**Session 2: Perform to design output primitives using OpenGL**

**Intended Learning Outcome:**

1. Able to demonstrate effective OpenGL programs to solve graphics programming issues including different shapes.
2. Able to appreciate the knowledge along axis (X,Y).

**Expected Skills:**

b. Make some design using OpenGL.

**Tools Required:**

1. CodeBlocks
2. OpenGL and GLUT using CodeBlocks.

**Session Detail:**

**Code for designing 8X8 cheese Board:**

#include <GL/gl.h>

#include <GL/glut.h>

void display(void)

{

 /\* clear all pixels \*/

 glClear (GL\_COLOR\_BUFFER\_BIT);

 /\* draw white polygon (rectangle) with corners at

 \* (0.25, 0.25, 0.0) and (0.75, 0.75, 0.0) \*/

 bool color\_change=false;

 for(int x=0; x<=800; x+=100)

 {

 for(int y=0; y<=800; y+=100)

 {

 if(color\_change)

 {

 glColor3f(1.0,0.75,0.0);

 color\_change=!color\_change;

 }

 else

 {

 glColor3f(0.0,0.0,0.0);

 color\_change=!color\_change;

 }

 glBegin(GL\_QUADS);

 glVertex2i(x,y);

 glVertex2i(x,y+100);

 glVertex2i(x+100,y+100);

 glVertex2i(x+100,y);

 glEnd();

 /\* don't wait!

 \* start processing buffered OpenGL routines \*/

 glFlush ();

 }

 }

}

void init (void)

{

 /\* select clearing (background) color \*/ glClearColor (0.0,0.0,0.0,0.0);

 /\* initialize viewing values \*/ glMatrixMode(GL\_PROJECTION);

 glLoadIdentity();

 //glOrtho(0.0, 800.0, 0.0, 800.0, -100.0, 100.0);

 gluOrtho2D(0.0,800.0,0.0,800.0);

}

/\*

\* Declare initial window size, position, and display mode

\* (single buffer and RGBA). Open window with "hello"

\* in its title bar. Call initialization routines.

\* Register callback function to display graphics.

\* Enter main loop and process events.

\*/

int main(int argc, char\*\* argv)

{

 glutInit(&argc, argv);

 glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

 glutInitWindowSize (800, 800);

 glutInitWindowPosition (0,0);

 glutCreateWindow ("Cheese Board");

 init ();

 glutDisplayFunc(display);

 glutMainLoop();

 return 0; /\* ISO C requires main to return int. \*/

}

**Sample output:**

****