Comment and self-evaluation:
I think that this lab is to get us be familiar with openGL by starting the demo program and I think that I have established this purpose. I have finished all the requirements in modifying and updating the code. Everything is straightforward. The 4th part with adjusting the parameters for gluOrtho2D() required more effort than the other parts as I am still not so familiar with the all the openGL functions in the code.
Below are the screenshots of the running program.

Draw Program:

Draw program with two triangles and two rectangles, dot size was changed.
Mouse_Key Program:

Mouse_Key Program after modifying the dot and brush size
Mouse_Key program after modifying gluOrtho2D() with parameters ( -1, 1, 0, 2 )
Source Code:

draw.cpp:
//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, 0.0f, 0.0f ); //set drawing color
  glPointSize( 40.0 ); //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, 1, 0 );
  glBegin( GL_TRIANGLES );
    glVertex2i( 10, 10 );
    glVertex2i( 200, 10 );
    glVertex2i( 100, 200 );
  glEnd();

  //second triangle blue
  glColor3f( 0, 0, 1 );
  glBegin( GL_TRIANGLES );
    glVertex2i( 250, 250 );
    glVertex2i( 300, 300 );
    glVertex2i( 500, 200 );
  glEnd();

  glColor3f( 0.6, 0.6, 0.6 ); //bright grey
  glRecti( 400, 400, 480, 480 );
  glColor3f( 1.0, 0.0, 0.0 ); //red
  glRecti( 350, 350, 380, 390 );

  glFlush(); //send all output to screen
}
mouse_key.cpp:

#include <GL/glut.h>
#include <stdlib.h>

#define screenHeight 500

int w_l = 0;
int w_r = 1;
int w_b = 0;
int w_t = 1;

//initialization
void init( void )
{
    //glViewport(250, 500, 250, 500);
    glClearColor( 1.0, 1.0, 1.0, 0.0 ); //get white background
    glColor(0.0f, 0.0f, 0.0f); //set drawing color
    glPointSize( 4 ); //a dot is 4x4
    glMatrixMode( GL_PROJECTION );
    glLoadIdentity();
    //gluOrtho2D( -1, 1, -1, 1);
    gluOrtho2D( -1, 1, 0, 2 );
    glColor(0.6, 0.6, 0.6); //bright grey
    // glRecti( -0.5, 0.5, -0.5, 0.5);
    glRecti( 0, 50, 0, 50 );
    glFlush();
} //init

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

void drawDot( int x, int y )
{
    glBegin( GL_POINTS );
        glVertex2i( x/500, y/500 ); //draw a points
    glEnd();
} //drawDot

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

void myMovedMouse(int mouseX, int mouseY)
{
    GLfloat x = (float)mouseX/250-1;
    GLfloat y = (float)(screenHeight - mouseY)/250;
    GLfloat brushsize = 0.05;
    glColor3f( 0.0, 1.0, 0.0 );
    glRectf( x, y, x + brushsize, y + brushsize );
    glFlush();
}  //myMovedMouse

void myKeyboard ( unsigned char key, int mouseX, int mouseY )
{
    GLfloat x = (float)mouseX/250-1;
    GLfloat y = (float)(screenHeight - mouseY)/250;
    switch( key )
    {
        case 'p':
            drawDot ( x, y );
            break;
        case 'e':
            exit ( -1 );
            default :
                break;
    }
}