

The Python Tank Challenge

Dallin Durfee 2021

If you succeed at this challenge, you will be promoted to the rank of “Python User.” If you can complete this challenge, you will have the skills to do basic calculations and graphs with Python.

The challenge

The challenge is to create a simple game in Python. In this game, the user will try to destroy an enemy tank. The game should do the following:

- The game should first randomly select a distance for the enemy tank somewhere from 50 to 150 meters away.
- Next, the game should tell the user how far away the enemy tank is, and then ask them how fast, and at what angle they would like to shoot their projectile at the enemy tank.
- The game should then create a plot with a size of 10 by 5 units. The x axis of the plot should go from -10 to 160 meters. The y axis of the plot should go from -10 to 80 meters. The area below $y=0$ (the ground) should be colored light brown. A green circle should be placed at the origin – the position of the user’s tank. A red x should be placed at the location of the enemy tank. A black line should show the path taken by the projectile, and an orange circle with a radius of 10 meters should be placed centered at the point of impact (the point at which the projectile hits the ground) to indicate the blast radius.
- The game should then tell the user how far from the enemy tank the projectile landed.
- If the user’s tank is within the blast radius, the program should congratulate the user on destroying their own tank and end the game. Otherwise, if the enemy tank is within the blast radius, the game should let them know that they destroyed the enemy tank and end the game. If neither tank is within the blast radius, the game should inform the user that they missed, and ask them if they would like to try again. If they do, it should repeat using the same enemy tank distance. If they don’t, the game should end.

Instructions

To complete this challenge,

- If you don’t have Python installed on your computer, go to <https://www.anaconda.com/products/individual> and download the free Anaconda Python installer.
- Open up Spyder (it’s a Python code editor installed as part of Anaconda Python) or another Python editor.
- Learn Python! Here are some resources to do that. Note that if any of the tutorials tell you to install something, don’t do it. Anaconda installs everything you need for most purposes. If, in the future, you need additional modules, Anaconda has a “Navigator” that you use to install them rather than using other methods that might be mentioned.
 - <https://www.learnpython.org/> has interactive lessons to get you started with Python. For this challenge you will need to do “Hello, World!” through “Loops,” tutorials as well as the “Modules and Packages” and “Numpy Arrays” tutorials.
 - It’s useful for this challenge to know about the numpy “arange” and “linspace” functions:
 - <https://numpy.org/doc/stable/reference/generated/numpy.arange.html>
 - <https://numpy.org/doc/stable/reference/generated/numpy.linspace.html>
 - That should cover what you need to know about Numpy for the challenge. But if you are interested, you can learn more about Numpy in this tutorial https://www.w3schools.com/python/numpy_intro.asp
 - An introduction to plotting with the matplotlib module can be found here: <https://matplotlib.org/stable/tutorials/introductory/pyplot.html> . Read up to, but not including “Plotting with keyword strings.”

- The “fill_between” plot is useful for creating the ground:
https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.fill_between.html
- You can use the markers “o” and “x” to mark the two tanks. But you might find the following useful for making the blast radius circle: <https://stackoverflow.com/questions/9215658/plot-a-circle-with-pyplot>
- Use the equations for projectile motion to find the trajectory. One way to do this calculation is to first calculate the time when the projectile hits the ground. Then make a Numpy array of times from zero to that time. Then calculate an x and y array from the time values. *You should not need to do any loops to calculate the different time, x, and y values* because Numpy can generate arrays and do operations on arrays. For example, if I wanted to plot the sine of x from x = 0 to 5, I could just use

```
x = np.linspace(0,5,100)
```

 to create an array of 100 values going from 0 to 5. Then I could calculate the sine at each point with

```
y = np.sin(x)
```

 and plot them using

```
plt.plot(x,y)
```
- Remember that there is a ton of information about Python online – just search for whatever it is you want to know. Also, don’t be shy about asking for help.

Examples

Below are examples of what the output should look like if they miss, if they hit, and if they destroy themselves.

