

# Computer Graphics (MIEIC)

## Practical work 6


### Final project



## Goals

- Apply the knowledge and skills acquired to date
- Utilizarelementos interaction with the scene, through the keyboard and interface elements  
print shop

## Practical work

Over the following points are described various tasks to accomplish. Some of them are noted

with the icon  (Image capture) .Nestes points should, with the program running, capturaruma image execution. Should nomearas images captured following the format "**CGFImage-TP6-TtGgg-xy.png**" ,on what **TtGgg** refers to the class and group number and **x** and **y** match point and sub-point corresponding to the task (eg "**CGFImage-TP6-T3G10-2.4.png**" or "**CGFImage-TP6-T2G08-extra.jpg**" ).

The tasks marked with the icon  (Code) must create a .zip file of your project, and nomeáo as "**CGFCode-TP6-TtGgg-xyz.zip**", (with **TtGgg**, **x** and **y** identifying the class, group and task as described above). When the icon  arise, it is expected to execute the program and observe the results. In the end, should submit all files via Moodle, using the link provided for this purpose. They should also include a file **ident.txt** with the elements of list group (name and number). Only one member of the group must submit the work.

## Preparation Desktop

This work should be based on a copy of the previous work (a classroom with at least two planes, two tables, two walls, floor and a cylinder, and an animated clock). We acrescentaruma interface class that will create a graphical interface area with some

interaction elements, which will also be responsible porgerireventos of teclado.Para such is provided the file **MyInterface.js** which should include the project as follows:



- Place the file in the same board of other JavaScript files project
- edit the file **main.js** and
  - add '**MyInterface.js**' the file list to include
  - replace the function code main reference to **CGFInterface** per **MyInterface**
- Edit your scene file (**LightingScene.js**) and
  - add in the method **LightingScene.init** The following variables:  
`this.option1 = true; this.option2 = false; this.speed = 3;`
  - the file to add the following method:  

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

## 1. Creation of MyDrone class (0.5 points)

In this exercise it is seeking to create a geometry to represent a *drone* to serve as *avatar* .

1. Crie uma classe **MyDrone** que represente um *drone*. Esta classe será responsável, inicialmente, desenhando o drone. Nesta fase será apenas para 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 apontando para +ZZ.
2. Aplique as transformações necessárias para colocar o drone no centro da sala, apontando (Aproximadamente) para a esquerda da tela.

(1.2  ) (1.2  ) 

## 2. Drone Control (3 values)

Neste ano está procurando criar um mecanismo de controle para o avatar criado acima. veja a classe **MyInterface.js** por exemplo para usar que vai ajudar a resolver estes pontos.

1. Crie um mecanismo para controlar o drone usando as teclas: virar para a esquerda ou para a direita conforme a tecla pressionada é **"THE"** ou **"D"** e mover na direção que você está enfrentando ou na direção oposta, conforme pressione **"W"** ou **"S"** respectivamente. Para subir e descer, devem ser usadas as teclas **"I"** e **"J"**, respectivamente. Você deve criar as variáveis ou métodos necessários para suportar esses movimentos na classe da cena e assim o drone pode mudar ou depender deles na classe da interface.

(2.1  ) 

## 3. GUI (2.5 points)

Neste ano está procurando criar uma interface gráfica de usuário (GUI) com alguns controles para alterar os parâmetros da cena em tempo de execução.

1. Adicione GUI a um grupo chamado "Lights" (remover / comentar / substituir o grupo amostra). Adicione o novo grupo para cada fonte de luz usada, um checkbox. Cada checkbox (On / off status) deve permitir que você altere o estado (respectivamente ligado / desligado) da fonte de luz em questão.



2. Adicione um botão para pausar / retomar o relógio da animação da cena;

(3.2  ) (3.2  ) 

## 4. Modelling of Drone (4 points)

Neste exercício procura-se criar uma geometria algo mais complexa para substituir a anterior representação do drone. O drone deve ser composto de vários elementos, todos com capacidade para conter aplicadas texturas. A textura deve ser criada / selecionada pelos alunos. Pretende-se que o drone tenha uma estrutura semelhante à da Fig. 1.



### 1. On the body of the Drone:

The body should be composed of two crossed cylinder-shaped arms, being the front one end of the  $+$  (More), representing  $+$ .

B. At each end of the arms there will be a cylinder as the basis for the propellers.

w. The center of the Drone will consist of a semi-sphere. (May use drums but suffer a penalty in the price).

### 2. The drone will have two "legs", each of which has:

The one parallelepiped base

b. Duas tiras (surfaces) joining the base body to the principal. Estas tiras devem ser curvilíneas (ex. parabólicas, segmentos circulares, bézier). The use of other simple geometries is possible, but will not have the full quote.

### 3. The Drone will have four propellers same as shown in the picture, the propeller center consists by a semi-sphere and the propeller blades will be comprised of flat rollers.

(**NOTICE** Number of slices and stacks should be reduced in view of the scale of objects)



### 4. Construa uma interface para seleção de texturas, integrada GUI application. must for this purpose, usar um control type "drop-down" type. For implementar este controls, it suggests:

a. Declarar scene in an array **droneAppearances** containing the various appearances possible

b. Declarar a dictionary **droneAppearanceList** that maps strings identifying each appearance to their index in **droneAppearances**.

c. Declarar na scene a variable **currDroneAppearance** that identifies the index of appearance selected / current.

d. Adicionar a control interface in that it is associated with **currDroneAppearance** and the **droneAppearanceList**

(Examples <http://workshop.chromeexperiments.com/examples/gui/#2--Constraining-Input>)

e. Ajustar the drawing code of the scene or the drone to be used to *appearance* correct.

f. Incluir at least a texture which easily identifying the front of the drone.

5. Execute and change the textures using the GUI.



## 5. Animation of the Drone (4 points)

In this exercise seeks to animate the Drone and its propellers.

1. The movement of a real Drone is controlled by the direction and speed of its Propellers in accordance with the diagram of Fig. 2.

- Consider three different speeds of slow rotation (G), corresponding to 0.2, 1 and 10 revolutions per second, respectively.
- Anime as hélices do seu Drone de forma a replicar o tipo de movimento exemplified in Fig. 2

Normal (N) and fast (R)

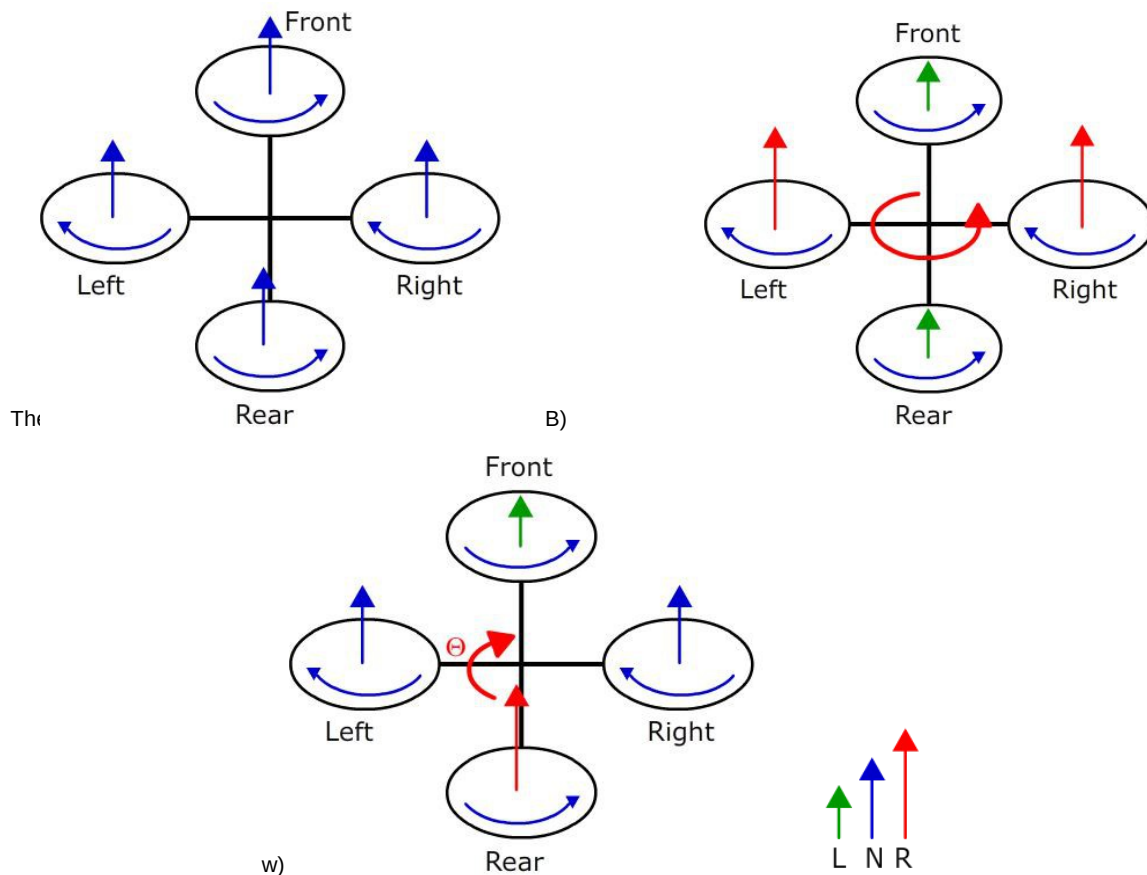


Figure 2: Examples of the three modes of motion a) static mode;. b) Rotational Movement (Yaw); c) move forward with tilt (pitch).

2. When the drone moved forward or backward, incline is slightly forwards or backwards, respectively. Implement this animation.

3. For display purposes, should include a scale factor in the GUI allowing see rotating propellers in "slow motion". This factor should range between 0.1 and 2.0. A value of 1.0 correspond to the rotational speeds mentioned above, a factor of 2 will propellers rotate to twice its original speed, a factor 0.5 by half. This factor should not change the drone's movement speed itself, only the rotation of the propeller.

(5.3  ) (5.3  ) 


## 6. transportation of cargo (4 points)

The drone must be able to collect a box through a wire / hook, and in another destination. The user location can raise or lower the hook using keys, and when it enters into contact with the box, it will automatically be "attached" to the hook. When the box is resting on destination, soltase automatically hook.

1. Implemente a class that represents the cable drone (a cylinder with 3 slices, variable length, with an object on the end to represent the hook) it's from
2. Inclua an instance of this object in the Drone, so that it descends down the center of the drone (for simplicity, the cable must not be affected by the drone of the slope, ie can always stay upright) .Must implement a method that allows to determine the position of end of the cable in the scene.
3. Acrescente support for the 'P' key and 'L' to rise and lower the cable, respectively.
4. Implemente a class that will represent the load to be implement a recolhida. Deve method whereby the load position in the scene.
5. Adicione an instance of this object to the scene.
6. Implemente a mechanism to determine whether the hook is touching the load (ie, the hook position is close to the load position). If they come into contact, should be given some visual indication (e.g. change the texture or color of the load or the hook), and the load must pass to keep the hook.

(6.6  ) 

7. Implemente a class that will represent the load destination. You must implement a method identifying the position of the object in the scene.
8. Adicione an instance of this object to the scene.
9. Implementeum mechanism que permitadeterminarseacarga (ouoganch) is close enough to the destination. When this happens, the load must be placed in destination and no longer "stuck" to the hook (and be some visual indication of this separation).

(6.9  ) (6.9  ) 

## Notes on evaluation of the work:

The statement includes, in each point, its highest rating, this corresponding to a great development, according to os critérios seguintes, which complies with todas as listed features. No loss of desired creativity in a work of this kind, will not be counted for the purposes

evaluation, any developments that are beyond the requests.

For work assessment purposes and taking into account the above quotations, the following criteria will be considered:

- Creation of Drone (0.5 points)
- Drone Control (3 values)
- GUI (2.5 points)
- Refinement of Drone (4 points)
- Animation of Drone (4 points)
- Animation and cable control (4 points)
- Software (2 values):
  - Structuring and efficiency of the most critical routines in terms of calculation time,
  - Criatividade e qualidade da Interação (intuitiveness, consistency, Ease-use);

According to the wording in the discipline of form, the evaluation of this work has to the final classification with a weight:



$50\% * 40\% = 20\%$  of the final grade.

### work Discussion

A avaliação do trabalho decorrerá durante a última aula prática, e consistirá numa presentation / discussion 10 minutes for each group with the appropriate teaching of practical classes.

## Check list

Until final do work must submit the following images and versions of code via Moodle, strictly respecting the rule of names, and the ident.txt file with the identification of Group members:

-  Pictures (6): 1.2, 3.2, 4.4, 5.3, 6.6, 6.9  
(Type names "CGFImage-TP6-TtGgg-xy.png")
-  Code zip file (6): 1.2, 2.1, 3.2, 4.4, 5.3, 6.9  
(Type names "CGFCode-TP6-TtGgg-xyz.zip")