Eugene Choi
Dec. 12, 2018
CSE40166

Ham Ham Revolution (Rhythm Game): Final Project Report

Project Description

Ham Ham Revolution is a rhythm game implemented with three.js. It has a 3D blender model of Basil the hamster imported which the player can control with the up, down, left, right arrows and the space bar. The game control screen (gray box on top left corner) shows the player what button to press. As of now, the game has a 30 second sample of “Humble” by Kendrick Lamar and its beats hardcoded. Moving forward with the project, user input for music and dynamic generation of the rhythm is planned to be implemented using a python library called librosa.

Main Functions

1. Initialization (init()):
    a. In this function, the scene is set up using three.js. The model is also imported and the setUpAnimations function is called as well as initGame.
2. Set Up Animation (setUpAnimations()):
    a. In this function, animations are connected with the button handler and their respective buttons.
3. Game Initialization (initGame()):
    a. In this function, the beats and the corresponding moves are set up. The beats are hardcoded as of now, and the moves are generated randomly. The button

handler for the start button is added. The beats for the music had to be edited because the sample music used was a fast song.

4. Game Start (startGame()):
    a. This function sets up audio and the game control screen for the game once the player has clicked the start button. It resets the clock and the score.
5. Game End (endGame()):
    a. This function displays the score and the corresponding message when the game ends.
6. Animate (animate()):
    a. This function updates the animation with the clock. It also handles the gameplay if the start button has been pressed. It matches up the beat timestamps and the clock; if they match within a margin, then it is checked if the player has pressed a button. If he or she has, the move is checked. If it is the correct move, the player receives a full point; otherwise, only half a point is given.

Usage

When the player opens “hamster.html”, they are taken to the initial screen which shows Basil the Hamster in resting animation. The player can still control the movements, but the game has not yet started. The game begins when the player clicks “start” button on the game control. The game control will show the player which button to press at what time.

1. Space bar: “Space” - Basil jumps
2. ^: “Up arrow” - Basil looks up
3. v: “Down arrow” - Basil looks down
4. ← : “Left arrow” - Basil twists left
5. → : “Right arrow” - Basil twists right

When the song finishes, the game control screen will display the player’s results (and a comment about their performance).

Results

Challenges

One of the biggest challenges was learning to use Blender, which I realize is not part of this class. It was fun but also frustrating to learn all the appropriate terms used with 3D modelling and the functionalities. I worked from scratch, building the 3D model to adding texture to appropriate UV map assignments, and animating the model.

Another big challenge was figuring out the synchronization of the player's button directions with the timestamps and the music. The example I used at first was difficult because it was a faster song. However, eventually I was able to match all of the timing including taking the player's response.

Assessment of Efforts

I worked alone on this project and thought I was a pretty collaborative team member.

What Was Learned

In addition to learning to use Blender from scratch, I learned how to export the model and import it to three.js. There was a steep learning curve with exporting to GTLF format from Blender correctly, but a simple process once learned. It was as simple as using a loader already provided with three.js to import the model. I also learned to control the model in three.js and to set up the appropriate environment on screen for the model.