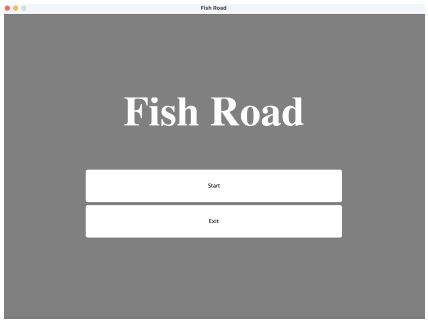
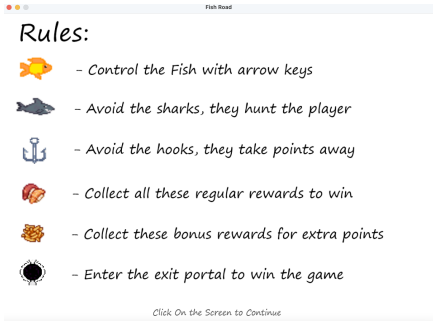
Phase 4 Report

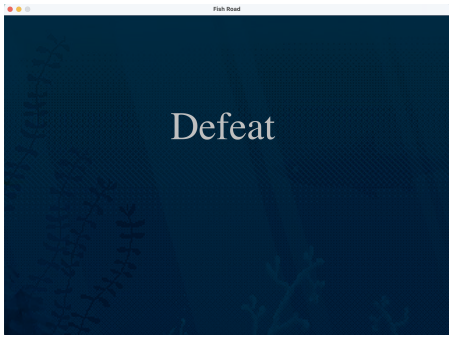
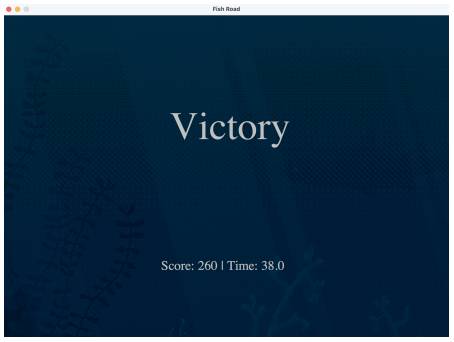
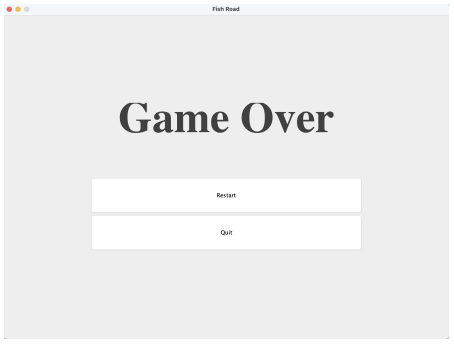
CMPT 276
Group 9

Overall Description of the Game:

The game is called “Fish Road”, and the main theme is underwater. The main character is the Nemo fish whose main job is to dodge hooks and avoid a shark, while collecting all regular rewards (sushi) to open up an escape portal and safely end the game. Users can control the main character using the arrow direction keys. Also, the game has point and timer systems where the player gets points for either collecting a regular award or a special reward, which spawns randomly until collected. On the other hand, the player can lose points for touching any of the hooks placed throughout the map. It is also important to note that the shark constantly tracks the player and the game will immediately end if the player gets caught by the shark.

Tutorial of Main Features of the Game:

	First screen displayed when the game is executed. It includes the title of the game along with the Start and Exit buttons for the player to choose.
	Once the player selects the Start button, the game shows brief explanations on all the items in the game.

	The screenshot of the gameplay. The shark moves closer to the player as soon as the player makes the first movement. The goal of the player is to collect all the sushi, the regular reward, and exit through the portal, which shows up after all the sushi are collected. The bonus reward shows at a random place.
	The defeat screen when the player loses either by losing points below 0 or getting caught by the shark.
	The victory screen will show up after the player collects all the sushi and reaches the escape portal.
	The ending screen will display after the defeat screen or the victory screen. It includes the Game Over sign along with the Restart and Quit buttons for the player to choose.

Plans and Designs Remained and Changed:

We made the very first blueprint of our game in the beginning of phase one where we spent a lot of time identifying different use cases and the overall design of the game. However, during the second and third phases of the project, we realized that we needed to make small changes to our original design in order to implement the game. One of the changes we made was to create an additional class in order to implement the JFrame class, which later on served as our game window. Also, we made a change to the class that generated the game map. The new class stored the coordinates of the objects on the screen in 2D arrays. We also had to do minor adjustments to some of the utilized packages. For instance, by using java.awt.event.KeyListener, we had to add an additional class that was not included in our original UML diagram. Moreover, after some discussions that we had as a group, it was decided to change some of the use cases regarding the user and enemy interactions. However, we tried our best to stick to our original plan and stay faithful to our design from phase one.

Most important lessons learned:

Throughout the development of our game, we have realized important lessons on how planning, communication, collaboration, and testing are all crucial in the joint development of a game. When it came to planning, we initially did not spend as much time on the UML diagram compared to the mockup drawings. This resulted in us being unsure about what to do during the implementation phase for the logic of our game and often finding the need to deviate from our initial UML diagram designs. However, when it came to the placement of objects and looks of our game, we found this quite simple since we already had well illustrated mockups of what we envisioned our game to look like. For communication, we learned about how critical effective communication was in developing a successful game. This included constant in-person and online meetings to ensure everyone was on the same page, and to help each other out. This continual stream of communication was what helped ensure collaboration within our group. It was through this collaboration where feedback was provided, ideas were shared, and problems were solved. Finally, the most important lesson we learned was the value of testing. Whether it be manually testing new features, or writing unit tests and integration tests, the importance of tests in development cannot be understated. For example, when modularizing the code for the start and end screen, we suddenly had issues where buttons would suddenly appear on the screen when the user’s cursor went over specific portions of the game window. If it was not for testing, there would still be many faults and errors remaining in our game.