CS 411: Lab 02

1751037 - Quang-Thuc Nguyen

May 8, 2020

1 Manager state of menu and drawing

Due to manager state of user selection and handle drawing event, I build two classes `MenuManager` and `DrawManager`.

`MenuManager` is a class that contains state as static variable. It stores current shape user wants to draw (i.e circle, ellipse, rectangle, polygon), current color (default is red), and current action (draw or fill).

`DrawManager` is used to handle drawing via mouse event.

2 Functionality

2.1 Context menu

The user can open context menu by right click. It will show some selections as Figure 1

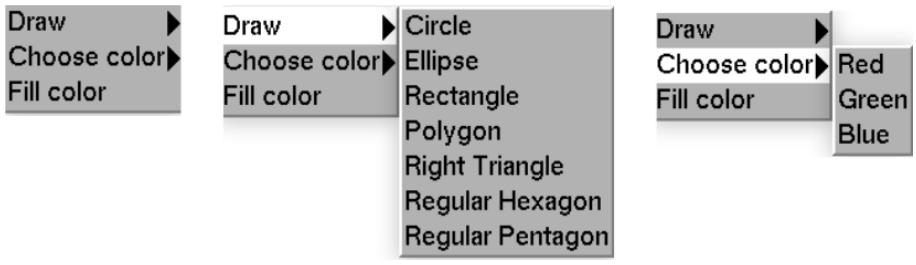

Figure 1: Context menu

2.2 Drawing with specific shape

Drawing on the frame can be done by using mouse only. With each mouse event, the `glutMouseFunc` will trigger `DrawManager` to store the information and call `glutPostRedisplay` for drawing.

2.2.1 Rectangle

For drawing rectangle, you have to select two opposite corner by following steps: hold down left button to choose the first point, drag and release the button at the place of the second point. I use `glRect` function, which is provided by OpenGL, for drawing.

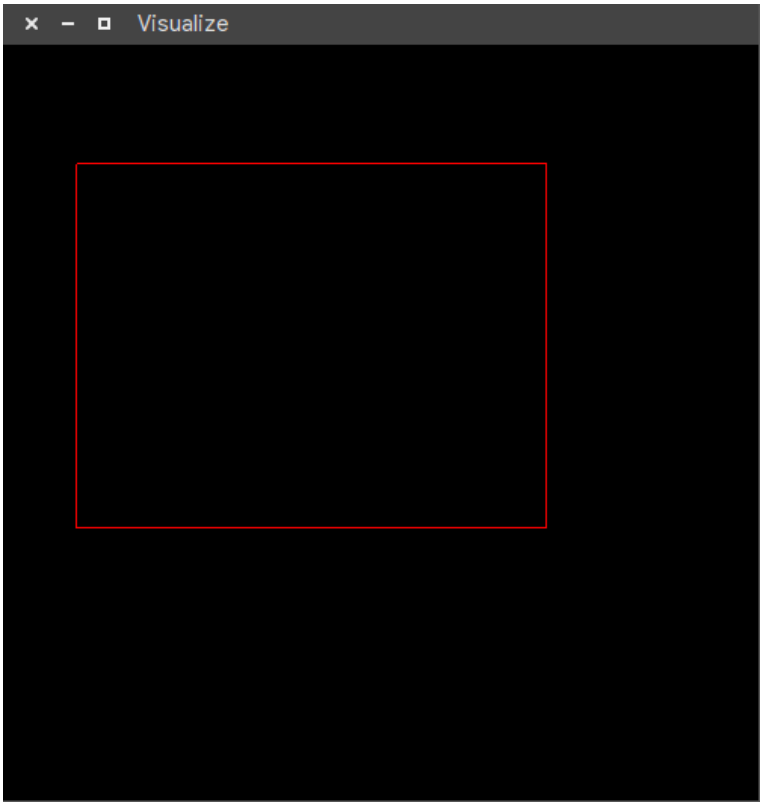

Figure 2: Demo for drawing rectangle

2.2.2 Circle and Ellipse

As same as rectangle, you have to define the rectangle and the algorithm will calculate the maximum inscribed ellipse or circle. In case of the shape is circle, the radius is half of minimum value between height and width of the rectangle. With ellipse, it considers the width and height as major and minor respectively. Both drawing use Midpoint algorithm which implemented in previous lab.

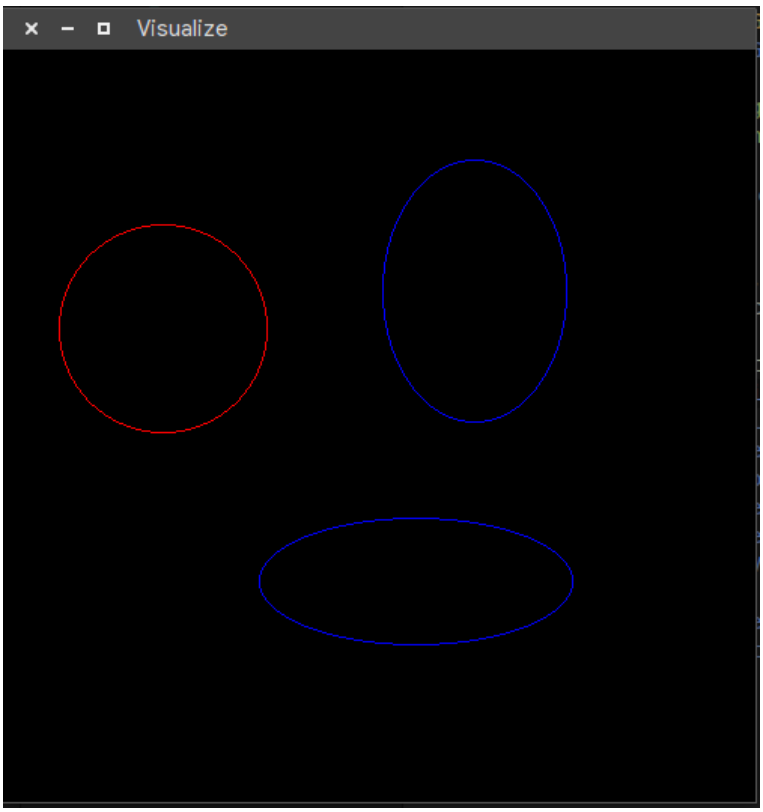

Figure 3: Demo for drawing circle and ellipse

2.2.3 Polygon

Drawing polygon can be done by drawing line by line; each line is defined by two points: start, and end. So, for choosing a point, you need to click left button. However, since the second line, the start point is the end point of

the previous line; the last line will be automatically connected with the start point of the first line by right click event. Due to the right button is using to trigger context menu, I detach menu away from the button at the starting and attach again when the process finish.

Figure 4: Demo for drawing polygon

2.2.4 Regular Polygon

First, I determine the circle as in 2.2.2. Thus, the regular polygon is inscribed the circle. The vertex i-th of the polygon is represented by r and φ_i in polar coordinates. The angle distance between two continuous vertex is α.

$$
\begin{aligned}
r &= \text{radius of circle} \\
\varphi_1 &= 0 \\
\varphi_i &= \varphi_{i-1} + \alpha
\end{aligned}
$$

After that I convert to Cartesian coordinates which has fomular as:

$$x = r\cos\varphi$$
$$y = r\sin\varphi$$

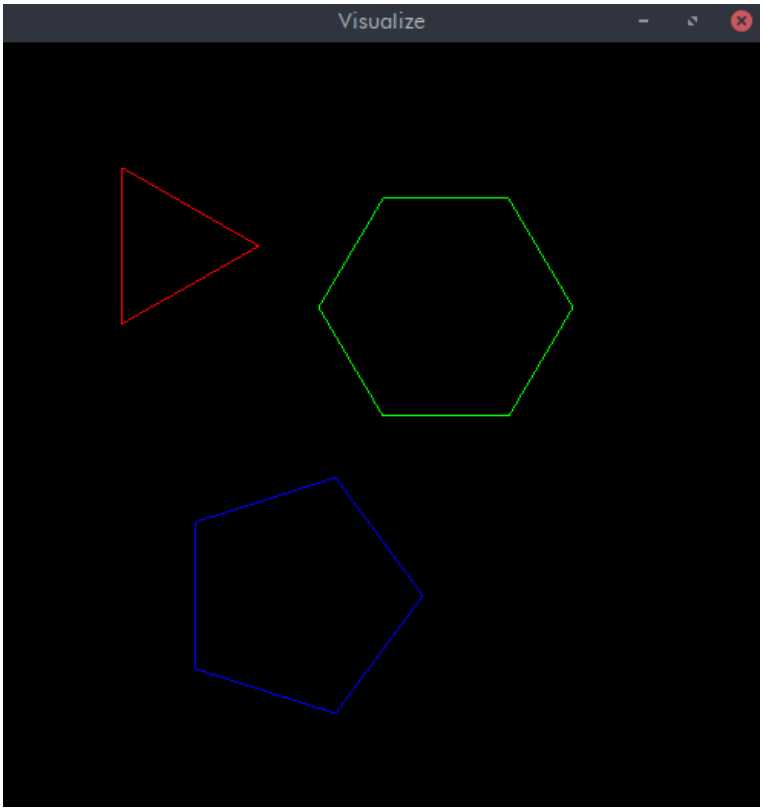

Figure 5: Demo for drawing regular polygon.

2.3 Fill color

I use Flood fill algorithm to fill the color of an bounded area.

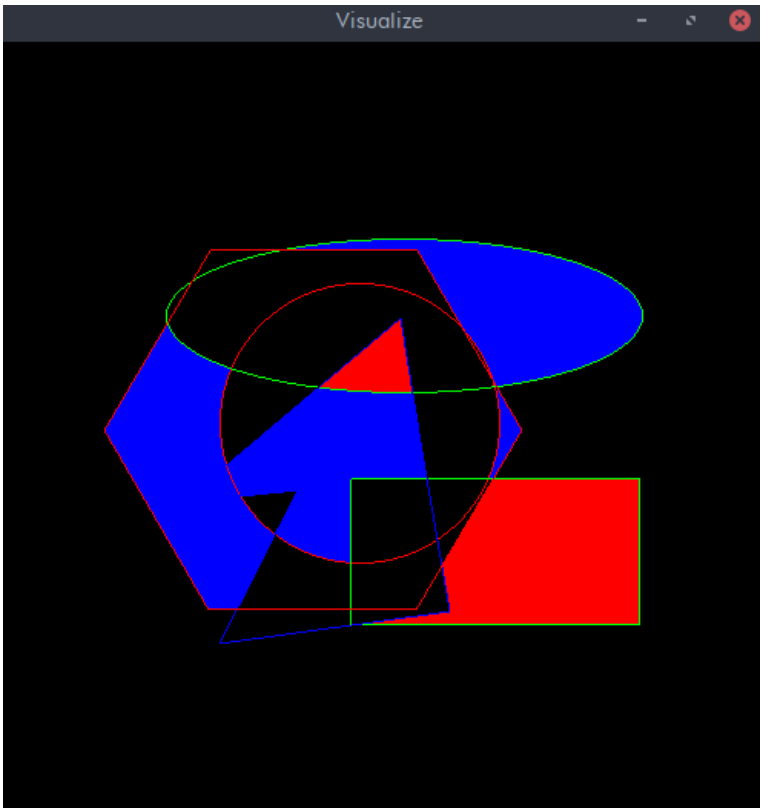

Figure 6: Demo for filling color