X-Treme Obstacles: Mini-Machine Learning

Instructions:

So, you’ve decided to train for the X-Treme obstacle course. That’s great! We’re here to help. Although, it’s worth noting that you’re kind of clumsy and not very smart. This may take a while.

Our super-very-intro-beginner courses start with just four obstacles: a hurdle that’s too-high hurdle (-), a pole (|), a ramp (/), and a tunnel (o). You’ll decide the length of your course, and it will look something like this:

```
!
---oo/o/|-/o|-/|/ooo-o-o/-o/o|
```

Your goal is to make it through the obstacle course without messing up, and the exclamation point denotes your location in a given attempt at the course.

You start off at the main menu, where you can choose one of the following options:

1. Start and learn a new game.
2. View your learning model.
3. Try your model on a new game!
4. Get some help (we know you need it)
5. Quit

Option 1:

Let’s learn a new obstacle course. First, we need to create the game by deciding how many obstacles you would like in your game.

Besides knowing which move to make, you also need to manage to stay upright. At each obstacle, you have a chance of falling down in the attempt. Choose your clumsiness factor next!

We believe that you know your physical limits the best, so you can choose how many times you’d like to attempt the obstacle course. Please don’t exhaust yourself. We here at X-Treme are not responsible for any injuries or loss of life, and we recommend that you consult your doctor before engaging in any sort of strenuous activity.

Decide whether you’d like to print the results of each attempt, or only print when the obstacle course is completed or failed after all attempts are exhausted.

Now comes the fun part- learning the game! You don't actually have to do anything here, so just sit back, relax, and let the computer tell you if you succeed or not.

We understand that you have absolutely no idea how to navigate each of these obstacles, so we've narrowed it down to just four options for you: under, around, over, and through.

On your first attempt, you'll have an equal (25%) chance of choosing each option. If you choose the correct move, you will advance to the next obstacle, and your chances of choosing that option for that type of obstacle will go up a bit. You'd think you'd only have to learn it once, but you're a very slow learner, apparently.

If you choose the incorrect move, you'll have to start the entire obstacle course over from the beginning, but your chances of selecting that move for that obstacle type again will go down, while your chances of selecting other moves will go up.

Well, good luck. Hope you can manage to figure it out!

Option 2:

Choose this option to check out the chances of choosing each move for each obstacle. It can be fun to see this before and after running the game!

Option 3:

This option will only work after you've run option one. This creates a new game (you get to choose your length and # of attempts again, but NOT your clumsiness) and applies the learning model that we created in option one to it. You should be able to complete this one in less attempts than the first one because the model has already been created. Clumsiness could keep you from solving it on your first try, though.

Option 4:

Choose this option for a little help spiel! It's the same stuff that's written at the top of this page, though, so it won't be that exciting.

Option 5:

Once you're bored, choose this option to go do something else with your life.

What I did/didn't complete:

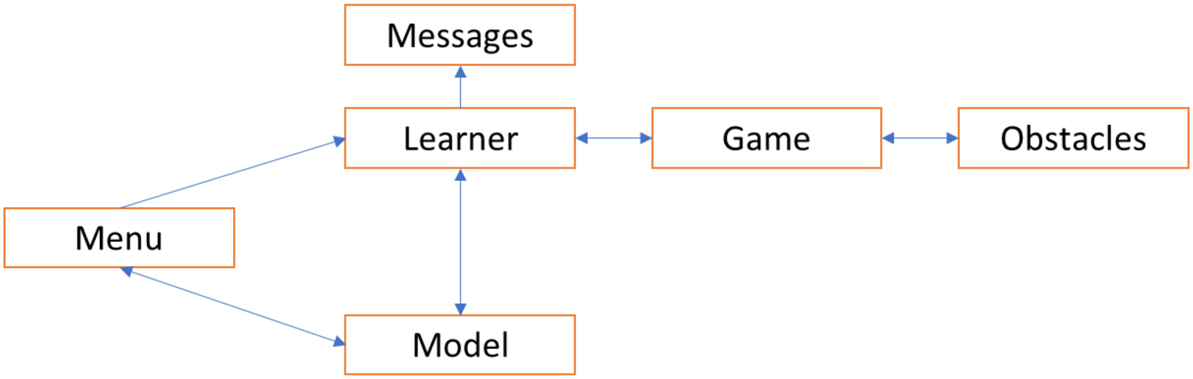
Besides switching projects halfway through because I couldn't figure out how to use a tree graph correctly, I more or less got everything into this code that I planned to for the project.

If I had unlimited time, though I would have liked to build a more complex learning model. I would have had obstacles randomly generate with random moves assigned to them each time, rather than just have 4. I also may have made the previous obstacle affect the next obstacle. For example, maybe after the ramp, it makes more sense to just jump over a tunnel, rather than just go through it. Or, maybe we add a time element so that you can complete the course by either going through or over that tunnel, but going over it would be faster.

Reflection:

This project was great! I had a tough time wrapping my head around OOP at first, especially when it came to the inheritance part of it. I forced myself to use inheritance in the project to make sure that I figured it out. Now, I have a much better grasp on it, and I can definitely see how it could be beneficial for large projects. It was nice to be able to chunk my work this way. I started with the creation of the game and was able to test it independently, which was nice. Then I moved onto creating the learner. Then I decided to break them up into further classes, so my 'game' class was split into Game, Obstacles, and the 4 Obstacles child classes. My 'learner' class was split up into Learner, Model, and Messages. The last piece that I build was the menu to run it all.

Splitting it up into smaller and smaller classes and functions allowed me to get a better grip of how classes pass information back and forth to each other. See the flow diagram below for my project:


Messages
Learner
Game
Obstacles
Menu
Model