

# Computação Gráfica (MIEIC)

## Trabalho Prático 5


### Animação


## Objetivos


- Animar objectos de uma cena
- Utilizar máquinas de estados para controlar a animação da cena

## 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 "CGFImage-tp5-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. "CGFImage-tp5-T3G10-2.4.png", ou "CGFImage-tp5-T2G08-extra.jpg").

Nas tarefas assinaladas com o ícone  (código), devem criar um ficheiro .zip do vosso projeto, e nomeá-lo como "CGFCode-tp5-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). Deve acrescentar aos ficheiros do projeto a textura **clock.png** fornecida no Moodle.

# 1. Animação de relógio de aula

Neste exercício procura-se criar um relógio para a sala de aula, criada nos trabalhos anteriores, que esteja animado de forma realista (ver Fig. 1).

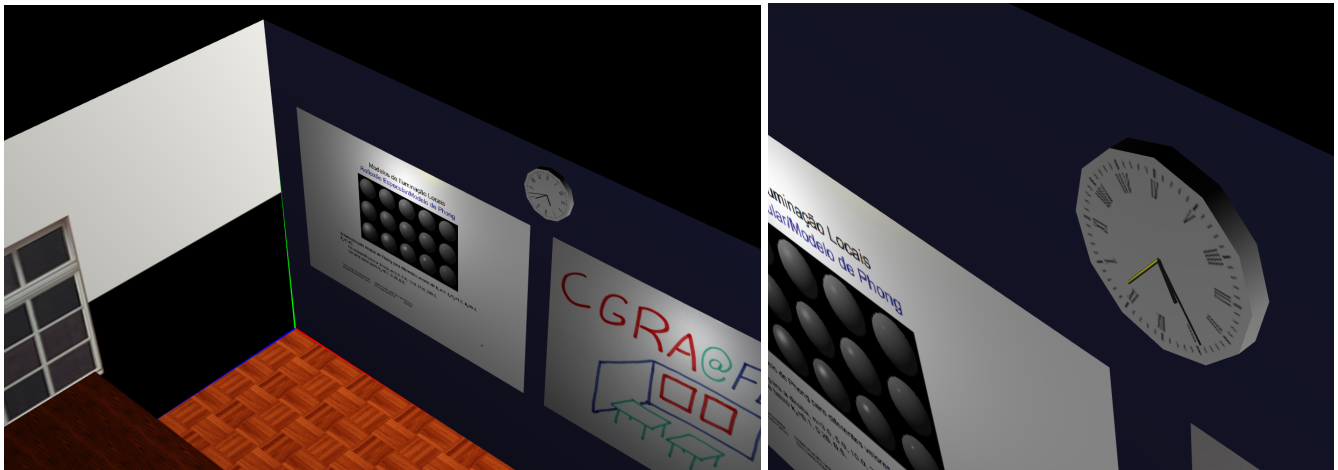




Figura 1: Exemplo do relógio e sua colocação na cena

1. Crie uma classe **MyClock** que represente um relógio, composto por um cilindro (de doze *slices* e uma *stack*) e que tenha pelo menos um dos topos, para o mostrador. Aplique no topo do cilindro a textura **clock.png** e adicione uma instância da classe à cena. Coloque o cilindro centrado, entre os dois quadros mas acima deles. 
2. Crie uma classe **MyClockHand**, que represente um ponteiro de relógio "normalizado" - base na origem, a apontar na direção +YY e comprimento unitário. Esta classe deve ter um método **setAngle(angle)**, que define o ângulo de rotação (em graus) do ponteiro à volta do eixo dos ZZ, no sentido dos ponteiros do relógio. Por exemplo, **setAngle(90)** deverá corresponder à posição "3 horas" - ponteiro na direção +XX.
3. Acrescente três objetos da classe **MyClockHand** ao **MyClock**, que devem ser desenhados com tamanhos diferentes correspondendo aos ponteiros das horas, minutos e segundos.
4. Use no construtor de **MyClock** o método **setAngle(angle)** de cada um dos ponteiros para os colocar inicialmente na posição correspondente a 3 horas, 30 minutos e 45 segundos. 
5. Acrescente à classe **LightingScene** a função **update(currTime)** (por agora sem código), e no final do método **init()** invoque o método da cena **this.setUpdatePeriod(100)**. Desta forma, o método **update** será executado regularmente, com uma cadência fixa **aproximada** de 100 milissegundos - taxa de atualização de 1/100 milissegundos. O argumento que será recebido pela função **update(currTime)** é o **tempo do sistema** em mili-segundos.
6. Crie uma função **update** na classe **MyClock**, e invoque-a na função **update(currTime)** da **LightingScene**, passando-lhe o valor de tempo recebido. Preencha a função **update(currTime)** da classe **MyClock** de forma a atualizar a rotação dos três ponteiros em função do tempo real recebido. Por exemplo, a cada segundo, o ponteiro dos segundos deve rodar o equivalente a 1/60 de uma rotação completa, e o dos minutos 1/60/60.

O cálculo deve ser feito em função do tempo decorrido, e independente da taxa de atualização - uma alteração à taxa de atualização não deve alterar a velocidade angular média dos ponteiros (podendo no entanto aparentar maior ou menor fluidez)

(1.6  ) (1.6  ) 



## Extra: Avião de papel

Crie um avião de papel (classe **MyPaperPlane**), com uma dimensão adequada à cena. Esse avião deverá levantar vôo de uma das mesas em direcção à parede da janela, numa trajectória ascendente, e ao bater na

janela, cair virado para baixo e ficar em repouso no chão. (extra  ) (extra  ) 

# 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 (2): 1.6, extra (nomes do tipo "CGFImage-tp5-TtGgg-x.y.png" )
-  Código em arquivo zip (2): 1.6, extra (nomes do tipo "CGFCode-tp5-TtGgg-x.y.zip")