



**FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI**

**BITM 3213 - INTERACTIVE COMPUTER GRAPHICS**

**LAB SESSION 5**

**C++ - OpenGL**

# Part 1 - C++ - Texture Mapping

1. Code the following program

```
#include <GL\glew.h>
#include <GL\freeglut.h>
#include <stdlib.h>
#include <math.h>

//angle of rotation
float xpos = 0, ypos = 0, zpos = 0, xrot = 0, yrot = 0, angle = 0.0;

//positions of the cubes
float positionz[10];
float positionx[10];

void cubepositions(void) { //set the positions of the cubes

    for (int i = 0; i<10; i++)
    {
        positionz[i] = rand() % 5 + 5;
        positionx[i] = rand() % 5 + 5;
    }
}

//draw the cube
void cube(void) {
    for (int i = 0; i<10; i++)
    {
        glPushMatrix();
        glTranslated(-positionx[i + 1] * 10, 0, -positionz[i + 1] *
            10); //translate the cube
        glutSolidCube(2); //draw the cube
        glPopMatrix();
    }
}

void init(void) {
    cubepositions();
}

void enable(void) {
    glEnable(GL_DEPTH_TEST); //enable the depth testing
    glEnable(GL_LIGHTING); //enable the lighting
    glEnable(GL_LIGHT0); //enable LIGHT0, our Diffuse Light
    glShadeModel(GL_SMOOTH); //set the shader to smooth shader
}

void camera(void) {
    glRotatef(xrot, 1.0, 0.0, 0.0); //rotate our camera on teh x - axis(left and
    right)
    glRotatef(yrot, 0.0, 1.0, 0.0); //rotate our camera on the y - axis(up and
    down)
    glTranslated(-xpos, -ypos, -zpos); //translate the screento the position of our
    camera
}

void display(void) {
    glClearColor(0.0, 0.0, 0.0, 1.0); //clear the screen to black
}
```

```

        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // clear the color buffer
and the depth buffer
        glLoadIdentity();
        camera();
        enable();
        cube(); //call the cube drawing function
        glutSwapBuffers(); //swap the buffers
        angle++; //increase the angle
    }

void reshape(int w, int h) {
    glViewport(0, 0, (GLsizei)w, (GLsizei)h); //set the viewport to the current
window specifications
    glMatrixMode(GL_PROJECTION); //set the matrix to projection
    glLoadIdentity();
    gluPerspective(60, (GLfloat)w / (GLfloat)h, 1.0, 1000.0); //set the perspective
(angle of sight, width, height, , depth)
    glMatrixMode(GL_MODELVIEW); //set the matrix back to model
}

void keyboard(unsigned char key, int x, int y) {
    if (key == 'q')
    {
        xrot += 1;
        if (xrot >360) xrot -= 360;
    }

    if (key == 'z')
    {
        xrot -= 1;
        if (xrot < -360) xrot += 360;
    }

    if (key == 'w')
    {
        float xrotrrad, yrotrrad;
        yrotrrad = (yrot / 180 * 3.141592654f);
        xrotrrad = (xrot / 180 * 3.141592654f);
        xpos += float(sin(yrotrrad));
        zpos -= float(cos(yrotrrad));
        ypos -= float(sin(xrotrrad));
    }

    if (key == 's')
    {
        float xrotrrad, yrotrrad;
        yrotrrad = (yrot / 180 * 3.141592654f);
        xrotrrad = (xrot / 180 * 3.141592654f);
        xpos -= float(sin(yrotrrad));
        zpos += float(cos(yrotrrad));
        ypos += float(sin(xrotrrad));
    }

    if (key == 'd')
    {
        yrot += 1;
        if (yrot >360) yrot -= 360;
    }
}

```

```

    if (key == 'a')
    {
        yrot -= 1;
        if (yrot < -360)yrot += 360;
    }
    if (key == 27)
    {
        exit(0);
    }
}

```

```

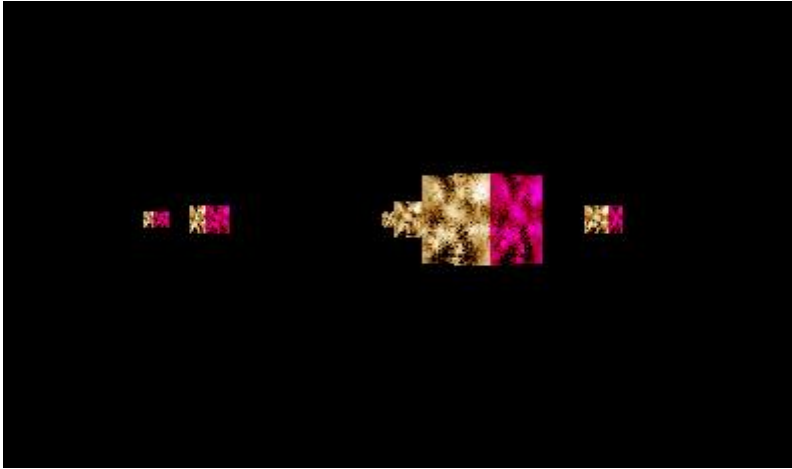
int main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_DEPTH); //set the display to Double
buffer, with depth
    glutInitWindowSize(500, 500); //set the window size
    glutInitWindowPosition(100, 100); //set the position of the window
    glutCreateWindow("A basic OpenGL Window"); //the caption of the window
    init(); //call the init function
    glutDisplayFunc(display); //use the display function to draw everything
    glutIdleFunc(display); //update any variables in display, display can be changed
to anything, as long as you move the variables to be updated, in this case, angle++;
    glutReshapeFunc(reshape); //reshape the window accordingly
    glutKeyboardFunc(keyboard); //check the keyboard
    glutMainLoop(); //call the main loop
}

```

EXERCISE:

1. Try to understand the code

## Part 2: Combine with Part 3 from Lab 4



SUBMIT VIA ULEARN BY 6 am 23 April  
2016.