

Computação Gráfica (MIEIC)

Trabalho Prático 6


Projeto Final



Objetivos

- Aplicar os conhecimentos e técnicas adquiridas até à data
- Utilizar elementos de interação com a cena, através do teclado e de elementos da interface gráfica

Trabalho prático

Ao longo dos pontos seguintes são descritas várias tarefas a realizar. Algumas delas estão anotadas

com o ícone  (captura de imagem). Nestes pontos deverão, com o programa em execução, capturar uma imagem da execução. Devem nomear as imagens capturadas seguindo o formato "CGImage-tp6-TtGgg-x.y.png", em que TtGgg referem-se à turma e número de grupo e x e y correspondem ao ponto e subponto correspondentes à tarefa (p.ex. "CGImage-tp6-T3G10-2.4.png", ou "CGImage-tp6-T2G08-extra.jpg").

Nas tarefas assinaladas com o ícone  (código), devem criar um ficheiro .zip do vosso projeto, e nomeá-lo como "CGFCode-tp6-TtGgg-x.y.zip", (com TtGgg, x e y identificando a turma, grupo e a tarefa tal como descrito acima). Quando o ícone  surgir, é esperado que executem o programa e observem os resultados. No final, devem submeter todos os ficheiros via Moodle, através do link disponibilizado para o efeito. Devem incluir também um ficheiro **ident.txt** com a lista de elementos do grupo (nome e número). Só um elemento do grupo deverá submeter o trabalho.

Preparação do Ambiente de Trabalho

Este trabalho deve ser baseado numa cópia do trabalho anterior (uma sala de aula com, pelo menos dois planos, duas mesas, duas paredes, chão e um cilindro, e um relógio animado).

Iremos acrescentar uma classe de interface que criará uma área de interface gráfica com alguns elementos de interação, e que será também responsável por gerir eventos de teclado. Para tal, é fornecido o ficheiro **MyInterface.js** que devem incluir no projeto da seguinte forma:

- Colocar o ficheiro na mesma diretoria dos restantes ficheiros Javascript do projeto
- editar o ficheiro **main.js** e
 - adicionar '**MyInterface.js**' à lista de ficheiros a incluir
 - substituir no código da função main a referência a **CGFInterface** por **MyInterface**
- Editar o vosso ficheiro de cena (**LightingScene.js**) e
 - acrescentar no método **LightingScene.init**, as seguintes variáveis:

```
this.option1=true; this.option2=false; this.speed=3;
```




- acrescentar ao ficheiro o seguinte método:

```
LightingScene.prototype.doSomething = function ()  
{ console.log("Doing something..."); };
```

1. Criação da classe MyRobot (1 valor)

Neste exercício procura-se criar uma geometria para representar um *robot* que servirá de *avatar*.

1. Crie uma classe **MyRobot** que represente um *robot*. Esta classe será responsável, inicialmente, pelo desenho do robot. Nesta fase será constituída apenas por um triângulo paralelo ao plano XZ, com coordenadas (0.5, 0.3, 0), (-0.5, 0.3, 0), (0, 0.3, 2), ou seja, um triângulo a apontar para +ZZ.
2. Aplique as transformações necessárias para colocar o robot no centro da sala, a apontar (aproximadamente) para o quadro da esquerda.

(1.2 ) (1.2 ) 

2. Controlo do Robot (3 valores)

Neste exercício procura-se criar um mecanismo de controlo para o avatar acima criado. Consulte a classe **MyInterface.js** para ver exemplos de utilização que ajudarão na resolução destes pontos.

1. Crie um mecanismo para controlar o robot utilizando as teclas: rodar para a esquerda ou para a direita, conforme a tecla premida é "A" ou "D", e mover-se no sentido para onde está virado ou no sentido contrário, conforme se pressione "W" ou "S", respetivamente. Para este efeito, deve criar as variáveis ou métodos na classe da cena de forma a poder alterá-las ou invocá-las na classe de interface

(2.1 ) 

3. GUI (3 valores)

Neste exercício procura-se criar uma interface gráfica (GUI) com alguns controlos para alterar parâmetros da cena em tempo de execução.

1. Adicione à GUI um grupo intitulado "Luzes" (remova/comente/substitua o grupo do exemplo). Acrescente ao novo grupo, por cada fonte de luz utilizada, uma checkbox. Cada checkbox (estado on/off) deve permitir alterar o estado (respetivamente acesa/apagada) da fonte de luz que lhe diz respeito.



2. Adicione um botão que pause/retome o mecanismo de animação do relógio da cena;

(3.2 ) (3.2 ) 

4. Modelação do Robot (4 valores)

Neste exercício procura-se criar uma geometria algo mais complexa para substituir a representação do robot anterior. O Robot deverá ser composto por vários elementos, todos eles com capacidade para conter texturas aplicadas. As texturas deverão ser criadas/selecionadas pelos alunos.

1. O corpo deverá ser composto por um cilindro (com pelo menos um topo). O Robot terá duas rodas (estilo Segway) sendo estas compostas por dois cilindros (com pelo menos um topo); as texturas das rodas deverão permitir perceber facilmente a sua rotação.



2. Construa a cabeça do Robot como uma semi-esfera com textura. Finalmente crie dois cilindros para os braços (com os dois topos, ou com semi-esferas nos topos).

(4.2 ) 

3. Construa uma interface para a seleção das texturas, integrada na GUI da aplicação. Deve para o efeito usar um controlo do tipo “drop-down”. Para implementar este tipo de controlos, sugere-se:
 - a. Declarar na cena um array **robotAppearances** que contenha as várias appearances possíveis
 - b. Declarar um dicionário **robotAppearanceList** que mapeie as strings identificando cada appearance ao seu índice em **robotAppearances**.
 - c. Declarar na cena uma variável **currRobotAppearance** que identifique o índice da appearance selecionada/atual
 - d. Adicionar um controlo na interface que fique associado a **currRobotAppearance** e a **robotAppearanceList**
(exemplos em <http://workshop.chromeexperiments.com/examples/gui/#2--Constraining-Input>)
 - e. Ajustar o código de desenho da cena ou do robot para que seja usada a **appearance** correta.
4. Execute e altere as texturas usando a GUI

(4.4 ) (4.4 ) 

5. Animação do Robot (4 valores)

Neste exercício procura-se animar os vários elementos que compõem o Robot.

1. Por forma a simular o movimento das rodas estas deverão girar de acordo com o movimento do Robot (estilo Segway). No movimento para a frente e para trás, ambas rodam na mesma direção. No movimento de rotação do robot, as rodas rodarão em direções opostas.




2. Os braços do Robot deverão oscilar alternadamente quando o robot se move para a frente e para trás (na rotação não se movem).



3. Ao ser pressionada a tecla ‘H’, o braço direito deve erguer-se, acenar três vezes e voltar à posição inicial (durante este tempo, o movimento oscilatório do braço direito, se ativo, deve ser ignorado).

(5.3 ) (5.3 ) 

6. Criação de um impostor (3 valores)

1. Altere/substitua a parede que contém a janela, dotando-a de um "buraco" que permita olhar de dentro da sala para o exterior, como se a janela estivesse aberta. A "moldura" da janela deve continuar a ser visível. 
2. Crie um plano no exterior da sala, paralelo ao plano ZY, de coordenada X negativa, ao qual deve aplicar, como textura, a imagem de uma paisagem. Deve adequar a posição e tamanho do plano, de forma a que, quando visto do interior da sala, dê a sensação de proporções realistas.

(6.2 ) (6.2 ) 

Notas sobre a avaliação do trabalho:

O enunciado incorpora, em cada alínea, a sua classificação máxima, correspondendo esta a um ótimo desenvolvimento, de acordo com os critérios seguintes, e que cumpra com todas as funcionalidades enunciadas.

Sem perda da criatividade desejada num trabalho deste tipo, não serão contabilizados, para efeitos de avaliação, quaisquer desenvolvimentos além dos que são pedidos.

Para efeitos de avaliação do trabalho e tendo em atenção as cotações mencionadas anteriormente, serão considerados os seguintes critérios:

- Software (2 valores):
 - Estruturação e eficiência das rotinas mais críticas em termos de tempo de cálculo,
 - Criatividade e qualidade da Interação (intuitividade, coerência, facilidade de utilização);
- Criação do Robot (1 valor)
- Controlo do Robot (3 valores)
- GUI (3 valores)
- Refinamento do Robot (4 valores)
- Animação do Robot (4 valores)
- Impostor (3 valores)

De acordo com a formulação constante na ficha de disciplina, a avaliação deste trabalho conta para a classificação final com um peso de:



$50\% * 40\% = 20\%$ da nota final.

Discussão do trabalho

A avaliação do trabalho decorrerá durante a última aula prática, e consistirá numa apresentação/discussão de 10 minutos de cada grupo com o respetivo docente das aulas práticas.

Checklist

Até ao final do trabalho deverá submeter as seguintes imagens e versões do código via Moodle, respeitando estritamente a regra dos nomes, bem como o ficheiro ident.txt com a identificação dos membros do grupo:

-  Imagens (6): 1.2, 3.2, 4.2, 4.4, 5.3, 6.2
(nomes do tipo " CGFImage-tp6-TtGgg-x.y.png")
-  Código em arquivo zip (6): 1.2, 2.1, 3.2, 4.4, 5.3, 6.2
(nomes do tipo "CGFCode-tp6-TtGgg-x.y.zip")