CIS 452 01 – Assignment 9 Reflection

Jake Buri

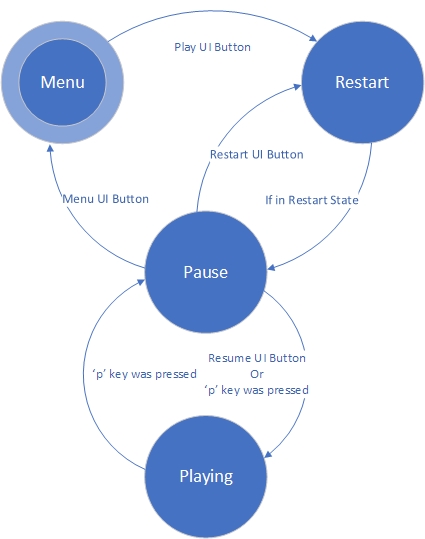
Starting Scene: Game

GitHub Link: https://github.com/jburi/CIS\_452\_Assignment\_9

Simmer.io Link: <https://simmer.io/@jburi/rhythm>

1. What is the internal state or process you are trying to control with the State design pattern?
   * I am trying to control whether the game is at the main menu, running, paused, or restarting.
2. What triggers changes between the states?
   * Booleans and if statements controlled by a UI button or current state.
3. What were the benefits of using the State Pattern to make your mini-game?
   * I found this pattern extremely useful and versatile. By using states to control UI elements and pausing the game allowed me to create everything in one scene. I also enjoy the process of using this and found that it helped redundancy. My favorite part however, is how it forces you to throw exceptions and errors which all good code should have.
4. Did you find any drawbacks to using the State Pattern? If so, what were they?
   * I sometimes found it difficult to track the process of how the code ran since it was switching between so many scripts and calling functions from other scripts. Commenting my code and drawing a state diagram helped a lot with this though.
5. What is the player’s goal in your mini-game and what makes it challenging?
   * The goal is to hit every note and keep a streak to gain points. What makes it challenging is the amount of notes you can miss before you fail.
6. How does the game communicate its goal(s) to the player?
   * The high score is displayed on the home screen as well as in the description I wrote that the goal is to full combo the song.
7. How can the player fail at the game and how does the game detect it?
   * The game has a counter to show how many notes you can miss in a row before you lose.
8. How does the game give players feedback about how well they are doing?
   * The game displays a score and multiplier that updates with each note.

**State Diagram**



**UML Diagram**

