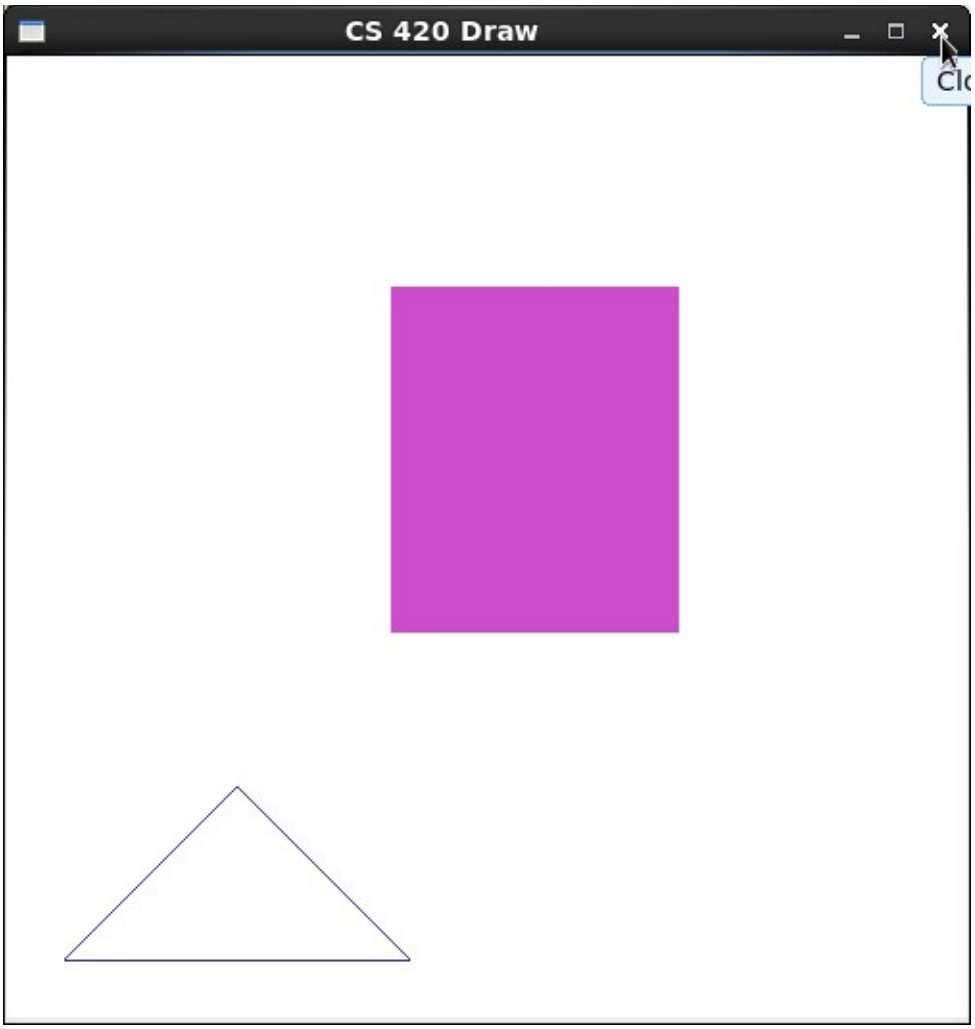
CSE420
Samuel Marrujo
Professor Yu
Lab 01

Draw01

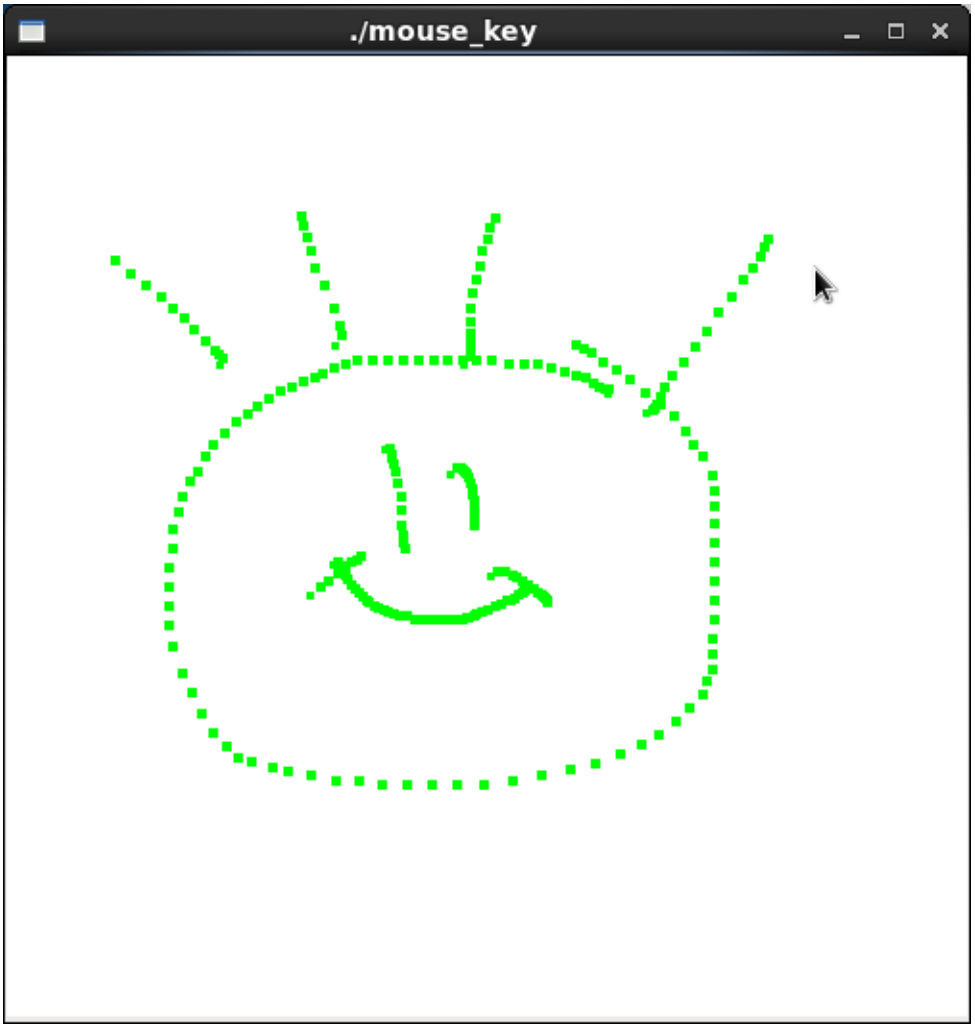
In this part of the lab, it was a simple drawing of a rectangle and a triangle. Using the rectangle function and polylines, I was able to accomplish this task successfully. Here are my results for this program:



```
//draw.cpp : demo program for drawing 3 dots, two lines, ploylines, rectangles
#include <GL/glut.h>

//initialization
void init( void )
{
  glClearColor( 1.0, 1.0, 1.0, 0.0 );    //get white background color
  glColor3f( 0.0f, 1.0f, 0.0f ); //set drawing color
  glPointSize( 8.0 );                    //a dot is 4x4
  glMatrixMode( GL_PROJECTION );
  glLoadIdentity();                      //replace current matrix with identity matrix
  gluOrtho2D( 0.0, 500.0, 0.0, 500.0 );
}

void display( void )
{
  glClear( GL_COLOR_BUFFER_BIT );    //clear screen


  glColor3f ( 0.2, 0.2, 0.6 );
  glBegin( GL_LINE_STRIP );
    glVertex2i( 30, 30 );
    glVertex2i( 120, 120 );
    glVertex2i( 210, 30 );
    glVertex2i( 30, 30 );
  glEnd();

  glColor3f( 0.8, 0.3, 0.8 );            //bright grey
  glRecti( 200, 200, 350, 380 );
  glFlush();                             //send all output to screen
}
```

Mouse_key

Lastly, in the mouse_key program, we are to modify the program to how we want it. So, I decided to change the brush size and the color, then decided to draw a happy face afterward! Here are my results for this program:



```
//mouse_key.cpp
#include <GL/glut.h>
#include <stdlib.h>

#define   screenHeight  500

//initialization
void init( void )
{
  glClearColor( 1.0, 1.0, 1.0, 0.0 );    //get white background color
  glColor3f( 0.0f, 0.0f, 0.0f ); //set drawing color
  glPointSize( 4.0 );                    //a dot is 4x4
  glMatrixMode( GL_PROJECTION );
  glLoadIdentity();
  gluOrtho2D( 0.0, 500.0, 0.0, 500.0 );
} //init

void display()
{
  glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
  glFlush();
}

void drawDot( int x, int y )
{
  glBegin( GL_POINTS );
    glVertex2i( x, y );             //draw a points
  glEnd();
} //drawDot

void myMouse( int button, int state, int x, int y )
{
  if ( button == GLUT_LEFT_BUTTON && state == GLUT_DOWN )
    drawDot( x, screenHeight - y );
  glFlush();                              //send all output to screen
}

void myMovedMouse(  int mouseX, int mouseY)
{
  GLint x = mouseX;
  GLint y = screenHeight - mouseY;
  GLint brushsize = 5;
  glColor3f( 0.0, 1.0, 0.0 );
  glRecti ( x, y, x + brushsize, y + brushsize );
  glFlush();
} //myMovedMouse
```

```
void myKeyboard ( unsigned char key, int mouseX, int mouseY )
{
  GLint x = mouseX;
  GLint y = screenHeight - mouseY;
  switch( key )
  {
    case 'p':
          drawDot ( x, y );
          glFlush();
          break;
    case 'r':
        glRecti ( x, y, x + 20, y + 30 );
          glFlush();
          break;
    case 'e':
          exit ( -1 );
    default :
          break;
  }
}
```