UNIVERSIDADE FEDERAL DA BAHIA
FACULDADE DE COMUNICAÇÃO

ATHOS CORREIA SAMPAIO

A UTILIZAÇÃO DE ELEMENTOS DA LINGUAGEM CINEMATOGRÁFICA EM AMBIENTES VIRTUAIS INTERATIVOS

Salvador
2006

ATHOS CORREIA SAMPAIO

A UTILIZAÇÃO DE ELEMENTOS DA LINGUAGEM CINEMATOGRÁFICA EM AMBIENTES VIRTUAIS INTERATIVOS

Memória descritiva e analítica de produto de natureza técnico-artística apresentada ao Curso de graduação em Comunicação/Jornalismo, Faculdade de Comunicação, Universidade Federal da Bahia.

Orientador: Prof. André Olivieri Setaro

Salvador
2006

RESUMO

O produto descrito nesta memória é uma série de ambientes virtuais interativos em 3 dimensões. Estes ambientes virtuais utilizam-se de alguns elementos sonoros e imagéticos comumente presentes no cinema tradicional e reconhecidamente responsáveis por parte do potencial imersivo da mídia cinematográfica, como: trilha sonora adequada ao clima da cena e à sensação que se espera causar no espectador; efeitos sonoros, em geral correspondentes a seres e elementos inanimados do cenário; elementos cenográficos climáticos que reforçam o significado das imagens e a localidade do ambiente mostrado, como por exemplo neve e névoa atmosférica; uma escolha de cores e variações de tons e texturas previamente pesquisadas, sempre tendo como parâmetro discriminatório a harmonia entre os elementos e o conceito visual do produto; Elementos animados nos cenários de fundo, com o intuito de criar dinâmica (cinestesia), conservar a atenção do espectador (no caso de ambientes virtuais: o usuário), além de despertar cadeias de significados e narrativas na mente do mesmo; entre outros elementos de igual potencial narrativo e sensorial.

Palavras-chave: Jogos; Realidade Virtual - Turismo Virtual; Computação Gráfica.

Este produto é dedicado a minha mãe, Leci, e a Paulinha, pelo grande suporte diário de ambas, que me permitiu levar adiante os meus sonhos e pela extrema paciência em ouvir diariamente sobre uma série de conteúdos altamente técnicos relacionados a jogos 3D e programação.

"Um bom artista de jogos [em 3 dimensões] é um alquimista dos dias atuais, criando não apenas meros objetos a partir de polígonos e pixels, mas mundos inteiros."

Matthew Omernick

SUMÁRIO

1. APRESENTAÇÃO

1.1 OS PRIMEIROS CONTATOS COM A CRIAÇÃO DE JOGOS: UM BREVE PRÓLOGO SE FAZ NECESSÁRIO

O meu envolvimento com jogos para computador e videogames data de muito antes do meu ingresso na Faculdade de Comunicação da UFBA. Na verdade, no início da adolescência, quando comprei meu primeiro videogame com a junção dos pagamentos pelos primeiros trabalhos profissionais de ilustração que me foram encomendados, aos 11 anos de idade. É interessante hoje perceber que o fato de ter começado a trabalhar com essa idade, me fez ver alguns produtos audiovisuais e de entretenimento (como ilustrações, quadrinhos, filmes, jogos, desenhos animados) do ponto de vista do autor, de certa forma – pois, apesar de ainda haver em minha mente uma enorme lacuna e uma correspondente curiosidade quanto ao processo e às técnicas por trás dessas obras, começava a dar os primeiros passos no sentido de me tornar um artesão de imagens, como hoje sou em certa medida. Quando digo "em certa medida", é porque tenho certeza que não cheguei ainda a explorar todo o meu potencial nem possuo técnica suficiente para expressar os incontáveis universos de fantasia e mistério que

povoam a minha mente enquanto artesão e "alquimista de mundos virtuais", os quais tenho a firme intenção de conceber e concretizar enquanto produtos prontos e acabados.

Provavelmente o pequeno empurrão necessário para passar de consumidor de jogos (leia-se "gamemaníaco") a aspirante a autor de jogos, numa cidade onde um mercado dessa área é praticamente inexistente, deu-se através de uma série de pequenos incidentes que mantiveram viva a chama da minha curiosidade e vontade de produzir e trabalhar com isso (também). Faz-se necessária uma breve narrativa, devido no mínimo ao pitoresco de se desenvolver jogos em três dimensões (3D) numa cidade – e um estado – onde, como dito antes, não há profissionais dessa área, nem estabelecimentos de ensino. Ou seja: nenhuma fonte de conhecimento e transmissão do ofício que não os livros e sites de tutoriais (pequenos aulas avulsas de uma técnica específica).

Sempre fui fascinado pelo fazer e pelo "como é feito", fato que com toda certeza foi estimulado e mantido vivo pela convivência com meu pai, Enéas Guerra - artista gráfico, designer e editor de livros. Por esse convívio, fui direcionado para as artes gráficas. Isso se somou à minha paixão e inclinação para as narrativas (provavelmente minha inclinação maior), a ficção e a literatura, pois o desenvolvimento de uma capacidade (embrionária ainda) de desenhar o mundo foi extremamente benéfica, pois me permitiu dar vida e forma aos meus personagens e universos de fantasia. Também creio ter sido marcante para a minha formação o fato de ter aprendido a ler e escrever em casa, com minha mãe, anos antes da alfabetização na escola. Ali se estruturou, dessa forma, o meu contato com o conhecimento e os livros: de uma forma prazerosa, estimulante e onde a minha postura diante dos livros era ativa, de iniciativa, de um pequeno pesquisador auto-didata.

Assim, com uma incipiente formação de artista gráfico e autodidata, veio o terceiro ingrediente que combinado aos demais me tornaria um artista de jogos no

contexto de uma cidade onde, como foi dito, não havia condições de estímulo: o contato com o computador e os programas de edição gráfica e desenho digital. Também alguns pequenos incidentes, como o contato, em momentos diferentes, com dois técnicos de informática e programadores que mediante "entrevista" me transmitiram o que sabiam sobre o processo de criação de um jogo para computador, videogame ou arcade (também conhecido como fliperama: aquelas máquinas de jogos encontradas em shopping centers e outros estabelecimentos).

Cheguei a descobrir o nome de dois programas de criação de jogos para computador. Jogos que nessa época ainda eram bidimensionais, feitos a partir de desenhos e pinturas digitais e pinturas tradicionais digitalizadas por meio de scanner. Ou seja: eram jogos basicamente feitos por artistas gráficos vindos de uma formação tradicional e programadores (profissionais que criam programas para computador). Porém, após uma exaustiva busca na internet por estes programas (em 1994 e 1995), com o pouco domínio que tinha das ferramentas de busca na internet e o ínfimo conhecimento que havia obtido por meio de pessoas, revistas importadas e textos em embalagens de jogos, não obtive sucesso.

No ano de 2000, após novas pesquisas, tomei conhecimento de um software livre[1] chamado Adventure Game Studio (http://www.adventuregamestudio.co.uk/) com uma funcionalidade exatamente correspondente à dos softwares que procurara anos antes. Cheguei a desenvolver alguns jogos demo (protótipo de um jogo completo) e um jogo que seria um jogo completo, não fosse uma contaminação por vírus de computador

[1] **Software livre**, segundo a definição criada pela Free Software Foundation é qualquer programa de computador que pode ser usado, copiado, estudado, modificado e redistribuído com algumas restrições. A liberdade de tais restrições é central ao conceito, o qual se opõe ao conceito de software proprietário, mas não ao software que é vendido almejando lucro (software comercial). A maneira usual de distribuição de software livre é anexar a este uma licença de software livre, e tornar o código fonte do programa disponível. (WIKIPEDIA, 2006). Disponível em: < http://pt.wikipedia.org/wiki/Software_livre>. Acesso em: 27 nov.2006.

dos arquivos que formavam tanto a parte visual (ilustrações e animações) quanto a parte funcional (scripts de programação) do jogo.

Por volta do mesmo período, tomei contato com outra classe de ferramentas essenciais para o ofício do artista / desenvolvedor de jogos: os programas de modelagem e animação 3D. A ferramenta específica foi o programa 3D Studio Max, na época pertencente à então Discreet (hoje chamada Autodesk). Eu já conhecia a computação gráfica e a animação 3D, de forma bem superficial, desde cerca de 1992, de revistas importadas e manuais de programas que adquirira em livrarias. Porém, assim como foi com as ferramentas de criação de jogos, não obtive êxito em minhas buscas de imediato, além de não dispor de um computador onde pudesse instalar os programas. Ou seja: do período que vai de 1992 a 2000, meu conhecimento acerca da modelagem e da animação 3D e do desenvolvimento de jogos era puramente conceitual e bastante superficial, devido à ausência de ferramentas com as quais pudesse experimentar, apreender a técnica e, em uma palavra, fazer.

Enfim, após mais 3 anos aprendendo a utilizar o 3D Studio Max de forma bastante desestruturada, esparsa e amadora – o único produto finalizado a que cheguei foi uma história em quadrinhos impressa, cujos elementos foram todos criados no 3D Studio Max, mas cuja estética e sofisticação deixou muito a desejar – migrei para o Maya, ferramenta concorrente do 3D Studio. Com o Maya, segui até 2004 com um aprendizado desorientado e "espasmódico": li inúmeros tutoriais na internet, porém não conseguia ter um conhecimento estrutural – e bem estruturado - acerca da animação 3D e da criação de jogos. Enfim, aprendia da mesma maneira como aprendeu a maioria dos profissionais pioneiros desta área aqui no Brasil. Com o Maya cheguei a produzir um curta de animação (*O Tédio do Abismo*), que ficou entre os finalistas do Festival da Imagem em Cinco Minutos, exibido na sala Walter da Silveira. Porém, me sentia refém

da falta de técnica e utilizava ainda os softwares como a maior parte dos usários comuns, quase sempre em conflito com a ferramenta e o seu incompreensível comportamento e procedimentos complicados e excessivamente técnicos. Resolvi então estruturar a minha pesquisa o máximo possível e criei (redigi) um cronograma de aprendizado estruturado, englobando todos os fundamentos da computação gráfica e da animação 3D e tradicional, tentando ser fiel ao formato de uma grade curricular de uma universidade americana qualquer de especialização em animação 3D e desenvolvimento de jogos. Adquiri os livros necessários e estabeleci um cronograma de leitura. Assim, desde 2004 até hoje venho aprendendo as diversas técnicas e teorias da área de forma organizada e aprofundada, com previsão para finalizar a formação pretendida (de artista 3D apto a criar jogos e curtas animados) em janeiro de 2007.

2. PONTO DE PARTIDA

2.1 OS EXPERIMENTOS ATUAIS

Tendo já acumulado uma base bastante sólida acerca dos fundamentos e práticas da criação de jogos para computador (em 2D e 3D), em fevereiro de 2006 adquiri um motor de jogos (programa que é a base de qualquer jogo para computador ou console), desenvolvido por um estúdio independente do Canadá (Gekido Design Group Inc). Apesar de seu valor acessível, é um motor de jogos capaz de desenvolver produtos do mesmo nível de sofisticação estética e jogabilidade dos jogos dos maiores estúdios do setor.

A minha meta para o futuro consiste em estabelecer, em Salvador, um estúdio de criação de jogos e ambientes virtuais interativos, que podem atender a diversos tipos de finalidades: artísticas - como instalações virtuais; arquitetônicas - como a recriação de construções modernas, sobrados, igrejas ou qualquer outra edificação; ecológicas – recriando ecossistemas de qualquer parte (e em qualquer era geológica) do Brasil, por exemplo; e educativas – através de qualquer um dos exemplos anteriores.

Porém, para se criar jogos interativos não basta ser um artista gráfico ou animador. A grosso modo, 50% de um jogo é composto pela sua arte (personagens, animações, texturas, cenários, efeitos animados etc.), mas 50% é composto de elementos geradores de interatividade – em geral arquivos de código-fonte[2] ou scripts programados[3]. Portanto, para desenvolver um jogo completo é necessário saber programar (dar instruções escritas à máquina numa linguagem que ela compreenda).

Apesar de já ter aprendido o básico necessário para programar em duas linguagens diferentes (Python e C++), não possuo a experiência e a desenvoltura de um programador profissional. Além, disso, até para um jogo simples, é necessária uma quantidade enorme de elementos de arte (texturas, animações, modelos, storyboards, manuais de personagens etc), bem como uma quantidade talvez ainda maior de código e de tempo dedicado à solução de problemas de interatividade. Logo, torna-se uma tarefa hercúlea desenvolver sozinho um jogo 3D de nível profissional e complexidade média. A equipe mínima ideal para se desenvolver jogos em 3D seria: um artista e um programador. Daí para o quadro de integrantes de um estúdio médio ou grande, os números variam mas a proporção é mantida. A única coisa que muda é a especialização

---

[2] **Código fonte** (**código-fonte**, ou até ***source code*** em inglês) é o conjunto de arquivos de texto escritos numa linguagem de programação (uma linguagem formal compreensível pela máquina) que forma a "essência" de um programa - sua funcionalidade. Cada linha de código é responsável por dar uma instrução (um input - entrada de dados) ao computador, fazendo com que ele execute um comando e retorne para o usuário algum tipo de resultado (output – saída de dados), formando assim uma experiência interativa. Um programa nada mais é do que um arquivo executável (composto de texto) originado ao se compilar o código-fonte, traduzindo-o para linguagem de máquina.

[3] **Scripts** são arquivos de texto, em alguma linguagem de programação, que acrescentam novas funções ao código-fonte, mas não o acessam diretamente. Diferente do código-fonte, que normalmente é compilado (ou seja: traduzido todo de uma vez só para a linguagem de máquina, formando um arquivo executável – um programa - e depois não mais modificado pelo usuário), os scripts são lidos e interpretados em tempo real pelo programa, toda vez que o mesmo programa é executado.

O código-fonte costuma ser implementado pelo desenvolvedor do programa e os scripts costumam ser adicionados posteriormente pelos usuários.

de cada membro em uma atividade específica dentro das etapas de produção de um jogo.

Portanto, além da finalidade principal de experimentar possibilidades de uso de elementos da linguagem cinematográfica, esta série de ambientes virtuais interativos, tem também o objetivo de demonstrar a minha capacidade técnica enquanto artista de jogos para possíveis parceiros programadores em uma instância futura.

Estes ambientes interativos são resultado (de experimentos) e "campo de testes" ao mesmo tempo. Eles surgem da necessidade de se testar a experiência de um jogo do ponto de vista do usário, tanto sob o critério estético quanto da interatividade. Diferente de um filme ou animação, o jogo é uma mídia interativa, logo questões de ergonomia, funcionalidade, eficiência etc. são de primeira ordem. Há muito de planejamento, design e testes exaustivos no processo de desenvolvimento de um jogo até sua forma final, existindo para isso até uma função (que em estúdios grandes converteu-se numa profissão): os *beta testers*, que são espécies de cobaias da qualidade interativa de um jogo em fase de protótipo. Assim, para se fruir um jogo a partir do ponto de vista do seu "espectador" final, e assim avaliar a eficiência de seus elementos estéticos e cenográficos – muitas vezes emprestados do cinema, é sempre necessário confeccionar alguns protótipos, como estes, antes de se lançar à empreitada de desenvolver um jogo completo.

3. O PROCESSO

3.1 PARALELOS ENTRE A CENOTÉCNICA POR TRÁS DO CINEMA E DOS JOGOS 3D

Ambiente virtual interativo exibindo uma paisagem urbana tipicamente européia (retirado do site http://www.game-stuff.com).

O produto consiste em alguns demos experimentais que, como foi dito, foram (e continuam sendo) campos de testes para utilização de elementos estéticos e cenográficos e a confirmação ou não de sua eficácia, em termos de resultado sensorial almejado.

Ao se explorar cada cenário com uma atenção mais analítica, nota-se o seu caráter eminentemente manufaturado. Assim como os cenários artificiais de cinema, e de certa forma como os cenários de teatro, há uma grande utilização de truques. Nos cenários de cinema (artificiais) e de teatro, alguns elementos são totalmente "falsos" (não existentes antes do filme / peça), confeccionados a partir de matérias-primas cuja forma e aspecto nada têm a ver com o produto final. Pedras falsas, feitas de fibra, isopor etc., árvores de fibra ou acrílico, névoa criada a partir de gelo seco. Porém, muitas vezes os objetos utilizados são reais, apenas deslocados de seu local de origem e modificados para atender às exigências das descrições visuais do roteiro do filme ou da peça. Já no caso de um jogo, o cenário é completamente artificial, pois o próprio espaço onde ele se desenvolve é artificial, virtual, não sendo sequer tridimensional de fato (apenas aparenta possuir 3 dimensões espaciais, mas existindo na verdade na superfície da tela do computador, que é plana, praticamente bidimensional, como uma folha de papel).

3.2 LÓGICA DA ECONOMIA DE DETALHES E TAXA DE QUADROS POR SEGUNDO

Modelo low-poly com simplicidade de detalhes evidenciada pelas linhas da sua estrutura – o detalhe está nas texturas (retirado do site http://www.game-stuff.com).

Outros paralelos podem ser apontados entre os cenários de cinema (ou teatro) e de jogos, como por exemplo: em ambos, *o que não será mostrado não precisa ser fabricado*. Assim, grande parte das cidades e ruas cenográficas do cinema exibem casas que na realidade se resumem a apenas uma fachada (não sendo, então casas, na

realidade). O mesmo ocorre com os jogos 3D: casas que não há razão para que se permita que o jogador explore podem também se resumir a uma fachada ou, no caso de o jogador ser capaz de explorar o terreno em torno, são estruturas ocas. A lei que vale é a mesma: só confeccione o que será visto, porque afinal, o que não será visto não tem função considerando-se os objetivos de um filme ou um jogo. No caso dos jogos, esta regra não só é observada, mas perseguida tenazmente, pois há um fator limitador sempre presente e que nunca deve ser esquecido, em qualquer que seja a etapa da produção: a performance dos computadores do público-alvo.

Há uma grande diferença, neste aspecto, entre as imagens de cinema e as imagens de um jogo: no cinema, as imagens são todas pré-confeccionadas. Portanto no momento da exibição, a quantidade de quadros por segundo mostrados é sempre constante (24 quadros/s), a menos que o projetor esteja danificado, é claro. Isso faz com que a imagem exiba sempre a mesma qualidade de movimento. Se de repente um trecho de um filme começasse a ser exibido a uma taxa de 6 quadros/s, por exemplo, o espectador instantaneamente notaria uma deterioriação na fluidez dos movimentos do que está sendo mostrado e a ilusão de movimento contínuo com certeza seria perdida. Já num jogo 3D, as imagens são confeccionadas durante o momento da exibição, o tempo todo, em tempo real. A peça do computador responsável por esse processo de confecção de imagens é a placa de vídeo, ajudada, dependendo da ação performada, pela memória RAM e pelo processador central. Digamos que a cada milésimo de segundo, caso haja uma mínima modificação na imagem a ser exibida - como numa mínima inclinação de uma câmera -, essa informação é processada pela placa de vídeo e convertida numa nova imagem, neste caso, um pouco inclinada.

Portanto, no momento da exibição de um jogo, a taxa de quadros por segundo pode variar - e efetivamente varia – a cada momento. Assim, esse tem sido o

principal divisor de águas entre a qualidade de imagem dos jogos e dos filmes de animação 3D (cujas imagens, assim como as do cinema, são pré-confeccionadas). Quando se trata de jogos, a única forma de manter a taxa de quadros por segundo das imagens num nível em que a ilusão de movimento seja mantida, "enganando" a percepção[4] visual do usuário é limitar a quantidade de detalhes exibida na tela a cada instante. Daí a suma importância de se eliminar todo e qualquer tipo de objeto – ou parte de um objeto – que não vá ser visto de forma alguma durante a imersão no jogo. Os exemplos são incontáveis, como: sofás encostados na parede não precisam ter a parte de trás (nem tampouco a parte de baixo), se o usuário não tiver como empurrá-lo para ver, árvores cujos galhos mais altos são ridiculamente simplificados, em termos de modelagem, caso o usuário não tenha como escalar a árvore, Pedras sobre o solo que não possuem a parte de baixo (a parte em contato com o solo). Enfim, quase todos os objetos cenográficos seguem a lógica das fachadas das cidades cenográficas de um filme, quase sempre sendo ocos e "desmembrados", possuindo apenas os pedaços que serão vistos.

O limite mínimo considerado "saudável" pelos estúdios para a experiência visual de um jogo, varia entre 24 e 30 frames (quadros) por segundo. Mas uma taxa considerada boa fica em torno de 40 a 60 quadros/s para o geral de um jogo – garantindo assim uma margem boa para que em momentos em que os detalhes de repente se tornem excessivos (como no caso de uma explosão, por exemplo) a taxa de quadros não fique em menos do que 24 quadros/s. Segundo estudos, a persistência da visão – e a conseqüente ilusão de movimento contínuo – tem como limiar a razão de 12

[4] *The Persistence of Vision with Regard to Moving Objects* ("A Persistência da Visão com Relação a Objetos em Movimento"), artigo de Peter Mark Roget, explicava que imagens eram retidas pela retina do olho humano por uma fração de segundo antes de serem substituídas pelas imagens seguintes. Se a sucessão fosse suficientemente rápida, o observador tinha a impressão de movimento, mesmo que estivesse olhando para imagens paradas.

quadros/s (padrão mais utilizado na indústria de animação tradicional, desenhada à mão), porém este é um limiar arriscado demais para jogos, pois o simples fato de o jogador virar bruscamente a câmera para olhar para outro lado, faria essa taxa cair em várias unidades, chegando talvez a 5 quadros/s ou até menos. É fácil entender o porquê: numa panorâmica rápida (rotação rápida da câmera em torno do seu próprio eixo) a quantidade de transformação na imagem a ser exibida é imensa.

Logo, de acordo com o que acabamos de entender, a quantidade de informação nova a ser processada pela placa de vídeo do computador aumentará bruscamente durante a panorâmica. Por conseqüência, a taxa de frames por segundo cairá bastante, caso atinja os limites de velocidade de processamento de imagens da placa de vídeo.

Este comportamento, apesar de à primeira vista parecer um detalhe meramente técnico, de pouca importância para a estética e os artistas de jogos, como foi dito tem sido o grande limitador da indústria de jogos e talvez a única coisa que ainda impeça os jogos de terem uma qualidade, riqueza de detalhes de imagem e características fotorrealísticas idênticas às do cinema. Enfim, é fácil perceber que será uma questão de tempo para que os jogos 3D se tornem espécies de filmes interativos, no que diz respeito à experiência visual do jogador (em termos de som e efeitos sonoros, isso já é uma realidade). Isso está intrinsecamente ligado à evolução dos computadores e ao aumento da capacidade de processamento (de imagens e dados não-visuais) que eles vêm sofrendo a cada ano.

Este paradigma de limitação da riqueza de detalhes dos objetos e personagens do jogo é conhecido como *low*-poly, que é uma abreviação de *low polygon count* – "baixa quantidade de polígonos", devido ao fato de que os objetos de um jogo são formados essencialmente (e "por trás dos panos") de polígonos simples, como

veremos mais adiante. Este, que pode ser considerado o paradigma atual da indústria de jogos 3D (desde o seu surgimento até hoje), acabou por definir o modo de fazer e uma enorme quantidade de técnicas – que forma o repertório de um artista de jogos 3D. Para se ter uma idéia, esse é o primeiro quesito a ser observado por um diretor de arte de um estúdio de jogos ao entrevistar alguém para uma vaga de artista 3D: se ele é capaz ou não de gerar modelos (objetos tridimensionais) "limpos".

Por modelos limpos entende-se uma série de coisas: modelos que sejam feitos com o mínimo possível de detalhes – só o essencial – mas ao mesmo tempo sem perderem o apelo visual; que não exibam defeitos ou propensão a apresentar defeitos quando colocados dentro do jogo; que façam uso inteligente e criativo das limitações – apesar de serem bem simples, pareçam complexos e detalhados – para isso o artista 3D deve ter uma grande desenvoltura com a utilização de texturas e detalhes *fake* ("falsos"). Um exemplo fácil: um modelo de uma caixa feita de lâminas de de madeira unidas com pregos: um artista 3D profissional muito provavelmente iria modelar um cubo totalmente liso e simples. Os detalhes da madeira, tarjas afixadas, pregos e inclusive os vestígios da "longevidade" da caixa (cortes, sujeiras, desgastes etc) seriam todos simulados na textura a revestir o cubo.

3.3 O PAPEL CRUCIAL DAS TEXTURAS NO JOGO

Modelos 3D vistos dentro do programa de modelagem, com sua geometria sendo evidenciada pelas linhas em destaque dos contornos – note a simplicidade das suas formas (retirado do site http://www.game-stuff.com).

Aqui vemos os mesmos modelos, sem destaque para os contornos da geometria, na sua forma final dentro do jogo, apoiada fortemente nas texturas (retirado do site http://www.game-stuff.com)

Assim somos introduzidos a outro grande fundamento da criação de arte para jogos 3D: a (inteligente) utilização de texturas. Assim como o axioma relacionado à quantidade de quadros por segundo (leia-se: fluidez dos movimentos) que vimos antes (*o que não será mostrado não precisa ser fabricado)*, agora chegamos a outro lema crucial: *o que puder ser conseguido utilizando texturas, não precisa ser modelado.* Que poderia ser colocado mais explicitamente: o que puder ser discretamente simulado usando-se texturas, não precisa estar presente como um detalhe tridimensional esculpido no modelo 3D. O exemplo da caixa de madeira se encaixa perfeitamente nesse parâmetro. Uma forma didática de se entender as texturas de um jogo é imaginá-las como papéis de parede colados sobre os objetos, envolvendo-os.

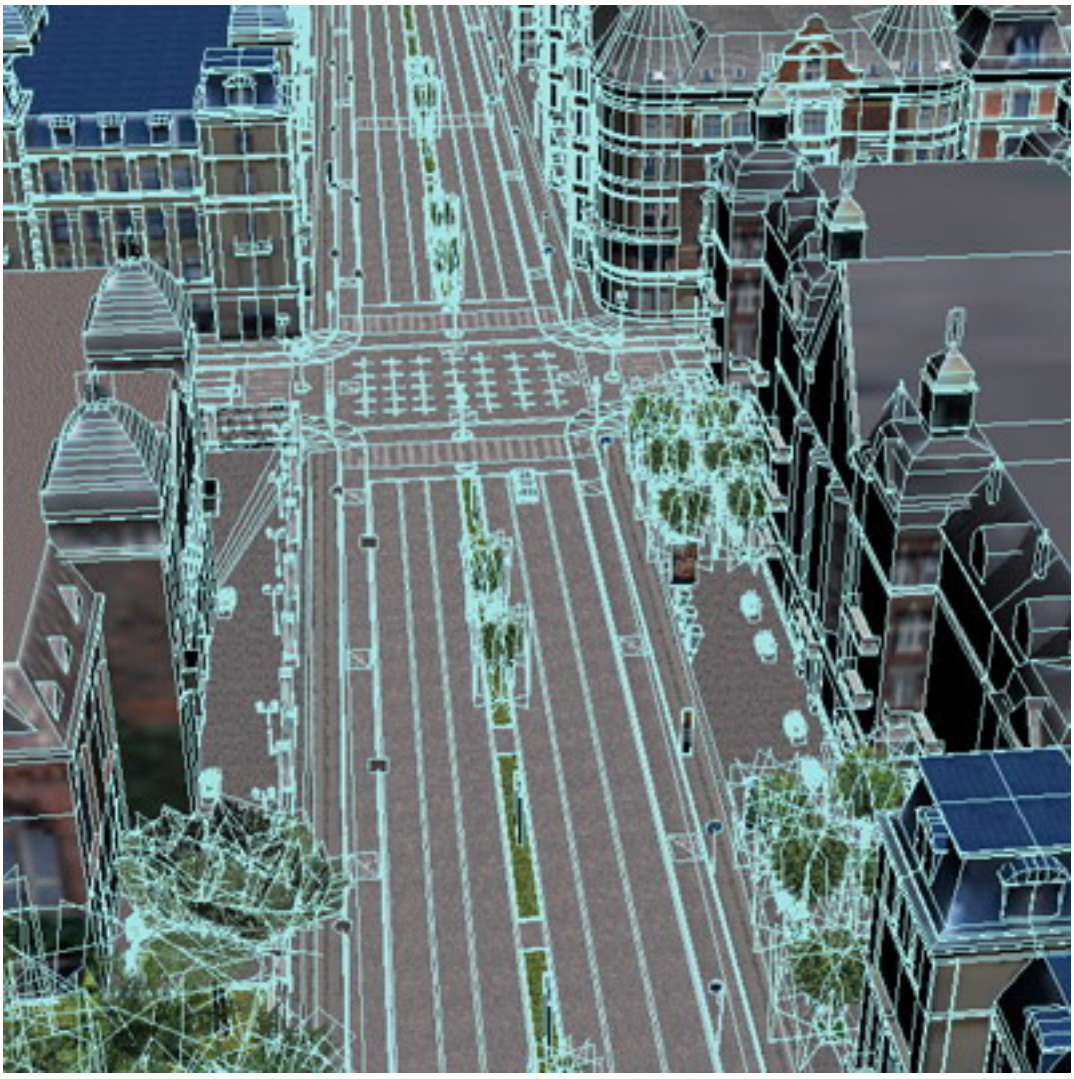
As texturas não são arquivos de aspecto tridimensional: de fato são bidimensionais como qualquer pôster ou capa de livro. Mais ainda possuem em comum com um pôster: são sempre retangulares. Mesmo que assim não pareçam se vistas dentro do jogo. Os arquivos de texturas são de fato chamados de *texture sheets*, ou folhas de texturas. Por exemplo: vemos no jogo uma laranja com detalhes de rugosidade em relevo, um pequeno entalhe e uma etiqueta com um preço colada nela.

Porém se desmontarmos este objeto, veremos que se trata de 3 elementos: 1) um polígono com faces suficientes para parecer uma esfera, porém liso e sem detalhes; 2) uma arquivo de imagem, retangular como uma foto, contendo a textura da laranja com a etiqueta do preço e o entalhe estampados nela. Esta é a textura de cor (*color map);* 3) uma textura de relevo cujas cores não correspondem às vistas na laranja, representando apenas informações sobre o relevo: no caso de uma textura/mapa de relevo (*bump map)*, os pontos mais elevados do relevo são mais claros e os menos elevados são escuros, formando uma imagem em preto-e-branco. No caso de um mapa das normais do objeto (*normal map*), de função semelhante ao mapa de relevo, a

imagem apresenta tons de cores de aspecto psicodélico baseado em quatro cores simples. Tanto o *bump map* quanto o *normal map* serão intepretados pelo programa e resultarão, dentro do jogo, numa ilusão de detalhes em relevo sobre a laranja.

Com isso, começamos a entender que o trabalho de um artista 3D envolve muita criatividade para a solução de problemas de natureza técnica (limitações a "driblar" e defeitos a evitar), poderia-se dizer até que, na maior parte dos casos, de natureza realmente cenotécnica. Isso acaba por definir bastante o perfil dos bons artistas de jogos: profissionais criativos e que gostem de desafios e de solucionar problemas. Isso acaba por se tornar um incômodo para artistas com menor inclinação técnica e de hábitos desorganizados, que passam a "lutar" com a máquina e os programas ao invés de compreenderem sua lógica de utilização (que no fundo é muito simples e exata) e trabalharem da forma para a qual o design destas ferramentas foi pensado.

3.4 OS FUNDAMENTOS DO ESPAÇO TRIDIMENSIONAL E DA REPRESENTAÇÃO DE OBJETOS 3D



A mesma cena da primeira imagem, vista sob sua forma original dentro do programa de modelagem 3D (retirado do site http://www.game-stuff.com).

Entender os fundamentos da tecnologia e da ciência por trás da representação de espaços e objetos tridimensionais (3D) é de grande ajuda para uma compreensão mais clara de muitos porquês técnicos e de grande valia para entender e poder contornar as limitações e solucionar alguns problemas. Seguindo uma breve explicação geral, teremos uma outra visão, muito mais clara do assunto.

Na sua essência a tecnologia de criação de imagens 3D é toda baseada em geometria euclidiana, trigonometria e no sistema de coordenadas cartesiano. Conhecimentos milenares que tomam como base o triângulo (como o Teorema de Pitágoras e algumas fórmulas que derivaram dele) e o sistema de coordenadas espaciais cartesiano (eixos x, y e z, para representar largura, altura e profundidade – as célebres três dimensões) foram conjugados de diversas formas para realizar cálculos de vetores, para achar sua posição, comprimento, direção etc.

Para entender o processo de confecção de uma imagem 3D fica muito mais fácil se dispensarmos uma dimensão e tentarmos entender primeiro o processo de geração de uma imagem 2D, que possui apenas largura e altura, como uma folha de papel (a título de exemplo, porque a profundidade de uma folha de papel é realmente desprezível).

A imagem 2D mais simples seria um ponto. O ponto é, digamos, o axioma visual, o "átomo virtual", a menor unidade desenhada numa tela de computador, de onde partem todas as estruturas mais complexas a serem representadas. Numa tela de computador, o menor ponto que se conseguir obter é equivalente a 1 pixel. Um monitor com uma definição de imagem de 800 x 600 pixels, nada mais é do que um aparelho eletrônico capaz de desenhar 800 pontos da esquerda para a direita e 600 pontos de cima para baixo. Se quisermos, nesta tela de 800 x 600, desenhar um ponto preto bem no seu centro, instruímos o computador, em linguagem de programação, a desenhar um objeto – ou seja, preencher um pixel com cor - nas coordenadas 400, 300 (ou seja: 400 unidades a partir da esquerda e 300 unidades a partir do topo). Temos então um ponto preto num espaço 2D. A origem deste espaço (posição 0, 0) seria o canto superior esquerdo, seguindo nosso raciocínio. E a posição 800, 600 seria o canto inferior direito.

Depois do ponto, o elemento fundamental seguinte seria uma reta. Para o computador, qualquer reta preta será simplesmente uma seqüência de pixels preenchidos com preto, em alguma direção. Para o programador, bastará fornecer a posição de dois pontos diferentes e instruir o computador a criar uma reta entre eles. Depois, da reta, o elemento geométrico mais simples é um triângulo, formado de 3 pontos e de retas entre eles. Por essa simplicidade, o triângulo é a base de todos os objetos 3D.

Não importa quão complexo seja este objeto: se modelarmos o Davi de Michelangelo em 3D, para o computador ele será uma nuvem de milhares de pontos diferentes - cada um com uma posição no espaço 3D - equivalentes a pontos na superfície do Davi original. O computador será então instruído a ligar estes pontos com retas, sempre em triângulos. Com isso, entendemos de uma maneira superficial, os elementos básicos que formam a geometria 3D: pontos, retas, triângulos e todos os outros objetos mais complexos, desde pirâmides, cubos e cilindros a esculturas, veículos, montanhas e todos os outros. Um cubo é formado, portanto, de quatro pontos ligados por retas e preenchido (um quadrado / dois triângulos) e depois mais quatro pontos deslocados no eixo da profundidade. Ao ligar todos estes pontos com retas e preenchermos, teremos um cubo *renderizado* (confeccionado em uma imagem).

Porém, até aqui teríamos apenas geometria 3D com cores “chapadas”. Só que como vimos antes, a arte 3D para jogos deve muito às texturas. Continuando a partir do exemplo do triângulo, para acrescentarmos a ele uma textura de tijolos, por exemplo, o programador instrui o computador (através de diversos cálculos e fórmulas algébricas e trigonométricas) a preencher apenas os *pixels* que estiverem na área entre os 3 pontos que formam o triângulo e em seguida fornece a ele uma imagem, como por

exemplo, uma foto frontal de uma parede de tijolos, ordenando que ele preencha o triângulo com esta foto.

Temos aqui 2 tipos de objetos digitais de natureza bastante distinta: um vetor (o triângulo) e um bitmap (a foto dos tijolos), que são as 2 matérias-primas fundamentais da criação de objetos para jogos, assim como para filmes e comerciais de animação 3D.

Um vetor é um objeto formado, essencialmente, de pontos com posições definidas (como o triângulo do exemplo) e uma informação de cor de preenchimento simples e chapada.

Já um bitmap, é uma mapa de pixels - um mosaico onde cada pixel possui uma informação de cor dizendo a intensidade de "vermelho", "verde" e "azul" (conhecido como RGB, sigla para *red*, *green* e *blue*). Se tivermos uma foto de 800 x 600 pixels, teremos uma informação RGB para cada pixel. Um pixel completamente branco nesta foto, por exemplo, terá basicamente 2 valores: um valor RGB igual a 255, 255, 255 (R=255 G=255 B=255) e um valor para a sua posição na tela, como 400, 300.

Não cabe aqui entrar no mérito de todos os pormenores dos cálculos e procedimentos por trás da confecção da imagem 3D, apenas entender de uma maneira superficial e didática como isso é possível. Além do que foi dito, basta dizer que a trigonometria também é muito utilizada para criar a ilusão de perspectiva necessária ao espaço 3D, a ilusão de objetos mais próximos e mais distantes. Pois no final de todos estes processos matemáticos sob uma lógica tridimensional, tudo isto será "projetado" (desenhado) numa tela retangular 2D, que é o monitor. Uma metáfora visual didática é imaginar que observamos uma paisagem urbana através de uma janela de vidro. Pegamos uma caneta hidrocor e riscamos o que vemos sobre o vidro. Esse desenho é uma projeção em 2D, plana, do espaço 3D lá fora. Assim termina a cadeia de cálculos

de confecção de imagens do computador na tela diante dos nossos olhos. Há uma infinidade de elementos que mereceriam ser explicados que compõem a matéria-prima das imagens 3D, mas não está dentro do escopo desta memória explicá-los todos ou a fundo, tarefa que demandaria uma enciclopédia. Com o que já vimos, temos uma base simples porém de grande valia para continuarmos com a compreensão do produto.

3.5 O DESIGN DE ÁUDIO PARA JOGOS E OS SONS 3D

Com relação ao áudio para jogos 3D, não há muito de específico comparando-se a outras mídias que se utilizem de áudio gravado, como o cinema ou o vídeo. Algumas similaridades são encontradas, bem como algumas discrepâncias.

Algumas similaridades pertinentes de serem citadas: é comum a presença de uma trilha sonora musical que é tocada à revelia do usuário, independente de qualquer ação sua (fato que não é regra, já que um evento dentro do jogo, como por exemplo tocar num objeto, pode ser programado para disparar uma trilha sonora musical qualquer); a utilização de efeitos sonoros, muitas vezes artificiais e gravados por sonoplastas e designers de áudio; o papel de destaque da trilha sonora quando se trata de criar, modificar, ampliar ou reduzir a intensidade dramática e o significado das imagens mostradas; a utilização de efeitos sonoros 3D, que simulam possuir uma fonte de emissão com localização espacial definida, que vêm sendo utilizados comumente há algum tempo no cinema, a partir da implantação do sistema Dolby Surround – se um carro atravessa a tela em direção à esquerda e se choca com um caminhão, ouvimos o

som da colisão como se estivesse vindo da esquerda do cinema. De maneira similar utiliza-se o áudio 3D nos jogos.

Provavelmente a diferença principal entre a forma como nos é mostrado o áudio do jogo, em relação ao cinema, está justamente no seu caráter interativo – algo que fica difícil até de obter comparação, já que o cinema simplesmente não é interativo - . Durante a produção de um jogo, cada arquivo de som é associado a um objeto 3D ou evento originado da interação entre o usuário e o ambiente do jogo, em geral através de linhas de código com instruções especialmente projetadas para a manipulação do áudio. Se o jogador anda sobre um chão de terra, é possível instruir o motor de jogos (o programa por trás do jogo) a acionar um som de passos sobre terra (previamente gravado). Já se ele andar sobre um chão de madeira, o motor de jogos aciona um som de passos sobre madeira. Todas essas interações são previamente programadas durante as etapas de desenvolvimento e testes do jogo, como tudo o mais, e integrado diretamente ao código-fonte ou ligados indiretamente através de scripts. Porém, ao final, em ambas as opções todo o código será compilado e encapsulado num arquivo executável fechado (em linguagem de máquina), pronto para ser instalado como qualquer outro programa.

3.6 A ANIMAÇÃO DE PERSONAGEM E A CONSTRUÇÃO DOS RIGS PARA ANIMAÇÃO

Como é sabido que a maioria dos jogos 3D apresenta de um a vários personagens no jogo – que são incorporados pelos jogadores, às vezes em jogos que comportam centenas, milhares de jogadores (gênero conhecido como MMOG, de

Massive Multiplayer Online Game, Jogo Online Multijogador Massivo) – acreditei necessário experimentar também com a modelagem e a animação de personagens 3D. Estão presentes nos experimentos alguns personagens, todos animais, notadamente: dois tipos de corvo (um medianamente detalhado e outro menos detalhado, para ser visto apenas ao longe, voando em altitudes elevadas), um cavalo, uma marmota e um tubarão.

Para cada um destes personagens animados foram necessárias todas as etapas normalmente presentes na implementação deste tipo de técnica / linguagem. Uma extensa pesquisa por imagens de referência destas espécies de animais – estáticas e exibindo seus movimentos mais comuns -; O aprendizado de técnicas de animação tradicional (coisa que venho pesquisando já há alguns anos, experimentando e tendo inclusive já realizado três trabalhos profissionais como animador), dos 12 princípios da animação, como postulados pelos excelentes animadores dos estúdios Disney do passado, que estabeleceram inúmeros procedimentos e descobertas de técnicas, entre outros; O aprendizado de técnicas e da teoria envolvendo o processo de construção dos *rigs* (as armaduras ou controles digitais necessários para se manipular os personagens 3D de forma intuitiva e mais simples, análogos aos controles de uma marionete real); e o aprendizado de diversas técnicas e noções de soluções de problemas envolvendo o processo de importar e exportar arquivos de personagens animados do programa de modelagem e animação utilizado (no caso, o Maya) para o motor de jogos Beyond Virtual – processo este que costuma ser um pouco complicado e propenso a apresentar alguns bugs cuja origem é difícil de determinar, por envolver muitos detalhes técnicos relacionados à conversão do arquivo de um formato nativo do Maya para o outro, um formato que seja possível ser lido e exibido corretamente pelo Beyond Virtual.

Infelizmente, todos os os personagens que foram animados, no final das contas apresentaram algum tipo de bug visual, às vezes mínimos, porém às vezes

críticos e impossíveis de se ignorar, impedindo a correta apreciação por parte do jogador / usário. Mas vale dizer que todos, sem exceção, foram de responsabilidade do Beyond Virtual e de suas atuais falhas técnicas com relação à importação de personagens animados para o ambiente do jogo. Este fato pôde ser facilmente comprovado, pois nenhum dos personagens animados apresentava defeitos no seu programa de origem, o Maya. Os defeitos se faziam notar quando os colocava dentro do Beyond Virtual. Isto até poderia ter sido gerado por algum descuido meu enquanto artista 3D, ao gerar modelos "sujos", mas essa possibilidade foi descartada após exaustivos testes utilizando todas as possíveis correções e configurações diferentes.

Enfim, alguns personagens estão disponíveis na sua forma animada (apesar dos bugs) dentro dos ambientes virtuais interativos, com exceção do cavalo, que acabou ficando estático, pois sua animação sofreu um bug extremo e não pode ser importada, na verdade. Além do cavalo, um personagem humano que pretendia que estivesse presente (animado) dentro dos ambientes virtuais também não pôde ser inserido nos produtos finais por apresentar defeitos por demais bizarros visualmente, quando dentro do motor de jogos Beyond Virtual.

Porém, todos estes personagens estão disponíveis em alguns arquivos nativos do maya e pequenos vídeos que capturei para uma posterior apreciação e analise, caso sua consulta seja desejada.

4. O PRODUTO

4.1 A FLORESTA, O DESERTO E A ILHA: ALGUMAS LOCAÇÕES BEM DISTINTAS

Com um foco claro nas características cenográficas e de ambientação, foram desenvolvidos alguns ambientes que exibem características geográficas distintas: uma floresta coberta de neve, um deserto montanhoso e uma ilha tropical (retratada durante o dia e à noite). Antes de mais nada, uma séria ressalva é necessária: o motor de jogos utilizado para criar os ambientes (Beyond Virtual versão 0.95, desenvolvido pelo estúdio Gekido Design Group Inc.), apesar de ser um programa formidável em meio à categoria de motores de jogos para desenvolvedores independentes[5], ainda estava em fase de testes quando os utilizei para criar os ambientes e por isso, como acontece com qualquer outro software ainda em versão beta[6], apresentava vários defeitos que geraram comportamentos indesejados e imprevisíveis (bugs) na interatividade dos ambientes.

[5] Ou seja: dotados de baixo orçamento, que custeiam jogos e protótipos com recursos "do próprio bolso", sem qualquer tipo de investidor ou vínculo com grandes estúdios.
[6] Etapa, no decorrer do desenvolvimento de um programa, logo depois da etapa chamada de Alpha, quando diversos defeitos são corrigidos, e antes da etapa chamada de Gold, quando uma versão final está pronta para ser lançada.

A bem da verdade, o Beyond Virtual, apesar de já estar na sua versão 1.0, ainda apresenta diversos bugs e, como os próprios desenvolvedores deste motor de jogos colocaram muito bem, “até o windows, quando foi lançado, apresentava uma lista de cerca de 1.000 bugs”. Apesar da quantidade de bugs ser algo que varia bastante entre produtos de uma empresa para outra, é algo que faz parte da realidade de programas que ainda estão sofrendo muitas modificações e melhorias, como é o caso do Beyond Virtual. Como conseqüência disso, os ambientes que desenvolvi apresentam alguns bugs, que tentei de diversas formas resolver, através de contato estreito com os desenvolvedores, no fórum de discussões e no serviço de suporte técnico através de email do Beyond Virtual.

Alguns bugs foram solucionados, mas outros permaneceram. Porém, isso nunca chegou a ser algo por demais desgastante, pois sempre tive claro que isso faz parte do processo e que os ambientes são protótipos experimentais e não produtos finalizados prontos para serem lançados no mercado de entretenimento. A sua função foi cumprida, de servir como arena de testes e verificação de alguns resultados estéticos, bem como para apontar pontos a serem melhorados e pesquisados mais a fundo.

Os elementos constituintes destes ambientes podem ser, de certa forma agrupados em algumas categorias abrangentes:

- Texturas de cenário – como a textura de terra e pedrinhas do deserto;
- Texturas dos personagens – como a textura do corvo, exibindo suas penas;
- Texturas de efeitos visuais e climáticos – a textura da fumaça etc.;
- Modelos 3D de personagens - no caso destes ambientes, alguns tipos de animais;
- Modelos 3D de terreno;

- Modelos 3D de objetos cenográficos – como pedras, arbustos etc.;
- Modelos 3D de colisão dos objetos – invisíveis, responsáveis por evitar que o usuário atravesse os objetos, criando a sensação de materialidade / substancialidade;
- Animações dos personagens – como o corvo voando ou o tubarão nadando;
- Animações de efeitos visuais e climáticos – como fumaça e neve;
- Menus – exibidos ao se iniciar ou encerrar a exploração dos ambientes;
- Fontes utilizadas nos menus;
- Scripts – arquivos de texto contendo código que definem certas características estáticas e interativas dos objetos e personagens exibidos;
- Trilha sonora musical – no caso da Floresta dos Sussurros (*Tristão e Isolda*, *Prelúdi*o, de Wagner);
- Efeitos sonoros – como o som do mar na ilha e o som de corvos grasnando no deserto.

Alguns elementos constituíram um grande volume de material acumulado, mas não chegaram a estar presentes nos componentes finais do jogo: estes são os incontáveis arquivos de imagem e som recolhidos durante a etapa de pesquisa que precedeu a produção de cada um destes ambientes. A maior parte são fotos e pequenos loops[7] de áudio representando sons de animais e sons ambientais, como vento e mar. Por terem sido pesquisados e armazenados de uma forma organizada e criteriosa, estes elementos que compõem o material de referência colhido durante a pesquisa poderão ser

[7] Trechos de certa duração, em geral curtos, onde o aspecto sonoro do começo coincide perfeitamente com o do final, permitindo que sejam repetidos quantas vezes forem necessárias sem causar uma mudança brusca no áudio, dando a impressão de um trecho contínuo.

utilizados muitas outras vezes, como já vêm sendo aliás. Antes de criar a floresta, por exemplo, acabei por criar um enorme banco de fotos de árvores, organizados em diferentes pastas baseadas na espécie de cada uma, cada uma contendo em geral quatro fotos: uma foto mostrando toda a árvore, da base ao topo; um close-up do tronco, para obter sua textura com boa qualidade; uma foto de um galho com folhas; e um close-up de apenas uma folha, para o caso de necessitar de uma textura mais definida ainda. E no caso de árvores decíduas, uma foto de seu aspecto outonal.

Outro exemplo interessante é o da textura de terra do deserto: ela havia sido recolhida muito antes, durante a pesquisa para a criação da ilha, mas tinha sido descartada do ambiente da ilha por não satisfazer às características visuais que eram necessárias para representar uma areia como a que se pretendia.

Além do material de pesquisa, nota-se, ao explorar os ambientes, que também modelos 3D presentes em um ambiente foram utilizados em outro, comos os corvos, a carroça e a nuvem de fumaça. Esta é uma característica notória da criação de arte para jogos: muita coisa é reutilizável em jogos que se venha a fazer depois, necessitando muitas vezes apenas de algumas modificações, em alguns casos mínimas. Se um modelo previamente produzido de um corvo adequar-se à arte de um outro jogo agora sendo feito, não há porque não reaproveitá-lo, apenas modificando o tom geral de suas texturas ou adaptando um pouco o design das suas formas. A reutilização de modelos e texturas prévios é uma prática comum na indústria de criação de jogos 3D.

5. CONSIDERAÇÕES FINAIS

A produção dos ambientes virtuais pode ser considerada uma experiência bem-sucedida, pois serviu ao seu propósito de confirmar alguns experimentos visuais relacionados à utilização de elementos normalmente associados ao cinema (principalmente à sua cenografia) bem como descartar outros como não sendo de muita eficiência. Muitos outros elementos estéticos pertencentes ao cinema ainda podem ser testados e verificados em ambientes virtuais como estes, sendo sempre de grande valia a experiência obtida após sua produção, para o artista de jogos 3D, por fazer com que este adquira mais confiança nas suas escolhas e crie as bases de um repertório de técnicas narrativas visuais (óbvias ou apenas sugestivas) o qual sempre poderá ser aprofundado e expandido.

Tendo uma base clara numa lógica de produção *low-poly*, assim como nos outros fundamentos atuais do modo de produção que para jogos 3D, podemos exercitar a nossa criatividade de diversas formas para obter os resultados almejados usando "atalhos" que dificilmente serão notados no produto final. Na própria produção destes ambientes virtuais interativos, alguns problemas foram solucionados apenas pelo fato de possuir uma base clara e sólida no modo de produção *low-poly* e suas adjacências somado ao exercício de uma liberdade criativa que é inclusive estimulada em muitos

estúdios de criação de jogos 3D (por ter sido a maneira como foi possível obter resultados estéticos expressivos e eficientes em muitos dos jogos produzidos desde o início da indústria de jogos 3D para computador e console).

Não só alguns problemas técnicos relacionados a estes ambientes como problemas associados a protótipos de outros usuários do motor de jogos Beyond Virtual, que por uma ajuda da minha parte puderam ultrapassar certas barreiras técnicas que os estavam levando a naufragar em seus progressos (como por exemplo Ken Donnovan, um usuário do Beyond Virtual que está desenvolvendo um simulador de vôo ambientado na I Guerra Mundial, onde o jogador pilotará biplanos alemães,franceses, ingleses etc). O contrário também é verdade: inúmeras vezes fui ajudado em questões técnicas por outros usuários do fórum e pelos desenvolvedores do Beyond Virtual, principalmente em problemas relacionados à parte da programação de alguns comportamentos interativos, em geral ocasionados pela minha falta de experiência quanto à criação de scripts de programação e à sintaxe da linguagem Angelscript (linguagem de programação utilizada para se criar scripts para o Beyond Virtual).

Alguns destes problemas técnicos onde conhecimentos mais aprofundados de programação para jogos se faziam necessários acabaram por não ser solucionados, o que me levou a descartar alguns elementos de interatividade que pretendia que estivessem presentes nos ambientes virtuais desenvolvidos. Porém, tendo em vista que até aqui em geral tenho sido capaz de pesquisar o conhecimento necessário em alguns locais da internet e em seguida aprendê-lo, creio que a não-solução de muitas pequenas questões relacionadas à parte de programação aconteceu devido à falta de uma documentação adequada nos manuais do Beyond Virtual (ainda bastante incompletos e em fase de implementação). Diversas vezes em que uma consulta a um procedimento específico no manual poderiam ter resultado em uma solução, não foi possível fazê-lo,

devido a esta falta de documentação. Esta deficiência do produto Beyond Virtual tem sido até aqui apontada por diversos usuários que assim como eu, foram *beta testers* deste motor de jogos e vêm acompanhando sua evolução e apostando na correção de seus bugs daqui para o próximo ano – que é algo que provavelmente irá ocorrer gradualmente, como já vem ocorrendo.

Porém, estas dificuldades quanto à parte da programação dos eventos interativos foi positivamente encarada como geradora de uma demanda por alguns ações futuras para o propósito de se implementar um estúdio de criação de jogos e ambientes virtuais interativos (conhecidos como *VR*, sigla para *Virtual Reality*, realidade virtual), mais especificamente duas ações: 1) continuar com o aprendizado de programação, principalmente programação para jogos – algo que já venho planejando e pesquisando os livros necessários para isto e uma instituição de ensino online (ensino à distância) disponível na internet mediante pagamento de matrícula e mensalidades como qualquer estabelecimento de ensino privado -; 2) contactar e firmar parceria com um ou mais programadores. Uma parceria por projeto, caso não seja possível pagar pelos seus serviços ou pagar pelos seus serviços no caso de dispor do capital necessário, seja originado de recursos próprios ou de algum tipo de edital ou lei de incentivo à produção de jogos ou produtos audiovisuais em geral.

Ficou clara para mim a falta de alguns experimentos que buscava implementar nestes ambientes virtuais – principalmente ligados à animação de personagens e à configuração de uma opção de câmera não-subjetiva onde o personagem vivido pelo jogador / usuário seria visto logo em frente à tela, como em qualquer outro jogo visto na terceira pessoa . Mas, novamente, estes experimentos mais elaborados utilizando animação de personagens não foram possíveis devido a bugs relacionados a animação no Beyond Virtual e em qualquer protótipo ou jogo gerado a

partir dele no seu estado atual de desenvolvimento. Porém, nada impede que sejam realizados daqui para a frente, pois os desenvolvedores deste motor de jogos já estão cientes destes bugs mediante uma série de *feedbacks* dos usuários no fórum do Beyond Virtual e no serviço de suporte técnico via email.

Ao final de todos estes experimentos, entendo que foi um aprendizado excelente, por tratar-se de uma espécie de simulação em pequena escala do que é realmente a realidade de se produzir um jogo e a partir dela estabelecer estratégias para a aproximação de problemas futuros relacionados tanto à parte técnica quanto à utilização de elementos cenográficos e em geral utilizados pelo cinema.

REFERÊNCIAS

ALIAS LEARNING TOOLS. **Discover the Game with Alias**. Born Digital, Inc., 2005.

_____. **Learning Maya 6: Dynamics**. Toronto: Sybex, 2004.

_____. **Learning Maya 6: Maya Unlimited Features**. Toronto: Sybex, 2004.

_____. **Learning Maya 6: Rendering**. Toronto: Sybex, 2004.

_____. **The Art of Maya: An Introduction to 3D Computer Graphics**. 3.ed. Maya Press, 2004.

BACKMANN, Patricia; YOUNG, Phil. **3D Animation with Maya 6**. New York: Thompson Delmar Learning, 2005.

BLAIR, Preston. **Cartoon Animation**. Laguna Hills: Walter Foster Publishing, Inc., 1994.

CHOI, Jae-jin. **Maya Character Animation**. 2.ed. Alameda: Sybex, 2004.

FORD, Michael; LEHMAN, Alan. **Inspired 3D Character Setup**. Premier Press, 2002.

KERLOW, Isaac V. **The Art of 3D Computer Animation and Effects**. 3.ed. Hoboken: John Wiley & Sons, Inc., 2004.

LUBISCO, Nídia M. L.; VIEIRA, Sônia Chagas. **Manual de Estilo Acadêmico: Monografias, Dissertações e Teses**. 2.ed. Salvador: EDUFBA, 2003.

MACIEL, Luiz Carlos. **O Poder do Clímax**. Rio de Janeiro: Editora Record, 2003.

NORLING, Ernest R. **Perspective Made Easy**. Mineola: Dover Publications, Inc., 1999.

OMERNICK, Matthew. **Creating the Art of the Game**. Indianapolis: New Riders, 2004.

OSIPA, Jason. **Stop Staring: Facial Modeling and Animation Done Right**. San Francisco: Sybex, 2003.

SMITH, Thomas G. **Industrial Light & Magic: The Art of Special Effects**. New York: Ballantine Books, 1986.

VINEYARD, Jeremy. **Setting Up Your Shots**. Studio City: Michael Wiese Productions, 2000.

VOGLER, Christopher. **A Jornada do Escritor: estruturas míticas para contadores de histórias e roteirirstas** (Traduzido por Ana Maria Machado). Rio de Janeiro: Ampersand Ed., 1997.

WATTS, Harris. **On Camera: O curso de produção de filme e vídeo da BBC** (Traducido por Jairo Tadeu Longhi). 4.ed. São Paulo: Summus, 1990.

WILLIAMS, Richard. **The Animator's Survival Kit**. London and New York: Faber and Faber, 2001.

GLOSSÁRIO

Arquivo Executável - Em informática, é um arquivo em que seu conteúdo deve ser interpretado como um programa por um computador. Normalmente, eles possuem a representação binária das instruções de máquina de um processador específico, mas podem conter também uma forma intermediária que podem ser necessários serviços de um interpretador para executar. Hoje em dia, a distinção entre um programa na sua forma original (em linguagem humana) e em sua forma executável (em linguagem de máquina) está se tornando menos distinta, já que o ato de transformar a forma original no formato máquina (por compilação) ou a interpretação pode ser feito de modo implícito. Desse modo, o significado do termo executável está geralmente sendo estendido de um arquivo que contém instruções de máquina para qualquer arquivo que possa ser executado pelo ambiente sem a necessidade de uma transformação explícita. Arquivos contendo linguagem interpretada, por outro lado, são normalmente chamados de arquivos de script ou scripts em vez de executáveis.

Bitmap - Imagens raster (ou bitmap, que significa mapa de bits em inglês) são imagens que contém a descrição de cada pixel, em oposição aos gráficos vectoriais. O tratamento de imagens deste tipo requer ferramentas especializadas, geralmente utilizadas em fotografia, pois envolvem cálculos muito complexos, como interpolação, álgebra matricial, etc.Um bitmap pode ser preto-e-branco ou colorido. Há um padrão chamado RGB, do inglês Red, Green, Blue, que utiliza três números inteiros para representar cada uma das cores primárias, vermelho, verde e azul. A cada ponto da imagem exibida na tela ou papel corresponde um pixel deste grade, de forma que a maioria das imagens requer um número muito grande de pixels para ser representada completamente. Por exemplo, uma imagem de 800 pixels de largura por 600 de altura e 3 bytes para representar cada pixel - padrão RBG completo - tem o tamanho de 1.440.054 bytes. Embora a representação em memória RAM seja geralmente em bitmaps, quando se fala em um grande número de imagens armazenada em discos magnéticos e trasmissão de dados via redes surge a necessidade de compressão desses arquivos, para reduzir o espaço ocupado e o tempo de transmissão.

Bump mapping - É uma técnica de computação gráfica em que, a cada pixel, uma

perturbação à superfície normal do objeto sendo renderizado é visto num heightmap, e aplicado antes que o cálculo de iluminação seja feito (veja, por exemplo, Phong shading). O resultado é mais rico, a apresentação da superfície é mais detalhada e mais próximas dos detalhes inerentes ao mundo natural. Normal mapping é a técnica de bump mapping mais usada, mas existem outras alternativas, como o Parallax mapping.

Código fonte - Código-fonte, ou até source code em inglês, é o conjunto de palavras escritas de forma ordenada, contendo instruções em uma das linguagens de programação existentes no mercado, de maneira lógica. Após compilado, transforma-se em software, ou seja, programas executáveis. Este conjunto de palavras, que formam linhas de comandos, deverão estar dentro da padronização da linguagem escolhida, obedecendo critérios de execução. Atualmente com a diversificação de linguagens, o código pode ser escrito de forma totalmente modular, podendo um mesmo conjunto de códigos ser compartilhado por diversos programas, e até mesmo linguagens.

Dolby Surround - Primeira versão do Dolby Stereo, formato de multicanais para som analógico de fílmes. Quando uma trilha sonora Dolby Surround é produzida, quatro canais de informação de áudio - esquerda, centro, direita e mono - são codificadas em duas trilhas de áudio. A informação estéreo é então levada a mídias estereofônicas, como fitas de video, discos compactos e canais de televisão, a partir das quais a informação pode ser decodificada por um processador para recriar o som espacial original, em quatro canais.

Geometria euclidiana - É a geometria sobre planos ou em três dimensões baseados nos postulados de Euclides de Alexandria. O texto de Os Elementos foi a primeira discussão sistemática sobre a geometria e o primeiro texto a falar sobre teoria dos números. Foi também um dos livros mais influentes na história, tanto pelo seu método quanto pelo seu conteúdo matemático. O método consiste em assumir um pequeno conjunto de axiomas intuitivos, e então provar várias outras proposições (teoremas) a partir desses axiomas. Muitos dos resultados de Euclides já haviam sido afirmados por matemáticos gregos anteriores, porém ele foi o primeiro a demonstrar como essas proposições poderiam ser reunidas juntas em um compreensivo sistema dedutivo.

Linguagem de Programação - É um método padronizado para expressar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador. Uma linguagem permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias. O conjunto de palavras (tokens), compostos de acordo com essas regras, constituem o código fonte de um software. Esse código fonte é depois traduzido para código de máquina, que é executado pelo processador.

Normal mapping - Em computação gráfica 3D, *normal mapping* é a aplicação de uma técnica conhecida como mapeamento de relevo (*bump mapping). Normal mapping* também é chamada às vezes de "Dot3 bum mapping", para distinguir do bump mapping comum. Enquanto o bump mapping perturba a normal (direção perpendicular à superfície) existente de um polígono, o normal mapping substitui completamente a normal do polígono. Assim como o bump mapping, é utilizado para acrescentar detalhes ao sombreamento sem utilizar para isso mais polígonos. Mas enquanto um bum map é geralmente calculado com base numa imagem com um único canal de cor (interpretado como uma escala de tons de cinza), a fonte das normais no normal mapping é geralmente uma imagem multi-canais (isto é, canais para "vermelho", "verde" e "azul" ao invés de uma única cor), criada a partir de um conjunto de versões mais detalhadas dos objetos.

Pixel - Aglutinação de Picture e Element, ou seja, elemento da imagem, é o menor elemento num dispositivo de exibição (como por exemplo um monitor), ao qual é possivel atribuir-se uma cor. De uma forma mais simples, um pixel é o menor ponto que forma uma imagem digital, sendo que o conjunto de milhares de pixels formam a imagem inteira. Num monitor colorido cada Pixel é composto por um conjunto de 3 pontos: verde, vermelho e azul. Cada um destes pontos é capaz de exibir 256 tonalidades diferentes (o equivalente a 8 bits) e combinando tonalidades dos três pontos é possível exibir 16 milhões de cores diferentes. Em resolução de 640 x 480 temos 307 mil pixels, a 800 x 600 temos 480 mil, a 1024 x 768 temos 786 mil e assim por diante.

RGB - É a abreviatura do sistema de cores aditivas formado por Vermelho (Red), Verde (Green) e Azul (Blue). É o sistema aditivo de cores, ou seja, de projeções de luz, como monitores e datashows, em contraposição ao sistema subtrativo, que é o das impressões (CMYK). A escala de RGB varia de 0 (mais escuro) a 255 (mais claro). Nos programas de edição de imagem, esses valores são habitualmente representados por meio de notação hexadecimal, indo de 00 (mais escuro) até FF (mais claro) para o valor de cada uma das cores. Assim, a cor #000000 é o preto, pois não há projeção de nenhuma das três cores; em contrapartida, #FFFFFF representa a cor branca, pois as três cores estarão projetadas em sua intensidade máxima. As cores são complementares às do sistema CMYK - Ciano (Cyan), Magenta (Magenta), Amarelo (Yellow) e Preto (blacK) - e a sua mistura forma a cor branca.

Trigonometria - Do grego trigonos = triangulo e metrûn = medida, etimologicamente, significa medida de triangulos. É o ramo da matemática que estuda as relações em triângulo, ângulos e funções trigonométricas como o seno e cosseno.

Vetor - Em computação gráfica, imagem vetorial é um tipo de imagem gerada a partir de descrições geométricas de formas, diferente das imagens chamadas mapa de bits, que são geradas a partir de pontos minúsculos diferenciados por suas cores. Uma imagem vetorial normalmente é composta por curvas, elipses, polígonos, texto, entre outros elementos, isto é, utilizam vetores matemáticos para sua descrição. Em um trecho de desenho sólido, de uma cor apenas, um programa vetorial apenas repete o padrão, não tendo que armazenar dados para cada pixel. As Curvas de Bézier são usadas para a manipulação dos pontos de um desenho. Cada linha descrita em um desenho vetorial possui nós, e cada nó possui alças para manipular o segmento de reta ligado a ele. Por serem baseados em vetores, esses gráficos geralmente são mais leves (ocupam menos memória no disco) e não perdem qualidade ao serem ampliados, já que as funções matemáticas adequam-se facilmente à escala, o que não ocorre com gráficos raster que utilizazam métodos de interpolação na tentativa de preservar a qualidade. Outra vantagem do desenho vectorial é a possibilidade de isolar objectos e zonas, tratando-as independentemente.