

# Otimização do processo de construção e animação de modelos 3D para jogos digitais utilizando técnicas de digitalização 3D e *Mocap* com Kinect®

Alessandro P. Lima Carolina B. Pillon Thiago da S. Krening Fábio P. da Silva  
José L. F. Aymone

Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Design, Brasil.

## Abstract

This paper aims to show the results of experiments on the 3D scanning and motion capture for games and animation. For accomplishing it, a previous study on the main accessible techniques of digitalization and motion capture is made, as well as a literature review on animation and games. On the last part of the paper, the chosen techniques are shown (white light scanning and Kinect® based motion capture), demonstrating the results and discussing the situations on which they would be of good use to improve these processes.

**Keywords:** 3D scanning, motion capture, animation, games

## Authors' contact:

{alessandropilma, carolinabpillon,  
thiago.krening}@gmail.com  
{fabio.silva, aymone}@ufrgs.br

## 1. Introdução

A computação gráfica hoje se encontra em um nível jamais visto anteriormente, graças ao avanço tecnológico de hardware e software. Mas tudo não passa de mera síntese da vida real, uma forma de imitar a vida dentro de ambiente computacional que faz da apresentação digital uma imagem crível aos olhos das pessoas. Tanto que, para Azevedo, pode-se “interpretar o processo de rendering como o ato de converter dados em uma imagem realística ou simplesmente sintetizar um objeto ou cena até ter-se deles uma aparência de algo real e não de formas inteiramente criadas no computador” [2003].

Tal processo que envolve a renderização está diretamente ligado à capacidade das pessoas em associar tudo aquilo que estão vendo a algo que já foi vivenciado, competindo aos designers produtores de imagens renderizadas (apresentadas, em uma tradução direta) produzir materiais que permitam as pessoas mais rapidamente realizar esta associação. Dentro desta perspectiva, Azevedo [2003] ainda aponta fatores fisiológicos (informações monoculares) que corroboram a este pensamento:

“As informações monoculares são inerentes à imagem formada na retina, são também chamadas de *static depth cues* (informações estáticas de profundidade). Entre as informações monoculares pode-se citar a noção de perspectiva linear, o

conhecimento prévio do objeto, a oclusão, a densidade das texturas, a variação da reflexão da luz e as sombras” [Azevedo 2003].

Com esta causa fisiológica, consegue-se associar imagens digitais a coisas que já vivenciamos, fazendo assim com que nosso cérebro acredite naquilo que vê. As técnicas de produção de modelos de hoje são bem variadas, podendo assumir diferentes perspectivas de produção. Segundo Ward, para a construção de um modelo tridimensional para jogo convencional, devem-se assumir as seguintes etapas: construção de um modelo básico, otimização, mapeamento UV, criação de texturas e testes na *Engine de Render* [2008].

A elaboração de modelos seguindo este procedimento gera arquivos excelentes, mas, como dito, novos procedimentos são incorporados ao processo de produção. Tais processos buscam muitas vezes aumentar a qualidade do trabalho, mas, em contra partida, podem onerar o tempo de produção. Em alguns casos, pode-se priorizar a qualidade ou o tempo de produção mas, às vezes, tanto a qualidade quanto os prazos são igualmente importantes. Neste caso, como mensurar cada etapa de modo a resultar em um modelo devidamente funcional e estético para o meio a qual deve fazer parte? É nesse cenário que alternativas surgem na produção de modelos digitais, e tais alternativas muitas vezes necessitam de certo aparato tecnológico para que efetivamente funcionem adequadamente.

Atualmente, as principais empresas de computação gráfica do mercado (neste ponto, não se faz distinção entre empresas de jogos ou de filmes), estão se equipando com verdadeiros arsenais de produção, para chegar a resultados cada vez mais rápidos e polidos frente à produção e animação de modelos. Como exemplo atual, o diretor James Cameron, no seu filme "Avatar" [2009], utilizou não apenas captura de modelos para desenvolver as personagens, como também os movimentos animados de atores reais foram inseridos em modelos digitais e em tempo real. O diretor pôde validar o desempenho das personagens virtuais, manipulando estes por meio de seus respectivos atores reais. A partir do exemplo de Cameron, muitos outros projetos seguiram-se desenvolvendo a tecnologia usada pelo diretor. Na ocasião do filme "Avatar", em 2009, existiam apenas três câmeras no mundo com tal tecnologia. Hoje este número de câmeras é bem maior e conta inclusive com

mais fornecedores para alimentar a produção de Hollywood.

Nasce uma questão frente a este cenário, que é o cerne principal deste trabalho: como estabelecer um processo de produção de baixo custo, buscando o máximo em resultados, com os menores tempos de produção? Em um mundo globalizado e cheio de informações e recursos disponíveis, o tempo e recursos financeiros seguem uma lógica inversa, sendo cada vez mais escassos.

## 2. Construção e animação de modelos 3D

Nesta sessão serão apresentados os principais meios, técnicas, ferramentas e softwares utilizados neste estudo, sendo identificados e contextualizados. Posteriormente suas aplicações práticas são vistas e explicadas.

### 2.1. Digitalização 3D

Os processos de digitalização de modelos tridimensionais são hoje muito usados tanto na indústria cinematográfica como na de jogos e publicidade. Encontram-se modelos ainda em aplicações educativas ou fisioterápicas, ajudando pacientes em sua reabilitação (tipicamente vistos em jogos, chamados de “fisiogames”). O processo de aquisição de dados nessas situações pode dar-se das mais variadas maneiras. A mais tradicional é a descrita por Ward [2008], em que o designer concebe o modelo a partir de desenhos planejados, realizando uma tradução do meio 2D para o 3D.

Tal processo descrito por Ward [2008], além de ser um dos mais tradicionais, também é um dos mais baratos, pois depende diretamente da capacidade do designer em representar digitalmente figuras reais ou imaginárias. Seu sucesso não depende tanto dos recursos que utiliza (hardware ou software) e sim de sua habilidade cognitiva em representar os dados que constituem o modelo de forma digital. Esta técnica gera um inconveniente: eventualmente o modelo gerado nestas condições sofre “interpretações” por parte do designer que o executa, e isso pode ser indesejado dependendo do projeto a que o modelo é submetido.

Para evitar estas situações, processos complexos de aquisição de dados podem ser utilizados, tornando a técnica mais precisa conforme os investimentos feitos. Uma das técnicas que parece oferecer uma boa porcentagem de sucesso, pois ela é eficiente e relativamente barata se comparada a outras técnicas que utilizam ferramentas próprias, é a fotogrametria. Em seu processo, para adquirir dados, ela precisa de boas câmeras fotográficas e local devidamente iluminado, ao passo que outras técnicas necessitam de hardware específico. Neste cenário surgem equipamentos que, juntamente com softwares, permitem a captura digital de elementos para facilitar seu processo de produção. Hoje existem diversos tipos

de fabricantes e tipos de aquisição de dados, cada um com um propósito e fins distintos.

#### 2.1.1. Equipamentos e técnicas para gerar modelos

Para a elaboração deste trabalho, optou-se primeiramente por procurar identificar quais seriam os requisitos de hardware que permitiriam desenvolver um bom modelo com aquisição de dados e, em seguida, escolher o aplicativo de software mais barato (ou mesmo gratuito) para gerar o modelo digital. Esperava-se realizar trabalho adicional em sua aquisição de dados, seja para modelagem, seja para textura.

Das alternativas avaliadas, o uso de notebooks pessoais dos participantes deste trabalho pareceu a escolha mais adequada, visto a possibilidade de se trabalhar em qualquer local e em qualquer horário. Para softwares foram utilizados o Autodesk 3DS MAX® pois já se tinha licença para uso em projetos como este, bem como o Adobe Photoshop. O software Autodesk 123DCatch® também foi utilizado para testes de geração de modelo pela técnica de fotogrametria (versão gratuita). Os softwares iPi Recorder® e Studio (versão demonstrativa) também foram usados, mas para a captura de dados animados.

#### 2.1.2. 123DCatch®

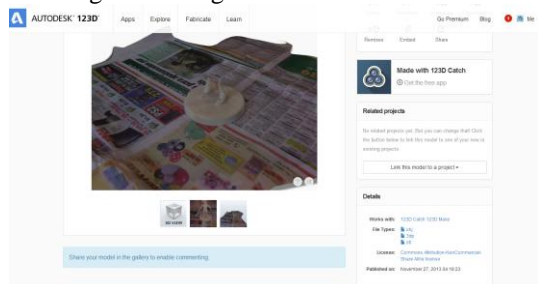
Para a aquisição de dados tridimensionais para construção do modelo digital, primeiramente optou-se pela fotogrametria utilizando o software 123DCatch®, disponível de forma gratuita *Standalone* ou via *Web browser*. A grande vantagem dessa ferramenta é que é possível adquirir dados não somente de modelo, mas também de imagem para o revestimento em textura.

Como experiência, foi utilizada a versão online do sistema 123DCatch®, enviando desta feita as imagens adquiridas da sessão fotográfica de captura dos modelos. O sistema recebe o material e de modo intuitivo, vai orientando o usuário para as demais etapas, que constituem em selecionar as fotos a serem utilizadas na construção do modelo tridimensional, seguido de uma tela em que se escolhem as melhores fotos para o sistema. Por fim, envia-se ao sistema online da Autodesk para que lá seja gerado o modelo. Neste momento, caso o usuário não esteja conectado com uma conta da Autodesk, ele pode escolher acessar o sistema por algum perfil de rede social que ele possua e que seja conectado aos serviços da empresa.

Após entrar no sistema e confirmar o envio das fotos para geração do modelo, o sistema processa as informações, e depois de certo tempo, disponibiliza o modelo digitalizado na mesma tela em que o usuário está utilizando o software em sua versão online, ou envia um aviso por email sobre a finalização do modelo digitalizado. O tempo de espera para que o modelo seja finalizado está diretamente relacionado ao tipo de conexão de internet que o usuário possui (quanto mais rápida, mais rápidos são os resultados), bem como o tipo de captura, se de baixa ou alta

qualidade (para o presente estudo foram feitas capturas de baixa qualidade). A Figura 1 ilustra a tela de interface do software.

Figura 1: imagem do software 123DCatch®.



Fonte: <http://apps.123dapp.com/catch/>.

### 2.1.3. Scanner de Luz Branca

Outra técnica que também foi explorada neste trabalho baseou-se na aquisição de dados tridimensionais por meio de scanner digital conhecido por luz branca. Consiste em um aparelho (hardware) da marca Artec, Modelo Eva, que pode ser manuseado livremente pelo usuário. Na verdade ele se parece com a forma de uso de aparelhos detectores de metais de aeroportos, onde o usuário passa o aparelho por sobre o modelo real e automaticamente o hardware passa estas informações ao software, e este vai exibindo o modelo digitalizado em etapas na tela.

## 2.2. Animação 3D e games

Animação é, basicamente, um processo que cria ilusão de movimento através e apresentação de imagens sequenciais em rápida sucessão [Chong 2011]. Dentro disto, existem diversas técnicas, desde animações em *stop-motion* - utilizando-se gravação quadro a quadro, da manipulação de um objeto inanimado em cenário físico espacial [Purves 2011] - e tradicionais - onde os *frames* são desenhados à mão - até a computação gráfica, mais recente técnica, desenvolvida com o auxílio de softwares e hardwares específicos.

Os processos de animação, apesar de terem nascido no cinema, espalharam-se por diversas mídias. A computação gráfica, como dito por Chong [2011], estava inicialmente sob o domínio da indústria. Isto se deu devido ao alto custo de produção e de investimentos necessários em equipamentos e pesquisa. Com o passar os anos, as tecnologias de computação gráfica se tornaram cada vez mais acessíveis e os animadores digitais, mesmo com ferramentas mais simples que estúdios e instituições, agora têm acesso e controle sobre as mesmas imagens e estética, devido "à digitalização da mídia e a democratização da tecnologia" [Chong 2011].

Paralelamente, a indústria de *games* cresceu de forma exponencial, adotando técnicas de animação em computação gráfica antes somente utilizadas em grandes produções cinematográficas. Isto foi possível devido à melhoria do poder de processamento dos consoles, permitindo que imagens com alto nível de

realismo e detalhes fossem exibidas na tela em tempo real.

Nesta seção são abordadas algumas questões básicas de animação e discutidas as tecnologias disponíveis, fazendo-se um paralelo com os testes realizados no projeto aqui descrito.

### 2.2.1. Tecnologia e técnicas

A animação de modo geral tem suas origens compartilhadas com o cinema. "As primeiras experiências para fazer imagens em movimento utilizavam figuras criadas à mão" [Chong 2011]. Depois se passou pela fotografia até o refinamento dos processos para captura de imagens em movimento, chegando-se ao cinema. Já nos primeiros anos do século XX, o precursor do cinema narrativo e de truques Georges Méliès produziu alguns filmes que se baseavam em efeitos especiais. Nestes filmes (como por exemplo, *Le Voyage à travers l'Impossible*, de 1904) Méliès incorporou efeitos de ilusão e animação ao *live-action*, ou seja, o cinema com atores e cenários reais. Estas ilusões de ótica deram origem aos efeitos especiais e contribuíram para a evolução da animação.

Alguns anos mais tarde, em 1915, Max Fleischer inventava o rotoscópio. Trata-se de um dispositivo que buscava trazer não só agilidade ao processo de animação tradicional como também mais realismo aos movimentos. A ação anteriormente filmada era projetada, um quadro por vez, em uma tela. Este movimento podia ser traçado no acetato, copiando-se a imagem original [Chong 2011]. A rotoscopia, portanto, foi uma precursora das atuais técnicas de captura de movimento.

Muitas vezes o uso de movimento real para animação, através do rotoscópio, se torna artificial na tela, pois se comparado a uma sequência totalmente animada à mão, acaba se destacando. Mesmo técnicas como o *pixilation*, também conhecida como *stop-frame*, onde o assunto a ser animado é o próprio ser humano, cria um "contexto teatral e particular" [Chong 2011]. Para que haja uma satisfatória integração entre animação manual e captura de movimento, a intervenção do artista, no caso o animador, é essencial. Esse ponto é demonstrado mais adiante no artigo, onde os testes de captura são realizados.

Os primeiros testes de animação computadorizada foram realizados ainda na década de 1950. Segundo Chong [2011], John Whitney Sr. foi um dos precursores destes experimentos, colaborando com o designer gráfico Saul Bass no projeto de aberturas de filmes. Um exemplo é a sequência de créditos de *Vertigo* (no Brasil, "Um corpo que cai"), de Alfred Hitchcock, lançado em 1958.

Nas décadas de 1950 e 1960 havia a preocupação de tornar os custos de produção de animações mais acessíveis. Conforme os curtas de animação passavam do cinema para a televisão, a necessidade de diminuir custos e otimizar processos se tornou mais evidente. Como explicado por Denis [2007], os estúdios passaram a utilizar a chamada animação reduzida. Essa técnica permitia utilizar planos em que um mínimo de

animação era aplicada. Um exemplo comum eram as cenas de diálogo, onde nada além da boca dos personagens tinha movimento. Além disso, muitas sequências (como corridas, caminhadas e outras ações corriqueiras) eram reutilizadas à exaustão.

Alguns anos mais tarde, já na década de 1970, os primeiros videogames começaram a se popularizar. Inicialmente, tratava-se de jogos bastante simples, como *PONG*, um simulador de tênis de mesa que demonstrava apenas alguns retângulos brancos em movimento. Em 1978 foi publicado *Space Invaders*, jogo que, segundo Thompson et al. [2007], foi o primeiro a realmente capturar o imaginário dos jogadores. Isso se deu, pois ele incorporou elementos narrativos ao sistema de jogo - o objetivo de salvar o planeta de uma invasão alienígena.

Por anos, antes que surgissem as primeiras instituições especializadas em efeitos especiais, as imagens geradas por computador (*CGI*) exigiam trabalho de empresas de computação ou laboratórios de pesquisa. Os primeiros filmes a utilizarem efeitos computadorizados os faziam para situações pontuais e mais simples, como as interfaces de computadores de "*Alien - O Oitavo Passageiro*", de 1979 [Chong 2011]. Nos anos 80, os videogames se tornavam cada vez mais populares e os efeitos especiais em cinema caminhavam para a digitalização. Em 1982 foi lançado "*Tron*", um dos primeiros filmes a usar animação digital de forma representativa no cinema comercial, possuindo inclusive influência dos videogames.

Em meados dos anos 80 e 90 começaram as primeiras experiências em se produzir personagens digitais. Para o filme "*O Enigma da Pirâmide*", lançado em 1985, a empresa *Industrial Light and Magic* usou um scanner 3D para capturar as coordenadas de uma maquete com o objetivo de construir um personagem tridimensional [Chong 2011], o que exigiu que a empresa criasse um software específico para utilização dos dados capturados. Chong [2011] ressalta que nesta produção o movimento do ator precisou ser monitorado manualmente para ser utilizado como *key frame* de um modelo digital, enquanto que em 2001, na produção do filme "*O Senhor dos Anéis*", toda a gravação de movimento era digitalizada e seu monitoramento, gravação e replicação realizados em computadores. Isto permitiu que a representação de personagens digitais se tornasse extremamente realista e sofisticada.

Com o intuito de buscar soluções acessíveis para a digitalização e captura de movimentos, o presente estudo buscou fazer alguns testes que são relatados aqui. O objetivo é relacionado com aquele que as empresas de efeitos especiais e games vêm buscando nas últimas décadas: a otimização de criação de imagens e animações digitais. A seguir, são descritas as técnicas de captura de movimento testadas, bem como os resultados da utilização destas em conjunto com as digitalizações descritas anteriormente.

### 2.3. Captura de movimento

O propósito desta seção é apresentar os conceitos sobre a captura de movimento, ou *mocap*, abreviação em

inglês para *motion capture*. Os dispositivos de captura de movimento que, de acordo com Gomide [2009], podem ser classificados como mecânicos, magnéticos ou óticos, são utilizados para rastrear os movimentos de atores reais e representá-los por meio de softwares.

Os sistemas mecânicos de captura de movimento podem ser inerciais, acústicos ou protéticos. Estes dispositivos funcionam de maneira semelhante, ou seja, possuem transmissores que são posicionados nas principais articulações fornecendo as posições e orientações das juntas em tempo real. Já a captura magnética possui transmissores que emitem campos magnéticos e os receptores calculam a orientação das articulações em relação a uma antena transmissora, a qual emite um sinal de pulso. Finalmente, na captura ótica, os transmissores são refletores, em geral emissores LED, que são posicionados nas juntas. As câmeras funcionam como receptores e geram as coordenadas bidimensionais de cada refletor. Desse modo, um conjunto de dados bidimensionais capturados pelas câmeras oferece as coordenadas tridimensionais dos refletores [Gomide 2009].

Segundo Kuhn [2005], a tecnologia de captura de movimento desperta interesse nas mais diversas áreas, como a indústria cinematográfica, jogos, ergonomia, desempenho desportivo e análise de movimento. Porém, o autor acredita que o sistema de *mocap* é uma realidade distante para a maioria das empresas, universidades e grupos de pesquisa no Brasil devido ao alto custo dos softwares e hardwares envolvidos.

Os sistemas de captura óticos de movimento utilizados em grandes produções de cinema e jogos custam algo em torno de 15 a 100 dólares por segundo capturado, dependendo da complexidade da cena e da quantidade de profissionais envolvidos [Chandler 2012]. Dessa maneira, a captura de movimento realizada com o sensor de movimento Kinect®, desenvolvido pela Microsoft para o para o Xbox 360® e Xbox One®, representa uma tecnologia de baixo custo para os estudantes e profissionais que desenvolvem animações tridimensionais para jogos e outras produções digitais.

Nesse sentido, a presente pesquisa pretende contribuir com a discussão sobre a utilização do Kinect® como recurso para realizar a captura de movimento de atores reais. A seguir são descritas as etapas de planejamento, requisitos e lista de tomadas de captura de movimento a partir dos equipamentos e técnicas utilizados na pesquisa.

#### 2.3.1. Equipamentos e técnicas para animação

O processo de animação tridimensional, conforme Novak [2010], utilizado nos jogos, pode ser realizado por meio de duas técnicas: quadros-chave (*Keyframing*) e captura de movimento. O primeiro método consiste em criar cada posição de um movimento e definir quadros-chave sequenciais para gerar os arquivos de animação. Já na captura de movimento o animador captura os movimentos de pessoas reais colocando marcadores em suas articulações, para monitorar, por intermédio de

sensores, as variações de movimento e gerar dados de movimento.

A captura de movimento, sejam movimentos reais de seres humanos, animais ou objetos, podem ser gravados e implementados para a obtenção de um efeito mais realista e natural. Além disso, à medida que a tecnologia avança, os jogadores esperam ver os personagens dos jogos se movendo e se comportando mais realisticamente [Chandler 2012]. A tecnologia utilizada para capturar os movimentos faciais no jogo L.A. Noire, em artigo publicado no site Gamasutra, por exemplo, serve tanto para transmitir as expressões faciais com mais realismo, quanto funciona como mecânica de jogo. Durante a cena de interrogatório, o jogador deve observar se o suspeito mente ou fala a verdade através de sua expressão facial.

Desse modo, a tecnologia de captura de movimento pode ser empregada nos jogos para retratar os movimentos e as expressões faciais dos personagens com mais realismo. Entretanto, é necessário planejar a sessão de captura de movimento com antecedência, estabelecendo os equipamentos e técnicas necessários para a produção. Pois, de acordo com Chandler [2012], dependendo da quantidade de movimentos capturada e de animadores trabalhando no projeto, isso pode levar vários meses de trabalho, mas o resultado é recompensador.

No planejamento da captura de movimentos, primeiramente o animador lista todos os movimentos necessários para o jogo, que incluem andar, correr, segurar itens, abaixar, deitar, virar, morrer, entre outras ações que são visualizadas no jogo. Então se decide quais animações serão realizadas através do método de quadros-chave (*Keyframing*), ou *mocap*. A vantagem de utilizar a técnica de quadros-chave é que o animador pode criar sequências de animações que uma pessoa real não conseguiria reproduzir. Ao passo que, a grande desvantagem de animar a partir do zero é que os movimentos podem não parecer tão realistas e naturais quanto o método de captura de movimento [Chandler 2012]. A técnica adotada nesta pesquisa consiste em capturar o movimento utilizando o sensor de movimento Kinect®.

Após o planejamento, os requisitos para a captura de movimento são determinados pela equipe de produção das animações do jogo. Isto inclui o gerenciamento dos dados de captura que são convertidos em ciclos de animação a serem usados nos jogos. Estes arquivos são denominados *assets*, e podem ser de vários tipos, como texto, arte e áudio. É importante, da mesma forma, estabelecer uma convenção de nomenclaturas dos arquivos gerados para que o animador possa identificar o movimento sem ter que abrir o arquivo. Por fim, convém determinar um formato de arquivo padrão para os dados de captura de movimento que estão sendo gravados [Chandler 2012]. O formato padrão utilizado neste trabalho foi o *bvh* (*Biovision Hierarchical Data*), desenvolvido pela Biovision. De acordo com Kuhn [2005], o *bvh* é um formato de arquivo padrão para armazenar os dados de captura de movimento, sendo suportado pelas

principais ferramentas de animação do mercado, como o 3DS MAX®.

Depois de definir os requisitos do projeto, é necessário que a equipe de animação elabore uma lista para determinar o número de atores que serão solicitados na sessão de captura de movimento. Os atores contratados devem ter as mesmas dimensões corporais, como altura e peso, dos personagens do jogo [Bates 2004]. Neste caso, a captura de movimento foi feita com a mesma pessoa que serviu como modelo para a digitalização, assim as medidas do esqueleto capturado correspondem com o modelo digitalizado.

Além disso, os animadores devem listar os movimentos que serão capturados e organizar a informação em uma planilha de *mocap*. É preciso definir o estúdio onde será realizada a captura de movimento e os custos com o serviço, mão-de-obra e equipamentos. Se o orçamento for restrito, algumas empresas também vendem os dados que podem ser comprados e aplicados nos modelos já criados [Bates 2004]. Neste projeto, a captura de movimento foi realizada no Laboratório ViD – Virtual Design da Universidade Federal do Rio Grande do Sul, UFRGS. Os alunos utilizaram um Kinect® que foi conectado a um notebook com os programas iPi Recorder® e iPi Mocap Studio® instalados.

Em uma etapa posterior, os arquivos de captura de movimento são importados em softwares específicos de animação tridimensional. Para facilitar a manipulação dos dados, em muitos programas existem ferramentas automatizadas que permitem reduzir a quantidade de quadros-chave, uma vez que as animações de capturas são construídas quadro a quadro do início ao fim da animação. No presente estudo, os arquivos no formato *bvh* foram carregados no Character Studio do 3DS MAX® com o intuito de refinar a animação de captura.

Portanto, a grande vantagem da captura de movimento é o realismo e a naturalidade que proporcionam na animação dos personagens. Esta tecnologia é especialmente útil para os jogos de esporte, já que é muito difícil reproduzir os movimentos característicos dos atletas através de métodos tradicionais de animação [Bates 2004]. A principal desvantagem é o orçamento da captura de movimento, contudo existem equipamentos de baixo custo, como o Kinect®, que permite capturar os movimentos reais dos atores por um preço mais acessível para pequenas empresas.

### 2.3.2. Kinect®

O Kinect® é um dispositivo periférico para o console de videogame Xbox 360® e Xbox One®. O projeto foi desenvolvido pela Microsoft Research em parceria com a empresa israelense PrimeSense que criou a tecnologia para detecção de movimentos. O Kinect® possui sensores com capacidade para realizar o reconhecimento gestual, assim como responde aos comandos de voz do usuário.

A interface baseada em gestos funciona com a utilização de sensores que mapeiam os movimentos do

usuário e são reconhecidos pelo sistema. O sensor do Kinect® inclui uma câmera RGB (*Color sensor*) que armazena os dados em três canais: um canal com resolução de 1280x960 a 12 fps (*frames* por segundo), um canal com resolução de 640x480 a 30 fps, e um canal YUV com resolução de 640x480 a 15 fps. Além disso, o sistema possui um emissor (*IR Emitter*) e um sensor de infravermelho (*IR Depth Sensor*). O emissor lança feixes de luz infravermelha que são refletidos e detectados pelo sensor de profundidade. Os feixes refletidos são convertidos em dados de profundidade que informam a distância entre o objeto e o sensor. O Kinect® dispõe, ainda, de um conjunto com quatro microfones (*Microphone Array*) para captar o som, e um motor de inclinação (*Tilt Motor*) que determina a orientação do sensor [Microsoft 2014].

Recentemente, a Microsoft lançou o Kinect® para o sistema operacional Windows, disponibilizando o *Software Development Kit – SDK*, e o *Windows Developer Toolkit* para o Windows 8, Windows 7, e Windows Embedded Standard 7. O SDK, além de permitir o desenvolvimento de aplicativos interativos utilizando as linguagens de programação C++, C# e Visual Basic, oferece recursos adicionais como o Kinect® Studio, que possibilita gravar e reproduzir ensaios de profundidade produzidos com o Kinect® [Microsoft 2014].

Desta maneira, tornou-se possível capturar o movimento de atores reais com a utilização do Kinect® para o Windows. O Kinect® possui a capacidade de reconhecer e rastrear os movimentos dos usuários em tempo real. O dispositivo reconhece vinte articulações do corpo, como cabeça, pescoço, ombros e braços, e representa a sua localização no espaço tridimensional. As juntas do modelo, tal como nos softwares tridimensionais, são criadas de forma hierárquica. O rastreamento das articulações é feito com base em três estados possíveis: *tracked*, articulação visível para o sensor; *not tracked*, articulação fora do alcance do sensor; *inferred*, a articulação é inferida por meio de interpolação entre as articulações adjacentes [Microsoft 2014].

Shotton et al. [2011] desenvolveu um método para rastrear as articulações do corpo através de um mapa de profundidade. Neste procedimento, o modelo é segmentado em diferentes partes, sendo que para cada segmento é atribuída uma cor. Desse modo, cada cor corresponde à probabilidade do pixel pertencer a uma determinada articulação. A posição aproximada da articulação no espaço tridimensional é calculada a partir do ponto médio de cada segmento. Assim, pode-se determinar a localização das articulações do corpo humano e, com base nos pontos que foram mapeados, torna-se possível criar o esqueleto do corpo em movimento.

Depois de identificadas as articulações no corpo do ator, os dados são manipulados nos softwares especializados na captura de movimento em tempo real. Neste caso, foram usados os programas iPi Recorder®, para gravar o vídeo de captura de movimento, bem como o iPi Mocap Studio®, para processar as informações que podem ser salvas em

diferentes formatos como o *fbx*, *bvh* e *txt*. Então, são exportados para os softwares de animação tridimensional, ou para *engine* Unity3D®.

## 2.4. Digitalização 3D e captura de movimento

Os processos de digitalização 3D e captura de movimento têm sido cada vez mais usados na indústria de cinema e jogos digitais por seu caráter agilizador de qualquer processo de produção de modelos e animações críveis. Ainda em 1999 foram utilizadas técnicas de captura de movimento em filmes como "Matrix" [1999] e "Final Fantasy: The Spirits Within" [2000], ambos com produção iniciada antes dos anos 2000. De lá para cá muita coisa mudou e se antes se utilizavam equipamentos caríssimos e roupas com no máximo 16 pontos de captura acoplados no corpo de atores reais, que eram então interpretados por softwares e gerados os modelos de esqueleto animados (informações do DVD Making of de Final Fantasy: The Spirits Within), hoje esta tecnologia está acessível a consoles de vídeo game como Microsoft Xbox e seu dispositivo Kinect®.

Nesta sessão, serão abordados os procedimentos de aquisição de dados volumétricos para os modelos tridimensionais.

### 2.4.1. Aquisição de dados por fotogrametria

Para esse trabalho, foram feitas algumas sessões de captura de imagens de modelos para fins de experimentos. Após ler a documentação do software 123DCatch® e entender os requisitos necessários para assegurar seu bom funcionamento (extrair o maior número de imagens possíveis, cuidar para um bom ambiente devidamente iluminado e dispor elementos que permitam ao software entender o que deve ser capturado), foi então possível realizar o experimento prático.

Nesta fase, escolheu-se um dia de sol e uma sala devidamente iluminada para se realizar a sessão fotográfica. A captura de imagens levou em torno de 30 minutos, não incluindo o tempo de configuração de máquinas fotográficas. As máquinas utilizadas foram duas Sony idênticas, utilizando as mesmas configurações de abertura de lente e obturador, bem como todas as demais configurações. Ao todo, foram extraídas 83 imagens para o modelo de exemplo deste trabalho.

Como uma primeira experiência, foram feitas imagens em uma sala com uma pessoa servindo de modelo. Cada câmera Sony adquiriu dados de uma parte do modelo (uma focando a parte superior do tronco até a cabeça, a outra a parte inferior do tronco até os pés). Ambas as câmeras fizeram uma volta de 360 graus em torno do modelo em sentido horário. Segundo o fabricante do software 123DCatch®, quanto mais imagens de boa qualidade, melhor para o sistema calcular os dados tridimensionais (respeitando o limite máximo de até 70 fotos, conforme orientações do sistema para envio aos servidores da Autodesk). É



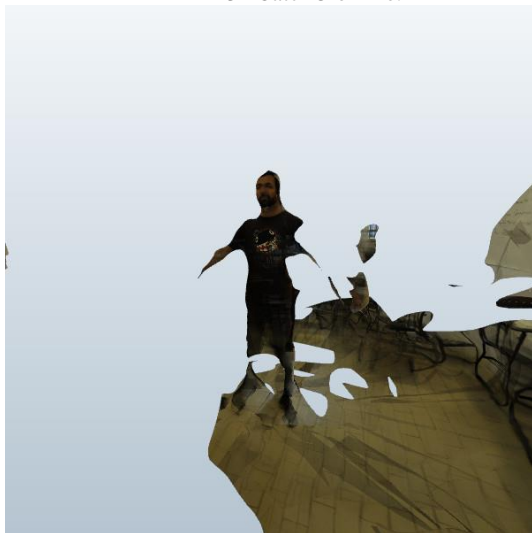
necessário ainda que o fundo da imagem a ser capturada não seja movimentado.

### 2.4.2. Processamento e criação de modelo 3D

Imediatamente após a aquisição de imagens pela fotogrametria, foi acessado o site do 123DCatch® para envio do material e início do processamento para gerar o modelo tridimensional. Como descrito anteriormente, após o envio dos arquivos das fotos, o sistema pede que seja confirmado o processamento das imagens 2D em um modelo 3D, sendo que logo após isto o usuário precisa estar conectado no sistema para finalizar a ação.

O processamento levou aproximadamente 15 minutos, utilizando uma conexão de internet de 5MB para download. O aplicativo liberou o modelo que pode ser visto na Figura 2. A ideia seria utilizar a fotogrametria para, ao mesmo tempo em que se captura o modelo, já se captura a textura. Infelizmente, os resultados não foram adequados à continuação do modelo (retopologia), mesmo realizando outros testes com configurações diferentes de cenário, modelo ou mesmo trabalhando com modelos do tipo miniaturas, para se averiguar se o tamanho do modelo poderia estar influenciado nos resultados inadequados. Infelizmente, para este caso com modelo vivo, foi necessária a troca de técnica pela de luz branca, com construção posterior de textura, utilizando técnicas de agilização de processos, diferente da ideia inicial.

Figura 2: Modelo digitalizado na interface do Autodesk 123DCatch® online.



Fonte: Autoria própria.

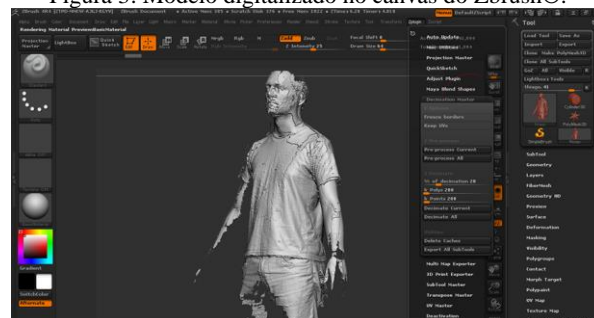
### 2.4.3. Retopologia

A realização da retopologia do modelo ocorreu por três motivos principais: otimizar a contagem de polígonos (visto que o modelo digitalizado possuía algo em torno de 8 milhões de polígonos); corrigir a sequência de *edge loops*, tornando o modelo funcional para a animação; editar a superfície do modelo.

Primeiramente o modelo foi importado no software Zbrush® (Figura 3) e em seguida foi utilizado o *plug-*

*in* Decimation Master para aperfeiçoar a malha. O modelo possuía 7.956.226 de faces inicialmente e depois de se aplicar este recurso, a contagem de polígonos diminuiu para 280.052 faces.

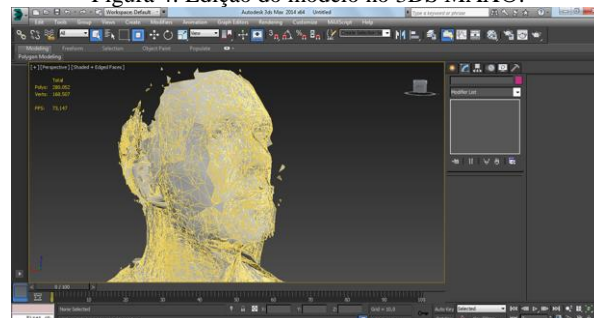
Figura 3: Modelo digitalizado no canvas do Zbrush®.



Fonte: Autoria própria.

A geometria foi exportada com a extensão *obj* (Wavefront 3D Object File), criada pela Wavefront Technologies, do Zbrush® para o 3DS MAX®, com o intuito de remover as superfícies duplas, inverter as normais do modelo, fechar as superfícies não conectadas e corrigir as coordenadas do modelo, como mostra a Figura 4.

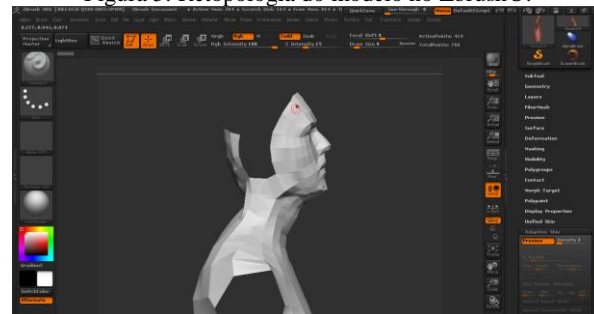
Figura 4: Edição do modelo no 3DS MAX®.



Fonte: Autoria própria.

O modelo tridimensional foi importado novamente no Zbrush® para realizar a retopologia (Figura 5), utilizando o recurso *retopology* que permite reconstruir a malha com base no modelo inicial.

Figura 5: Retopologia do modelo no Zbrush®.



Fonte: Autoria própria.

No 3DS MAX® foi feita a edição da superfície utilizando a ferramenta de extrusão para criar áreas que não foram digitalizadas, como a parte interna dos braços, mãos e topo da cabeça. Também foi espelhado

o modelo com o modificador *symmetry* para melhor visualizar o conjunto inteiro. O modelo finalizado possui 1450 polígonos, como pode ser visualizado na Figura 6.

Figura 6: Modelo reconstruído com 1450 polígonos.

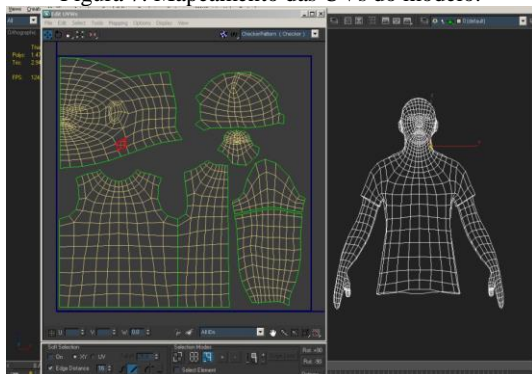


Fonte: Autoria própria.

#### 2.4.4. Mapeamento e texturização

Após a reconstrução, o modelo foi mapeado com o modificador *Unwrap UVW* (Figura 7) para que fosse então aplicadas as texturas de pele, roupas e cabelos.

Figura 7: Mapeamento das UVs do modelo.



Fonte: Autoria própria.

Para efetuar a texturização, utilizaram-se as fotografias criadas no momento da captura do modelo por luz branca, conforme a Figura 8.

Figura 8: Fotografias geradas na ocasião de captura de arquivo de dados do modelo 3D.

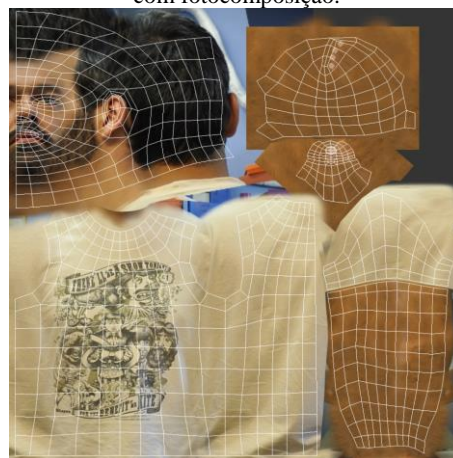


Fonte: Autoria própria.

A seguir, para efetuar a texturização, optou-se pela técnica de revestimento com fotocomposição (Figura 9), que é a aplicação de partes de imagens, extraídas de

fotografias, com aplicação direta ou manipulada no gabarito UV do modelo, conforme Fox [2003] explica em seu livro. Inicia-se este processo com tamanho de arquivo definido em 1024 x 1024 pixels.

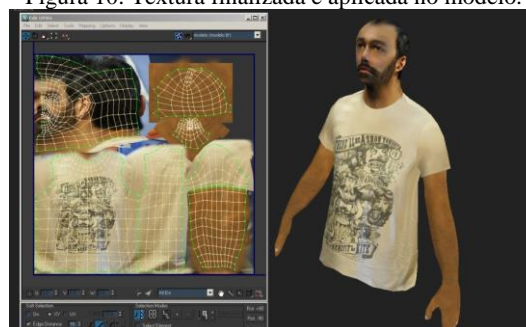
Figura 9: Texturização utilizando a técnica de revestimento com fotocomposição.



Fonte: Autoria própria.

Em seguida, a textura finalizada é aplicada ao modelo, com seu leiaute UV devidamente ajustado e a textura devidamente finalizada (Figura 10). O modelo pode então passar à próxima etapa, que é a de *rigging* seguido de *skinning*.

Figura 10: Textura finalizada e aplicada no modelo.

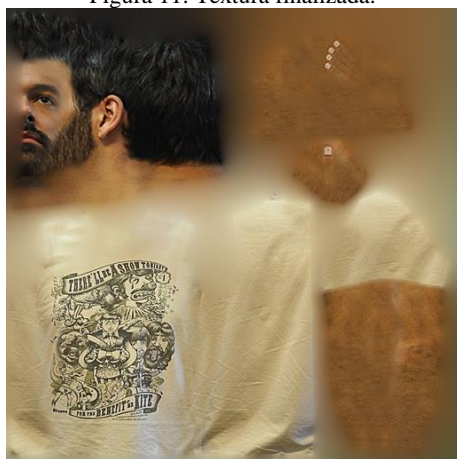


Fonte: Autoria própria.

Para finalizar a textura, a mesma foi reduzida para 512 x 512 pixels, a fim de melhorar o desempenho dentro de aplicativo tridimensional, ao mesmo tempo em que recebeu filtros de Adobe Photoshop para não perder os detalhes. Foi aplicado um filtro especial (*Solidify*), que expande da borda das ilhas de *clusters* (a região mapeada em si, que é uma representação bidimensional de superfícies tridimensionais) do mapeamento, até o limite do arquivo, o último pixel, cruzando com os vizinhos e por fim, mesclando todos. Isto é feito para evitar o efeito de “*mip-map*” que ocorre em *engines* de jogos, na qual se veem as emendas das texturas estando os modelos a certa distância da câmera. Prolongando o último pixel, isto deixa de acontecer e o modelo adquirir maior qualidade de renderização (Figura 11).



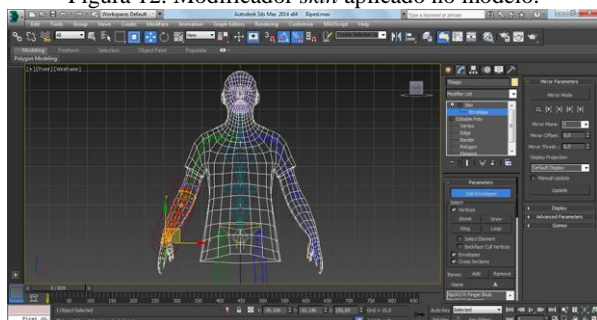
Figura 11: Textura finalizada.



Fonte: Autoria própria.

### 2.4.5. Esqueletização

Após a texturização, os modelos passaram pelo processo de *rigging* (que é o momento de se criar a estrutura óssea que controlará o personagem) utilizando uma estrutura de *biped*, que foi construído de acordo com a anatomia do modelo. Feito isso, foi aplicado o modificador *skin* (Figura 12), para criar uma relação da geometria com o esqueleto, e determinar o peso com que o *bone* influencia na deformação da malha.

Figura 12: Modificador *skin* aplicado no modelo.

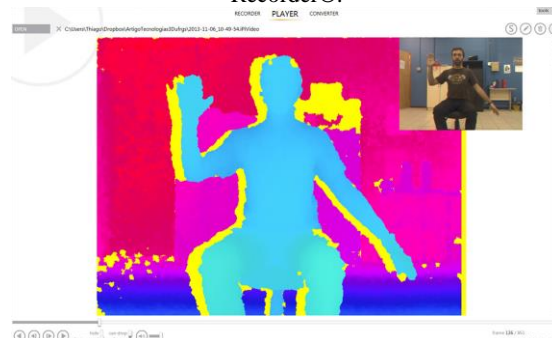
Fonte: Autoria própria.

### 2.4.6. Aquisição de dados para animação usando Kinect®

Tendo em vista que o espaço apropriado para capturar os movimentos do ator situa-se entre 1,2 a 3,5 metros, este projeto foi realizado a uma distância aproximada de dois metros do aparelho. Além disso, optou-se por realizar a captura do modelo na posição sentado. Ainda que isto reduza o número de articulações detectadas pelo Kinect® de 20 para 10 juntas, optou-se por utilizar este método devido à limitação de equipamentos, visto que para capturar o esqueleto em pé seriam necessários dois ou mais dispositivos. A gravação do vídeo foi feita com o programa iPi Recorder® (Figura 13), e o processamento da captura foi realizado no iPi Mocap Studio® que converteu o *mocap* em uma sequência de *keyframes*. Esta etapa de processamento durou algo em torno de seis horas. Na

sequência, a animação foi exportada para Character Studio do 3DS MAX®.

Figura 13: Captura de vídeos e volumes com o iPi Recorder®.

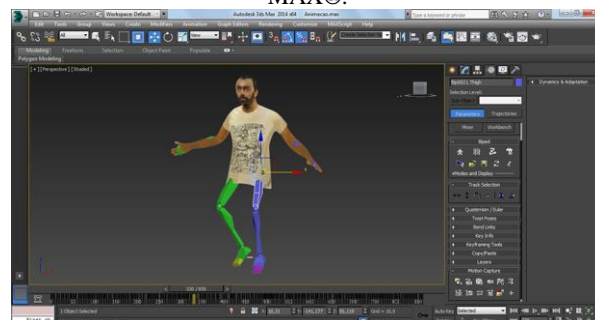


Fonte: Autoria própria.

### 2.4.7. Integração de modelo digitalizado com animação capturada

Posteriormente, no Character Studio do 3DS MAX® (Figura 14) utilizou-se o comando *Load capture* na aba *Motion* para inserir a captura no *biped*. Dessa forma o esqueleto carrega todas as informações de movimento da captura importada. Além disso, foi feita uma redução dos *keyframes* durante a importação do arquivo, já que a técnica de *mocap* é construída quadro a quadro. Convém ressaltar que foi feita a captura apenas dos movimentos do tronco e da cabeça, devido à limitação de instrumentos, uma vez que é necessária a utilização de, no mínimo, dois Kinets® para capturar corretamente todo o corpo. No Laboratório ViD, onde foram realizadas as sessões de captura de movimento, os alunos tiveram acesso somente a um Kinect®, que foi conectado a um notebook com os programas iPi Recorder® e iPi Mocap Studio® instalados.

Figura 14: Edição da animação no Character Studio do 3DS MAX®.



Fonte: Autoria própria.

### 2.4.8. Adaptação para Unity

Como forma de aplicação em tempo real do modelo, optou-se pelo uso do mesmo dentro do software de *engine* de jogo denominada Unity3D® Engine. Esta *engine* é uma das mais usadas hoje para desenvolvimentos de jogos de baixo custo, para nichos específicos de consumo (*Indie Games*), seu custo de aquisição também é relativamente baixo se comparado

a outras *engines* de jogos disponíveis no mercado, havendo ainda uma versão gratuita com recursos reduzidos para ela.

Nesta etapa se exportou do software 3D de modelagem o modelo e seu esqueleto de animação atrelado a sua malha tridimensional, juntamente com os movimentos animados vindos do iPi Studio. Uma vez que os movimentos vindos deste software haviam sido transformados em um formato compatível com a estrutura óssea utilizada (*Character Studio*), foi possível exportar em formato *FBX* (novo formato de conversação entre softwares 3D desenvolvido pela Autodesk) para a engine Unity3D®.

Dentro da Unity3D® Engine, precisou-se apenas atribuir o arquivo de textura ao modelo juntamente com seu material, pois a animação já vem atrelada ao modelo. Dentro da linguagem da Unity3D®, todo modelo em formato *FBX* que for exportado com animação, este já aparece no sistema de hierarquias da *engine* e em modo *Play*, os movimentos apresentam-se no modelo em tempo real. Embora simples, os movimentos foram funcionais de modo que se confirma o fato de haver boa integração entre as ferramentas para facilitar o processo de animação também para aplicações de tempo real, como as *engines* de jogos digitais.

### 3. Considerações Finais

A presente pesquisa se apresentou útil para mostrar que nem sempre o caminho mais curto pode ser o melhor para se construir modelos, mas que se bem executado e bem compreendidas as ferramentas usadas, com as devidas adaptações elas podem ser muito funcionais para a aquisição de dados. A pesquisa contemplou as etapas de modelagem, mapeamento de superfícies, texturização, captura de movimentos para animação e exportação integradora entre ferramentas 3D convencionais e *engine* de jogos.

Durante todo o processo foi esperado que as técnicas e ferramentas utilizadas realizassem a maior parte do trabalho, mas, em certas circunstâncias, elas não se mostraram efetivas. Pode-se dizer que o trabalho de construção de modelos de modo tradicional poderia ser o caminho mais curto, mesmo deixando para o designer que executa o modelo a responsabilidade de torná-lo idêntico e funcional ao seu conceito. Em contra partida, a aquisição de dados para animação foi muito interessante. Com mais estudos e testes acredita-se que poderia se chegar a resultados muito bons. Para um estudo posterior, poderia se construir um modelo 3D de modo tradicional (de imagens 2D para a confecção do modelo 3D em softwares convencionais como o 3DS MAX®, por exemplo) e confrontar seu tempo de produção com o tempo de produção realizado no modelo desta pesquisa.

Ao fim, pode-se perceber que o designer, parte humana dentro do processo de produção aqui estudado e experimentado, se mantém indispensável, pois sem ele o processo não continua e não se intercala entre as

diferentes ferramentas. Mesmo com ferramentas que prometem resultados rápidos e maravilhosos, muitas vezes a custos baixos ou altos, o designer precisa atuar para que o modelo seja criado de modo coerente para o projeto a qual será utilizado faça parte.

### Agradecimentos

Os autores agradecem o Laboratório de Design e Seleção de Materiais – LdSM, e o Laboratório de Virtual Design – ViD, da Universidade Federal do Rio Grande do Sul, UFRGS.

### Referências

- AZEVEDO, Eduardo. Computação Gráfica I - Teoria e Prática. Rio de Janeiro: Elsevier, 2003, 5ª reimpressão.
- BATES, Bob. Game Design. 2. ed. Boston: Cengage Learning PTR, 2004.
- CATCH. Disponível em: <http://apps.123dapp.com/catch/> Acesso em 11 dez. 2013.
- CHANDLER, Heather Maxwell. Manual de Produção de Jogos Digitais. 2. ed. Porto Alegre: Bookman, 2012.
- CHONG, Andrew. Animação digital. Porto Alegre: Bookman, 2011.
- DENIS, Sébastien. O cinema de animação. Lisboa: Edições Texto & Grafia, 2007.
- FOX, Barret. Animação Com 3DS MAX® 6. Rio de Janeiro: Ciência Moderna, 2003.
- GAMASUTRA. Disponível em: [http://www.gamasutra.com/view/news/122183/LA\\_Noire\\_s\\_Pioneering\\_Motion\\_Capture\\_Draws\\_Line\\_In\\_The\\_Sand\\_For\\_Animation.php](http://www.gamasutra.com/view/news/122183/LA_Noire_s_Pioneering_Motion_Capture_Draws_Line_In_The_Sand_For_Animation.php). Acesso em 05 de mai. 2014.
- GOMIDE, João Victor Boechat. FLAM, David Lunardi. PACHECO, Daniel. ARAÚJO, Arnaldo de Albuquerque. Captura de Movimento e Animação de Personagens em Jogos. In: SIMPÓSIO BRASILEIRO DE GAMES, 2009, Rio de Janeiro.
- IMDB. Disponível em: <http://www.imdb.com/title/tt0173840/>. Acesso em 10 dez. 2013.
- KUHN, Giovane Roslindo; GOMES, Paulo César Rodacki. Animação de um Personagem Virtual Utilizando Captura Óptica de Movimento com Marcações Especiais. Anais do IX Seminfo (JIT3C), 2005.
- MICROSOFT. Kinect® for Windows. Disponível em: <http://www.microsoft.com/en-us/Kinect@forwindowsdev/default.aspx>. Acesso em: 9 fev. 2014.
- NOVAK, Jeannie. Desenvolvimento de Games. 2. ed. São Paulo: Cengage Learning, 2010.
- PURVES, Barry. Stop-motion. Porto Alegre: Bookman, 2011.
- SHOTTON, Jamie. FITZGIBBON, Andrew. COOK, Mat. SHARP, Toby. FINOCCHIO, Mark. Moore, Richard. KIPMAN, Alex. BLAKE, Andrew. Real-time human pose recognition in parts from single depth images. In: CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. 2011, Colorado.
- THOMPSON, Jim. BERBANK-GREEN, Barnaby. CUSWORTH, Nic. Videojuegos: Manuela para diseñadores gráficos. Barcelona: Editorial Gustavo Gili, 2008.
- WARD, Antony. Game Character Development. EUA: Cengage, 2008.