HBM 601E Computational Geometry

Homework 1 Report

Kadriye Tuba Turkcan

25.11.2020

This report represents drawing a circle passing through given 3 points using OpenGl in C programming language.

First, 3 points are asked from the user by left clicking the mouse. To do this, first we defined maximum number of points N_max equal to 3. Then we assigned 0 to N_points which shows current number of clicked points and defined `X_pts_array and Y_pts_array` to hold these points inside. After defining window and its size, we defined and set the mouse position to zero. To show the command to the user, **`drawBitmapText`** function with font `GLUT_BITMAP_TIMES_ROMAN_24` is added. At this part, since the text is represented by bitmap characters, it was impossible to change the font size and get a bigger one. Inside the main, **`glutInitWindowSize()`** is called to represent a window with given size on the screen, followed by the position and the title of the window. Inside Display() function, `"Draw 3 points by left clicking the mouse!"` is written by **`drawBitmapText`** to show the text on the top left corner of the window. This text is shown if the number of points is less than 3. Inside the same if loop, the color(green) of the points is determined by **`glColor3f`**`(0.0, 1.0, 0.0)` which gives the colors according to RGB system. Than each point is represented from the points arrays by clicking the left mouse. To do this, a mouse function is added before display and inside it, a condition is proposed to show the points when the right click is less than or equal to 3.

Second, to show the center of the circle to the user, the following algorithm is used. The given 3 points can be represented by two points by moving one of them (here P3) to (0,0) position. Then calculating the new positions for P01 (x01,y01) and P(x02, y02). By doing the calculations for the new points we get c=0, and new values for b and a, which are easier to represent inside the code. Using these b and a, we get the center (X0M,Y0M) equal to (-a/2, -b/2). This is the center for the new coordinates, we need to translate it back to its space by adding back the coordinates of P3. At this point, I couldn't see the center at first, then after translation, it became clear. Also, radius is calculated from this center point. It is given by the following code:

```
x01= X_pts_array[0] - X_pts_array[2];
y01= Y_pts_array[0]- Y_pts_array[2];

x02= X_pts_array[1] - X_pts_array[2];
y02= Y_pts_array[1]- Y_pts_array[2];

d = (y02 * x01) - (y01 * x02);

z01 = -((x01*x01)+(y01*y01));
z02 = -((x02*x02)+(y02*y02));

a = (y02*z01-y01*z02)/(y02*x01-y01*x02);
b = (z02*x01-z01*x02)/(y02*x01-y01*x02);

//center of the circle with translated coordinates
cX = -(a/2);
cY = -(b/2);

//radius of the circle
```

```
r = sqrt(cX*cX+cY*cY);

//center of the original circle
cX = cX + X_pts_array[2];
cY = cY + Y_pts_array[2];
```

We wanted to show this center with a right click. First, a center variable is defined and set to 0 to show the center after 3 points are clicked. When the number of points is 3, 1 is assigned to center and the following is used to show the text to the user

```
drawBitmapText("Center of the circle",cX-60,cY+30);

drawBitmapText("Right click to see the circle",50,50);
drawBitmapText("passing through these points",50,70);
```

Here the text for center is a little bit below the center position and others are on left top corner. A condition is set to show this center before the right click.

Third, when right clicked, the circle is drawn. To do this a right_click variable is defined and assigned to 0 in the beginning, to set a condition inside the mouse function. Also, a circle function which draws the circle with given center and radius is given before display function. Inside the circle function, with given radian values, the points on the circle are found and by connecting these points with `glBegin(GL_LINE_LOOP);` the circle is drawn. Again a color is set before the drawing. The following code shows the function:

```
void drawcircle( double r,  double cX, double cY)
{
      glLineWidth(4);
      glColor3f(1, 0, 1);
      glBegin(GL_LINE_LOOP);
      int i;
      int precision = 100;
      double theta = 0;
        for(i = 0; i < precision; i++){
          theta = i * 2 * M_PI/precision;
          glVertex2d(cX + r*cos(theta), cY + r*sin(theta));
        }
        glEnd();
}
```

However, to use this function was a tricky one. First, I put it inside the mouse function with a condition of right button but it didn't work. Also, when I called it from the display function, there was no problem, it could be seen on the screen but I couldn't set the condition right click. Since everything is seen inside the Display function, I needed something to use both mouse and display functions. Finally, I decided to use a right_click variable and set it to zero, and change it to 1 once right clicked inside the mouse function. Inside the display function, if right_click is equal to 1 then, the circle function is called. This can be seen from the following code block:

```
if (right_click == 1 && d!= 0) {
```

```
        drawcircle(r,cX,cY);
        drawBitmapText("Press F1 to exit!",50.0,50.0);
        drawBitmapText("Left click to draw a new circle",50,70);

    }
```

Here d is the denominator of b and a variables and if they are equal to 0, the points lie on a straight line and we get infinite curvature so the circle will not be able to be seen on the screen. Actually, after writing the code, I tried to put the points aligned on a line, however, it was impossible to do that by hand. In every situation, although it didn't show the whole circle, there was a circle that can be seen on the window. As it can be seen from above, the user is asked to press F1 or right click to decide what to do. This F1 key function is given inside the **`processSpecialKeys()`** by destroying the window when F1 pressed.

In the following figure, we can see a circle with given 3 points on different quadrants of the circle.

The following figure shows the circle with given 3 points on the same quadrant of the circle.

In conclusion, here a program that asks from the user to draw 3 points by left clicking the mouse and draws a circle from these points is explained. To do this program C language is used on an Eclipse platform with OpenGl graphics tool.