Graphics Systems and Interaction

Recourse Season 2015-02-12

N.o _____ Name _____

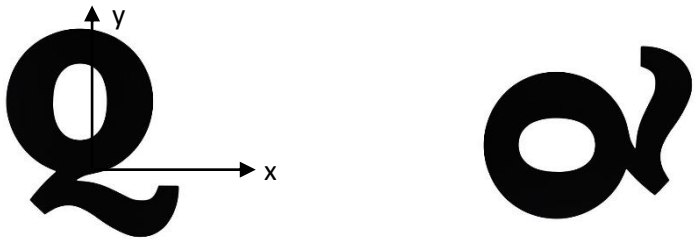
Assessment duration: 75 minutes **Value of each question:** marked with brackets **Multiple choice questions:** each wrong answer deducts 1/3 of the question's value

Theoretical Part **30%**

a. **[2.5]** What's the size in bytes of an RGBA 1024 x 1024 x 32 bits frame buffer?

   i. 1 Megabyte
   ii. 2 Megabyte
   iii. 4 Megabyte
   iv. None 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)*?



a) *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, 0.0, 1.0, 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 $R = (1 - \alpha)P + \alpha Q$, ($\alpha = 0.3$)

   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 is less efficient?

   i. Explicit
   ii. Pointers to a vertex list
   iii. Pointers to an edge list
   iv. Winged-Edge

e. **[2.5]** The equation system $x = k \cdot \cos(u)$, $y = k \cdot \sin(u)$, $z = v$, $0 \leq u < 2\pi$ e $0 \leq v \leq 1$, k = constant 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]** Phong's diffuse component

   i. Can only be defined for directional light sources
   ii. Is characteristic of materials such as shiny metal
   iii. Does not depend of the viewer's position
   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 triples when the distance between the light source and the lit object decreases to one third?

   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 GL_NEAREST_MIPMAP_NEAREST filtering method

   i. Is not used for magnification, i.e. when a single pixel on the screen corresponds to a tiny portion of a texel
   ii. Uses a weighted linear average of the 2 x 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. None of the above

Graphics Systems and Interaction

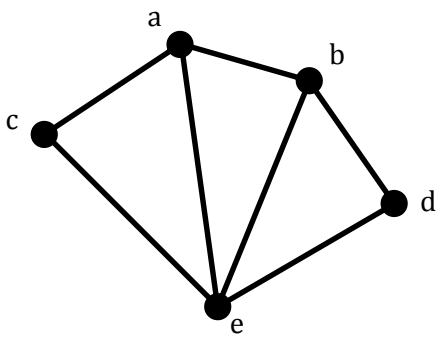
Recourse Season 2015-02-12

N.o _____ Name _____

Practical Part **40%**

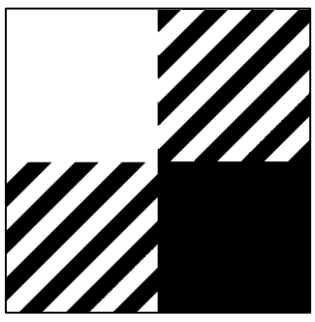
Multiple choice questions: each wrong answer deducts 1/3 of the question's value
Note: Unless otherwise specified, always assume the default camera position

a. **[2.0]** Assume that you want to model the object represented in the figure. Complete the following code fragment, filling the blanks in instructions `glVertex3fv()` with the letters *a* to *e*.




```
glBegin(GL_TRIANGLE_FAN);
     glVertex3fv(____);
     glVertex3fv(____);
     glVertex3fv(____);
     glVertex3fv(____);
     glVertex3fv(____);
glEnd();
```

b. **[2.0]** Complete the following code fragment in order to apply the texture image *a)* to the object represented in figure *b)*.



a)

b)

```
glBegin(GL_QUADS);
     glTexCoord2f(____,____);
     glVertex2f(-2.0, -1.0);
     glTexCoord2f(____, ____);
     glVertex2f(-2.0, 1.0);
     glTexCoord2f(____,____);
     glVertex2f(0.0, 1.0);
     glTexCoord2f(____,____);
     glVertex2f(0.0, -1.0);
glEnd();
```

c. **[1.0]** Assume a scene composed by a single light source that emits only diffuse light with the following components: `{1.0, 0.0, 0.5, 1.0}`; and an object made with a material that reflects diffuse light in the following way: `{0.2, 0.5, 1.0, 1.0}`. What color will result from this combination (in terms of its primary components)?

R = _____ G = _____ B = _____

d. **[3.0]** Assume that you want to simulate the view of a submarine periscope. The position of the periscope base is given by variables `obj.x`, `obj.y` and `obj.z`; the height of the viewfinder in relation to the base is given by variable `obj.height`; and the direction by variable `obj.dir`. Complete the following code fragment:

_____

_____

_____

```
gluLookAt( ________, ________, ________,
           ________, ________, ________,
           ________, ________, ________);
```

Graphics Systems and Interaction

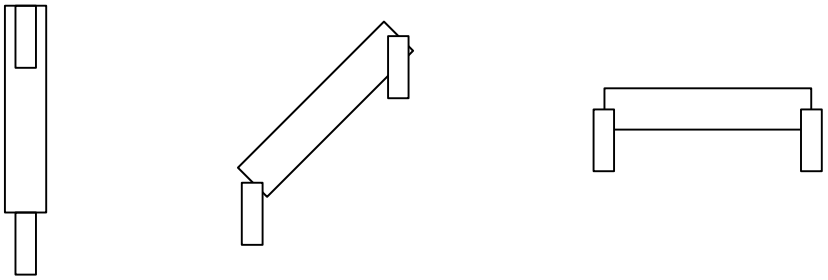
Recourse Season 2015-02-12

N.o _____ Name _____

e. **[4.0]** Draw the tree of the object represented in the figure; assume the existence of the function `box()`, which draws the big rectangle, **centered in the origin**, with the orientation shown in *a)*.

Also assume:

- That the big rectangle rotates around its center and the smaller ones rotate as shown in figures;
- That the size of the smaller rectangles is 30% of the size of the big one;
- The existence of variable `model.rotation`.



a) *b)* *c)*

f. **[3.0]** Write the sequence of OpenGL instructions that draws the big rectangle and one (**only one**) of the smaller ones. Don't forget to write the transformations needed to animate the object.

```
void drawScene()
{
```

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

```
}
```

Graphics Systems and Interaction

Recourse Season 2015-02-12

N.o _____ Name _____

g. **[1.0]** Assume that you want the object to move when you press the arrow keys. Which GLUT callback(s) do you need?

_____

_____

h. **[4.0]** Complete the *timer* function in order to animate the object in the following way:

- Continuously rotates between positions *a)* and *c)* in the figures;
- When it reaches position *c)*, it starts rotating in the opposite direction until it returns to position *a)*;
- Continuously repeat the previous animation.

Use the variables you need.

```
void Timer(int value)
{
    glutTimerFunc(100, Timer, 0);
```

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

```
    glutPostRedisplay();
}
```