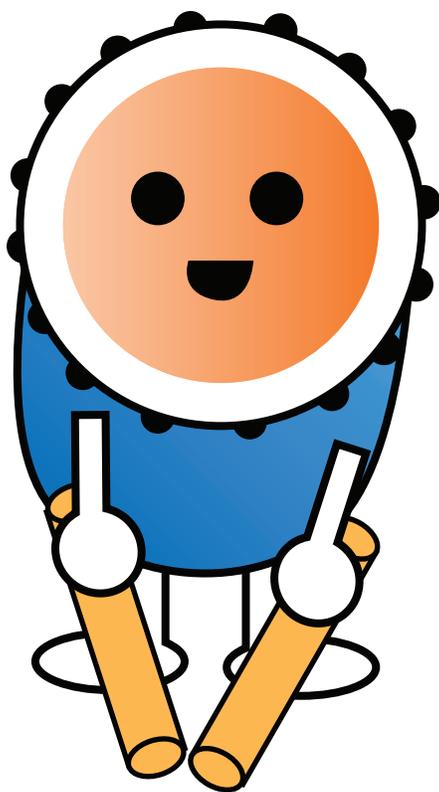


Final Project Report

Taiko Master



Marie Cheung

301090275

IAT 265 - D103

Taiko Master

Overview

The game is an interactive rhythm game where players must hit moving beats with the drum icon. Its concept is similar to the popular Dance Dance Revolution game and the inspiration of the game is from the Japanese arcade game called Taiko no Tatsujin.

Beats are created using the SmartEllipse into circle shapes and the getMethod helps find coordinates to set the location to move the beats. Multiple beats are created and held inside an ArrayList, and if there are indexes in the list where there are no beats, it will shift indexes to make sure each index carries a beat. Beats will move left on a running timeline, and the player must press the spacebar on the keyboard in order to hit the beats.

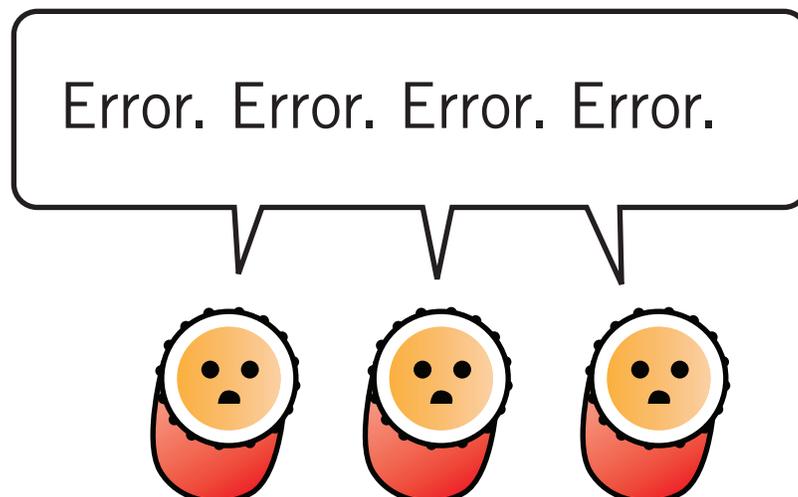
Using the paintComponent function, the score is drawn inside a speech bubble on the top right corner of the screen and acts as the score counter for the number of beats hit correctly. Players must achieve at least 2000 points before the game ends in order to win the game. If players fail to achieve 2000 points before the beats run out, then the game will end. However, in both cases, players can restart the game by pressing the letter R on the keyboard only when the game has ended. If the spacebar is hit when there are no beats under it, or when beats are not hit on time, 10 points will subtract from the total score.

Taiko Master

Technical Challenges

One of the major technical challenges involved with creating this game was to create beats that would move left along the timeline, but spaced apart. When I first created them, they were really close together and I could not figure out how to solve this problem. I tried setting the location of the beats outside of the screen/window, but the beats just all bunched up together as they moved. Thus, rather than separating them by setting their locations, I learned how to implement the built-in Java `System.currentTimeMillis()` function to create my beats through the use of time in milliseconds. The beats would be created according to when x number of milliseconds had passed since the game had started.

Another major challenge in this project was when I initially created the drum icon and used the intersects function to detect collisions because the score system would act weirdly when I hit the spacebar. 10 points should be added to the total score each time a beat was correctly hit. However, sometimes the score would add 20 from the total score when I only hit the drum once with the spacebar. To fix this problem, I implemented `keyPause`, which helps decrease the key rate, so that each time after the spacebar is hit, 80 frames would pass before the spacebar would add any points when it hit beats correctly. Thus, there would be a small gap between when the spacebar is pressed and released, so that points would not be duplicated.

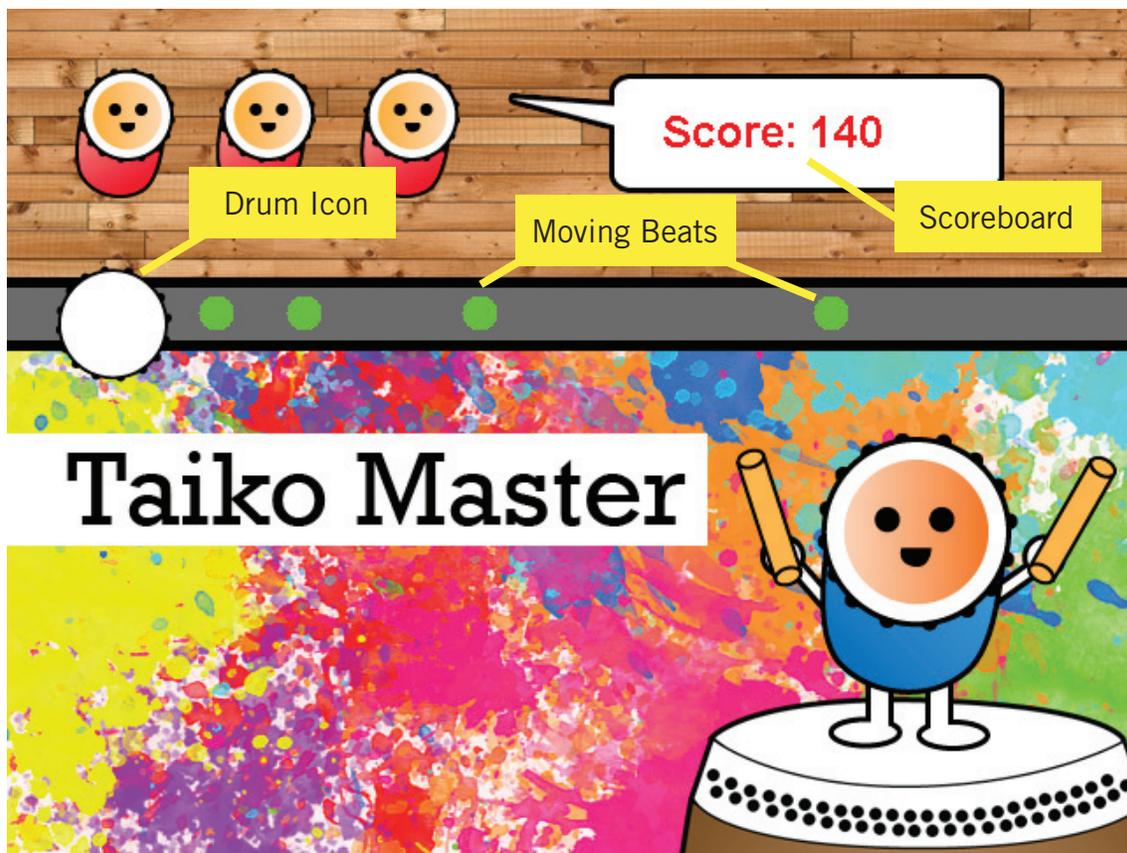


Taiko Master

User Manual

The game is an interactive rhythm game where players must hit moving beats with the drum icon. Its concept is similar to the popular Dance Dance Revolution game and the inspiration of the game is from the Japanese arcade game called Taiko no Tatsujin. The beats will move left on a running timeline, and the player must hit the spacebar on the keyboard in order to hit the beats. The speech bubble on the top right corner of the screen is the score counter for the number of beats hit correctly. Players must achieve at least 2000 points before the beats run out in order to win the game. If the spacebar is hit when there are no beats under it, or when beats are not hit on time, 10 points will subtract from the total score.

To run the game in Eclipse, you must hit Ctrl + F11. The game will start immediately as the song will start playing as soon as the game is run. To hit the beats, press the spacebar on the keyboard. After the game has ended, press the letter R on the keyboard to restart the game.



Taiko Master

Conclusion

All in all, my goals to follow my proposal and vision of creating a Taiko beat game, and to develop an actual working game were successful. This project was highly challenging since I had never created a game before. The class labs were helpful and similar to some extent, but a lot of resources and references outside of class had to be found in the creation of this game. However, I believe I now have a better understanding of the concepts of methods, arrays, constants, timers, and game development in general after I completed this project. Although I had struggled with trying to modify the pong game to develop my game in the beginning, I then learnt how to utilize some aspects of that lab and not to fully rely on it.

If time permits, I would like to fix the ending of the game. Currently, when a player wins, the timer stops and pauses the game (except for the song), even when there are beats still on the screen. However, as soon as the player hits the letter R to restart the game, the paused beats carry on moving instead of disappearing and actually restarting the game, and only the timeline is restarted/new beats appear as the old beats move off the screen.

