta por um espelho, tapete, toalheiro e alguns *posters*. O avatar encontra-se, então, a efetuar uma animação de repouso, antes de passar para a cena seguinte. Do lado esquerdo, é apresentado um pequeno menu com várias hipóteses de partes personalizáveis e também um botão que permite trocar de avatar. Inicialmente, o avatar que aparece na cena é o Dr.X, que pode ser trocado pelo engenheiro Wilson. Contudo, o protótipo apenas apresenta um avatar completamente funcional: Dr.X. Esta funcionalidade tem apenas a intenção de mostrar que é possível utilizar mais do que uma personagem.

⁵<http://unity3d.com/support/resources/example-projects/charactercustomization>



Figura 4.8: Cenário do vestíbulo onde ocorre a personalização do avatar

No que diz respeito à personalização, é possível alterar certas partes do avatar, como o cabelo, camisola e calças. Isto é possível dado que o programa acede à textura atual do avatar e troca-a por outra. Isto acontece através da troca das texturas e dos materiais do avatar, que consistem em componentes criados aquando da sua exportação. Quando o utilizador decidir que está terminada a aparência do avatar, pode passar para a cena seguinte, carregando no botão "Ok".

4.4.3.3 Disco

Depois de personalizar o seu avatar, o utilizador é reencaminhado para a cena seguinte: discoteca. O cenário, como se pode ver na Figura 4.4.3.3, é semelhante ao anterior, mas conta com detalhes diferentes, de forma a se assemelhar, o mais possível, ao conceito de uma pista de dança. Para complementar este ambiente, foi criada uma bola de espelhos, que vai rodando sobre si mesma no cenário e uma série de luzes de diversas cores. Este efeito foi conseguido, utilizando um sistema de partículas, que consiste em dezenas de partículas adicionadas ao cenário 3D. Cada partícula é emitida com uma certa velocidade e tem uma duração específica, até se desvanecer. Neste caso, foram emitidas partículas elipsoides, com uma rotação e pouca duração de vida. Também possuem diferentes cores, que vão aparecendo e desaparecendo à volta de um eixo invisível.

O avatar aparece localizado, sensivelmente, na posição anterior, vestindo as roupas escolhidas na cena anterior, o que significa que há uma passagem de características do modelo 3D de uma cena para a outra. A sua animação é de repouso até que o utilizador escolha a música pretendida, altura em que a animação muda para dança. A animação de dança varia conforme um de dois estilos musicais identificados pelo programa: *pop/rock* ou *hip-hop*. Para cada um destes dois estilos musicais, existem também dois tipos de animações diferentes, para que a dança não se torne monótona. Na secção seguinte, é analisado, com detalhe, como a música é processada e como a animação de dança do avatar se adapta à mesma.



Figura 4.9: Último cenário da aplicação

4.4.4 Som e Movimento com a Música

Ainda tendo como cenário a pista de dança, quando o utilizador escolhe o botão "Dança" é aberta uma aplicação externa. Trata-se de uma interface que permite ao utilizador escolher um ficheiro de música do tipo ".wav". Este ficheiro é, então, processado, de forma transparente ao utilizador, através do serviço IBT. Este *software* devolve um ficheiro de texto com o instante de tempo de cada batida detetada. Da análise feita resulta a identificação de padrões de batidas, como se pode ver num exemplo da Figura 4.10. Nesta imagem, a cinzento está representada a onda musical do trecho escolhido e as linhas laranja referem-se aos instantes de batida reconhecidos pelo sistema.

Depois de algum tempo a processar a música e a gerar o ficheiro de texto com os dados, o avatar está preparado para começar a sua animação. No instante em que o processamento termina, a música que foi escolhida pelo utilizador é posta a tocar e a animação tem de acompanhar a linha temporal que corresponde à duração da música. Como explicado no Capítulo 3, a animação do avatar tem de estar sincronizada com a reprodução da música escolhida e não o contrário. Ao tentar sincronizar o tempo da música com o tempo da animação, o resultado obtido foi o de a música reproduzida estar alterada, com efeito de "solavanco" ao longo da faixa. Contudo, alterando a abordagem para a sincronização da animação com o tempo da música, a frequência alta dos *frames* faz com que a animação pareça sempre fluida aos olhos do utilizador.

Aplicando o algoritmo explicado no capítulo anterior, obtém-se uma dança fluida e natural por parte do avatar. Dependendo do tamanho da música, a análise às suas propriedades de ritmo pode demorar mais tempo. Isto acontece já que o algoritmo do IBT está a ser utilizado nesta aplicação na sua versão *offline*.

Foi escolhida esta abordagem, já que, como foi visto no Capítulo 3, embora a versão *offline* tenha a desvantagem de demorar mais ou menos tempo a proceder à análise da música, é mais vantajoso o seu uso em comparação com a versão *online*. Esta última não se adapta tão bem a mudanças de ritmo e a informação que devolve não é tão fiável.

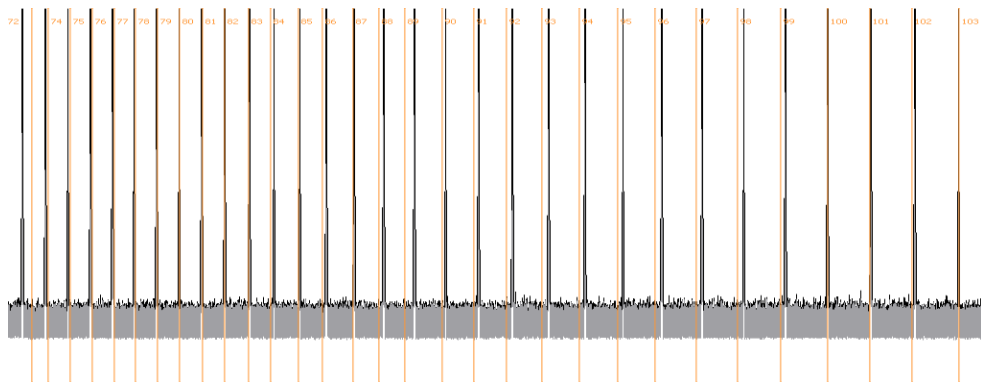


Figura 4.10: Resultado visual da análise musical

Para além disto, sofrendo da ambiguidade métrica do ritmo musical, o IBT poderá intercalar ou extrapolar o nível métrico a que atende ao longo da análise de uma mesma peça musical que resultará em alterações do *tempo* musical (e.g., para o dobro ou metade) que serão confundidas com alterações da "velocidade" rítmica da música, que na verdade não acontecem. Isto acontece, uma vez que uma música, sendo uma peça complexa, não possui apenas um padrão de ritmo (mas sim um padrão hierárquico a vários níveis métricos) o que leva a que, várias vezes, o padrão que é reconhecido por um algoritmo possa não ser o mais óbvio para quem ouve a música.

De notar ainda que alterações bruscas do ritmo musical (em termos de *tempo* ou *timing*) poderão demorar a ser atendidas pelo algoritmo dependendo da proeminência das hipóteses atuais face a hipóteses passadas.

4.5 Aplicação de envio de batidas

A aplicação de envio de batidas, desenvolvida nativamente em C#, apresenta a interface representada na Figura 4.11. É um programa de utilização simples e intuitiva, cujo objetivo é a escolha de um trecho musical, o reconhecimento do seu perfil de batidas e o subsequente envio das mesmas para a aplicação principal. Através da seleção do botão "Choose file", é apresentada ao utilizador uma janela de escolha de ficheiros, restrita a ficheiros que possuam a extensão "wav".

Uma vez escolhido o ficheiro, o utilizador tem a possibilidade de escolher o género da música de entre uma lista de opções pré-definidas. Alternativamente, o utilizador tem a hipótese de recorrer à API do Last.fm para obter o género da música que escolheu a partir do seu título e respetivo nome do artista, selecionando o botão "Lookup Lastfm". Para o efeito, o nome do ficheiro é analisado na tentativa de preencher automaticamente os campos referentes ao artista e ao título; em particular, este processo funciona quando o nome do ficheiro se encontra no formato "Artista - Título.wav". Tanto o género da música como o par artista-título podem ser introduzidos manualmente pelo utilizador.

Finalmente, tendo sido selecionada toda a informação necessária, o utilizador deve escolher a opção "Send to server", que desencadeia o processamento do ficheiro escolhido e o subsequente

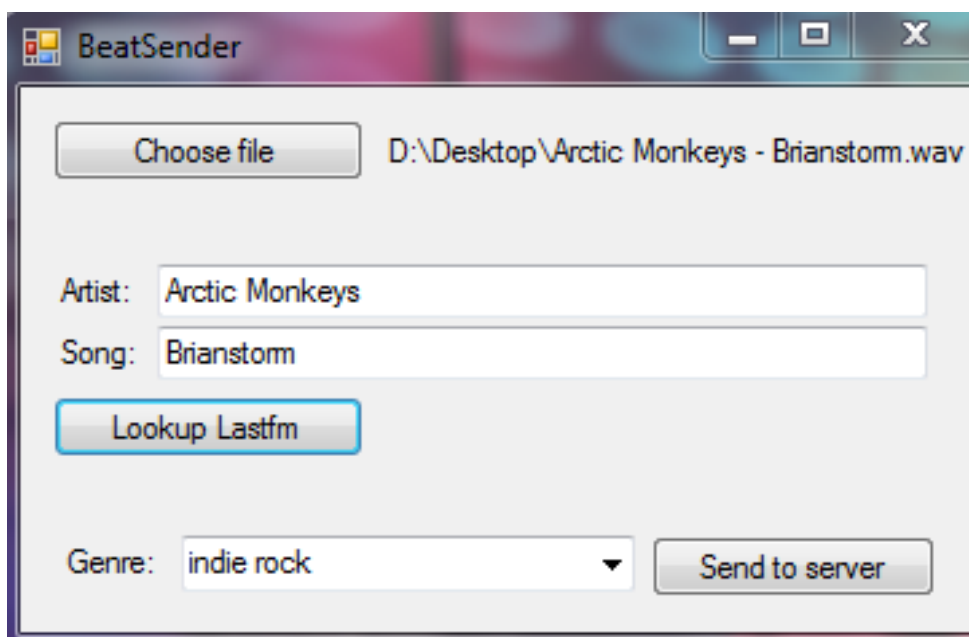


Figura 4.11: Aplicação de envio de batidas

envio dos resultados para a aplicação principal.

4.5.1 Comunicação com a Aplicação Principal

A aplicação principal desenvolvida comunica com a aplicação secundária, que trata da ligação com o sistema IBT, para a análise dos padrões de batidas da música. A aplicação secundária pode ser vista na Figura 4.12. Na interface com o utilizador, este pode escolher o botão "DANCE!", que despoleta a abertura da aplicação denominada *Beat Sender*. Esta ação desencadeia o processo de comunicação entre a aplicação principal - desenvolvida em Unity - e o *Beat Sender*, que serve de ponte de ligação para o *software* de reconhecimento de batidas. Ao surgir, então, a janela da aplicação secundária, o programa principal fica à escuta da receção de informação.

Assim que o *Beat Sender* é iniciado, é possível escolher a opção "Lookup Lastfm". Esta operação tem o objetivo de classificar a música segundo o seu género musical. Para isto, são usados métodos da API do serviço do site da Last.fm (explicado em mais detalhe na Secção 4.5.2). Esta chamada de métodos destes *WebServices* usa o título da música e nome do artista - se estiverem presentes na sua base de dados -, de forma a devolver a etiqueta de maior peso que lhe está associada, que corresponde, geralmente, ao género do trecho musical. De referir que esta ação de categorização da música escolhida não é obrigatória, podendo proceder-se ao reconhecimento de batidas da mesma sem ter este dado.

Posteriormente, quando o utilizador escolhe o botão "Send to Server", a aplicação secundária envia os dados necessários ao sistema do IBT para que este faça a análise da música. Para isto, o programa *Beat Sender* executa o comando "ibt", juntamente com o nome do ficheiro que o utilizador escolheu, como parâmetro da linha de comandos. O IBT é baseado num sistema multiagente

Implementação do Sistema

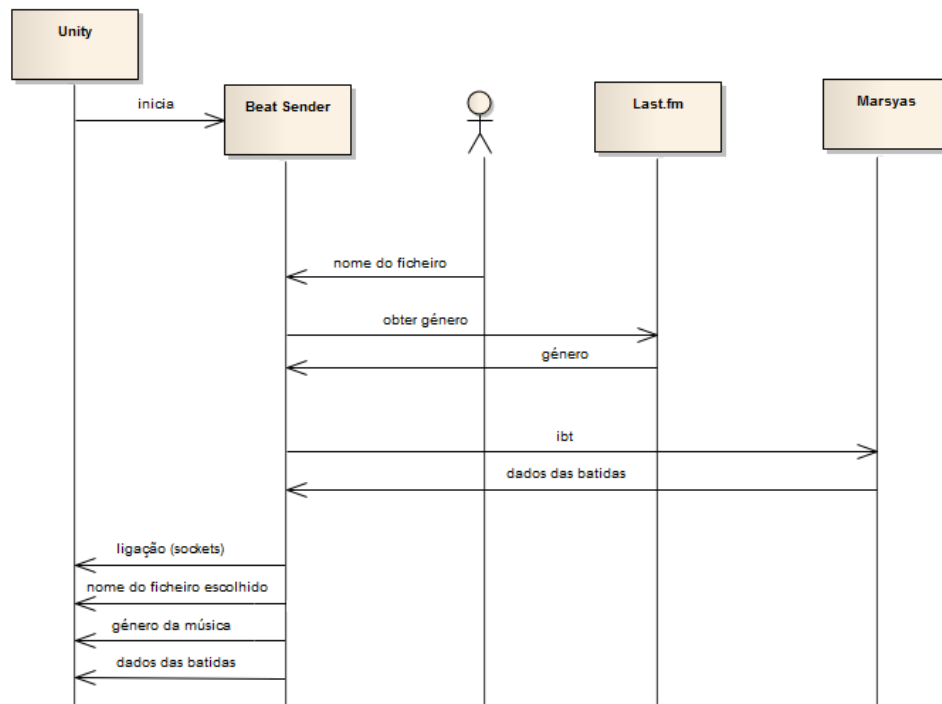


Figura 4.12: Comunicação entre as diferentes aplicações

que considera hipóteses paralelas relativamente ao andamento e batidas de um trecho musical. Os dados pedidos pelo *Beat Sender* são-lhe então retornados, sob a forma de um ficheiro de texto, que contém os instantes de tempo de cada batida reconhecida pelo programa.

Assim finalizado o processo de análise da música, os dados deverão ser enviados para a aplicação principal, de forma a moldar a dança do modelo 3D. O *Beat Sender* vai, para o efeito, ligar-se à aplicação principal através de *sockets*, começando a enviar os dados obtidos. Estes mesmos dados são enviados na seguinte ordem: nome do ficheiro escolhido, género musical e informações das batidas analisadas. O primeiro - nome do ficheiro - é utilizado na aplicação principal de maneira a reproduzir a música escolhida ao mesmo tempo que a dança se inicia. Já o género musical tem o objetivo de selecionar o tipo de dança a ser protagonizado pelo avatar. Por fim, os tempos das batidas musicais determinam a velocidade a que a animação é reproduzida, ou seja, se a música for mais lenta, a animação também o será, e vice-versa.

4.5.2 Integração com Last.fm

Na aplicação desenvolvida, quando o utilizador escolhe uma música, esta é processada segundo os seus padrões de batidas. Assim sendo, esse processamento apenas tem em consideração o andamento da música, não fazendo distinção se se trata de uma música de um género ou outro. As animações ou danças do avatar não teriam, por isso, correspondência com o género musical que foi escolhido, já que os dados fornecidos não continham essa informação. A solução encontrada

para este problema foi a de recorrer a uma entidade externa que conseguisse identificar o género do trecho musical. Para este efeito, foi escolhido o serviço da Last.fm, que consiste numa rádio *online*, que possui uma comunidade extensa de utilizadores. O *website* possui uma base de dados que, segundo informações de [Ric09], contém mais de 12 milhões de músicas individuais, o que é vantajoso, já que esse facto permite que haja pouca margem para músicas não reconhecidas.

Na aplicação desenvolvida, e como explicado na Secção 4.5, a interface com o utilizador é dotada de um campo para o género musical. Contudo, esse campo é, na verdade, preenchido pela etiqueta de maior peso correspondente à música escolhida. As etiquetas das músicas são variadas e podem incidir sobre determinados aspetos, como a classificação musical, tema do trecho, características do artista, entre outros. Como a API disponível pelo serviço da Last.fm não permite a recolha direta do género musical, optou-se por fazê-la a partir destas etiquetas. Regra geral, a etiqueta de maior peso corresponde, então, ao género musical.

O pedido à API foi feito, usando bibliotecas em C# próprias do serviço da Last.fm, de forma a comunicar com o mesmo. Por razões de segurança, é necessário obter uma chave da API antes de poderem ser utilizados os *Web Services* disponíveis. Foi criada uma variável do tipo *Track*, composta pelo nome do artista e título da música. Estes dados são recolhidos dos campos da interface com o utilizador, que foram analisados aquando da escolha do ficheiro, ou introduzidos manualmente. É, então, utilizado o método *GetTopTags*, que tem como parâmetro a variável do tipo *Track* supracitada. Este método da API devolve uma lista das etiquetas mais populares, ordenadas por grandeza. Finalmente, é devolvido o género musical correspondente à música escolhida.

Capítulo 5

Conclusões e Trabalho Futuro

Desde o surgimento da Internet e da exploração do conceito social neste âmbito, foram aparecendo as mais diversas formas de interação entre utilizadores. Uma dessas formas encontradas foi a utilização de pequenos modelos tridimensionais que se assemelhassem ao ser humano e, se possível, às características do próprio utilizador. Foi uma questão de tempo até que esta nova forma de representatividade num meio tão vasto como é o mundo *online*, passasse de um mero modelo estático ou sem funcionalidades complexas para um meio de interação ativa. O conceito de avatar tomou uma dimensão maior do que a esperada inicialmente e é agora utilizado como personagem de jogos, assistentes *online* ou até dançarinos.

O primeiro avatar dançarino surgiu no ano de 1994 com o objetivo de tornar a experiência musical de um utilizador em algo mais do que escuta passiva. Adicionando o fator da imagem a algo tão simples como a audição de música pode tornar esta interação bastante mais interessante. Ainda mais se essa experiência for representada por um objeto semelhante ao utilizador e reagir automaticamente com a música. O tema desta dissertação surgiu com a necessidade de incorporar os dois meios - áudio e imagem - num só.

No decorrer da elaboração desta dissertação foram analisadas diversas áreas, das mais diferentes componentes. O estudo efetuado incidiu sobre três temas principais: processamento de som, modelação 3D e controlo de animações. Em cada uma delas foram analisadas as contribuições já existentes e metodologias adotadas e tomadas decisões sobre a implementação a desenvolver. O resultado final consiste numa integração de elementos variados: música, movimento e personagens 3D. A aplicação construída está preparada para a personalização de alguns aspetos de um avatar e permite escolher, inclusive, o próprio avatar. Ainda ao recorrer a ferramentas que efetuem o reconhecimento de batidas de uma música, é capaz de determinar a velocidade da animação do avatar, criando uma dança fluida e realista, sincronizada com a música escolhida. Possui ainda mais do que um movimento de dança e também danças para géneros musicais específicos, que podem corresponder ao género da própria música seleccionada pelo utilizador. Os resultados são, por isso, motivadores e vão de encontro ao proposto. Uma aplicação deste género possui aplicabilidade em diversas áreas maioritariamente relacionadas com multimédia e jogos.

Conclusões e Trabalho Futuro

Existem, no entanto, diversas funcionalidades que podem ser adicionadas ao trabalho existente, que poderiam contribuir para o seu melhoramento:

- envio de batidas em tempo real, de forma a não haver tempo de espera entre o início do processamento da música e o início da dança.
- adaptação da aplicação ao meio televisivo, ou a qualquer outro que possua um reprodutor de música a que se possa aliar a imagem.
- preparar a aplicação de forma a poder ser utilizada em rede por vários utilizadores, aumentando o fator social.
- possibilidade de personalização de aspetos diferentes do avatar como características morfológicas que influenciem a dança.
- definição de regras que controlam a sucessão de unidades de batida numa música, de forma a permitir a definição de coreografias conhecidas ou, no mínimo, coerentes.
- entrada de dados recolhidos por elementos externos, como o microfone.
- criação de animações através do método de captura de movimento.
- implementação de uma linha de tempo na interface, em que o utilizador pudesse escolher a altura da música que quer ouvir.

Pode, desta forma, concluir-se que, apesar do trabalho desenvolvido até ao momento e dos seus resultados satisfatórios, existe ainda bastante margem para melhoramentos e investigação mais profunda. É ainda um projeto que se pode adaptar e agregar a outros de áreas diferentes.

Referências

- [AB98] J.M. Allbeck e N.I. Badler. Avatars á la Snow Crash. In *Proceedings Computer Animation '98 (Cat. No.98EX169)*, pages 19–24. IEEE Comput. Soc, 1998.
- [ARCN09] Senaka Amarakeerthi, Rasika Ranaweera, Michael Cohen e Nicholas Nagel. Mapping selected emotions to avatar gesture. *Proc. IWAC*, 1, 2009.
- [BA00] N.I. Badler e J.M. Allbeck. Towards behavioral consistency in animated agents. *Departmental Papers (CIS)*, 68:213, 2000.
- [Bai06] JN Bailenson. Transformed social interaction in collaborative virtual environments. *Digital media: Transformations in human communication*, pages 255–264, 2006.
- [BBET97] Ronan Boulic, Pascal Bécheiraz, Luc Emering e Daniel Thalmann. Integration of motion control techniques for virtual human and avatar real-time animation. In *Proceedings of the ACM symposium on Virtual reality software and technology*, number September, pages 111–118. ACM, 1997.
- [BDA⁺05] JP Bello, Laurent Daudet, Samer Abdallah, C. Duxbury, M. Davies e M.B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, September 2005.
- [BNZP07] Koray Balci, Elena Not, Massimo Zancanaro e Fabio Pianesi. Xface open source project and smil-agent scripting language for creating and animating embodied conversational agents. In *Proceedings of the 15th international conference on Multimedia - MULTIMEDIA '07*, page 1013, New York, New York, USA, 2007. ACM Press.
- [BPW93] Norman I. Badler, Cary B. Phillips e Bonnie Lynn Webber. *Simulating humans: computer graphics animation and control*. Oxford University Press, Inc., New York, NY, USA, 1993.
- [CB05] EWG Clua e J.R. Bittencourt. Desenvolvimento de Jogos 3D: Concepção, Design e Programação. In *XXIV Jornadas de Atualização em Informática (JAI) Part of XXIV Congresso da Sociedade Brasileira de Computação*, pages 22–29, 2005.
- [dAT08] Vicente Ruiz de Angulo e Carme Torras. Learning inverse kinematics: reduced sampling through decomposition into virtual robots. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 38(6):1571–7, December 2008.
- [DWYW09] Nicolas Ducheneaut, M.H. Wen, Nicholas Yee e Greg Wadley. Body and mind: a study of avatar personalization in three virtual worlds. In *Proceedings of the 27th*

REFERÊNCIAS

- international conference on Human factors in computing systems*, pages 1151–1160, New York, New York, USA, 2009. ACM.
- [GAO07] Quan-Sheng GAO. A Method for Character Animation Generation and Control. *Journal of Software*, 18(9):2356, 2007.
- [GEKO09] David Grunberg, Robert Ellenberg, Youngmoo Kim e Paul Oh. Creating an autonomous dancing robot. *Proceedings of the 2009 International Conference on Hybrid Information Technology - ICHIT '09*, pages 221–227, 2009.
- [GM96] Masataka Goto e Yoichi Muraoka. Beat tracking based on multiple-agent architecture-a real-time beat tracking system for audio signals. In *Proceedings of the Second International Conference on Multiagent Systems*, pages 103–110, 1996.
- [GNM97] Masataka Goto, R. Neyama e Y. Muraoka. RMCP: Remote Music Control Protocol. In *Proceedings of the ICMC*, pages 446–449, 1997.
- [Got07] M. Goto. Active music listening interfaces based on signal processing. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–1441. IEEE, 2007.
- [Hem06] Paul Hemp. Avatar-based marketing. *Harvard Business Review*, 84(6):48–57, 2006.
- [HW10] S. Huh e D. Williams. Dude looks like a lady: Gender swapping in an online game. *Online worlds: Convergence of the real and the virtual*, pages 161–174, 2010.
- [KB01] Sonu Chopra Khullar e Norman I. Badler. Where to look? automating attending behaviors of virtual human characters. *Autonomous Agents and Multi-Agent Systems*, 4:9–23, 2001. 10.1023/A:1010010528443.
- [KFC07] Y.B. Kafai, D.A. Fields e Melissa Cook. Your second selves: avatar designs and identity play in a teen virtual world. In *Proceedings of DIGRA*, volume 2007, 2007.
- [KM95] I.a. Kakadiaris e D. Metaxas. 3D human body model acquisition from multiple views. In *Proceedings of IEEE International Conference on Computer Vision*, number June, pages 618–623. IEEE Comput. Soc. Press, 1995.
- [KWS07] Gunwoo Kim, Yan Wang e Hyewon Seo. Motion Control of a Dancing Character with Music. *Computer, (Icis)*, 2007.
- [Mac] Machinimart. Maya vs. 3ds Max. <http://machinimart.com/maya-vs-3ds-max/>. [accessed Jun 18, 2012].
- [Mac08] John V. MacConaill, Michael Aloysius; Basmajian. *Muscles and movements: a basis for human kinesiology*. Universidade de Michigan, 2008.
- [Net09] João Neto. *Framework para geração de personagens humanóides 3D para jogos computacionais*. PhD thesis, Universidade Federal de Pernambuco, 2009.
- [NL10] Luiz Naveda e Marc Leman. The spatiotemporal representation of dance and music gestures using Topological Gesture Analysis (TGA). *Music Perception*, 28(1):93–111, September 2010.

REFERÊNCIAS

- [NNIY02] Atsushi Nakazawa, Shinichiro Nakaoka, Katsushi Ikeuchi e Kazuhito Yokoi. Imitating human dance motions through motion structure analysis. In *IEEE/RSJ International Conference on Intelligent Robots and System*, volume 3, pages 2539–2544. IEEE, 2002.
- [NZB00] T. Noma, L. Zhao e N.I. Badler. Design of a virtual human presenter. *IEEE Computer Graphics and Applications*, 20(4):79–85, 2000.
- [OCFT⁺08] Ferda Ofli, C. Canton-Ferrer, J. Tilmanne, Yasemin Demir, E. Bozkurt, Y. Yemez, Engin Erzin e A. M. Tekalp. Audio-driven human body motion analysis and synthesis. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2233–2236. IEEE, March 2008.
- [ODY⁺08] Ferda Ofli, Yasemin Demir, Yücel Yemez, Engin Erzin, A Murat Tekalp, Koray Balci, İdil Kızıoğlu, Lale Akarun, Cristian Canton-Ferrer, Joëlle Tilmanne, Elif Bozkurt e A Tanju Erdem. An audio-driven dancing avatar. *Journal on Multimodal User Interfaces*, 2(2):93–103, May 2008.
- [OGMR10] Lobato Oliveira, Fabien Gouyon, L.G. Martins e L.P. Reis. IBT: A real-time tempo and beat tracking system. *ISMIR 2010*, (Ismir):291–296, 2010.
- [ONG10] J.L. Oliveira, L.A. Naveda e Fabien Gouyon. Synthesis of variable dancing styles based on a compact spatiotemporal representation of dance. *Robots and Musical*, (i), 2010.
- [RB12] Christian Rau e Guido Brunneth. Anatomically correct adaption of kinematic skeletons to virtual humans. In *GRAPP/IVAPP*, pages 341–346, 2012.
- [Ric09] Richard Jones. Last.fm Radio Announcement. <http://blog.last.fm/2009/03/24/lastfm-radio-announcement>, 2009. [accessed Jun 18, 2012].
- [RZS10] Cheng Ren, Liming Zhao e Alla Safonova. Human Motion Synthesis with Optimization-based Graphs. *Computer Graphics Forum*, 29(2):545–554, June 2010.
- [SM05] H.C. Sun e D. Metaxas. Animation of human locomotion using sagittal elevation angles. *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications*, pages 429–430, 2005.
- [Ste00] N Stephenson. *Snow crash*. Spectra, United States, 2000.
- [Tay96] Richard. Taylor. *The encyclopedia of animation techniques / Richard Taylor*. Simon & Schuster, East Roseville, NSW., 1996.
- [TWR⁺05] Gerard J To, John Wiley, Bonnie Roesch, Lisa Wojcik, Brian Salisbury, Karin Gerdes e Howard Grossman. *Principles of Anatomy and Physiology*. Wiley, 11th edition, 2005.
- [YB06] Nick Yee e J.N. Bailenson. Walk a mile in digital shoes: The impact of embodied perspective-taking on the reduction of negative stereotyping in immersive virtual environments. *Proceedings of PRESENCE*, pages 24–26, 2006.
- [YB07] Nick Yee e Jeremy Bailenson. The Proteus Effect: The Effect of Transformed Self-Representation on Behavior. *Human Communication Research*, 33(3):271–290, July 2007.

REFERÊNCIAS

- [YNT⁺07] Kazuyoshi Yoshii, Kazuhiro Nakadai, Toyotaka Torii, Yuji Hasegawa, Hiroshi Tsujino, Kazunori Komatani, Tetsuya Ogata e Hiroshi G. Okuno. A biped robot that keeps steps in time with musical beats while listening to music with its own ears. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1743–1750. IEEE, October 2007.