CCG3013 Computer Graphics
Final Exam Answer Script
January 2022
Set A

Section A (60 marks)

1. Discuss three advantages and three disadvantages of virtual reality (VR) as an application in working environment. (12 marks)

   Advantages:
   - ✓ It only requires virtual reality (VR) equipment to simulate the virtual world. **(2 marks)**
   - ✓ Users can jump into the virtual environment at any time, without the waste of time to commute. **(2 marks)**
   - ✓ Cost saving at long-term, which excluded the expenditure or budget to purchase physical materials for product developments. **(2 marks)**

   Disadvantages:
   - ✓ Users unable to distinguish the reality and virtual world in the near future. **(2 marks)**
   - ✓ Time spent on the perfect fictional world, rather than the imperfect real world. **(2 marks)**
   - ✓ Users could have face mental and physical health problems, due to various differences between the two worlds. **(2 marks)**

2. Explain a unit form in two statements. Then, state two dimensional (2D) primitive elements and the corresponding quantities for a unit form in Figure 1 below. (10 marks)

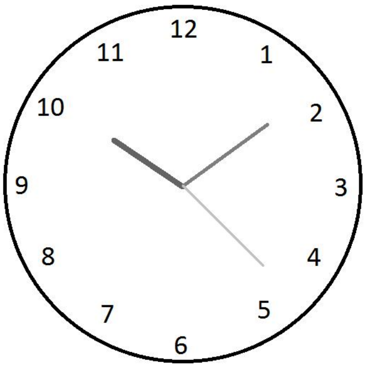

Figure 1: Unit form

A combination of simple objects that generate a more refined object. **(2 marks)**
It can be a composition of 2D shapes, such as the points, lines, triangles, rectangles, circles, ellipses, etc. **(2 marks)**

Shape	Quantity	Marks
Line	3	**(2 marks)**
Circle	1	**(2 marks)**
Text glyph	15	**(2 marks)**

3. Given a three-dimensional (3D) point origin at (23, 18, 15) in the 3D space. Compute the corresponding image point with the following matrix transformations.

(a) Translate with a vector of T(-2, 6, -12) (4 marks)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \text{(2 marks)}$$

$$= \begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & -12 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 23 \\ 18 \\ 15 \\ 1 \end{bmatrix} \quad \text{(1 mark)}$$

$$= \begin{bmatrix} 21 \\ 24 \\ 3 \\ 1 \end{bmatrix} \quad \text{(1 mark)}$$

(b) Rotate clockwise at 45 degrees along x-axis, R(45) (4 marks)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} \quad \text{(2 marks)}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 45 & \sin 45 & 0 \\ 0 & -\sin 45 & \cos 45 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 23 \\ 18 \\ 15 \\ 0 \end{bmatrix} \quad \text{(1 mark)}$$

$$= \begin{bmatrix} 23 \\ 18\cos 45 + 15\sin 45 \\ -18\sin 45 + 15\cos 45 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 23 \\ 23.33 \\ -2.12 \\ 0 \end{bmatrix} \quad \text{(1 mark)}$$

(c) Rotate counter-clockwise at 20 degrees along z-axis, R(-20) (4 marks)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} \text{ (2 marks)}$$

$$= \begin{bmatrix} \cos 20 & -\sin 20 & 0 & 0 \\ \sin 20 & \cos 20 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 23 \\ 18 \\ 15 \\ 0 \end{bmatrix} \text{ (1 mark)}$$

$$= \begin{bmatrix} 23\cos 20 - 18\sin 20 \\ 23\sin 20 + 18\cos 20 \\ 15 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 15.46 \\ 24.78 \\ 15 \\ 0 \end{bmatrix} \text{ (1 mark)}$$

(d) Scale with a factor of (2, 2.5, 1.5) (4 marks)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} \text{ (2 marks)}$$

$$= \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2.5 & 0 & 0 \\ 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 23 \\ 18 \\ 15 \\ 0 \end{bmatrix} \text{ (1 mark)}$$

$$= \begin{bmatrix} 46 \\ 45 \\ 22.5 \\ 0 \end{bmatrix} \text{ (1 mark)}$$

4. Define a glyph and a texture atlas. Then, state and discuss two font libraries that are available for OpenGL. (12 marks)

Glyph is a textured quad for a letter. **(1 mark)**

Texture atlas is a textured quad for a set of letters. **(1 mark)**

Font libraries	**Discussion**
OpenGL native font library	It renders characters in a Bitmap. It only renders two font types, which are Times Roman and Helvetica. The font for each glyph or letter can be customized and rendered pixel by pixel. The library can be extended by using the display list.
OGLFT	OGLFT stands for OpenGL-FreeType Library. It renders text using FreeType font library. It supports Qt graphics user interface (GUI).
OpenGLText	This library is developed by Tristan Lorach, Nvidia developer from Santa Clara, CA. It renders text using TrueType font library. It may support mobile application.
FreeType	It is written in C language. It renders text using FreeType in Bitmap. Cross-platform, which included Linux, iOS, Android, ChromeOS, ReactOS, and Ghostscript.

1 mark will be awarded for each correct font library above, up to two only.

4 marks will be awarded for the corresponding valid discussion, up to two only.

5. Discuss the procedure involved for mesh generation approach to construct a 3D model. (10 marks)

a. In the 3D space, render a 3D model using a set of 3D primitives. **(2 marks)**

b. Join the 3D primitives with union operations. **(2 marks)**

c. Subtract the excess 3D shape with intersect operations. **(2 marks)**

d. Sub-divide 3D faces into a mesh that consisted of either triangles or quadrilaterals. **(2 marks)**

e. Fit the mesh into a UV polynomial function to smooth the surface. **(2 marks)**

Section B (30 marks)

1. Briefly describe Disney's principles of animation. Then, evaluate and justify with three suitable principles of animation for a waterfall of a musical fountain. (10 marks)

It describes twelve basic nature of animation in detailed by Johnston and Thomas (1981) in the book of Disney Animation: The Illusion of Life. **(1 mark)**

Principles	**Justifications**
Staging **(1 mark)**	The plots and scenes that simulate the water particles and form like a sound wave, when the music is playing. **(2 marks)**
Straight ahead action **(1 mark)**	Water flow that output from the water sprinklers is a continuous animation, which drive using the water pump, while it falls along with the gravity. **(2 marks)**
Arc **(1 mark)**	The water pump pushes the water particles that form a trajectory path. For example, a 45-degree angled water sprinkler will generate a maximum arc. **(2 marks)**

Section C (15 marks)

1. Write a function definition in C++ OpenGL to draw an ellipse. (10 marks)

```
void drawEllipse(GLint x, GLint y, (1 mark)
                 GLint r, GLint s, (1 mark)
                 GLfloat o) (1 mark)
  glPushMatrix(); (0.5 marks)
  GLint xp, yp; (1 mark)
  glBegin(GL_POLYGON); (0.5 marks)
    for(int i=0; i<s; i++){ (1 mark)               360*
      xp = (int)(x + w*r*cos(o + 2*Pi/s*i)); (1 mark)
      yp = (int)(y + h*r*sin(o + 2*Pi/s*i)); (1 mark)
      glVertex2i(xp, yp); (1 mark)                  360*
    }
  glEnd(); (0.5 marks)
  glPopMatrix(); (0.5 marks)
}
```

2. Write a function definition in C++ OpenGL to render a wireframe sphere in cyan colour, with radius five for the size, 25 lines of latitudes, and 24 lines for longitude. (5 marks)

```
void Sphere::draw(){ (1 mark)
 glColor3f(0.0f, 1.0f, 1.0f); (1 mark)
 glFrontFace(GL_CW); (1 mark)
 glutWireSphere(5, 25, 24); (1 mark)
 glFrontFace(GL_CCW); (1 mark)
}
```

Section D (15 marks)

1. Develop a complete render function in C++ OpenGL to generate a path of looping wave in yellow colour for a point moving along the x-axis, based on sine function with amplitude 0.1, starting at the coordinates, (0, 400). (15 marks)

```
void render(){ (1 mark)
  glClearColor(1.0f, 1.0f, 1.0f, 1.0f); (1 mark)

  glMatrixMode(GL_PROJECTION); (0.5 marks)
  glLoadIdentity(); (0.5 marks)
  gluOrtho2D(0, 800, 0, 600); (1 mark)

  glClear(GL_COLOR_BUFFER_BIT); (1 mark)

  glColor3f(1.0f, 1.0f, 0.0f); (1 mark)
  sprite01.translate(sprite01.xVel, 0.0); (1 mark)
  sprite01.y += 0.1*sin(sprite01.x*sprite01.Pi()/180); (2 marks)
  if(sprite01.x >= 799){ (1 mark)
    sprite01.x = 0; (0.5 marks)
    sprite01.y = 400; (0.5 marks)
  }
  sprite01.drawPoint(20.0); (1 mark)

  glutSwapBuffers(); (1 mark)
  glutPostRedisplay(); (1 mark)

  glFlush(); (0.5 marks)
  glFinish(); (0.5 marks)
}
```

****