Graphics Systems and Interaction

Normal Call 2023-02-01

No. _____ Name _____

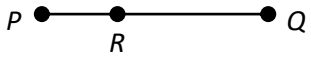
Assessment duration: 45 minutes
Value of each question: marked with brackets
Multiple choice questions: each wrong answer deducts 1/3 of the question's value

Part I **10%**

a. **[3.3]** The visualization, on the screen of a common computer, of a SVG (Scalable Vector Graphics) image

   i. Is only possible with systems equipped with a graphics processing unit (GPU)
   ii. Requires sophisticated pattern recognition techniques
   iii. Requires a rasterization operation
   iv. None of the above

b. **[3.3]** Given two different points P and Q and the affine combination $R = \alpha P + (1 - \alpha)Q$, what value should be assigned to α so that point R gets positioned two times closer to P than to Q?



   i. $\alpha = -0.33$
   ii. $\alpha = 0.33$
   iii. $\alpha = 1 - 0.33$
   iv. None of the above

c. **[3.3]** When using a perspective projection

   i. The viewing volume has the shape of a parallelepiped
   ii. The viewing volume has the shape of a pyramid trunk
   iii. The apparent size of objects being displayed decreases with decreasing the distance from the camera
   iv. None of the above

d. **[3.3]** Which of the following polygon mesh coding technique avoids drawing each edge twice?

   i. Explicit
   ii. Pointers to a vertex list
   iii. Pointers to an edge list
   iv. None of the above

e. **[3.3]** Knowledge of the normal vector is not necessary to compute

   i. Ambient lighting component
   ii. Diffuse lighting component
   iii. Specular lighting component
   iv. None of the above

f. **[3.3]** A texture mapping function

   i. Returns, for each point of the texture space, the corresponding point of the object's surface
   ii. Describes the shape used to wrap the object
   iii. May be based on the parametric description of the object's surface
   iv. All the above

Graphics Systems and Interaction

Normal Call 2023-02-01

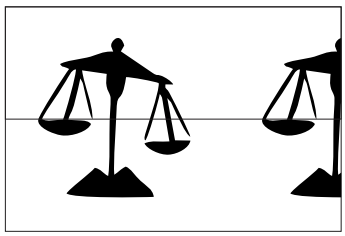
No. _____ Name _____

g. Part II 20%

a. **[4.0]** Suppose you want to map the texture represented in Figure 1 to a rectangle so that it looks like the one shown in Figure 2. Enter the texture coordinates corresponding to each polygon vertex.

Figure 1

Figure 2



v0: _____, _____ v1: _____, _____

v2: _____, _____ v3: _____, _____

b. **[2.0]** Consider a cube made of a pink material (1.0, 0.5, 1.0) illuminated by a single sky-blue light source (0.0, 0.5, 1.0). What are the primary components (R, G, B) of the resulting color? Specify the calculations made.

R = _____ G = _____ B = _____

c. **[1.4]** An *Object3D* and a *Group* are different mainly because *Group*

i. Allows sub-objects
ii. Allows you to embed meshes
iii. Makes code easier to understand
iv. Allows you to integrate materials

d. **[1.4]** In an *OrthographicCamera*, the *far* option is useful

i. For setting the field-of-view (*fov*)
ii. To avoid drawing objects that are too distant from the camera
iii. For controlling the scene resolution
iv. To enable shadows

e. **[1.4]** Library *lil-gui* allows you to easily

i. Change object properties
ii. Inform users of changes in object properties
iii. Group object properties in folders
iv. All the above

No. _____ Name _____

v.

f. **[1.4]** Specifying the vertices of 3D solid faces always with the same winding (clockwise – CW – or counterclockwise – CCW – when viewed from the outside) allows you

i. To speed up rendering
ii. To group faces easily
iii. To map textures easily
iv. To use indices in vertex identification

g. **[1.4]** In order to get shadows in a 3D scene you must enable them

i. In lights
ii. In materials and renderer
iii. In lights, meshes and renderer
iv. In lights, materials and renderer

h. **[1.4]** An *Environment Map* is particularly useful for simulating

i. Shadows
ii. Lighting
iii. Reflections
iv. Physics

i. **[1.4]** In order to quickly implement an user-controlled viewing system you can use

i. *OrbitControls*
ii. *CameraControls*
iii. *ViewControls*
iv. *GUIControls*

j. **[1.4]** A *RayCaster* is very useful in three.js to allow users to

i. Get more realistic renderings
ii. Change the lighting in the scene
iii. Show information related to 3D objects
iv. Interact with 3D objects in the scene

k. **[1.4]** A *RoughnessMap* allows you to

i. Adjust the height of vertices in a mesh
ii. Adjust the position (*x*, *y*) of vertices in a mesh
iii. Define more or less smooth/diffuse areas
iv. Define more or less transparent/opaque areas

l. **[1.4]** Materials' *fog* property is useful for the following scenario

i. To simulate fog in a forest background
ii. When you want to define variable fog densities
iii. To adjust the influence of lighting on fog
iv. To represent a fog-free room with a foggy exterior view