

## Mundo IME Virtual

Agora que você já sabe modelar objetos, transformações geométricas, projeções, e aprendeu a trabalhar com texturas, está na hora de elevar o nível de realismo do seu trabalho. Neste exercício-programa, dividido em 2 fases, você deverá: (i) modelar um mundo virtual, que consistirá dos prédios e detalhes estruturais do próprio IME, circundado pelos outros institutos da USP e por um céu artificial, e (ii) desenvolver um sistema de partículas que simule chuva com vários graus de intensidade e que interaja fisicamente com o seu modelo. Para tanto você deverá utilizar OpenGL e a biblioteca auxiliar GLUT. É altamente recomendado que você utilize a linguagem C++ na sua implementação. No final do curso, você fará uma demonstração da sua simulação para a turma.

### Primeira Fase (60%) – Modelagem e dinâmica do mundo IME

Em games de First Person Shooter (FPS), um ou mais personagens estão imersos em um ambiente interno contendo diversos objetos rodeados por uma atmosfera exterior. Nesta primeira fase, você deverá implementar esse paradigma para criar o mundo IME virtual, de acordo com os requisitos abaixo.

#### 1. Ambiente

O universo da sua simulação consistirá dos prédios do IME-USP e do terreno ao seu redor. Você modelará os prédios do IME como objetos geométricos que devem ser mapeados com texturas obtidas por fotos reais dos prédios, que você deverá adquirir com uma câmera digital. A visão do seu personagem será representada pela posição da câmera que deverá ser movimentada pelo usuário utilizando o teclado (teclas seta). Ou seja, o seu personagem deve ser capaz de mover-se dentro deste universo - para frente, trás, esquerda e direita. Você não deve permitir que o seu personagem atravesse as paredes do modelo e nem chegue perto do horizonte. Árvores e carros podem ser omitidos. Você não precisa desenhar o seu personagem.

#### 2. Céu e horizonte

Assim como os antigos acreditavam que o céu era um domo localizado a uma distância intangível, onde nuvens e estrelas eram pinturas em movimento, você se utilizará dessa ilusão para simular um céu infinito ao redor do seu mundo. Para tanto utilizaremos a técnica *skybox*, normalmente utilizada em games (<http://www.opengl.org/wiki/Skybox>). Esta técnica descreve o mapeamento de ambientes externos em um cubo com dimensões razoáveis, cujas faces são texturizadas com fotos panorâmicas de ambientes externos. A câmera da cena é posicionada dentro do cubo de forma a criar um ambiente externo ilusório. Você deverá criar esse *skybox* (Fig. 1) utilizando fotos do ambiente ao redor dos prédios do IME. Para criar este mapeamento de forma eficiente você poderá usar ferramentas como o **gimp** e **blender**. Ou seja, o seu mundo interior deverá consistir somente dos prédios do IME, porém o *skybox* criará a ilusão de que todo o mundo externo (i.e. outros institutos, céu, etc) existem ao redor. O seu mundo também deverá apresentar um certo grau de dinamicidade. Você utilizará inicialmente o relógio do sistema para regular a posição do sol que deverá nascer e se por. Você deverá prover meios para acelerar ou desacelerar o tempo para o usuário observe a troca entre dia e noite, de forma



Figura 1: Exemplo de textura utilizada em um *skybox*.

suave. Para alternar entre ambientes diurno e noturno, você poderá fazer uso de múltiplas texturas ou simplesmente mudar os parâmetros de iluminação (ou ainda uma combinação de técnicas).

## 2. Sol e sombras

Tendo em vista que temos a posição do sol, também é requisitado que você modele as sombras dos objetos do seu modelo em relação ao solo. Você poderá usar técnicas simples como projeções de perspectiva ou técnicas de mapeamento de sombras utilizando o buffer de profundidade ([http://en.wikipedia.org/wiki/Shadow\\_mapping](http://en.wikipedia.org/wiki/Shadow_mapping)).

## Segunda Fase (40%) – Chuva, outra vez ?

Na segunda fase deste EP, você deverá implementar um sistema de partículas que simule chuva. Você deverá começar mudando o seu ambiente externo, alterando luzes, e adicionando um pouco de neblina, caso ache necessário. Em seguida, simule a queda de um número de gotas de água a partir do *skycube*, fazendo com que elas atinjam o solo ou os prédios. O seu objetivo é maximizar o *frame rate*. Talvez você somente deseje que chova dentro do campo visual da câmera. Para desenhar as gotas, você pode utilizar pontos, esferas, ou até mesmo texturas. É importante que você deixe o usuário aumentar ou diminuir a intensidade da chuva utilizando alguma tecla. Seja criativo.

## Bônus para realismo

Serão ofertados até 2 pontos extras pelo realismo e riqueza de detalhes demonstrado na sua simulação.

## Observações finais

### Sobre a implementação:

- Mostre o *frame rate* da sua simulação superposto ao seu mundo virtual.

- Economize recursos no sistema de partículas, reuse aquelas que já atingiram o solo ou prédios.
- Não é necessário que o seu personagem entre dentro dos prédios. No entanto você pode utilizar um mapeamento de ambiente interno para cada prédio para aumentar o realismo.
- Embora este EP possa ser realizado sem o uso de *shaders*, você poderá utilizá-los para melhor eficiência da sua simulação, mas use somente GLSL.
- Caso você utilize pedaços de código de outras fontes, referencie-as de forma apropriada no seu arquivo LEIAME (veja abaixo) e no seu código fonte.

### Links úteis:

- <http://www.gamerendering.com>
- <http://developer.amd.com>

### Sobre a elaboração:

É aconselhável que este EP seja elaborado por equipes de dois alunos, respeitando as seguintes regras.

- Os alunos devem trabalhar sempre juntos cooperativamente.
- Caso em um grupo exista um aluno com maior facilidade, este deve explicar as decisões tomadas. E o seu par deve participar e se esforçar para entender o desenvolvimento do programa (chamamos isso de *programação em pares*, que é uma excelente prática que vocês devem se esforçar para adotar).

### Sobre a avaliação:

- É sua responsabilidade manter o código do seu EP em sigilo, ou seja, apenas você e seu par devem ter acesso ao código.
- **Não serão toleradas cópias!** Exercícios copiados (com ou sem eventuais disfarces) levarão à reprovação da disciplina e o encaminhamento do caso para a Comissão de Graduação do aluno.
- Exercícios com erros de sintaxe (ou seja, erros de compilação) receberão nota zero.
- É muito importante que seu programa seja elegante, claro e bem indentado, ou seja, digitado de maneira a ressaltar a estrutura de subordinação dos comandos do programa (conforme visto em aula). A qualidade do seu trabalho sob esse ponto de vista influenciará sua nota!
- As informações impressas pelo seu programa na tela devem aparecer da forma mais clara possível. Este aspecto também será levado em consideração no cálculo da sua nota.
- Uma regra básica é a seguinte: do ponto de vista do monitor responsável pela correção dos trabalhos, quanto mais convenientemente apresentado estiver o seu programa, melhor avaliado será seu trabalho.

### Sobre a entrega:

- Entregar apenas um arquivo de nome **MundoIME.zip** contendo todas as classes de seu exercício. Entregue também um arquivo texto contendo uma explicação sobre o programa (e.g. teclas e funcionalidades), dê a ele o nome **LEIAME**.
- No início de cada arquivo, acrescente um cabeçalho bem informativo, como o seguinte:

```

/*****
/**  MAC 420 - Introdução à Computação          **/
/**  IME-USP - Primeiro Semestre de 2012        **/
/**  <turma> - <nome do professor>              **/
/**                                              **/
/**  Segundo Exercício-Programa                  **/
/**  Arquivo: NomeDoArquivo.cpp                  **/
/**                                              **/
/**  <nome do(a) aluno(a)>                       <número USP>      **/
/**  <nome do(a) aluno(a)>                       <número USP>      **/
/**                                              **/
/**  <data de entrega>                          **/
*****/

```

Não é obrigatório que o cabeçalho seja idêntico a esse, apenas que contenha pelo menos as mesmas informações.

- Para a entrega, utilize o Paca. Você pode entregar várias versões de um mesmo EP até o prazo, mas somente a última será armazenada pelo sistema.
- Não serão aceitas submissões por email ou atrasadas. Não deixe para a última hora, pois o sistema pode ficar congestionado e você poderá não conseguir enviar.
- Guarde uma cópia do seu EP pelo menos até o fim do semestre e, novamente, você é responsável por manter o sigilo de seu código-fonte.