

## From Performative Animation to Film Production Da Animação Performativa à Produção Cinematográfica

Marcello Ferreira

Escola Superior de Media Artes e Design – Instituto Politécnico do Porto, Portugal

Bárbara Cleto

Escola Superior de Media Artes e Design – Instituto Politécnico do Porto, Portugal

Luís Leite

Escola Superior de Media Artes e Design – Instituto Politécnico do Porto, Portugal

### Abstract

*A few years ago, motion capture was a technology used exclusively in high-cost film productions. With technological developments, this type of resource has become more accessible to the general public, and so the need arose to look for appropriate methods that serve as an impetus for the development of 3D animations in real time, either through open source software, through equipment that is more accessible to the general public. The importance here lies in reducing costs and discarding any dependency on rendering in 3D animation.*

*This article aims to expose some areas of study associated with performance animation and film production. As a case study, this article presents the development of a prototype for a multimedia installation within the scope of the short-film Animata. With this prototype, it's intended to explore the limits of the animated character in real time, through the integration between different devices and software, connected in a multimedia ecosystem.*

**Keywords:** Motion capture, Performative Animation, Multimedia Installation, Real time Interaction, Virtual Production.

### Introdução

*Animata* é uma curta-metragem de animação que utiliza a tecnologia produzida em motion capture, desenvolvida no âmbito do projeto final da licenciatura em Multimédia da ESMAD. Este artigo tem como objetivo explorar a relação entre as tecnologias multimédia e o cinema de animação.

A curta-metragem é destinada a um público jovem-adulto e tem uma duração de aproximadamente onze minutos. A narrativa aborda a relação entre o cientista Alexandre e sua filha (Inês), após a morte de sua esposa (Sofia). Obcecado com o passado, Alexandre tenta recuperar a sua falecida mulher construindo um robô à sua imagem, contudo, esta experiência terá graves consequências, nomeadamente a perda das memórias que Alexandre tinha de Sofia.

Este projeto reflete não só a grande paixão do autor pela animação, pela ilustração e pelo cinema; mas também o culminar de estudos, investigações e experiências desenvolvidas durante a licenciatura em Multimédia. Para além do desenvolvimento conceptual e visual, houve um foco maior no desenvolvimento de

soluções técnicas em diferentes fases da produção. Apesar de ser um projeto ambicioso, revelou-se não ser impossível.

Para a produção da curta-metragem, utilizaram-se técnicas de captura de movimento aplicadas num contexto experimental, combinando técnicas de animação 3D e 2D. No contexto da animação 3D, a animação foi produzida através de sistemas de captura inerciais e óticos aplicados a personagens modelados em 3D, num ambiente virtual tridimensional.

Neste artigo pretende-se demonstrar as potencialidades de alguns dos dispositivos de captura que foram explorados e testados durante a investigação do projeto. A partir deste estudo, do material produzido e dos recursos disponíveis, surgiu a inspiração para recriar uma das cenas do filme através de uma instalação multimédia que pudesse ser interativa. Ao contrário do filme, em que a captura de movimento foi processado de forma faseada, em pós-produção; nesta instalação pretendeu-se explorar formas de interação transmediáticas entre a história e o público, através da animação performativa em tempo real.

### Estado da Arte

#### *O Cinema, a Animação e a Performance*

A maior inspiração para o filme foi o próprio termo “animação”. Tal como Alexandre, o propósito do autor era encontrar a qualidade dos seres animados, as suas “almas”. Mais do que um conceito usado para designar imagens em movimento, o termo “animação”, tem origem na palavra “*animare*”, significa “dar vida a”. A partir desta definição é possível ir além dos domínios da animação e refletir sobre o significado filosófico da vida e do movimento, um movimento que gera vida, uma indicação de um sujeito que respira, a presença de uma alma e da ilusão da vida.

Perante uma eminente rutura entre animação e cinema encontram-se diversas relações na imersão nos espaços virtuais, na perceção do movimento e na necessidade de contar uma história. O mesmo é aplicado nas artes performativas, mas com uma sensação diferente de continuidade e de presença.

Tal como afirma Paulo Simões Rodrigues <sup>1</sup>, “a experiência da arte performativa não pode ser preservada, não só devido a sua duração, mas por causa das suas mudanças culturais. Resta então a perpetuação da memória e da identidade, pois a preservação passa pela prática e pela transformação daquilo que chamamos de arte.”

Ou seja, novas formas de criação irão levar-nos a novos públicos e novos interesses. A intangibilidade e a fragilidade do ato performativo são características essenciais da experiência e tornam este tipo de acontecimentos único para quem os realiza e para quem os usufrui. Por outras palavras, cada performance é uma história única.

### *Animação Expandida*

Segundo Marcel Duchamp <sup>2</sup>, “o ato de criação de um artista não é tanto a invenção de novos objetos, mas a revelação de relações desconhecidas entre eles.” Se refletirmos sobre a história da arte, podemos deduzir que o homem tornou-se capaz de reproduzir o real ou alterá-lo, mas foi através do cinema que foi possível reinventá-lo. E se a realidade é cada vez mais o que fazemos dela, é natural surgirem novas relações entre o público e o cinema, e consequentemente, a animação.

Ainda não existe um termo concreto para descrever estas novas formas de interação, mas podemos referir alguns conceitos como “*Live-Cinema*” ou “*Augmented Storytelling*” <sup>3</sup> – apresentados por Luís Leite na tese *Virtual Marionette – Interaction Model for Digital Puppetry* - para designar novas expressões baseadas na animação em tempo real e o termo “*Interactor*” para designar o público que passa a ser um ator interativo na cena. Tal como em outras áreas, como na fotografia ou no cinema, também podemos invocar aqui o termo “*Animação Expandida*” <sup>4</sup> - introduzido no livro *Experimental and Expanded Animation* (2018) - de Vicky Smith e Nicky Hamlyn, que parece ser o mais apropriado para descrever o cinema de animação como uma experiência imersiva, que ultrapassa os limites do cinema e rejeita a relação tradicional entre o público e o ecrã.

Com a evolução dos meios digitais e com as novas necessidades da realidade atual no contexto da pandemia, nos deparamos com novas formas de “assistir” e de “fazer” animação. Devido ao seu caráter multidisciplinar, é possível encontrar diversas relações e diferentes desafios na construção da forma, que procura o seu meio e a sua justificação, num processo de expansão digital.



Figura 1 – As Aventuras de Tintim – O Segredo de Licorne, de Steven Spielberg (2011)

### *Animação Performativa*

Quando se aborda a temática da animação performativa, é importante entender o que foi feito no passado, o que está a ser feito no presente e como isso irá influenciar o futuro. Ao invocar a vanguarda da arte digital e das origens do motion capture, é impossível não referir Myron Krueger, um pioneiro na

### *Captura de Movimento/ Motion Capture*

O Motion capture, ou captura de movimento, consiste num sistema tecnológico que possibilita registar, em suporte digital e através de recursos computacionais, movimentos provenientes de um ator. Consequentemente, confere um meio técnico ao animador para reproduzir os comportamentos da realidade, economizando em simultâneo, o tempo de produção e recursos humanos de uma obra fílmica animada ou parcialmente animada.

O desenvolvimento de personagens em tempo real, possibilita o surgimento de uma nova abordagem: a interatividade na arte de animar. O tempo real enfatiza a criação contínua de movimento de maneira espontânea e natural e esta metodologia permite não só a otimização dos processos de produção, mas também abre um vasto leque de novas aplicações e novos contextos para a animação.

Este ecossistema de interação tornou-se fundamental nas artes digitais para romper barreiras e flexibilizar o trabalho criativo, permitindo aos artistas mapear os seus artefactos e objetos virtuais, independentemente de especificações de software ou hardware.

Atualmente, podemos ver o quanto estes sistemas evoluíram e são agora aplicados em diferentes contextos e de diferentes formas. Contudo, apesar da introdução destas técnicas digitais na produção cinematográfica revolucionar o universo das indústrias criativas, curiosamente o mesmo não aconteceu com o cinema de animação, apesar do sucesso de alguns filmes como *A Casa Fantasma* (2006), *Um Conto de Natal* (2009) ou *As Aventuras de Tintim – O Segredo de Licorne* (2011).

Entre estes, a referência mais relevante foi o filme *As Aventuras de Tintim – O Segredo de Licorne*, dirigido por Steven Spielberg (2011), que utilizou a mesma tecnologia do filme *Avatar* (2009). Enquanto *Avatar* serviu para consolidar a tecnologia na indústria do cinema, *As Aventuras de Tintim – O Segredo de Licorne* (figura 1) consegue equilibrar o que é real e estilizado, tanto na direção de arte, quanto na animação, alcançando desta forma um resultado credível do motion capture aplicado neste contexto.



arte interativa e na visão por computador na década de 1970. Krueger desenvolveu várias instalações artísticas, entre elas o Videoplace (1985) (figura 2), onde explorava o espaço de interação física, definindo um vocabulário de interação em que os participantes usam os seus próprios corpos para interagir com o computador. O tempo real também foi um aspeto

essencial na animação performativa, pois enfatizava o feedback e a criação contínua do movimento de maneira espontânea e natural.

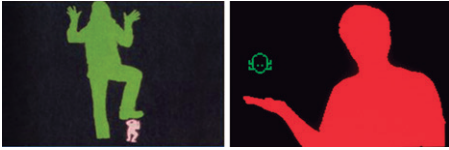


Figura 2 – Instalação multimídia de Myron Krueger, Videoplace (1985)

Para este projeto em específico, houve uma reflexão sobre como adaptar o que é do cinema, no teatro. Uma referência importante é o espetáculo *Prometeu* (figura 3), um peça multimídia inspirada no teatro de sombras, desenvolvida por Luís Leite e Marcelo Lafontana em 2014. Neste espetáculo de marionetas, verificou-se o cruzamento entre o performativo e o autômato, que inspiraram uma visão ampla da integração multidisciplinar da tecnologia em função da dramaturgia e do *storytelling*.



Figura 3 – Prometeu – Espetáculo multimídia realizado no Teatro Municipal de Vila do Conde (2014).

### Motivações e Contextualização

A proposta inicial da curta-metragem era desenvolver um sistema de produção virtual para fazer toda a gravação do filme em tempo real. Como não foi possível concretizar este modelo por limitações técnicas e temporais, a animação em motion capture do filme foi feita de forma faseada, em pós-produção e em diferentes sessões de gravação por ator, para cada parte do personagem (corpo, rosto e mãos). Também foram utilizados sistemas de motion capture mais avançados, nomeadamente o Xsens e o Manus VR, por ter uma maior precisão na interpretação dos dados.

Por outro lado, foram utilizados sistemas óticos, nomeadamente a Kinect e o Facecap que, neste contexto, mostrou-se ser uma solução mais adaptável e acessível (figura 4).

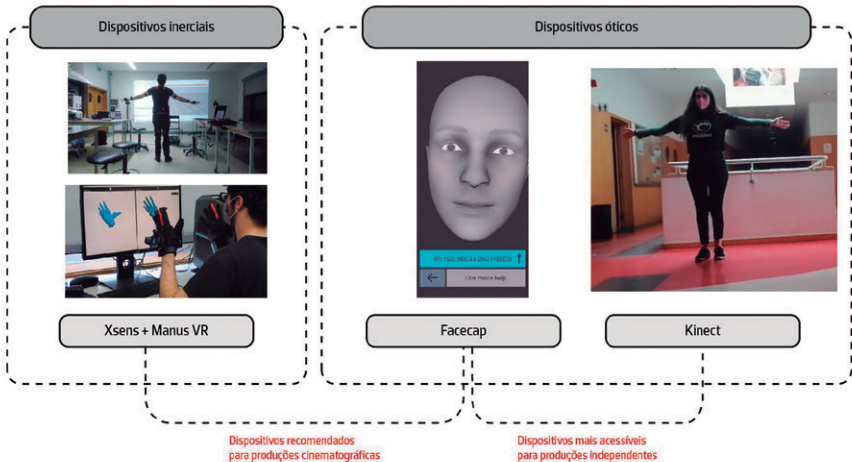


Figura 4 – Aplicações de dispositivos de motion capture

Em relação aos personagens, foi utilizado apenas o personagem Alexandre, inserido na recriação de uma das cenas do filme. O “marionetista” ao entrar na instalação, passa a ser um dos personagens do filme, que poderá interagir com o público em tempo real.

A cena recriada na instalação e abordada neste artigo é a do começo do 2º ato do filme, quando Alexandre revê as memórias que tinha de Sofia através de registos de vídeo no seu laboratório. Através deste monólogo, Alexandre decide dar um novo propósito ao projeto que estava a ser desenvolvido com a

sua esposa – o projeto “Animata”, que consistia na construção de um autômato construído pela sua esposa Sofia.

### Metodologia

Apesar de ser um processo complexo, procura-se demonstrar que é possível adaptar as metodologias utilizadas em produções de alta escala num contexto académico ou independente. Serão abordadas e descritas as etapas fundamentais para

a comunicação entre os diferentes dispositivos e software, demonstrando também as potencialidades de exploração em outros contextos.

Quando o filme é criado em animação linear, há um controle maior nas gravações e nas diferentes fases do projeto. Neste caso, como a animação é em tempo

real, é necessário confirmar que cada ligação funcione corretamente em cada fase, para garantir que também funcionem corretamente em simultâneo. Para fazer a ligação entre os diferentes sistemas (software e hardware) estabeleceram-se diferentes protocolos de comunicação representados na figura 5:

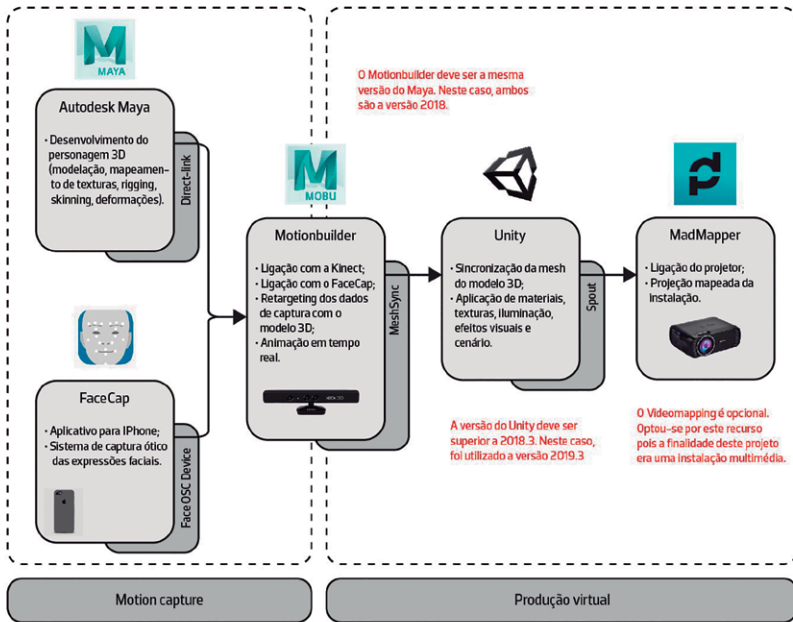


Figura 5 – Aplicações de dispositivos de motion capture

O projeto está dividido em duas etapas - enquanto que o motion capture está relacionado com a animação do personagem, a produção virtual está relacionado não só com a cenografia, mas com todo o sistema desenvolvido para produzir a performance.

Usou-se o software Motionbuilder para integrar diferentes dispositivos de motion capture e permitir trabalhar com diferentes formas de animação (linear ou interativa). Através da Kinect, é feita a captura de movimento do corpo, e através da aplicação Facecap, para iPhone, a captura do rosto. Através do plugin MeshSync, é feito o sincronismo do personagem animado para o software Unity, onde foi feita a integração com o cenário e a pós-produção cinematográfica. Por fim, através do protocolo Spout, é feito o output da instalação em vídeo mapping através do software MadMapper.

Numa produção de alta escala, o ator seria o “marionetista” principal, que controlaria o personagem através do corpo e do rosto, possibilitando a captura de vários atores ao mesmo tempo. Devido a limitações técnicas, sendo a principal, a falta de um suporte para fixar o iPhone, foi necessário recorrer a dois atores – um para animar o corpo e outro para animar o rosto do modelo 3D. O ator 1 irá animar então o personagem através do esqueleto gerado pela Kinect,

enquanto que o ator 2 irá animar as expressões faciais do personagem, através do Facecap no iPhone. O esquema da instalação está representado na figura 6:

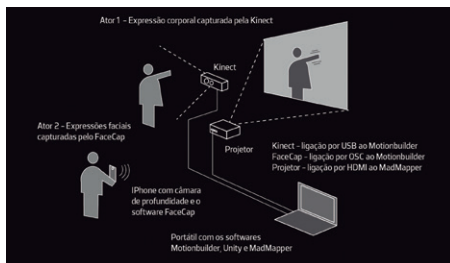


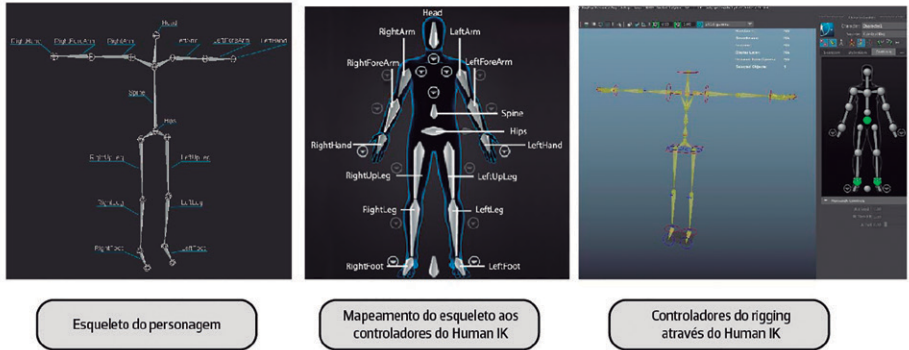
Figura 6 – Esquema de funcionamento da instalação

### Configuração do Personagem

O elemento mais importante neste projeto é o personagem. O personagem Alexandre foi desenvolvida no software Autodesk Maya, mas poderia ter sido desenvolvido com outras ferramentas de modelação 3D, como o Blender ou o 3D Studio Max, no entanto, optou-se pelo Maya pela sua compatibilidade e ligação direta com o Motionbuilder.

Na animação performativa, é importante distinguir os conceitos de “personagem” e “marioneta” (figura 7). Neste caso, o personagem é o modelo 3D associado a um esqueleto, enquanto que a marioneta, é o sistema de controladores associado às partes do esqueleto e que são utilizadas para animá-lo. Outra vantagem do Maya é o Human IK, o modelo de manipulação de esqueletos que também é utilizado no Motionbuilder. Caso o personagem tenha sido desenvolvido em outro software, seria necessário fazer o mapeamento do esqueleto com o Human IK no Motionbuilder.

Enquanto o design de personagens define a aparência do modelo 3D, o design mecânico define o comportamento do personagem, que é geralmente orientado por tarefas que determinam a maneira como o modelo 3D é construído, operado e interpretado. Sendo assim, o modelo 3D precisa estar devidamente modelado e configurado para ser animado no Motionbuilder e reproduzido no Unity.



Esqueleto do personagem

Mapeamento do esqueleto aos controladores do Human IK

Controladores do rigging através do Human IK

Figura 7 – O Personagem e a Marioneta

Antes de enviar o personagem para o Motionbuilder ou para o Unity é importante que ele cumpra alguns requisitos (figura 8). Em primeiro lugar, o modelo 3D deve estar preparado para as deformações previstas na animação, ou seja, deve-se configurar a relação entre a geometria e o esqueleto. Em segundo, o personagem deve estar devidamente texturado, com os materiais padrão do software, neste caso o Maya.

Em terceiro, o personagem deve conter as diferentes expressões faciais correspondentes ao modelo de referência do ARKit do iPhone. O que não seria necessário, se esta animação não fosse efetuada. Por fim, todos os ficheiros do projeto (Maya, Motionbuilder, Unity, MadMapper) devem estar na mesma pasta, num caminho com poucos caracteres, por exemplo, no ambiente de trabalho.

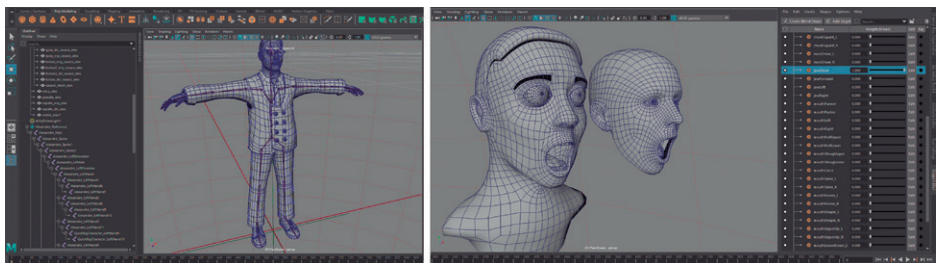


Figura 8 – Requisitos do Personagem 3D

### Marioneta Virtual

A marioneta é uma das mais antigas expressões dentro das artes performativas, mas é uma das áreas mais adaptáveis as novas tecnologias. Ao abordar este conceito, podemos invocar os termos “marioneta virtual” ou, segundo Luís Leite, “*animateer*”<sup>5</sup>, para descrever a fusão entre o ator, o marionetista e o animador. Por outras palavras, o objeto performativo.

Em termos técnicos, o processo de transposição do movimento capturado do ator para o personagem virtual é chamado de retargeting (figura 9).



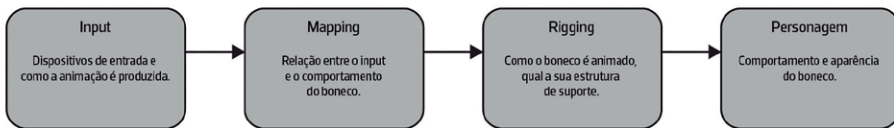


Figura 9 – Processo de retargeting entre dispositivos de motion capture (corporal) e personagens virtuais

### Animação corporal - Kinect

O Motionbuilder trabalha com o Sistema Human IK para criar as relações de retargeting entre os diferentes modelos em cena. Este software permite trabalhar com diferentes dispositivos de motion capture, como o Xsens, Manus VR, Leap motion, entre outros. Neste caso, foi escolhida a Kinect por conseguir fazer a captura completa do esqueleto do ator.

Neste projeto foram utilizadas as versões de 2018 do Maya e do Motionbuilder por questões de compatibilidade entre plugins e dispositivos. Para o Motionbuilder reconhecer a Kinect, é preciso ter o SDK da versão do dispositivo instalada no computador.

No Motionbuilder, é adicionado o dispositivo da Kinect na cena do projeto. Após verificar que o dispositivo foi reconhecido, configura-se o caminho das pastas onde as gravações serão feitas e alteramos o Character Solver para MB Character Solver. Após esta configuração, faz-se a calibração da Kinect para testar os limites de animação do modelo 3D de referência. Como a Kinect é um dispositivo de captura ótico, deve-se evitar ter ruído visual no espaço em que será feito a performance. Caso seja possível, o espaço deverá ser amplo e vazio de preferência.

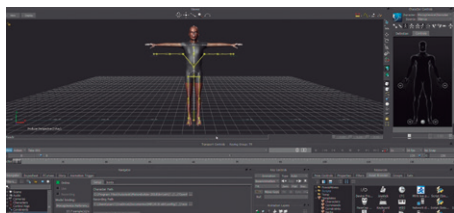


Figura 10 – Dispositivo da Kinect e Modelo de referência no Motionbuilder

Após verificar que o dispositivo funciona corretamente é possível importar o nosso personagem do Maya, através da sua ligação direta com o Motionbuilder (Send to Motionbuilder). É feito o retargeting do esqueleto da Kinect ao personagem Alexandre no Human IK, para então, verificar novamente a sua animação. O personagem não deve ter materiais complexos, pois eles não são interpretados pelo Motionbuilder pelo que serão configurados no Unity.

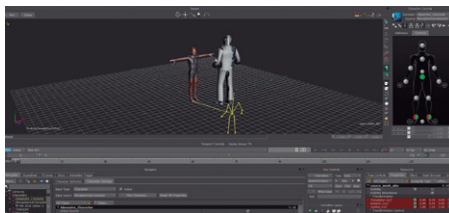


Figura 11 – Retargeting entre a Kinect e o Personagem 3D

### Animação facial - Facecap

Ao contrário do habitual nas produções cinematográficas, em vez de utilizar um sistema de marcações para manipular a geometria facial do modelo 3D, a aplicação mobile Facecap é baseada na câmara True Depth dos iPhones e nos algoritmos de reconhecimento facial da Apple utilizado no ARKit. Este sistema utiliza um sistema de parametrização baseado em FACS (Facial Acting Coding System), que é o sistema padrão utilizado por animadores e psicólogos para categorizar as expressões faciais humanas. Por outras palavras, em vez de utilizar controladores, o Facecap trabalha com deformadores, o que torna a gravação mais acessível e flexível tanto a produção, quanto a performance do ator.

Em animações mais simples, o personagem deve ter as expressões faciais básicas e os fonemas principais. Neste caso, para o personagem Alexandre foram modeladas 52 micro expressões que correspondem a parametrização facial da aplicação. Como é possível verificar na figura 12, a aplicação interpreta os dados capturados como valores de intensidade das expressões do modelo 3D (da aplicação). Após fazer a correspondência da parametrização facial, estes valores são enviados para a geometria facial do personagem.

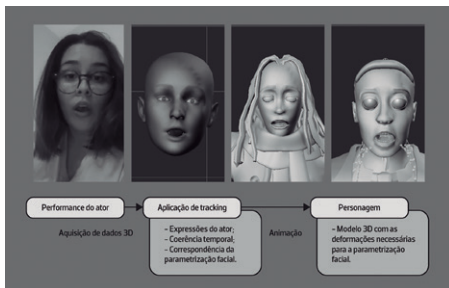


Figura 12 – Processo de retargeting entre dispositivos de motion capture (facial) sem marcação e personagens virtuais

É necessário ter em atenção que o Facecap só funciona em modelos de iPhone equipados com câmaras de profundidade. Ele permite tanto gravar e exportar clips de animação (animação offline), quanto transmitir por OSC (Open Sound Control) em tempo real os dados para o modelo 3D no Motionbuilder (animação online) (figura 13). A versão gratuita tem uma duração limite tanto para gravações quanto para o streaming, mas pode ser o suficiente para testar a aplicação com o modelo 3D desenvolvido.

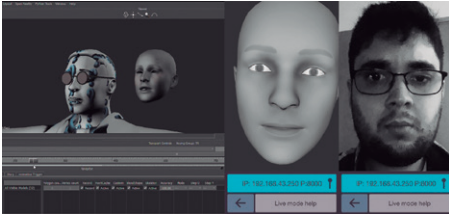


Figura 13 – Processo de retargeting entre dispositivos de motion capture (facial) sem marcação e personagens virtuais

Neste projeto, é utilizado o Facecap online, visto que é a animação será em tempo real. É adicionado em cena o dispositivo Facecap OSC e é criado uma relação de constraints (figura 14) entre os blend shapes do dispositivo (que será o source object) e os do personagem (que será o target object).

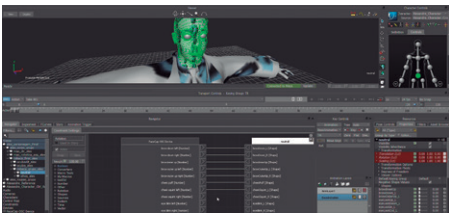


Figura 14 – Ligações entre os blend shapes do dispositivo Facecap OSC e o personagem 3D através de constraints

Após criar as relações entre os blend shapes, é feita a ligação entre o Facecap e o Motionbuilder através do dispositivo Facecap OSC. Como a comunicação é feita por OSC, é necessário adicionar o nosso endereço de rede no Facecap. Sendo este um protocolo de comunicação por rede, é recomendado não haver muitos utilizadores na mesma rede, nem utilizar outras aplicações online, para evitar falhas de latência. Caso não funcione na primeira ligação, é possível experimentar também outras portas disponíveis.

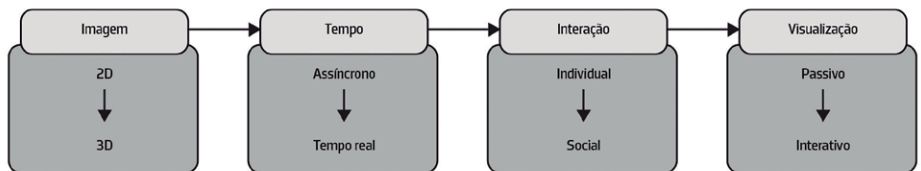


Figura 16 – Evolução do processamento de imagem e da computação gráfica

Deve-se confirmar que a ligação funciona corretamente na janela de propriedades do personagem, através da variação dos blend shapes. Nesta fase há a possibilidade de criar uma câmara auxiliar para acompanhar o rosto do personagem durante a performance, e verificar se é necessário configurar algum parâmetro facial no Facecap.

### Produção Virtual

Num mundo cada vez mais remoto, as tecnologias que até 2019 eram emergentes, foram se tornando cada vez mais o padrão na indústria a partir de 2020. A partir das tecnologias desenvolvidas em várias áreas do cinema e dos jogos, surgiu o conceito de produção virtual, que é um tipo de produção cinematográfica que combina filmagens em live-action e em (CGI) em tempo real. Por outras palavras, a essência de uma produção virtual é a de gravar um filme em computação gráfica em tempo real. Como é possível constatar nos exemplos a seguir (figura 15), o conceito de produção virtual vai além do motion capture, pois abrange todos os componentes cinematográficos.



Figura 15 – Produções Virtuais – The Lion King (2019) e The Mandalorian (2019), ambos dirigidos por Jon Favreau

Este tipo de produção foi impulsionado pela situação pandêmica, pois possibilitou a colaboração em tempo real e remoto entre cineastas, artistas e programadores. O processamento de imagens de alta-definição, que até então era o fator mais dispendioso na indústria do VFX, passa a ser feito em tempo real. O conceito de pós-produção torna-se irrelevante, pois todos os ajustes passam a ser feitos na pré-produção. Como se pode constatar na figura 16, a tendência na evolução da computação gráfica é cada vez mais a imersão e a interação entre o homem e a tecnologia.

Nas áreas da computação gráfica e do processamento de imagem, é importante distinguir os conceitos de CPU (Central Process Unit) e GPU (Graphics Processing Unit). Na última década a tecnologia baseada em GPU provou-se ser a chave para uma gama crescente de aplicações, não só nas produções virtuais, mas nos jogos, na robótica, na engenharia, na indústria automotiva e na saúde, por exemplo. Contudo, o GPU não pretende substituir o processamento de imagens em CPU, mas sim acelerar e otimizar as práticas e fluxos de trabalho existentes, maximizar a produção e compensar os custos de processamentos. Por isso, ambos os métodos devem ser ponderados para cada contexto e cada propósito.

### *Look development*

Ao contrário do Motionbuilder que é um software voltado para simulações e prototipagem, o Unity é um motor de jogos com inúmeros recursos para processamento de imagem, animação e interação. Por outras palavras, enquanto a marioneta virtual é desenvolvida no Motionbuilder, a produção virtual é desenvolvida em paralelo, no Unity, para no final, ser criado a integração entre os softwares.



Figura 17 – WINDUP, de Yibing Jiang – Unity 2020

Para este projeto, optou-se por utilizar o Unity não só pelos seus recursos, mas também pela sua flexibilidade e compatibilidade com o Maya e o Motionbuilder. Uma alternativa ao Unity é o Unreal Engine, que também possui recursos de alto desempenho, contudo possui uma pipeline diferente, ou seja, possivelmente envolveria ferramentas diferentes, como o LiveLink (ligação entre o Motionbuilder e o Unreal Engine). Entretanto, ambos os softwares são referências importantes não só na indústria de jogos, mas nas diversas áreas de visualização gráfica (arquitetura, cinema, animação, VFX, etc.). Tanto o Unity quanto o Unreal Engine fizeram um grande investimento na área de animação nos últimos anos. Na figura 17, pode-se constatar a qualidade gráfica de um dos exemplos mais recentes feitos no Unity, a curta-metragem WINDUP, dirigido por Yibing Jiang (2020).



A partir da versão 2019.3 o Unity consolidou novos modos de processamento de imagem, nomeadamente o URP (Universal Render Pipeline), mais voltado para jogos e performance, e o HDRP (High Resolution Render Pipeline), mais voltado para gráficos de alta-qualidade e produções virtuais.

Ao criar um novo projeto, existe a opção de escolher entre estes dois modelos. O modelo escolhido será o HDRP Template. Nas versões superiores a 2018.1 era necessário adicionar manualmente o Asset HDRP para configurar e adaptar o projeto a este modelo de render. Nas versões superiores a 2019.3, o projeto é configurado automaticamente, e é aberto com uma cena de exemplo com todos os recursos deste template.

Nesta etapa importam-se os assets do Maya para configurar os materiais. Para tal, os modelos 3D devem ter os materiais padrão e as texturas devidamente mapeadas. Neste projeto serão importados não só o modelo 3D do personagem, mas também o cenário desenvolvido para a curta-metragem, que será utilizado na instalação.

Ao importar os assets para a cena, os materiais vão estar desconfigurados. Para resolver isso, é criado uma nova pasta de materiais, onde se guardam os novos materiais extraídos dos assets importados – Inspector » Import Settings » Materials » Extract Materials. Desta forma configuramos os materiais dos assets para o material Lit Shader, o material padrão do modelo HDRP. Todos os assets devem ter este material configurado devidamente em cada caso.

### *Materiais e texturas*

O processamento de materiais no Unity (figura 18) envolve a combinação entre os seguintes componentes (texturas, shaders, materiais e mesh):



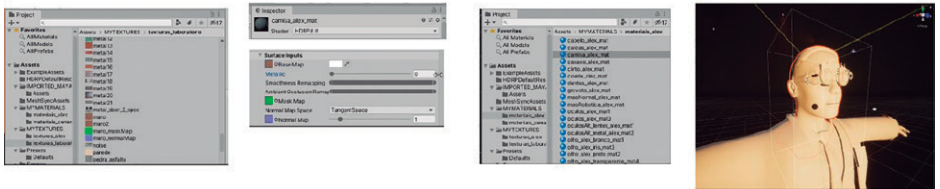
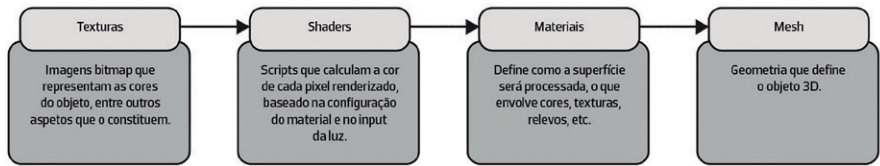


Figura 18 – Processamento de materiais no Unity

Em suma, um material é como uma interface que apresenta as propriedades e funcionalidades definidas no shader utilizado, que por sua vez, determina quais opções estão disponíveis no material. Nas versões mais antigas do Unity tínhamos o Standard Shader, contudo no HD Render Pipeline, temos o Lit shader, que é utilizado para desenvolver materiais fotorealistas – materiais PBR (Physically Based Rendering). O Unity também disponibiliza shaders PBR de outros softwares, como o Arnold Standard Surface do Maya, por exemplo.

Agora já com os materiais prontos para serem configurados, deve-se adicionar novamente as texturas criadas anteriormente para os modelos 3D. Se fosse uma produção cinematográfica, era necessário explorar um pouco mais os shaders e as configurações de cada material. Neste caso, apesar de se pretender alcançar o resultado mais credível possível, a prioridade é ter um bom desempenho em tempo real. Por isso, serão apenas trabalhados os níveis de detalhe somente do que aparece mais em cena, ou seja, o personagem (figura 19), e alguns elementos do cenário.



Figura 19 – Aplicação de materiais no Unity

Um material PBR no Unity é composto por diferentes mapas:

- Diffuse map » relacionado com a informação de cores da textura, sem luzes nem sombras;
- Normal map (bump) » relacionado com a informação dos relevos do material, baseado nas influências da luz;

- Height map » aumenta os detalhes do bump map;
- Mask map » informação comprimida dos mapas de Ambient Occlusion, Metallic e Smoothness.

Uma das peculiaridades do HDRP Template em relação aos mapas de texturas são os Mask maps (figura 20), que servem para otimizar a performance das texturas e dos materiais, que passam a ter 4 canais em vez de 12.

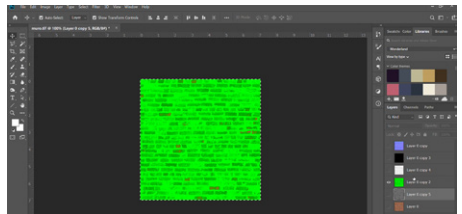


Figura 20 – Criação de Mask Maps no Photoshop

Para criar os mapas de texturas utilizamos o software Materialize para criar novos mapas de texturas e o software Photoshop para criar o Mask Map. Para criar o Mask Map no Photoshop basta adicionar cada mapa no respectivo canal de cor: Red (Metallic map – relacionado com as reflexões do material), Green (AO map – aumenta a profundidade das texturas e adiciona a sombra de contato na superfície), Blue (Detail map – detalhes complexos na textura (opcional)), Alpha (Smoothness map – se o material é áspero ou liso). Para este projeto optou-se por desenvolver manualmente os materiais, mas também é possível importar materiais PBR disponíveis na Asset Store do Unity.

**Iluminação e Processamento de imagem**

Após a configuração dos materiais criados, é feito a iluminação da cenografia (figura 21). Neste projeto foram utilizados luzes simples do Unity, como a Spot Light e a Point Light. Numa fase inicial, foram adicionadas todas as fontes de iluminação em cena

para analisar o comportamento da luz no cenário, para então, ajustar as luzes criadas para alcançar o efeito cinematográfico pretendido.

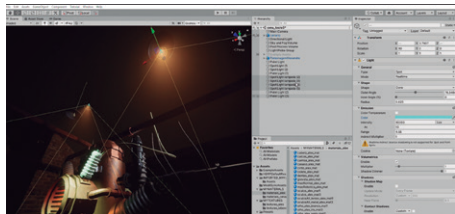


Figura 21 – Aplicação de luzes no Unity

Em relação as luzes, foram configurados alguns parâmetros simples, como a intensidade e a temperatura de cor. O modo de todas as luzes deve ser Realtime, visto que elas irão ser renderizadas em tempo real, e não em pós-processamento. Como tal, a opção Shadow Map também deve estar desativada, e a opção Volumetrics » Enable, deve estar ativada.

Foi utilizado também alguns recursos de animação do Unity para criar o efeito de iluminação do projetor (figura 22). Primeiro, foi criado um componente Animator na Spot Light inserida no projetor, que teve o parâmetro Intensity animado em Loop. Em segundo, foi utilizado o Game Object Video Player, para replicar o ficheiro de vídeo adicionado como textura animada. Por fim, aumentamos o alcance da luz, para refletir a iluminação no chão e na parede, alcançando desta forma, um resultado mais realista.



Figura 22 – Animação de componentes da luz do projetor, no Unity

Outro recurso em destaque no modelo HDRP, é o Post-Processing, uma funcionalidade de pós-processamento de imagem que permite editar a imagem final da performance. Os Overrides permitem configurar o processamento de correção de cor, efeitos visuais, efeitos de lente e configurações da câmara. Este recurso afeta diretamente a performance da nossa cena, por isso deve ser uma das últimas etapas do processo e deve ser utilizado com moderação.

O Post-Processing pode ser aplicado em toda a cena, ou aplicado como volumes, o que é mais recomendado para criar transições de efeitos entre cenas. Como já tínhamos o objeto Global Volume criado automaticamente no projeto, não foi necessário adicionar este recurso manualmente.

Nesta etapa foi definida a cinematografia, ou seja, como a iluminação irá afetar a nossa cena. Optou-se

por apagar as restantes luzes, deixando somente a luz do projetor e o seu reflexo na parede, criando desta forma, um ambiente mais sombrio e melancólico. No Post-Processing, para além dos ajustes de cor e luminosidade, foram adicionados alguns efeitos que são únicos no Unity, como o Bloom, o Grain e o Depth of Field, que ao contrário de outros softwares de produção audiovisual ou 3D, neste caso, são processados em tempo real.

### Interação

O componente mais importante na cinematografia é a câmara. Entretanto, ao contrário do audiovisual em que a câmara define a estética do frame, nas produções virtuais a estética já foi definida, por isso a câmara servirá para criar a edição e a continuidade.

O Unity possui recursos cinematográficos avançados para edição, como o Cinemachine e as funcionalidades de animação. Neste caso, como não se trata de uma animação linear e sim de uma performance em tempo real, precisamos de controlar as câmaras durante a performance.

Como o Unity é um motor de jogos, existe a possibilidade de controlar os objetos em cena através de diversos dispositivos de controle remoto, como Joysticks, Controladores MIDI, Sensores, etc. Neste caso, optou-se por uma solução mais simples, recorrendo a programação em C#. Foram criados 4 targets (figura 23) para a câmara principal que serão controladas através do teclado do computador.

Em Project Settings, definiram-se os inputs e os controladores que serão utilizados. Foi criado um script em C# chamado CamaraSwitch.cs que permite mudar o posicionamento da câmara através dos targets, durante a reprodução da cena. Numa fase experimental, podem ser criadas mais câmaras para analisar os melhores enquadramentos, mas no output final e por uma questão de performance, deve ser utilizado somente uma câmara, que servirá para fazer a partilha de imagens através do protocolo Spout para o MadMapper.

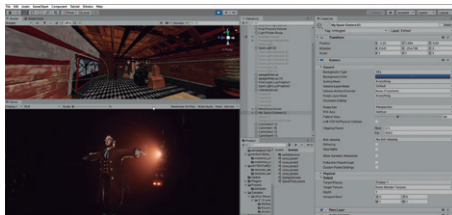


Figura 23 – Posicionamento de câmaras no Unity

### Sincronização

Como é possível constatar, é necessário testar a funcionalidade de cada etapa antes de avançar para as restantes conexões. Em relação à animação performativa, temos o sistema de motion capture a funcionar corretamente com o personagem no Motionbuilder e em relação à cenografia, temos os assets prontos para receber o personagem no Unity.

Para fazer a ligação entre o Motionbuilder e o Unity utilizaremos o plugin MeshSync. O MeshSync está disponível para as versões superiores do Unity 2019 e diferentes softwares como o Maya, Blender, 3D Max e o Motionbuilder. O plugin serve para sincronizar modelos 3D de diferentes DCC (Digital

Content Creators) com o Unity em tempo real. Esta funcionalidade não só possibilita a otimização de diferentes tipos de produção no Unity, como também a criação de experiências multimídia interativas, como é o caso deste projeto (figura 24).

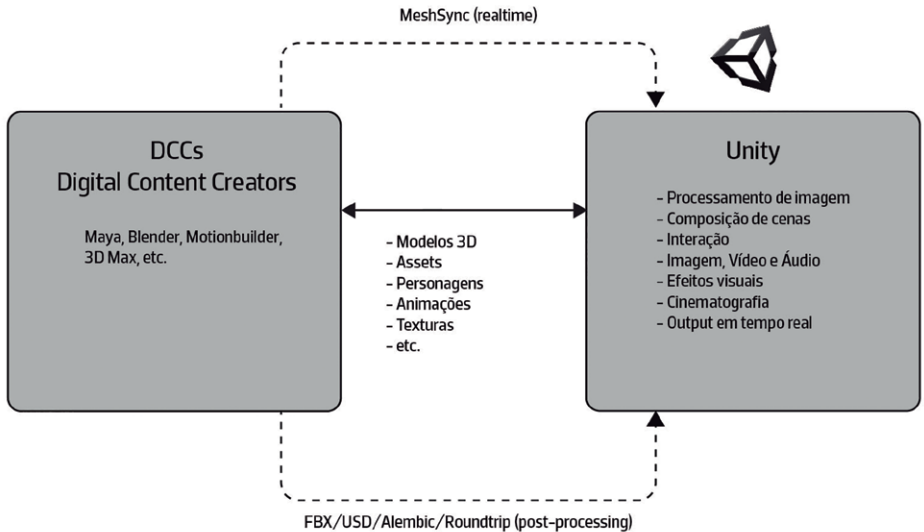


Figura 24 – Pipeline entre DCCs e o Unity, em pós-processamento e em tempo real

No Motionbuilder é preciso adicionar o plugin manualmente na pasta de dispositivos, enquanto no Unity, instalamos através do Package Manager. No Motionbuilder basta adicionar o dispositivo MeshSync ao viewport e ativar a ligação ao Unity (auto sync). Devemos ativar também a opção bake deformers para tentar regular as anomalias que possam aparecer na geometria durante a performance. No Unity, é criado um novo servidor Game Object » MeshSync » Create Server e ativamos o Auto Start para receber os dados do Motionbuilder (figura 25).

personagem é diretamente reproduzido no Unity, com os novos materiais aplicados automaticamente. Uma vantagem peculiar deste plugin é que como toda a geometria é sincronizada, conseguimos reproduzir as expressões faciais do personagem no Unity, o que até as versões anteriores a 2019 não era possível por falta de compatibilidade com deformadores.

É normal haver alguma latência, pois estamos a trabalhar com vários dispositivos em tempo real. Caso haja alguma anomalia mais notável poderá derivar da calibração do sistema de motion capture, ou pelas configurações do modelo 3D em si. Por isso que é importante verificar todos os parâmetros em cada etapa – por um lado, para evitar estes erros, por outro, caso estes erros persistam, ser possível resolvê-los mais facilmente.

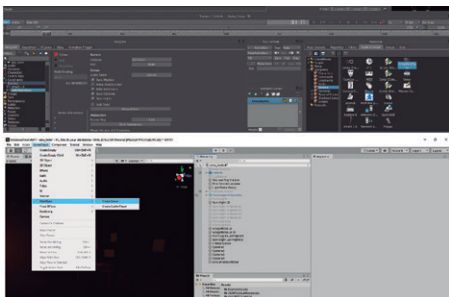


Figura 25 – Ativação do plugin MeshSync no Motionbuilder e no Unity

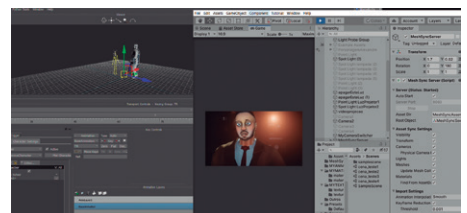


Figura 26 – Sincronização entre o Motionbuilder e o Unity através do MeshSync

A animação só é reproduzida se tivermos o Motionbuilder ativo como aplicação principal (figura 26). Após ativar o plugin podemos verificar que o

## Configurações Finais

Como é possível constatar, tanto o Unity quanto o Motionbuilder oferecem soluções para diferentes contextos da animação (figura 27). Entretanto, ainda falta configurar o output da instalação multimídia.

Para o output da instalação, é utilizado o software MadMapper (figura 28). Optou-se por este software por ser o mais flexível e adequado para o vídeo mapping, mas também poderia ter se utilizado qualquer software que reconheça o protocolo Spout, como o Cycling Max ou o Touch Designer, por exemplo. O Spout é um protocolo utilizado para partilhar imagem e vídeo em tempo real entre diferentes softwares no mesmo computador. Caso estivéssemos a trabalhar com mais de um computador, seria necessário trabalhar com o protocolo NDI. No Unity, é utilizado o package Spout4Unity para adicionar este recurso. Após importar o package e a câmara de exemplo da cena SpoutSender, adicionamos o script CamaraSwitch.cs criado anteriormente, para manipular o posicionamento da câmara. No MadMapper, a câmara Spout é adicionada à cena, tendo o cuidado de verificar se a reprodução das imagens funciona. Com o MadMapper podemos configurar a saída do projetor e dar como concluída a instalação. Como foi dito anteriormente, o Motionbuilder deve ser o software ativo principal para a animação funcionar, por isso, caso haja dois monitores, o ideal é ter o Motionbuilder e o Unity no primeiro monitor, e o MadMapper, como output, no segundo monitor ou no projetor.



Figura 27 – Output final no Unity

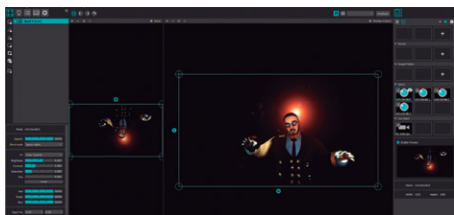


Figura 28 – Sincronização entre o Unity e o MadMapper através do Protocolo Spout.

## Conclusões e Trabalhos Futuros

Com o desenvolvimento deste protótipo<sup>6</sup>, pretendeu-não só explorar a vertente multidisciplinar da animação performativa num carácter experimental, mas também desmistificar algumas soluções para o motion capture e para produções virtuais, num contexto académico e independente <sup>7</sup>.

Neste protótipo existem várias limitações de desempenho – nomeadamente, as configurações utilizadas para a cenografia desenvolvida no Unity – e de motion capture – como as limitações da Kinect aplicadas no personagem no Motionbuilder – que devem ser levadas em consideração ao serem aplicadas numa produção virtual.

Em relação aos dispositivos utilizados, apesar da Kinect ser o dispositivo mais acessível, também é o que oferece mais constrangimentos – principalmente no que diz respeito aos limites de reconhecimento do sensor de profundidade, nas anomalias e instabilidades do esqueleto e nos limites de rotação do ator. Em produções de custo mais elevado, os sensores inerciais do Xsens, por exemplo, mostraram-se ser uma solução mais flexível e otimizada, permitindo um resultado mais credível e estável ideal para o cinema de animação. Por outro lado, apesar do Facecap ser um sensor ótico, tal como a Kinect, a aplicação mostrou-se ter o melhor custo e benefício se comparada com outras aplicações, como por exemplo, a captura facial feita por marcações.

Apesar da proposta inicial da curta-metragem *Animata*<sup>8</sup> fosse realizar a performance total em tempo real, os estudos desenvolvidos permitiram chegar a conclusões que auxiliaram tanto as gravações feitas em motion capture, quanto no desenvolvimento de novas pipelines relacionadas com as produções virtuais.

Em relação ao conceito da instalação, podemos dizer que o sistema desenvolvido permite criar novos laços de empatia com o personagem Alexandre e com a história através da interação, sendo este o objetivo primordial da construção de qualquer animação – o impacto emocional.

O presente artigo não apresenta conclusões definitivas, contudo propõe reflexões sobre as novas formas de fazer animação e de contar histórias, através de dispositivos e tecnologias que durante alguns anos, não eram tão acessíveis quanto agora. Este tipo de solução híbrida permite novas formas de encarar não só a animação, como também as artes performativas, o cinema ou os jogos, abrindo novos horizontes para a construção do que há em comum entre estas áreas: o personagem animado.

## Notas finais

<sup>1</sup> (Rodrigues, 2021).

Alma d'Arme. 2021. Mesa redonda: O que sabemos nós das Marionetas?. YouTube: <https://www.youtube.com/watch?v=yVyxGnuBRrk&t=1s>

<sup>2</sup> (Hodge 2019, 28).

Hodge, Susie. 2019. Breve História da Arte Moderna: um guia de bolso para os principais movimentos, obras, temas e técnicas, São Paulo: Editorial Gustavo Gili, SL.

<sup>3</sup> (Leite, 2018, 397)

Leite, Luís. 2018. Virtual Marionette – Interaction Model for Digital Puppetry. Tese de Doutoramento, Faculdade de Engenharia da Universidade do Porto.

<sup>4</sup> (Hamlyn, Smith, 2018, 10)

Smith, V., Hamlyn, N. 2018. Experimental and Expanded Animation: New Perspectives and Practices.

<sup>5</sup>(Leite, 2018, 18)

Leite, Luís. 2018. Virtual Marionette – Interaction Model for Digital Puppetry. Tese de Doutoramento, Faculdade de Engenharia da Universidade do Porto.

<sup>6</sup>(Ferreira, 2021).

Ferreira, M. 2021. *Animata | 2º ato – O pretérito ainda não Imaginado*. Vimeo: <https://vimeo.com/536769827>

<sup>7</sup>(Ferreira, 2021).

Ferreira, M. 2021. *Animata | A Performance em Tempo Real*. Vimeo: <https://vimeo.com/536980117>

<sup>8</sup>(Ferreira, 2021).

Ferreira, M. 2021. *Animata | Teaser Trailer (2021)*. Vimeo: <https://vimeo.com/536762128>

## Livros

Hodge, Susie. 2019. Breve História da Arte Moderna: um guia de bolso para os principais movimentos, obras, temas e técnicas, São Paulo: Editorial Gustavo Gill, SL.

Menache, A. 2011. Understanding Motion Capture for Computer Animation.

Nogueira, L. 2010. Manuais de Cinema II – Géneros Cinematográficos. Covilhã, Portugal: LabCom Books.

Ribeiro, Nuno. 2004. Multimédia e Tecnologias Interativas. Edição FCA. Lisboa.

Smith, V., Hamlyn, N. 2018. Experimental and Expanded Animation: New Perspectives and Practices.

Williams, Richard. 2001. The Animator's Survival Kit, Faber and Faber Limited, Londres.

## Bibliografia

Alípio, L., Cleto, B., Ferreira, M. 2020. Da Produção Cinematográfica a Animação performativa - Comunicação apresentada no CONFIA – 8ª Conferência Internacional de Ilustração e Animação.

Alves, P. 2013. A Importância da Personagem Virtual no Contexto da Animação 3D em Tempo Real. Tese de Mestrado, Instituto Politécnico do Cávado e do Ave.

Alves, P., Teixeira, P. M., Tavares, P. 2011. Captura de desempenho: à procura das melhores técnicas e estéticas no contexto da animação digital.

Bessa, Diogo. 2019. The Uncanny Valley of the Notion of Corrupted Being. Comunicação apresentada no CONFIA.

Bond, Rose. 2019. Boundary Crossings: A Practice-Based Intensive in Expanded Animation. Comunicação apresentada no CONFIA.

Costa Luz, Filipe. 2009. Animação Digital: Reflexos dos Novos Médios nos Conceitos Tradicionais de Animação. Comunicação apresentada no 8º Congresso LUSOCOM.

Ferreira, A., Teixeira, P. M., Tavares, P. 2013. Estudos de Expressões Faciais para Animação 3D. Comunicação apresentada no CONFIA - 2ª Conferência Internacional de Ilustração e Animação.

Gomes, Nuno. 2021. O Motion Capture no Cinema de Animação – Um estudo de caso sobre o filme Aventuras de Tintin: O Segredo de Licorne. Dissertação de Mestrado, Escola Superior de Comunicação Social.

Júnior, Cassimiro. 2009. Arte, Técnica e Estética: animação cinematográfica. Dissertação de Mestrado, Universidade Estadual de Campinas, Instituto das Artes.

Krueger, Myron W., et al. 1985. VIDEOPLACE — An Artificial Reality. In: Proceedings of the SIGCHI conference on Human factors in computing systems series CHI '85. Estados Unidos: Association for Computer Machinery (ACM).

Leite, Luís. 2014. Prometeu – Ilusão em Tempo Real. Comunicação apresentada na AVANCA – Conference

Cinema – Art, Technology, Communication.

Leite, Luís. 2018. Virtual Marionette – Interaction Model for Digital Puppetry. Tese de Doutoramento, Faculdade de Engenharia da Universidade do Porto.

Teixeira, P. M., Tavares, P. Félix, M. J. 2011. A Representação de Personagens Virtuais: Como a Tecnologia obriga à criação de estilos e estéticas interdisciplinares na animação contemporânea.

## Filmografia

*A Casa Fantasma*, 2006. De Gil Kenan. Estados Unidos: Columbia Pictures.

*As Aventuras de Tintim, O Segredo de Licorne*. 2011. De Steven Spielberg. Holanda: Paramount Pictures, Colombia Pictures.

*Avatar*, 2009. De James Cameron. Estados Unidos: 20th Century Fox.

*Mandalorian*. 2019. De Jon Favreau. Estados Unidos: Disney Media Distribution.

*O Rei Leão*. 2019. De Jon Favreau. Estados Unidos: Walt Disney Studios Motion Pictures.

*WINDUP*. 2020. De Yibing Jiang. China: Unity Technologies.

## Webgrafia

Alma d'Arame. 2021. Mesa redonda: O que sabemos nós das Marionetas?. Youtube: <https://www.youtube.com/watch?v=VyXGnuBRrk&t=1s>

ARFaceAnchor, Apple Developer Documentation: <https://developer.apple.com/documentation/arkit/arfacedanchor>

Facecap. Bannaflak: <https://www.bannaflak.com/facecap/>

Ferreira, M. 2021. *Animata | Teaser Trailer (2021)*. Vimeo: <https://vimeo.com/536762128>

Ferreira, M. 2021. *Animata | 2º ato – O pretérito ainda não Imaginado*. Vimeo: <https://vimeo.com/536769827>

Ferreira, M. 2021. *Animata | A Performance em Tempo Real*. Vimeo: <https://vimeo.com/536980117>

Motionbuilder Support. *Connecting Motionbuilder to Kinect*. Autodesk: <https://knowledge.autodesk.com/support/motionbuilder/learn-explore/caas/CloudHelp/cloudhelp/2017/ENU/MotionBuilder/files/GUID-60A372EC-154D-402A-B56F-DC41FE3FF41F-htm.html>

Plugin Spout4Unity. Github: <https://github.com/sloopidoopi/Spout4Unity>

Plugin MeshSync. Github: <https://github.com/unity3d-jp/MeshSync>