EBU7405 A

Joint Programme Examinations 2020/21

EBU7405 3D Graphics Programming Tools

Paper A

Time allowed 2 hours

Answer ALL questions

Use Answer book to answer the questions.

NOT allowed: electronic calculators and electronic dictionaries.

INSTRUCTIONS

1. **You MUST use the supplied Answer book to answer the questions.**
2. You must NOT take **question papers or answer books**, used or unused, from the examination room.
3. Write only with a black or blue pen **and in English**.
4. Do all rough work in the answer book – **do not tear out any pages**.
5. If you use Supplementary Answer Books, tie them to the end of this book.
6. Write clearly and legibly.
7. **Read the instructions on the inside cover.**

Examiners

Dr Marie-Luce Bourguet, Dr Pengwei Hao

Filename: 2021_EBU7405_A

Instructions

Before the start of the examination

1) Place your BUPT and QM student cards on the corner of your desk so that your picture is visible.

2) Put all bags, coats and other belongings at the back/front of the room. All small items in your pockets, including wallets, mobile phones and other electronic devices must be **placed in your bag in advance**. **Possession of mobile phones, electronic devices and unauthorised materials is an offence.**

3) Please ensure your mobile phone is switched off and that no alarm will sound during the exam. **A mobile phone causing a disruption is also an assessment offence.**

4) Do not turn over your question paper or begin writing until told to do.

During the examination

1) You must not communicate with or copy from another student.

2) If you require any assistance or wish to leave the examination room for any reason, please raise your hand to attract the attention of the invigilator.

3) If you finish the examination early you may leave, but not in the first 30 minutes or the last 10 minutes.

4) For 2 hour examinations you may **not** leave temporarily.

5) For examinations longer than 2 hours you **may** leave temporarily but not in the first 2 hours or the last 30 minutes.

At the end of the examination

1) You must stop writing immediately – **if you continue writing after being told to stop, that is an assessment offence.**

2) Remain in your seat until you are told you may leave.

Question 1

a) What is the general format of the `glVertex` function? Explain why it needs several versions and provide examples.

[5 marks]

b) Write an OpenGL orthographic projection statement which guarantees that a sphere of radius 5 can be entirely contained in the viewing volume without distortion.

[5 marks]

c) This question is about the OpenGL camera.

[6 marks]

i) What are the characteristics of the default OpenGL camera?

(3 marks)

ii) Which OpenGL function allows you to set the location and orientation of the camera? How does it work?

(3 marks)

d) Consider the code shown in **Code Box 1** on the next page.

[9 marks]

i) Which two statements can be removed from the code without modifying its output?

(2 marks)

ii) Explain the purpose of the following statement:

```
glGetIntegerv (GL_VIEWPORT, viewport);
```

(4 marks)

iii) Modify the code so that a click on the right button of the mouse terminates the program.

(3 marks)

```
#include <GL/glut.h>

GLdouble W = 640.0;
GLdouble H = 480.0;
GLint size = 30;

void myInit(void) {
   glClearColor (0.0, 0.0, 0.0, 1.0);
   glMatrixMode (GL_PROJECTION);
   glLoadIdentity();
   gluOrtho2D (0, W, 0, H);
}

void mydisplay(){
}

void drawSquare(int x, int y)
{
    GLint viewport[4];
    glGetIntegerv (GL_VIEWPORT, viewport);
    y = viewport[3] - y;
    glShadeModel(GL_SMOOTH);
    glBegin(GL_POLYGON);
        glVertex2i(x+size, y+size);
        glVertex2i(x-size, y+size);
        glVertex2i(x-size, y-size);
        glVertex2i(x+size, y-size);
     glEnd();
     glFlush();
}

void mymouse(int btn, int state, int x, int y)
{
   if(btn==GLUT_LEFT_BUTTON && state==GLUT_DOWN)
      drawSquare(x, y);
}

int main(int argc, char** argv){
      glutInit(&argc, argv);
      glutInitWindowSize(W, H);
      glutCreateWindow("simple");
      glutDisplayFunc(mydisplay);
      glutMouseFunc(mymouse);
      myInit();
      glutMainLoop();
}
```

Code box 1.

Question 2

a) What is Euler's formula for a convex polyhedron? Explain your variables.

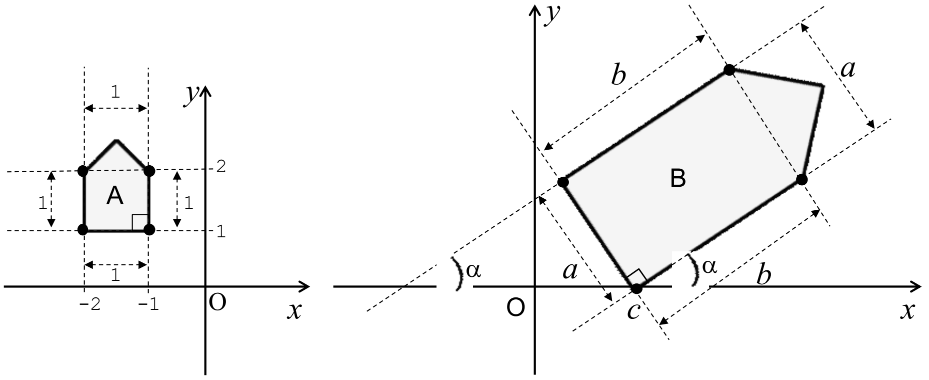
[4 marks]

b) This question is about geometric transformations.

[11 marks]

i) Consider Model A and Model B in **Figure 1**. Give a chain of the basic transformation matrices for translation, scaling and rotation which, when post-multiplied by the homogeneous coordinates of the vertices of Model A, will transform the vertices of Model A into their corresponding vertices of Model B, such that the vertex at $(-1,1)$ of Model A is transformed to the vertex at $(c,0)$ of Model B.

(7 marks)



(Model A) **(Model B)**

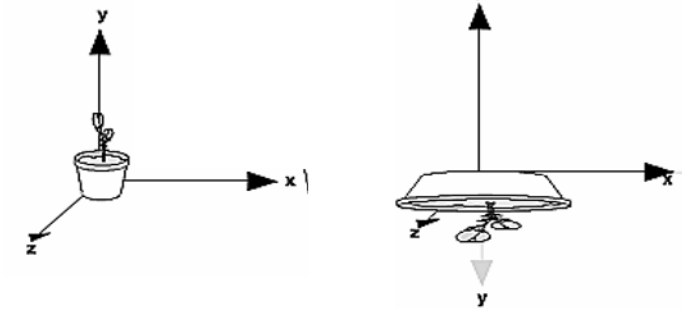
Figure 1.

ii) Compute the composite 2D transformation matrix for the transformations found in question i) above.

(4 marks)

c) Starting from the object model shown on **Figure 2 a)**, find the single modelling transformation that can create the object instance shown in **Figure 2 b)**. Give the corresponding OpenGL command with appropriate arguments. Justify your choice of transformation and arguments.

[6 marks]



a) b)

Figure 2.

d) What are the four printing primary colours used in printing? Give the relations between the four printing primary colours and the three additive primary colours.

[4 marks]

Question 3

a) This question is about lighting calculations in OpenGL. Describe the appearance (colour and shininess) of an object illuminated under the conditions declared in **Code box 2**.

[6 marks]

```
float light_ambient[] = {0.5, 0.5, 0.5, 1.0};
float light_diffuse[] = {1.0, 1.0, 1.0, 1.0};
float light_specular[] = {1.0, 1.0, 1.0, 1.0};
float material_ambient[] = {0.2, 0.3, 0.2, 1.0};
float material_diffuse[] = {0.0, 0.4, 0.4, 1.0};
float material_specular[] = {1.0, 1.0, 1.0, 1.0};
float shininess = 100.0;
```

Code box 2.

b) This question is about WebGL.

[9 marks]

i) What are the roles of the vertex shader and of the fragment shader, respectively? How do they relate to each other and to the framebuffer?

(5 marks)

ii) Consider the WebGL code in **Code box 3**. Briefly explain the following two elements: (1) gl; and (2) program.

(4 marks)

```
function initProgram() {
      const vertexShader = getShader('vertex-shader');
      const fragmentShader = getShader('fragment-shader');

      program = gl.createProgram();
      gl.attachShader(program, vertexShader);
      gl.attachShader(program, fragmentShader);
      gl.linkProgram(program);

      gl.useProgram(program);
      program.aVertexPosition = gl.getAttribLocation(program,
                                         'aVertexPosition');
    }
```

Code box 3.

c) This question is about 3D modelling types and tools.

[10 marks]

i) What are the differences between polygons and NURBS?

(4 marks)

ii) What is the role of the control points in surface modelling?

(3 marks)

iii) In Blender, would you model a donut using polygons or NURBS? Justify your answer.

(3 marks)

Turn over until you reach the "END OF PAPER" line

Question 4

a) This question is about projection.

[16 marks]

i) Consider **Figure 3**. Give the perspective projection transformation matrix for point P(x,y,z) projecting to the view plane at z=2 if the centre of projection is the origin.

(6 marks)

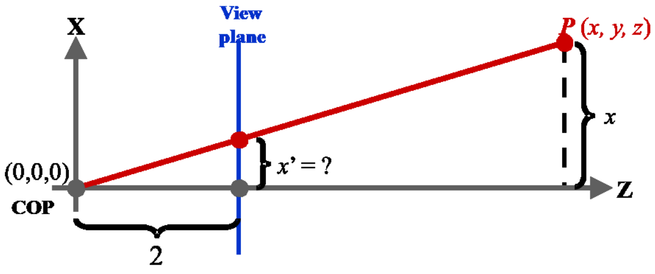


Figure 3.

ii) In the world coordinate system, if the view plane normal is [4,3,0], the view-up vector is [0,0,1], the view reference point is [1,2,3], find the transformation matrix for parallel projection.

(10 marks)

b) This question is about rasterisation. Use the line equation for a line from (x0, y0) to (x1, y1) to derive the mid-point algorithm for line generation.

[9 marks]

END OF PAPER