

Representação de Ambiente Arquitetônico com Tecnologia de Jogos Digitais: uma análise da modelagem geométrica

Architectural Environment Representation with Digital Games Technology: an Analysis of Geometric Modeling Techniques

Wagner Costa Oliveira

Pós-Graduando em Gráfica Digital - Universidade Federal de Pelotas - Brasil
wenver@gmail.com

Felipe Etchegaray Heidrich

Professor Assistente FAUrb - Universidade Federal de Pelotas - Brasil
prof.felipeh@gmail.com

ABSTRACT

This paper aims to describe part of a study about the visualization of architectural designs through digital interactive environments generated with gaming technology. Thus, this text reports the analyses concerning the geometrical modeling procedures of an architectural environment specifically for this usage. The experiment consisted in using a modeling software to create one object using several different techniques, to analyze the number of polygons, the behavior of the surface normals, the stability of the geometries and collision behavior for each one of the objects, in the digital environment.

KEYWORDS: Modelagem Geométrica, Ambientes Digitais Interativos, Motor de Jogos Digitais.

Introdução

A representação gráfica arquitetônica tradicional é composta por uma série de desenhos técnicos projetivos. Embora estes desenhos consigam realizar a comunicação das intenções do arquiteto aos profissionais envolvidos na execução da obra, por vezes eles se mostram ineficientes para a comunicação entre o arquiteto (especialista) e o cliente (não especialista). Assim, outros artifícios acabam sendo buscados para tornar mais clara esta comunicação, como maquetes físicas executadas em diversas escalas e materiais, perspectivas e maquetes eletrônicas. Neste contexto, os motores de jogos mostram-se como uma possibilidade para o desenvolvimento de ambientes interativos digitais, capazes de gerar simulações em tempo real com texturas realistas, iluminação direta e indireta, sombras, áudio e suporte a múltiplos usuários (Hoon e Kehoe, 2003, p. 348).

O presente artigo, portanto, busca descrever parte de um estudo sobre visualização de ambientes arquitetônicos através de ambientes digitais interativos gerados a partir de tecnologia de jogos digitais. Desta forma, o texto descreve as análises realizadas no que

diz respeito às questões de modelagem geométrica de um ambiente arquitetônico especificamente para este uso. No experimento, um mesmo objeto foi criado em um programa de modelagem, utilizando diferentes procedimentos, com o objetivo de analisar as seguintes questões: número de polígonos que definem cada um dos modelos, comportamento das normais dos polígonos geradores das superfícies, estabilidade das geometrias e comportamento das colisões do avatar com o modelo tridimensional do ambiente.

Motor de Jogos Digitais

Segundo Lewis e Jacobson (2002), o termo motor de jogos refere-se a uma coleção de módulos de códigos de simulação capazes de gerenciar as diversas partes componentes de um jogo eletrônico digital. É formado de partes responsáveis por lidar com a entrada de dados (teclado, mouse, joystick), saída de dados (renderizações tridimensionais, desenhos bidimensionais, sons) e simulações de efeitos físicos e dinâmicos para os mundos virtuais. São, portanto, ferramentas para o desenvolvimento de jogos eletrônicos para computadores ou consoles de videogame.

Vale salientar, contudo, que o motor de jogos não é o jogo em si. Deste modo, “os motores oferecem componentes reutilizáveis que podem ser manipulados para trazer um jogo à vida. Carregar, mostrar e animar modelos, detecção de colisão entre objetos, física, entrada, interfaces gráficas de usuário, e até mesmo porções da inteligência artificial de um jogo podem ser componentes integrantes do motor.” (Ward, 2008)

Entre os recursos propiciados por estes programas, encontram-se a interatividade total, a solidez tátil do mundo virtual, o áudio integrado, a possibilidade de múltiplos usuários simultâneos, a inclusão de avatares, a simulação de movimento e do sítio, e a interação com objetos (Hoon e Kehoe, 2003, p. 351).

Desta forma o presente estudo é parte de um estudo maior o qual visa verificar a aplicação de motores de jogos digitais no desenvolvimento de ambientes digitais interativos com finalidade de visualização de ambientes arquitetônicos. Para tal estudo está sendo utilizado neste momento o motor de jogos *Unreal Development Kit*, ou UDK, desenvolvido pela empresa *Epic Games*.

Modelagem Geométrica

A primeira etapa para a construção do ambiente digital interativo é a modelagem geométrica dos elementos que o compõem. Este processo pode ser realizado de diversas maneiras, de acordo com o programa utilizado e os comandos escolhidos para se realizar os procedimentos. Cada técnica de modelagem gera uma malha que, embora tenha a mesma aparência de outra malha criada através de uma técnica diferente, pode apresentar outras propriedades diferentes.

Neste sentido, o estudo descreve os experimentos realizados para avaliar como o motor de jogos interpreta os objetos construídos no aplicativo de modelagem escolhido, o *software* 3DS Max.

Com o objetivo de avaliar as diferenças entre geometrias criadas através de diferentes processos dentro de um mesmo programa, foi realizado o experimento que consistiu na modelagem de uma parede de 3,50m de comprimento, 25cm de espessura e 2,70m de altura, com um vão centralizado de 2,00m de largura, 1,20m de altura e peitoril de 1,00m. Através da utilização de onze técnicas diferentes para criar este modelo geométrico, esperava-se identificar as mais adequadas para a criação de geometrias para a utilização no motor de jogos, bem como as que não podem ser utilizadas, caso houvesse. Vários dos objetos foram criados através de operações de edição de malha, porém utilizando operações diferentes para atingir o resultado desejado.

O processo de elaboração de cada um dos modelos estudados neste experimento foi:

- Objeto 01 – primeiramente construiu-se uma primitiva

de programa do tipo *Box*, com as dimensões de toda a parede. A seguir, criou-se, com o mesmo comando, um paralelepípedo com as dimensões do vão. Após posicionar corretamente o vão com os comandos de translação de objetos, realizou-se uma operação booleana de subtração com o comando *Boolean*, abrindo assim o vão na parede.

- Objeto 02 – realizou-se a conversão de uma primitiva do tipo *Box* em uma malha editável, do tipo *Editable Poly*. A seguir, foi realizada a adição de arestas para a abertura do vão com os comandos *Connect Edges* e *Chamfer Edges*. Após apagar as faces desnecessárias, as arestas foram conectadas com o comando *Bridge Borders*, fechando a geometria.

- Objeto 03 – a partir do objeto 02, foi realizada a translação de alguns dos vértices da malha, sobrepondo-os aos vértices dos cantos externos da parede. Realizou-se então a união dos vértices sobrepostos com o comando *Weld Vertices*. Este tipo de modelagem foi utilizado por ter sido recomendado em um tutorial visto na internet.

- Objeto 04 – o *Box* inicial foi criado com as dimensões da parede lateral à janela (0,75m X 0,25m X 2,70m) e convertido em *Editable Poly*. Após a divisão de uma das faces com as operações *Connect* e *Chamfer Edges*, o restante do modelo foi criado através da extrusão de faces, com o comando *Extrude*. As últimas faces foram criadas com a operação *Bridge Faces*.

- Objeto 05 – neste caso, o modelo de origem foi uma primitiva de programa do tipo *Plane*, transformado em malha editável do tipo *Editable Poly* para a criação do vão. A espessura da parede, então, foi dada com o modificador *Shell*.

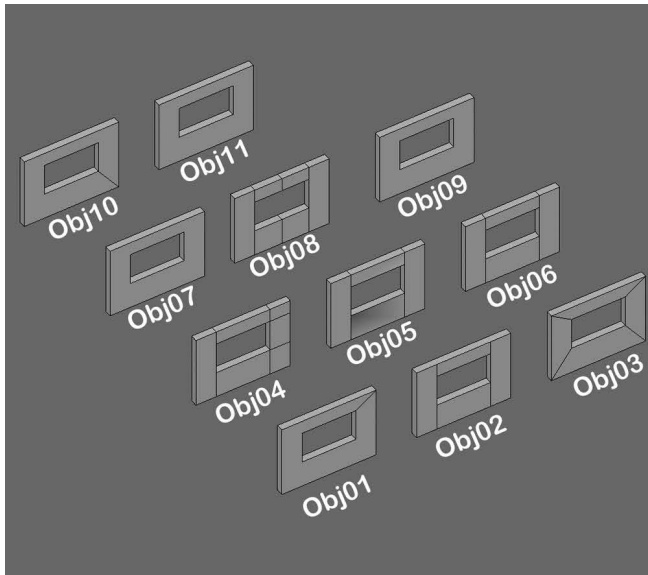
- Objeto 06 – a parede foi construída com quatro primitivas do tipo *Box* independentes.

- Objeto 07 – criado a partir de uma operação booleana, assim como o Objeto 01. Porém, neste experimento utilizou-se o operador *ProBoolean*, que gera uma geometria final com menos arestas visíveis.

- Objeto 08 – executado com um procedimento semelhante ao Objeto 04 (geometria modelada através de extrusão de faces). Neste caso, contudo, modelou-se apenas metade do objeto. A outra metade é proveniente de um modificador *Symmetry* aplicado no eixo Y do modelo, criando assim uma simetria axial.

- Objeto 09 – o modelo foi executado com o comando especial *AEC* (*Architecture, Engineering and Construction*) *Wall*. O vão para a janela foi aberto com um objeto *AEC Window*. A parede então foi convertida em uma malha editável do tipo *poly*, e a janela foi apagada.

- Objeto 10 – primeiramente foram criadas duas *polylines* do tipo *Rectangle*, para representar os contornos



Identificação	Tamanho do Arquivo .MAX (KB)	Número de Vértices	Número de Triângulos	Número de Polígonos
Objeto 01	200	16	32	32
Objeto 02	204	24	48	16
Objeto 03	204	16	32	16
Objeto 04	204	28	56	26
Objeto 05	204	24	48	48
Objeto 06	204	32	48	48
Objeto 07	204	16	32	32
Objeto 08	204	40	80	40
Objeto 09	200	18	36	10
Objeto 10	204	16	32	32
Objeto 11	200	16	32	32

Fig. 1. Esquerda: objetos modelados no experimento visualizados no software 3DS Max. Direita: tabela com as características dos objetos.

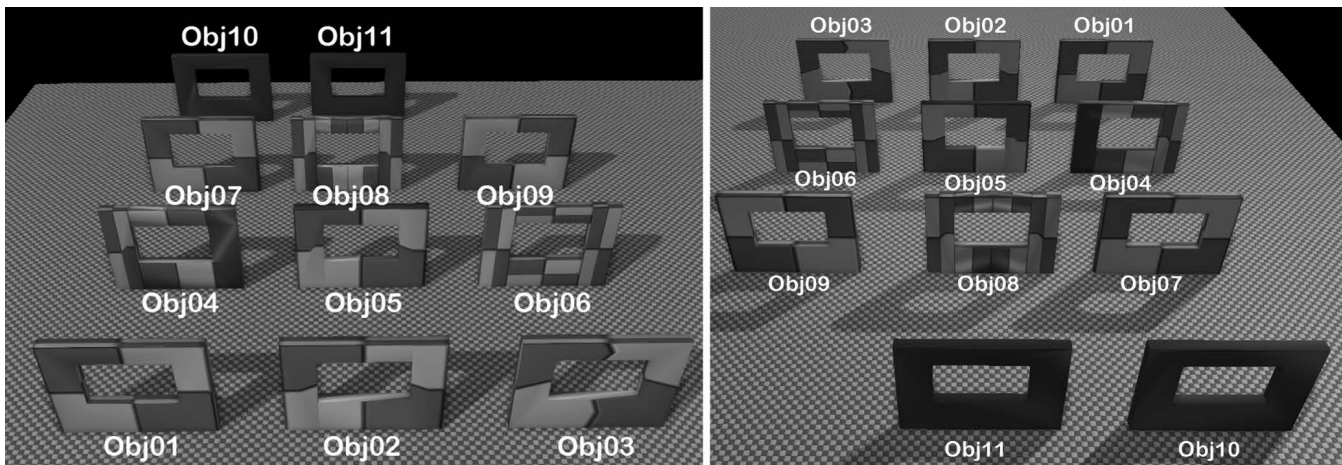


Fig. 2. Objetos modelados no experimento visualizados no motor de jogos.

externos da parede e do vão. Estes polígonos então sofreram extrusão ao longo de um eixo reto normal ao plano da figura, para assim representar a espessura. Finalmente, uma operação booleana de subtração foi utilizada para abrir o vão na parede.

- Objeto 11 – processo semelhante ao anterior, porém neste caso as duas *polylines* foram combinadas em uma única entidade do tipo *spline* antes do processo de extrusão.

Após a modelagem as características dos objetos analisadas foram: número de vértices, polígonos e triângulos em cada um dos modelos geométricos estudados, o tamanho de cada arquivo criado, a estabilidade e a aparência da geometria quando importada no motor de jogos.

A partir dos resultados obtidos, fizeram-se algumas constatações em relação às geometrias e aos arquivos produzidos. Em primeiro lugar, embora todos os objetos tenham a mesma aparência após a renderização, há diferenças significativas nas malhas geométricas,

reflexo do processo de sua construção. Com relação aos arquivos produzidos, averiguou-se que o número de vértices, triângulos ou faces não teve relação direta com o tamanho final do arquivo, visto que o arquivo cuja geometria apresenta a maior complexidade (com quarenta triângulos e oitenta vértices) é um dos menores em tamanho. Da mesma forma, o número de entidades individuais que compõem o modelo não influenciou diretamente no tamanho do arquivo produzido, visto que o modelo criado com quatro entidades diferentes apresentou o mesmo tamanho que outros produzidos com uma única entidade.

O próximo passo foi converter as geometrias para o formato *ASCII Scene Exporter* (ASE), compatível com o motor de jogos. Este tipo de arquivo pode ser aberto em um editor de texto, e contém informações organizadas em blocos definindo as propriedades das entidades geométricas e dos materiais aplicados na cena. Muitos destes blocos são ignorados pelo motor de jogos, como as definições relacionadas às luzes e câmeras existentes na cena.

Percebe-se que, com exceção dos Objetos 1 e 9, todos os outros modelos apresentam algum tipo de anomalia na superfície. Em alguns casos, são faces ou triângulos mais claros ou mais escuros em pontos onde não deveriam; em outros, surgiram arestas aparentes em faces.

A primeira hipótese considerada foi a de que estas irregularidades seriam alterações nas normais das superfícies ou no mapeamento das texturas, decorrentes do próprio processo de modelagem dos objetos. Ao modificar vértices, arestas e faces, estas características dos objetos também são modificadas.

Para verificar esta hipótese, foi realizado um segundo experimento, que consistiu em aplicar, a cada um dos objetos no 3ds Max, um mapeamento de textura do tipo *Box* e um modificador *Smooth*, cuja função é reorganizar as normais das superfícies de maneira a representar transições suaves ou rígidas entre as faces dos modelos geométricos.

Após os procedimentos descritos, os objetos foram novamente exportados no formato ASE e importados no ambiente do motor de jogos.

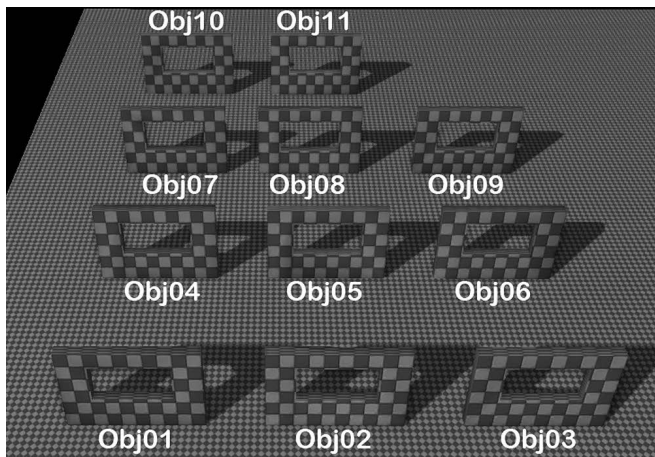


Fig. 3. Objetos do segundo experimento visualizados no motor de jogos.

Em relação à aparência geral dos objetos modelados, com segundo experimento a modelagem conseguiu ordenar o mapa aplicado no motor de jogos, eliminando as linhas inclinadas e as diferenças de repetição do padrão que ocorriam em alguns objetos.

Considerações finais

Analisa-se que os experimentos de modelagem geométrica realizados pelo estudo, foram importantes na verificação da aplicação das diferentes técnicas oferecidas pelo programa de modelagem utilizado para a criação de objetos a serem utilizados no processo de geração de ambientes digitais interativos com motor de jogos digitais. Neste sentido, concluiu-se que os objetos a serem utilizados no motor de

jogos podem ser modelados utilizando-se qualquer uma das técnicas apresentadas no que diz respeito à modelagem geométrica. Entretanto outras questões foram percebidas e serão melhor analisadas em outros experimentos relacionados às etapas de modelagem visual dos objetos.

Agradecimentos

Este trabalho se insere no projeto PROBARQ (Produção e compartilhamento de objetos de aprendizagem para o projeto de arquitetura), financiado pelo CNPq, e apoiado pela FAPERGS e CAPES, o qual busca estruturar materiais didáticos para os estágios iniciais de formação, a serem disponibilizados de forma irrestrita às comunidades acadêmicas e profissionais da área de arquitetura, por meio da INTERNET.

Referências

Hoon, M.; Kehoe, M. 2003. Enhancing architectural communication with game engines. *Crossroads of Digital Disclosure: Proceedings of Annual Conference of the Association for Computer Aided Design in Architecture* (pp. 349-355). Indianapolis.

Lewis, M.; Jacobson, J. 2002. Game engines in scientific research. *Communications of the ACM*, 45 (1), 27-31.

Ward, J. 2008. *What is a game engine?* Acesso em março de 2012, de http://www.gamecareerguide.com/features/529/what_is_a_game_.php?page=1