

Graphics Systems and Interaction

Normal Season

2015-01-30

N.o _____ Name _____

Exam duration: 75 minutes

Question points: in brackets

Multiple choice questions: a wrong answer deducts 1/3 of the question points

Theoretical Part

30%

- a. [2.5] In a graphics system with an RGBA 1024 x 1024 x 32 bits frame buffer
- i. It's possible to display images with 32 million colors
 - ii. Each pixel consists of 11 bits to the red component, 11 bits to the green and 10 bits to the blue, in a total of 32 bits
 - iii. It's possible to display images with 256 levels of transparency
 - iv. All of the above
- b. [2.5] Consider the planar object represented in Figure (a). Which one of the following transforming sequences transforms the object into the one in Figure (b)?

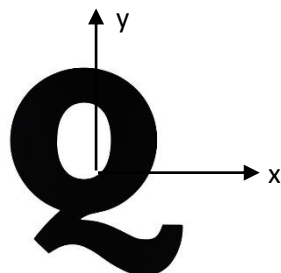


Figure (a)



Figure (b)

- i. `glRotated(-90.0, 0.0, 0.0, 1.0); glScaled(-1.0, 1.0, 1.0);`
 - ii. `glScaled(-1.0, 1.0, 1.0); glRotated(90.0, 0.0, 0.0, 1.0);`
 - iii. `glRotated(90.0, 0.0, 0.0, 1.0); glRotated(180.0, 1.0, 0.0, 0.0);`
 - iv. All of the above
- c. [2.5] Given two different points P and Q , the point that results of the affine combination $R = (1 - \alpha)P + \alpha Q$, ($\alpha = 1.5$)
- i. Matches point P
 - ii. Matches the midpoint of line segment PQ
 - iii. Is nearer to Q than to P
 - iv. None of the above

- d. **[2.5]** Which one of the following polygon mesh coding techniques allows computing in constant time the nine types of adjacencies between vertices, edges and faces?
- i. Pointers to a vertex list
 - ii. Pointers to an edge list
 - iii. Winged-Edge
 - iv. None of the above
- e. **[2.5]** The equation system $x = v \cdot \cos(u)$, $y = v \cdot \sin(u)$, $z = v$, $0 \leq u < 2\pi$ e $0 < v \leq 1$ corresponds to
- i. The implicit surface of a sphere
 - ii. A parameterization of the surface of a cylinder
 - iii. The implicit surface of a cone
 - iv. None of the above
- f. **[2.5]** Which properties should the material of a highly polished object exhibit?
- i. Strongly reflect ambient lighting and a shininess constant = 1.0
 - ii. Strongly reflect diffuse lighting and a low shininess constant
 - iii. Strongly reflect specular lighting and a high shininess constant
 - iv. None of the above
- g. **[2.5]** Which values should the attenuation factors exhibit in order to simulate a situation in which the reflected light intensity decreases to $\frac{1}{4}$ when the distance between the light source and the lit object doubles?
- i. Constant factor = 1.0; linear factor = 0.0; quadratic factor = 0.0
 - ii. Constant factor = 0.0; linear factor = 1.0; quadratic factor = 0.0
 - iii. Constant factor = 0.0; linear factor = 0.0; quadratic factor = 1.0
 - iv. None of the above
- h. **[2.5]** In OpenGL texture mapping, the trilinear filtering method
- i. Uses the texel that lies nearest to the center of the pixel within the nearest mipmap
 - ii. Uses a weighted linear average of the 2×2 array of texels that lie nearest to the center of the pixel within the nearest mipmap
 - iii. Uses the nearest texel in each of the two nearest best choice of mipmaps and then interpolates linearly between these two values
 - iv. Uses a weighted linear average of the 2×2 array of texels in each of the two nearest best choice of mipmaps and then interpolates linearly between these two values

Graphics Systems and Interaction

Normal Season

2015-01-30

N.o _____ Name _____

Practical Part

40%

Multiple choice questions: a wrong answer deducts 1/3 of the question points

Note: Unless otherwise specified, always assume the default camera position

- a. Consider a scene consisting of a pendulum (supposedly hanging from the ceiling of a room) with geometry and movement as shown in Figure (a).

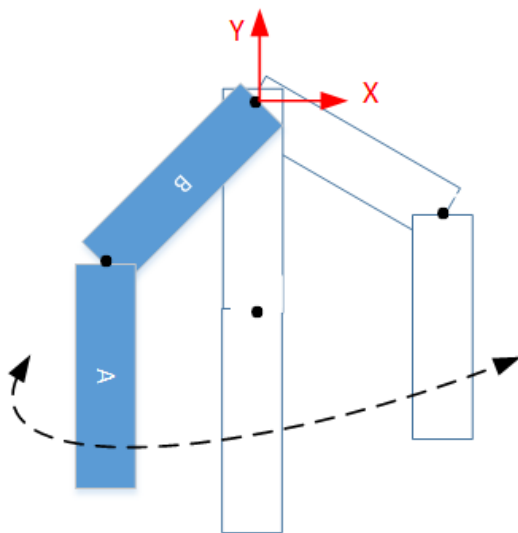


Figura (a)

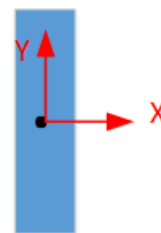


Figura (b)

- i. **[4.0]** Write the pendulum drawing function; it receives as a parameter the leaning angle of segment B with relation to the vertical direction (i.e., 0° corresponds to the resting position; -45° corresponds to the far left; 45° to the far right). Consider the existence of a function `drawSegment()` that draws an origin-centered rectangle, as shown in Figure (b).

```
void drawPendulum(float angle)
{
```

}

- ii. **[4.0]** Write the draft code of a timer that animates the pendulum (assume a frictionless continuous movement).

```
void Timer(int value)
```

```
{
    glutTimerFunc(100, Timer, 0);
```

}

Graphics Systems and Interaction

Normal Season

2015-01-30

N.o _____ Name _____

- b. Assume that you want to model a scene consisting of two pendula side by side and that you want to reuse the function `drawPendulum()` of question a.i.

- i. **[0.5]** Which constants should you define for this scene?

- ii. **[0.5]** Which model variables should you define for this scene?

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

```
void drawTwoPendula()
{
```

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

}

Graphics Systems and Interaction

Normal Season

2015-01-30

N.o _____ Name _____

c. **[4.0]** Enumerate the necessary steps to draw an overlay over a scene that presents helps to the user; for example the character health level or the score.

- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____

Graphics Systems and Interaction

Normal Season

2015-01-30

N.o _____ Name _____

- d. **[4.0]** Assume that you want to set a light source to simulate a flashlight in the hand of a game character; it should always be pointing in the same direction than the character. Write the code to set such light source, pointing out any constants and/or model variables you may need.

```
void setFlashLight()  
{
```

```
}
```

```
void display()  
{
```

```
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
```

```
    ...
```

```
    glTranslatef(modelo.x, modelo.y, modelo.z);
```

```
    glRotatef(modelo.dir, 0.0F, 1.0F, 0.0F);
```

```
    drawCharacter();
```

```
    ...
```

```
    glFlush();
```

```
}
```