

Graphics Report

K.vamshi

Execute the project by using make command in the terminal. It outputs a executable file transform which can be run using ./transform. The code has been documented well to understand the algorithms which were used in the code.

The functionalities which are implemented in this assignment are drawing of lines (bold and dotted lines), parabola, ellipse, circle, "UP" anti aliasing letter and face figure as mentioned in the assignment. In this assignment we have to implement the line, other polynomial drawing algorithms and compare them with openGL drawings. The computer screen is made up of discrete set of pixels, We have to chose the appropriate pixel while drawing images on the screen. We have used midpoint algorithm to choose a appropriate pixel and plot the required figures on the computer screen.

LINE DRAWING

The first screen which appears after running the program has the functionality to draw line objects on the window. User can draw any number of objects by selecting the start and end points on the screen. I have implemented Midpoint Gupta-sproull's algorithm to plot the lines. User can start drawing lines by left clicking on a point on the window which is created by running the program. The line is dynamically drawn along with the mouse until the user fixes other end of the line by left clicking on any another part of the screen. Algorithm draws the line between start and end points. Right clicking the mouse does end the plotting of this object and user can draw one more object beside the current object. Below figure shows a screenshot of the implementation of "MAKE" where **left had side is made up of midpoint** Gupta-sproull's algorithm and the **right hand side is made up of openGL GL_LINES function**.



Functionality

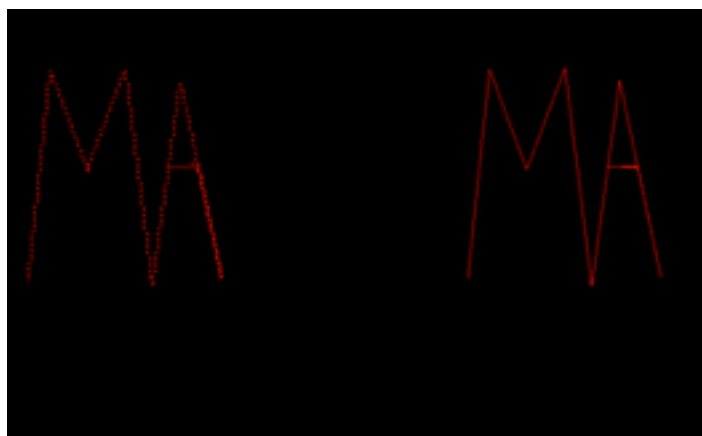
1. **Escape Button:** Pressing the escape button closes the window and stops the program.
2. **Left button click (LButtonDown):** *add a new control point*
3. **Mouse moving without clicking any buttons (MouseMove):** *draw a line from the previously added control point to the current cursor position. The line drawing should be refreshed dynamically as the mouse moves until the left button is clicked again.*
4. **Right button click (RButtonDown):** *add last control point. Your program should draw all the line segments created up to this point until the user clicks the left button again. If the user clicks the left button again after clicking the right button, the last line segment list should be cleared and a new line segment list started.*

OTHER OPTIONS

Clicking on the **middle button** of the mouse user is provided a menu where he can select various options like drawing a dotted line, circle, parabola, ellipse, "UP" which is made up of parabola, line and a semicircle and a face which is made up of 3 circles and a parabola

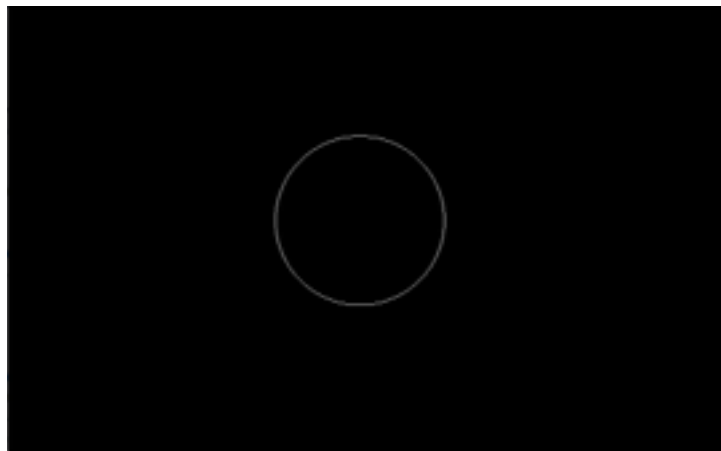
Dotted Lines

User can also draw dotted lines similar to bold lines which is mentioned above. I have used midpoint algorithm which was used in the previous case. I have used plotting the lines for even x,y points so that the lines look dotted leaving the odd pixels empty although we compute their values. The functionalities of the dotted line remain the same as that of drawing a normal line which was explained in the previous section.



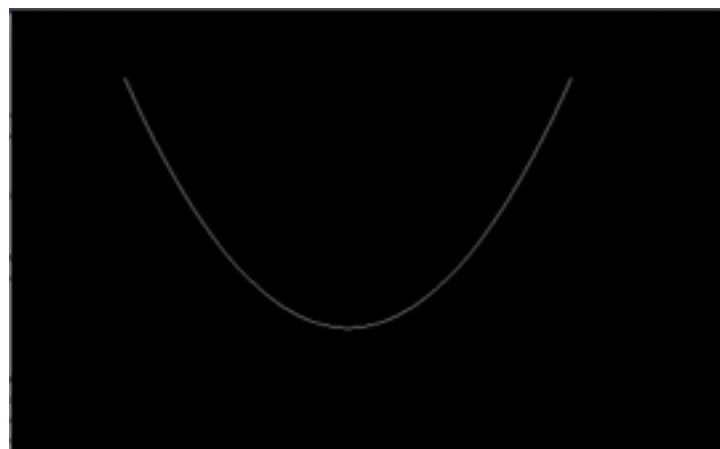
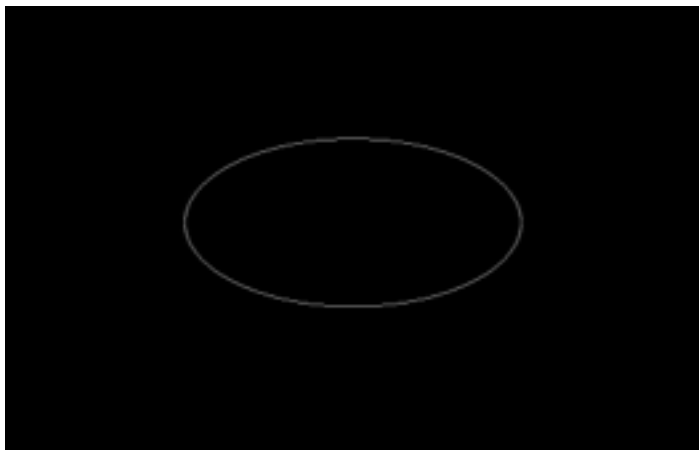
CIRCLE

Midpoint algorithm has been implemented to draw a circle. Circle has a 8 way symmetry, so its enough if we are able to plot the circle in any one of the 8 co-ordinates. I have implemented the algorithm which plots the circle in 45-90 deg and I am reflecting the arc in other co-ordinates to make it form a circle. The basic concept of the midpoint algorithm is to check whether the midpoint joining to pixels is present inside the circle or outside the circle. If it is inside the circle we select the SE point else we select the east point. Computations are made using this concept to plot the circle.



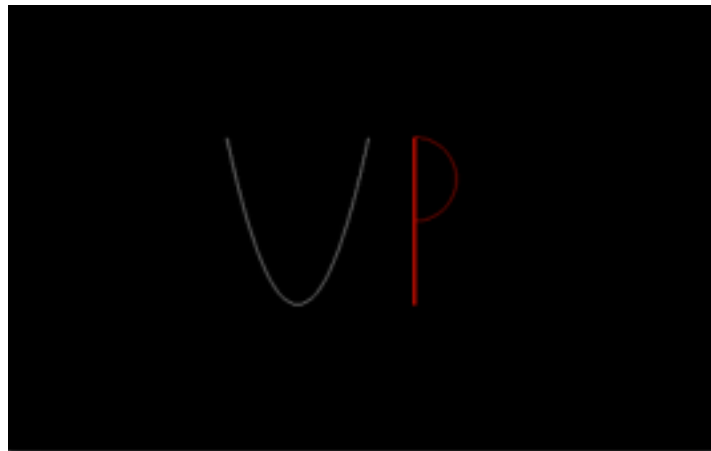
Ellipse and parabola.

We can consider ellipse and parabola as the same case. In both cases there is 4 way symmetry. We have to take two cases into consideration $dy/dx > 1$ and $dy/dx < 1$. The curves show different properties in these two cases. For example, if we consider ellipse which is shown in the figure we can see that from $(0, dy/dx=1)$ the curves 'x' co-ordinate values are increasing rapidly and from $dy/dx > 1$, the y values are increasing rapidly. Similarly case occurs in parabola also. So I have used the midpoint algorithm to compute the value where $dy/dx = 1$ and divided the problem into 2 sub problems and solved them individually using the midpoint algorithm which we have used in case of circles.



“UP” WORD

The “UP” word is implemented using the parabola, a straight line and a semi circle. We can see that all the curves are anti-anti aliased and the semi circle has been implemented using the circle which was previously made. Gupta-sproull’s anti aliasing has been used to vary the intensity of the line which we can see in the figure.



FACE WITH CIRCLES AND PARABOLA

Since we have constructed parabola and circles in the previous cases, I have used them to construct the face. As we can see in the figure, the mouth region has been made using the parabola, eyes and face are constructed using the circle which was previously made. As we can see the animation, The face smiles and opens and closes eyes in the terminal. This was implemented by running the display function in the loop.

