Chenlin Ye
W200 Wednesday 6:30PM

Project 1 Reflections – Slot Machine Game

First of all, I have to say that this project development experience had been challenging yet rewarding. When I was watching the async videos on object-oriented programming (OOP), I thought everything seems straightforward. It was not until I really started developing an actual project myself, I realized how much I did not know and how powerful OOP is.

I have always been intrigued by slot machines and how casinos make money yet still giving out huge jackpots by playing the game of probability. Although I am not able to design the same kind of random number generation algorithm that real slot machines use at the moment, I got to have some fun playing with probability while sharpen my skills in OOP. I started of with four simple classes (player, account, slot machine, and casino), wrote them down on a piece of paper with associated actions and attributes of each, and started to sketch out how they interact in a real-world scenario. From there, I started building the classes in Jupyter notebook and the iterative process of programming. In the end, I dropped the account class as it was not necessary for the game design.

One of the challenges I faced was to make classes interact with each other. Although it flows naturally now, but the idea of passing a class instance to another class instance as argument/input seemed abstract to me at the beginning because when I thought about arguments, I thought about static objects such as numbers, lists, text strings. It was difficult for me to imagine passing something that is “alive” as arguments to another object. But eventually I realized that since everything in Python is an object, passing a class instance that’s defined and initiated by myself is the same thing as passing a number as an argument. After going through various online tutorials and studying past projects, I finally understood how an instance of a class can be initiated inside a different class and be called upon using the self.instancename.method/attribute method just like calling that instance from its own class.

One other thing that popped up during debugging was that the input function always returns a text string, even when a number is entered. In order to compare the user entry with the dictionary items, either the input needs to be “integerized” or the dictionary keys need to be “stringized”. Lastly, documenting all the classes and their attributes/methods on a piece of paper really helped streamline the debugging process, especially when edits needs to be made to some elements in the code. Writing everything down on a high-level also helps the logic flow when designing class interactions.

I know that OOP is a deep field and I only know a piece of an iceberg at this point, but I am still impressed by its power. Encapsulation makes each class instance do what it supposed to do only. When changes need to be made, we can always make them without messing up other classes. For instance, the spin_wheel method only spins wheels and no need to worry about determining payouts; the player class instances only track balances, deposits, and withdraws; then the casino class manage all the interactions and make sure the logic flows. When we need to change how the payout is determined, we can easily do that under the pay_out method in the slot machine class and the whole algorithm will keep flowing. I am excited to know more about OOP and dig deeper as I move along my data science career at UC Berkeley and beyond.

Chenlin Ye
W200 Wednesday 6:30PM

In order to run the game, simply run the Ye_Project_1.py file in command prompt or in Jupyter notebook. Upon initiation, it will ask for your username and passcode. Please enter anyone in the W200 Wednesday 6:30PM session with passcode 1234 to proceed (i.e. “Chenlin Ye” for username and 1234 for passcode). After that, the menu options should be straightforward to follow along. One interesting thing I noticed was that the time.sleep(x) method results in different pause intervals on different platforms and programs for the same x number. For instance, time.sleep (0.5) pauses for a longer period of time on Windows than on Mac using Jupyter notebook. Furthermore, the animation feeling of reel spinning does not show up when executing the game in the command prompt. Other than that, I was excited to be able to apply some of the Unicode lessons we learned last week on the spin_wheels () method. To close, I just wanted to thank Gerry and all the instructors who put this project (and the course in general) together.

Have fun!