Computer Graphics (IT331)  
Task 3- Problem1

Names:

* Lamya Raed Aly 20170399 3CS-4
* Alaa Farouk 20170058 3CS-3

‣Steps:

* **Line Drawing (Using Line Directed Algorithm):**

- Given the endpoints (x1, y1) and (x2, y2), we want to draw an approximation of the line connecting (x1, y1) to (x2, y2) :   
Y = y1+(x-x1)\*slope .

Algorithm  
 DrawLine(xs ,ys , xe ,ye){

1. Dx = xe - xs

2. Dy = ye - ys

3. Slope = dy/dx

4. If xs > xe then

swap with xs with xe

swap with ys with ye

end if

5. for x=xs to xe

y= ys+(xe-xs)\*slope

yi=round(y)

drawpixel(x,yi)

end for

end Algorithm

* **Circle Drawing (Using Circle Iterative Polar):**

- Given the circle center point (xe ,ye) and radius R, we want to draw an approximation of the circle.

* **We have equations:**
* **For Polar Form Equation:**

- If the angle between the vector from the center point (xe,ye ) to the point (x, y) and the positive x axis direction is theta then the coordinates of the point (x, y) are :

X= xe + R Cos(theta).

Y= ye + R sin(theta).

* **For Iterative Polar Algorithms:**

-Next point (x’, y’) to draw on the circle is obtained by increasing theta by delta theta , thus it has polar coordinates (R , theta + delta theta).

X’ = R Cos(theta + delta theta)

Y’ = R Sin (theta +delta theta).

Expanding x’ and y’ using some trigonometric identities:

* X’ = R (Cos theta Cos delta theta - sin theta sin delta theta )

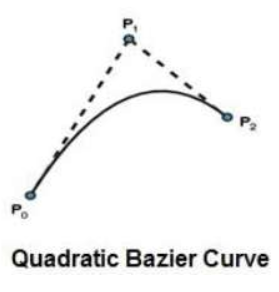
= (R cos theta )cos delta theta - R(sin theta ) sin delta theta

= x cos delta theta - y sin delta theta .

* Y’ = R (Cos theta Sin delta theta - Sin theta Cos delta theta )

= (R cos theta ) Sin delta theta + R(sin theta ) sin delta theta

= x sin delta theta - y cos delta theta .

* **Curve Drawing (Using quadratic Bezier curve (determined by three control points)):**
* Given three control points P0, P1 and P2 we develop a divide procedure that is based upon a parameter t, which is a number between 0 and 1 (the illustrations utilize the value t = 0.0001).

**Derivation Steps:**

* First let P (1) 1 be the point on the segment P0P1 defined by

P (1) 1 = (1 − t)P0 + tP1 (= P0 + t(P1 − P0)).

* then let P (1) 2 be the point on the segment P1P2 defined by

P (1) 2 = (1 − t)P1 + tP

* finally let P (2) 2 be the point on the segment P (1) 1 P (1) 2 defined by

P (2) 2 = (1 − t)P (1) 1 + tP

* Define P(t) = P(2)2.   
   P(t) becomes a functional representation of the Bezier curve  
   P(t) = P (2) 2 (t) = (1 − t)P (1) 1 (t) + tP (1) 2 (t)

where P (1) 1 (t) = (1 − t)P0 + tP1, and P (1) 2 (t) = (1 − t)P1 + tP2 .  
 (Note that we have now denoted P (1) 1 and P (1) 2 as functions of t.)

* Substituting these two equations back into the original, we have   
  P(t) = (1 − t)P (1) 1 (t) + tP (1) 2 (t)   
   = (1 − t) [(1 − t)P0 + tP1] + t[(1 − t)P1 + tP2]   
   = (1 − t) 2P0 + (1 − t)tP1 + t(1 − t)P1 + t 2P2   
   = (1 − t) 2P0 + 2t(1 − t)P1 + t 2P2

**The Analytical Formula**

– Given points P0, P1, and P2, we can construct a curve P(t) by the following  
 P(t) = (1 − t) 2P0 + 2t(1 − t)P1 + t 2P2 for t ∈ [0, 1].

* **Clipping Technique** :
* In our program we have a getRadius Function which used to calculate the circle radius (between two entered points).
* At first, when user enters the face points we calculate the radius and store it in a global variable to be easily accessed whenever needed
* Then, when we start drawing the lines(Hair Lines) we check each point (pixel) whether it is outside the circle or not using LineColour function which calculates the radius between this point and the face center.
* If the calculated radius > the actual face radius, then we return the colour red, otherwise we return blue.

**Steps For our program:**

* First we have number of points like as a counter for clicks .
* When the user enters 2 clicks for drawing face so we call function draw iterative circle. we took from user two points x , y and we compute Radius by
* getRadius function which use the 2 points to calculate the circle’s radius .
* Then, the user enters another two clicks for right eye so we call function draw iterative circle .
* Then the user enters anther two clicks for left eye so we call function draw iterative circle .
* Then the user enter three clicks for smile . so we call function draw quadratic curve .
* The the user enters 6 clicks for the hair . so we call function draw line .
* And we took from the user 4 points from 2 clicks .