

Computer Graphics Project

Report

Cartoon chosen : Tom

(using python-turtle module)

Submitted by:

Aryan Gupta - 21ucs248

Vaibhav Goyal - 21ucs223

Submitted To:

Dr. Alope Dutta

Course Instructor

Github Link:

<https://github.com/algoviber/CG-Project-Tom>

INTRODUCTION

For making Tom, we are using Python turtle programming . Turtle is a Python library that enables users to create pictures and shapes by providing them with a virtual canvas. The onscreen pen that we use for drawing is called the turtle and this is what gives the library its name. To install the turtle library, one needs to run the following command in command prompt: **pip install turtle**

OUTPUT OF CODE



CODE BREAKDOWN

⇒ Drawing the Outline of its Body:

```
def body(): #Function for body of Tom(structure of Tom)

    # TOM FACE*****
    t.hideturtle() # Used for hiding the drawing cursor(turtle)
    t.pensize(2.5)
    t.fillcolor(BODY_COLOR)
    t.speed(15)

    # setting position of turtle
    t.up()
    t.setpos(-50, 300)
    t.down()
    t.begin_fill()
    t.left(85)
    t.circle(25, 35)
    t.circle(200, 20)
    t.left(30)
    t.circle(100, -45)
    t.back(20)
    t.left(80)
    t.circle(80, -30)
    t.right(30)
    t.back(20)

    t.circle(200, 40)
    t.left(5)
    t.circle(200, -40)
    t.right(180)
```

```
t.circle(250, -20)
t.left(5)
t.circle(280, 25)
t.right(115)
t.forward(20)
t.circle(250, -25)
t.right(10)
t.circle(250, -12)
t.left(15)
t.circle(250, 15)
t.circle(250, 20)

# *****

t.pensize(2.5)
t.back(5)
t.setheading(90)
t.left(5)
t.circle(150, -15)
t.right(75)
t.circle(20, 80)
t.circle(100, -15)
t.left(110)
t.circle(20, -30)
t.left(45)
t.circle(40, 20)
t.right(75)
```

```
t.circle(20, -20)
t.left(20)
t.forward(10)
t.circle(5, 110)
t.left(25)
t.forward(10)
t.circle(20, 20)
t.forward(10)
t.left(200)
t.circle(200, -7)
```

```
# ear checkpoint

t.right(5)
t.circle(150, -15)
t.setheading(123.0)
t.left(45)
t.circle(200, 15)
t.right(15)
t.back(20)
t.left(15)
t.circle(150, 20)
```

```
# elbow
t.right(60)
t.back(200)
t.circle(25, -120)
```

```
t.circle(500, -15)

# Hand curves
t.left(20)
t.circle(150, 10)
t.right(20)
t.circle(150, -10)
t.left(25)
t.circle(150, 5)
t.right(25)
t.circle(150, -5)
t.left(25)
t.circle(150, 5)
t.right(25)
t.circle(150, -5)
t.circle(10, -100)
```

```
t.left(120)
t.forward(50)
t.circle(20, 60)
```

```
t.left(70)
t.forward(15)
t.right(5)
t.circle(200, 15)
```

```
t.left(90)
```

```
t.left(90)
t.circle(400, 15)
t.left(15)
t.circle(250, 15)
t.left(180)
t.circle(100, -50)
t.backward(10)
t.right(120)
t.circle(50, 80)
t.left(180)

#white portion of tail starts

t.circle(150,-50)
t.circle(150,-20)
t.left(80)
t.right(180)
t.circle(40, -40)
t.back(30)
t.right(20) # tail white portion end
t.left(180)
t.circle(100, 60)
t.right(180)
t.circle(130, -20)
t.right(30)
t.circle(50, -20)
t.back(30)

# upto tail
```

```
t.left(90)
t.circle(100, -20)
t.right(30)
t.circle(100, -20) # 1st leg white portion start
t.circle(100, -10)
t.circle(30, -70)
t.circle(30, 70)
t.circle(100, 10) # returning to 1st leg white portion

t.circle(100, 20)

t.left(180)
t.circle(20, -80)
t.circle(250, -20)
t.left(60)
t.circle(30, -60)
t.left(90) # 2nd leg starts (101.0, -281.0)

t.circle(20, 40)
t.circle(20, -180)
t.left(40)
t.circle(20, 110)
t.circle(20, -200)
t.left(90)
t.circle(20, 120)
t.circle(20, -240)
t.back(10)
```

```
t.left(30)
t.circle(400, -18)
t.circle(10, -80)
t.back(60)

# # #####
t.left(90)
t.circle(70, -35)
t.right(20)
t.circle(70, 20)
t.left(10)
t.circle(100, -35)
t.back(60)
t.circle(100, -15)

# Stomach border starts
t.left(30)
t.circle(200, -35)
t.back(100)
t.right(180)
t.circle(150, 10)
t.right(70)
t.back(100)
t.circle(50, -90)
t.left(20)
t.circle(50, 30)
t.right(20)
```

```
t.circle(50, -95)

# left hand white border starts
t.right(60)
t.right(180)
t.circle(80, -40)
t.left(30)
t.circle(80, -50)
t.circle(10, -120)
t.back(10)
t.left(110)
t.circle(120, 35)
t.circle(120, -35)
t.circle(15, -160)
t.back(10)
t.right(30)
t.left(180)
t.circle(80, 50)
t.circle(80, -50)
t.circle(10, -140)
t.right(180)
t.right(20)
t.circle(150, 25)
t.circle(150, -5)
t.left(70)
t.left(180)
t.circle(50, -35)
```

```
t.circle(50, 35)
t.left(180)
t.right(70)
t.circle(150, -2)
t.right(70)
t.circle(80, -30)
t.circle(50, -30)
t.circle(20, -80)
t.right(10)
t.circle(400, -15)
t.circle(400, 10) # hand outline complete

t.right(50)
t.forward(120)
t.right(180)
t.circle(15, -125)
t.circle(15, 60)
t.right(15)
t.circle(150, 20)
t.left(40)
t.back(15)
t.right(20)
t.circle(100, 25)
t.left(60)
t.circle(70, -70)
t.right(20)
t.circle(100, 15)
```

```
t.left(20)
t.circle(180, -5)
t.right(10)
t.circle(180, -10)
t.left(100)
t.circle(100, -30)
t.right(20)
t.circle(80, -35)
t.setposition(-24.0, 311.0)
t.circle(50, -30)

t.end_fill()
```



⇒ Drawing his right hand :

```
def righthand(): #Function for Right hand
    t.up()
    t.setposition(60.0, -61.0)
    t.down()
    t.setheading(8.0)
    t.color("black", "white")

    t.begin_fill()
    t.left(20)
    t.circle(150, 10)
    t.right(20)
    t.circle(150, -10)
    t.left(25)
    t.circle(150, 5)
    t.right(25)
    t.circle(150, -5)
    t.left(25)
    t.circle(150, 5)
    t.right(25)
    t.circle(150, -5)
    t.circle(10, -100)

    t.right(50)
    t.forward(30)
    t.circle(10, 50)
    t.circle(80, 85)
    t.circle(15, 90)
```

```
t.circle(60, 30)
t.circle(10, 180)
t.forward(10)
t.back(10)
t.circle(10, -180)
t.circle(60, -35)
t.right(30)
t.circle(60, 40)
t.circle(10, 180)
t.circle(10, -180)
t.circle(60, -25)
t.right(40)
t.circle(80, 20)
t.circle(10, 182)
t.circle(10, -182)
t.circle(80, -20)
t.left(40)
t.circle(60, 25)
t.circle(10, 180)
t.right(160)
t.circle(80, 30)

t.end_fill()

t.up()
t.setposition(46.0, -58.0)
t.down()
```

```
t.setheading(0)
t.right(50)
t.right(180)
t.circle(70, -30)
```



⇒ Drawing his white tail portion :

```
def tailcolor(): #Function for filling Tail white shade
    t.up()
    t.setposition(249.49,-215.80)
    t.down()
    t.setheading(163.0)
    t.color("black","white")

    t.begin_fill()

    t.circle(150,-20)
    t.left(80)
    t.right(180)
    t.circle(40,-40)
    t.back(30)
    t.right(20)

    t.pensize(1)
    t.left(30)
    t.circle(50,30)
    t.right(60)
    t.right(180)
    t.circle(50,40)
    t.left(10)
    t.circle(50,-20)
    t.right(20)
    t.circle(50,20)
    t.left(30)
```

```
    t.circle(50,-20)
    t.right(20)
    t.circle(50,30)
    t.left(20)
    t.circle(50,-40)

    t.end_fill()
```



⇒ Drawing his ear:

```
def ear(): #Function for Ear
    t.up()
    t.setposition(143.0, 331.0)
    t.down()

    # *****
    t.begin_fill()
    t.color("black", "pink")

    t.setheading(208.0)
    t.circle(250, 20)

    t.pensize(2.5)
    t.back(5)
    t.setheading(90)
    t.left(5)
    t.circle(150, -15)
    t.right(75)
    t.circle(20, 80)
    t.circle(100, -15)
    t.left(110)
    t.circle(20, -30)
    t.left(45)
    t.circle(40, 20)
    t.right(75)
    t.circle(20, -20)
    t.left(20)
```

```
t.left(20)
t.forward(10)
t.circle(5, 120)
t.left(25)
t.forward(10)
t.circle(20, 20)
t.forward(10)
t.left(200)
t.circle(200, -7)

t.right(120)
t.forward(10)
t.circle(50, 75)
t.forward(10)
t.left(180)
t.circle(150, -40)
t.setposition(180.0, 348.0)

t.end_fill()
```



⇒ Drawing his left hand:

```
def leftHand(): #Function for Left hand
    t.up()
    t.setposition(-294.0, 36.0)
    t.down()
    t.begin_fill()
    t.color("black", "white")
    t.right(60)
    t.right(180)
    t.circle(80, -40)
    t.left(30)
    t.circle(80, -50)
    t.circle(10, -120)
    t.back(10)
    t.left(110)
    t.circle(120, 35)
    t.circle(120, -35)
    t.circle(15, -160)
    t.back(10)
    t.right(30)
    t.left(180)
    t.circle(80, 50)
    t.circle(80, -50)
    t.circle(10, -140)
    t.right(180)
    t.right(20)
    t.circle(150, 25)
    t.circle(150, -5)
```

```
    t.left(70)
    t.left(180)
    t.circle(50, -35)
    t.circle(50, 35)
    t.left(180)
    t.right(70)
    t.circle(150, -2)
    t.right(70)
    t.circle(80, -30)
    t.circle(50, -30)
    t.circle(20, -80)
    t.right(10)
    t.circle(400, -15)
    t.circle(400, 10)
    t.right(50)
    t.circle(400, -12)

    t.end_fill()
```

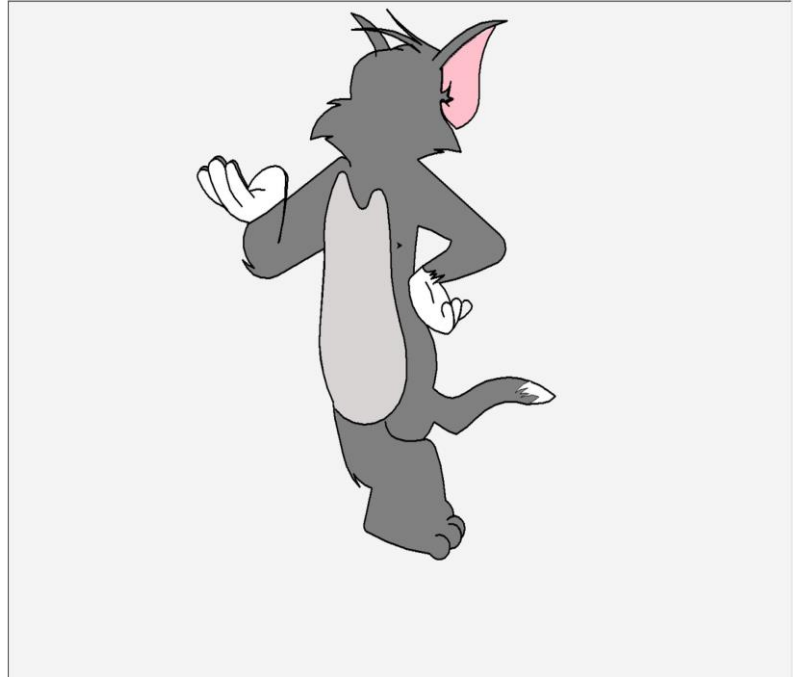


⇒ Drawing his Stomach:

```
def stomach(): #Function for Stomach
    t.up()
    t.setposition(-132.0, -249.0)
    t.down()
    t.setheading(270.0)
    t.begin_fill()
    t.color("black", "#D7D4D4")
    t.left(30)
    t.circle(200, -35)
    t.back(100)
    t.right(180)
    t.circle(150, 10)
    t.right(180)
    t.circle(200, -35)
    t.circle(10, -140)
    t.right(180)
    t.circle(100, 30)
    t.circle(10, 120)
    t.right(180)
    t.circle(200, -5)
    t.circle(15, -90)
    t.right(25)
    t.circle(50, -30)
    t.circle(500, -10)
    t.right(180)
    t.circle(300, 20)
    t.right(180)
```

```
t.circle(200, -25)
t.circle(80, -130)

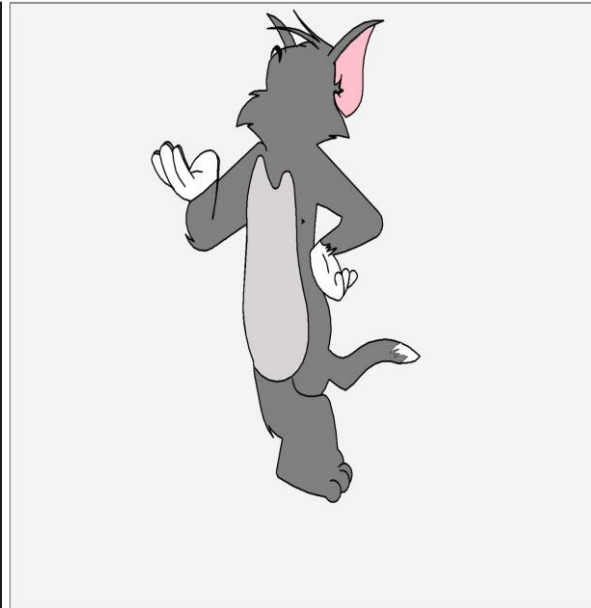
t.end_fill()
```



⇒ Drawing his left eyebrow:

```
def lefteyebrow(): #Function for Left Eyebrow
    t.up()
    t.setposition(-87.0, 290.0)
    t.down()
    t.color("black", "black")
    t.begin_fill()
    t.setheading(90)
    t.left(5)
    t.right(180)
    t.circle(20, -70)
    t.circle(10, -90)
    t.circle(20, -70)
    t.left(30)
    t.circle(20, 60)
    t.circle(20, 20)

    t.end_fill()
```



⇒ Drawing the Outline of his eyes:

```
def eyes(): #Function for both the eyes
    t.up()
    t.setposition(-95.0, 215.0)
    t.down()
    t.color("black", "#F9EE6D")
    t.begin_fill()
    t.setheading(270)
    t.left(7)
    t.circle(150, -25)
    t.circle(15, -170)
    t.forward(10)
    t.back(80)

    t.end_fill()
    # *****
    t.color("black", "#D7D4D4")
    t.begin_fill()
    t.forward(70)
    t.left(180)
    t.circle(5, -180)
    t.back(10)
    t.forward(10)
    t.right(180)
    t.circle(100, -15)
    t.circle(5, -100)
    t.circle(10, -20)
    t.circle(10, -30)

    t.back(25)
    t.right(10)
    t.forward(10)
    t.left(180)
    t.circle(5, -180)
    t.right(180)
    t.right(10)
    t.circle(300, 15) # grey portion ends

    t.end_fill()

    # right eye starts
    t.color("black", "#F9EE6D")
    t.begin_fill()

    t.circle(300, -15)
    t.right(40)
    t.circle(200, -5)
    t.circle(15, -150)
    t.left(20)
    t.circle(250, -18)

    t.end_fill()
```



⇒ Drawing his mouth:

```
def mouth(): #Function for structure of mouth
    t.color("black","white")
    t.begin_fill()

    t.setheading(180)
    t.right(25)
    t.circle(100, 25)
    t.left(10)
    t.circle(50, 15)
    t.right(60)
    t.circle(100, 25)
    t.circle(15, 180)

    # # lower region of mouth starts
    t.right(85)
    t.circle(150,20)
    t.left(70)
    t.right(180)
    t.circle(100,-30)
    t.right(180)
    t.circle(15,70)
    t.circle(15,-90)
    t.back(10)
    t.forward(10)
    t.circle(15,90)
    t.circle(100,25)
    t.circle(100,-15)
    t.right(20)
    t.circle(100,25)
    t.circle(20,50)
    t.right(90)
    t.circle(100,10)
    t.circle(100,-10)
    t.back(10)
```

```
#red portion of mouth starts(tongue)

t.color("black","red")
t.begin_fill()

t.circle(100,-31)
t.circle(10,-40)
t.circle(20,-40)
t.setheading(0)
t.left(70)
t.right(180)
t.circle(30,-50)
t.circle(5,-80)
t.left(90)
t.circle(20,-50)

t.end_fill()
```

```
t.circle(10,-130)
t.pensize(1)
t.circle(50,-30)
t.circle(50,25)
t.pensize(2.5)
t.right(90)
t.setposition(-4.0,193.0)

t.end_fill()
t.up()
t.setposition(-105.0,183.0)
t.down()
t.setheading(0)
t.right(30)
t.circle(50,40)
t.right(50)
t.circle(60,75)

t.color("black","black")
t.begin_fill()

t.circle(60,-75)
t.left(50)
t.circle(50,-10)
t.setheading(320.0)
t.left(180)
t.circle(50,-45)
t.right(180)
t.circle(20,80)
t.circle(10,40)
t.circle(100,31)

t.end_fill()
```



⇒ Drawing his nose:

```
def nose(): #Function for Nose
    t.up()
    t.setposition(-58.0,203.0)
    t.down()
    t.color("black","black")
    t.begin_fill()

    t.setheading(180)
    t.circle(20,50)
    t.circle(5,90)
    t.circle(20,50)
    t.circle(5,90)
    t.circle(20,50)
    t.left(90)

    t.end_fill()
```



⇒ Drawing his right eyebrow: Drawing the Corneas of his

```
def righteyebrow(): #Function for Right Eyebrow
    t.up()
    t.setposition(-5.0, 298.0)
    t.down()
    t.color("black", "black")
    t.begin_fill()

    t.setheading(90)
    t.right(20)
    t.right(180)
    t.circle(20, -180)
    t.left(30)
    t.circle(20, 90)

    t.end_fill()
```



⇒ Drawing his eye

```
def eyecornea(): #Function for Internal Structure of Eye

    # Left eye green cornea starts
    t.up()
    t.setposition(-95.0, 215.0)
    t.down()
    t.color("black", "green")
    t.begin_fill()

    t.setheading(270)
    t.left(7)
    t.circle(150, -15)
    t.right(30)
    t.circle(10, -120)
    t.circle(50, -50)
    t.end_fill()

    # Left eye Black Cornea starts
    t.up()
    t.setposition(-95.0, 215.0)
    t.down()
    t.color("black", "black")
    t.begin_fill()
    t.setheading(270)

    t.left(7)
```

```
t.setheading(272.0)
t.color("black", "black")
t.begin_fill()

t.circle(300, -7)
t.right(30)
t.circle(8, -120)
t.circle(25, -70)
t.back(10)

t.end_fill()
```

```
t.circle(150, -10)
t.right(30)
t.circle(8, -120)
t.circle(25, -80)

t.end_fill()

# right eye Green cornea starts

t.up()
t.setposition(-25.0, 199.0)
t.down()
t.setheading(272.0)
t.color("black", "green")
t.begin_fill()

t.circle(300, -10)
t.right(30)
t.circle(10, -120)
t.circle(50, -63)

t.end_fill()

# Right eye black cornea starts
t.up()
t.setposition(-25.0, 199.0)
t.down()
```



⇒ Drawing his Right Leg:

```
def rightleg(): #Function for Right leg
    t.up()
    t.setposition(6.54,-309.66)
    t.down()
    t.setheading(353.0)
    t.color("black","white")
    t.begin_fill()

    # returning to 1st leg white portion
    t.circle(100,20)
    t.left(180)
    t.circle(20, -80)
    t.circle(250, -20)
    t.left(60)
    t.circle(30, -60)
    t.left(90)

    # 2nd leg starts
    t.right(180)
    t.circle(150,-35)
    t.left(90)
    t.circle(50,60)
    t.right(90)
    t.right(180)
    t.circle(50,-20)
    t.circle(50,10)
    #*****
```

```
t.pensize(1)
t.left(30)
t.circle(30,-30)
t.right(20)
t.circle(40,20)
t.left(40)
t.circle(50,-20)
t.right(60)
t.circle(50,30)
t.right(180)
t.circle(50,30)
t.right(30)
t.circle(30,-60)

t.end_fill()

t.pensize(2.5)
t.up()
t.setposition(68.0,-426.0)
t.down()
t.setheading(270)
t.left(40)
t.circle(20,-60)

t.up()
t.setposition(32.0,-416.0)
t.down()
```

```
t.setheading(270)
t.circle(20,-60)

t.up()
t.setposition(22.43,-353.31)
t.down()
t.setheading(198.0)
t.circle(100,15)
t.right(180)
t.circle(110,-30)
```



⇒ Drawing his Left Leg:

```
def leftleg(): #Function for Left leg

    t.up()
    t.setposition(100.00,-429.27)
    t.down()
    t.setheading(183.0)
    t.color("black","white")
    t.begin_fill()

    t.circle(20, 40)
    t.circle(20, -180)
    t.left(40)
    t.circle(20, 110)
    t.circle(20, -200)
    t.left(90)
    t.circle(20, 120)
    t.circle(20, -240)
    t.back(10)
    t.left(30)
    t.circle(400, -18)
    t.circle(10, -80)
    t.back(60)
    t.forward(30)
    t.left(180)
    t.right(30)

    t.pensize(1)
```

```
    t.right(40)
    t.circle(30,60)
    t.right(180)
    t.left(40)
    t.circle(30,40)
    t.left(110)
    t.circle(20,40)
    t.left(20)
    t.circle(50,-25)
    t.right(45)
    t.left(180)
    t.circle(40,-30)
    t.left(30)
    t.circle(50,30)
    t.right(30)
    t.circle(50,-35)

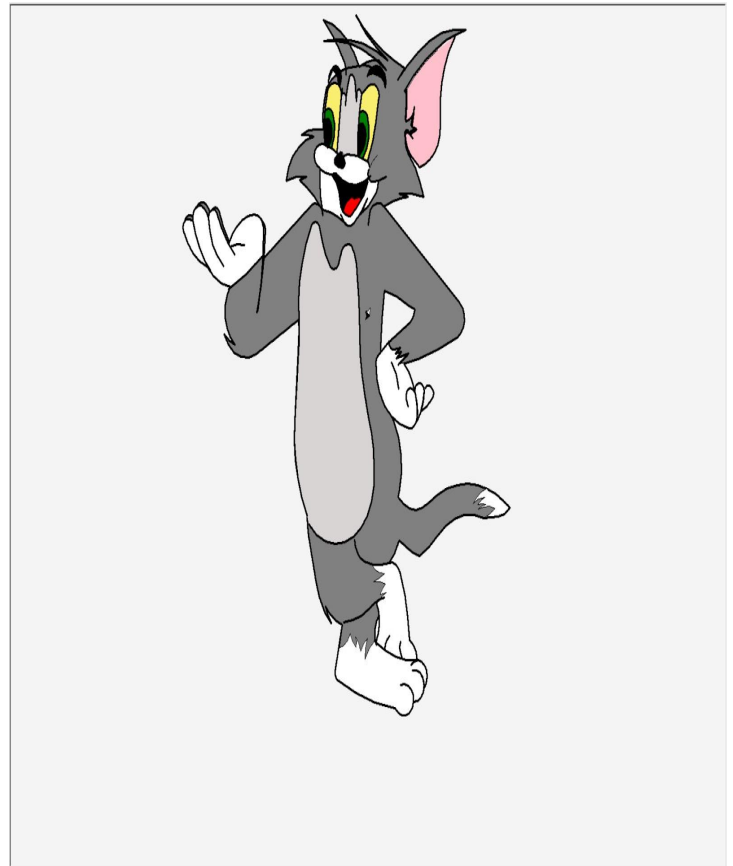
    t.pensize(2.5)
    t.setposition(12.66,-406.69)
    t.setheading(328)
    t.circle(150,35)

    t.end_fill()

    t.up()
    t.setposition(12.66,-406.69)
    t.down()
```

```
    t.color("black","white")
    t.begin_fill()
    t.setheading(58.0)
    t.circle(50,40)
    t.right(50)
    t.right(180)
    t.circle(35,65)
    t.end_fill()

    t.up()
    t.setposition(0,0)
```



TURTLE FUNCTIONS USED

- **up()** : Moves turtle without leaving the canvas.
- **down()** : Stops the pen, letting the turtle draw a line behind it as it moves from up().
- **hideturtle()**: Makes the turtle hide from the screen
- **begin_fill()** : Directs the user to start the desired colour filling.
- **end_fill()** : Directs the user to stop the desired colour filling.
- **circle()** : Allows user to draw a circle by using the turtle rotating in 360 degrees direction.
- **speed()** : Allows the user to adjust the speed of the turtle while joining pixels.
- **fillcolor("nameofthecolour")** : Allows the user to fill the desired colour in.
- **forward()** : Allows user to move the turtle forward by a certain amount passed as parameter of the function.
- **backward()** : Allows user to move the turtle backward by a certain amount passed as parameter of the function.

- **left()** : Allows user to rotate the turtle in anti-clockwise direction by specified angle which is passed as parameter to the function.
- **right()** : Allows user to rotate the turtle in clockwise direction by specified angle which is passed as parameter to the function.
- **screensize()** : Allows the user to adjust the screen size of the window.
- **pensize()**: Sets the line thickness to width or return it. If no argument is given, the current pensize is returned.
- **circle(radius, extent)** : Draws a circle with given radius. The centre is radius units left of the turtle; extent – an angle – determines which part of the circle is drawn.
- **setheading(angle)**: Sets the orientation of the turtle to the given angle.
- **turtle.listen()**: Sets focus on TurtleScreen (in order to collect key-events).
- **turtle.exitonclick()**: Binds the bye() method - bye() method shuts the turtlegraphicswindow - to mouse clicks on the Screen.
- **turtle.done()**: it is the last statement in a turtle graphics programme. It Starts event loop - calling Tkinter's mainloop function.
- **turtle.getscreen()**: Returns the TurtleScreen object the turtle is drawing on.

TOPICS TAUGHT IN CLASS AND USED IN CODE

⇒ Bresenham Line Algorithm:

- It is used to draw the line from one pixel to another.
- **setposition()** function is used to move the turtle from one pixel to another and a line is created.
- Sometimes the line is not smooth for diminishing jagged edges. It is an efficient method because it involves only integer addition, subtractions, and multiplication operations. These operations can be performed very rapidly so lines can be generated quickly .

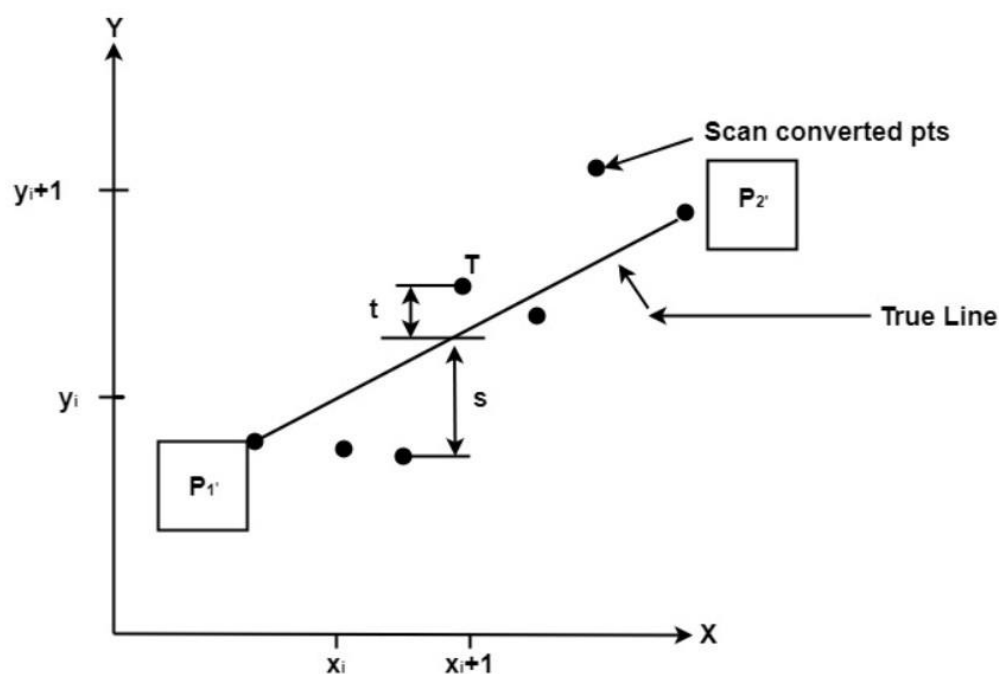


Fig: Scan Converting a line.

⇒ Affine Geometry (Transformations) :

- Affine transformations are precisely those maps that are combinations of translations, rotations, shearings, and scalings. It's the fundamental to repositioning and resizing objects in space.
- The **turtle** knows three things: it's position, its heading, and its scale. The turtle's position is represented by a point in the affine plane; the turtle's heading and scale are stored in a 2-dimensional vector
- Let $Q = (q_1, q_2)$ is a fixed point, $w = (w_1, w_2)$ is a vector, d is a scalar indicating distance, and s is a scale factor.

Suppose we want to translate the turtle using **forward()** and **backward()** functions, then the following transformation will be implemented:

$$\text{Translate}(w, d): \quad P \rightarrow (p_1 \ p_2) * \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + d(w_1 \ w_2)$$

Drawing an arc using the **circle(radius, angle)** function makes use of the following transformation:

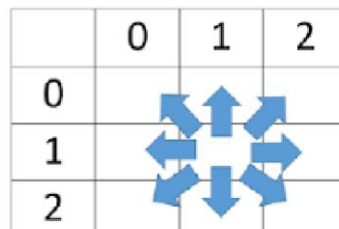
$$\text{Rotate}(Q, a): \quad P \rightarrow (p_1 \ p_2) * \begin{pmatrix} \cos(a) & \sin(a) \\ -\sin(a) & \cos(a) \end{pmatrix} + (q_1 \ q_2) * \begin{pmatrix} 1 - \cos(a) & -\sin(a) \\ \sin(a) & 1 - \cos(a) \end{pmatrix}$$

Changing the heading direction of the turtle using the **left(angle)** and **right(angle)** functions is implemented by the following transformation:

$$(w_1 \ w_2) \rightarrow (w_1 \ w_2) * \begin{pmatrix} \cos(a) & \sin(a) \\ -\sin(a) & \cos(a) \end{pmatrix}$$

⇒ FloodFill Color Filling Algorithm:

- Starting with a single pixel, this algorithm checks if a pixel has the region's original colour. If yes, then we fill it with the new colour. Recursively, it calls each neighbour (either 4-nbd or 8-nbd) till the entire region is coloured.



CONCLUSION

This project helped us to get hands-on experience about how to implement a code for making graphics using Python's turtle library, as we were able to draw Tom (Cartoon Character) with the help of CG concepts like Bresenham line drawing algorithm, 2-D Transformations, and FloodFill Color Filling Algorithm. We thank our course instructor for providing us this opportunity. Knowledge of graphics will help us in our future to serve the tech world.

REFERENCES

1. Abelson, H. and diSessa, A. (1986), Turtle Geometry: The Computer as a Medium for Exploring Mathematics, MIT Press, Cambridge, Mass.
2. https://en.wikipedia.org/wiki/Turtle_graphics
3. Goldman, Ron; Schaefer, Scott; Ju, Tao. "Turtle Geometry in Computer Graphics and Computer Aided Design"
4. <https://docs.python.org/3/library/turtle.html>