**Game play**

Our game has 2 key features:

1. Classic car racing game. Player uses W, A, S, D keys to control a car to go through the whole track and the goal of this game is to finish a round of track within one minute.

2. QTE (Quick Time Event) system. Another important feature of this game is the QTE system. Player needs to press the corresponding key (One of J, K, L) according to certain QTE button occurring in game randomly. If the correct key pressed in time, the player will get a short-time acceleration bonus, which gives the player a chance to achieve better race results, but makes the car harder to manipulate as well.

Therefore, while controlling the car running smoothly along the track, the left and right hands coordinate to complete the QTE to get acceleration bonus will be the fun and challenge of this game.

**Game Models**

Most of our game models are downloaded from the unity asset store. Unity assets store have many mature, excellent, and even free models for game developers to use. For example, our car model is downloaded from unity asset store whose elements including salon, engine, bottom, wheels, even windows can be modified easily.

**Car’s direction control**

Considering the comfort of the left and right hands, we set W, A, S, D to control the car instead of UP, DOWN, LEFT, RIGHT.

When the user presses the direction button, the script will record the direction, and control the front wheel turn to corresponding direction. Our car is rear wheel drive, front wheels are mainly used to controls steering direction.

**Implementation details of acceleration and fire trail**

The acceleration feature is developed based on a PowerUp class, which will add a number to the speed and acceleration of the car.

The fire trail is a cool effect during acceleration. We obtain it from the Unity Assets Store and set it in each wheel.

We write a script to monitor the results of QTE. When player presses a correct key, the script will enable the fire trails and create a powerup object to speed up the car. Then there is a short cooling down time, and the speed of the car will gradually slow down to the normal number.

**Implementation details of QTE**

In this QTE part, we set a button-like image to represent the QTE content in the canvas and the car will speed up if the player pressed the correct button. The default background color of QTE button is blue. If the player did not press the correct button according to the hint on the button or player did not react in 3.5 seconds after the QTE button appears, the background color of button will be red. If the QTE is correct, the background color will turn to green. It is a intelligible mechanism to represent the QTE outcome.

The QTE function is mainly accomplished with two coroutines. The QTE event will be triggered with 3-10s intervals which is controlled by a Random Function. One coroutine counts the seconds the player uses to react to the QTE event. Another coroutine monitors the event of key pressing (GetkeyDown) and help judge if the pressed key is correct.

**Discrepancies**

1. Considering the comfort of the left and right hands, we set W,A,S,D to control the car instead of UP, DOWN, LEFT, RIGHT.

2. We do not implement the drift system, because we think it is complex enough to control the car and complete QTE, a drift system may reduce the playability of the game.