

***Framework* para geração de personagens
humanóides 3D para jogos
computacionais**

TRABALHO DE GRADUAÇÃO

Novembro de 2009

Framework para geração de personagens humanóides 3D para jogos computacionais

por

João da Rocha Pascoal Neto

Monografia apresentada ao Centro de Informática
da Universidade Federal de Pernambuco como
requisito parcial para obtenção do Grau de
Bacharel em Ciência da Computação.

Orientador: Sílvio de Barros Melo

Novembro de 2009

“A vingança nunca é plena, mata a alma e a envenena.”
(Ramón Valdez)

Agradecimentos

Em primeiro lugar agradeço aos meus pais, Ricardo e Roberta, por chegar até aqui. Pelo apoio incondicional, pelo incentivo jamais contestado e por todo o carinho demonstrado apesar das noites mal dormidas e dos desgastes gerados pela vida acadêmica.

Agradeço aos meus tios Hermes e Mônica, pela chance que me deram, pela total confiança depositada numa pessoa que mal conheciam e por todo o carinho dado. Sem eles dificilmente eu estaria aqui.

A Elayne, minha namorada, que me acompanhou durante toda essa reta final, me ajudou e me incentivou a continuar mesmo nos momentos mais complicados. Foi graças ao convite dela para participar de uma equipe de jogos que comecei minha trajetória e cheguei até aqui.

A Ricardo Lessa, que me deu minha primeira oportunidade na vida profissional e apostou no meu potencial. Graças a ele, dei o meu primeiro passo na carreira de programador na Educandus, onde cultivei grandes amizades lá dentro.

A Álvaro, Luis Henrique, Valeska, Priscilla e Cynthia, minha gratidão por todo o tempo de convivência na Educandus e pela amizade que sei que levarei comigo para o resto da vida.

A Maurício Carvalho e a Paulo Souza da Playlore, pela grande oportunidade dada. Sem eles não estaria desenvolvendo este projeto. Juntamente com eles, meus agradecimentos a Antônio, Neto e Leozito pela paciência e pela ajuda prestada em todo o tempo que trabalhamos juntos e para Bruce, Leandro, Rael, Makako, Gustavo, Totonho, Juliano, Dalton, Shushu entre outros que testaram, sugeriram, opinaram ou colaboraram de alguma maneira para que o projeto se desenvolvesse mais e com maior qualidade. Além de grandes profissionais, são excelentes pessoas e ótimos amigos.

Um agradecimento mais do que especial ao pessoal do China 48: Mário, Formiga, Apebão, Peter, JP, Lhama e Riffa. Amigos de coração e pessoas de grande qualidade que vivenciaram comigo vários momentos da vida acadêmica e na amizade que todos nós compartilhamos.

E finalmente, a Sílvia Melo por toda a ajuda e disposição dada nesse trabalho.

Resumo

O mercado de jogos computacionais está cada vez mais exigente em relação à qualidade. Cenários realistas e personagens com características comportamentais diferenciadas estão em alta, o que aumenta drasticamente a quantidade de trabalho para o artista 3D. Em jogos que possuem dezenas de personagens distintos, a construção dos mesmos pelos artistas leva muito tempo pelo alto nível de complexidade o que, para as empresas do ramo, acarreta em um prazo maior ou mesmo um adiamento para a entrega do produto final. Esse trabalho tem como objetivo a construção de um framework para criação e exportação de personagens em arquivo Collada para que os mesmos sejam importados para games computacionais.

Palavras-chave: Jogos, Modelos 3D, Personagens Humanóides, Criação de Personagens, Exportação de Personagens, Modelagem, Texturização, Interpolação Parametrizada.

Sumário

1. INTRODUÇÃO	8
1.1 MERCADO DE JOGOS	8
1.2 MOTIVAÇÃO.....	10
1.3 OBJETIVO E ESTRUTURA GERAL.....	10
2. CRIAÇÃO DE PERSONAGENS 3D	12
2.1 MODELAGEM.....	12
2.1.1 Construção por Geometria Sólida	13
2.1.2 NURBS.....	14
2.1.3 Modelagem Poligonal.....	15
2.2 ANIMAÇÃO.....	16
2.2.1 Keyframing	16
2.2.2 Rotoscopia.....	17
2.2.3 Motion Capture	18
2.3 TEXTURIZAÇÃO.....	19
2.3.1 Color Mapping	20
2.3.2 Bump Mapping	21
2.3.3 Normal Mapping.....	22
2.3.4 Displacement Mapping	23
2.3.5 Lightmapping	24
3. GERAÇÃO AUTOMÁTICA DE PERSONAGENS 3D.....	25
3.1 GERAÇÃO POR PRESETS	25
3.2 EXTRAÇÃO AUTOMÁTICA POR FOTOGRAFIA.....	26
3.3 MANIPULAÇÃO DE BONES.....	28
3.4 INTERPOLAÇÃO POR PARAMETRIZAÇÃO	30
4. GCM – GAME CHARACTER MAKER	32
4.1 FACEMAKER.....	32
4.1.1 Edição de geometria.....	33
4.1.2 Encaixe de textura.....	34
4.1.3 Filtragem da textura.....	36
4.2 BODYMAKER.....	39
4.3 TORQUE GAME ENGINE	39
5. AVALIAÇÃO E RESULTADOS	41
5.1 AVALIAÇÃO DE DESEMPENHO	41
5.2 CONSIDERAÇÕES SOBRE O ENCAIXE DE TEXTURA POR PONTOS DE CONTROLE	41
5.3 JUSTIFICATIVA PARA TRATAMENTO DE IMAGEM NO FRAMEWORK.....	42
5.4 INSERÇÃO DO MODELO GERADO EM UM JOGO EXEMPLO.....	42
6. CONSIDERAÇÕES FINAIS.....	44
7. REFERÊNCIAS	46

Lista de Figuras

Figura 1. Crescimento de faturamento e vendas no mercado de games	8
Figura 2. Distribuição de jogadores de console / PC.....	9
Figura 3. Exemplo de modelagem com CSG.....	13
Figura 4. Superfície construída com NURBS.....	14
Figura 5. Malha da cabeça um personagem 3D.....	15
Figura 6. Animação de um modelo através de rotosopia.....	18
Figura 7. Animação de um modelo através de Motion Capture	19
Figura 8. Cubo sofrendo aplicação de Color Mapping.....	20
Figura 9. Uma laranja, seu respectivo Bump Map e o resultado final	21
Figura 10. Bump Map (à esquerda) e Normal Map	22
Figura 11. A malha original (cima), o Displacement Map (meio) e o resultado final..	23
Figura 12. Lightmap utilizado na iluminação de uma sala.....	24
Figura 13. Gerador de personagem de Lineage II com aplicação de presets	26
Figura 14. Demonstração do AvMarker. Foto (esquerda) e a cabeça gerada.	28
Figura 15. Gerador de personagem de TopSpin 3 com manipulação de bones	29
Figura 16. Gerador de personagem de Oblivion com interpolação paramétrica.....	31
Figura 17. Modo de Edição de Geometria	34
Figura 18. Modo de Encaixe de Textura	36
Figura 19. Modo de Filtragem – Redução de Brilho e Sombreamento Excessivo.....	37
Figura 20. Modo de Filtragem – Imagem Normal	38
Figura 21. Modo de Filtragem – Rosto Recortado e Esticado para Encaixe.....	38

1. Introdução

1.1 Mercado de jogos

Os jogos computacionais estão em alta no cenário nacional e mundial. Apesar de crise, o mercado de games continua sólido e, com o advento de novas tecnologias para hardware e software, a expectativa é que os futuros lançamentos levem aos consumidores inovações tanto na jogabilidade quanto na qualidade gráfica dos cenários e dos personagens.

A ESA, Entertainment Software Association, divulga anualmente uma pesquisa voltada ao crescimento e ao desenvolvimento de softwares de entretenimento chamada ESA Facts. Segundo a ESA Facts 2009, o mercado americano, um dos mais afetados pela crise mundial, obteve um crescimento de 22% nas vendas com um total de mais de 11 bilhões de dólares faturados e mais de 200 milhões de unidades vendidas para consoles ou PC's [ESA09].

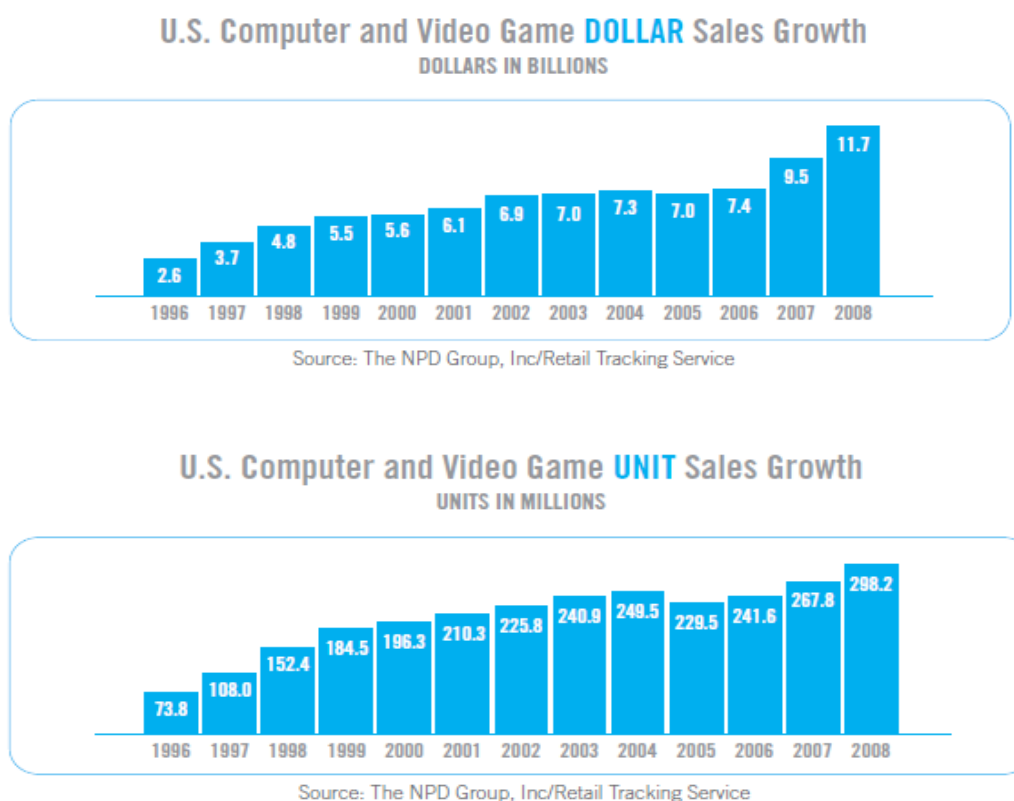


Figura 1. Crescimento de faturamento e vendas no mercado de games

Também segundo a ESA Facts, os games possuem adeptos entre quase todas as faixas etárias e os dois gêneros. 25% dos jogadores estão acima dos 50 anos e 43% dos jogadores online são mulheres. Outra característica marcante apresentada é que a média tempo de vida de um jogador é de 12 anos [ESA09].

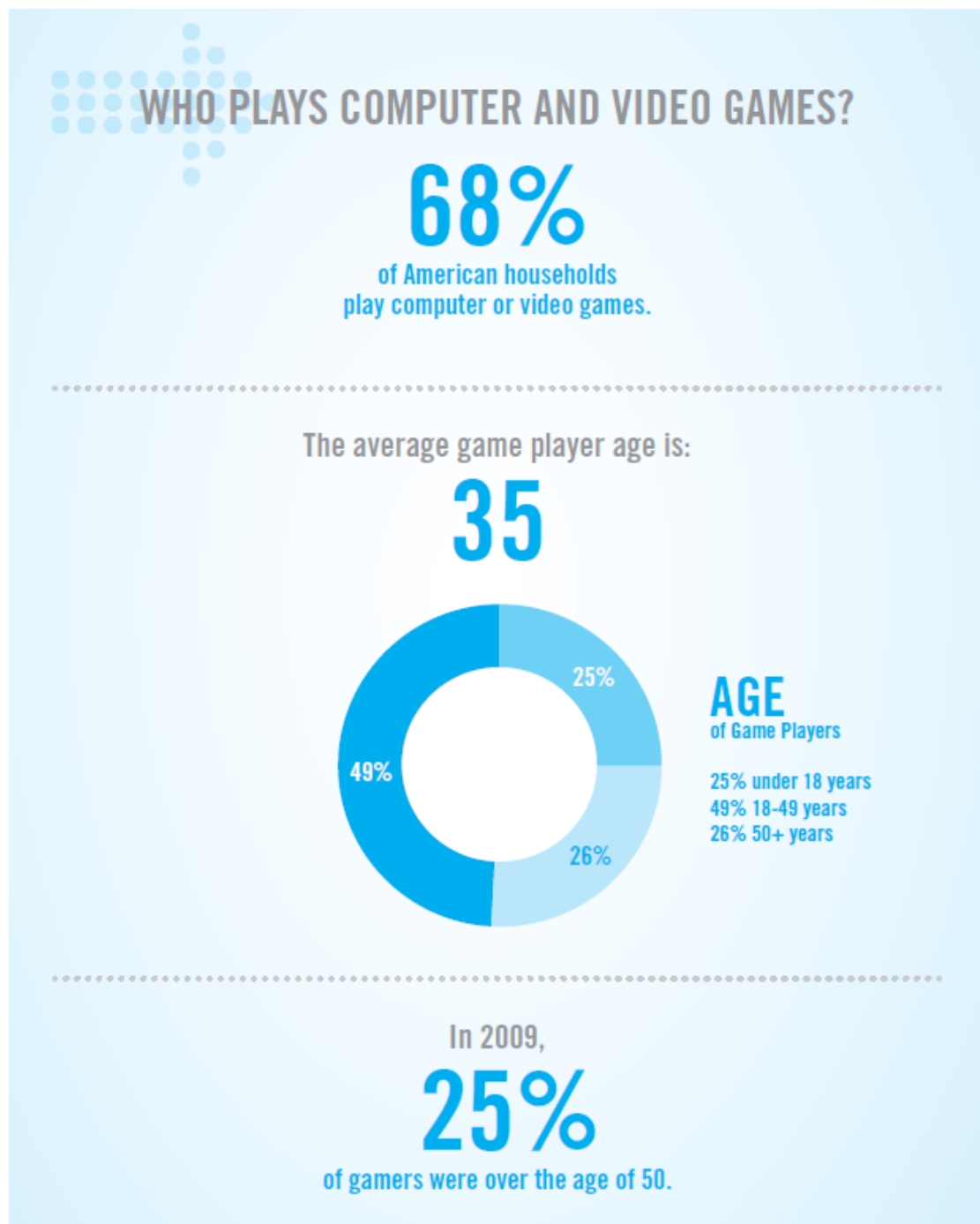


Figura 2. Distribuição de jogadores de console / PC

1.2 Motivação

Com o advento dos jogos 3D no cotidiano dos *gamers* há alguns anos atrás, a cobrança por novidades também cresceu. Para o *gameplay* surgiram novos sistemas de combate dinâmicos, gerenciamento complexo de grupos, sistemas de cooperação em tempo real, novas interações musicais, maior interferência do ambiente na jogabilidade, entre outras.

O visual dos jogos, com a rápida evolução das placas gráficas, foi elevada a uma nova dimensão. Iluminação e sombreamento em tempo real, aplicação de *shaders* para renderização de efeitos mais complexos, personagens com alto grau de realismo são apenas uma pequena parcela do quão complexos e elaborados estão os games.

1.3 Objetivo e estrutura geral

O foco desse trabalho está voltado para a criação de personagens humanos realistas para jogos computacionais. Esse tipo de personagem está presente em muitos títulos conhecidos atualmente de grandes empresas e aplicado nos mais diversos tipos como aventura, ação, esportes, FPS, RPG e etc.

A criação de um personagem varia de acordo com o nível de visual proposto pelo jogo como também pelo papel que ele representa, ou seja, quanto maior for a sua importância no enredo do jogo, pode-se levar mais tempo na construção. Em um jogo com gráficos mais modestos é possível criar um personagem em uma semana de trabalho porém esse tempo pode ficar entre 3 a 5 vezes maior dependendo da complexidade das texturas e próprio personagem em si. Quando os produtos são baseados em filmes ou famosos do mundo real, os detalhes são ainda mais exigidos e qualquer falta percebida pode torná-lo suscetível a críticas, acarretando mais tempo na construção. Como um jogo completo, com um tempo razoável de desenvolvimento, contém dezenas de personagens, o tempo gasto para a produção de todo o pipeline pode levar meses.

Com estas orientações em mente, este estudo oferece uma abordagem construída para agilizar a criação de personagens humanos através de barras de controle que gerenciam os vários parâmetros que constituem um corpo humano como altura, largura, formato da cabeça entre outros. A partir dos parâmetros modificados pelo usuário, é exportado um modelo no formato Collada, que pode ser utilizado e posteriormente modificado nos mais conhecidos softwares de modelagem do mercado como Maya e 3DStudio e importado para qualquer ambiente de jogo. Esse trabalho ainda provê uma ferramenta chamada GCM que implementa as técnicas propostas nesta pesquisa.

No capítulo 2 é descrito o processo tradicional de criação de um modelo 3D para jogos e seus principais passos. No capítulo 3 são descritas as técnicas utilizadas pelos games para a geração automática de personagens a partir do usuário. No capítulo 4, é descrito o produto gerado pela pesquisa e as técnicas utilizadas. No capítulo 5 são descritas as conclusões da pesquisa bem como futuras melhorias para o framework.

2. Criação de Personagens 3D

O processo de criação de um personagem para jogos exige conhecimento e tempo. É necessário conhecimentos em modelagem, animação, iluminação e animação além de noções em anatomia, perspectiva e pintura [Rosa]. Ainda segundo Rosa, para executar a criação também é necessário domínio em software's especializados nesse tipo de atividade como 3DMax, Maya ou Blender, que são os mais conhecidos no mercado [Rosa].

Existem várias etapas para a criação de um personagem tridimensional assim como diferentes técnicas para cada tarefa. Nesta pesquisa, são citadas as três principais etapas de criação, de forma resumida, e suas técnicas mais conhecidas. As descrições dos métodos de modelagem e dos softwares tem como idéia desenvolver comparações de tempo de construção e qualidade dos modelos com o produto gerado por esta pesquisa. Essa comparações serão descritas e informadas com mais detalhes no capítulo 4.

2.1 Modelagem

A modelagem é a etapa da construção do objeto real no mundo virtual [Cal06]. Um modelo 3D é uma representação de um objeto tridimensional, do mundo real ou não, geralmente utilizando uma coleção de pontos, ou vertices, no espaço 3D. Esses pontos são conectados através de polígonos onde os mais tipos mais utilizados são os triângulos e os retângulos.

Para dar forma a um objeto desejado, é preciso começar de uma forma inicial e, de preferência, simples, para, em seguida, estendê-la com mais polígonos, modelando-a na definição desejada.

Existem vários tipos de classificação de modelagem 3D tanto no meio acadêmico quanto fora dele. A seguir, são descritas as principais técnicas segundo Calife, Coimbra e Toscano [Cal06]

2.1.1 Construção por Geometria Sólida

Conhecida como CSG (Construtive Solid Geometry, em inglês), essa técnica é baseada na utilização de poliedros básicos tais como cubos, esferas e pirâmides para, em seguida, serem aplicadas operações sobre eles como união, interseção, adição ou subtração.

A vantagem dessa técnica é a preservação dos modelos originais que compõem a estrutura final e todas as operações realizadas sobre os poliedros iniciais livrando a estrutura final de qualquer ambiguidade [Fog07]. Em contrapartida é difícil seu reuso, pois a geometria final não é armazenada, tornando difícil seu acesso pelos softwares de modelagem mais comuns. [Miya08]

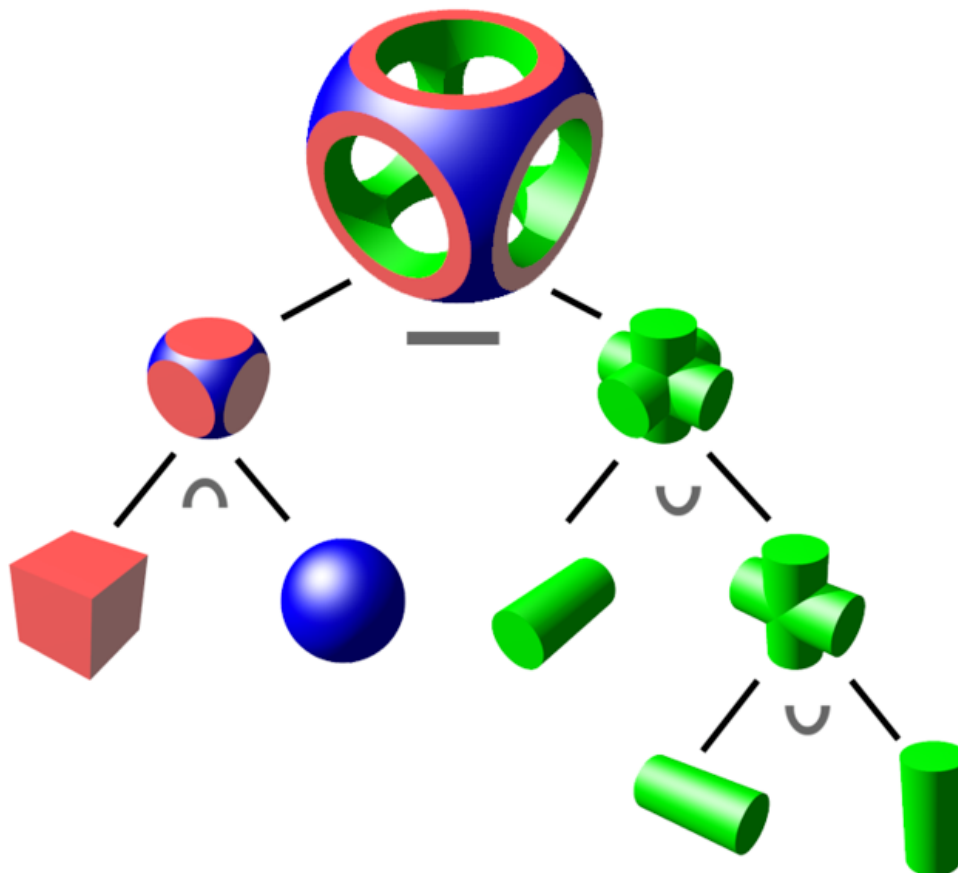


Figura 3. Exemplo de modelagem com CSG

2.1.2 NURBS

Do inglês, *Nonuniform rational B-Spline*, a modelagem NURBS tem como base a descrição matemática de superfícies através de pontos de controle. Segundo Minetto, NURBS possui a liberdade de representar tanto formas analíticas quanto formas livres como carros, armários entre outros [Min03].

A vantagem dessa técnica é a facilidade de representação de superfícies com grande qualidade. O grau de qualidade atingido por essa técnica é muito elevado e, por consequência, utilizado em vários segmentos da indústria de produtos e no cinema [Min03]. Como desvantagem, é pouco utilizado nos games, pelo simples fato de que os algoritmos de renderização para jogos tridimensionais implementados nas placas de vídeo usem como base a geometria dos modelos, que não é explorada pelos NURBS.

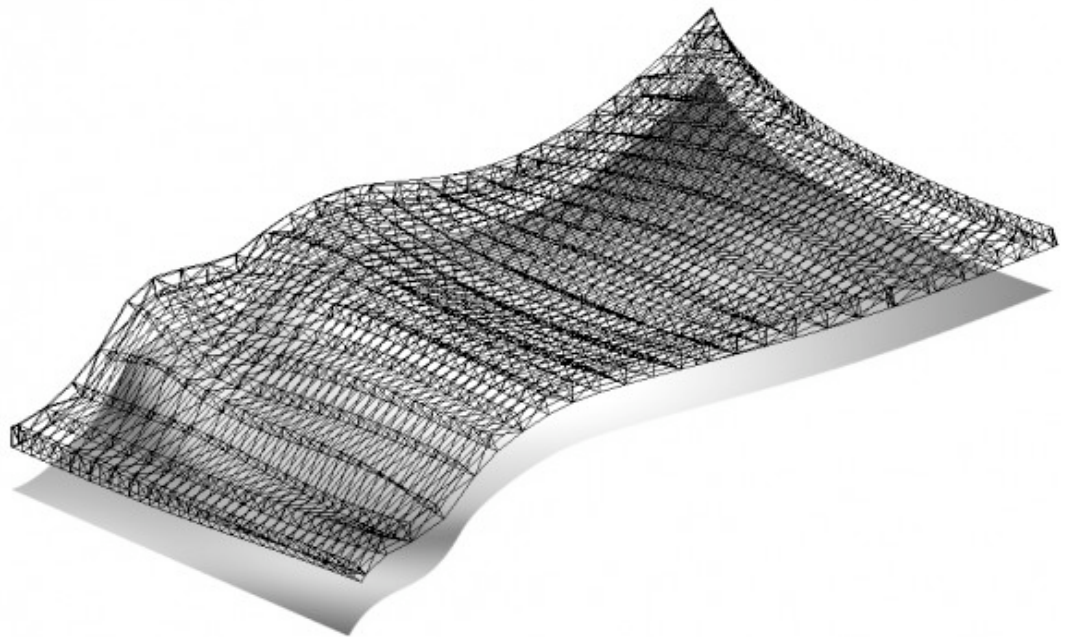


Figura 4. Superfície construída com NURBS

2.1.3 Modelagem Poligonal

A modelagem poligonal representa uma ou mais superfícies através de uma malha de polígonos [Cal06] normalmente quadriláteros ou triângulos. Um objeto 3D possui uma ou mais malhas (*mesh*, em inglês) e cada malha representa uma parte desse objeto. Em alguns casos, a malha pode ser todo o objeto.

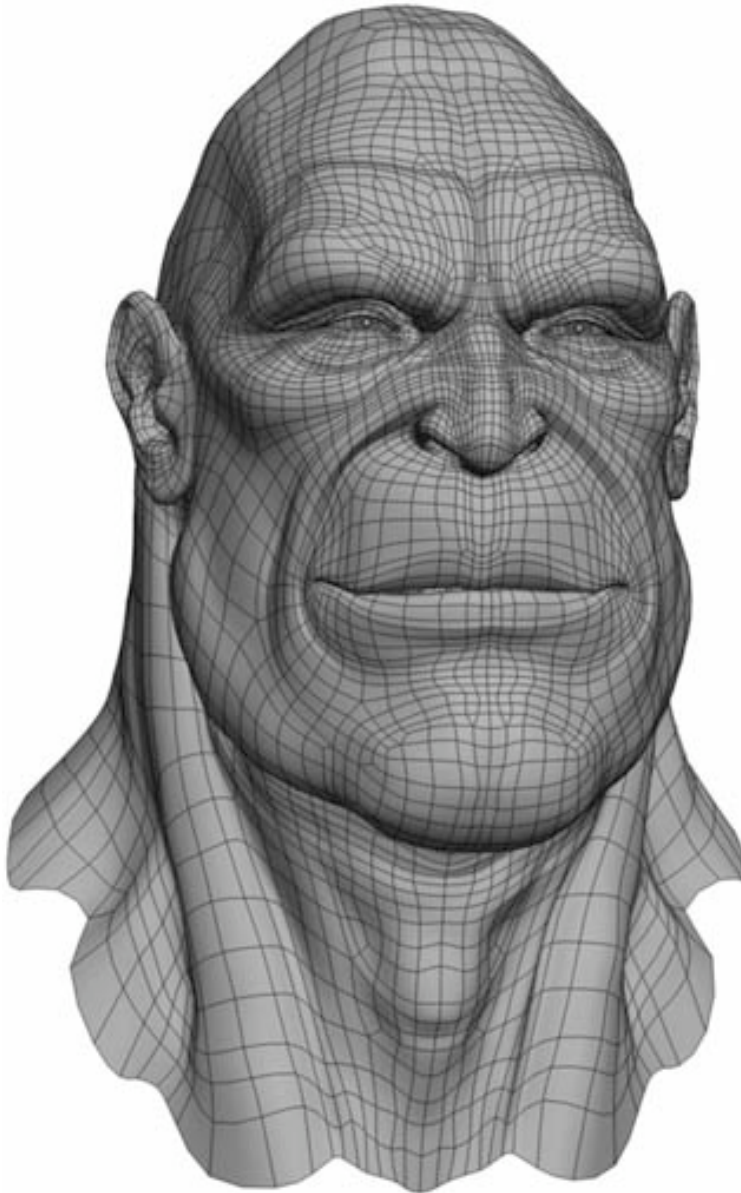


Figura 5. Malha da cabeça um personagem 3D

As informações contidas em um modelo com baseado em modelagem poligonal são: os vértices que compõem todo o espaço tridimensional do modelo e as n-tuplas que representam os polígonos de n lados, onde cada elemento da tupla é

o índice de um vértice do espaço 3D. Normalmente as tuplas possuem tamanho 3 ou 4, indicando a constante frequência de triângulos ou quadrados na formação das malhas.

Com essa técnica é possível ter um maior controle sobre a qualidade dos modelos gerados através da quantidade de polígonos usada na construção. Com mais polígonos, é possível representar superfícies mais elaboradas, com mais curvas e detalhes, gerando um modelo de ótima qualidade [Sgarbi]. Em compensação, o custo computacional é maior, acarretando mais tempo na renderização do modelo e na carga em tempo de execução de um jogo.

Essa técnica é a mais utilizada na geração de cenários e personagens para jogos e, por isso, os modelos devem ter um cuidado especial na sua construção, pois devem ter uma boa qualidade sem gerar um alto custo computacional para o usuário [Bat00][Clua05]. Para isso são utilizadas várias técnicas de otimização e texturização para aparentar uma maior qualidade. Mais detalhes sobre texturização na seção 2.3.

2.2 Animação

A animação consiste em aplicar movimentos a um objeto tridimensional. Ao término de uma modelagem, de qualquer um dos tipos citados acima, o usuário terá um modelo estático, sem qualquer movimento e as técnicas de animação são utilizadas para dar a sensação de movimento e de maior interação do modelo. As técnicas descritas abaixo estão de acordo com Battaiola[Bat00].

2.2.1 *Keyframing*

A primeira técnica de animação utilizada na história dos games foi o *keyframing*, ou animação por quadros-chave. Essa técnica surgiu com os desenhos

animados, sendo adaptada para jogos em 2D e atualmente é utilizada também nos jogos tridimensionais.

O keyframing consiste em uma animação feita a partir de keyframes (quadros-chave em inglês), onde cada quadro contém uma posição-chave necessária para a execução da animação[Louro]. Uma vez determinadas todas as poses do personagem para animação, é feita uma interpolação entre os quadros para o modelo transmitir a sensação de movimento. É possível manipular cada keyframe em separado, modificando iluminação, posicionamento e rotação, entre outros fatores, e cada mudança feita em qualquer um dos quadros modificará toda a execução da animação.

Apesar da simplicidade e do potencial dessa técnica, segundo Silva e Cavalcanti [Silva96], o keyframing torna-se um problema quando se deseja representar com perfeição movimentos complexos do corpo humano. Segundo Battaiola *et al*, o animador necessitaria de um conhecimento muito grande sobre a movimentação e toda a anatomia do processo e, ainda assim, o resultado pode não sair como o desejado [Bat00].

2.2.2 Rotoscopia

A rotoscopia é uma técnica de animação onde usamos uma situação criada no mundo real como referência. Ao invés de aplicar diretamente em cada quadro cada pose do movimento, a rotoscopia usa como referência um vídeo contendo uma filmagem real da animação desejada. Desse vídeo são extraídos vários quadros em intervalos regulares onde o animador pode usá-los de referência para desenvolver sua animação e aplicar outras técnicas como o keyframing[Bat00].

Essa técnica é melhor aplicada em animações de personagens humanos quando se deseja representar movimentos mais elaborados, difíceis de prever utilizando apenas o keyframing.

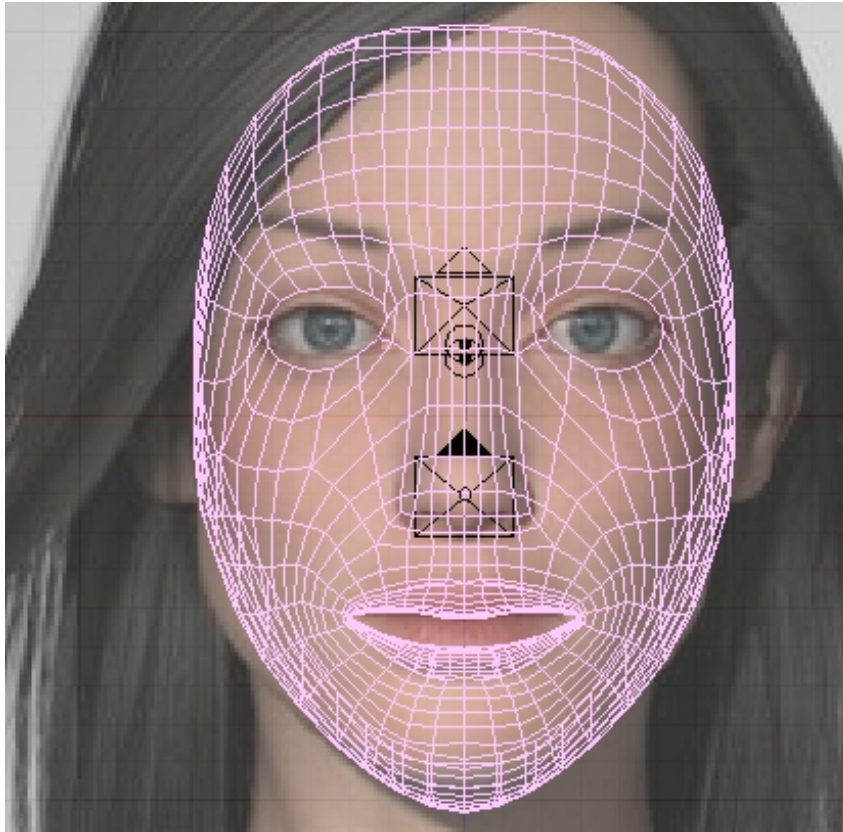


Figura 6. Animação de um modelo através de rotosopia

2.2.3 Motion Capture

Motion Capture é uma técnica que teve sua origem relacionada ao cinema, segundo Silva e Cavalcanti [Silva96]. Essa técnica moderna captura movimentos com alto grau de realismo, através do uso de vários aparelhos eletrônicos e eletromagnéticos.

O MoCap (abreviação de Motion Capture) utiliza um sistema de câmeras para filmar uma pessoa executando o movimento desejado para animar o personagem. A pessoa filmada por esse sistema utiliza uma roupa especial, contendo vários sensores eletromagnéticos que são acoplados em várias partes do seu corpo. A função desses sensores é mapear, de maneira automática, toda a movimentação capturada pelas câmeras em um modelo 3D previamente produzido [Bat00]. Além de câmeras e sensores, segundo Battaiola *et al*, outros equipamentos podem ser

utilizados na execução dessa técnica como aparelhos em infravermelho e sincronizadores [Bat00].

A vantagem dessa técnica é a alta qualidade da animação produzida, podendo reproduzir com fidelidade, movimentos mais suaves e naturais, que dificilmente seriam produzidos com roscopia e keyframing. Em contrapartida, o custo pra produzir animações com MoCap é muito alto em comparação com as demais técnicas em função da aquisição dos dispositivos necessários como câmeras e sensores.

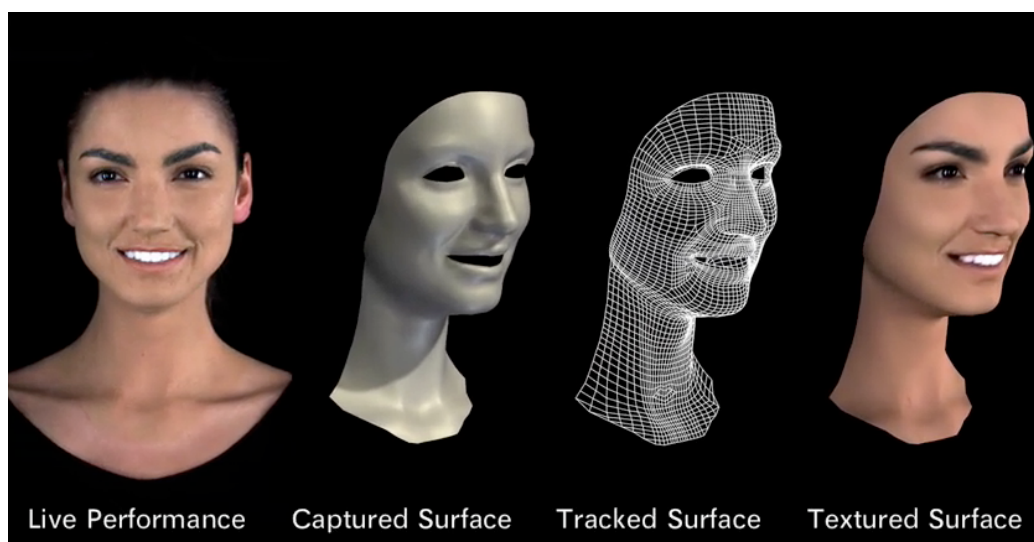


Figura 7. Animação de um modelo através de Motion Capture

2.3 Texturização

A texturização é uma técnica desenvolvida para dar ao modelo 3D uma nova visualização, fazendo-o parecer mais próximo do real para o usuário [Pozzer07]. A unidade básica do processo de texturização é a textura que é uma imagem bidimensional aplicada sobre um polígono pertencente a um objeto 3D [Evan07]. Embora a maioria das texturas utilizadas sejam 2D, existem também as texturas em 1D e em 3D.

A textura, quando aplicada sobre o modelo, em conjunto com a iluminação torna o modelo tridimensional mais realista e com maior aceitação para o usuário.

Todos os jogos fazem uso de texturas para caracterizar não apenas seus personagens, como também os cenários, itens e etc.

Nos jogos mais atuais, as texturas não são aplicadas isoladamente sobre os modelos 3D para replicar a informação binária sobre as faces. Outras texturas com diferentes aplicações também podem ser utilizadas sobre o mesmo modelo, agregando novas características à visualização final para torná-lo mais realista. A seguir são descritas algumas das aplicações mais difundidas para as texturas no processo de criação de personagens.

2.3.1 Color Mapping

O Color Mapping utiliza a textura para modificar a propriedade dos materiais, modificando a cor de cada polígono do modelo 3D [Camozz07]. É a aplicação mais básica da texturização, onde a imagem é replicada diretamente sobre a superfície do objeto, dando ao mesmo, a aparência da imagem contida na textura. Também é conhecida como Material Mapping ou Diffuse Mapping.

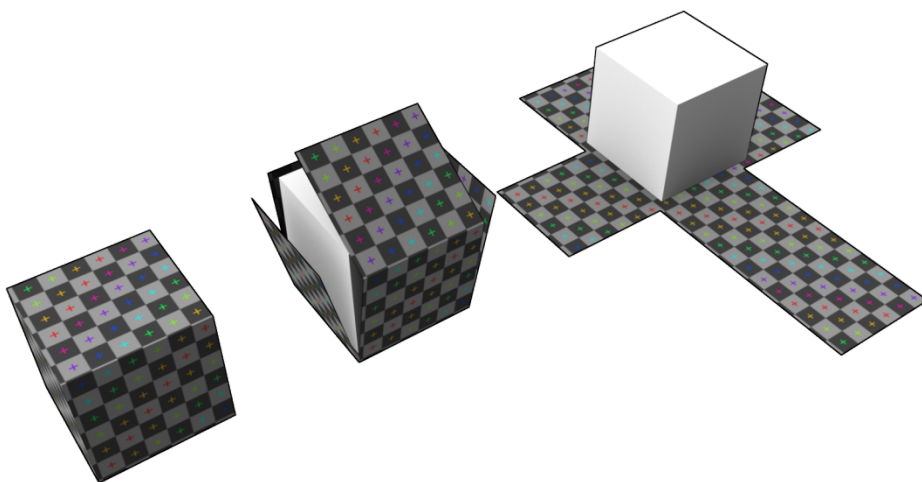


Figura 8. Cubo sofrendo aplicação de Color Mapping

2.3.2 Bump Mapping

Para um usuário normal que não está relacionado com a área, o *bump mapping* pode perfeitamente passar invisível por seus olhos. Essa técnica não altera diretamente a cor do ponto da superfície mas, em conjunto com o Color Mapping e outras técnicas de texturização e iluminação, o Bump Mapping adiciona um maior realismo ao objeto sem necessitar de modificações em sua geometria.

O Bump Mapping modifica a direção dos vetores normais dos vértices do objeto [Pozzer07]. Como os vetores normais são utilizados na iluminação dos objetos, então pode-se afirmar que o Bump Mapping modifica a reflexão dos raios de luz na superfície do objeto 3D.

Essa técnica é aplicada graças ao bump map (mapa de perturbação, in inglês), uma textura em preto e branco que também é mapeada sobre o objeto. Porém as cores dos pixels da textura determinam a perturbação que cada vetor normal do vértice da superfície irá sofrer na renderização [Pozzer07]. Quanto mais escuro, maior é a perturbação sofrida.

A grande vantagem dessa técnica é a fácil criação de rugosidades no objeto sem precisar incrementar a quantidade de polígonos no próprio objeto havendo, assim, um ganho de performance apesar da maior qualidade aparente.

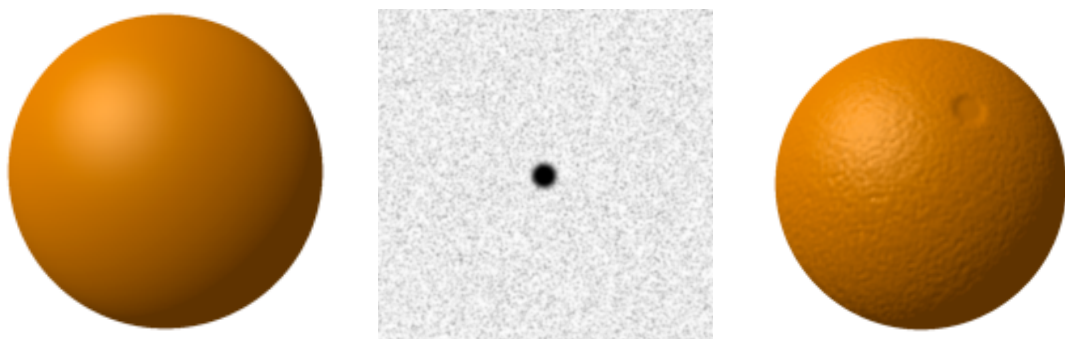


Figura 9. Uma laranja, seu respectivo Bump Map e o resultado final

2.3.3 Normal Mapping

O normal mapping também é uma técnica que cria irregularidades na superfície do objeto. Porém, ao contrario do bump map, a textura utilizada no processo, conhecido como normal map, não é em tons de cinza mas em um mapa de RGB [Camozz07].

A vantagem do normal map em relação ao bump map é a eficiência na aquisição das informações. Enquanto o bump mapping calcula a normal através de uma curva de nível baseada no tom de cinza do pixel, o normal map adquire a informação diretamente na textura sem nenhum cálculo adicional, pois o canal RGB é mapeado diretamente no espaço XYZ [Evan07].

O normal mapping é capaz de gerar as saliências com mais suavidade e qualidade pois, enquanto existem 256 variações de tons de cinza, ao todo são mais de 16 milhões de variações de RGB.

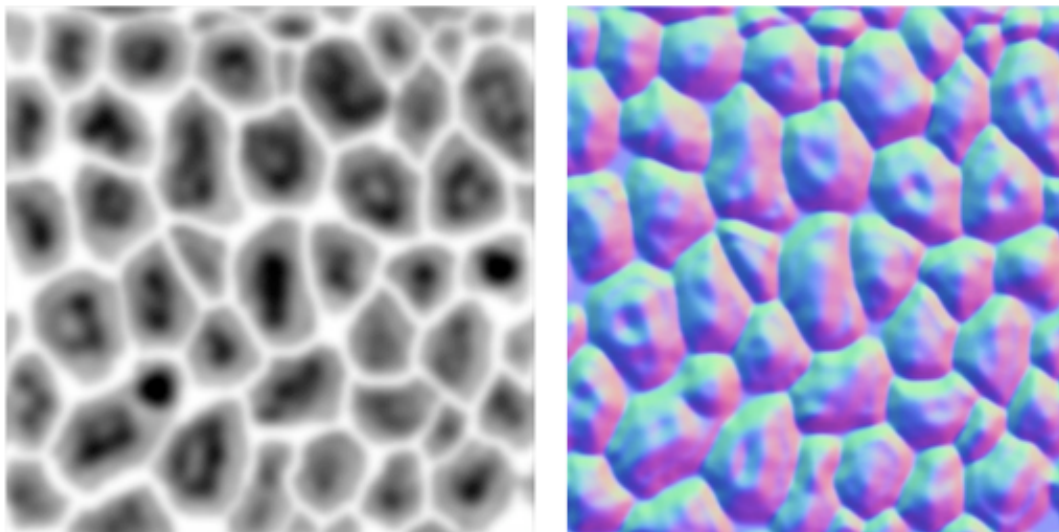


Figura 10. Bump Map (à esquerda) e Normal Map

2.3.4 Displacement Mapping

Como as técnicas citadas acima, o *Displacement Mapping* também tem como finalidade gerar imperfeições e saliências na superfície dos objetos 3D, porém ao contrários dos demais, essa técnica modifica diretamente a geometria do objeto [Camozz07].

A textura, como no Bump Mapping, é em tons de cinza e cada pixel determina o quão distante é o ponto em relação ao estado original [Dogg00]. Nesse caso, quanto mais claro for o pixel, mais distante o ponto será e, portanto, maior é a modificação naquele local.

A vantagem dessa técnica é a realidade apresentada. Ao contrario de Normal Mapping e Bump Mapping que constroem uma visao diferente do objeto para o usuário, o Displacement Mapping utiliza a textura para modificar a estrutura da geometria e, por consequência, para dar uma aparência diferenciada sem nenhum artifício. Em compensação, o custo computacional no geral é um pouco maior devido à geração de novos polígonos para serem renderizados.

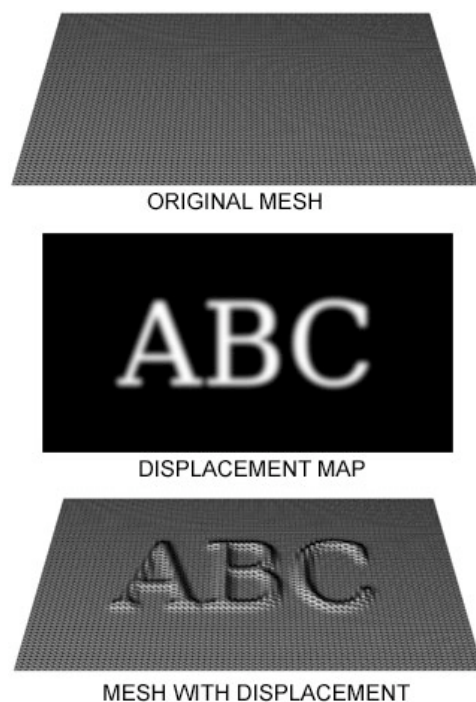


Figura 11. A malha original (cima), o Displacement Map (meio) e o resultado final

2.3.5 Lightmapping

A iluminação de um objeto é um parâmetro bastante considerado no momento em que o usuário visualiza um cenário tridimensional. Os efeitos da iluminação permitem com que o usuário visualize com clareza os objetos que estão na cena bem como suas respectivas posições [Hey02]. Uma cena bem iluminada pode passar uma maior sensação de realismo.

Para criar os efeitos de iluminação, pode-se utilizar diversos algoritmos que causam grande impacto, porém possuem um alto custo computacional. Para aplicações em tempo real, como os jogos, o cálculo de iluminação pode trazer consequências no tempo de resposta do programa [Hey02]. Uma alternativa de baixo custo é o uso de uma técnica de texturização chamada Lightmapping.

A técnica utiliza uma textura conhecida como lightmap que mostra a distribuição da iluminação em uma cena ou em um objeto. Em compensação essa técnica não é capaz de gerar sombra dinâmica em tempo real como na iluminação original, obrigando o usuário a utilizar uma outra técnica para tal tarefa.



Figura 12. Lightmap utilizado na iluminação de uma sala

3. Geração automática de personagens 3D

Nesse capítulo, serão mostradas as diferentes técnicas utilizadas na geração automática de personagens 3D para jogos, sem necessidade de conhecimento de modelagem, exemplificando a criação de cabeças. Além disso, será detalhada a técnica proposta nessa pesquisa, podendo ser comparada com as demais. Para descrever a classificação abaixo, foram estudados vários jogos 3D que contêm geradores de personagens com diferentes características. Algumas das técnicas abaixo são conhecidas na literatura e outras foram incluídas neste trabalho. As plataformas utilizadas nesse estudo, além do PC, foram o Playstation 3 e o Xbox 360, consoles de última geração com uma grande gama de jogos e grande impacto nas vendas.

3.1 Geração por presets

O modo mais fácil de criação de personagens, tanto a cabeça quanto o corpo, é através de geração por modelos pré-fabricados, conhecidos também como *presets*. É um método muito difundido, principalmente no início da explosão dos jogos 3D, que é utilizado até hoje pela praticidade na execução.

Essa técnica utiliza um modelo base, onde a partir dele é montado todo o personagem. Em cima desse modelo base, o usuário pode escolher cada uma das características de seu personagem como olhos, nariz, boca, cabelo, roupa entre outros, através de um conjunto limitado, fabricado durante a produção do jogo, de modelos, gerados por modelagem poligonal, representando essas características.

A vantagem dessa técnica é a facilidade para implementar a abordagem. Com um conjunto fechado de modelos para moldar um personagem, o programador tem pouca dificuldade para desenvolver o processo de criação, apenas precisando dar a liberdade suficiente ao usuário para escolher os presets desejados e aplicá-los no modelo base. A desvantagem é a pouca variedade de personagens possíveis no jogo.

Em jogos multiplayer massivos (conhecidos como MMO's) não é incomum encontrar dois ou mais personagens com as mesmas características. Nesse caso as empresas procuram investir em conjuntos cada vez maiores de presets ou adotar novas abordagens de construção de personagem. Lineage II, da empresa NCSoft, é um jogo de RPG online multiplayer cujo gerador de personagens é baseado em aplicação de presets. Cada raça disponível tem uma quantidade fixa de cabelos, olhos, cor de cabelo e boca de no máximo cinco variações, ou seja, o jogo permite apenas seiscentos e vinte e cinco variações de personagem.



Figura 13. Gerador de personagem de Lineage II com aplicação de presets

3.2 Extração automática por fotografia

Um dos grandes desafios para a indústria de jogos e outras áreas de entretenimento digital, segundo Shan et. al [Shan00], é a capacidade de prover um

rosto de um personagem 3D com características humanas com grande aceitação. Se o personagem em questão é, de alguma forma, conhecido pelo público, essa aceitação é ainda mais passível de críticas.

Uma solução rápida adotada para esse problema é a criação de cabeças de personagens baseadas em fotografias. Duas fotos, uma de frente e a outra de perfil, são utilizadas para extração de características fundamentais na composição de um rosto humano, como olhos, nariz, orelhas, boca e sobrancelhas, através de técnicas de processamento de imagens [Shan00]. Para gerar um personagem completo, gera-se a cabeça para, em seguida, ser encaixada em modelo 3D pré-fabricado de um corpo inteiro. Outra aplicação é o uso da foto utilizada na geração como textura por *Color Mapping*.

Com os pontos chaves extraídos da foto tanto frontal quanto lateralmente, é possível determinar toda a superfície 3D da cabeça, além das características faciais já citadas, e gerar, como resultado final, uma cabeça tridimensional com a malha triangularizada, obtida de maneira semelhante à modelagem poligonal, descrita na seção 2.1.3. A vantagem dessa técnica é a possibilidade que o jogador possui de criar um personagem com a cabeça e o rosto iguais a ele. Além disso, para produção em série, criar personagens baseados em fotos pode acarretar um ganho no tempo de desenvolvimento. Em contrapartida, a qualidade dos modelos gerados não é garantida e é diretamente proporcional à qualidade da foto utilizada na extração das características. Fotos com muita iluminação ou com baixa qualidade podem gerar ruído no modelo final.

Um software que utiliza essa técnica, em parte, é o AvMarker da empresa Cyber Extruder. O Cyber Extruder utiliza apenas uma única foto, na posição frontal, para a extração das características faciais. É possível manipular os pontos detectados e as linhas entre esses pontos, dando uma maior liberdade ao usuário. Em compensação a ausência da foto lateral implica em um tamanho padrão para todas as cabeças geradas. A parte traseira da cabeça é um modelo pré-fabricado (preset) que é encaixado com a parte frontal, gerada pelo software.

Jogos que utilizam essa ferramenta como uma solução direta para a criação de personagens não possuem um alto impacto no visual, porém o resultado pode ser satisfatório de acordo com a proposta do mesmo.

O Second Life, criado pela Linden Labs, é um jogo que utiliza o AvMarker para criação de seus personagens. O foco deste game são as interações sociais dos diversos jogadores espalhados pela internet, em um mundo virtual, em uma série de atividades como esportes, shows, baladas, negócios, relacionamentos e muitos outros. Como o Second Life exige uma demanda maior na parte de rede e seu foco é voltado para a interatividade, o uso de personagens criados por extração de fotografias é uma solução prática, com menor custo computacional e abre a possibilidade para o jogador criar personagens com o próprio rosto, sem necessitar de conhecimento em modelagem.

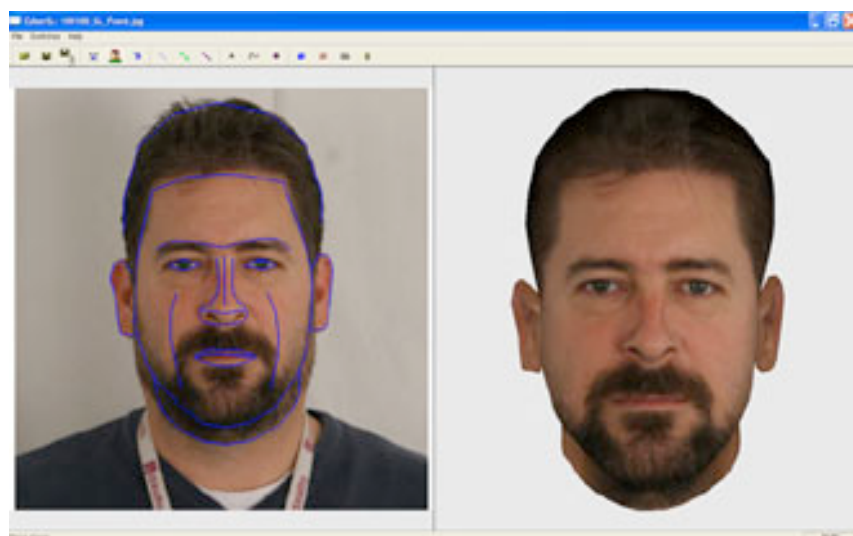


Figura 14. Demonstração do AvMarker. Foto (esquerda) e a cabeça gerada.

3.3 Manipulação de bones

Um modelo 3D não é constituído apenas de vértices e polígonos. Existem, internas à sua malha, estruturas não-renderizáveis chamadas *bones*. Os *bones* são

responsáveis pelas articulações dos modelos, exercendo influência nos vértices mais próximos [Oga03], ou seja, se a posição de um bone é alterada, os vértices influenciados também são deslocados.

Graças a essa propriedade, os bones podem ser utilizados para realizar alterações na estrutura de um modelo 3D. É possível aplicar operações de rotação, translação e escala em um bone e, conseqüentemente, rotacionar, transladar e escalonar partes específicas de um modelo. A vantagem dessa técnica é um maior controle sobre as alterações exercidas, principalmente pequenos detalhes, porém, para alterações mais gerais, essa técnica tende a ser mais trabalhosa.

O TopSpin 3, da empresa PAM Development, é um jogo de tênis para o console Xbox 360, que implementa um gerador de personagens baseado em manipulação de bones. Na parte facial, o usuário pode manipular as diversas articulações existentes na cabeça, esculpindo ou deformando o rosto conforme sua vontade. Na parte corporal, é possível deixar o personagem mais alto ou baixo e mais largo ou mais magro, através da operação de escala sobre os *bones* do tórax e das pernas.



Figura 15. Gerador de personagem de TopSpin 3 com manipulação de bones

3.4 Interpolação por parametrização

Seo e Magnetat-Thalmann descrevem uma maneira de geração de personagens através do controle de parâmetros passados pelo usuário [Seo03]. Essa abordagem envolve a utilização de um modelo base que será o modelo padrão do processo e mais uma série de modelos adicionais, todos pré-fabricados, que sofrerão operações de interpolação com o modelo base.

Ainda segundo o método de Seo e Magnetat-Thalmann, os modelos adicionais são criados de forma semelhante ao modelo base, porém com uma característica diferente. Essa característica pode ser controlada pelo usuário, determinando o quão essa característica em particular irá se manifestar no modelo base que é o centro dessa abordagem [Seo03]. Esse fator de interpolação está contido em um intervalo de zero a cem por cento.

A interpolação é feita pela soma dos vértices do modelo base com a diferença entre os pontos correspondentes do modelo base e do modelo adicional e essa diferença é multiplicada pelo fator de interpolação. Quanto maior for o fator, maior é o deslocamento dos vértices do modelo base para a posição dos vértices do modelo adicional.

Assim, com diferentes modelos adicionais e, conseqüentemente, com diferentes características manipuláveis, a estrutura final do modelo é representada pela soma das características, multiplicadas pelos seus respectivos fatores de interpolação.

A vantagem dessa abordagem é a visualização que o usuário possui das alterações realizadas sobre o modelo base e pela facilidade em fazer grandes alterações no modelo em pouco tempo. Outra facilidade exposta por esta técnica, é a possibilidade de aplicar efeitos de envelhecimento no modelo onde o único requisito é que o sistema possua um modelo adicional que representa uma pessoa idosa.

A empresa Inverse Solutions, desenvolveu um framework que implementa essa abordagem para a criação de cabeças 3D chamada FaceGen. O FaceGen possui uma série de controles, onde cada controle manipula uma característica diferente do rosto como nariz e queixo, através de *sliders*. Após modificar o modelo conforme sua vontade, o usuário pode exportar o resultado final em um arquivo com formato padrão para softwares de animação e modelagem como o Maya e o Blender. Este framework é utilizado em alguns jogos conhecidos do público como The Elder Scrolls: Oblivion, da empresa Bethesda Softworks.



Figura 16. Gerador de personagem de Oblivion com interpolação paramétrica

4. GCM – Game Character Maker

O estudo dos processos de criação de personagens 3D para jogos e as atuais técnicas utilizadas pelos geradores de personagens nos jogos modernos tem como objetivo a criação de um framework que atenda aos requisitos do mercado para geração de personagens humanóides e que seja de fácil uso para pessoas com ou sem conhecimentos em modelagem e animação. Para os usuários que conhecem técnicas de modelagem é possível modificar internamente a estrutura do software e aumentar a capacidade de geração do mesmo.

O resultado final da pesquisa foi a elaboração de um projeto de software apelidado de GCM (sigla de Game Character Maker). O GCM é dividido em dois módulos: o FaceMaker, módulo responsável pela geração e texturização de cabeças, e o BodyMaker, módulo responsável pela geração do corpo do personagem. Na fase de implementação, o FaceMaker foi concluído e será descrito no decorrer do capítulo e no capítulo seguinte com a amostra dos resultados e conclusões. O BodyMaker também será descrito porém sem a análise de resultados devido ao estado atual da implementação.

4.1 FaceMaker

O FaceMaker, módulo responsável pela geração de faces e cabeças, foi desenvolvido e implementado durante o tempo decorrente desta pesquisa. Este módulo foi quebrado em três diferentes sub-módulos, devido às funcionalidades bem definidas e diferenciadas de cada uma delas. As três divisões do FaceMaker serão detalhados abaixo.

4.1.1 Edição de geometria

O primeiro sub-módulo do FaceMaker é o responsável pela edição da geometria do personagem. Nesta etapa, o usuário poderá definir o formato da cabeça e as feições do rosto do personagem. O editor de geometria possui duas janelas visíveis para o usuário: o visualizador principal e a janela de controle de parâmetros.

A janela de parâmetros contém uma série de controles, que determinam os diversos parâmetros presentes na cabeça do personagem. Cada controle possui um slider, por onde o usuário através do mouse determina o valor do parâmetro associado ao controle, um campo de texto estático mostrando o nome do controle em questão e um campo de texto dinâmico onde o usuário poderá passar através do teclado o valor do parâmetro entre 0 e 100, sem a necessidade do uso do slider. O slider e o campo de texto são sincronizados entre si e cada mudança feita em um dos componentes será refletida no outro.

A abordagem utilizada neste sub-módulo é a interpolação por parametrização, descrito na seção 3.4. Nos geradores mais tradicionais que utilizam essa técnica, cada slider é associado a um par de modelos adicionais cujas características são completamente opostas como “modelo de nariz largo”(P1) e “modelo de nariz fino”(P2), por exemplo. Quanto maior o valor do slider, maior é a interpolação para o modelo adicional com a característica mais proeminente como o P2, citado no exemplo anterior, e maior é a interpolação para o modelo adicional de menor expressão quando o valor do slider é baixo como P1. No FaceMaker, cada slider é associado a apenas um único modelo adicional, ou seja, P1 e P2 estão representados em controles diferentes e são independentes um do outro. Essa abordagem foi utilizada para dar maior liberdade na edição da malha da cabeça. Também na janela de controles, existe um botão para o usuário zerar todos os parâmetros sem necessidade manipular cada controle para atingir o estado inicial.

O visualizador principal mostra ao usuário o resultado final da cabeça após ser aplicada a interpolação com os parâmetros definidos pelo usuário. As mudanças

de parâmetros feitas pelo mesmo são aplicadas diretamente no visualizador em tempo real, sendo possível ver a geometria transitar entre os estados antes e depois da execução das mudanças dando ao usuário a sensação de transformação do modelo. Caso todos os parâmetros estejam zerados, o resultado do visualizador é o modelo base já contido no pipeline do framework.

O modelo base e os modelos adicionais foram criados a partir de modelagem poligonal conforme descrito na seção 2.1.3.

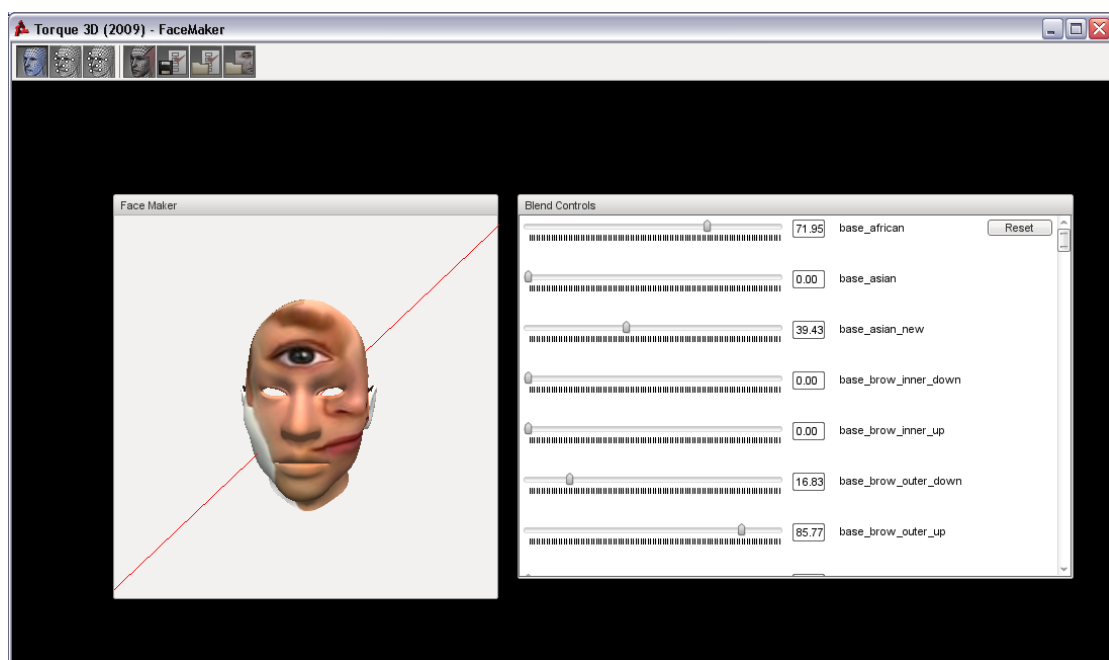


Figura 17. Modo de Edição de Geometria

4.1.2 Encaixe de textura

O segundo sub-módulo do FaceMaker é a parte de encaixe de textura. No FaceMaker é possível o usuário aplicar uma foto qualquer como textura para a cabeça. Como as fotos em questão podem ser tiradas de vários ângulos e posições diferentes, é necessário que o usuário tenha poder para encaixar a textura na cabeça moldada do personagem a partir de alguns controles.

Essa divisão possui 4 janelas visíveis com mais uma opcional para o usuário: o visualizador principal, visualizador de imagem, visualizador de UV e janela de pontos

de controle. O visualizador de deformação de textura é opcional e pode ser ocultado.

O visualizador principal é o mesmo citado na seção anterior. A finalidade de tê-lo nesse sub-módulo é poder ver em tempo real as modificações de textura aplicadas diretamente sobre a cabeça.

O visualizador de imagem e o visualizador de UV estão interligados. Os dois são responsáveis pela deformação e encaixe da textura no modelo final da cabeça. O visualizador de imagem mostra a textura que será aplicada na cabeça e o visualizador de UV mostra a UV do modelo final no modo wireframe. Em ambos os visualizadores é possível a inserção de pontos de controle para o encaixe da textura e cada ponto de controle possui coordenadas em ambos os visualizadores ou seja, cada vez que um ponto de controle é inserido em um dos dois visualizadores, ele também é visível no outro. Cada ponto de controle possui seis parâmetros: coordenadas x e y para textura, coordenadas x e y para UV, peso e raio de influência. Ao manipular o ponto de controle no visualizador de imagem, o usuário determinará a posição da foto que irá representar e ao manipular o mesmo ponto no visualizador de UV, o usuário determinará em que região da cabeça será aplicada a região da imagem determinada no visualizador de imagem. Assim, o usuário tem total controle de onde e como a textura será aplicada no modelo final da cabeça. O raio de um ponto de controle determina a área da imagem que o ponto de controle influencia a partir da posição do próprio ponto e o peso determina quanto a região de influência do ponto de controle é atingida por outros pontos de controle. Quanto maior o peso, menor é a interferência exercida por outros pontos.

Os pontos de controle são vermelhos e, quando selecionados pelo mouse simplesmente passando por cima deles, ficam de cor verde em ambos os visualizadores, destacando assim em que ponto correspondente o usuário está manipulando.

A quarta janela é o repositório de pontos de controle. Esta janela permite ao usuário ter as informações de todos os pontos visíveis nos visualizadores de imagem e de UV. Cada ponto possui um rótulo com todas as informações de coordenadas de

textura e de UV, o raio e o peso do ponto. Nesses rótulos é possível manipular essas variáveis através de botões de incremento e decremento. Outra possibilidade é a remoção do próprio através de um botão em formato de lixeira contido no próprio rótulo.

A janela opcional é o visualizador de deformação de textura. Quando o usuário insere pontos de controle e controla seus parâmetros para encaixar a imagem corretamente no modelo, essa imagem sofre várias deformações de acordo com o estado inicial da imagem e esse visualizador permite ao usuário ver a deformação aplicada na imagem. A imagem deformada é a textura final da cabeça.

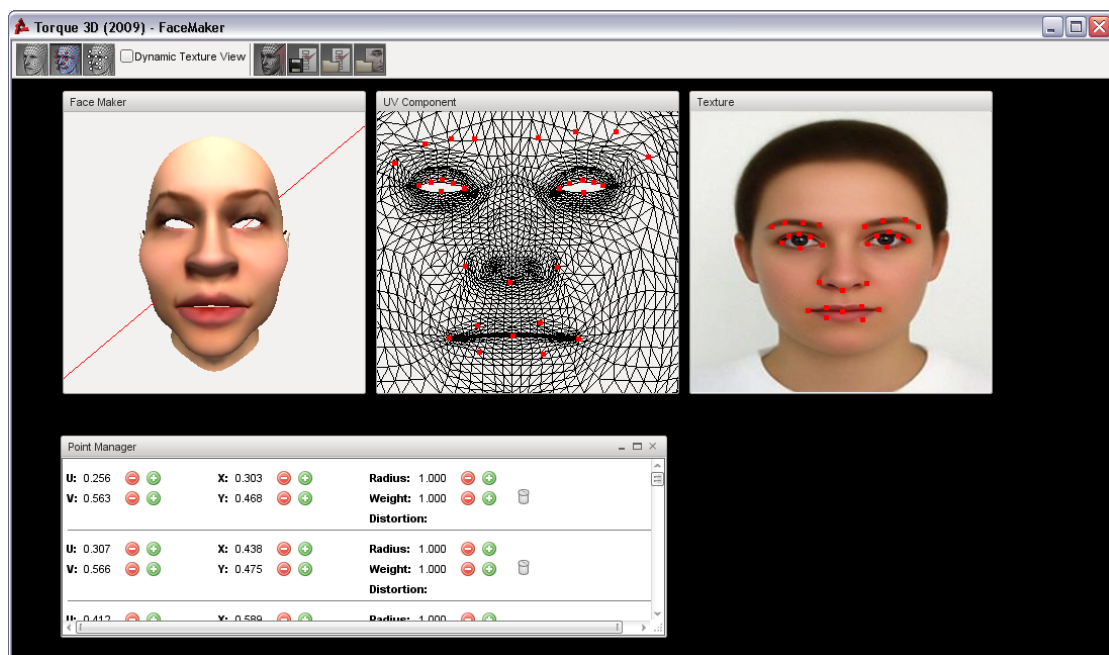


Figura 18. Modo de Encaixe de Textura

4.1.3 Filtragem da textura

O terceiro sub-módulo do FaceMaker é a aplicação de filtros na textura da face. O usuário pode carregar uma foto para que ela seja aplicada como textura para a cabeça, porém a foto pode estar com baixa qualidade devido à forte iluminação, o

que gera muito brilho na foto e muitas regiões de sombra. Essa parte tem como função, aplicar filtros de correção na imagem que removem o excesso de brilho.

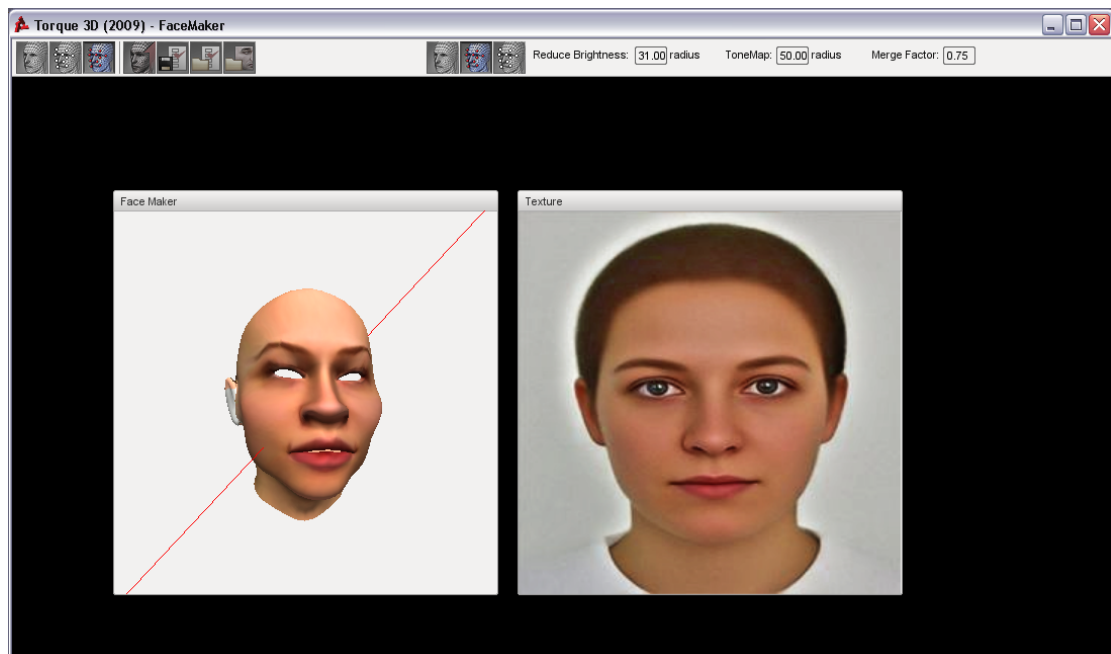


Figura 19. Modo de Filtragem – Redução de Brilho e Sombreamento Excessivo

Difícilmente uma imagem carregada pelo usuário no framework está apta a se tornar uma textura eficiente para a cabeça. A razão disto é a presença de vários componentes que não são necessários para a composição da textura final como o plano de fundo da imagem e o cabelo da foto. Nos jogos 3D, a textura da cabeça corresponde a imagem da face esticada em um plano sem o cabelo, que é tratado como um objeto separado e encaixado posteriormente sobre a cabeça. Este submódulo também possui como funcionalidade recortar a face na imagem e esticá-la sobre a textura. Tanto o recorte da face quanto a escolha de tom da pele esticada são feitos pelo usuário.

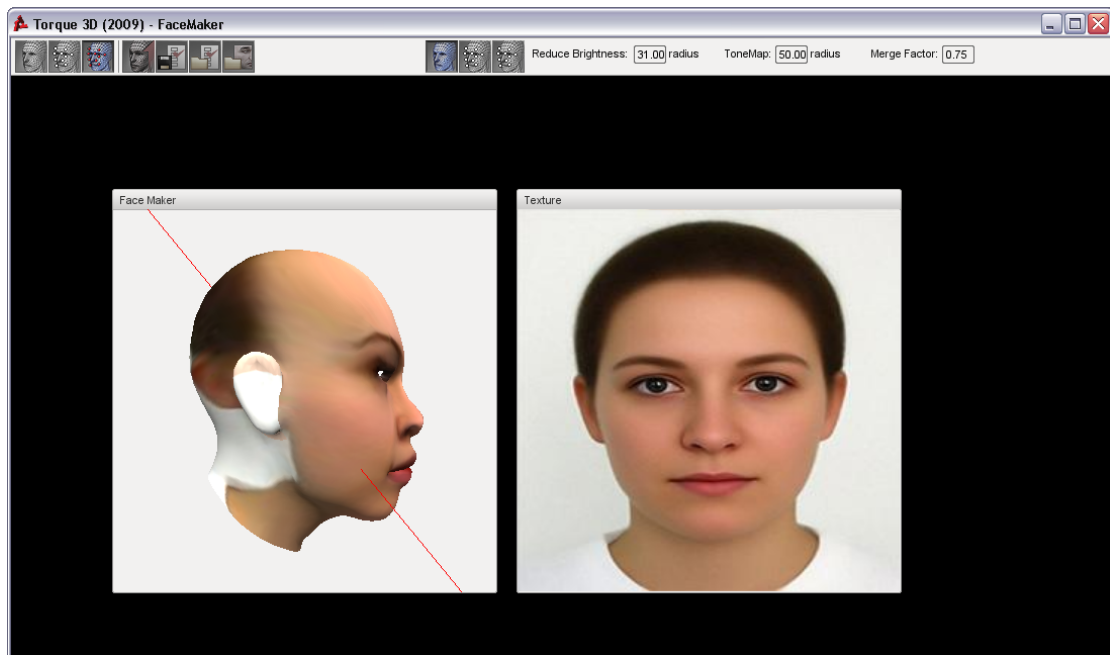


Figura 20. Modo de Filtragem – Imagem Normal

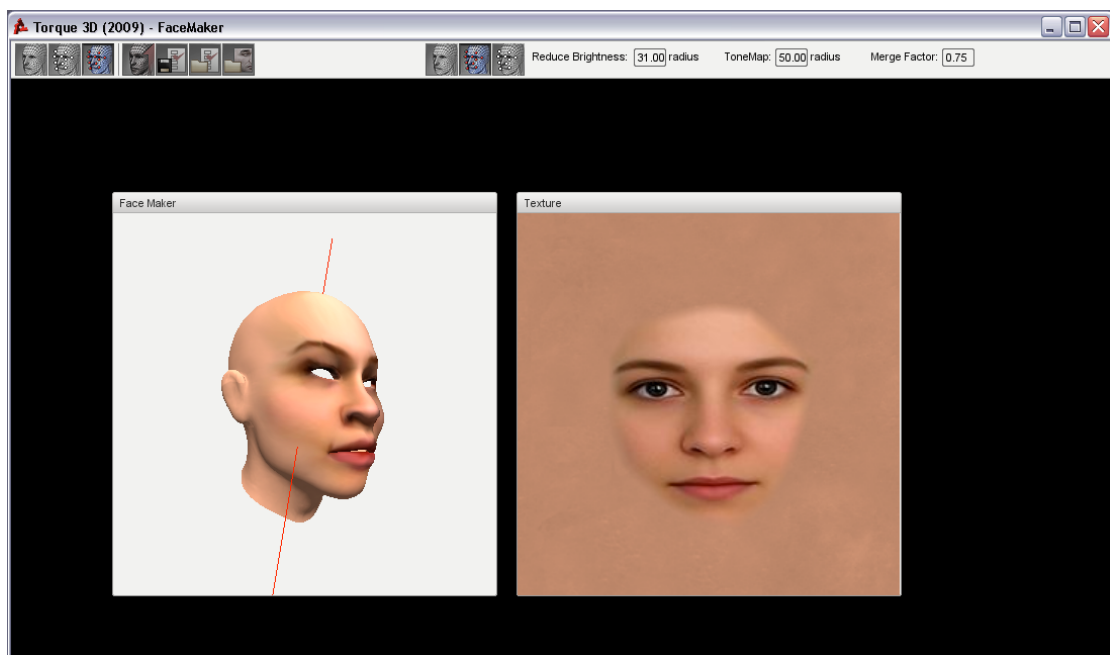


Figura 21. Modo de Filtragem – Rosto Recortado e Esticado para Encaixe

4.2 BodyMaker

O BodyMaker é o segundo módulo elaborado para o GCM. Este módulo, ainda não implementado, tem como objetivo definir a estrutura do corpo do personagem humanóide.

Nesta parte, o usuário poderá definir a altura e o peso do personagem através da técnica de manipulação de bones, descrita na seção 3.3. Duas janelas serao utilizadas nessa fase: um visualizador principal e uma janela de controle de parâmetros.

O visualizador principal tem como função, exibir o personagem completo ao usuário, incluindo a cabeça definida no FaceMaker. A janela de controle possui sliders como descritos na seção 4.1.1, um para a altura e outro para o peso do personagem. Porém, ao contrário do FaceMaker que trabalha diretamente com a manipulação dos vértices da malha, o BodyMaker utilizada a alteração da escala dos bones que compoem o corpo. Ao modificar a altura, o usuário estará alterando a escala dos bones que estão nas pernas e ao modificar o peso, o usuário estará alterando a escala dos bones que estão no tronco do personagem.

4.3 Torque Game Engine

O FaceMaker foi construído na Torque Game Engine (TGE), uma engine de jogos 3D de classe AAA desenvolvida pela GarageGames para desenvolvedores independentes [Torque] e atualmente a TGE é conhecida como Torque 3D, ou T3D, por causa de modificações e novas funcionalidades atribuídas à engine ao longo dos anos.

A T3D foi desenvolvida em C++, possui código-fonte aberto e uma linguagem de script própria para desenvolvimento chamada TorqueScript. Graças a baixa curva de aprendizado para o script, a possibilidade de desenvolver novas funcionalidades e componentes à engine e às funcionalidades de manipulação de objetos 3D já

presentes na engine, a T3D se mostrou uma ótima ferramenta para o desenvolvimento do projeto.

5. Avaliação e Resultados

5.1 Avaliação de desempenho

Um dos objetivos desse trabalho é permitir ao usuário que ele crie personagens humanóides através do controle de parâmetros em tempo hábil, com o mínimo de atraso possível. No GCM é possível um usuário com experiência em modelagem ampliar a capacidade do sistema com a inserção de novos modelos adicionais. Para isso basta abrir o arquivo “base2.dae”, modificar a característica facial desejada e adicionar na pasta que contém os modelos adicionais já prontos. Quanto maior for a quantidade de modelos adicionais para a interpolação, maior é o processamento da aplicação.

O FaceMaker, modulo do GCM, foi desenvolvido e testado em um computador [configuração do computador] e foram utilizados, além do modelo base, 90 modelos adicionais. O resultado do teste de desempenho é satisfatório, pois o usuário tem a sensação de que o modelo base está se transformando com a mudança do parâmetro. Porém, ainda é perceptível um atraso mínimo principalmente se o usuário arrasta bruscamente com o mouse o slider de parâmetro. Como a operação de interpolação é composta basicamente de soma, subtração e multiplicação em largae scala, a solução encontrada para eliminar o atraso é a utilização da linguagem ASM x86 para realizar essas operações devido ao fato que o ASM acessa diretamente os registradores da CPU e, conseqüentemente, as operações seriam executadas muito mais rapidamente.

5.2 Considerações sobre o encaixe de textura por pontos de controle

O encaixe da imagem na cabeça é essencial na geração da cabeça do personagem. No FaceMaker, alguns usuários testaram o sub-módulo de encaixe de textura, ao delimitar e encaixar os olhos, o nariz, as sobrancelhas e a boca.

O desempenho do sub-módulo foi bem aceito pelos usuários, principalmente aqueles com experiência em modelagem e texturização. Utilizando uma malha 32 x 32 para distorcer e encaixar a imagem, o desempenho foi considerado ótimo, não havendo nenhuma observação sobre atrasos no processamento, mesmo com uma grande quantidade de pontos para realizar a tarefa (cerca de 20 pontos).

Em relação ao resultado final da imagem, foi constatado que certas regiões da cabeça, já texturizada, podem apresentar algumas falhas na visualização ao utilizar dois pontos de controle próximos e que não coincidam originalmente com o estado inicial da imagem. Isso é resolvido ao inserir mais um ponto de controle no local afetado e encaixá-lo apropriadamente como citado na seção 4.1.2. No geral, o resultado final foi considerado ótimo e útil para agilizar a produção de personagens.

5.3 Justificativa para tratamento de imagem no framework

Como já mencionado na seção 4.1.3, o FaceMaker tem como proposta reduzir o brilho e as áreas sombreadas da imagem que será utilizada como textura para a cabeça gerada. A razão dessa funcionalidade foi o problema detectado por alguns usuários que testaram a aplicação e verificaram que a maioria das imagens utilizadas poderiam conter brilho excessivo ou serem muito escuras. Isso se deve ao fato que várias imagens utilizadas foram tiradas sem qualquer preparo especial para a aplicação, ou seja, são fotos comuns.

5.4 Inserção do modelo gerado em um jogo exemplo

Finalmente foi analisado o comportamento do modelo gerado pelo GCM em um ambiente de jogo fornecido pela Torque Engine. No caso, foi gerado uma cabeça com diferentes parâmetros através do FaceMaker em conjunto com uma textura gerada através do encaixe de uma imagem selecionada pelo usuário na cabeça 3D

em questão. Em seguir, a cabeça texturizada com material foi inserida em um ambiente 3D próprio para um jogo computacional, através de TorqueScript.

O resultado final foi considerado satisfatório. Todas as características visualizadas pelo usuário durante a geração da cabeça no FaceMaker foram confirmadas com sucesso no ambiente de jogo. Todas as características peculiares determinadas pelo usuário e os detalhes da pele da cabeça feitos durante o tratamento da imagem, também no FaceMaker, foram comprovados em sua totalidade sem nenhuma ausência ou falha durante o processo.

6. Considerações Finais

Os objetivos traçados para este trabalho foi atingidos parcialmente, em cerca de 75%. Foi proposta e elaborada uma abordagem para gerar um personagem humanóide 3D através de um framework que contivesse controles de baixa curva de aprendizado para que um usuário, acostumado a jogos 3D e criação de personagens, pudesse criar personagens personalizados com uma enorme variedade tanto na geometria quanto na textura final. Para um usuário com experiência em modelagem e animação, também seria possível adicionar novos modelos de interpolação e aumentar ainda mais a capacidade de geração. O GCM, produto final desse trabalho, pode ser incorporado diretamente em um jogo ou utilizado para gerar conteúdo 3D em grande escala e, posteriormente, inserir o conteúdo gerado em um jogo tridimensional.

No entanto foi concluído apenas o FaceMaker, módulo responsável pela geração de cabeças para o personagem. Isso se deve ao fato de o tempo total necessário para a conclusão do projeto ser estimado em 6 meses, dos quais 4 meses focados apenas no FaceMaker, devido a uma maior complexidade pelos detalhes envolvidos tanto na geometria do objeto quanto no tratamento de imagens com aplicação de diversos filtros para a texturização da cabeça. Nessa etapa, os objetivos foram alcançados com sucesso.

Porém existe muito espaço para melhorias no FaceMaker. Como citado na seção 5.1, o algoritmo de interpolação deve ser substituído para oferecer maior eficiência durante o processo e eliminar os atrasos. No encaixe de textura, novas e diversas funções matemáticas para deformações na textura deverão ser utilizadas e poderão, no futuro, serem aplicadas independentemente para cada ponto. Atualmente é aplicada uma função exponencial para tal tarefa, mas foi cogitado a utilização de funções quadráticas, bilineares e gaussianas e deverão ser testadas.

Outro ponto que deve ser considerado com atenção é o tratamento de imagem para texturização. A textura base de pele utilizada na edição deve ser melhorada e modificada para dar maior realismo à cabeça e os algoritmos de

redução de brilho e seleção de rosto na imagem podem ser escritos em shaders para reduzir o processamento e possibilitar ao usuário aplicá-los em tempo real sem a necessidade esperar o fim do evento de mouse para ativar os algoritmos.

Uma característica que deve ser testada é a geração de modelos com animações faciais. Atualmente, os modelos base e adicionais são estáticos e as cabeças são também são. Isso se deve ao fato de não haver nenhum profissional em animação estar disponível no momento do desenvolvimento do projeto porém, isso sera solucionado em poucas semanas.

Já o BodyMaker, modulo responsável pela edição da estrutura corporal do personagem, será desenvolvido e integrado ao FaceMaker para finalizar o GCM. O tempo estimado para a execução desse módulo foi de 2 meses.

7. Referências

- [Torque] Maurina, E. F., III. **The Game Programmer's Guide To Torque: Under The Hood of the Torque Game Engine**. GG Press, 2006.
- [Rosa] Rosa, D. R. **A Representação da Figura Humana em Meio Digital**. UDESC – CEART, n/d
- [ESA09] **Essential Facts about Computer and Video Game Industry**. Pesquisa publicada pela Entertainment Software Association, 2009, Acessado em Outubro / 2009, Disponível em: <http://www.theesa.com>.
- [Cal06] Calife, D.; Coimbra, W. A. P.; Toscano, W. **Engines Gráficos 3D**. Escola Politécnica – Universidade de São Paulo (USP), 2006.
- [Bat00] Battaiola, A. L. **Jogos por Computador – Histórico, Relevância Tecnológica e Mercadológica, Tendências e Técnicas de Implementação**. JAI/2000 SBC, 2000.
- [Fog07] Foggiato, J. A.; Volpato, N.; Bontorin, A. C. B. **Recomendações para Modelagem em Sistemas CAD-3D**. 4º Congresso Brasileiro de Engenharia de Fabricação, 2007.
- [Miya08] Miyamoto, R. T.; Medeiros Filho, D. A.; Sartori, A. R. **Modelagem Tridimensional com Geometria Construtiva de Sólidos para Projetos de Engenharia e Arquitetura em Sistemas CAD visando a Portabilidade e Estabilidade de Objetos 3D**. XIV Convención Científica de Ingeniería y Arquitectura, 2008.
- [Min03] Minetto, C. F. **Um Estudo sobre Curvas NURBS**. Dissertação submetida como requisito parcial para a obtenção do grau de Mestre em Matemática Aplicada, 2003.
- [Sgarbi] Sgarbi, R.; Fernandes, A. M. R. **Um Tutorial para Ensino de Modelagem de Modas baseado em Ambiente 3D**. Universidade do Vale do Itajaí (UNIVALE), n/d.
- [Clua05] Clua, E. W. G.; Bittencourt, J. R. **Desenvolvimento de Jogos 3D: Concepção, Design e Programação**. XXV Congresso da Sociedade Brasileira de Computação, 2005.

[Louro] Louro, S.; Reis, L. P.; Oliveira, E. **Virtual 3D: Sistema Multi-Agente de Visualização com Controlo Inteligente de Câmara**. Laboratório de Inteligência Artificial e Ciência de Computadores, Universidade do Porto, Portugal, n/d.

[Silva96] Silva, F. W. S. V.; Cavalcanti, P. R. **Animações Complexas em Tempo Real Utilizando Movimentos Capturados**. LCG – Laboratório de Computação Gráfica, COPPE – Sistemas / UFRJ, 1996.

[Pozzer07] Pozzer, C. T. **Síntese de Imagens e Realismo**. Universidade Federal de Santa Maria, 2007.

[Evan07] Evangelista, B.; Silva, A. **Criando Efeitos Fotorealistas e Não-fotorealistas para Jogos**. Universidade Federal de Minas Gerais, 2007.

[Camozz07] Camozzato, D. **Avaliação e Desenvolvimento de Técnicas para o Aumento do Espaço de Validade em Impostores**. Pontifícia Universidade Católica do Rio Grande do Sul, Programa de Pós-Doutorado, 2007.

[Dogg00] Doggett, M.; Hirche J. **Adaptive View Dependent Tessellation Of Displacement Maps**. WSI / GRIS, University of Tübingen, Germany, 2000.

[Hey02] Hey, H.; Purgathofer W. **Real-Time Rendering Of Globally Illuminated Soft Glossy Scenes With Directional Light Maps**. Institute of Computer Graphics and Algorithms, Vienna University Of Technology, 2002.

[Shan00] Shan, S.; Gao W.; Yan, J.; Zhang, H.; Chen, X. **Individual 3D Face Synthesis Based On Orthogonal Photos and Speech-Driven Facial Animation**. 7th IEEE International Conference Of Image Processing (ICIP 2000), Vancouver, British Columbia, Canada, 2000.

[Oga03] Ogawara, K.; Takamatsu J.; Hashimoto, K.; Ikeuchi, K. **Grasp Recognition using a 3D Articulated Model and Infrared Images**. IEE/RSJ International Conference on Intelligent Robots and Systems, 2003.

[Seo03] Seo, H.; Magnenat-Thalmann N. **An Automatic Modeling Of Human Bodies from Sizing Parameters**. Miralab, University Of Geneva, Proceedings of the 2003 Symposium on Interactive 3D Graphics, 2003.