Project _1 Instructions/Write-Up - Steve Dille

Included with the code you should do a ~one page write up of what you completed, any challenges you faced, as well as how we should go about testing and using your project.git.

Background:

I completed the game of Battleship in an old looking 1970s/80s looking line editor format. I found this project to be fun as I enjoyed writing scrabble. Also, long ago, you may have heard, there was a famous star trek program written in Basic that was included on UNIX mini computers everywhere. I used to waste a lot of time playing that in the computer lab in the middle of the night. This game is a throwback to that with the matrix printed out, shooting at locations and with the continual print out nature of it all (like the old PDP-writers) that we used to use. I think I found the source code in Basic online (for Star Trek) and I'm going to consider rewriting it in Python if I ever get time. I think I could now.

Instructions:

You really want to run this game in Terminal and I gave you the .py file too. The experience is awful in Jupyter as you have this little window so you can't see the whole progression of the game very well and the matrices all at once.

What I learned:

I learned that objects are OK and I didn't struggle that much once we completed our first homework assignment. This project was good for that as I could create object classes for the boards and the ships and there are 2 instances of them that actually run so it made it easy for me to understand why objects are good. This was similar to the drone example.

The first question my 14 year old asked me was will the computer try and shoot around the ship when you first get a hit? He tried to stump dad of course. Fortunately, I had just coded that and it was one of the challenges. I called it the shot optimizer and I made it miss the first shot which would

always be adjacent to the hit (just like a human guessing) so the computer would not have an unfair advantage of needing fewer shots to sink the ship.

As I played more, I found I can beat the computer it seems most of the time, so the game is still tilted towards the human who has a field of view of where the most productive shots might be placed vs. picking random shots. However, I'm already at 580 lines of code which was at the top of the guidance so didn't want to keep improving. An opportunity for extension would be an object that optimizes the computer's random shot making into areas that cover the most possibilities just like I do when I play. That is probably why I have an advantage still and find ships faster.

I solved a nagging small problem for me and finally learned a simple way to validate input with While True. I have been reinventing that every assignment. Now I have it in a few lines which I'll just use in the future. I also learned more about printing in color as I found I needed to do that to bring visibility to the matrix or it was hard to see the differences in ships, hits, misses and open holes waiting for a peg. You will see my new Class Colors. I also used some timers in this project to pause the progression to give it more of a back and forth feel between players. The experience was just too fast and difficult to keep up with without me slowing the progression down between computer and player.

The whole logic of the game is all in objects.