ASSIGNMENT 1B : OPENGL

FUNDAMENTAL FOR COMPUTER GRAPHICS (SCSV 2213-01)

DUE DATE : 4TH MARCH 2020

GROUP MEMBERS :-

1. AIMI BINTI RUSDI B19EC0001
2. AQILAH HANI BINTI MOHD TAUFIK B19EC0006

```
#include <GL/glut.h> // Include the GLUT header file
#include <fstream>
#include <stdlib.h>
#include <iostream>

using namespace std;

void renderbitmap(float x, float y, void* font, char* string) {

      char* c;
      glRasterPos2f(x, y);
      for (c = string; *c != '\0'; c++) {
            glutBitmapCharacter(font, *c);
      }
}

void introscreen() {
      //glColor3f(1.f, 0.f, 0.f);
      char buf[100] = { 0 };
      glColor3f(0.f, 0.f, 1.f);
      sprintf_s(buf, "Eid Mubarak!");//blue
      renderbitmap(-0.27, -0.6, GLUT_BITMAP_TIMES_ROMAN_24, buf);
      glColor3f(0.13f, 0.37f, 0.31f);//green
      sprintf_s(buf, "Maaf Zahir Batin");
      renderbitmap(-0.19, -0.7, GLUT_BITMAP_HELVETICA_12, buf);
      glColor3f(0.f, 0.f, 0.f);//black
      sprintf_s(buf, "Best Wishes from Aimi binti Rusdi");
      renderbitmap(-0.30, -0.8, GLUT_BITMAP_HELVETICA_10, buf);
      glColor3f(0.f, 0.f, 0.f);//black
      sprintf_s(buf, "& Aqilah Hanim binti Mohd Taufik");
      renderbitmap(-0.30, -0.9, GLUT_BITMAP_HELVETICA_10, buf);
}
```

```
void display(void) {
	glClearColor(0.29f, 0.46f, 0.43f, 1.0f); // Clear the background of 
our window to red
	glClear(GL_COLOR_BUFFER_BIT); //Clear the colour buffer (more 
buffers later on)
	glLoadIdentity(); // Load the Identity Matrix to reset our drawing 
locations

	glFlush(); // Flush the OpenGL buffers to the window
}

void play(void) {
	glClearColor(0.29f, 0.46f, 0.43f, 1.0f);
	glClear(GL_COLOR_BUFFER_BIT);

	glBegin(GL_QUADS); //Centre Box
	glColor3f(0.80f, 0.80f, 0.80f); //White
	glVertex2f(-0.4f, -0.4f); //clockwise
	glVertex2f(-0.4f, -1.2f);
	glVertex2f(0.4f, -1.2f);
	glVertex2f(0.4f, -0.4f);
	glEnd();

	glBegin(GL_QUADS); //Left1LeftKetupat
	glColor3f(1.0f, 1.0f, 0.0f); //Yellow
	glVertex2f(-0.8f, 0.3f); //clockwise
	glVertex2f(-0.9f, 0.2f);
	glVertex2f(-0.8f, 0.1f);
	glVertex2f(-0.7f, 0.2f);
	glEnd();

	glBegin(GL_QUADS); //Left1BottomKetupat
	glColor3f(0.0f, 0.8f, 0.0f); //Dark Green
	glVertex2f(-0.7f, 0.2f); //clockwise
	glVertex2f(-0.8f, 0.1f);
	glVertex2f(-0.7f, 0.0f);
	glVertex2f(-0.6f, 0.1f);
	glEnd();

	glBegin(GL_QUADS); //Left1RightKetupat
	glColor3f(1.0f, 1.0f, 0.0f); //Yellow
	glVertex2f(-0.6f, 0.3f); //clockwise
	glVertex2f(-0.7f, 0.2f);
	glVertex2f(-0.6f, 0.1f);
	glVertex2f(-0.5f, 0.2f);
	glEnd();

	glBegin(GL_LINES);
	glVertex2f(-0.7f, 1.4f);
	glVertex2f(-0.7f, 0.3f);
	glEnd();
```

```
        glBegin(GL_QUADS); //Left1TopKetupat
        glColor3f(0.0f, 0.8f, 0.0f); //Dark Green
        glVertex2f(-0.7f, 0.4f); //clockwise
        glVertex2f(-0.8f, 0.3f);
        glVertex2f(-0.7f, 0.2f);
        glVertex2f(-0.6f, 0.3f);
        glEnd();

        glBegin(GL_POLYGON);//Left1KetupatPolygon
        glColor3f(1.0f, 1.0f, 0.0f);
        glVertex2f(-0.7f, 0.0f);
        glVertex2f(-0.8f, -0.4f);
        glVertex2f(-0.7f, -0.3f);
        glVertex2f(-0.6f, -0.4f);;
        glVertex2f(-0.7f, 0.0f);
        glEnd();

        glBegin(GL_QUADS); //Left2LeftKetupat
        glColor3f(1.0f, 1.0f, 0.0f); //Yellow
        glVertex2f(-0.4f, 0.8f); //clockwise
        glVertex2f(-0.5f, 0.7f);
        glVertex2f(-0.4f, 0.6f);
        glVertex2f(-0.3f, 0.7f);
        glEnd();

        glBegin(GL_QUADS); //Left2BottomKetupat
        glColor3f(0.0f, 0.8f, 0.0f); //Dark Green
        glVertex2f(-0.4f, 0.6f); //clockwise
        glVertex2f(-0.3f, 0.5f);
        glVertex2f(-0.2f, 0.6f);
        glVertex2f(-0.3f, 0.7f);
        glEnd();

        glBegin(GL_QUADS); //Left2RightKetupat
        glColor3f(1.0f, 1.0f, 0.0f); //Yellow
        glVertex2f(-0.3f, 0.7f); //clockwise
        glVertex2f(-0.2f, 0.6f);
        glVertex2f(-0.1f, 0.7f);
        glVertex2f(-0.2f, 0.8f);
        glEnd();

        glBegin(GL_LINES);
        glVertex2f(-0.3f, 1.4f);
        glVertex2f(-0.3f, 0.9f);
        glEnd();

        glBegin(GL_QUADS); //Left2TopKetupat
        glColor3f(0.0f, 0.8f, 0.0f); //Dark Green
        glVertex2f(-0.3f, 0.9f); //clockwise
        glVertex2f(-0.4f, 0.8f);
        glVertex2f(-0.3f, 0.7f);
        glVertex2f(-0.2f, 0.8f);
        glEnd();
```

```
glBegin(GL_POLYGON);//Left2KetupatPolygon
glColor3f(1.0f, 1.0f, 0.0f);//yellow
glVertex2f(-0.3f, 0.5f);
glVertex2f(-0.4f, 0.1f);
glVertex2f(-0.3f, 0.2f);
glVertex2f(-0.2f, 0.1f);;
glVertex2f(-0.3f, 0.5f);
glEnd();

glBegin(GL_LINES);
glVertex2f(0.7f, 1.4f);
glVertex2f(0.7f, 0.3f);
glEnd();

glBegin(GL_QUADS); //Right1LeftKetupat
glColor3f(1.0f, 1.0f, 0.0f); //Yellow
glVertex2f(0.8f, 0.3f); //clockwise
glVertex2f(0.9f, 0.2f);
glVertex2f(0.8f, 0.1f);
glVertex2f(0.7f, 0.2f);
glEnd();

glBegin(GL_QUADS); //Right1BottomKetupat
glColor3f(0.0f, 0.8f, 0.0f); //Dark Green
glVertex2f(0.7f, 0.2f); //clockwise
glVertex2f(0.8f, 0.1f);
glVertex2f(0.7f, 0.0f);
glVertex2f(0.6f, 0.1f);
glEnd();

glBegin(GL_QUADS); //Right1RightKetupat
glColor3f(1.0f, 1.0f, 0.0f); //Yellow
glVertex2f(0.6f, 0.3f); //clockwise
glVertex2f(0.7f, 0.2f);
glVertex2f(0.6f, 0.1f);
glVertex2f(0.5f, 0.2f);
glEnd();

glBegin(GL_QUADS); //Right1TopKetupat
glColor3f(0.0f, 0.8f, 0.0f); //Dark Green
glVertex2f(0.7f, 0.4f); //clockwise
glVertex2f(0.8f, 0.3f);
glVertex2f(0.7f, 0.2f);
glVertex2f(0.6f, 0.3f);
glEnd();

glBegin(GL_POLYGON);//Right1KetupatPolygon
glColor3f(1.0f, 1.0f, 0.0f);
glVertex2f(0.7f, 0.0f);
glVertex2f(0.8f, -0.4f);
glVertex2f(0.7f, -0.3f);
glVertex2f(0.6f, -0.4f);;
glVertex2f(0.7f, 0.0f);
glEnd();
```

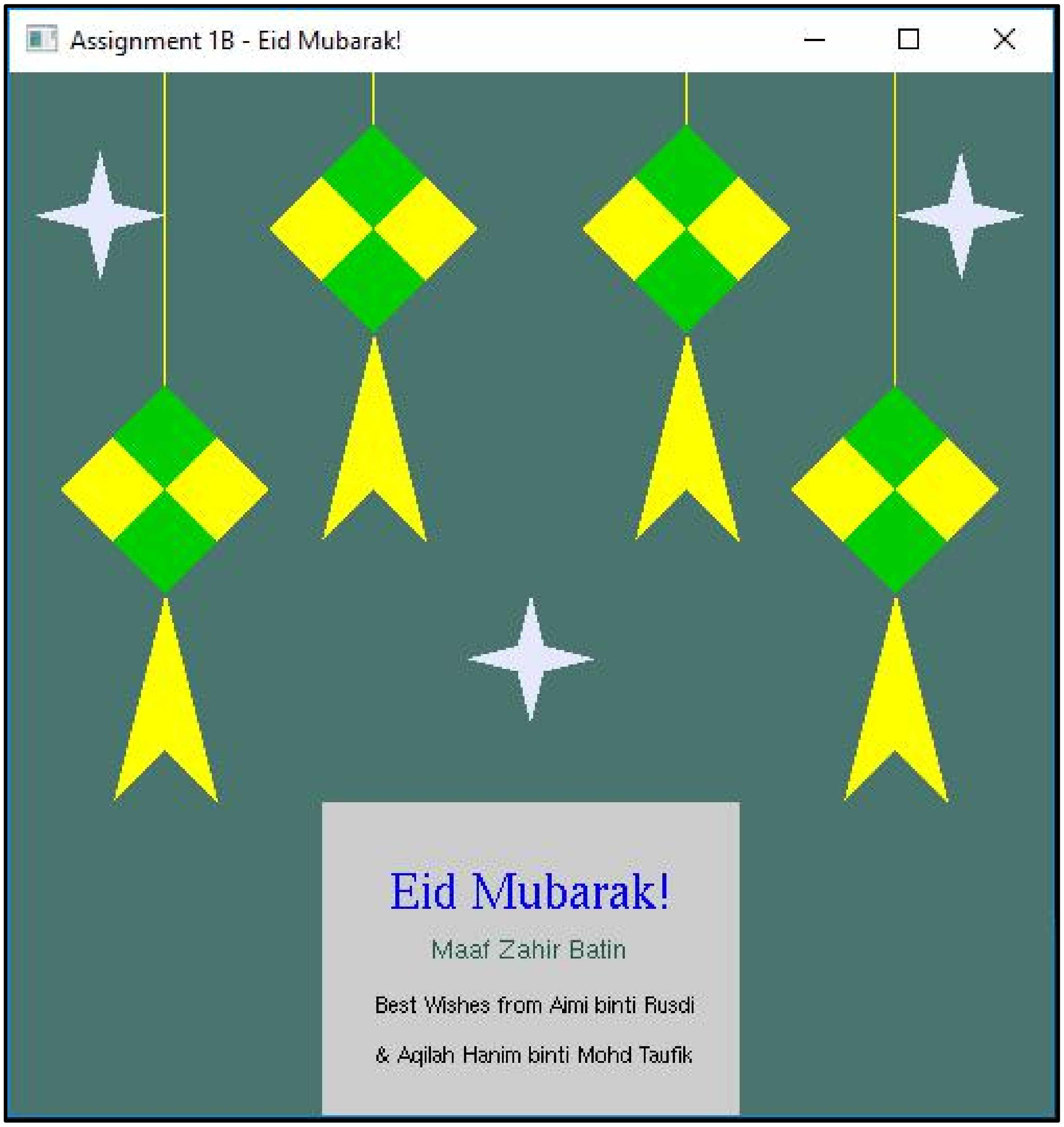
```
    glBegin(GL_QUADS); //Right2LeftKetupat
    glColor3f(1.0f, 1.0f, 0.0f); //Yellow
    glVertex2f(0.4f, 0.8f); //clockwise
    glVertex2f(0.5f, 0.7f);
    glVertex2f(0.4f, 0.6f);
    glVertex2f(0.3f, 0.7f);
    glEnd();

    glBegin(GL_QUADS); //Right2BottomKetupat
    glColor3f(0.0f, 0.8f, 0.0f); //Dark Green
    glVertex2f(0.4f, 0.6f); //clockwise
    glVertex2f(0.3f, 0.5f);
    glVertex2f(0.2f, 0.6f);
    glVertex2f(0.3f, 0.7f);
    glEnd();

    glBegin(GL_QUADS); //Right2RightKetupat
    glColor3f(1.0f, 1.0f, 0.0f); //Yellow
    glVertex2f(0.3f, 0.7f); //clockwise
    glVertex2f(0.2f, 0.6f);
    glVertex2f(0.1f, 0.7f);
    glVertex2f(0.2f, 0.8f);
    glEnd();

    glBegin(GL_LINES);
    glVertex2f(0.3f, 1.4f);
    glVertex2f(0.3f, 0.9f);
    glEnd();

    glBegin(GL_QUADS); //Right2TopKetupat
    glColor3f(0.0f, 0.8f, 0.0f); //Dark Green
    glVertex2f(0.3f, 0.9f); //clockwise
    glVertex2f(0.4f, 0.8f);
    glVertex2f(0.3f, 0.7f);
    glVertex2f(0.2f, 0.8f);
    glEnd();

    glBegin(GL_POLYGON);//Right1KetupatPolygon
    glColor3f(1.0f, 1.0f, 0.0f);//yellow
    glVertex2f(0.3f, 0.5f);
    glVertex2f(0.4f, 0.1f);
    glVertex2f(0.3f, 0.2f);
    glVertex2f(0.2f, 0.1f);;
    glVertex2f(0.3f, 0.5f);
    glEnd();

    glBegin(GL_POLYGON);//Star1Square
    glColor3f(0.90f, 0.91f, 0.98f);
    glVertex2f(-0.8, 0.75);//a
    glVertex2f(-0.85, 0.75);//b
    glVertex2f(-0.85, 0.7);//c
    glVertex2f(-0.8, 0.7);//d
    glEnd();
```

```
	glBegin(GL_POLYGON);//Star1RightTriangle
	glVertex2f(-0.7, 0.725);//e
	glVertex2f(-0.8, 0.75);//a
	glVertex2f(-0.8, 0.7);//d
	glEnd();

	glBegin(GL_POLYGON);//Star1TopTriangle
	glVertex2f(-0.825, 0.85);//f
	glVertex2f(-0.85, 0.75);//b
	glVertex2f(-0.8, 0.75);//a
	glEnd();

	glBegin(GL_POLYGON);//Star1LeftTriangle
	glVertex2f(-0.95, 0.725);//g
	glVertex2f(-0.85, 0.75);//b
	glVertex2f(-0.85, 0.7);//c
	glEnd();

	glBegin(GL_POLYGON);//Star1BottomTriangle

	glVertex2f(-0.825, 0.6);//h
	glVertex2f(-0.8, 0.7);//d
	glVertex2f(-0.85, 0.7);//c
	glEnd();


	glBegin(GL_POLYGON);//Star2Square
	glColor3f(0.90f, 0.91f, 0.98f);
	glVertex2f(0.8, 0.75);//a
	glVertex2f(0.85, 0.75);//b
	glVertex2f(0.85, 0.7);//c
	glVertex2f(0.8, 0.7);//d
	glEnd();

	glBegin(GL_POLYGON);//Star2RightTriangle
	glVertex2f(0.7, 0.725);//e
	glVertex2f(0.8, 0.75);//a
	glVertex2f(0.8, 0.7);//d
	glEnd();

	glBegin(GL_POLYGON);//Star2TopTriangle
	glVertex2f(0.825, 0.85);//f
	glVertex2f(0.85, 0.75);//b
	glVertex2f(0.8, 0.75);//a
	glEnd();

	glBegin(GL_POLYGON);//Star2LeftTriangle
	glVertex2f(0.95, 0.725);//g
	glVertex2f(0.85, 0.75);//b
	glVertex2f(0.85, 0.7);//c
	glEnd();
```

```
      glBegin(GL_POLYGON);//Star2BottomTriangle
      glVertex2f(0.825, 0.6);//h
      glVertex2f(0.8, 0.7);//d
      glVertex2f(0.85, 0.7);//c
      glEnd();

      glBegin(GL_POLYGON);//Star3Square
      glColor3f(0.90f, 0.91f, 0.98f);
      glVertex2f(0.025, -0.1);//a
      glVertex2f(-0.025, -0.1);//b
      glVertex2f(-0.025, -0.15);//c
      glVertex2f(0.025, -0.15);//d
      glEnd();

      glBegin(GL_POLYGON);//Star3RightTriangle
      glVertex2f(0.125, -0.125);//e
      glVertex2f(0.025, -0.1);//a
      glVertex2f(0.025, -0.15);//d
      glEnd();

      glBegin(GL_POLYGON);//Star3TopTriangle
      glVertex2f(0.0, 0.0);//f
      glVertex2f(-0.025, -0.1);//b
      glVertex2f(0.025, -0.1);//a
      glEnd();

      glBegin(GL_POLYGON);//Star3LeftTriangle
      glVertex2f(-0.125, -0.125);//g
      glVertex2f(-0.025, -0.1);//b
      glVertex2f(-0.025, -0.15);//c
      glEnd();

      glBegin(GL_POLYGON);//Star3BottomTriangle
      glVertex2f(0.0, -0.25);//h
      glVertex2f(0.025, -0.15);//d
      glVertex2f(-0.025, -0.15);//c
      glEnd();

      introscreen();

      glFlush();

      glutPostRedisplay();
}
```

```
int main(int argc, char **argv) {
      glutInit(&argc, argv); // Initialize GLUT
      glutInitDisplayMode(GLUT_SINGLE); // Set up a basic display buffer
(only single buffered for now)
      glutInitWindowSize(500, 500); // Set the width and height of the
window
      glutInitWindowPosition(100, 100); // Set the position of the window
      glutCreateWindow("Assignment 1B - Eid Mubarak!"); // Set the title
for the window

      glutDisplayFunc(display); // Tell GLUT to use the method "display"
for rendering
      glutDisplayFunc(play);


      glutMainLoop(); // Enter GLUT's main loop
}
```

OUTPUT



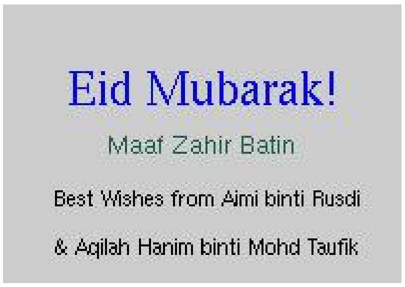
Eid Mubarak!

Maaf Zahir Batin

Best Wishes from Aimi binti Rusdi

& Aqilah Hanim binti Mohd Taufik

EXPLANATION



Eid Mubarak!

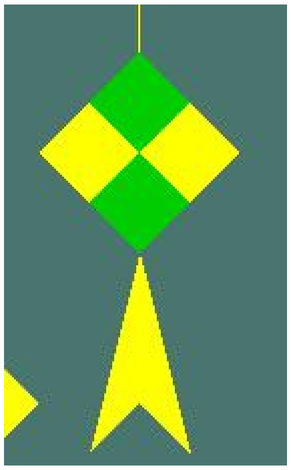
Maaf Zahir Batin

Best Wishes from Aimi binti Rusdi

& Aqilah Hanim binti Mohd Taufik

Used introscreen to make the text visible by inserting the fonts and size at a certain coordinate (x, y).

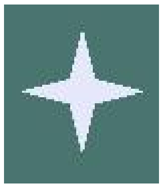
```
glColor3f(0.f, 0.f, 1.f);
      sprintf_s(buf, "Eid Mubarak!");//blue
      renderbitmap(-0.27, -0.6,
GLUT_BITMAP_TIMES_ROMAN_24, buf);
```



In the function play, we inserted shapes with the combination of colour and coordinates to show an image of a "ketupat".

Each were made by keying in squares combined together, polygon for the end and using gl_lines to make a string.

```
glBegin(GL_QUADS); //Left1LeftKetupat
      glColor3f(1.0f, 1.0f, 0.0f); //Yellow
      glVertex2f(-0.8f, 0.3f); //clockwise
      glEnd();

glBegin(GL_LINES);
      glVertex2f(-0.3f, 1.4f);
      glVertex2f(-0.3f, 0.9f);
      glEnd();

glBegin(GL_POLYGON);//Left1KetupatPolygon
      glColor3f(1.0f, 1.0f, 0.0f);
      glVertex2f(-0.7f, 0.0f);
      glEnd();
```



For the stars, we used gl_polygon by creating one square and four triangles combining at a certain coordinate.

The arrangements of coordinates start from top right of the square, followed by the top left, bottom left and lastly bottom right.

To make things easier, we labeled each coordinate a, b, c, d, e, f, g and h.

```
glBegin(GL_POLYGON);//Star1RightTriangle
      glVertex2f(-0.7, 0.725);//e
      glVertex2f(-0.8, 0.75);//a
      glVertex2f(-0.8, 0.7);//d
```

a) glRasterPos2f(x, y);

This function is used to position the pixels and bitmap write operations.

b) glutBitmapCharacter(font, *c);

This function is used to create the fonts and characters that will be displayed.

c) glColor3f(0.f, 0.f, 1.f);

This function is used to give colour to the object from each vertex. Each of the colour is in the order red, green and blue.

d) glClearColor(0.29f, 0.46f, 0.43f, 1.0f);

This function is used to set the background colour to either black or opaque.

e) glClear(GL_COLOR_BUFFER_BIT);

This function is used to clear the background colour buffer.

f) glLoadIdentity();

This function is used to load the Identity Matrix to reset our drawing locations

g) glClear(GL_COLOR_BUFFER_BIT);

This function is used to tell glClear() function which buffers you want to clear.

h) glBegin(GL_QUADS);

This function is used to create a quad which has 4 vertices each shape.

i) glVertex2f(-0.4f, -0.4f);

This function plots where the x and y coordinates of the vertex will be displayed at.

j) glEnd();

This function limits the vertices that defines the created shape.

k) glFlush();

Flush the OpenGL buffers to the window

l) glutPostRedisplay();

This function is used to mark the normal plane of the current window if it is needed to be displayed again.

m) glutInit(&argc, argv);

This function is used to initialize GLUT.

n) glutInitDisplayMode(GLUT_SINGLE);

Set up a basic display buffer (only single buffered for now).

o) glutInitWindowSize(500, 500);

This function is used to set the window's width & height.

p) glutCreateWindow("Assignment 1B - Eid Mubarak!");

This function is used to create a window with the given title.

q) glutDisplayFunc(display);

This function is used to tell GLUT to use the method "display" for rendering.

r) glutDisplayFunc(play);

This function is used to set the display callback for the *current window*.

s) glutMainLoop();

This function is used to enter the event-processing loop.