Laboratório de Aplicações com Interface Gráfica

MIEIC – 2017/2018
Avaliação individual após Trabalho T1
(duração: 2:30 horas + 0:30 de tolerância)

O presente enunciado diz respeito a uma prova de avaliação prática da unidade curricular LAIG e contém um conjunto de três desenvolvimentos a efetuar sobre a base de trabalho desenvolvida no trabalho T1 das aulas práticas. Os computadores das salas reservadas para a prova encontram-se privados de rede. O único acesso permitido é ao software "SIGEX" para efeitos de acesso ao enunciado e documentação e *upload* dos trabalhos realizados (http://sigex.fe.up.pt; a validação é a normal do FEUPSIG).
Introduzir o "Código Público do Exame" correspondente:

- **B104** (WINDOWS), Código Público do Exame: **PMF326**
- **B207** (WINDOWS), Código Público do Exame: **RNV872**
- **B207** (LINUX), Código Público do Exame: **TGH082**
- **B208** (WINDOWS), Código Público do Exame: **HHT107**
- **B213** (WINDOWS), Código Público do Exame: **KIJ552**

O resultado é uma página onde pode ser encontrada alguma informação assim como facilidades de *upload* de ficheiros.

Instruções para *Upload*

Siga cuidadosamente as seguintes instruções:

- Para cada problema, use sempre como base o **código original** do trabalho **T1** (não resolver um problema proposto sobre a solução do problema anterior).
- **Assinale no próprio código** as secções que alterou para resolver o problema, **com um ou mais blocos de comentários** do tipo (dependendo do problema que está a resolver):
  - //LAIGPROB1_inicio, //LAIGPROB1_fim;
  - //LAIGPROB2_inicio, //LAIGPROB2_fim;
  - //LAIGPROB3_inicio, //LAIGPROB3_fim.
- Para cada problema, crie um ficheiro de texto **ident.txt**, contendo:
  - Identificação do estudante (número e nome),
  - Identificação do Grupo,
  - Situação da resolução do problema em questão: "prob. completo", "não considera o detalhe xpto", "não resolvido"...
  - Algumas instruções julgadas pertinentes para o funcionamento do software, nomeadamente a forma de identificar o ficheiro "teste.xml", se for o caso.
- Arquive os seguintes ficheiros num ficheiro .zip com o nome "prob1.zip"... "prob3.zip" (esta regra de nomeação deve ser ESTRITAMENTE seguida, **sem espaços nem extensões diferentes de .zip**):
  - o ficheiro **ident.txt**;

    - o **código-fonte**, incluindo o próprio **index.html** de arranque;
    - um ou mais **ficheiros LSX/XML** necessários para demonstrar a resolução do problema em questão;
    - todas as **texturas** utilizadas.
- Cada problema, tendo em atenção o ponto anterior, deverá ser arquivado num diretório independente ("prob1",... "prob3") e cada um destes, comprimido em um ficheiro "prob1.zip", ... "prob3.zip".
- Cada ficheiro obtido "prob?.zip" deverá ser sujeito a *upload* utilizando as facilidades do SIGEX.

Notas:

- Para efeito de teste, garanta que, após arrancar o servidor, **o *browser* usa o endereço 127.0.0.1:8080** (se necessário altere).
- Cada ficheiro "prob*.zip" deverá, isoladamente, ser sujeito a upload utilizando as facilidades do SIGEX.
- Caso não tenha resolvido um problema, deve submeter na mesma o respetivo ficheiro .zip, nem que contenha apenas o ficheiro **ident.txt** com a indicação de não-resolução. Isto é, no final, deve ter sempre três ficheiros submetidos: **prob1.zip, prob2.zip**, **prob3.zip**.
- Os identificadores finais dos arquivos serão construídos pelo sistema SIGEX, acrescentando-lhes um prefixo igual ao identificador do estudante respetivo.
- O sistema SIGEX permite eliminar ficheiros submetidos erradamente, e/ou submeter ficheiros com o mesmo nome, que serão sobrepostos aos existentes.

Enunciado - Novas funcionalidades sobre o trabalho T1

1. Nova primitiva Rect2 (7 valores)

Pretende-se disponibilizar um novo tipo de geometria **"rect2"** que crie um retângulo com as seguintes especificações:

1. A sintaxe é semelhante à da primitiva rectangle:

```
<LEAF type="rect2" args="ff ff   ff ff" corneroffset="ff" />
```

2. O parâmetro "corneroffset" é um valor percentual (Ex: 20%) a aplicar aos dois valores X e Y do segundo canto do retângulo;
3. Assim, as coordenadas ***X,Y*** do segundo vértice são afetadas por aquele valor (Ex: (X*1.2; Y*1.2), para os 20% de *offset*);
4. À primitiva devem ser aplicadas texturas com *amplification factors*, tal como na primitiva *rectangle*, considerando as dimensões do retângulo **após** a aplicação do offset descrito.

    - Exemplo: `<LEAF type="rect2" args="-15 -30  10 20" corneroffset="20"/>`
        - Retângulo desenhado com (-15, -30); (10*1.2; 20*1.2)
        - Textura mapeada ao retângulo modificado

a) Adeque o código desenvolvido nas aulas de forma a dar cumprimento à especificação anterior.
b) Faça um ficheiro LSX de teste que contenha, além do sistema de eixos e das fontes de luz, três retângulos **rect2**, dispostos lado a lado, com diferentes *offsets* e com a mesma textura. O ponto de vista e as fontes de luz devem permitir uma boa visualização da cena, nomeadamente os detalhes do efeito da aplicação de texturas

com *amplification factors*.

2. Marcação de Nós (8 valores)

Na interface da aplicação passa a existir a possibilidade de ligar/desligar a opção "visualizar objetos marcados" através de uma *"checkbox"*(por semelhança com a que foi desenvolvida para ligar/desligar as fontes de luz). Os objetos/nós podem ser marcados na linguagem LSX com mais uma propriedade que é o seu material de marcação; quando a opção "visualizar objetos marcados" estiver ativa, esses objetos e seus descendentes são desenhados com esse material alternativo (se opção está inativa, o material normal deve ser o usado).

Assim, para cada objeto suscetível de ser marcado dever-se-á definir, na linguagem LSX, um material de marcação, além do material usual. Seja, para o efeito, uma nova propriedade **matmark**, opcional, associada a um nó:

```
<NODE id="ss">
    <MATERIAL id="ss" />
    <TEXTURE id="ss" />
    <MATMARK id="ss" />   <!-- instrução opcional -->
    etc...
```

- O valor (string) do atributo "id" define o material de marcação.
- O material de marcação não deve ser "null".
- Sempre que se ativa a opção "visualizar objetos marcados", todos os nós que possuam a propriedade **matmark** passam a ser visualizados com os respetivos materiais de marcação. Os restantes objetos mantêm o seu material normal.
- A propriedade **matmark** de um nó ***P*** propaga-se obrigatoriamente a todos os seus descendentes: qualquer nó ***Q*** descendente direto ou indireto de ***P***, **com ou sem** uma propriedade **matmark** definida, passa a usar o material de marcação do nó ***P***.

a) Adeque o código desenvolvido nas aulas de forma a dar cumprimento à especificação anterior.
b) Faça um ficheiro LSX de teste que contenha, além do sistema de eixos e das fontes de luz, um conjunto de objetos simples que mostre claramente as facilidades implementadas e cubra os diferentes casos de descendência.

3. *Patches* múltiplos (6 valores)

Seja o *objeto* representado na figura 2. Pretende-se que seja desenhado com base na primitiva **patch** da linguagem LSX, por replicação rotativa do patch apresentado na figura 1.

Especificação de um patch:

- Deve ter as dimensões dadas na figura 1.

- Deve ser definido com a ordem mínima necessária para cada dimensão, **U** e **V**.
- Deve ser dividido em 15 partes em cada dimensão, **U** e **V**.

Especificação do objeto:

- O número de patches a usar é 12.
- Os vários patches são dispostos, em ângulos de 30º, à volta do eixo central.
- As dimensões do objeto podem ser controladas por meio de transformações geométricas.

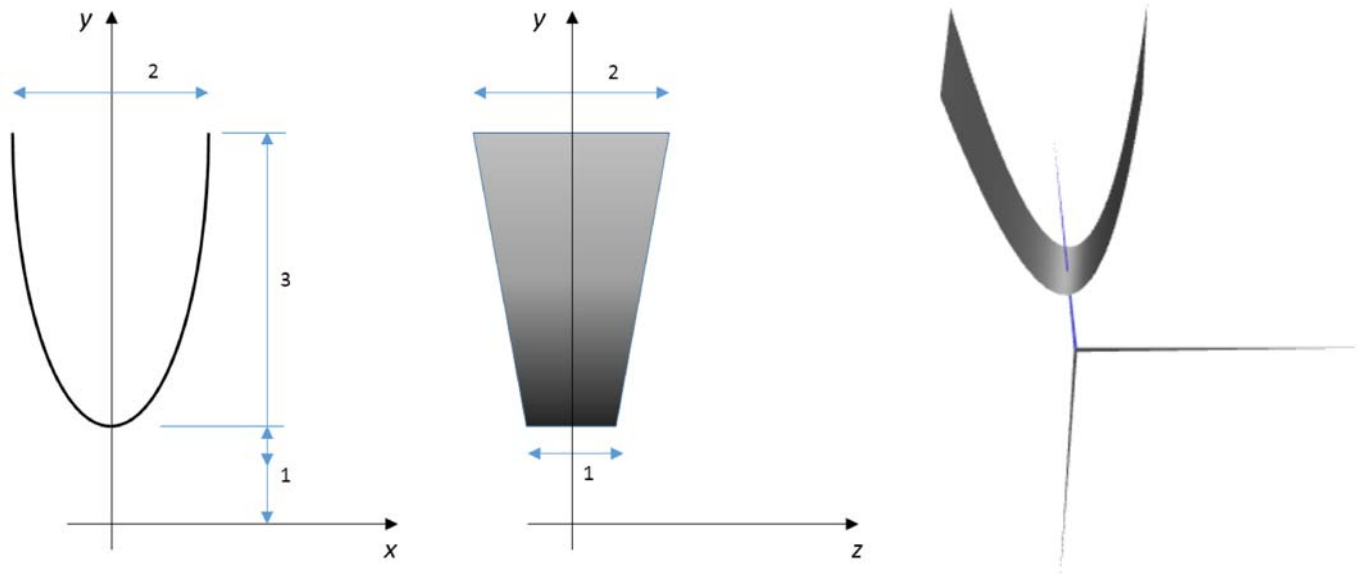

Figura 1: Vista de frente, lateral e em perspetiva, da superfície curva pretendida.

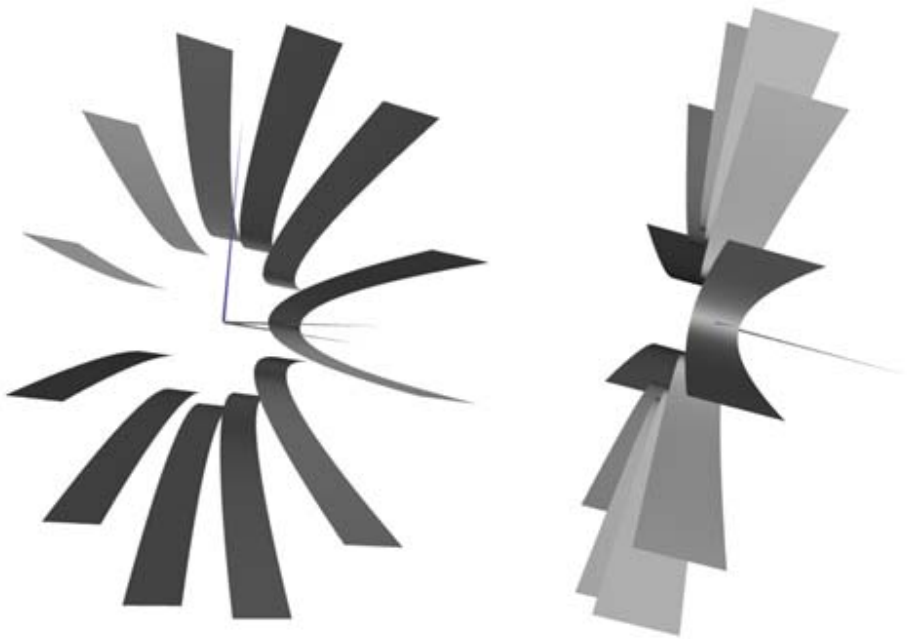

Figura 2: Objeto a visualizar (algumas faces não são visíveis por efeito do *back face culling*).

Desenvolva um ficheiro LSX que desenhe no ecrã um objeto como o que é apresentado na figura 2 (não devem existir outros objetos em cena). Considere uma ou mais fontes de luz, caracterizadas de modo a possibilitar uma fácil observação das características do objecto.

Porto e FEUP, 25 de outubro de 2017
A. Augusto de Sousa,
Jorge Barbosa,
Rui Rodrigues,
Alexandre Carvalho

Resumo da Linguagem LSX modificada

Nota: as alterações à especificação inicial encontram-se marcadas a ***vermelho, bold, itálico.***

```
<SCENE>


<INITIALS>
    <frustum near="ff" far="ff"/>                       <!-- frustum planes-->

    <!-- All the following transformations are mandatory; neutral values may be used -->
    <translate x="ff" y="ff" z="ff" />              <!-- initial translate -->
    <rotation axis="cc" angle="ff" />               <!-- initial rotation 3 -->
    <rotation axis="cc" angle="ff" />               <!-- initial rotation 2 -->
    <rotation axis="cc" angle="ff" />               <!-- initial rotation 1 -->
    <scale sx="ff" sy="ff" sz="ff" />               <!-- initial scaling -->
    <reference length="ff" />                       <!-- axis length; "0" means no axis-->
</INITIALS>


<ILLUMINATION>
    <ambient r="ff" g="ff" b="ff" a="ff" />         <!-- global ambient -->
    <doubleside value="tt" />                       <!-- double or single side illum. -->
    < background r="ff" g="ff" b="ff" a="ff" />     <!-- background color -->
</ILLUMINATION>


<LIGHTS>
    <LIGHT id="ss">                                     <!-- light identifier -->
        <enable value ="tt" />                          <!-- enable/disable -->
        <position x="ff" y="ff" z="ff" w="ff" />
                <!-- light position; w=1: point light; w=0: directional light -->

        <ambient r="ff" g="ff" b="ff" a="ff" />         <!-- ambient component -->
        <diffuse r="ff" g="ff" b="ff" a="ff" />         <!-- diffuse component -->
        <specular r="ff" g="ff" b="ff" a="ff" />        <!-- specular component -->
    </LIGHT>

    <!-- NOTE: this block "LIGHT" must be repeated as necessary with
                different "id". At least one light should be present.
 -->

</LIGHTS>
```

```
<TEXTURES>
    <TEXTURE id="ss">
        <file path="ss" />                              <!-- path to file -->
        <amplif_factor afs="ff" aft="ff" />             <!-- dx/afs, dy/aft -->
    </TEXTURE>

    <!-- NOTE: this block "TEXTURE" must be repeated as necessary with different "id" -->

</TEXTURES>



<MATERIALS>

    <MATERIAL id="ss">
        <shininess value="ff" />
        <specular r="ff" g="ff" b="ff" a="ff" />        <!-- specular reflection -->
        <diffuse r="ff" g="ff" b="ff" a="ff" />         <!-- diffuse reflection -->
        <ambient r="ff" g="ff" b="ff" a="ff" />         <!-- ambient reflection -->
        <emission r="ff" g="ff" b="ff" a="ff" />        <!-- emission component -->
     </MATERIAL>

    <!-- NOTE: the "MATERIAL" block may be repeated as required. Each defined material
             requires a distinct "id". At least one material should be present. -->

</MATERIALS>

<NODES>

    <ROOT id="ss" />     <!-- identifier of root node of the scene graph; this node must -->
                              must be defined in one of the following NODE declarations -->

    <NODE id="ss">       <!-- defines one intermediate node; may be repeated as necessary -->

        <!-- next two lines are mandatory -->
        <MATERIAL id="ss" />      <!-- this superimposes the material received from parent node
                                       id="null" maintains material from parent node -->
        <TEXTURE id="ss" />       <!-- declared texture superimposes the texture received from parent node
                                         id="null" maintains texture from parent node
                                         id="clear" clears texture declaration received from parent node →

        <MATMARK id="ss" />      <!-- opcional -->

        <!-- geom. transf. are optional and may be repeated, in any order, as necessary: -->
        <TRANSLATION x="ff" y="ff" z="ff" />
        <ROTATION axis="cc" angle="ff" />
        <SCALE sx="ff" sy="ff" sz="ff" />

        <!-- declaring descendants, at least one node or one leaf must be present
               descendents may be mixed, nodes and leafs -->
        <DESCENDANTS>
            <NODEREF id="ss" />   <!-- "ss" is the identifier of a node or of leaf; -->
                                  <!"--     may be repeated as necessary. It can refer an -->
                                  <!--  identifier of another node, before or later defined in the file. -->

            <!-- next lines define nodes of type leaf; may be repeated, in any order, as necessary →

            <LEAF type="rect2" args="ff ff   ff ff" corneroffset="ff" />

            <LEAF type="rectangle" args="ff ff   ff ff" />
                  <!-- 2D coordinates for left-top and right-bottom vertices. -->

            <LEAF type="cylinder" args="ff ff ff ii ii tt tt" />
                  <!-- height, bottom radius, top radius, sections along height (stacks), parts per
                       section (slices), top cap(0 or 1), bottom cap (0 or 1) -->
```

```
            <LEAF type="sphere" args="ff ii ii" />
                  <!-- radius, sections along radius, parts per section -->

            <LEAF type="triangle" args="ff ff ff  ff ff ff  ff ff ff" />
                  <!-- coordinates of each vertex -->

            <LEAF type="patch" args="ff ff" >
                <!-- Non-Uniform rational basis spline (NURBS), divisions across U and across V-->

                <CPLINE>      <!-- a controlpoint line declaration;
                                   use as many CPLINE elements as required to characterize the U dimension -->
                    <CPOINT xx="ff" yy="ff" zz="ff" ww="ff" />     <!-- a controlpoint declaration;
                                                                       use as many CPOINT as required to
                                                                       characterize the V dimension -->
                </CPLINE>

            </LEAF>

        </DESCENDANTS>
    </NODE>
</NODES>

</SCENE>
```