 Nome ______________________________ Número ________

COMPUTAÇÃO GRÁFICA E INTERFACES

FCT/UNL – Ano letivo 2016/2017
Teste 2A – 2016.12.12
Don't remove the staples. You can use the back pages as scracth paper.
You have 2h to finish the test.

1. (3.5/20)

Classify each sentence as (T)rue or (F)alse. Each wrong answer will subtract 50% of its score.

In the axonometric projection, line parallelism is preserved.	
In the perspective projection, the line parallelism is never preserved.	
In the axonometric projection, when both parameters are set to nonzero values, the angles are never preserved.	
In the perspective projection, the number of vanishing points is related to the location of the centre of projection.	
The projection type more appropriate to photo realism is the axonometric projection.	
The lookAt() function returns a matrix that includes a translation and a rotation.	
The lookAt() function can be used to specify an arbitrary axonometric projection.	
The matrix returned by the lookAt() function allows us to convert points from camera coordinates to world coordinates.	
All the aliasing problems can be alleviated by an increase of the device resolution.	
The anti-alising through the filtering method allows us to increase image sharpness by increasing the contrast.	
The Sutherland-Hodgeman algorithm can be generalized to non-rectangular windows, if they are convex in shape.	
In a scene graph, if we change the order of the branches, the resultant scene does not change..	
In an optimized scene graph, we don't need to use a transformation stack and we only need to keep the current transformation matrix.	
Ina scene graph, the order of the transformations found in a certain branch is irrelevant for the final result since they are always applied in the sequence T.R.S.	

2. (2.5/20)

Consider the ambient and diffuse reflection models discussed in the lectures, $I_{rgb}= I_a K_a + I_p K_d. \cos \theta$, evaluated at an arbitrary visible point on the surface of an object. Classify each sentence as (T)rue or (F)alse. Each wrong answer will subtract 50% of its score.

Light reflected towards the viewer depends on the orientation of the surface with respect to the viewer.	
Light reflected towards the viewer depends on the orientation of the surface with respect to the light source.	
I_a is a scalar value.	
I_p is a vetor.	
I_p is associated with a specific light source.	
I_a is associated with a specific light source.	
K_d represents the colour of the object when lit by a white light.	
For multiple light sources the second term is repeated with a varying value for K_d.	
The diffuse reflection is maximum when light strikes the surface in a perpendicular direction..	

3. (3/20)

Consider that a 3D application wants to offer the user with the possibility of specifying a view volume and a type of projection as similar as possible to the ones obtained with a normal photo camera.

a) From the following functions that serve to specify a view volume:

1. `frustum(left, right, bottom, top, near, far)`

2. `perspective(fovy, aspect, near, far)`
3. `ortho(left, right, bottom, top, near, far)`,

which one would you consider more appropriate to the task at end? Justify you answer!

b) As you know, the functions referred in a) produce a matrix capable of transforming the specified view volume into a canonical (standard) one that spans from -1 to 1 in all the three axes. What is the importance of this transformation to the implementation of the 3D pipeline?

c) The lookAt(eye, at, up) allows us to establish a relation between the camera and the world coordinate frames. Let the matrix returned by that function be known as **M**. Fill the gaps with values that will validate the identities below.

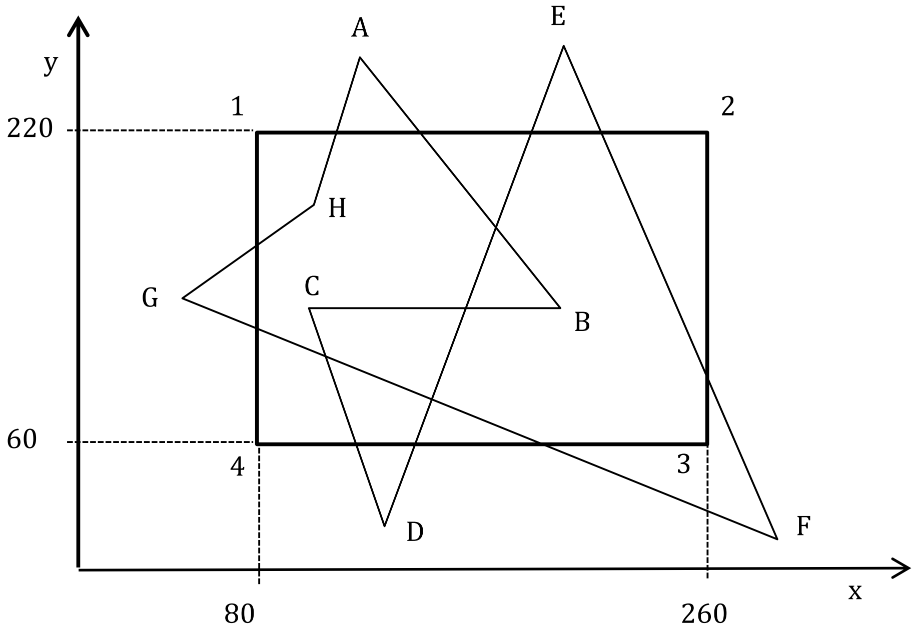
M . eye = ____________________

M . ____________________ = $[0\ 1\ 0]^T$

(1/||at-eye||) **M**(at-eye)= ____________

4. (5/20)

Given the polygons P=[A, B, C, D, E, F, G, H] and Q=[1, 2, 3, 4], being this last one the clipping window, with the coordinates given in the figure:



Polygon P will be clipped against window Q by applying the Sutherland-Hodgeman algorithm, for which the following sequence has been established: Clip Left -> Clip Bottom -> Clip Right -> Clip Top.

a) Specify the polygons that result from each stage of the algorithm:

Clip Left:

Clip Bottom:

Clip Right:

Clip Top:

b) How many edges does the clipped polygon contain? ____________________

Let us imagine that each edge of polygon P will now be clipped by window Q using the Cohen-Sutherland line clipping algorithm.

c) Fill the table below with the decision of the algorithm for each edge of P (**accept/reject** or **intersection** with a window limit line), as well as the total number of possible intersections with the delimiting lines and the indication of the first intersection that is effectively computed. Admit that the bits are assigned from left to right using the same order the algorithm in a) used: Left, Bottom, Right, Top.

d)

Edge	Decision	**total number of intersections** with the limits	line equation for the **1st intersection**
AB			
BC			
CD			
DE			
EF			
FG			
GH			
HA			

e) Paint, in the figure, that area that would be filled by the fill area (scanline) algorithm applied to the **initial** polygon P!

f) How many non-empty entries would the edge table (TA) contain? Justify with the indication of the indices of each of those entries, as well as the respective contents. **Note**: You can identify the index of an entry by providing the y-value of a known point (for instance y_B):

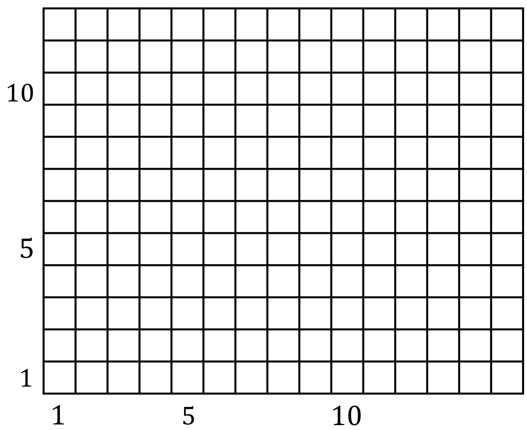
g) What is the contents of the Active Edges Table (TAA) immediately before filling the current scanline for each of the following scanlines?

y_3: y_D:

y_H: y_C:

5. (3/20)

In the following pictures each square represents a pixel in a grayscale device. White (value 255) is the background color and black (value 0) will be used to draw the primitives.



1/32	4/32	1/32
4/32	12/32	4/32
1/32	4/32	1/32

a) Fill, in the left figure, the squares that would be painted by the midpoint line algorithm when applied to a line segment that goes from pixels (2,2) to (13,5).

b) To the result obtained in a) we want to apply the filter presented on the right side to reduce the aliasing problems. What is the final colour of the pixel with coordinates (3)?

c) Can you imagine any inconvenient of applying the method in b) to smooth text in a screen? Justify your answer!

d) Explain in detail how you would apply the filter on the edges of the screen.

6. (3.0/20)

The following WebGL program is already optimized with respect to the number of push and pop operations.

 Nome ______________________________ Número ________

1. multTranslate([10,0,0])
2. multScale([2,2,1])
3. PushMatrix()
4. multRotY(40°)
5. multScale ([2,1,2])
6. PushMatrix()
7. multTranslate ([0,1,0])
8. multRotZ(5°)
9. multRotX(30°)
10. P1()
11. PopMatrix ()
12. multRotX(-30°)
13. multRotZ(20°)
14. P2()
15. PopMatrix()
16. multRotY(35°)
17. multScale([3,2,2])
18. multRotX(25°)
19. PushMatrix()
20. multRotZ(10°)
21. P3()
22. PopMatrix()
23. multScale([3,1,0.5])
24. multTranslate(2,0,0])
25. P4()

a) Draw the corresponding scene graph:

b) If, in a certain system we could use a kind of super node, allowing us to group a sequence of transformations in a pre-determined order (T.Rz.Ry.Rx.S), What would be the total number of nodes required?

Mark the corresponding braches of your graph that resulted in a simplified (less nodes) version.