Design of Lab3: Random Walks – Jacob Bollinger

Problem:

Create a program to design and run random walks.

Plan:

I will start by importing randint and choice from random and turtle to the program. It will not work without importing these things.

Next I will define some variables like SIZE, STEP\_SIZE, STEPS, RADIUS, DIRECTIONS, ESCAPE, and TITLE. These will be used throughout the program. SIZE will be defined as 500. STEP\_SIZE will be defined as 8 for 8 pixels. STEPS will be defined as a random integer between 1 and 100. The RADIUS will be defined as 200. DIRECTIONS will be defined as the four strings n, e, s, and w. ESCAPE will be set to False. The TITLE variable will be defined Random Walk.

Next I’ll create a function to test if turtle has escaped the circle named hasEscaped. It will use the inputs turtle and radius. In it I will create an if statement to decide if turtle is still in the circle. Test if pos[0] \*\* 2 is greater than RADIUS \*\* 2. If so, set ESCAPE equal to True and write turtle.dot(). Follow that by else if pos[1] \*\* 2 is greater than RADIUS \*\* 2. If this is true, set ESCAPE equal to True and write turtle.dot().

Next I will create a function to draw the walks named drawWalk. The function will use the input steps and radius. Assign steps\_left to be steps. Tell Turtle to create a circle radius big around the origin and go back to the origin. Write turtle.dot() so there will be a dot at the beginning of the walk. Create a while loop to repeat as long as steps\_left is greater than 0 or until the variable ESCAPE is True. Set a value to be choice(DIRECTIONS). This will randomly choose from the four items assigned to DIRECTIONS.

In the while loop create an if statement for if the value is n to tell turtle to move positive on the y-axis STEP\_SIZE pixels. Follow that by an else if statement that tests if the value is e. If it is, tell turtle to move positive on the x-axis STEP\_SIZE pixels. Follow that by an else if statement that tests if the value is s. If it is s, tell turtle to move negative on the x-axis STEP\_SIZE pixels. Follow that by an else if statement that tests if the value is w. If it is the value w, tell turtle to move negative on the x-axis STEP\_SIZE pixels.

While still in the while loop, call the function hasEscaped with input variables turtle and RADIUS. Finish the while loop with assigning steps\_left to be itself minus 1.

Next I will create a function to count the steps. I start by creating an if statement. In the if statement it will test if steps\_left is greater than 0. If it is greater print STEPS. Then follow that by else and print STEPS ­- steps\_left. This will decide whether turtle used up all of its STEPS or broke through the circle before using all of them.

Next I will define my screen and turtle variables. Set screen equal to turtle.Screen(). Set turtle equal to turtle.Turtle(). Set pos equal to turtle.position(). Write turtle.\_tracer(0, 0), screen.colormode(255), screen.title(TITLE), and screen.setworldcoordinates(-50, -50, 300, 300).

I will call the function drawWalk() and give it input variables STEPS and RADIUS. Then I will call the function stepCount() with no additional input variables.

I will finish the code off with turtle.hideturtle() and screen.mainloop()