

CS4303 Video Games Practical 3

Revised Design of a Video Game.

Student ID: 080010830

December 16, 2011

Word Count: 1142

Revised Design of Video Game on the Processing/Android platform.

1 Title

The title of my game is “Reysoid”, a combination of the sounds of the words race and android which are respectively the genre and platform of the game.

2 Genre

The genre of my game is a driving game, it can also be considered a racing game although there is only one car on the track.

3 Environment

In this section I will describe the setting for my game and all its features.

3.1 What is the setting for your game?

The setting for my game is a 2D racing/driving track. The track itself is very similar to “zeRace” [1] race track below:

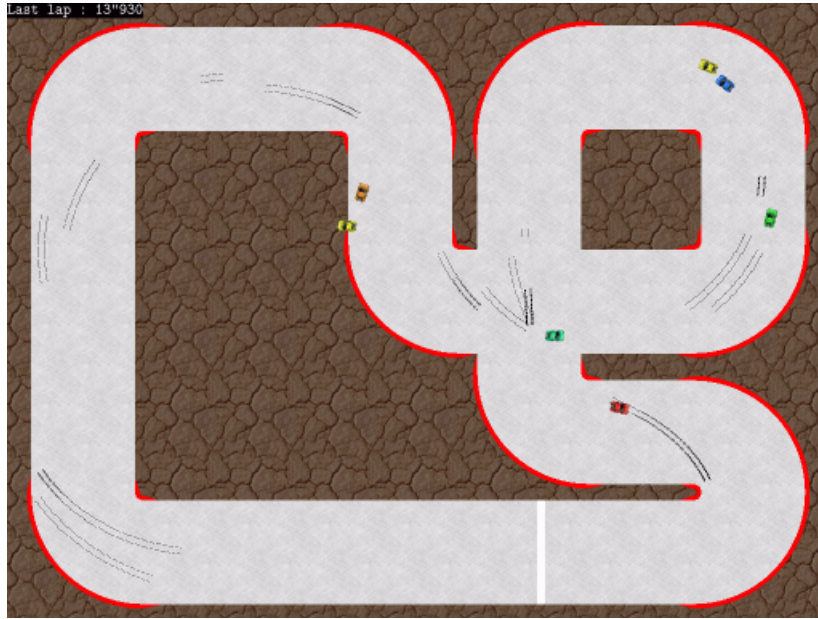


Figure 1: Screenshot of “zeRace” [1]

The setting for my game is thus rather simple compared with most modern racing games for smartphones, its features are described below in more detail.

3.2 What are the features of this environment?

The features of this environment include a track and out of track. The player described on section 4 will be on the track, and will try to avoid leaving it because that will slow down his speed.

3.2.1 Track

The track allows the player to keep a steady speed if the smartphone is tilted frontward as long as he is on the track, as soon as he leaves the track the speed is reduced.

The track shown below was taken from supertuxkart [2].

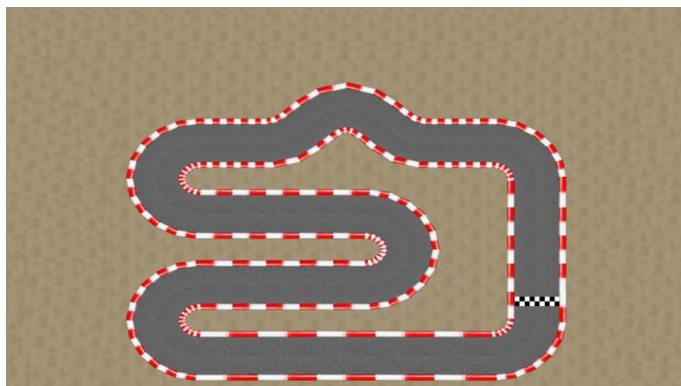


Figure 2: Screenshot of empty track.

3.2.2 Out of Track

The out of track is the part of the environment which is not the track itself. In the case of the player leaving the track the speed of the car will become half as fast as when the player is on the track itself. This feature will force the player to keep the car on the track in order to achieve the fastest speed possible, and with that complete the track in the least amount of time possible.

3.2.3 Information Panel

There is an information panel at the top left, displaying the current inclination of the smartphone in the x axis, the current coordinates of the car, the place of the car - if on track or off track, the state of the car - if accelerating or reversing, the speed of the car and when the player presses on one of the black circles to change direction a text will appear where one can read “Turning Left” if the left circle is pressed or “Turning Right” in the case of the right circle. On the top right the current time in seconds is displayed.

4 Player

The player should feel that he is in control of the car described below in section 4.1. The player goal is to keep the car on track and avoid the out of track (section 3.2.2). The player will “win” the game by beating previous high scores, this is described in more detail in section 7.3.

4.1 How is the player represented in your game?

The player is represented by a Lamborghini Aventador LP 700-4 from a rear top-view perspective. The car image was provided by TinyCars [3].



Figure 3: Top-view of Lamborghini LP 700-4 Aventador [3].

The figure 3 above will be used for the representation of the user. It is a view from the top of Lamborghini LP 700-4 Aventador.

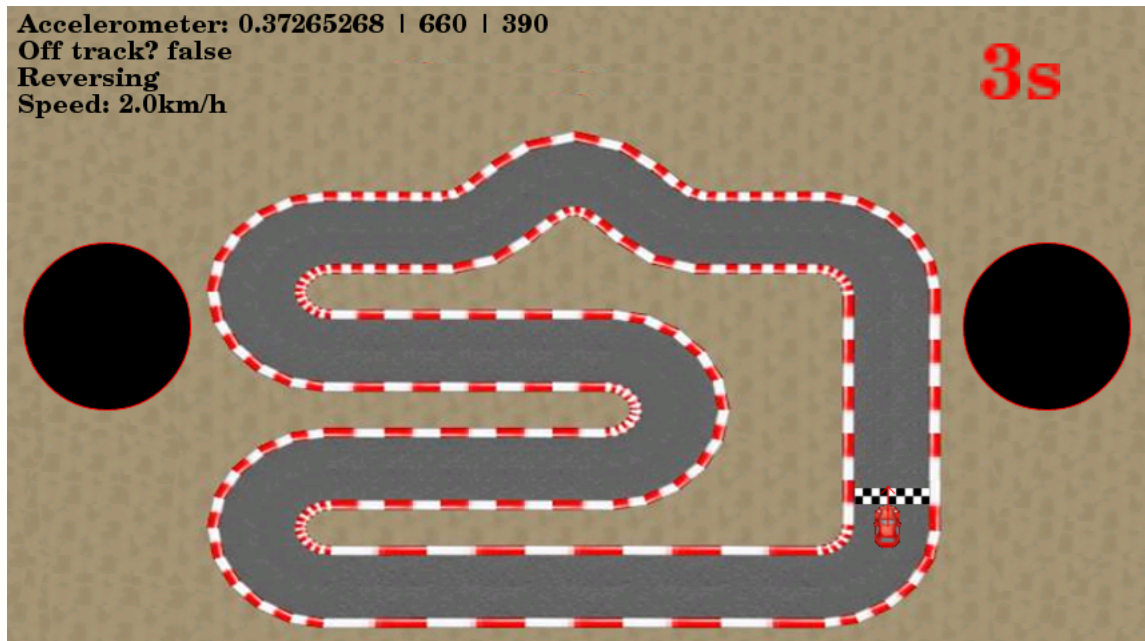


Figure 4: Final environment with the player representation.

The player will see the car with an arrow that points the current direction the car is going.

5 Opponent

There is not really an opponent in my game, however, time can be considered the opponent since the player will try to complete the track in the minimum time possible. Moreover, if the player wants to get a better score, he will have to beat the previous highest scores. The clock is displayed on the top right of the screen, as seen in figure 4.

6 Rules/Mechanics

The rules of my game are fairly simple. The player has to drive on the racing track and finish it as fast as possible.

6.1 What abilities does the player have?

The player has the ability to turn left, turn right, accelerate and reverse the car. These four abilities will allow the player to interact fully with the environment, as described next.

6.2 How does he/she interact with the environment?

The player interacts with the environment by tilting the smartphone, making use of the built-in accelerometer and by clicking on the circles to the left and right of the track. There are two inclinations to produce the acceleration and braking interactions with the environment.

- **Accelerate.** Tilt the smartphone forward, making the x axis decrease.
- **Reverse.** Tilt the smartphone backwards, making the x axis increase.
- **Turn left.** Press the left black circle.
- **Turn Right.** Press the right black circle.

7 Goals

7.1 What is the overall goal of your game?

The overall goal of my game is to complete the track in the least amount of time possible. Furthermore, we also consider the fact that the player can beat previous higher scores to be a secondary goal of the game. Basically, completing the racing track and trying to be faster than the previous higher scores is the goal.

7.2 How does the player progress?

The player does not progress in my game, the racing track is the same and the only factor that can change is the time that takes the player to complete the track. There will not be a next level, due to limitations on the time to implement the game.

7.3 How does the player win the game?

It is not possible to win the game, since it is a racing game and the main goal of the game is to complete the track as soon as possible. However, the player can beat previous higher scores and therefore win against those higher scores. In other words, the player will “win” if its time is higher than the other highest scores.

8 End Game

When the user finishes the track, the screen below is displayed. It shows the finishing time and gives the user the option to play the game again or exit.



Figure 5: Game over screen with options to play again and exit.

8.1 Main game loop

```
check for input
  if smartphone tilted
    if tilted front
      increase speed (e.g. accelerate)
    else if tilted back
      decrease speed (e.g. brake)
  else if click left circle
    turn left
  else if click right circle
    turn right
  else
```

stop

8.2 End game loop

if click play again
 go to main loop
else if click exit
 exit

9 State Diagram

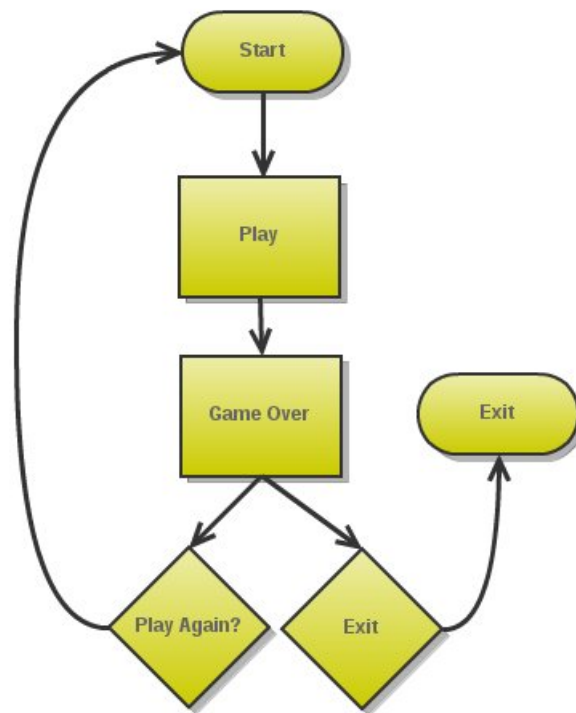


Figure 6: State diagram created using gliffy [4].

10 In Context

Generically, racing videos games date back to the 70s, more precisely 1973, when Space Race [5] was first released by Atari. Space Race is basically a racing game with two players represented as spaceships that race upwards on a monochrome screen and trying to keep away from asteroids that come from the left. The following year, the first driving racing game was released which was named Speed Race [6]. The game Speed Race introduced to the video game industry the notion of collision detection and scrolling graphics, more specifically overhead vertical scrolling [7]. Thus, car games became one of the most popular types of games and a good option for this project.

References

- [1] zeRace. In pictures: Commodore computers. <http://royale.zerezo.com/zerace/>. Accessed October 30, 2011.
- [2] supertuxkart. Classic racetrack. http://supertuxkart.sourceforge.net/Track_design. Accessed December 16, 2011.
- [3] tinycars. Tinycars. <http://www.realore.com/games/tinycars/>. Accessed December 16, 2011.
- [4] Gliffy. Gliffy - state diagram. <http://www.gliffy.com/>. Accessed November 1, 2011.
- [5] The International Arcade Museum. Space race - videogame by atari. http://www.arcade-museum.com/game_detail.php?game_id=9681. Accessed October 30, 2011.
- [6] C. Kohler. Power-up: How japanese video games gave the world an extra life. 2004.
- [7] B. Loguidice and M. Barton. *Vintage games: an insider look at the history of Grand Theft Auto, Super Mario, and the most influential games of all time*. Focal Press, 2009.