**Session 9: Two-dimensional Object Animation (Spinning & Sound)**

**Intended Learning Outcome:**

1. Students will be able to adopt the movement control of object in the display.
2. Students will be able to better understand about axis (X, Y) wise movement and controlling.

**Expected Skills:**

1. Better Understanding about axis (X,Y)
2. Clear idea about polygon drawing with any variable

**Tools Required:**

1. CodeBlocks
2. OpenGL and GLUT using CodeBlocks.

**Session Detail:**

***Moving 2D object & Spinning with sound by using key Functionalities:***

******

/\*

 \* GLUT Shapes Demo

 \*

 \* Written by Nigel Stewart November 2003

 \*

 \* This program is test harness for the sphere, cone

 \* and torus shapes in GLUT.

 \*

 \* Spinning wireframe and smooth shaded shapes are

 \* displayed until the ESC or q key is pressed. The

 \* number of geometry stacks and slices can be adjusted

 \* using the + and - keys.

 \*/

#include<windows.h>

#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

#include <stdlib.h>

#include <unistd.h>

#include <math.h>

//#include<stdarg.h>

static GLfloat spin = 0.0;

static float tx = 0.0;

static float ty = 0.0;

void display(void)

{

 glClear(GL\_COLOR\_BUFFER\_BIT);

 glPushMatrix();

 glRotatef(spin, 0.0, 0.0, 1.0);

 glColor3f(1.0, 1.0, 1.0);

 glTranslatef(tx,ty,0);

 glRectf(-25.0, -25.0, 25.0, 25.0);

 glPopMatrix();

 glFlush();

}

void spinDisplay\_left(void)

{

 spin = spin + 10;

 glutPostRedisplay();

}

void spinDisplay\_right(void)

{

 spin = spin - 10;

 glutPostRedisplay();

}

void init(void)

{

 glClearColor (1.0, 0.0, 0.0, 0.0);

 glOrtho(-100.0, 100.0, -100.0,100.0, -1.0, 1.0);

}

void my\_keyboard(unsigned char key,int x, int y)//In main(), To register a callback to the keyboard function,

{

 switch (key) {

 case 'l':

 spinDisplay\_left();

 break;

 case 'r':

 spinDisplay\_right();

 break;

 case 's':

 glutIdleFunc(NULL); //idle callback is continuously called when events are not being received

 break;

 default:

 break;

 }

}

void spe\_key(int key, int x, int y) // glutSpecialFunc sets the special keyboard callback for the current window

{

 switch (key) {

 case GLUT\_KEY\_UP:

 ty +=5;

 PlaySound("Horn Honk-SoundBible.com-1162546405.wav", NULL, SND\_ASYNC|SND\_FILENAME);

 glutPostRedisplay();

 break;

 case GLUT\_KEY\_DOWN:

 PlaySound("Horn Honk-SoundBible.com-1162546405.wav", NULL, SND\_ASYNC|SND\_FILENAME);

 ty -=5;

 glutPostRedisplay();

 break;

 case GLUT\_KEY\_RIGHT:

 PlaySound("Horn Honk-SoundBible.com-1162546405.wav", NULL, SND\_ASYNC|SND\_FILENAME);

 tx +=5;

 glutPostRedisplay();

 break;

 case GLUT\_KEY\_LEFT:

 PlaySound("Horn Honk-SoundBible.com-1162546405.wav", NULL, SND\_ASYNC|SND\_FILENAME);

 tx -=5;

 glutPostRedisplay();

 break;

 default:

 break;

 }

}

void my\_mouse(int button, int state, int x, int y)

{

 switch (button) {

 case GLUT\_LEFT\_BUTTON:

 if (state == GLUT\_DOWN)

 glutIdleFunc(spinDisplay\_left);

 break;

 case GLUT\_MIDDLE\_BUTTON:

 case GLUT\_RIGHT\_BUTTON:

 if (state == GLUT\_DOWN)

 glutIdleFunc(spinDisplay\_right);

 break;

 default:

 break;

 }

}

int main()

{

 //PlaySound("Horn Honk-SoundBible.com-1162546405.wav", NULL, SND\_ASYNC|SND\_FILENAME);

 glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

 glutInitWindowSize (500, 500);

 glutInitWindowPosition (100, 100);

 glutCreateWindow ("LAB3");

 init();

 glutDisplayFunc(display);

 glutKeyboardFunc(my\_keyboard);

 glutSpecialFunc(spe\_key);

 glutMouseFunc(my\_mouse);

 glutMainLoop();

 return 0;

}