Snake Game Final Report

Joey Luyben
David St. Pierre
Collin Crow
William Wilkins

The software is a basic snake game, with wonderful background art and plenty of customization available. When starting the game, the user can either start their game with the default settings or go to the settings and change things like the snake speed, color, and size. Once the user has played a round of the game, they can enter a name and that name will be stored into a local database along with every other player and score and the top 5 will be shown under the leaderboard section of the main menu. The target audience is anyone looking to kill a bit of time, maybe have some friendly competition around a public computer space with some friends. This game isn't high enough level to really intrigue anyone who is serious about competitive gaming, but can distract someone for a few minutes.

The easiest thing to implement was the database integration. It was one of the last things i integrated so I knew exactly what methods I needed to make and when to call them. I was able to used the code given in lecture and slightly modify it to get a desired result. When it came to getting top 5 scores in the database, I was easily able to find code online and modify it as well.

The most difficult component of the project was the implementation of the Snake mechanics. While numerous components of the mechanics proved to be uncooperative, the most apparent issue that arose was the inability of the game itself to show when prompted in the Main program. While the original implementation of the game worked as a separate program, it did not translate well into the use of controllers. It was found the main cause of this was the snake itself. The ObservableList class was used to initialize the snake, but there were numerous errors with creating and adding components of it to the game window. Attempts to fix this included implementing the snake with an ArrayList, altering the method in which the animation was created, and defining a new class to create the game components. In the end the solution was creating a new class called "Entity" which acted as the mechanism to initialize the snake and food, as well as add new components to the snake.

The thing I would have done differently is use java scene builder from the start to make the gui. I started with just hard coding every line and making basically the same functionality as the final interface, but it didn't look as nice as the final product which was made with scene builder and had everything lined up and spaced much nicer. However, doing this GUI I learned basically everything I know about using javafx and would have a much easier time making a GUI in the future with this method and probably other assuming there's a decent amount of crossover to other methods.