System and Unit Test Report
TokyoDriftr
Luke Harvey, Conrad Menchine, Kevin Pinney

For our testing, we tested the entire game at once by playtesting. Our testing methods were mostly based around loading the game, driving around for a moment, and seeing if anything broke. Below are formalized versions of our testing methodologies for each sprint

Sprint 1:

A “As a player, I want the ability to control my car”
B “As a player, I want my ability to control the car to matter when placed inside the world”
C “As a player, I want to open the website, understand the controls, and start playing quickly”
D “As a player, I want a new unique map to try to compete on a regular basis”
Scenario:

1. Load the webpage. Ensure that the instructions are visible
2. Press the “1” key to start the game
3. Use the WASD keys to drive around the track. Attempt to drift.
4. Before reaching the end of the course, drive into a wall in order to test that the collision is working
5. Refresh the page. See if the course is different than it was before

E “As a user, I would like to know how my performance compares to others”
This User story was not completed during the sprint, and there was no test for it

Sprint 2:
A “As a user, I want the road to have no artifacts as well as have collisions to be accurate to the model so that roads look clean and the gameplay reflects what I see.”
B “As a user, I want the game client to recognize when I have completed the race and to display, record and compare my time in order for my actions in game to have meaning.”
C “As a user, I want the game to have audio to indicate in-game events and add atmosphere”

Scenario:

1. Load the webpage. Ensure that the instructions are visible
2. Press the “1” key to start the game
3. Use the WASD keys to drive around the track. Attempt to drift.
4. Keep an eye on the walls of the course, ensure that there is no clipping or breaking anywhere
5. Before reaching the end of the course, drive into a wall in order to test that the collision is working
6. Cross the finish line, see if the game reacts to this event.

7. Turn up your computer volume and see if there is music or car sounds.

D “As a developer, I want development tools in-game in order to speed up the testing and development process.”

This User story was not completed during the sprint, and there was no test for it

Sprint 3

A “As a player I want the controls to feel intuitive and nice”

B “As a player I want the main menu to be clear about how each car functions”

C “As a player I want to be able to see terrain and visuals outside of just the road, so that I feel more immersed in the gameplay”

Scenario:

1. Load up the webpage, mouse over each of the cars and ensure that the stats for each car is being displayed
2. Click on a car to start the game
3. Drive around. Make sure that the “feeling” of the car matches up with the stats given on the menu. In general:
    a. The RX7 should be the fastest but handle badly
    b. The ae86 should be the slowest, but handle well and have a large boost for drifting
    c. The Civic should handle very well and be fast, but not drift at all.
4. Ensure that there are decorations visible on the sides of the course
5. Repeat this process with each car, note if any of the cars feel significantly better or worse than the others.

Unit Testing:

We didn’t create any automated testing during the course of this project, but we did specific steps on each module each time that it was modified:

Collision Physics:

1. Load the game and select a car
2. Drive around the course and attempt to drive into the walls.
3. Try:
    a. Driving into the walls head-on
    b. Glancing off at an angle
    c. Drifting into a wall

Road generator:

1. Load the game and select a car
2. Drive around the course and look for overlapping bits of road or awkward turn angles
3. Repeat this process about ten times

Sound Testing:

1. Load the game and before everything is loaded try out each slider and see its effect
2. Drive around and make sure that the engine noise changes with the car's speed
3. On each game transition change sliders and listen for if it actually works and if the game breaks as you move sliders

GameState loading:

1. After gamestate transition check to see if every object inside of this.objects was instantiated eventually
2. Check that sound isn't null and that sound eventually is sound.isPlaying