Project 1 Reflections

7/10/19

Instructions for Running Code:

- Run tu_veux_manger.py from the command line
- Enter username (8 characters or less)
- Enter character (either 'Charlie' or 'Caline')
- To end game, exit window

Completed Elements of Project:

In looking back at my original project design document, many elements of the project unfolded the way I anticipated, although I did have to make adjustments to my design as I learned more about the nature of the python arcade library that I was using. I used classes for the objects that I envisioned at the outset, although the nature of the classes was slightly different than I originally planned. The game engine was the major class of the program as I expected. The dogs are also created as instances of their own class. Instead of having separate classes for the dog treats and dog catchers, these objects are both created from the same classes of "falling objects". I did need to create separate "falling object" classes for each level, however, to account for the changing speed in each level (I imagine a single class for each could be used for all falling object with a method in the game engine to change the speed variable but I was not able to figure this out).

I had imagined that a large portion of my work might center around trying to determine how to build methods for object movement and detection of collisions. Happily, as I learned about the arcade library it turned out these problems were easily solved by built in methods. This left me with time to create additional details in the game, such as multiple levels, different "game states" (instruction screen, between level screens and game over screen), a way to record high scores and I was able to include both of my dogs in the game. Those challenges (while still keeping the game stable) ended up taking the majority of my time.

Uncompleted Elements:

Some aspects of the game I didn't get to that I would like to add:

- **Additional falling objects:** I would like to add other falling objects that would have different effects on the user. For example, I might add a falling chicken wing that would add 3 points to the user score.
- **More elegant user input and exit to game:** I would like to make the user input built into the gui instead of at the command line. From what I have seen, the arcade library does not support this. Also, the only way I could figure out to allow user to exit the game is by exiting the window. This sometimes causes the code to crash and produces an error message. I was not able to figure out how to get around this.
- **Highlight active user on high scores:** I would like the current user to be highlighted in a different color when they are added. This would likely be an easy addition but I didn't have time.

Challenges:

- Implementing the different “game states” and integrating those with the different levels was a challenge. I found myself losing track of which level methods were being called under which conditions. This was a lesson for me in staying organized and cognizant of code flow.
- Implementing the high scores took me some time. I ran into some difficulties with global vs relative variables when modifying the external module.
- The extra time needed to learn the ins and outs of the arcade library was a bit of a challenge. After getting down the basics, however, I found it a fun library to work with and enjoyed the process. It was good practice in working with an existing library and adding the necessary elements to create my desired program.