Computer Graphics

Lab-2: Basic Primitives in OpenGL

```
#include <GL/glut.h>
void myInit(void) {
glClearColor(1.0, 1.0,1.0, 1.0); // white backgrd
glMatrixMode(GL_PROJECTION);
glLoadIdentity( );
gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
void display(void)
{
glClear(GL_COLOR_BUFFER_BIT);
GLint p1[] = {200, 100};
GLint p2[] = {50, 0};
GLint p3[] = {100, 200};
GLint p4[] = {150, 0};
GLint p5[] = {0, 100};
glBegin(GL_LINES);
glColor3f(1.0, 0, 0); //red
 glVertex2i(200, 350);
 glVertex2i(50, 250);
 glVertex2i(100, 450);
 glVertex2i(150, 250);
 glVertex2i(0, 350);
glEnd( );
glBegin(GL_LINE_STRIP);
glColor3f(0, 1.0, 0); //green
 glVertex2i(450, 250);
 glVertex2i(300, 150);
 glVertex2i(350, 350);
 glVertex2i(400, 150);
 glVertex2i(250, 250);
glEnd( );
glLineWidth(4.0);
glBegin(GL_LINE_LOOP);
glColor3f(0, 0, 1.0); //blue
 glVertex2iv(p1);
 glVertex2iv(p2);
 glVertex2iv(p3);
 glVertex2iv(p4);
 glVertex2iv(p5);
glEnd( );
glLineWidth(3.0);
glBegin(GL_LINE_LOOP);
glColor3f(0.0, 1.0, 0.0);
glVertex2i(300, 200);
glColor3f(1.0, 1.0, 0.0);
glVertex2i(300, 50);
glColor3f(1.0, 0.0, 0.0);
glVertex2i(500, 50);
glEnd( );
//Circle Drawing
GLint x=0,y,p,r = 100,xc=500,yc=350;
y = r;
p = 3-2*r;
glBegin(GL_POINTS);
glColor3f(1.0,0,1.0);
while(x<=y)
 {  if(p<0)
    p = p+(4*x+6);
    else
    { y = y-1;
      p = p+4*(x-y)+10;
    }
glVertex2i(xc+x, yc+y);
glVertex2i(xc-x, yc+y);
glVertex2i(xc+x, yc-y);
glVertex2i(xc-x, yc-y);
glVertex2i(xc+y, yc+x);
glVertex2i(xc-y, yc+x);
glVertex2i(xc+y, yc-x);
glVertex2i(xc-y, yc-x);
x = x+1;
}
glEnd( );
glFlush( ); // send all output to the display
}
int main(int argc, char *argv[ ])
{   glutInit(&argc, argv);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(10, 10);
    glutCreateWindow("Basic primitives");
    glutDisplayFunc(display);
    myInit( );
    glutMainLoop( );
}
```

Lab-3: Primitives in OpenGL

```
#include <GL/glut.h>
#include<math.h>
#define PI 3.14159265358979323846264338327 95
  void myInit(void) {
  glClearColor(1.0, 1.0,1.0, 1.0); // white backgrd
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity( );
  gluOrtho2D(0.0, 640.0, 0.0, 480.0);
  }
  void display(void)
  {
  glClear(GL_COLOR_BUFFER_BIT);
  //Circle Drawing
  GLint x = 0, y, p, r = 100, xc = 150, yc = 150;
  y = r;
  p = 3 - 2*r;
  glBegin(GL_POINTS);
  glColor3f(0,1.0,0);
  while(x<=y)
  {  if(p<0)
  p = p+(4*x+6);
  else
  { y = y-1;
    p = p+4*(x-y)+10;
  }
  glVertex2i(xc+x, yc+y);
  glVertex2i(xc-x, yc+y);
  glVertex2i(xc+x, yc-y);
  glVertex2i(xc-x, yc-y);
  glVertex2i(xc+y, yc+x);
  glVertex2i(xc-y, yc+x);
  glVertex2i(xc+y, yc-x);
  glVertex2i(xc-y, yc-x);
  x = x+1;
  }
  glEnd( );
  //Ellipse drawing
  GLint i, r1 = 75, r2 = 150;
  xc = 500, yc = 250;
  GLfloat t, d;
  glBegin(GL_POINTS);
  glColor3f(1.0,0,1.0);
  for(i = 0; i<360; i++)
  {
  t = PI/180;
  d = i*t;
  x = xc+ceil(r1*cos(d));
  y = yc+ceil(r2*sin(d));
  glVertex2i(x,y);
  }
  glEnd( );
  glFlush( );
  }
  int main(int argc, char *argv[ ])
  {   glutInit(&argc, argv);
     glutInitWindowSize(640, 480);
     glutInitWindowPosition(10, 10);
     glutCreateWindow("Primitives in OpenGL");
     glutDisplayFunc(display);
     myInit( );
     glutMainLoop( );
  }
```

OpenGL fill area routines

```
#include<Gl/glut.h>
void myInit(void) {
   glClearColor(1.0,1.0,1.0,1.0);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
void display(void)
{
glClear(GL_COLOR_BUFFER_BIT);
glColor3f(0,0,1.0);
glRecti(50, 450, 150, 275);
glBegin(GL_POLYGON);
  glColor3f(1.0,0,1.0);
  glVertex2i(200,150);
  glVertex2i(150,50);
  glVertex2i(50,50);
  glVertex2i(0,150);
  glVertex2i(100,250);
 glEnd( );
 glBegin(GL_TRIANGLES);
  glColor3f(0,1.0,1.0);
  glVertex2i(200,350);
  glVertex2i(300,450);
  glVertex2i(400,350);
  glVertex2i(250,300);
  glVertex2i(350,300);
  glVertex2i(300,200);
 glEnd( );
 glBegin(GL_TRIANGLE_STRIP);
 glColor3f(1.0,1.0,0);
   glVertex2i(600,350);
   glVertex2i(500,450);
   glVertex2i(400,350);
   glVertex2i(550,250);
   glVertex2i(500,150);
   glVertex2i(350,250);
 glEnd( );
 glBegin(GL_TRIANGLE_FAN);
 glColor3f(0,1.0,0);
   glVertex2i(450,125);
   glVertex2i(350,225);
   glVertex2i(250,125);
   glVertex2i(400,25);
   glVertex2i(300,25);
   glVertex2i(200,75);
glEnd( );
glClear(GL_COLOR_BUFFER_BIT); // Clear Screen
glBegin(GL_QUADS);
glColor3f(0.0,1.0,1.0);
  glVertex2i(350,400);
  glVertex2i(250,450);
  glVertex2i(150,400);
  glVertex2i(300,300);
  glVertex2i(200,250);
  glVertex2i(150,100);
  glVertex2i(50,150);
  glVertex2i(100,300);
  glEnd( );
glBegin(GL_QUAD_STRIP);
glColor3f(1.0,1.0,0.0);
  glVertex2i(450,250);
  glVertex2i(350,350);
  glVertex2i(250,250);
  glVertex2i(400,150);
  glVertex2i(300,150);
  glVertex2i(200,200);
  glEnd( );
//Character primitives in openGL
glColor3f(1.0,0,1.0);
glRasterPos2i(400, 50);
glutBitmapCharacter(GLUT_BITMAP_9_BY_15, 'a');
glRasterPos2i(400, 30);
glutBitmapCharacter(GLUT_BITMAP_HELVETICA_10,'a');
glRasterPos2i(300, 100);
glutStrokeCharacter(GLUT_STROKE_MONO_ROMAN, 'a');
glFlush( );
}
int main(int argc, char** argv)
{   glutInit(&argc, argv);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(50, 50);
    glutCreateWindow("OpenGL fill area Primitives");
    glutDisplayFunc(display);
    myInit( );
    glutMainLoop( );
    return 0;
}
```