



CATÓLICA  
ESCOLA DAS ARTES

---

PORTO

## ESTÁGIO – SABER PORTO

Relatório de Estágio apresentado à Universidade Católica Portuguesa  
para obtenção do grau de Mestre em Som e Imagem

*Miguel Carriço*

Porto, Setembro de 2023



CATOLICA  
ESCOLA DAS ARTES

---

PORTO

## ESTÁGIO – SABER PORTO

Relatório de Estágio apresentado à Universidade Católica Portuguesa  
para obtenção do grau de Mestre em Som e Imagem

- Especialização em -  
Animação por Computador

*Miguel Carriço*

Trabalho efetuado sob a orientação de

Prof. Sahra Kunz

Porto, Setembro de 2023

## **Dedicatória**

Quero dedicar este trabalho à minha família, amigos e namorada.

Aos meus pais, Henrique Carriço e Cristina Carriço, que me proporcionaram a oportunidade de integrar no mestrado e sempre me apoiaram nas minhas escolhas de estudos e carreira profissional. Por me incentivarem a ser, cada vez mais, a melhor versão de mim mesmo. Ao meu irmão Filipe Carriço, por me apoiar e partilhar o entusiasmo pelos nossos gostos. Por me ensinar que devemos trabalhar para aquilo que mais gostamos.

Ao meu amigo João Cardoso por me acompanhar nas várias etapas da minha vida, incluindo este mestrado. Por ser uma força positiva que me motiva a fazer sempre mais e melhor.

À minha namorada Margarida Santos que partilha comigo todos os momentos bons e maus. É alguém que posso sempre encontrar conforto e carinho nos momentos de maior ansiedade.

A todas as pessoas especiais na minha vida que me ajudaram em cada circunstância, fico muito agradecido pelo apoio e dedicação.

## **Agradecimentos**

Gostaria de agradecer de uma forma geral a todas as pessoas que direta ou indiretamente estiveram envolvidos neste projeto. Pela possibilidade de ter concretizado o estágio curricular no estúdio a que me propus, que no fim me acabou por contratar.

Um agradecimento ao professor Ricardo Megre, pela disponibilidade, pela franqueza na sua orientação numa fase inicial deste relatório. Um agradecimento à professora Sahra Kunz pela orientação, pela disponibilidade e pelo apoio que me deu nesta fase final do meu percurso neste mestrado.

Gostaria de agradecer à *Saber* Porto por ter aceite a minha proposta de estágio, por terem demonstrado interesse e confiança em mim e pela experiência que me proporcionaram durante os seis meses de estágio. Em especial gostava de agradecer ao Team Leader da equipa de animação Pedro Oliveira, que desde o início demonstrou paciência e dedicação na minha instrução. Ao Daniel Vicente pelo companheirismo e apoio constante. Ao Romain Perrault e Roc Bisbal pela orientação e boa disposição.

Por fim queria também agradecer aos meus colegas de mestrado que partilhamos muitos dos mesmos entraves, por me fazerem companhia nesta jornada e por nos puxarmos uns aos outros para cima.

## Resumo

Este relatório insere-se no contexto de animação em videojogos reportando uma experiência de estágio no seio da equipa de animação do estúdio da *Saber* Porto Este estágio permitiu a experiência de trabalhar numa produção profissional como animador 3D. Foi possível experienciar as dinâmicas e técnicas de uma equipa de animação, que produz desde *rigs* a animação 3D utilizando também a tecnologia de *motion capture*, usando diversos programas de animação 3D e um motor de jogos.

O estágio decorreu durante seis meses e foi possível a execução de dezassete tarefas que envolveram não só a prática de animação, mas como implementação de elementos num motor de jogos, tendo assim uma versatilidade e variedade nas tarefas de um animador técnico permitindo o ajuste do aluno e do estúdio em diferentes vertentes da *pipeline* de animação.

**Palavras-Chave:** Animação 3D, Videojogos, *Rig*, *Mocap*, *keyframing*

## **Abstract**

This document reports the experience of an internship done in a team of animation at the video-game studio *Saber* Porto. This internship allowed for the experience of working on a professional production as a 3D animator. It was possible to experience all the techniques and dynamics of an animation team, whose production ranges from creating rigs to 3D animations, using the technology of motion capture, several 3D animation software, and a game engine.

The internship lasted six months and there were seventeen tasks that were completed in total. These tasks involved, not only the practice of animation, but also the implementation of some elements in a game engine. This way, it was possible to have versatility and variety in the tasks of a technical animator, allowing for an adjustment between the student and the studio within the varied work of the animation pipeline.

**Keywords:** 3D Animation, Videogames, Rig, Mocap, keyframing

# Índice

1. Introdução.....	10
1.1 <i>Saber Interactive</i> .....	10
1.2 Videojogos <i>AAA</i> e Animação .....	11
1.3 Objetivos e Tarefas .....	12
1.4 Estado da Arte.....	13
1.5 Estrutura do relatório .....	16
2. Cronograma e Planeamento.....	17
2.1 Objetivos e tarefas: .....	17
2.2 Cronograma .....	21
2.3 Conclusão de Capítulo .....	22
3. Desenvolvimento .....	23
3.1 Referências.....	23
3.2 <i>Rigging</i> .....	24
3.3 <i>Motion Capture (mocap)</i> .....	28
3.4 Tratamento de dados <i>mocap</i> .....	29
3.6 Conclusão do Capítulo.....	39
4. Ilações Finais e Trabalho Futuro.....	40
5. Referências e Bibliografia.....	41

## Lista de Figuras

Figura 1- Atriz a usar um fato <i>mocap</i> .....	13
Figura 2- Transmissor e um ator a usar um fato com sensores magnéticos .....	14
Figura 3- Ator com um fato de <i>mocap</i> e câmaras por trás. O resultado digital à direita	14
Figura 4- Logótipo do Unreal Engine.....	18
Figura 5- Logótipos do Motion Builder e Maya .....	20
Figura 6- Cronograma .....	21
Figura 7- Exemplo de utilização de duas referências.....	23
Figura 8- Ferramenta de criação de <i>Joints</i> no Maya .....	24
Figura 9- Opções automáticas de distribuir influências nos vértices .....	25
Figura 10- Representação visual da influência de uma <i>joint</i> do ombro de uma personagem numa mesh de roupa .....	25
Figura 11- Exemplo de um <i>Rig</i> com controladores customizados.....	26
Figura 12- Manipuladores <i>forward kinematics</i> no braço permitindo a rotação independente de cada joint .....	27
Figura 13- Manipulador <i>inverse kinematics</i> do pé permitindo a manipulação da perna apenas com um controller.....	27
Figura 14- Área de trabalho do Motion Builder.....	29
Figura 15- Diferentes sistemas de <i>joints</i> para efeitos de Retargeting.....	29
Figura 16- Criação de um <i>control rig</i> no Motion Builder.....	30
Figura 17- <i>Storymode</i> no Motion builder .....	31
Figura 18- Editor de curvas de animação .....	31
Figura 19- Exemplo de uma correção usando a translação do controlador do pélvis ....	32
Figura 20- <i>Control rig</i> do Motion Builder.....	32
Figura 21- Filtros proporcionados pelo Motion Builder e Biblioteca de poses.....	33
Figura 22- Opções de <i>Bake</i> e <i>Export</i> .....	34
Figura 23- Exemplo de uma curva com ease out acentuado .....	35
Figura 24- Exemplo de <i>squash and stretch</i> numa personagem 3D .....	36
Figura 25- <i>Spritesheet</i> de uma animação do StreetFighter.....	36
Figura 26- Dois exemplos de simulação de físicas na roupa de uma personagem.....	37
Figura 27- Interface do Maya .....	37
Figura 28- <i>Graph Editor</i> do Maya.....	38
Figura 29- <i>Tweener plug-in</i> para o Maya .....	38

## Glossário

*Pipeline* – Processo ou conjunto de processos;

*AAA* – Standard de qualidade de um videogame feito com um grande orçamento;

*Rigging* – Processo de estabelecer uma ligação entre um conjunto de *Joints* e uma *Mesh*;

*Motion capture/Mocap* – Captura digital de movimentos através de um sistema de câmaras e sensores;

*Unreal Engine /Game Engine/ Engine* – *Software* usado para a criação de videogames;

*Moodboards* – Conjunto de vários tipos de referências;

*Keyframing* – Tipo de animação na qual se usa interpolações entre os estados inicial e final do objeto;

*Maya Motion Builder* – *Software* de animação 3D

*Joints* – Esqueleto criado para manipulação de uma *mesh*;

*Encoder* – Objeto de captura de informação num sistema de *mocap*;

*Mesh* – Estrutura 3D de um modelo constituída por polígonos;

*Looping* – A repetição de algo, normalmente uma sequência de imagens;

*Skinning* – Processo de atribuição de influências de uma *mesh* a *joints*;

*Takes* – Gravações de performances normalmente de atores;

*Controllers* – Manípulos criados para guardar a informação da animação;

*Retargeting* – Processo de transferência de informação de animação entre dois sistemas diferentes;

*Bake/Plot* – Gravação da informação em todos os *frames* da animação

*Frames* – Uma imagem numa sequência de imagens. Em 3D também poderá ser a unidade base de uma animação.

*T-Pose* – Uma posição geralmente de uma pessoa com os braços esticados num ângulo de 90° em relação ao corpo criando a forma de um T;

*Ease in* – Começo de uma animação de uma forma lenta que vai acelerando;

*Ease out* – Fim de uma animação partindo de um estado rápido que vai ficando cada vez mais lento à medida que acaba;

Pintar pesos – Atribuição de influências de uma *mesh* a um esqueleto usando um pincel digital;

Fbx – Formato de um ficheiro usado em software 3D

## 1. Introdução

A área de trabalho desenvolvida neste relatório foi a animação para videojogos. A entidade acolhedora para o estágio curricular foi a *Saber* Porto. Os objetivos deste relatório são de documentar e relatar todos os aspetos que um animador precisa de conhecer para desenvolver videojogos *AAA* em estúdios profissionais de videojogos.

Devido aos direitos do estúdio de um contrato de confidencialidade, este relatório não pode relatar e identificar determinadas tarefas que foram realizadas. Consequentemente o relatório irá tocar sobre assuntos de um ponto de vista mais técnico e sempre que possível apresentar a aplicação das técnicas desenvolvidas com um exemplo real aplicado no estágio ou um exemplo feito posteriormente que ilustre a tarefa desenvolvida sem colocar a confidencialidade em causa.

### 1.1 *Saber Interactive*

A *Saber Interactive* emprega mais de dois mil e quinhentos trabalhadores em mais de vinte estúdios globalmente, conhecidos pelo desenvolvimento dos jogos *Warhammer 40,000: Space Marine 2*, *World War Z*, *Evil Dead: The Game*. A *Saber* Porto é um dos mais recentes investimentos do grupo *Saber Interactive*, localizando-se em Vila Nova de Gaia e sendo composta atualmente por três edifícios com mais de cem trabalhadores. (Saber Interactive, 2023). A *Saber* Porto é responsável pelo desenvolvimento do jogo *Dakar Desert Rally* e agora está a desenvolver um novo projeto no qual este estágio se enquadrou.

Foi feita a integração numa equipa composta por quatro elementos, o *Team Leader* que coordena a equipa, um *Senior Animator* e um *Senior Technical Animator* e um *Junior Animator*. Geralmente, a equipa teve uma reunião de duas em duas semanas para discutir sobre o que precisava de ser feito nos próximos quinze dias. Dentro desta equipa, o *Team Leader* tem a responsabilidade de coordenar e rever todo o trabalho produzido, os *Seniors* têm trabalho especializado que é tecnicamente mais exigente e o *Junior* em conjunto com o aluno completam tarefas de animação e implementação de *rigs* ou dessas mesmas animações no *game engine*.

Um videojogo *AAA* é uma produção de uma grande dimensão que requer também um grande investimento. Dentro dos vários departamentos da *Saber* Porto, o departamento de animação tem acesso a programas de animação 3D e uma sala de *mocap*.

Cada trabalhador tem um posto de trabalho composto por um computador, dois ou três ecrãs, teclado, rato e um controlador para testar os jogos.

Apesar da equipa de animação da *Saber* Porto ser pequena encontra-se em expansão. Devido à necessidade de adquirir profissionais para trabalhar nos seus atuais projetos, foi possível entrar na produção desde cedo no estágio.

## 1.2 Videojogos AAA e Animação

A definição de videojogos AAA foi mudando ao longo dos anos tendo ela várias interpretações, mas geralmente um jogo AAA é visto como um *standard* elevado da indústria. São jogos tipicamente publicados por empresas bem conhecidas, como a Ubisoft, Blizzard ou Rockstar Games, que produzem jogos de propriedades intelectuais já bem estabelecidas com grandes orçamentos.

Os videojogos são das formas mais rentáveis da área de entretenimento, sendo a animação de personagens uma grande parte do seu desenvolvimento. Muitas personagens de videojogos são icónicas devido ao seu design e à sua animação, que representa muito da personalidade dessas personagens. Desde os clássicos como Super Mário (Nintendo, 1985) e Sonic (Sega, 1991) até a Kratos do God of War (Santa Monica, 2005) ou a Alloy do Horizon Zero Dawn (Guerrilla Games, 2017), a animação é o que dá a vida a personagens digitais. Este estágio permitiu uma intervenção prática no processo por detrás da animação de personagens deste estilo.

A animação de personagens era a parte específica do processo que se pretendia aprofundar apesar de todos os processos e tarefas que revolvem à sua volta serem aqui documentados.

Completo-se uma licenciatura em Jogos Digitais e Multimédia no Instituto Politécnico de Leiria antes de ingressar no mestrado em Som e Imagem na especialização de Animação por Computador da Universidade Católica Portuguesa. Querendo complementar os conhecimentos gerais de videojogos com um aprofundamento de técnicas de animação, ingressou-se no mestrado com o objetivo de se candidatar a um estágio profissional num estúdio de videojogos.

### 1.3 Objetivos e Tarefas

Este estágio teve como objetivo a preparação para o mercado de trabalho profissional na área dos videojogos integrando uma equipa profissional de animação. A possibilidade de integrar o desenvolvimento de um projeto como um videojogo *AAA*, contribuiu para o crescimento pessoal e profissional. Por fim, foi possível documentar as técnicas e estratégias aprofundadas durante o estágio, adotando-as e também desenvolvendo novas.

Inicialmente houve um teste de competências onde foram realizados pequenos trabalhos com os programas utilizados pelo estúdio. Os testes decorreram nas primeiras semanas de estágio e consistiram de tarefas fundamentais para a familiarização com a metodologia e cuidados a ter com a realização das mesmas. O objetivo seria a introdução às ferramentas básicas e necessárias para realizar o estágio, tais como os programas de animação, *game engine* e programas de gestão utilizados pelo estúdio, que são precisas para realização de futuras tarefas para o jogo.

Após os testes, deu-se a entrada na produção realizando uma variedade de tarefas, como criação de *rigs*, limpeza de *motion capture* e animação *keyframe*, que foram de encontro às propostas iniciais feitas na entrega da Proposta de Estágio.

## 1.4 Estado da Arte

Dentro da área dos videojogos existem vários géneros, dos mais estilizados aos mais realistas. O trabalho inseriu-se no âmbito de animação realista de personagens para jogos digitais. Há duas abordagens para fazer este tipo de animação, animar a partir de *motion capture (mocap)* ou através de interpolação (*keyframing*).

“*Mocap* é uma tecnologia de gravação digital de movimentos específicos de uma pessoa (como um ator) e traduzi-los para animação. Em *keyframe animation* as *keyframes* são animadas manualmente pelo animador começando com uma personagem parada” (Newman, 2020).

Existem vários sistemas de captura de *mocap* que podem funcionar com câmaras e sensores. “Motion Capture é a gravação do movimento do corpo humano (ou outro movimento) para visualização imediata ou análise e reprodução posterior” e, “O uso de captura de movimento para animação de personagens por computador é relativamente novo, tendo começado no final dos anos 1970, e só agora começa a se tornar difundido.” (Sturman, 1994: 1) Várias técnicas foram desenvolvidas para obter os melhores resultados possíveis através da captura de movimento. Mocap mecânico: “Esta técnica de captura de movimento é conseguida através do uso de um exoesqueleto. Cada *joint* é então conectada a um *encoder* angular. O valor do movimento de cada *encoder* (rotação) é registado por um computador que conhecendo a posição relativa dos *encoders (joints)* pode reconstruir estes movimentos num ecrã ao usar um *software*. Um *offset* é aplicado a cada *encoder* porque é muito difícil acertar exatamente a sua posição com a sua relação real”. (Rhaul, 2018: 2)



Figura 1- Atriz a usar um fato mocap

Com *mocap* é preciso primeiro contratar atores que possam representar a personagem pretendida. Os atores vestem os fatos de captura e realizam os movimentos necessários para posteriormente serem trabalhados pelo animador.

“Sistemas de *magnetic mocap* também são às vezes chamados de *magnetic trackers*. Com um sistema *mocap* magnético, doze a vinte sensores são colocados num sujeito de captura para medir a relação espacial a um transmissor magnético. Os sensores de monitorização produzem as suas translações e rotações” (Kitagawa, 2008: 10)



Figura 2- Transmissor e um ator a usar um fato com sensores magnéticos

*Optical Motion Capture*. “A captura é baseada numa gravação ótica de várias câmaras sincronizadas, a síntese das coordenadas de (X, Y) do mesmo objeto de diferentes ângulos possibilita a dedução das coordenadas (X, Y, Z). Este método envolve a consideração de vários problemas complexos como *parallax* ótico, distorção da lente usada, etc. O sinal então sofre muitas interpolações. No entanto, uma calibração correta destes parâmetros vai ajudar numa alta precisão dos dados colecionados” (Rhaul, 2018: 2)

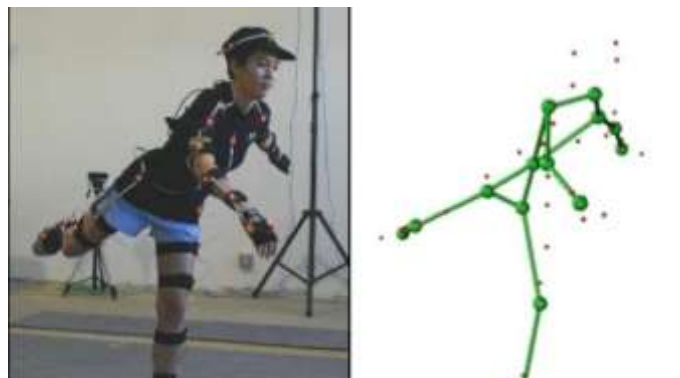


Figura 3- Ator com um fato de mocap e câmaras por trás. O resultado digital à direita

A animação por *keyframing* implica começar a animação do zero e animar apenas com base nas ferramentas criadas na fase de *rigging*.

O *mocap* pode ajudar a animação a chegar a um alto nível de realismo rapidamente, porém perde-se a facilidade em manipular a personagem para fazer exatamente aquilo que o artista quiser.

Segundo o vídeo produzido pelo canal do *YouTube Inside Gaming*, que explica como funciona a captura de movimentos, “*Motion Capture* tornou-se a técnica standard para filmes e videogames com grandes orçamentos para produzir efeitos especiais e personagens credíveis” (Inside Gaming, 2021). Para qualquer jogo de grande orçamento que contenha personagens humanoides realistas, muito provavelmente é utilizado *mocap* uma vez que é a maneira mais eficaz de captar movimento realista e transformá-lo em animação.

O realismo na animação pode nem sempre ser a melhor escolha, na medida em que pode criar personagens com pouca personalidade devido à restrição no seu movimento que é condicionado pelas forças da natureza podendo resultar em estranheza. Este fenómeno normalmente associado ao *uncanny valley* introduzido pela roboticista Masahiro Mori (Mori, 1970) para descrever um acontecimento idêntico quando os robôs eram construídos com características demasiado realistas. Jogos de nova geração sofrem com o uso desta tecnologia sendo que o objetivo é chegar cada vez mais perto de movimentos realistas imitando a realidade. “Quando os gráficos ultras realísticos estavam a ser perseguidos, nós tivemos resultados muito interessantes como *Heavy Rain*” (Quantic Dream, 2010) “onde expressões faciais e movimentos eram estranhos” (Rose, 2019)

Analisando estas técnicas, é benéfico desenvolver ambas para um melhor entendimento das possibilidades de um animador. Os videogames introduzem estes dois tipos de animação, e no caso da Saber Porto, os projetos requerem conhecimento tanto de *keyframing* como de *mocap*.

## 1.5 Estrutura do relatório

O relatório refere todo o processo do estágio e o seu planeamento. Também inclui várias partes mais informativas que descrevem processos e introduzem comparações.

Primeiramente são apresentadas as tarefas e o cronograma do estágio explicando todos os processos que serão desenvolvidos, incluindo a descrição das tarefas e como elas decorreram introduzindo já alguns conceitos que serão explorados posteriormente.

No desenvolvimento, as tarefas são descritas com mais precisão, entrando mais em detalhe no que diz respeito a certos conceitos e funcionamento de *software*. Sendo impossível a exata descrição das tarefas realizadas no estágio, são utilizados exemplos alternativos que ilustram o conceito e o processo das tarefas sem quebrar confidencialidades.

No final é feita uma conclusão sobre o trabalho feito e quais os seus resultados sobre desenvolvimento de técnicas, obtenção de experiência e resultados consequentes do trabalho em estágio.

Esta estrutura permite uma leitura organizada deste relatório que começa por introduzir temas que possam ser desconhecidos ao leitor e por fim explicar em maior detalhe todo o processo.

## 2. Cronograma e Planeamento

### 2.1 Objetivos e tarefas:

#### **Integração na equipa.**

Alguns dos objetivos principais na realização deste estágio foram: conhecer colegas e local de trabalho, também como as principais técnicas e metodologias utilizadas, ganhar uma noção do ritmo de trabalho e em que tarefas ele se insere e troca de ideias através de comunicação com colegas promovendo um ambiente de entreajuda.

As tarefas foram divididas por dificuldade e tipo, que definiam o tempo de duração de cada uma. As tarefas mais específicas foram direcionadas para os trabalhadores mais experientes. Geralmente estas tarefas demoravam mais tempo do que as tarefas mais gerais. Normalmente um elemento era responsável por um tipo de tarefa específico, como criação de *rigs*, para que todo o processo fosse uniforme.

Qualquer tarefa realizada sofria um processo de revisão feito pelos colegas. A revisão consistia numa avaliação do trabalho apontando falhas ou inconsistências em relação ao objetivo desejado. Apenas quando todos os detalhes apontados nas revisões fossem abordados e concluídos, se podia dar como terminada essa tarefa.

#### **Pesquisa de Referências.**

Um animador tem como dever inserir-se no contexto do projeto que está a ser realizado. Deve pesquisar tudo o que estiver relacionado com estilo artístico, caso o jogo seja estilizado ou realista, técnicas e ferramentas utilizadas, como *scripts* (código que possa ser aplicado num *software* permitindo ações que não vêm de raiz) e ferramentas que podem ajudar o processo, e criação de *moodboards* (um conjunto de imagens, materiais ou texto que ajudam a definir um estilo ou um conceito) partilhado pela equipa para que todos possam estar em sintonia com o projeto.

Deve ainda encontrar ou criar referências visuais que ajudem no desenvolvimento do projeto-, tais como vídeos ou imagens que contenham movimentos e poses que ajudem à criação de animações realistas.

### **Criação de *Rigs*.**

Trata-se de um estudo de proporções e fisionomia da personagem, de forma a criar um esqueleto e manípulos que permitam controlar as personagens.

“O esqueleto é uma estrutura que permite ao animador um controlo sobre as *motion dynamics* em animação *keyframe*” (Burtnyk, 1976). O esqueleto, em conjunto com os manípulos, torna o processo de animação mais fluido diminuindo assim o tempo de trabalho. É uma técnica extremamente necessária no que toca a animação 3D, pois sem a sua existência não seria possível criar muitos dos movimentos que as personagens apresentam.

O *Rigging* requer um modelo 3D vindo do departamento de modelação. Com base no modelo o animador cria um esqueleto com referência em estudos do mundo real ou que satisfaçam as necessidades do seu propósito.

*Rigging*, por si só, pode ser um cargo dentro de um estúdio de videojogos executado por um profissional que não seja necessariamente um animador. Apesar disto, devem trabalhar juntos, visto que o trabalho de um criador de *rigs* influencia diretamente o trabalho do animador.

### **Implementação no *Engine***

Um motor de jogos ou *game engine* é um programa especificamente dedicado à criação de jogos. Todo o processo de criação de jogos concentra-se no *game engine* que processa todos os *assets* e toda a programação criando o produto final.

O *game engine* escolhido pela Saber Porto é o Unreal Engine, que segundo o seu site, “é a ferramenta de criação 3D em tempo real mais aberta e avançada do mundo” (Epic Games, 2023). O Unreal Engine é utilizado com muita regularidade em jogos *AAA*, sendo alguns dos mais conhecidos o Fortnite (Epic Games, 2017) e Final Fantasy VII Remake (Square Enix, 2020). Este motor de jogos tornou-se num *standard* da indústria devido à constante evolução por parte dos criadores e também à liberdade que eles promovem,



*Figura 4- Logótipo do Unreal Engine*

permitindo aos estúdios desenvolver as suas próprias ferramentas e versões do *engine*. Por fim, a sua popularidade também surge devido ao facto de ser grátis e disponível para qualquer pessoa utilizar, seja esta parte de uma grande empresa ou o utilize apenas para uso próprio.

A implementação de personagens e animações no jogo deve ser feita de forma adequada com base em implementações anteriormente realizadas. A forma como as implementações vão sempre atualizando tornam importante manter um documento atualizado com cada mudança que é feita no seu processo.

### ***Motion Capture***

Foi feita captura de movimentos usando o sistema de *mocap* nos estúdios da *Saber Porto*. O sistema de *mocap* da *Saber Porto* é um sistema ótico, composto por câmaras e fatos com sensores que se colocam em cada *joint* que será necessária.

Fez-se a calibração do sistema e dos fatos. Foram contratados atores para representarem, vestidos com os fatos de *Mocap*, tendo estes recebido as instruções necessárias para a execução das suas ações. Foi feita a gravação de vários *takes* até obter o resultado desejado, posteriormente organizando cada gravação de forma a perceber que movimentos foram gravados e para que personagem se destinariam.

### **Tratamento dos dados *Mocap***

A maior parte das animações que vão ser implementadas no jogo serão com base em *Mocap*, por isso foi necessário adequar as capturas feitas, em animações fluidas e sem falhas, prontas a serem implementadas no jogo. O tratamento de dados foi feito em *Motion Builder*, que é um *software* da Autodesk específico para animação *mocap*. A versão de *Motion Builder* utilizada durante este estágio foi a de 2019. Este *software* foi comprado pela Autodesk para o desenvolvimento de cinematografia virtual e *mocap*. As funcionalidades que este programa apresenta irão ser mais aprofundadas ao longo deste relatório. Alguns dos tratamentos que a informação pode ter são *filtragens*, *looping* ou correção de movimentos não naturais.

### **Animação *Keyframe***

A animação foi feita de raiz usando Maya ou Motion Builder. Estes programas são usados para animação 3D e ambos têm os seus prós e contras. O Maya tem mais ferramentas e *plugins* que permitem um processo de animação mais simples e rápido. O Motion Builder, sendo um *software* mais específico para *mocap* tem ferramentas mais limitadas para animação *keyframe*, mas permite um enquadramento fácil se for necessário introduzir uma animação *keyframe* entre animações *mocap*.



*Figura 5- Logótipos do Motion Builder e Maya*

### **Criação de lógica para animação**

A implementação de personagens num *game engine* necessita de uma estrutura lógica para que possa funcionar, por isso parte do trabalho de um animador técnico é criar esta lógica. As propriedades da importação para o *engine* precisam de ser ajustadas dependendo do *software* de animação. No *engine* é necessário criar uma lista de personagens que partilhem o mesmo sistema de *rig* e aplicar a animação a esse sistema para que essas personagens possam reproduzir as animações.

### **Atualização de progresso e Correções**

O estúdio funciona com um sistema de progresso em que cada trabalhador vai atualizando o tempo que demora em cada tarefa. Quando conclui a tarefa deve dizer que esta está terminada ou pronta para revisão. As revisões são feitas por colegas que sejam tecnicamente capazes de o fazer, apontando detalhes a melhorar ou falhas para corrigir. As correções são feitas após as revisões, fazendo com que a tarefa volte a ser revista. Poderão existir momentos em que a equipa se reúne e analisa o trabalho feito até ao momento sugerindo maneiras de melhorar ou decidir mudar/acrescentar algo por completo.

## 2.2 Cronograma

Para simplificar e manter a integridade do trabalho desenvolvido foi dado um número às tarefas e estas foram divididas em cinco categorias: *Rig*, *Implementação*, *Mocap*, *Bugs* e *Keyframe Animation*.

	Nov.	Dezembro	Janeiro	Fevereiro	Março	Abril	Maio
Tarefa 1 (Rig)	█						
Tarefa 2 (Implementação)	█						
Tarefa 3 (Mocap)		█					
Tarefa 4 (Rig)		█					
Tarefa 5 (Implementação)		█					
Tarefa 6 (Mocap)		█	█				
Tarefa 7 (Rig)			█				
Tarefa 8 (Implementação)			█				
Tarefa 9 (Bugs)			█				
Tarefa 10 (Mocap)				█			
Tarefa 11 (Keyframe Anim)				█	█	█	█
Tarefa 12 (Rig)						█	
Tarefa 13 (Implementação)						█	
Tarefa 14 (Rig)						█	
Tarefa 15 (Implementação)						█	
Tarefa 16 (Mocap)						█	█
Tarefa 17 (Keyframe Anim)							█

Figura 6- Cronograma

### 2.3 Conclusão de Capítulo

Neste capítulo foram descritos os objetivos e tarefas principais delineadas para este estágio. A aprendizagem da metodologia utilizada pelo estúdio e pelas pessoas com as quais se interage é fundamental na integração de uma equipa de desenvolvimento de videojogos. Neste caso a equipa de animação, a sua metodologia, horário e *software*.

A gestão de tempo e expectativas em relação ao resultado final de cada trabalho é importante para que se possam fazer estimativas realistas em relação ao tempo que o animador tem para concretizar a sua tarefa. Estas estimativas devem incluir o tempo que os animadores passam a rever o trabalho uns dos outros, já que, a revisão é uma parte muito importante na filtragem e correção de erros.

O conhecimento dos sistemas de *mocap* e conseqüentemente dos programas que se relacionam com o processo é fulcral para o bom desenrolar do processo de animação. O planeamento nesta situação deve ser feito com muito cuidado para evitar problemas na limpeza das animações e confusões entre gravações. Deve, por exemplo, dar-se um nome claro e objetivo a cada gravação de acordo com o seu objetivo, manter a gravação dentro do tempo delineado previamente, fazer toda uma preparação de itens que sejam necessários para ajudar as gravações e por fim certificar que cada gravação não teve qualquer tipo de problema com uma pré-visualização rápida que o programa oferece.

A equipa de animação tem um trabalho muito delicado e demorado que requer uma grande preparação para que todo o processo avance sem contratempos. Nem sempre é possível prever todas as variáveis, mas, em geral, um bom planeamento é seguido de uma boa produção.

### 3. Desenvolvimento

#### 3.1 Referências

Antes de qualquer tarefa, existe sempre um processo de pesquisa e/ou criação de referências que um animador deve fazer. Estas referências servem para direcionar a tarefa mantendo o foco e objetivo de acordo com o material pesquisado. As referências normalmente são imagens ou vídeos que ajudam o animador no contexto da sua tarefa sendo que estas podem ser obtidas *online* ou se for possível gravadas pelo próprio animador.

Para a utilização das referências, estas podem sofrer algumas alterações passando por um processo de edição utilizando programas de edição de imagem ou vídeo. É importante que a referência seja importada para o *software* de animação que esteja a ser utilizado para que o animador possa a colocar alinhada com a sua personagem para uma verificação e comparação mais precisa. Se as referências forem vídeos pode ser pertinente transformá-las em sequências de imagens para tornar a utilização do *software* mais fluida.

Neste caso, foi feita uma gravação de um movimento que serviu de referência para uma animação *keyframe*.

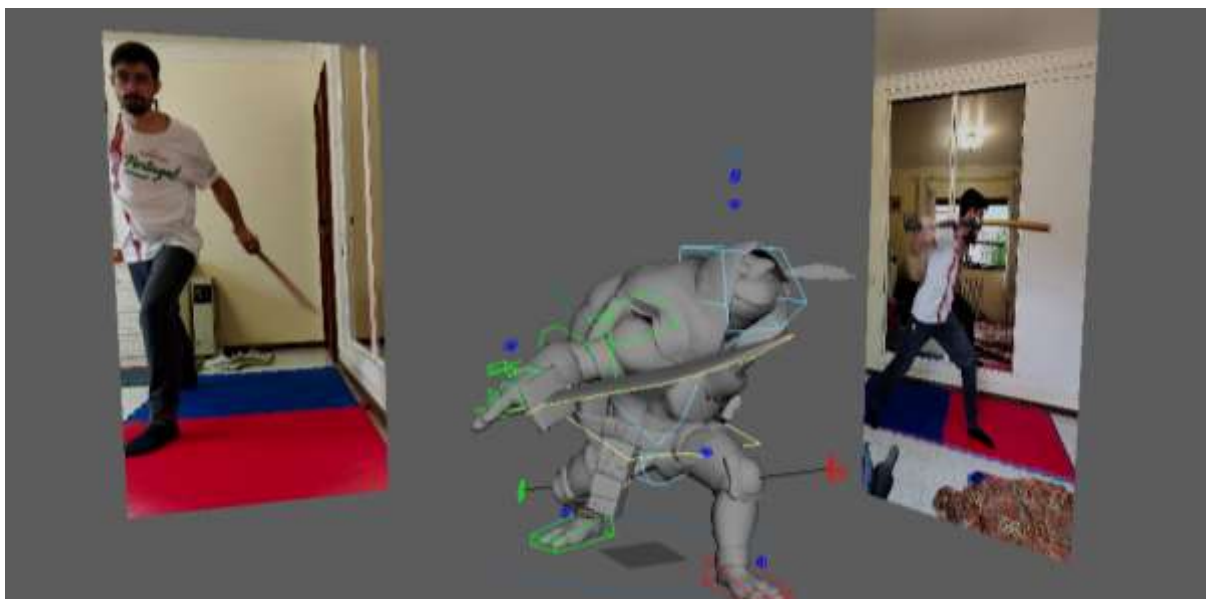


Figura 7- Exemplo de utilização de duas referências.

*O modelo e as gravações servem apenas de exemplo, não foram usadas em nenhum projeto da Saber Porto*

O movimento foi gravado de dois ângulos diferentes, permitindo um estudo do movimento mais detalhado.

Em relação a animações *mocap*, deve ser feito um estudo de referências antes dos movimentos serem gravados para que o ator possa representar o movimento exato. Também deve ser gravado um vídeo do ator quando este faz a representação dos movimentos para a gravação de *mocap*. Esse vídeo depois é usado para comparar com resultado final da animação.

### 3.2 Rigging

O processo de criação de um sistema de *rig* é desenvolvido no *software* Autodesk Maya e começa a ser feito quando a equipa de modelação disponibilizar o modelo 3D do personagem que será utilizado no jogo. O *rig* serve para definir o sistema que vai controlar o movimento do modelo disponibilizado, para isso, é criado um esqueleto com o sistema de *joints* do Maya. *Joints* são o que permitem dar movimento à personagem após a alocação das influências da *mesh* da personagem em cada *joint*.

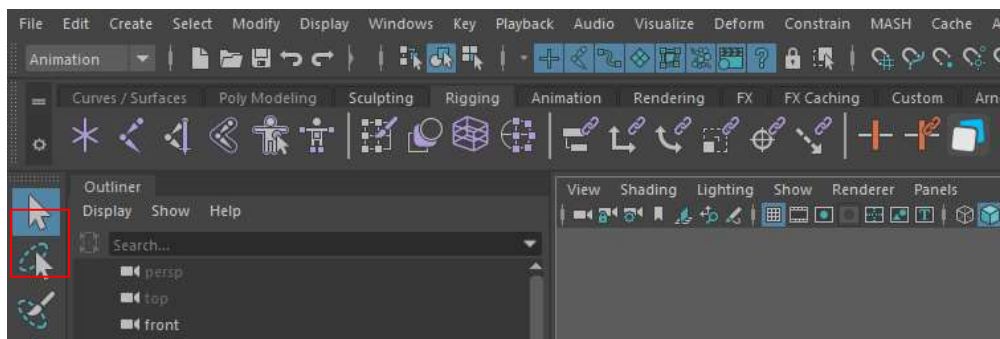


Figura 8- Ferramenta de criação de Joints no Maya

A *Root* é a *joint* base do esqueleto e deve começar sempre na *origem*, (no ponto 0,0,0 nos eixos de X, Y e Z). A partir dessa *joint* é feito o resto do esqueleto que terá influência nas várias partes da *mesh* do modelo. No final da estrutura é feito o *skinning* que é o processo de gestão de influências de cada *joint* em cada vértice da *mesh* do modelo. O Maya oferece várias formas automáticas de realizar este processo, mas geralmente é sempre necessário um ajuste do animador para que as deformações da *mesh* sejam corretas e consistentes.

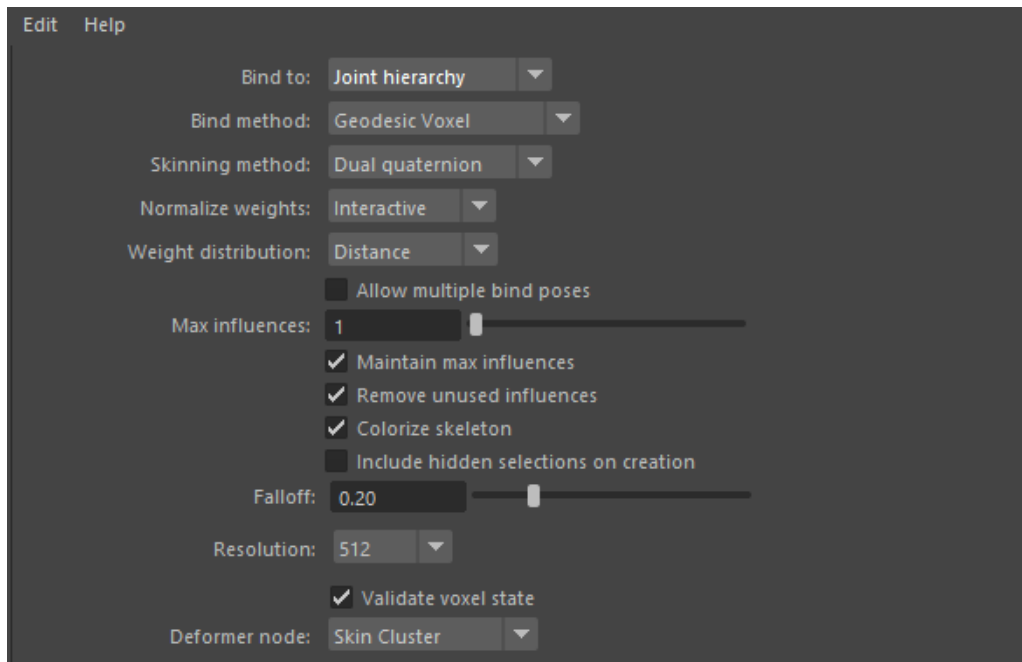


Figura 9- Opções automáticas de distribuir influências nos vértices

Existem dois conceitos importantes que são: *soft skinned* e *hard skinned*. A primeira refere-se a partes do modelo que são influenciadas por *joints* diferentes fazendo com que a sua *mesh* seja moldável, a segunda refere-se a uma parte do modelo que é apenas influenciada por uma *joint*, fazendo com que essa parte seja rígida. Para fazer a atribuição das influências o animador pode usar uma *brush tool* e pintar os pesos diretamente nos vértices.

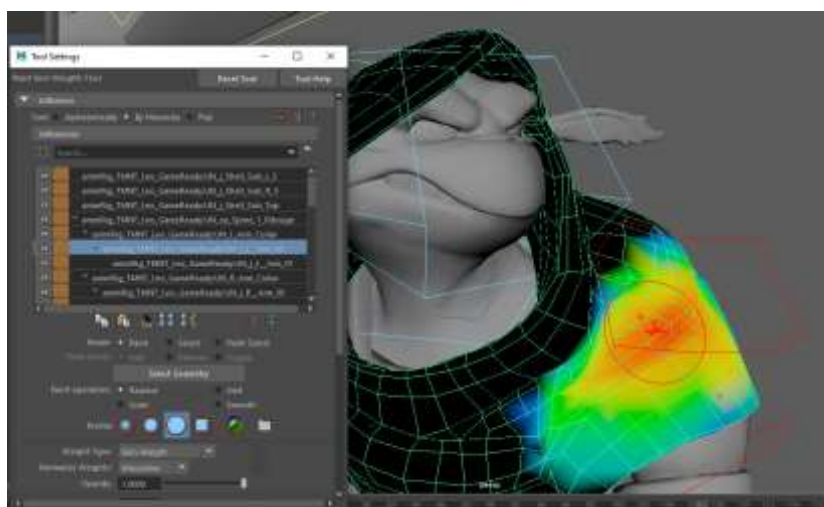


Figura 10- Representação visual da influência de uma joint do ombro de uma personagem numa mesh de roupa

O modelo só necessita do sistema de *joints* para ser implementado no *engine*, mas para completar um *rig* pronto a animar é necessário criar um sistema de manipulação do esqueleto chamados de *controllers*. Estes *controllers* podem ter várias formatos e feitios e o seu objetivo é tornar a manipulação do modelo mais acessível e intuitiva, criando também sistemas que permitem certos tipos de movimentos de *joints* conjuntos como *forward kinematics* e *inverse kinematics*.

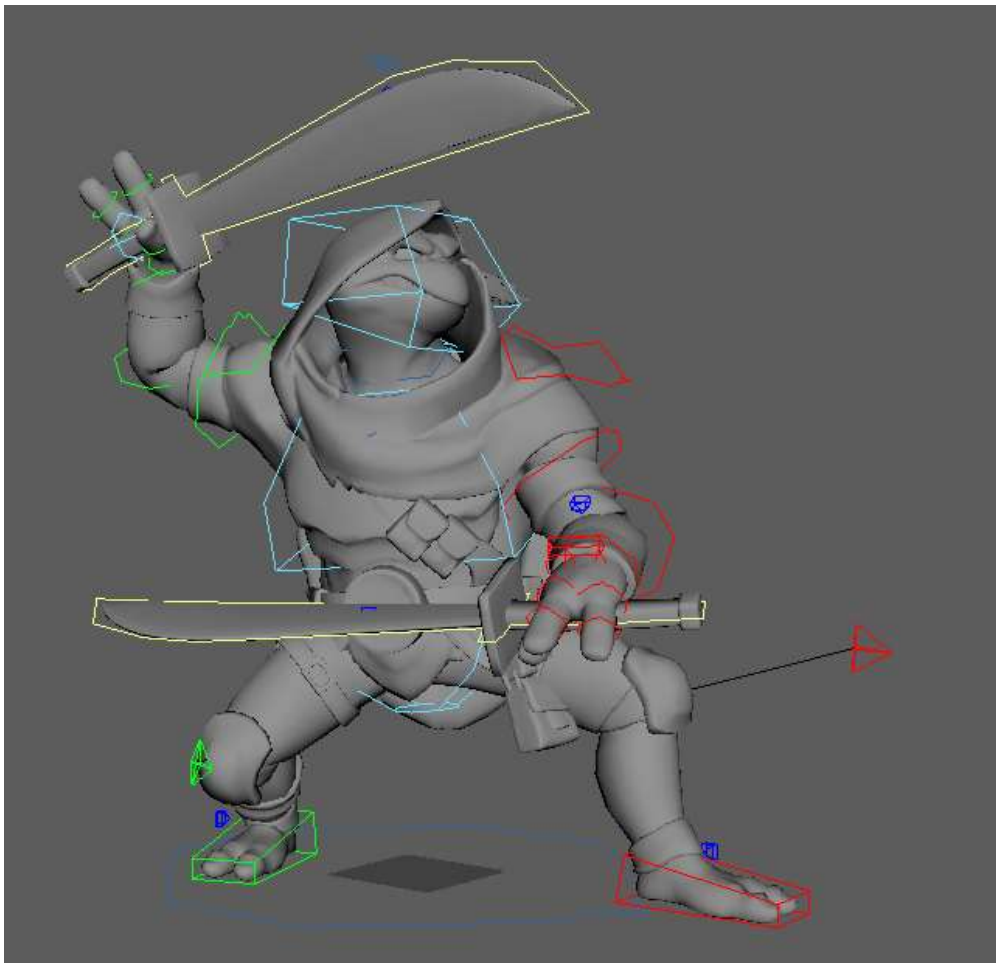


Figura 11- Exemplo de uma Rig com controladores customizados.

Os controladores têm uma cor diferente dependendo do lado da personagem em que se encontram, e formas diferentes para que se possam selecionar sem dificuldade

*Forward kinematics* refere-se à maneira de como as *joints* se comportam perante a sua rotação ou translação. “O cálculo da posição e orientação do *end-effector* em termos das variáveis da *joint* é chamado de *Forward Kinematics*” (Kucuk S, 2006). Este método é utilizado para a animar uma personagem de uma maneira mais livre, permitindo rotações e translações que *Inverse Kinematics* não permite.

*Inverse kinematics* permitem ao animador manipular a posição e orientação de várias *joints* especificando o seu ponto final o seu ponto final. “A conversão da posição e a orientação de um *end-effector* do manipulador do espaço cartesiano para o espaço da *joint* é chamado de problema de *inverse kinematics*.” (Kucuk, 2006). O sistema de *inverse kinematics* também permite bloquear uma *joint* fazendo com que, mesmo que exista movimento no resto do corpo da personagem, todo esse movimento será adaptado àquele ponto que se encontra fixo. Um exemplo muito prático deste sistema encontra-se geralmente nos pés das personagens que se encontram em pontos específicos necessitando de ter o seu movimento bloqueado para não atravessar o chão ou para não deslizar enquanto andam.

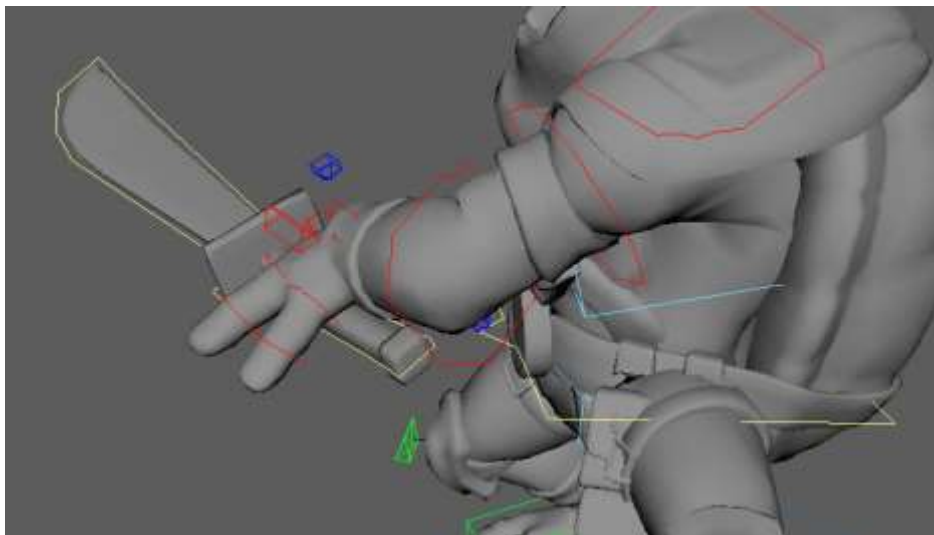


Figura 13- Manipuladores *forward kinematics* no braço permitindo a rotação independente de cada *joint*

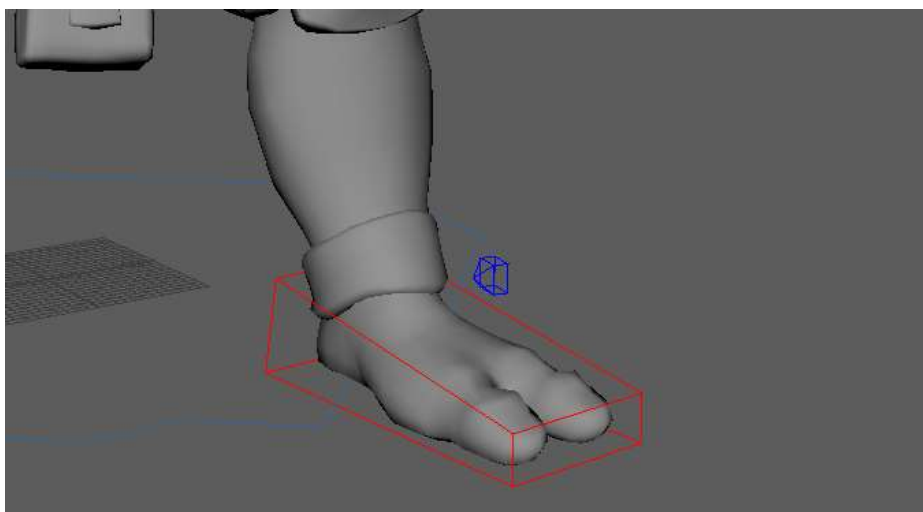


Figura 12- Manipulador *inverse kinematics* do pé permitindo a manipulação da perna apenas com um controller

### 3.3 *Motion Capture (mocap)*

Existem vários sistemas de *mocap*, aquele presente na *Saber Porto* é um sistema ótico que usa câmaras e sensores de movimento capturando a informação para ser aplicada em animações. Este processo é muito rigoroso e necessita de uma preparação prévia para que se possa tirar o máximo proveito desta tecnologia. São sempre necessárias pelo menos duas pessoas para a captação, um técnico e um ator. O ator veste o fato de *mocap* e executa os movimentos que serão gravados, o técnico precisa de estar no computador associado ao sistema para iniciar, finalizar e guardar as gravações. Os fatos de *mocap* são compostos por várias peças, nomeadamente, casaco, calças, luvas, sapatos e touca. Em todos os componentes do fato são colocados os pontos refletores que são capturados pelas câmaras.

É sempre necessário fazer calibrações antes da gravação, tanto às câmaras como ao fato. As câmaras são calibradas com um objeto que contém vários sensores fazendo movimentos circulares à sua frente. Para calibrar o fato, é necessário que o ator faça alguns movimentos que estabelecem as diferentes capacidades de amplitude e rotação do corpo. O *software* de *mocap* permite uma visualização dos movimentos num modelo 3D em tempo real facilitando a deteção de algum erro ou necessidade de ajustar algo. São feitos vários *takes* para cada uma das animações previamente definidas, e no final, são escolhidas as que melhor se adaptam à situação ou que menos erros apresentam.

Uma sessão de gravação é um processo moroso que requer sempre uma preparação prévia muito rigorosa. Não é algo que esteja sempre a acontecer, por essa razão são sempre feitas listas de capturas necessárias para gravar, e quando é decidido fazer uma sessão, são feitas o maior número de gravações possíveis.

### 3.4 Tratamento de dados *mocap*

A maior parte do trabalho durante este estágio foi desenvolvido no *software* Motion Builder da Autodesk. Este programa foi concebido para especificamente ajudar os animadores a trabalhar em animações conseguidas a partir de *mocap*. O processo começa sempre com um *setup* prévio da personagem que vai ser animada em *T-pose* e com a *rig* que vai ser usada para a implementação no *Engine*. A *T-pose* é a pose padrão que se usa em animação para personagens humanoides porque é espelhada, os braços encontram-se num ângulo de 90° do tronco de forma a facilitar o processo de *skinning*.

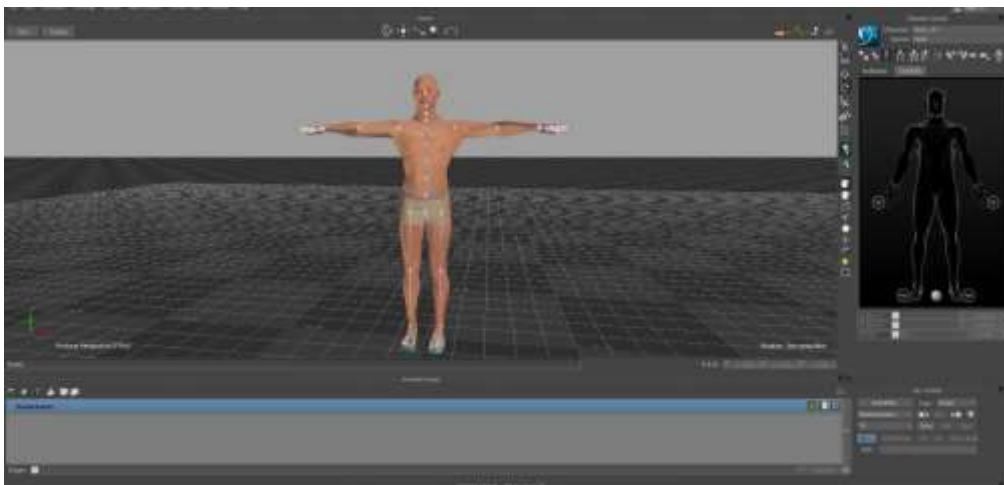


Figura 14- Área de trabalho do Motion Builder

Uma vez tendo o *setup* do modelo em *T-pose* e a animação *Mocap*, essa captura é importada para o Motion Builder para fazer *Retargeting*. *Retargeting* é o processo de transferência de informação entre *rigs* diferentes. Neste caso a informação da captura de *Motion capture* precisa de ser transferida para o *rig* criado para a personagem para que esta tenha a animação pretendida.

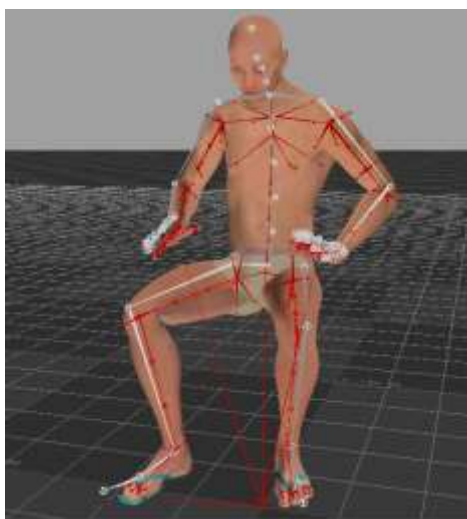


Figura 15- Diferentes sistemas de joints para efeitos de Retargeting

Uma vez feita a transferência de movimento do ator para a personagem, para que a informação possa ser editada e limpa, é necessário criar um novo *rig* chamado de *Control Rig* através das ferramentas do programa. Este *rig* é temporário e apenas serve para adicionar ou substituir informação já existente na animação. Possíveis diferenças entre o *rig* proveniente do *mocap* com o *rig* da personagem podem ter alguns resultados inesperados na transição de movimento de um para o outro.

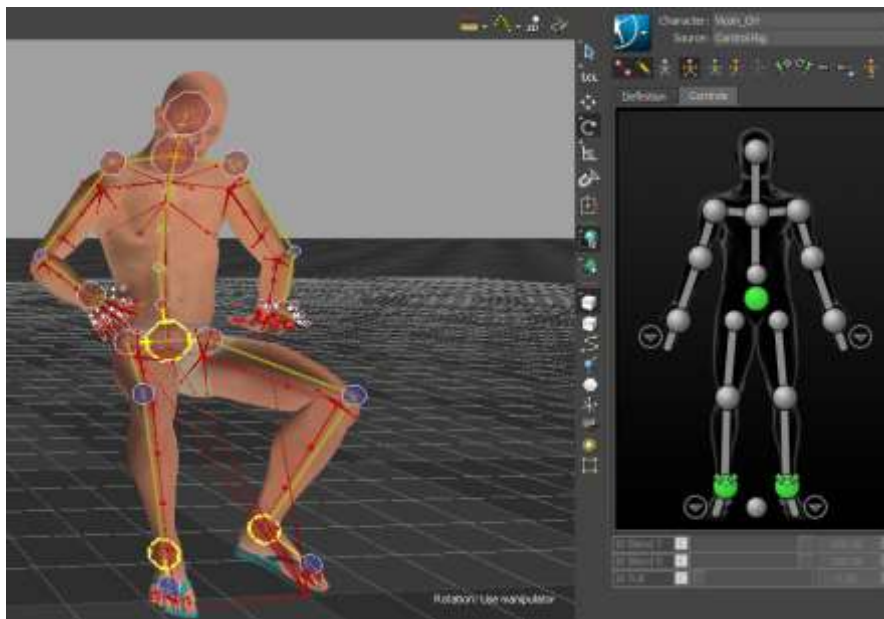


Figura 16- Criação de uma control rig no Motion Builder

No caso do ator que gravou com o fato *mocap* ter uma altura diferente da personagem do jogo, quando o *retargeting* for feito, os movimentos poderão ter inconsistências devido à diferença de alturas. Por isso, no processo de gravação, já tem de se ter noção de que se devem contratar atores que correspondam o melhor possível à personagem que eles vão representar.

*Storymode* é uma funcionalidade do Motion Builder que permite editar os *clips* de animação como se fosse um editor de vídeo. Isto permite definir muito precisamente o início e o fim da animação como também criar um *loop* muito facilmente com técnicas como *ease in* e *ease out*. Ao intercalar dois *clips* de animação ele cria automaticamente uma interpolação linear de transição de um para o outro sendo possível alterar para uma interpolação *ease in* e *ease out*, criando uma transição mais suave entre animações. Esta funcionalidade é muito utilizada na criação de um *loop* dentro da animação, querendo dar a impressão de que ela se repete indefinidamente. Se o ator começou e terminou na mesma posição, ou em posições muito parecidas, é possível então criar o *loop*.

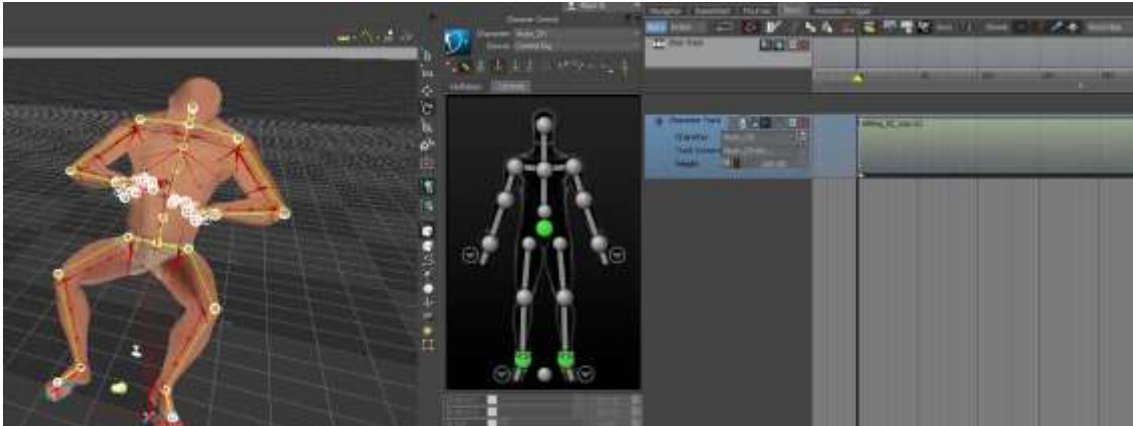


Figura 17- Storymode no Motion builder

O processo de *plotting/baking* permite gravar a animação após alterações ou edições feitas tornando-se irreversíveis. Este processo ajuda a edição em certas situações, mas tendo sempre o cuidado de gravar o projeto regularmente para que haja sempre um ponto de retorno, mesmo pós *plotting/baking*.

A edição é a parte do trabalho onde os conhecimentos de animação são aplicados, apesar de grande parte do trabalho ter sido feito pelo *mocap*, vai sempre haver necessidade de corrigir informação ou de a ajustar para algum objetivo em específico. Uma janela muito importante para a edição são as curvas que representam toda a informação de cada *joint* numa *timeline* (representação de informação ao longo do tempo). Principalmente trabalha-se com a translação e rotação de cada *joint* sendo que existe uma linha para cada eixo, x, y e z, sendo possível a alteração de informação de uma maneira muito precisa e minuciosa.

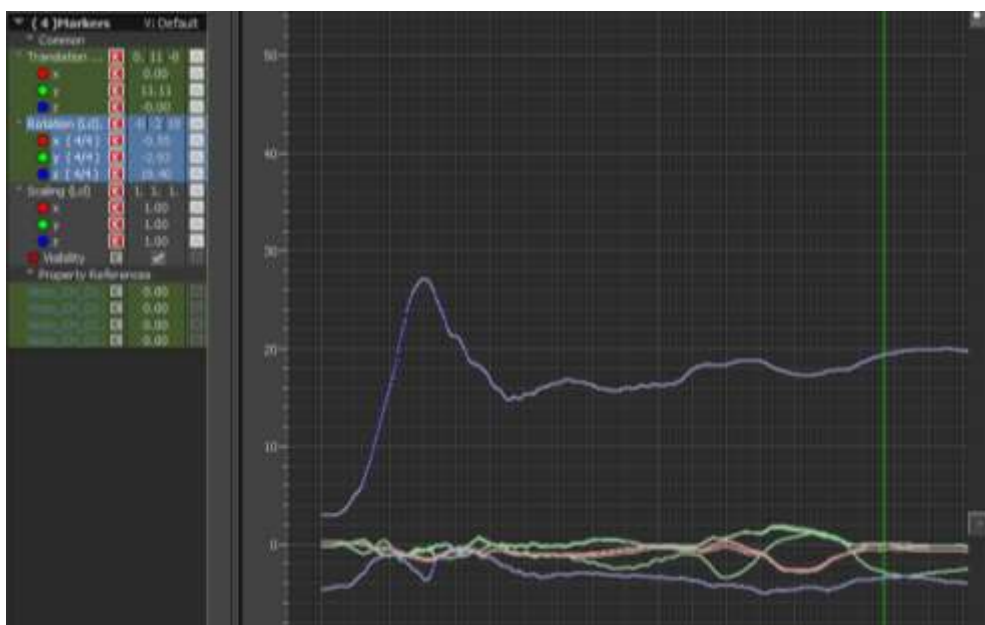


Figura 18- Editor de curvas de animação

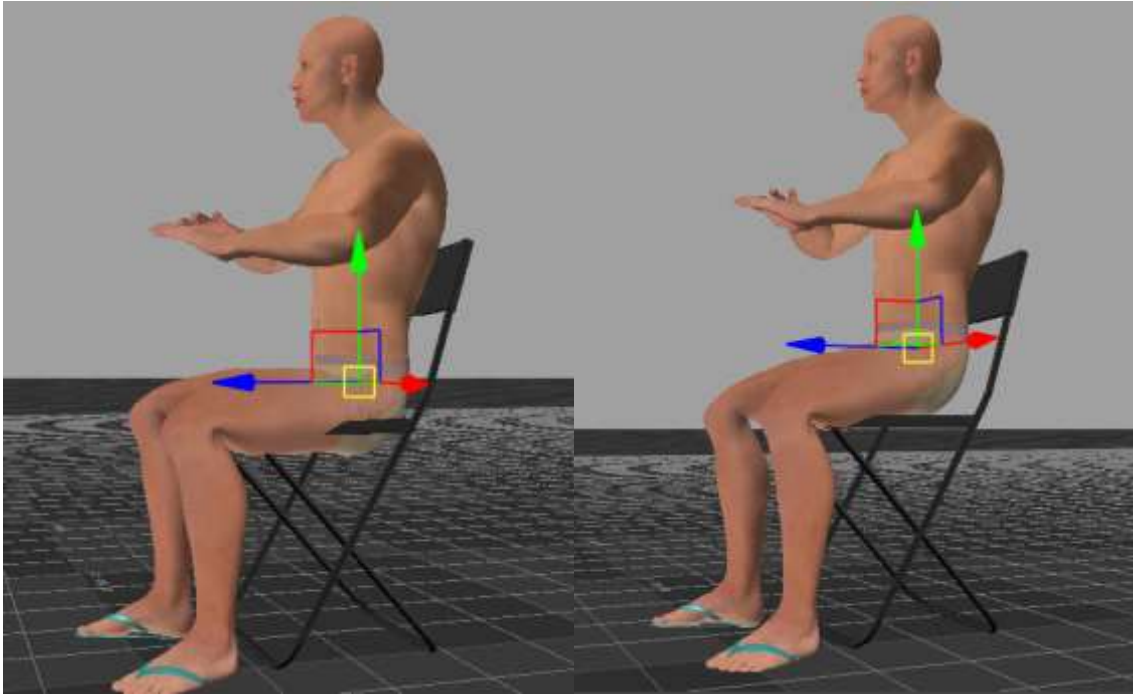


Figura 19- Exemplo de uma correção usando a translação do controlador do pélvis

No Motion Builder também estão presentes os sistemas de *forward kinematics* e *inverse kinematics*. Sabemos quais dos sistemas estão ativos se os manipuladores (representados pelos círculos) estiverem verdes ou cinzentos. Se estiverem verdes (no caso dos pés) quer dizer que o sistema de *inverse kinematics* está ligado e que a informação que aparece na janela das curvas é referente ao movimento dos manípulos das *inverse kinematics*. É possível ativar apenas a translação ou apenas a rotação ficando visualmente com metade do círculo verde. Se estiverem a cinzento quer dizer que a informação estará guardada nos manípulos retangulares que estão entre os círculos. Esses manípulos funcionam com *forward kinematics*.

É possível bloquear cada controlador de *inverse kinematics* através do ícone de um *pin* que se encontra acima da representação da *control rig*, permitindo um ajuste de alguma parte do corpo sem mexer aquela que estiver bloqueada.

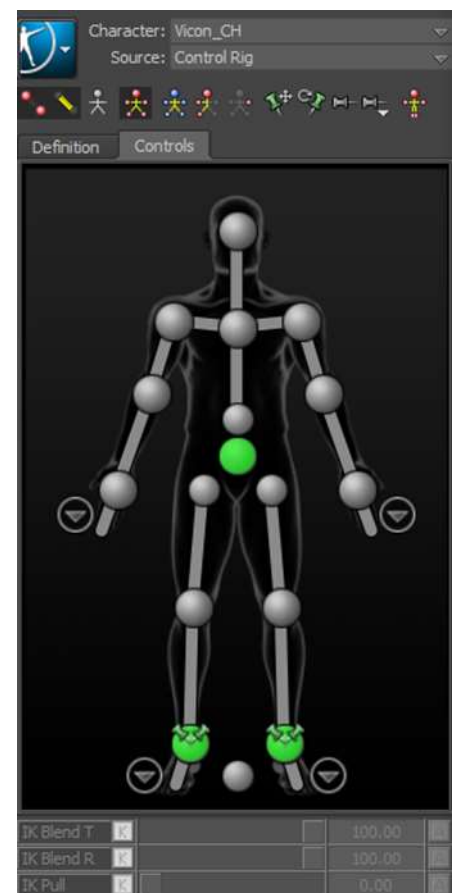


Figura 20- Control rig do Motion Builder

O Motion Builder oferece alguns filtros padrão para ajudar com a edição. Alguns deles podem ser aplicados em toda a animação para diminuir *jittering* (mudanças erráticas nos valores da informação da animação) e *sliding* (deslizes indesejados em partes como os pés ou mãos), como o filtro de *butterworth* e *smooth* que tornam as curvas de informação mais consistentes e suaves.

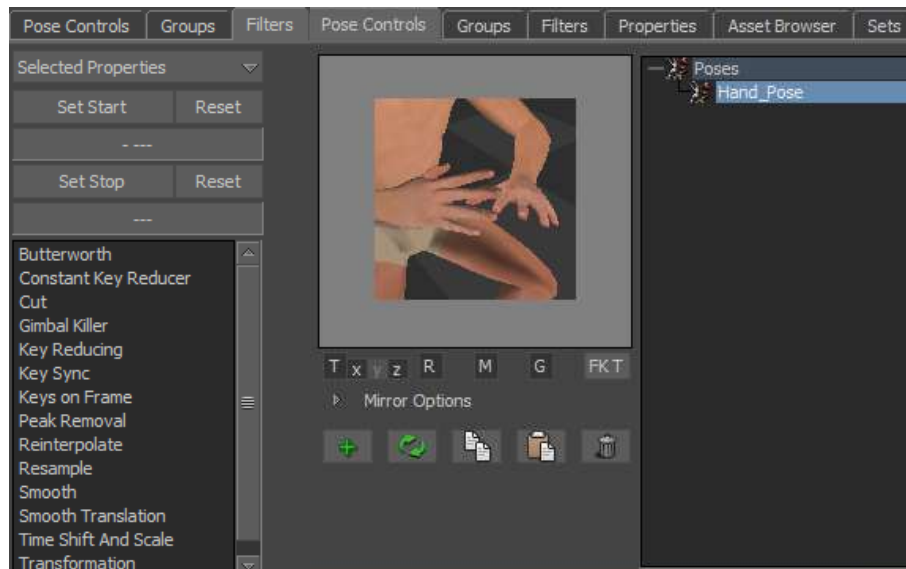


Figura 21- Filtros proporcionados pelo Motion Builder e Biblioteca de poses

À medida que se vai animando podem-se guardar poses para serem reutilizadas futuramente, sendo possível criar uma biblioteca de poses transferível de projeto para projeto ganhando assim algum tempo. Isto é especialmente útil para poses de dedos que são sempre das partes mais morosas do processo.

Para a edição completa é necessário realizar o *bake* final, mas desta vez diretamente para a *rig* da personagem que se quer implementar no jogo, para que a informação da animação esteja lá gravada. Para exportar a animação é selecionado o esqueleto pretendido e escolhida a opção de *Motion File Export* criando um ficheiro de fbx apenas com a informação da animação no esqueleto de implementação.

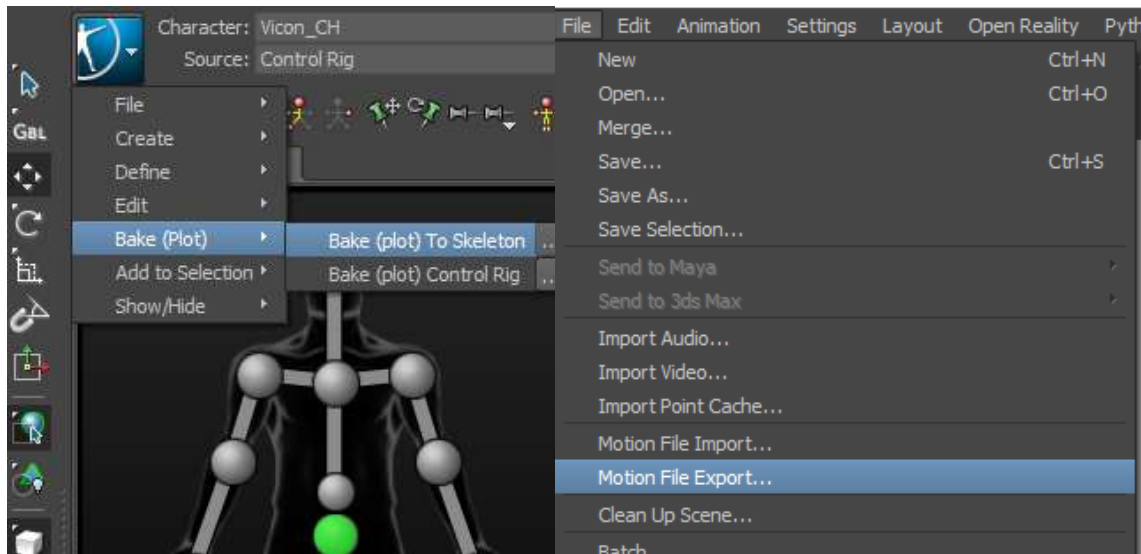


Figura 22- Opções de Bake e Export

### 3.5 Animação *Keyframe*

Algumas das tarefas realizadas durante o estágio envolviam animação *keyframe*. Estas animações foram feitas no Maya e no Motion Builder. Ambos programas permitem a criação destas animações apesar que o Maya apresenta ferramentas mais diversas e acessíveis, enquanto o Motion Builder tem uma estrutura mais limitada. No entanto os conceitos de animação são transversais a qualquer *software* e através destas tarefas conseguiu-se avaliar o nível de conhecimento e técnica por parte do estúdio permitindo uma evolução e reconhecimento do seu trabalho.

Na animação por *keyframes* é preciso aplicar conceitos de animação como *timing*, aceleração e desaceleração, *squash and stretch*, antecipação e *follow through and overlapping action*. Estes princípios, descritos pelos animadores da Disney Frank Thomas e Ollie Johnson (Johnson & Thomas, 1981) são intemporais e também se aplicam à animação 3D realista.

*Timing* refere-se ao número de desenhos que um animador teria de desenhar para que essa animação tivesse mais ou menos tempo. Isto significa que mais desenhos dariam uma animação mais lenta, e menos desenhos, uma animação mais rápida. Este conceito no 3D converte-se diretamente no número de *frames* que é deixado entre cada *key*. Por exemplo, se um movimento começar com uma *keyframe* no zero e acabar com um *keyframe* no cem, será bem mais lento do que um movimento que comece no zero e acabe

com um *keyframe* no cinco. A interpolação entre *keyframes* é feita automaticamente tendo essa vantagem sobre o 2D tradicional.

A aceleração e desaceleração, tal como o *timing*, refere-se ao número de desenhos que são precisos entre poses extremas. Para obter uma aceleração o número de desenhos tem de ir diminuído à medida que nos afastamos da pose original. Para uma desaceleração o processo inverte, tem de haver um aumento do número de desenhos à medida que nos aproximamos da pose final. No 3D existem várias formas de obter este resultado, a primeira será imitar um pouco a forma do 2D e criar mais ou menos *keyframes* junto das iniciais. A segunda maneira seria manipular as curvas do editor de curvas para criar um *ease in* ou um *ease out* sem ter de criar *keyframes* adicionais.

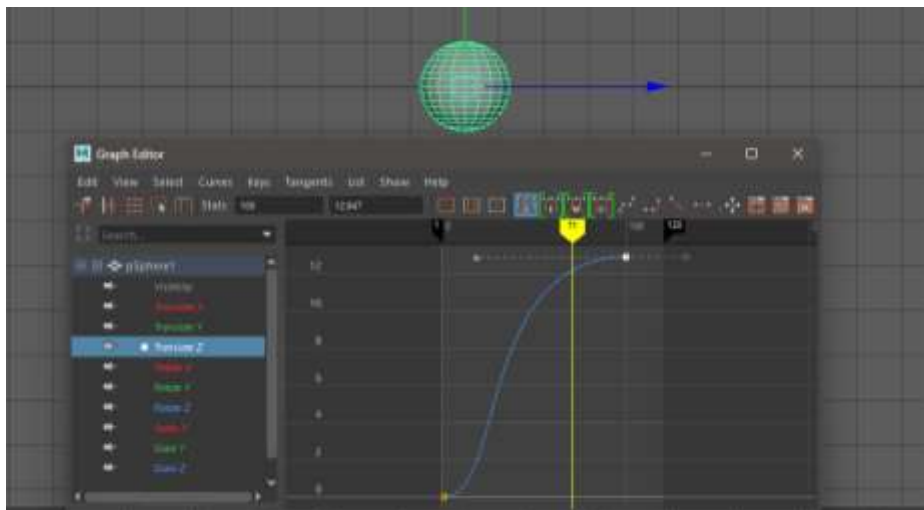


Figura 23- Exemplo de uma curva com *ease out* acentuado

O *squash and stretch* remete para que o animador tenha de pensar na consistência dos objetos e personagens que está a animar, como esses corpos se comportam em situações de aceleração e desaceleração. Se mudam de forma, como uma bola saltitona, ou se se mantêm constantes, como uma bola de *bowling*. Em 3D as ferramentas de escala e translação permitem fazer este tipo de deformações, apesar de que em animações *mocap* não é algo que venha por definição, e teria de ser acrescentado posteriormente se fosse esse o objetivo. À medida que a animação se aproxima de um estilo mais realista, mais se perde a noção do *squash and stretch*, apesar de nunca verdadeiramente desaparecer.

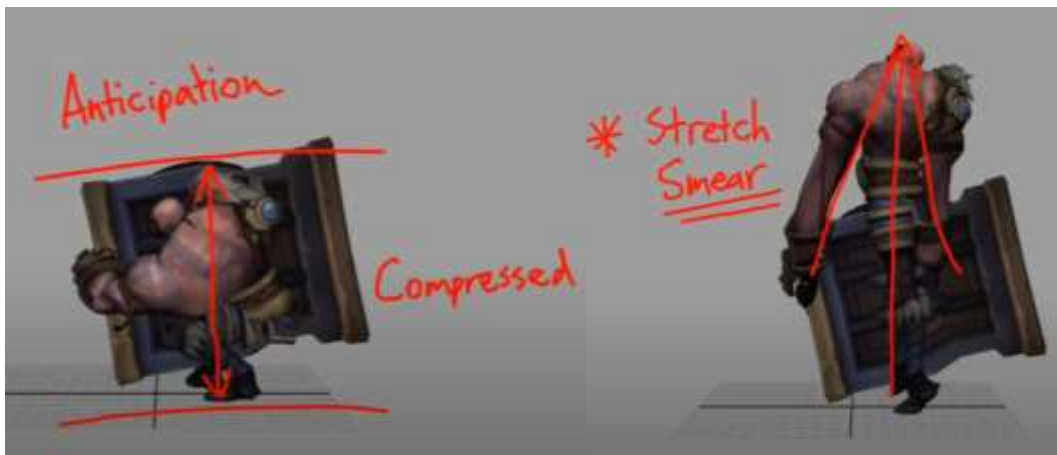


Figura 24- Exemplo de squash and stretch numa personagem 3D

A antecipação refere-se ao movimento antes da ação, de que o público precisaria de uma pista do que será a ação antes dela acontecer para lhe dar ênfase. Trata-se de um efeito muito prático aplicado a videojogos, principalmente jogos que necessitem de reações rápidas por parte do jogador. Por exemplo, num jogo de luta, uma personagem antes de atacar tem sempre uma antecipação indicando ao oponente que tipo de ataque vai fazer, dando a oportunidade desse ataque ser defendido. As animações neste caso servem como indicadores de que o jogador precisa de estar atento para poder jogar o jogo ao mais alto nível.



Figura 25- Spritesheet de uma animação do StreetFighter

*Follow through and overlapping action* é a continuidade do movimento de um corpo que vai para além da ação principal. Pensando em cabelo comprido, que vai acompanhando o movimento da cabeça até esta parar ou mudar de direção, nesse caso o cabelo vai continuar o seu movimento na direção inicial durante mais algum tempo até voltar a sincronizar com a cabeça. Este conceito aplica-se a muitas partes do corpo humano e a apêndices ou roupa que as personagens possam usar. Em 3D é possível criar um sistema de físicas para simular o movimento destas peças adicionais em vez de as ter de animar. Este processo pode tornar os ficheiros mais pesados e perder um pouco em *performance*, mas ganha-se muito em termos de poupança de tempo e movimento realista.



Figura 26- Dois exemplos de simulação de físicas na roupa de uma personagem

Para animar no Maya é preciso conhecer e entender algumas ferramentas importantes.

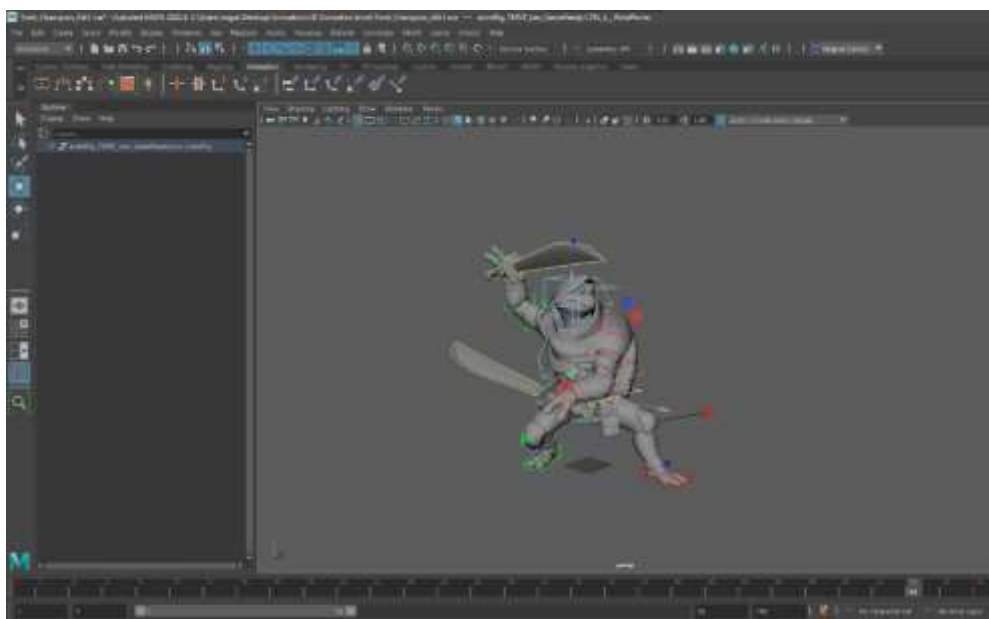


Figura 27- Interface do Maya

De lado, encontramos as ferramentas de seleção, às quais também se pode aceder através de atalhos específicos do *software*. Se clicarmos no W ativamos a translação, no E a rotação e no R, a escala. Estas três ferramentas são essenciais para interagir com o *rig* usando a deformação desejada.

Na figura seguinte pode ver-se a *timeline*. Na *timeline* estão as *keys* a vermelho e é aqui que controlamos o *timing* e duração da animação.

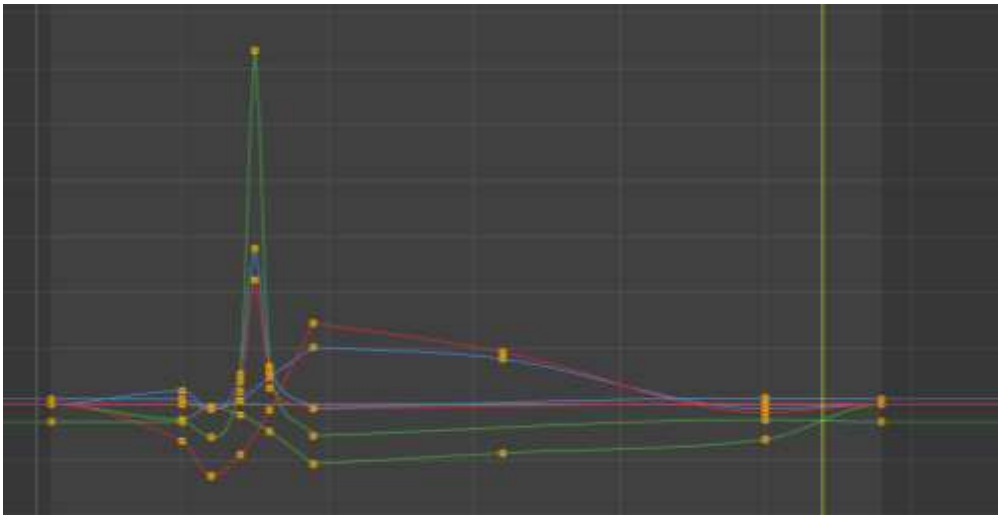


Figura 28- Graph Editor do Maya

O *graph editor* do Maya corresponde ao editor de curvas no Motion Builder. Aqui é representada a informação da animação em formato de um gráfico, onde cada eixo é representado por uma linha de cor com a sua variação consoante o tempo. Cada ponto amarelo é um *keyframe*. É possível editar a animação a partir deste editor mexendo nos *keyframes* ou mudando as curvaturas das linhas.

Uma ferramenta externa ao Maya que pode ser instalada de uma forma gratuita é o Tweener. O Tweener permite criar *keyframes* com base nos *keyframes* já existentes. Basta estar na *timeline* entre duas *keyframes* e carregar num dos botões que têm círculos. Eles representam a percentagem da proximidade que o *keyframe* vai ser criado vai ter em relação aos *keyframes* da sua esquerda e direita. Esta ferramenta ajuda-nos a aplicar os princípios de aceleração e desaceleração.

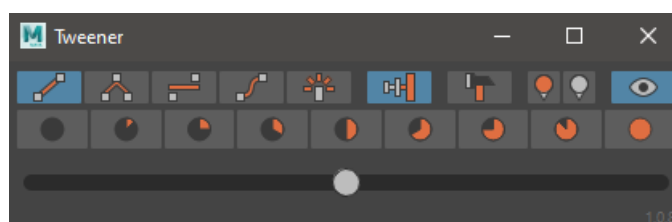


Figura 29- Tweener plug-in para o Maya

### 3.6 Conclusão do Capítulo

As tarefas propostas no âmbito do estágio foram cumpridas e com bons resultados. A aprendizagem e aprofundamento dos programas Maya e Motion Builder trouxeram novas perspetivas e novas oportunidades de desenvolver animações de uma maneira mais profissional e eficiente.

Todo o processo de *mocap* é delicado e requer toda uma preparação prévia para que realmente seja proveitoso o resultado das gravações. Apesar de ser tecnologia que agiliza muitos processos, também leva o seu tempo tanto para preparar, executar e até mesmo interpretar a informação obtida no final.

Os princípios da Animação desenvolvidos pelos estúdios Disney continuam presentes seja em que meio for: videojogos, animação 3D realista, conseguem-se sempre aplicar. O *software* veio simplificar muitos destes princípios, mas é fácil esquecer que eles precisam de ser aplicados.

A criação de *rigs* dá uma perspetiva nova no que toca a criar as ferramentas e maneiras necessárias para que as personagens possam ser animadas de uma forma livre e intuitiva.

#### 4. Ilações Finais e Trabalho Futuro

Durante os seis meses de estágio houve uma demonstração de competências já adquiridas nos anos de estudo que proporcionaram uma adaptação rápida aos processos de desenvolvimento do estúdio. No entanto, o conteúdo desenvolvido durante o estágio foi de acordo com aquilo que exatamente era expectável, levando o trabalho pessoal e profissional a um novo nível, evoluindo de uma maneira diferente comparativamente ao estudo desenvolvido no contexto académico. O nível de exigência foi aumentando à medida que o estágio se aproximava do fim, pedindo um alto nível de organização e concentração sendo que o objetivo do estúdio era a contratação após a finalização do estágio.

No geral os objetivos foram cumpridos, dependendo da tarefa, com mais ou menos facilidade. Todas as tarefas foram instrutivas e importantes, mas um ponto mais negativo foi a diversidade ou a repetição de uma tarefa em específico como *rigging* e a falta de mais tarefas como as de animação.

O estágio serviu como uma boa preparação para a vida profissional na área dos videojogos, abordando todas as fases técnicas de criação e implementação de animações em projetos profissionais criando um sentido de responsabilidade mesmo estando apenas num estágio.

A contratação do aluno após estágio foi concretizada. Presentemente este continua a desenvolver animações para videojogos na *Saber Porto*.

## 5. Referências e Bibliografia

All3DP. (15 de 9 de 2021). *The Best 3D Animation Software of 2021 (Some are Free)*. Obtido de all3dp.com: <https://all3dp.com/1/best-20-3d-animation-software/#:~:text=3ds%20Max%20is%20one%20of,effects%20production%2C%20and%20architectural%20visualization>.

Alderton, R. (22 de 09 de 2023). *League Animation Deepdive I*. Obtido de Youtube: <https://www.youtube.com/watch?v=WnKUI7sRaPE>

Burtnyk, N., & Wein, M. (1971). *Computer Generated Keyframe Animation*. Journal of the SMPTE, vol 80, no. 3, 149-153. doi: 10.5594/J07698.

Burtnyk, N., & Wein, M. (1976). *Interactive Skeleton Techniques for Enhanced Motion Dynamics in Key Frame Animation*. Communications of the ACM 19 (10), 564-669.

Catmull, E. (1972). *A system for computer generated movies*. Proceedings of the ACM annual conference - Volume 1 (ACM '72). Association for Computing Machinery, New York, NY, USA, 422–431. DOI:.

Catmull, E. (1978). *The problems of computer-assisted animation*. Proceedings of the 5th annual conference on Computer graphics and interactive techniques (SIGGRAPH '78). Association for Computing Machinery, New York, NY, USA, 348–353. DOI: [org/10.1145/800248.807414](https://doi.org/10.1145/800248.807414)

David J. Sturman (1994). *A Brief History of Motion Capture for Computer Character Animation*. [https://www6.uniovi.es/hypgraph/animation/character\\_animation/motion\\_capture/history1.htm](https://www6.uniovi.es/hypgraph/animation/character_animation/motion_capture/history1.htm)

Guerrilla Games. (s.d.). *Home*. Obtido de Website de Guerrilla Games: <https://www.guerrilla-games.com/>

Inside Gaming. (27 de 05 de 2021). Obtido em maio de 2023, *How Motion Capture Works in Video Games*. Obtido de Youtube: <https://www.youtube.com/watch?v=aAGoBziTBDY>

Kitagawa, M. Windsor, B. (2008). *MoCap for Artists: Workflow and Techniques for Motion Capture*. Elsevier Inc.

Kucuk, S., & Bingul, Z. (2006). *Robot kinematics: Forward and inverse kinematics* (pp. 117-148). London, UK: INTECH Open Access Publisher.

Mori, M. (1970). *The Uncanny Valley*.

Newman, H. (10 de 12 de 2020). *Motion Capture vs Handkey Animation - Maya Power Tips EP8*. Obtido de Youtube:  
<https://www.youtube.com/watch?v=7gZt3RCmgfM>

Rhaul M (2018). *Review on Motion Capture Technology*. <https://computerresearch.org/index.php/computer/article/view/1851/1835>

Rose, I. R. (21 de 09 de 2019). *The Longevity of 2D Animation in Video Games*. Obtido de Youtube: <https://www.youtube.com/watch?v=Du2uFv8iuvo>

Saber Interactive. (13 de 06 de 2022). Obtido em setembro de 2023, *Saber Interactive: Home*. Obtido de Site da Saber Interactive: [saber.games](http://saber.games)

Saber Interactive. Obtido em setembro de 2023, Saber Interactive Home page. Obtido de linkedin: <https://www.linkedin.com/company/saberinteractive/>

Santa Monica Studio. *Home*. Obtido em setembro de 2023 Obtido de Web site de Santa Monica Studio: <https://sms.playstation.com/>

Sito, T. (2013). *Moving Innovation: a History of Computer Animation, 2013*. The MIT Press.

Sturman, D. (1994). *A Brief History of Motion Capture for Computer Character Animation*. [Conference Presentation]. SIGGRAPH 1994.

SuperMacBrother. (10 de 19 de 2018). *Star Citizen | Planet Weather, Real Time Cloth & Clothing Simulation*. Obtido de Youtube:  
<https://www.youtube.com/watch?v=BHzGJtMAOB4>

Thomas, F. & Johnston, O. (1981). *The Illusion of Life*. Disney Productions