Exercise 7: Progress Report

Sharon Ku

Progress made since proposal stage:

Since the proposal stage, I switched topics for my project. Before, I wanted to make a game where a frog needed to defeat flies, but I realized that the idea was too complicated because it involved creating a lot of sprites by hand, and calculating a lot of the positions within the sprite (e.g. tongue position of an image) to write code.

I decided to switch my topic to expand on what I had for project 1, which is a simulation where the player had to feed a fish. I made progress by organizing my code into separate classes to allow for modularity and reuse. I transferred my existing code from Project 1 into approximately 20 new classes, which proved to be a challenging feat since I had a lot of interacting objects.

I created numerous chains of inheritance: for my buttons, for my fish, and for my food trackers. It was tricky because I had to figure out which child in the chain was responsible for which method. For instance, I wanted to incorporate random movement for my button, but the text inside the button had to keep the same position as the button's circular shape, so I had to be careful in choosing where to assign my methods.

I added a more complex behavior to the buttons when the finger is hovering over it. Before, the button only grew bigger, but now, it enlarges and shrinks (similar to a pulsing behavior).

I also successfully added a new baby fish to my tank: a neon goby! It comes with its own food tracker. It was difficult to adopt this new fish because I had to create a new foodTracker array and a new fishes array to take care of all my fish and the food they ate. However, since my food tracker and fish interacted with a lot of objects in the tank, I was extremely careful with how I proceeded (although I still had to do a lot of debugging). In particular, I struggled with making the food tracker update based on the specific fish that has eaten the food, but I pulled through in the end.

Plans for future work:

I plan on incorporating interactivity in the Instructions state by adding more animated elements and featuring a "page" for each rule so that the user can familiarize themselves with the actions. Currently, the Instructions state is simply a static image that displays the rules. The challenge here is that I need to find a way to clearly communicate each instruction and to keep the interactions simple and to the point.

I want to change the appearance of my "More Food" button (maybe feature an image with food pebbles floating around it?).

I plan on adding two more creatures in the tank. I will have four fishes (two clown fishes, one firefish, and one neon goby), corals, and two snails. The challenge will be to code the behavior of different fishes and creatures. I will need to use more conditionals and deal with interacting objects.

- The fishes and creatures interact with the user's finger and with each other.
- Clownfishes interact with the sea anemone by feeding it.
- Neon goby interacts with the sea anemone by getting too close to it and getting stung.
- Snails interact with the tank's background (climb on rocks) and with each other (sometimes they travel on top of each other).
- Neon goby is most attracted to the food so it gets to the food the fastest.
- Firefish easily gets scared and hides behind the rocks.

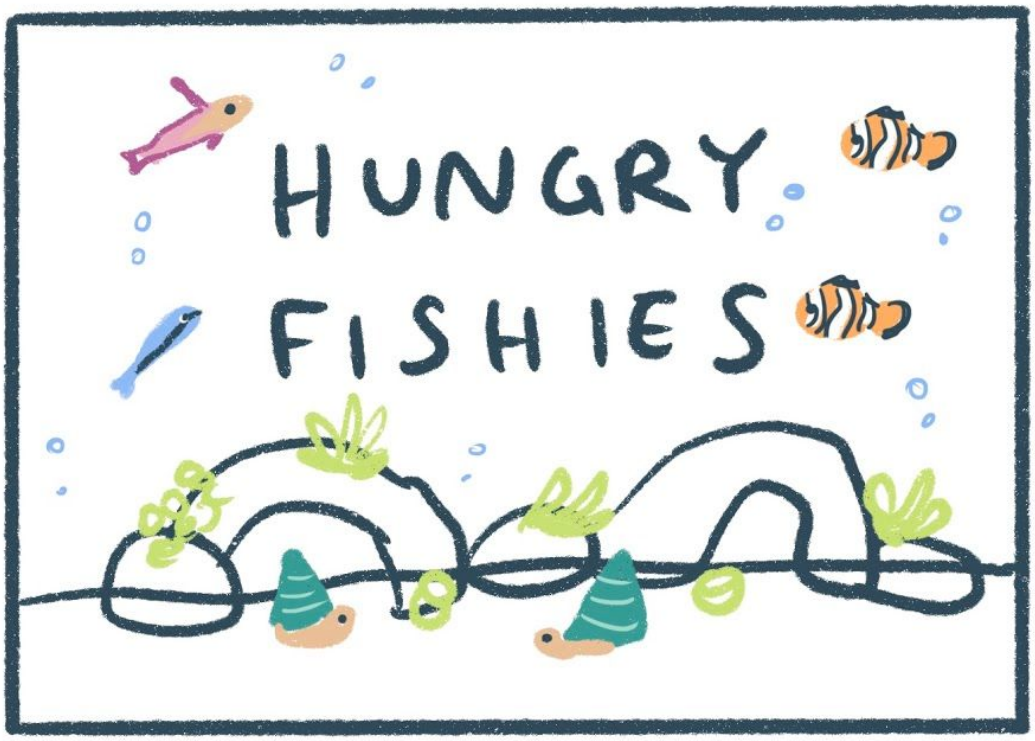

Figure 1: sketch of all the new creatures I will add for project 2

Finally, I plan on changing the way the current flows and carries the food. Instead of changing the current with left and right arrow keys, the user will use the up and down arrow keys to increase and decrease the flow of water inside the tank. When there is a lot of flow, the current pushes the food from right to left, i.e. apply negative x acceleration to food. When there is no flow, the food flows downwards in a "natural" way.