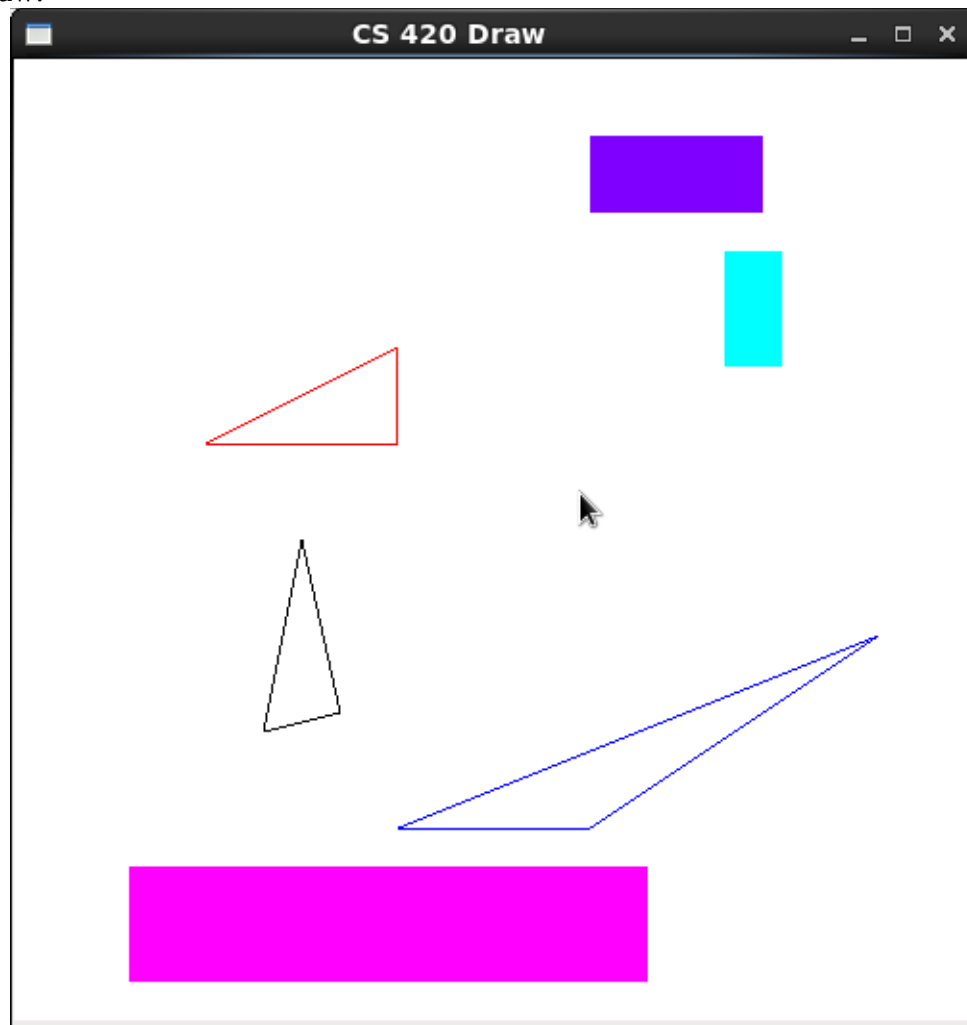


Yazhuo Liu

Output for draw:



Code for draw:

//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 ( 0.0, 1.0, 0.0 );
glBegin( GL_POINTS );           //draw points
    glVertex2i( 100, 50 );       //draw a point
    glVertex2i( 100, 150 );      //draw a point
    glVertex2i( 200, 200 );      //draw a point
glEnd();

glColor3f ( 1.0, 0.0, 0.0 );
glBegin( GL_LINES );           //draw lines
    glVertex2i( 20, 20 );        //horizontal line
    glVertex2i( 400, 20 );
    glVertex2i( 20, 10 );       //vertical line
    glVertex2i( 20, 400 );
glEnd();
*/
glColor3f ( 0.0, 0.0, 1.0 );
glBegin( GL_LINE_STRIP );      //draw polyline
    glVertex2i( 200, 100 );
    glVertex2i( 300, 100 );
    glVertex2i( 450, 200 );
    glVertex2i( 200, 100 );
glEnd();

glColor3f( 1.0, 0.0, 0.0 );
glBegin( GL_LINE_STRIP );
    glVertex2i( 100, 300 );
    glVertex2i( 200, 300 );
    glVertex2i( 200, 350 );
    glVertex2i( 100, 300 );
glEnd();

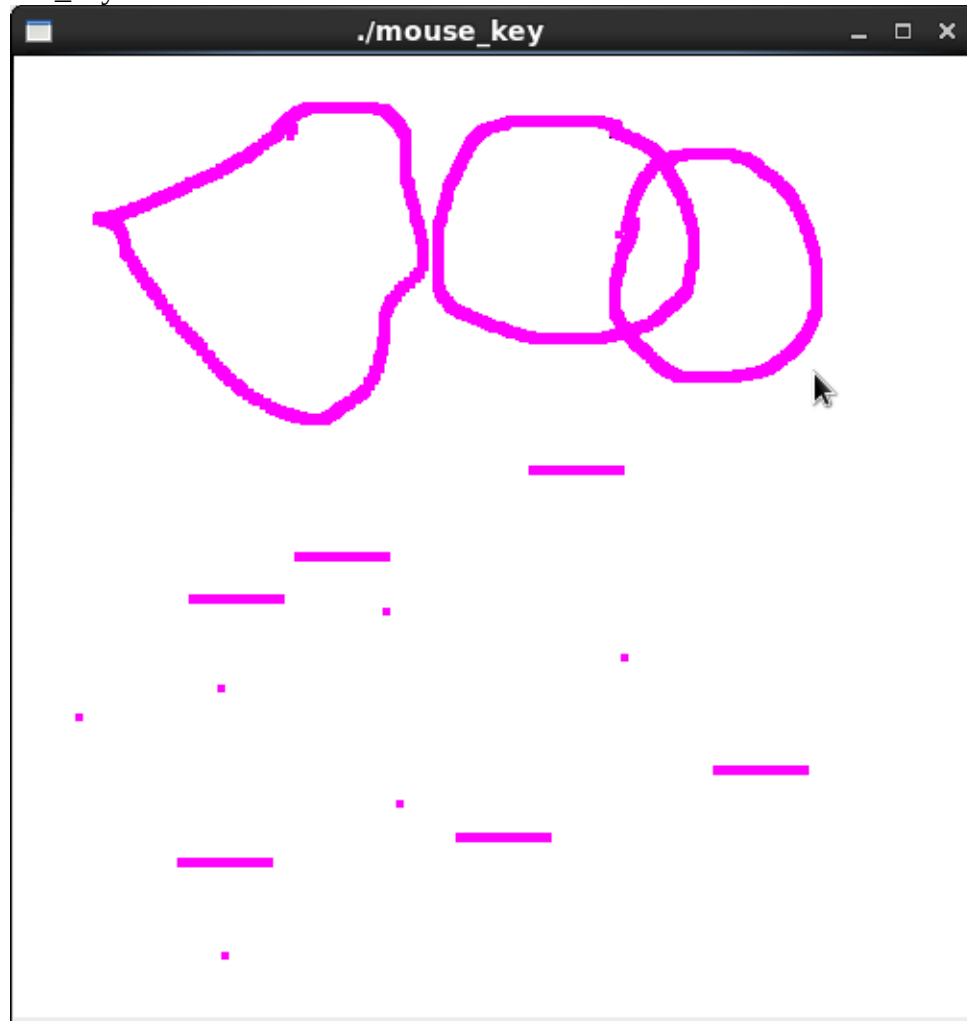
glColor3f( 0.0, 0.0, 0.0 );
glBegin( GL_LINE_STRIP );
    glVertex2i( 130, 150 );
    glVertex2i( 150, 250 );
    glVertex2i( 170, 160 );
    glVertex2i( 130, 150 );
glEnd();

//draw rectangles
glColor3f( 0.5, 0.0, 1.5 );    //bright grey
glRecti( 300, 420, 390, 460 );
glColor3f( 0.0, 1.0, 2.0 );    //red
glRecti( 370, 340, 400, 400 );
glColor3f( 1.5, 0.0, 2.0 );
glRecti( 60, 20, 330, 80 );

glFlush();                     //send all output to screen
}

```

Output for mouse_key:



Code for mouse_key:

```
//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 = 6;
    glColor3f( 1.0, 0.0, 2.0 );
    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 + 50, y + 5 );
            glFlush();
            break;
        case 'e':
            exit ( -1 );
        default :
            break;
    }
}
```

Report:

In the draw program, I deleted the horizontal line and vertical line, and the three dots. I added some more triangles and rectangles, and changed the color of them. In the mouse_key program, I changed the color of the dots, and I changed case “r” so that now when you hit “r”, it displays a short line instead of a rectangle.