

Computação Gráfica (MIEIC)

Projeto Prático

Versão Provisória 0.1

Projeto A

Objetivos

- Aplicar os conhecimentos e técnicas adquiridas até à data
- Criação de uma cena complexa, com diferentes geometrias, materiais, texturas e luzes

Descrição

Pretende-se com este projeto a criação de uma cena que combine os diferentes elementos explorados nas aulas anteriores. Para este trabalho deve usar como base o código que é fornecido no Moodle, que corresponde a uma cena vazia. Terá posteriormente de adicionar alguns dos objetos criados anteriormente.

A cena deve ser genericamente constituída por:

- Um terreno,
- Vários conjuntos de árvores,
- Uma casa
- Algumas elevações
- Uma paisagem envolvente
- Luzes a simular dia, noite de luar e uma lanterna

Os pontos seguintes descrevem as principais características dos diferentes elementos pretendidos. É dada alguma liberdade quanto à composição dos mesmos na cena, para que cada grupo possa criar a sua própria cena.

1. Criação de novas Primitivas

1.1. Prisma (sem topos)

Pretende-se que crie a classe **MyPrism** para que esta consiga desenhar um prisma com um número variável de "lados" (slices) como se estivesse contido (inscrito) num cilindro de raio igual a uma unidade, base coincidente com o plano XZ e centrada na origem, e com comprimento também unitário, em Y (ver Figura 1). O cilindro pode ser aberto nas extremidades (sem tampas).

Defina as normais para **MyPrism**, tendo em conta que, para cada face, as normais dos seus vértices devem ser perpendiculares a essa face (Figura 2). Poderá por isso ter de **definir o mesmo vértice** mais do que uma vez, na lista de vértices, de forma a atribuir-lhe normais diferentes.

Para cada vértice defina as **coordenadas de textura** por forma a mapear uma textura à volta do prisma.

Nota: Analise o código da classe *MyPyramid*, fornecido na TP3, e verifique as semelhanças.

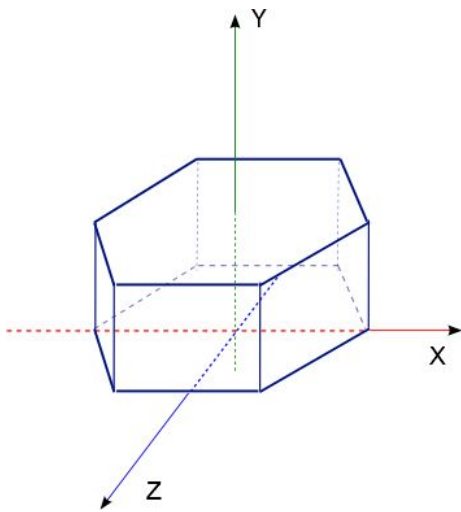


Figura 1: Prisma de seis lados (*slices*) e comprimento unitário.

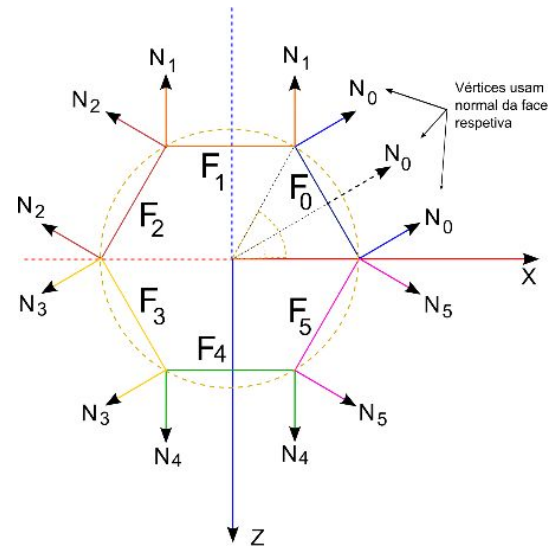


Figura 2: Ilustração das normais a atribuir a cada vértice no caso de seis lados

1.2. Criação de um cilindro (sem topos)

Recorde como o cone é aproximado pela geometria da pirâmide manipulando as suas normais. Crie uma nova classe **MyCylinder** que aproxime um cilindro com a geometria de prisma criada no ponto anterior. Ou seja, a normal de cada vértice do prisma deve ser perpendicular à superfície do cilindro perfeito em que o prisma original está inscrito. À semelhança dos vértices da base do cone, cada vértice que seja usado em duas faces adjacentes, terá sempre a mesma normal (Figura 3), pelo que não será necessário repetir vértices neste caso.

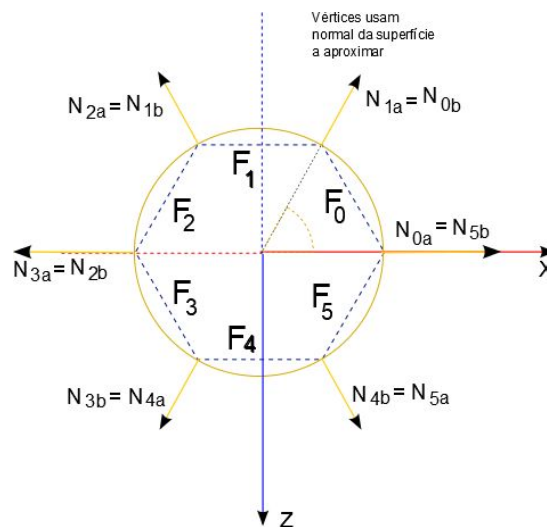


Figura 3: Ilustração das normais a atribuir a cada vértice no caso de um cilindro aproximado por seis lados.

Para cada vértice defina as **coordenadas de textura** por forma a mapear uma textura à volta do cilindro. Note que como a textura dá a volta ao cilindro poderá necessitar de repetir a primeira linha vertical de vértices pois estes são, simultaneamente, o início e o fim da textura.

2. Criação de objetos compostos

Nos seguintes exercícios, irá usar os objetos desenvolvidos até agora para criar objetos compostos para incluir na cena.

Adicione os ficheiros das classes das aulas práticas anteriores **MyPyramid**, **MyCone**, **MyQuad**, e **MyUnitCubeQuad** (composto por 6 **MyQuads**), e inclua no ficheiro **main.js**.

2.1 Árvore (MyTree)

Crie uma nova classe **MyTree**, para representar uma árvore. Este objeto será composto por tronco (cilindro) e copa (cone), centrados no eixo Y, e base no plano XZ, com materiais aplicados de forma adequada. O cilindro e o cone deverão ter entre 5 e 10 lados. A altura e raios do tronco e da copa devem ser fornecidos no construtor da **MyTree**.

No construtor devem também ser identificadas as texturas a aplicar ao tronco e copa.

MyTree.constructor (*scene*, *trunkHeight*, *trunkRadius*, *treeTopHeight*, *treeTopRadius*, *trunkTexture*, *treeTopTexture*)

2.2. Arvoredo (MyTreeGroupPatch, MyTreeRowPatch)

Crie duas classes que representem dois tipos de grupos de árvores, para depois poderem ser espalhados pela cena. A classe **MyTreeGroupPatch** define um conjunto de 9 árvores distribuídas numa grelha de 3x3, mas não perfeitamente alinhadas, e com alguma variedade de dimensões (fig. 4). A classe **MyTreeRowPatch** define um grupo de 6 árvores em linha, ligeiramente desalinhadas de forma arbitrária (fig. 5).

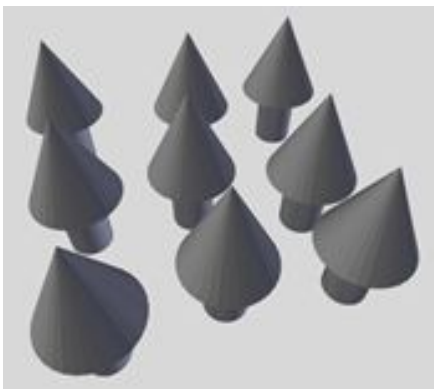


Figura 4: Grupo de 9 árvores distribuídas em 3x3 (**MyTreeGroupPatch**).

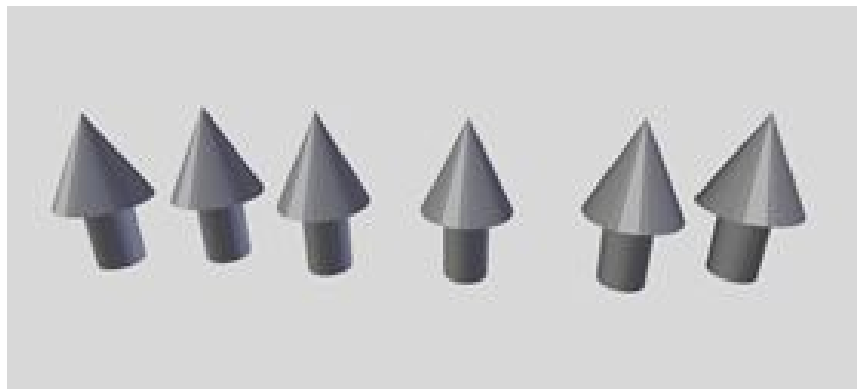


Figura 5: Grupo de 6 árvores distribuídas em linha (**MyTreeRowPatch**).

2.3. Casa (MyHouse)

Crie uma nova classe **MyHouse** para criar um objeto semelhante a uma casa, de forma à escolha. Este objeto deverá ser composto no mínimo por:

- 4 paredes representadas por um cubo (**MyUnitCubeQuad**),

- um telhado (**MyPyramid**)
- Pelo menos duas colunas exteriores (**MyPrism**). Estas duas colunas deverão ter entre 5 e 10 lados.

Adicione e aplique materiais e texturas a cada objeto constituinte de **MyHouse**, à escolha do grupo.

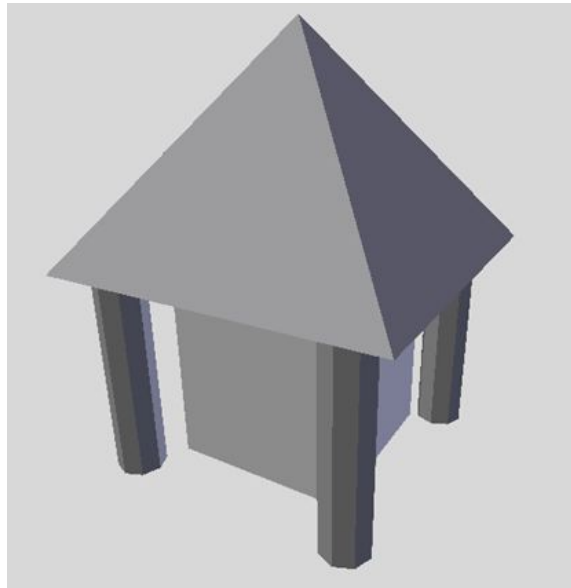


Figura 6: Exemplo de estrutura da casa. Podem fazer variantes.

2.4. Colina (MyVoxelHill)

Crie uma nova classe **MyVoxelHill** para criar um objeto composto por um número variável de **MyUnitCubeQuad** com as texturas aplicadas como pedido no TP4. O construtor deve ter como parâmetro a altura, expressa em número de níveis **levels**, seguindo a regra ilustrada na figura 7.

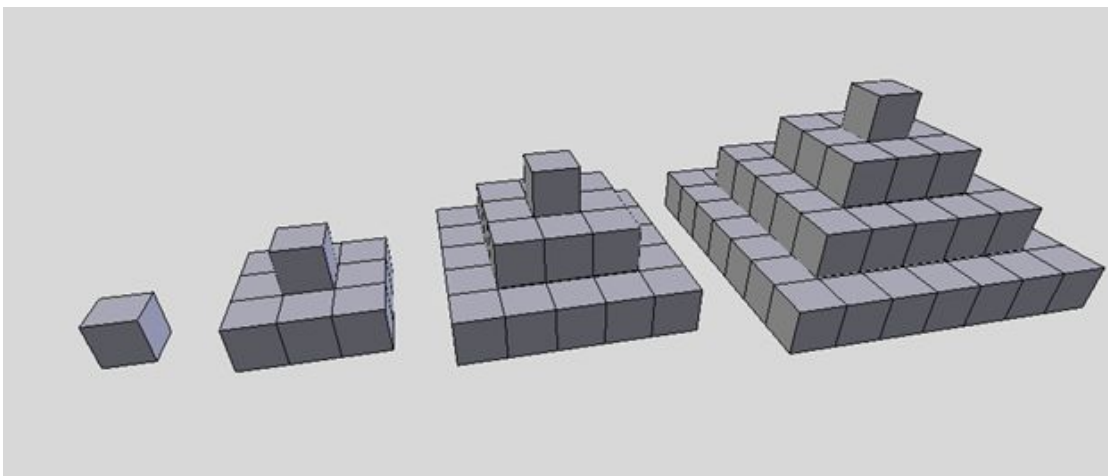


Figura 7: Da esquerda para a direita, **MyVoxelHill** com 1, 2, 3, e 4 níveis respetivamente.

3. Criação de um cubemap

Com este exercício pretende-se que crie um *cubemap* que sirva como ambiente de fundo da cena. Um *cubemap* pode ser definido como:

- um cubo de grandes dimensões (bastante superiores à da cena),

- apenas com as faces interiores visíveis,
- e às quais são aplicadas texturas que representam a envolvente da cena (uma paisagem, por exemplo; ver Figura 8).

Inclua na pasta do projeto uma cópia do código da classe **MyUnitCube**, o cubo unitário criado ao longo de TP2 e TP3, e use essa cópia para criar uma classe **MyCubeMap**, de forma a ser visível por dentro, e com coordenadas de textura que mapeiem texturas com o formato da Figura 8).

Se necessário, poderá ter de alterar a posição da câmara de forma a que fique no interior e centrada dentro do *cubemap*.

Como imagens para o cubemap podem utilizar recursos como os indicados abaixo:

<http://www.custommapmakers.org/skyboxes.php>

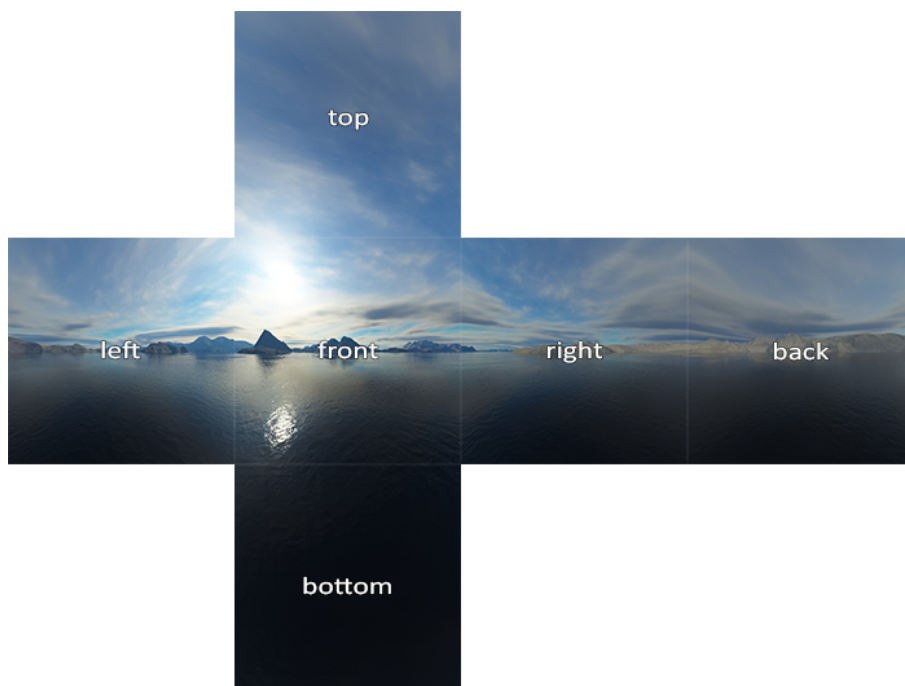


Figura 8: Imagem do tipo skybox, para textura de cubemap

4. Cena final

Ao longo dos próximos exercícios, será criada uma cena composta pelos diferentes objetos criados até ao momento, para criar um ambiente semelhante a uma floresta.

4.1 Composição da cena

Utilize os objetos criados anteriormente, e coloque-os na cena, aplicando transformações geométricas para criar um cenário de uma floresta. O cenário deverá ser composto por, pelo menos:

- Cubemap para criar o ambiente de fundo, que terá de ter duas versões (uma para dia e outra para noite, ver 4.3)
- 1 plano de terreno (**MyQuad**), assente no plano XZ, com textura de terreno/relva com repetição, sem esticar a textura com o escalamento
- 1 casa centrada na cena (**MyHouse**)
- 2 ou mais colinas (**MyVoxelHill**) distribuídas à escolha

- 2 ou mais arvoredos de cada tipo (**MyTree*Patch**) distribuídas à escolha (sem sobreposições)
- 1 objeto a representar uma fogueira/lanterna (poderá utilizar uma combinação de objetos à escolha)
- Instâncias adicionais de objetos (se necessário) para aplicação dos materiais referidos em 4.2

Nota: a cena e a câmara inicial devem ser definidas de forma a ser possível visualizar a cena na totalidade sem zoom in/out de câmara.

4.2 Materiais

Deve aplicar materiais nos elementos a incluir de forma a explorar a iluminação e a dar aspetos diferenciados às superfícies. Deve em particular garantir que a cena contém pelo menos:

- 1 material especular (ex: metal/água/vidro/mármore)
- 2 materiais difusos (ex: pedra/madeira/terra)

Para permitir verificar as características dos materiais, **inclua na GUI uma *checkbox* para ativar/desativar as texturas** (verifique a documentação da função ***enableTextures*** da ***CGFscene***)

4.3 Modo dia e noite



Serão definidas várias luzes na cena utilizadas para criar uma iluminação de um ambiente diurno ou noturno. Deverá criar na interface um menu *dropdown* com as opções “Day” e “Night”, que deverão ativar as luzes associadas ao ambiente selecionado (e desativar as restantes).

As luzes a criar deverão ser, no mínimo:

- Luz 1 (dia) - Sol, posição com elevado Y, cor quente, e mínima atenuação
- Luz 2 (noite) - Lua, posição com elevado Y, cor fria, e pouca atenuação
- Luz 3 (noite) - Fogueira ou lanterna de campismo, posição com baixo Y, cor quente, e maior atenuação



A textura aplicada no cubemap deverá ser diferente nos dois ambientes, com imagens adequadas ao ambiente selecionado.

Capture duas imagens onde se consiga visualizar (no modo dia e noite respetivamente) a totalidade do terreno e seus elementos. Capture três imagens adicionais de detalhes que considere relevantes. Submeta as imagens e a versão final do código.

( 1,2,3,4,5) ( 1)

Checklist

Até ao final do trabalho deverá submeter as seguintes imagens e versões do código via Moodle, **respeitando estritamente a regra dos nomes**:

-  Imagens (5): (nomes do tipo "projA-t<turma>g<grupo>-n.png")
-  Código em arquivo zip (1): (nomes do tipo "projA-t<turma>g<grupo>-n.zip")