



เรื่อง

Assignment 5

วิชา

060213291 Computer Graphics

เสนอ

อ. สมชัย เชียงพงศ์พันธุ์

จัดทำโดย

นาย ชีษณุพงศ์ บุญเสริม 6106021610081

นาย ศุภกร สิริพิสิฐวงศ์ 6106021610111

ภาควิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีและการจัดการอุตสาหกรรม

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

คำนำ

รายงานนี้เป็นส่วนหนึ่งของ 060213291 Computer Graphics ที่จัดทำขึ้นเพื่อเป็นการอธิบายและแสดงตัวอย่างของการใช้งานโปรแกรมเพื่อความเข้าใจอย่างง่ายในการทำงานของโปรแกรมมีข้อผิดพลาดประการใดขออภัยมา ณ ที่นี้

คณะผู้จัดทำ

สารบัญ

Assignment 5 ข้อ 1

แนวคิดการออกแบบ	1
ภาพตัวอย่างโปรแกรม	2
Source Code	4

Assignment 5 ข้อ 2

แนวคิดการออกแบบ	10
ภาพตัวอย่างโปรแกรม	11
Source Code	14

แนวคิดการออกแบบ

a. การทำงานกับเมาส์ ดังนี้

- i. สามารถใช้เมาส์ซ้าย เมื่อคลิกทำการเคลื่อนไหวโดยวัตถุหมุนรอบแกน Z
- ii. สามารถใช้เมาส์ขวา หยุดการเคลื่อนไหว

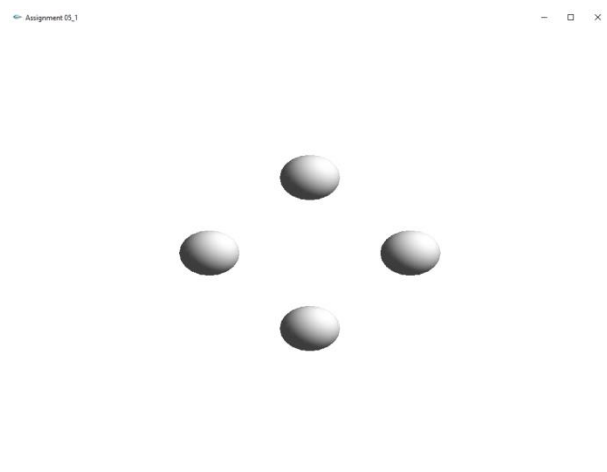
b. การทำงานกับปุ่มคีย์บอร์ด ดังนี้

- i. กดปุ่ม Esc – ออกจากโปรแกรม
- ii. กดปุ่ม R – กำหนดแสงเป็นแสงสีแดง
- iii. กดปุ่ม G – กำหนดแสงเป็นแสงสีเขียว
- iv. กดปุ่ม B – กำหนดแสงเป็นแสงสีน้ำเงิน
- v. กดปุ่ม W – กำหนดแสงเป็นแสงสีขาว
- vi. กดปุ่ม S – กำหนดวัตถุเป็นทรงกลม(Sphere)
- vii. กดปุ่ม C – กำหนดวัตถุเป็นลูกบาศก์(Cube)
- viii. กดปุ่ม T กำหนดวัตถุเป็นโดนัท(Torus)

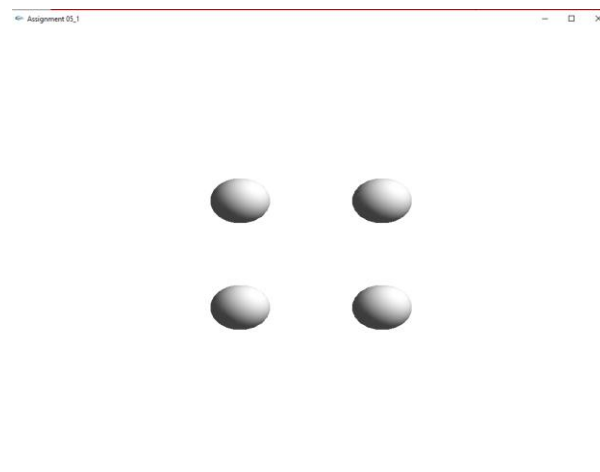
และให้กำหนดตำแหน่งของแสงไว้ที่ $x = 30, y = 30, z = 30$ หรือตำแหน่งที่เหมาะสม

ภาพตัวอย่างโปรแกรม

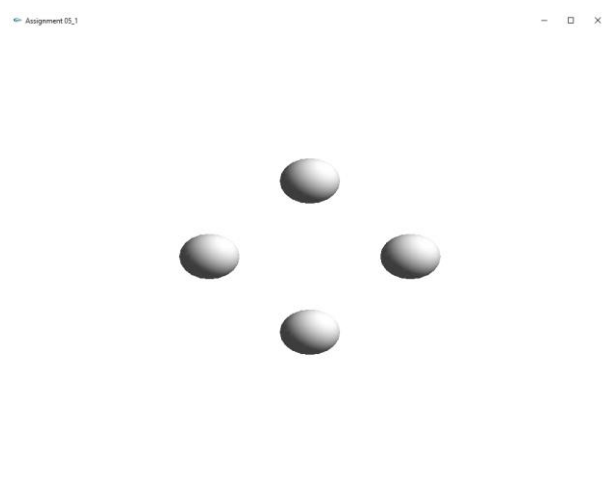
เมื่อเปิดโปรแกรม



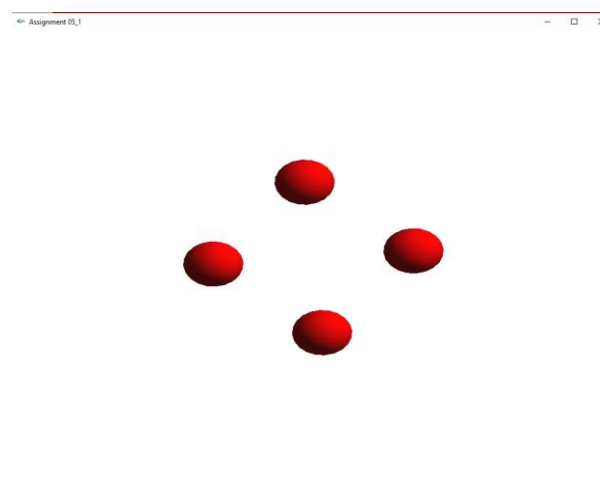
เมื่อคลิกเมาส์ขวา



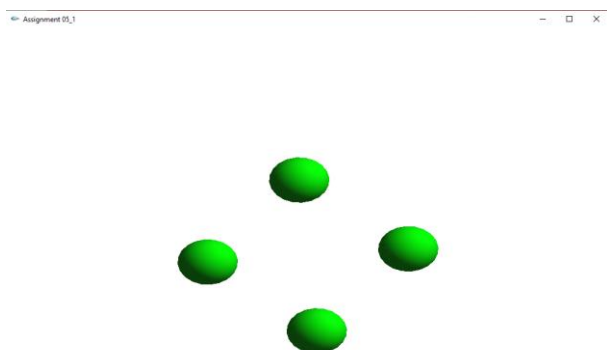
เมื่อคลิกเมาส์ซ้าย



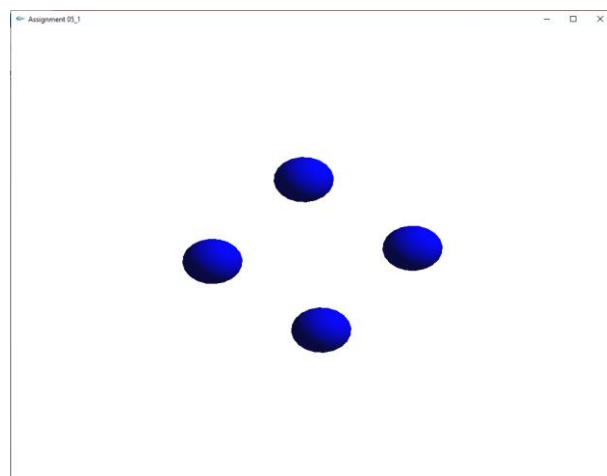
เมื่อกดตัว R



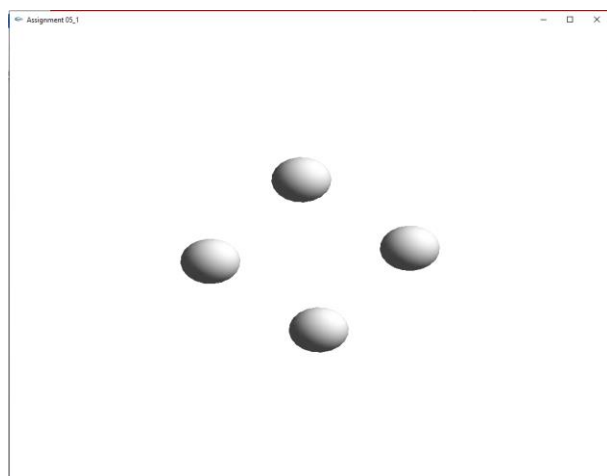
เมื่อกดตัว G



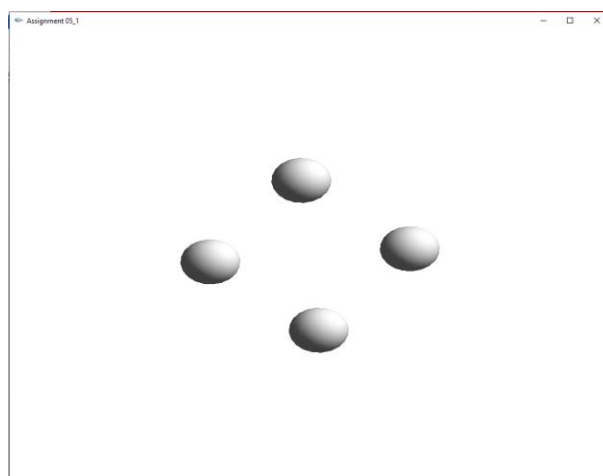
เมื่อกดตัว B



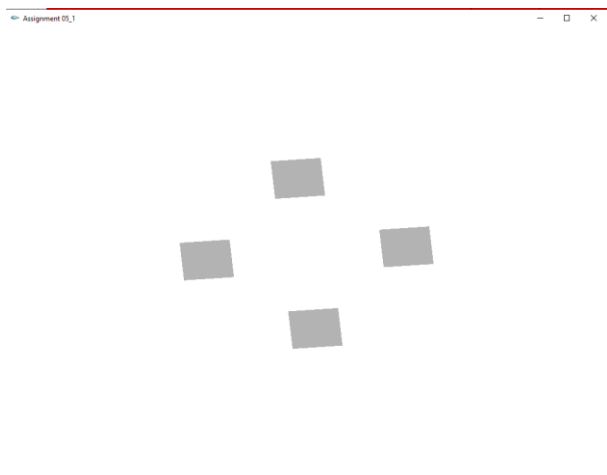
เมื่อกดตัว W



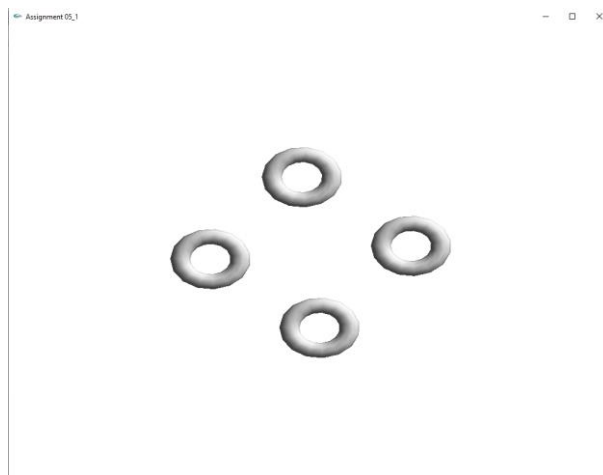
เมื่อกดตัว S



เมื่อกดตัว C



เมื่อกดตัว T



Source Code

```

#include <iostream>
using namespace std;

#include<GL\glut.h>

void init(void);
void myDisplay(void);
void myMouse(GLint button,GLint action,GLint x,GLint y);
void myKeyboard(GLubyte key,GLint x,GLint y);
void defineLight();
void define3dView();
void drawShape(int type);
void drawSphere();
void drawTorus();
void drawCube();
void Rotate(int value);

#define ESCAPE 27

int Type = 1 ;
GLfloat light_color[] = {1.0,1.0,1.0};
GLfloat light_position[] = {30.0,30.0,30.0,0.0};
int angle = 0,vX = 0,vY = 0,vZ = 1;
bool Play = false;

void main(int argc,char** argv){
    //Function Main ใช้เรียก การสร้างหน้าต่าง GUI และเรียกใช้ Function สำหรับ glut ในการ
    ทำงาน
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowPosition(50,50);
    glutInitWindowSize(1024,768);
    glutCreateWindow("Assignment 05_1 ");
    init();
    glutDisplayFunc(myDisplay);
    glutKeyboardFunc(myKeyboard);
    glutMouseFunc(myMouse);
    glutPostRedisplay();
    glutMainLoop();
}

void init(void){
    //Function สำหรับเชื่อมต่อ display และ หน้าต่าง GUI

```

```

    glClearColor(1.0,1.0,1.0,0);
    define3dView();
    defineLight();
}

void define3dView(){
    //Function จัดมุมมอง 3D ในการวาดภาพ
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-30,30,-30,30,-30,30);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void defineLight(){
    //Function สำหรับการจัดแสง
    glShadeModel(GL_SMOOTH);
    glLightfv(GL_LIGHT0,GL_POSITION,light_position);
    glLightfv(GL_LIGHT0,GL_AMBIENT,light_color);
    glLightfv(GL_LIGHT0,GL_DIFFUSE,light_color);
    glLightfv(GL_LIGHT0,GL_SPECULAR,light_color);

    glEnable(GL_LIGHT0);
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_LIGHTING);
}

void myDisplay(void){
    //Function ที่ใช้สำหรับวาดขึ้น GUI
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);

    drawShape(Type);

    glutSwapBuffers();
    glFlush();
}

void myKeyboard(GLubyte key,GLint x,GLint y){
    //Function รับค่าจาก keyboard เพื่อทำการ เช็ทสีของแสงและเช็ท Type ของรูปทรง
    key = toupper(key);
    if(key == 'R'){

        light_color[0] = 1.0;
        light_color[1] = 0.0;
        light_color[2] = 0.0;
    }
}

```



```

        glLightfv(GL_LIGHT0, GL_DIFFUSE, light_color);
        glLightfv(GL_LIGHT0, GL_AMBIENT, light_color);
    }
    else if(key == 'G'){

        light_color[0] = 0.0;
        light_color[1] = 1.0;
        light_color[2] = 0.0;

        glLightfv(GL_LIGHT0, GL_DIFFUSE, light_color);
        glLightfv(GL_LIGHT0, GL_AMBIENT, light_color);
    }
    else if(key == 'B'){

        light_color[0] = 0.0;
        light_color[1] = 0.0;
        light_color[2] = 1.0;

        glLightfv(GL_LIGHT0, GL_DIFFUSE, light_color);
        glLightfv(GL_LIGHT0, GL_AMBIENT, light_color);
    }
    else if(key == 'W'){

        light_color[0] = 1.0;
        light_color[1] = 1.0;
        light_color[2] = 1.0;

        glLightfv(GL_LIGHT0, GL_DIFFUSE, light_color);
        glLightfv(GL_LIGHT0, GL_AMBIENT, light_color);
    }
    else if(key == ESCAPE){

        exit(0);
    }
    else if(key == 'S'){

        Type = 1;
    }
    else if(key == 'C'){

        Type = 2;
    }
    else if(key == 'T'){

        Type = 3;
    }

```

```

    }

    glutPostRedisplay();
}

void myMouse(GLint button, GLint action, GLint x, GLint y){
    //Function รับค่าจาก Mouse เมื่อคลิกซ้ายให้ทำการหมุน และ เมื่อคลิกขวาให้ทำการหยุด
    if (button == GLUT_LEFT_BUTTON && action == GLUT_DOWN){
        if(!Play){

            Play = true;
            glutTimerFunc(30, Rotate, 30);
        }
    }
    else if (button == GLUT_RIGHT_BUTTON && action == GLUT_DOWN){

        Play = false;
    }
    glutPostRedisplay();
}

void Rotate(int value){
    //Function ใช้สำหรับการหมุนวัตถุโดยหมุน 360 องศา
    angle +=1;
    if(angle == 360){
        angle = 0;
    }
    glutPostRedisplay();
    if(Play) glutTimerFunc(value, Rotate, value);
}

void drawTorus(){
    //Function สำหรับวาดรูป โดนัท
    glPushMatrix();
    glRotatef(angle, vX, vY, vZ);
    glPushMatrix();
        glTranslatef(0.0f, 10.0f, 0.0f);
        glutSolidTorus( 1, 3, 10, 15);
    glPopMatrix();
    glPushMatrix();
        glTranslatef(-10.0f, 0.0f, 0.0f);
        glutSolidTorus( 1, 3, 10, 15);
    glPopMatrix();
    glPushMatrix();

```

```

        glTranslatef(10.0f,0.0f,0.0f);
        glutSolidTorus( 1, 3, 10, 15);
    glPopMatrix();
    glPushMatrix();
        glTranslatef(0.0f,-10.0f,0.0f);
        glutSolidTorus( 1, 3, 10, 15);
    glPopMatrix();
glPopMatrix();
}

```

```

void drawCube(){
    //Function สำหรับวาดรูป ลูกบาศก์
    glPushMatrix();
        glRotatef(angle,vX,vY,vZ);
        glPushMatrix();
            glTranslatef(0.0f,10.f,0.0f);
            glutSolidCube( 5);
        glPopMatrix();
        glPushMatrix();
            glTranslatef(-10.0f,0.0f,0.0f);
            glutSolidCube( 5);
        glPopMatrix();
        glPushMatrix();
            glTranslatef(10.0f,0.0f,0.0f);
            glutSolidCube( 5);
        glPopMatrix();
        glPushMatrix();
            glTranslatef(0.0f, -10.0f,0.0f);
            glutSolidCube( 5);
        glPopMatrix();
    glPopMatrix();
}

```

```

void drawSphere(){
    //Function สำหรับวาดรูป วงกลม
    glPushMatrix();
        glRotatef(angle,vX,vY,vZ);
        glPushMatrix();
            glTranslatef(0.0f,10.0f,0.0f);
            glutSolidSphere(3,20,20);
        glPopMatrix();
        glPushMatrix();
            glTranslatef(-10.0f,0.0f,0.0f);
            glutSolidSphere(3,20,20);
        glPopMatrix();
    glPopMatrix();
}

```

```

        glPushMatrix();
            glTranslatef(10.0f,0.0f,0.0f);
            glutSolidSphere(3,20,20);
        glPopMatrix();
        glPushMatrix();
            glTranslatef(0.0f,-10.0f,0.0f);
            glutSolidSphere(3,20,20);
        glPopMatrix();
    glPopMatrix();
}
void drawShape(int Type){
    //Function สำหรับเช็คค่า ในการวาดวัตถุ
    switch(Type){
        case 1:drawSphere();
            break;
        case 2 :drawCube();
            break;
        case 3 :drawTorus();
            break;
    }
}

```

แนวความคิดการออกแบบ

a. สร้างเมนูหลักที่ประกอบ

- i. เมนูย่อย Animation มีรายการ 2 รายการคือ “Play Animation”, “Stop Animation” เพื่อให้วัตถุเคลื่อนไหวหรือหยุดเคลื่อนไหว ตามรายการเมนูที่เลือก
- ii. เมนูย่อย Camera View มีรายการ 3 รายการคือ “Front View”, “Right View”, “Perspective” เพื่อกำหนดมุมมองของตำแหน่งกล้องตามรายการเมนูที่เลือก
- iii. เมนู Exit Program เพื่อออกจากโปรแกรม

b. การท างานกับเมาส์ ดังนี้

- i. สามารถใช้เมาส์ขวา ส าหรับแสดงเมนู

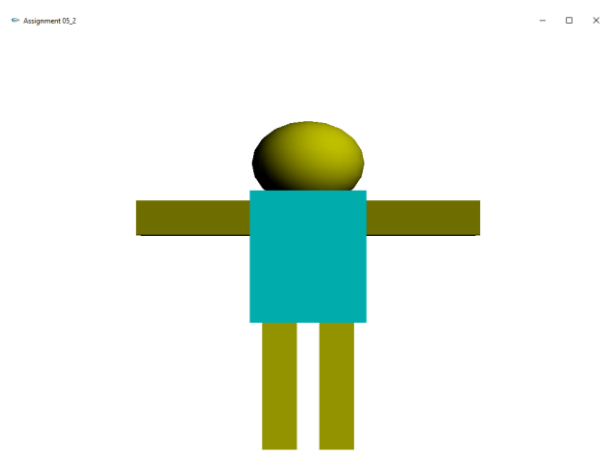
c. การท างานกับปุ่มคีย์บอร์ด ดังนี้

- i. กดปุ่ม Esc – ออกจากโปรแกรม
- ii. กดปุ่ม K – กำหนดพื้นหลังเป็นสีดำ
- iii. กดปุ่ม W – กำหนดพื้นหลังเป็นสีขาว

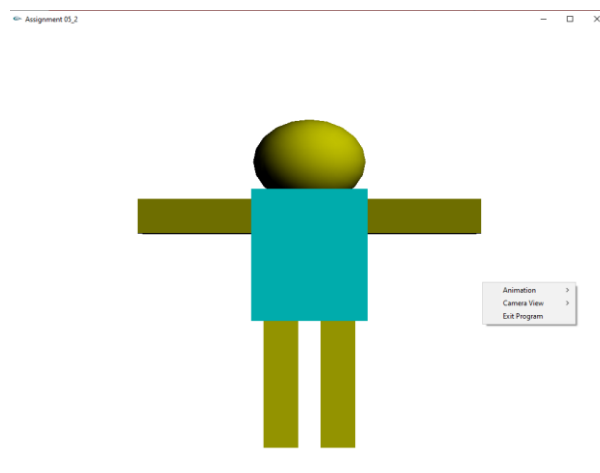
กำหนดตำแหน่งของแสงไว้ที่ $x = 40$, $y = 40$, $z = 40$ เป็นแสงสีขาวเท่านั้น

ภาพตัวอย่างโปรแกรม

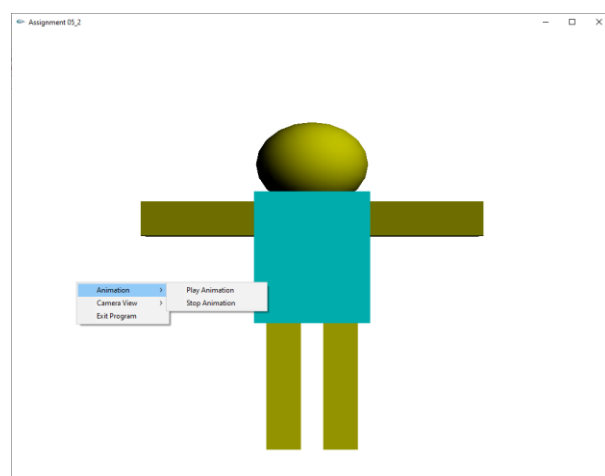
เมื่อเปิดโปรแกรมขึ้นมา



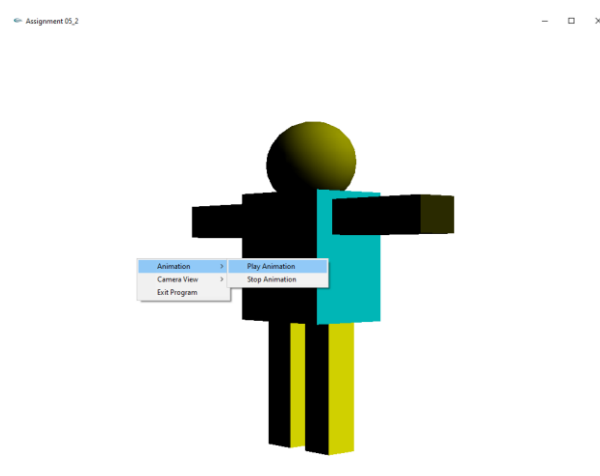
เมื่อทำการคลิกซ้าย



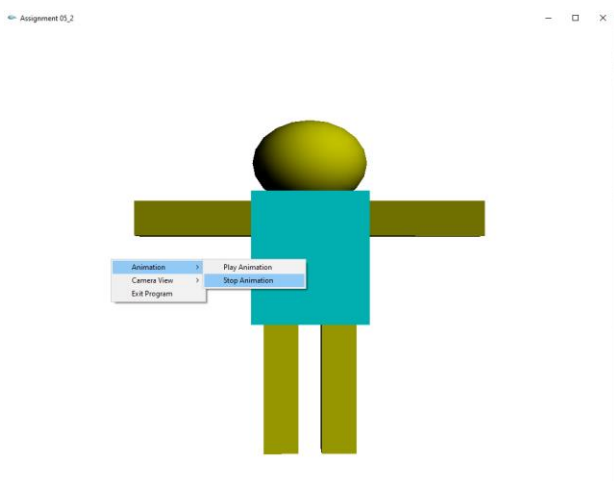
เมื่อ mouse ชี้ที่เมนู Animation



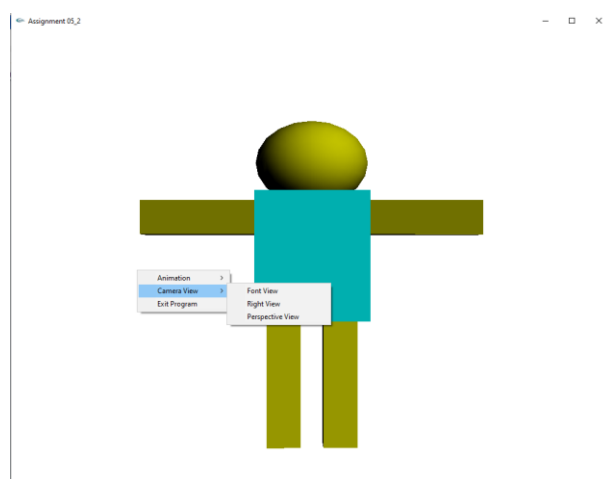
เมื่อคลิกเมนู Play Animation



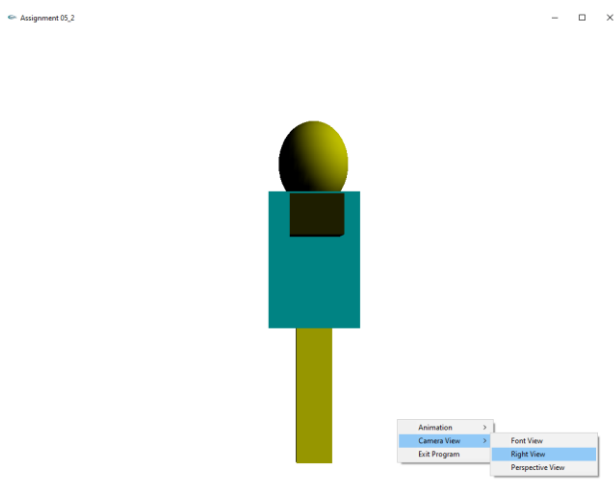
เมื่อคลิกที่เมนู Stop Animation



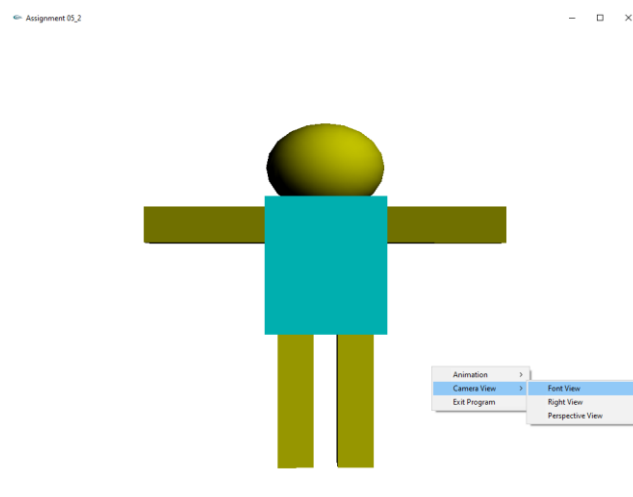
เมื่อ mouse ชี้ที่เมนู camera



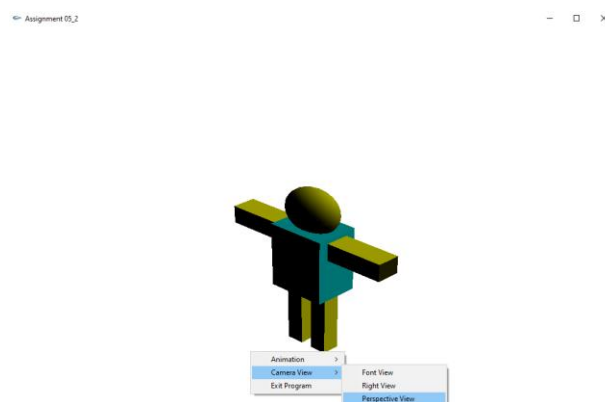
เมื่อคลิกที่เมนู Right view



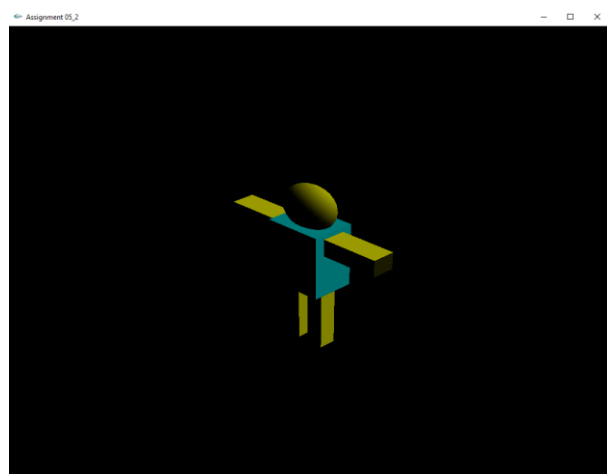
เมื่อคลิกเมนู Front view



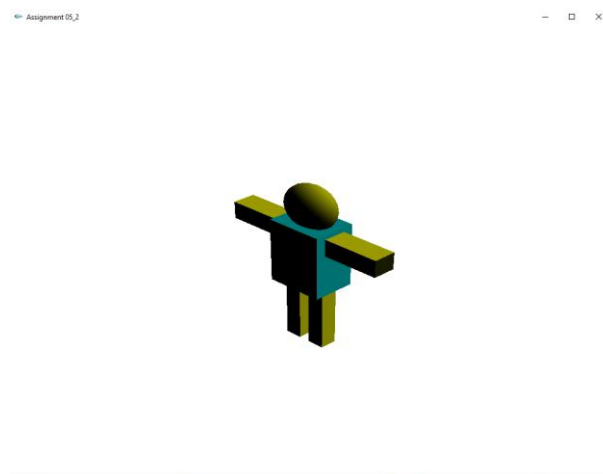
เมื่อคลิกที่เมนู Perspective view



เมื่อทำการกดตัว K



เมื่อทำการกดตัว W



SourceCode

```

#include<iostream>
using namespace std;

#include<GL\glut.h>

void init(void);
void myDisplay(void);
void drawShape();
void myKeyboard(GLubyte key,GLint x,GLint y);
void defineLight();
void definePerspective();
void defineRightPerspective();
void defineFontPerspective();
void Rotate(int Value);
void anif(int option);
void camf(int option);
void mainMenu(int option);

#define ESCAPE 27

GLfloat light_color[] = {1.0,1.0,1.0};
GLfloat light_position[] = {40.0,40.0,40.0,0.0};
int angle = 0,vX = 0,vY = 0,vZ = 1;
bool Play = false;

void main(int argc,char** argv){
    //Function Main ใช้เรียก การสร้างหน้าต่าง GUI และเรียกใช้ Function สำหรับ glut ในการ
    ทำงาน
    GLint ani, cam;
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowPosition(50,50);
    glutInitWindowSize(1024,768);
    glutCreateWindow("Assignment 05_2 ");
    init();
    glutDisplayFunc(myDisplay);
    glutKeyboardFunc(myKeyboard);
    ani = glutCreateMenu (anif);
    glutAddMenuEntry ("Play Animation", 1);
    glutAddMenuEntry ("Stop Animation", 2);

    cam = glutCreateMenu(camf);
    glutAddMenuEntry ("Font View", 1);

```

```

glutAddMenuEntry ("Right View", 2);
glutAddMenuEntry ("Perspective View", 3);

glutCreateMenu (mainMenu);
glutAddSubMenu ("Animation", ani);
glutAddSubMenu ("Camera View", cam);
glutAddMenuEntry ("Exit Program", 3);

glutAttachMenu (GLUT_RIGHT_BUTTON);

glutPostRedisplay();
glutMainLoop();
}

void init(void){
    //Function สำหรับเชื่อมต่อ display และ หน้าต่าง GUI
    glClearColor(1.0,1.0,1.0,0);
    defineFontPerspective();
    defineLight();
}

void defineLight(){
    //Function สำหรับการจัดแสง
    glShadeModel(GL_SMOOTH);
    glLightfv(GL_LIGHT0,GL_POSITION,light_position);
    glLightfv(GL_LIGHT0,GL_AMBIENT,light_color);
    glLightfv(GL_LIGHT0,GL_DIFFUSE,light_color);
    glLightfv(GL_LIGHT0,GL_SPECULAR,light_color);

    glEnable(GL_LIGHT0);
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_LIGHTING);
}

void myDisplay(void){
    //Function สำหรับการวาดบน GUI
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);

    drawShape();

    glutSwapBuffers();
    glFlush();
}

void drawShape(){
    //Function สำหรับการ วาดวัตถุลงบน หน้าต่างของ GUI

```

```

GLfloat no_mat[]={0.0,0.0,0.0,1.0};

GLfloat G_ambient[]={0,0,0,1.0};
GLfloat G_diffuse[]={0,0.7,0.7};
GLfloat G_specular[]={0,0,0};

GLfloat Y_ambient[]={0,0,0,1.0};
GLfloat Y_diffuse[]={0.6,0.6,0};
GLfloat Y_specular[]={0,0,0};

GLfloat shininess[]={12};

glPushMatrix();
    glRotatef(angle,0,1,0);

    glPushMatrix();
        glScalef( 0.8, 1.2, 0.6);
        glMaterialfv(GL_FRONT,GL_AMBIENT,G_ambient);
        glMaterialfv(GL_FRONT,GL_DIFFUSE,G_diffuse);
        glMaterialfv(GL_FRONT,GL_SPECULAR,G_specular);
        glMaterialfv(GL_FRONT,GL_SHININESS,shininess);
        glMaterialfv(GL_FRONT,GL_EMISSION,no_mat);
        glutSolidCube(2.5);
    glPopMatrix();

    glPushMatrix();
        glTranslatef(0,2.2,0.0);
        glScalef(1,1,0.6);
        glMaterialfv(GL_FRONT,GL_AMBIENT,Y_ambient);
        glMaterialfv(GL_FRONT,GL_DIFFUSE,Y_diffuse);
        glMaterialfv(GL_FRONT,GL_SPECULAR,Y_specular);
        glMaterialfv(GL_FRONT,GL_SHININESS,shininess);
        glMaterialfv(GL_FRONT,GL_EMISSION,no_mat);
        glutSolidSphere(1.0,20,20);
    glPopMatrix();

    glPushMatrix();
        glTranslatef(-1.5,0.9,0.0);
        glScalef(3,0.8,0.8);
        glMaterialfv(GL_FRONT,GL_AMBIENT,Y_ambient);
        glMaterialfv(GL_FRONT,GL_DIFFUSE,Y_diffuse);
        glMaterialfv(GL_FRONT,GL_SPECULAR,Y_specular);
        glMaterialfv(GL_FRONT,GL_SHININESS,shininess);
        glMaterialfv(GL_FRONT,GL_EMISSION,no_mat);
        glutSolidCube(1);
    glPopMatrix();

```

```

        glPushMatrix();
            glTranslatef(1.5,0.9,0.0);
            glScalef(3,0.8,0.8);
            glMaterialfv(GL_FRONT,GL_AMBIENT,Y_ambient);
            glMaterialfv(GL_FRONT,GL_DIFFUSE,Y_diffuse);
            glMaterialfv(GL_FRONT,GL_SPECULAR,Y_specular);
            glMaterialfv(GL_FRONT,GL_SHININESS,shininess);
            glMaterialfv(GL_FRONT,GL_EMISSION,no_mat);;
            glutSolidCube(1);
        glPopMatrix();

        glPushMatrix();
            glTranslatef(-0.5,-2,0.0);
            glScalef(0.6,5,0.6);
            glMaterialfv(GL_FRONT,GL_AMBIENT,Y_ambient);
            glMaterialfv(GL_FRONT,GL_DIFFUSE,Y_diffuse);
            glMaterialfv(GL_FRONT,GL_SPECULAR,Y_specular);
            glMaterialfv(GL_FRONT,GL_SHININESS,shininess);
            glMaterialfv(GL_FRONT,GL_EMISSION,no_mat);
            glutSolidCube(1);
        glPopMatrix();

        glPushMatrix();
            glTranslatef(0.5,-2,0.0);
            glScalef(0.6,5,0.6);
            glMaterialfv(GL_FRONT,GL_AMBIENT,Y_ambient);
            glMaterialfv(GL_FRONT,GL_DIFFUSE,Y_diffuse);
            glMaterialfv(GL_FRONT,GL_SPECULAR,Y_specular);
            glMaterialfv(GL_FRONT,GL_SHININESS,shininess);
            glMaterialfv(GL_FRONT,GL_EMISSION,no_mat);
            glutSolidCube(1);
        glPopMatrix();
    glPopMatrix();
}

void Rotate(int value){
    //Function สำหรับทำการหมุนรูปวัตถุที่ 360 องศา
    angle +=1;
    if(angle == 360){
        angle = 0;
    }
    glutPostRedisplay();
    if(Play) glutTimerFunc(value,Rotate,value);
}

```

```

}
void myKeyboard(GLubyte key, GLint x, GLint y){
    //Function การรับค่าจาก Keyboard เพื่อทำการเปลี่ยนสี พื้นหลัง
    key = toupper(key);
    if(key == 'K') glClearColor(0.0,0.0,0.0,0.0);
    else if(key == 'W') glClearColor(1.0,1.0,1.0,0);
    else if(key == ESCAPE) exit(0);
    glutPostRedisplay();
}

void definePerspective(){
    //Function การจัดมุมมองแบบ perspective
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(30,1,1,1000);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(20,20,20,0,0,0,0,1,0);
}

void defineRightPerspective(){
    //Function สำหรับการจัดมุมมองทางขวา
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(30,1,1,1000);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(20,00,00,0,0,0,0,1,0);
}

void defineFontPerspective(){
    //Function สำหรับการจัดมุมมองข้างหน้า
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(30,1,1,1000);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(00,00,20,0,0,0,0,1,0);
}

void anif(int option){
    //Function สำหรับการรับค่าจาก pop-up menu ของการหมุน
    switch(option){
        case 1: Play = true; glutTimerFunc(30, Rotate, 30); break;
        case 2: Play = false; angle = 0; break;
    }
}

```

```

    }
    glutPostRedisplay ( );
}

void camf(int option){
    //Function สำหรับการรับค่าจาก pop-up menu ของมุมมอง
    switch(option){
        case 1: defineFontPerspective(); break;
        case 2: defineRightPerspective(); break;
        case 3: definePerspective(); break;
    }
    glutPostRedisplay ( );
}

void mainMenu(int option){
    //Function สำหรับการรับค่าจาก pop-up menu หลัก
    switch (option){
        case 3: exit(0); break;
    }
    glutPostRedisplay ( );
}

```