T3Dmake

Traffic Racing complete kit

2017

Thank you for purchasing the Traffic Racing complete kit

Contents

Introduction

I made this kit to let you create traffic racing games easily. Add your own models and UI and play the game. Choose your car, buy it and choose one of the locations. Drive endless roads and avoid collisions with the randomly spawned cars.

About this project

I started this project after finishing the 3D city run asset. While I made the 3D city run asset, I learned a lot about endless spawning and scrolling textures. For this project, I just wanted to make a game, that's more fun to play. So I made fast cars, 2 endless worlds, cars that prevent accidents and switch lines and a garage with some unlock able cars. There are lots of traffic racing games on the Google play store already which gave me inspiration. The hardest part of making an endless 3D racing game was to create the illusion of moving forward while actually having a static car. All cars seem to move forward, while there moving backwards, the trees and houses aren't static, they move backwards even faster than the cars and even the distance isn't real distance, it's just calculated using car speed.

General system

To create the illusion of moving forward, everything needs to move the right way with the right speed. Otherwise it would turn out looking pretty weird. For me, one of the hardest things in this system was to have different cars with different high speeds. Because everything is moving accept the car, I needed the car to control everything around it. All together, the system works as follows:

- *Car selected by player gets instantiated*
- *Car gives it's high speed to the game manager*
- *The scrolltexture script finds the game manager*
- *Scrolltexture has a float as well, called 'scrollspeed'*
- *Scrollspeed increases till it reaches game manager highspeed (to create the accelerating effect)*

Then, all ground textures scroll using the scrollspeed of scrolltexture script. Houses, trees, bridges etc. move with scrollspeed * 6, to move as fast as the ground textures. Cars move the same direction as all other objects but slower, to make it look like they move forward and even the time between spawning new objects and cars is based on scrollspeed, to make sure the concentration of cars and objects doesn't depend on the car speed.

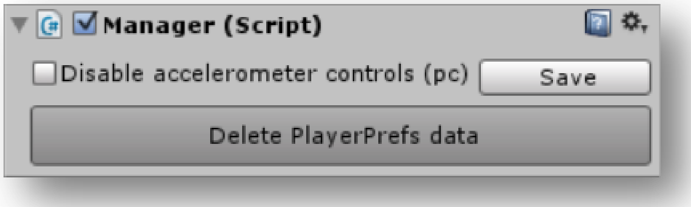
In the comments of for example, the 'EnvironmentSpawner' script, you'll see things like *'based on scrollspeed (car speed)',* that's because it than uses scrollspeed, while scrollspeed is most of the time the same as car speed.

Controls

By default, in the in-game options menu, you'll find two different control types. Touch/click and accelerometer controls. Touch/click works on both mobile devices and pc. Accelerometer only works on mobile. I you are going to build your game for pc, you can go to one of the racing scenes and in the manager script turn on pc only (click 'save'). To make the pc only setting work for all scenes, it uses PlayerPrefs. That's why I recommend first to delete PlayerPrefs, than choose your controls type and then build the game. **By deleting PlayerPrefs, accelerometer controls are enabled automatically**. If you're building for mobile devices, there's nothing to worry about, since both controls are enabled by default.

Resetting the game

Resetting the game (deleting PlayerPrefs to lock all cars and set coins to 0) is very easy. Just go to one of the racing scenes, go to the manager script and press delete PlayerPrefs data. The only thing to keep in mind is that the control type automatically resets too. So if you want to build for pc, you'll need to save the pc only setting again (it might be better to make your game, than delete PlayerPrefs and than setting the controls type).



☑ Manager (Script)

□ Disable accelerometer controls (pc)

Garage

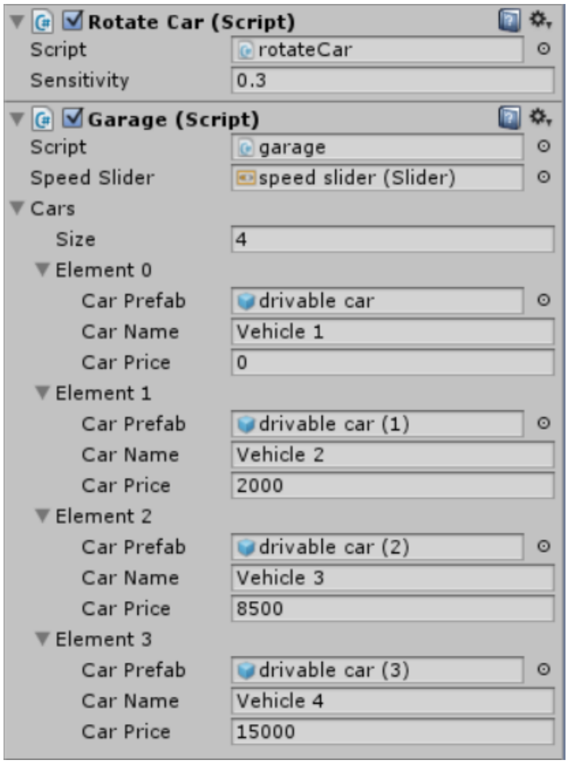
One of the main goals was to create an easy to customize garage. The garage is used to select and unlock new cars. Since coins and new cars are the main goal of the game, the garage is pretty important. The garage scene includes a start screen as well and that's also why the garage script handles the start panel. It's very easy to add your own cars to the garage and set a price for them. Also from the garage scene, you can open the race locations.

Setting up and adding your cars

Using your own cars in the game is not that hard. There are a few steps to set up a car and use it in the garage.

- First, import your car model and drag it into the scene.
- Now, give your car one big box collider and a small one at the front.
- Set the small collider to 'Is Trigger'.
- Add the car controls script and a rigid body. Copy rigid body values from one of the drivable car prefabs.
- Then add the crash audio and turn off 'play on awake'.
- Copy smoke of one of the drivable car prefabs and add it to your car.
- The car is finished now. Go to the platform object and in the garage script, add a new car.
- Make a prefab of your car and add it to the garage script.
- Set a name and price for your car. The name is used to save your car with PlayerPrefs too.
- Finally, add your car prefab to the game manager of **each scene**.

<u>Please note that the order of the cars in your garage and in the scenes should be exactly the same, otherwise the cars don't correspond.</u>



```
▼ ☑ Rotate Car (Script)
  Script              rotateCar
  Sensitivity         0.3
▼ ☑ Garage (Script)
  Script              garage
  Speed Slider        speed slider (Slider)
▼ Cars
    Size              4
  ▼ Element 0
      Car Prefab      drivable car
      Car Name        Vehicle 1
      Car Price       0
  ▼ Element 1
      Car Prefab      drivable car (1)
      Car Name        Vehicle 2
      Car Price       2000
  ▼ Element 2
      Car Prefab      drivable car (2)
      Car Name        Vehicle 3
      Car Price       8500
  ▼ Element 3
      Car Prefab      drivable car (3)
      Car Name        Vehicle 4
      Car Price       15000
```

Viewing your cars

To let your players have a good impression of the car they are going to unlock and to show your cars, I added the possibility of rotating cars in the garage by dragging your mouse (or finger). This is done by writing a script, called 'rotateCar' and adding it to the platform. Every instantiated car gets instantiated as a child of the platform, so they rotate together with the platform.

Line switching cars

For a long time, I doubt about implementing line switching cars (changing track), because I thought it was going to be way to hard. Actually, with some raycasting, it was quite doable. I think line switching makes the game way more interesting and a bit more difficult as well. Only the smaller cars switch lines, to prevent accidents.

Setting up your own line switching cars

To set up your own line switching cars, there are a few simple steps.

- Import your car model and add it to the scene. Give it a move object script and check car, change track and set a slow down distance. (Distance between car and the car in front of this car. I called it slow distance because it looks like the car slows down, while it actually moves faster…) If you have a normal size car, you should set this distance between 6 and 8, if you have a large vehicle like a bus, it should be around 10.
- Now, add a box collider and a rigid body and copy rigid body values of one of the 'simple vehicles' prefabs.
- Finally, copy the left/right lights from one of the 'simple vehicles' prefabs and make sure you name them exactly the same.

After you've set up your car, add it to the car spawner script in both scenes

Preventing accidents

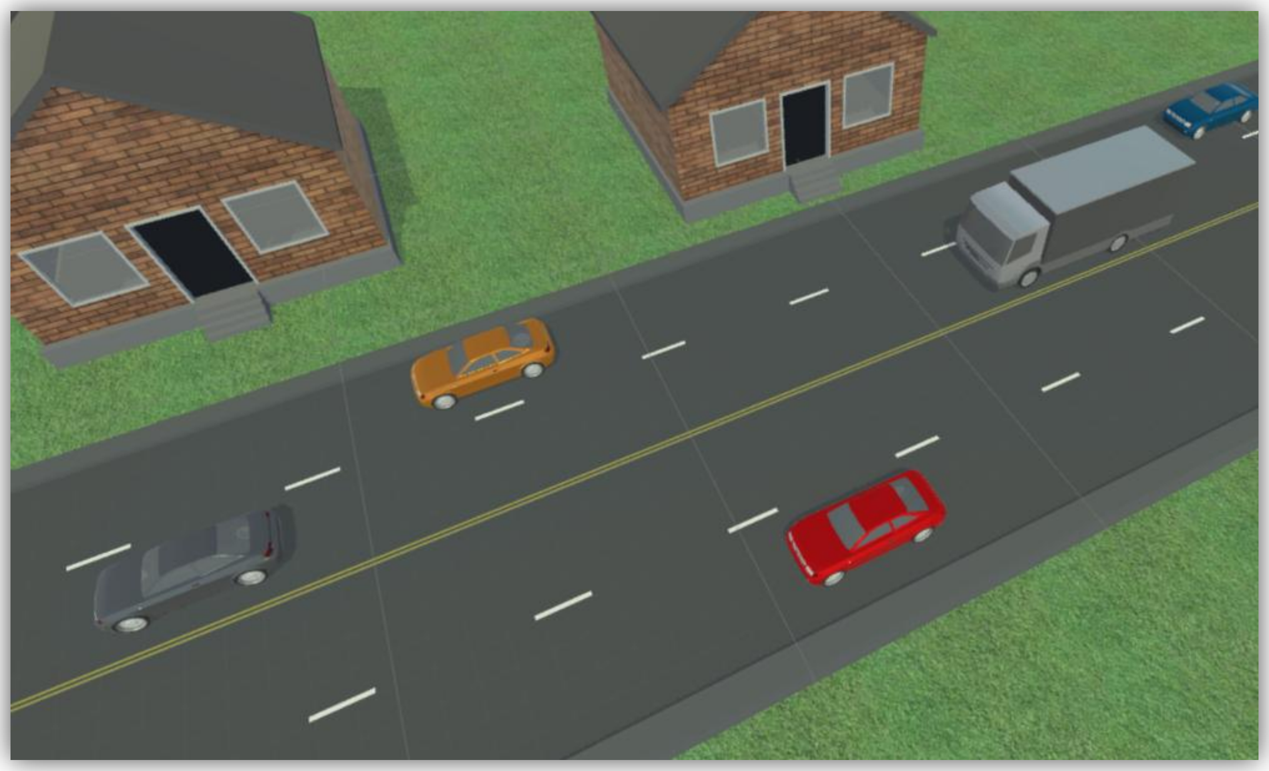
To prevent accidents between cars, the line switching cars check the road with rays, before moving. Also all moving objects with car checked, have a slow down distance. That distance is the length of the ray which checks for other cars. If another car hits the ray, the car moves faster to prevent accidents (in game it looks like the car slows down).

Manager script

The manager script is very important to the game, since it handles all the UI. Things like count down, Game UI, game over menu and resetting the game using PlayerPrefs. The manager script is the only script which inspector is controlled by an editor script. I think using an editor script is handy to delete PlayerPrefs, because you don't have to run the game that way. The manager holds the high speed variable too, but it doesn't really use it, it just gives it to the scrolltexture script.

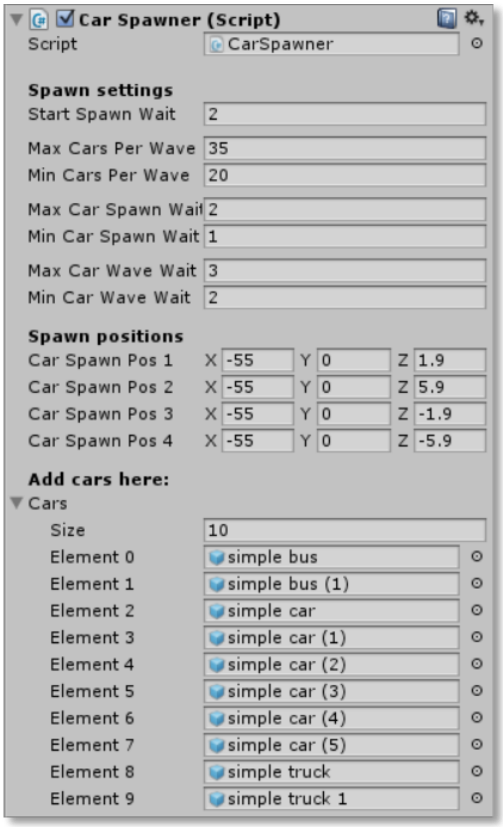
Creating your own scenes

To make your own endless game scenes, just copy one of the included scenes. Than go to the road object and add your own textures (the tiling must be the same as other textures to get the right speed). After you've added your materials, add them to the materials array of the 'scrolltextures' script. You can now add your own cars and environment too. If you want to use your scene in the game, you'll need to add it to the build settings and add an extra button to the locations panel in the garage.



Spawning cars & environment

The car and environment spawner work almost the same. The biggest difference is the spawning position. For the environment, there are two main positions (each side of the road) with a random spawn range to create some variety. The car spawner has 4 exact positions (each track) without a random range. With the min and max variables in the inspector, you can fully control the wave spawning and in the environment spawner, there's a separate variable called bridge. That's because bridges are always spawned at the same position (in the middle).



Car Spawner (Script)

Script	CarSpawner

Spawn settings

Start Spawn Wait	2
Max Cars Per Wave	35
Min Cars Per Wave	20
Max Car Spawn Wai	2
Min Car Spawn Wait	1
Max Car Wave Wait	3
Min Car Wave Wait	2

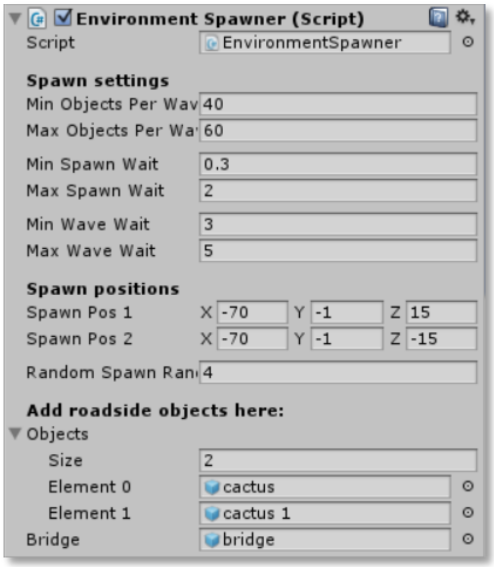
Spawn positions

	X	Y	Z
Car Spawn Pos 1	-55	0	1.9
Car Spawn Pos 2	-55	0	5.9
Car Spawn Pos 3	-55	0	-1.9
Car Spawn Pos 4	-55	0	-5.9

Add cars here:

Cars

Size	10
Element 0	simple bus
Element 1	simple bus (1)
Element 2	simple car
Element 3	simple car (1)
Element 4	simple car (2)
Element 5	simple car (3)
Element 6	simple car (4)
Element 7	simple car (5)
Element 8	simple truck
Element 9	simple truck 1



Environment Spawner (Script)

Script	EnvironmentSpawner

Spawn settings

Min Objects Per Wav	40
Max Objects Per Wa	60
Min Spawn Wait	0.3
Max Spawn Wait	2
Min Wave Wait	3
Max Wave Wait	5

Spawn positions

	X	Y	Z
Spawn Pos 1	-70	-1	15
Spawn Pos 2	-70	-1	-15

Random Spawn Ran	4

Add roadside objects here:

Objects

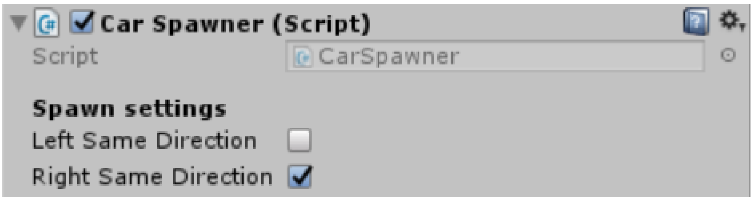
Size	2
Element 0	cactus
Element 1	cactus 1
Bridge	bridge

Crashing

The drivable cars (cars controlled by the player) have two ways of crashing and making the game over screen appear. First, they have a trigger on the front which immediately causes a crash. Secondly, they count the amount of collisions with other cars and after 5 collisions; they set the smoke effect active. If the smoke effect is active and the car collides again, it will crash.

Two directions

Since v1.4, cars can drive in two directions. Two enable/disable two directions simply use the checkboxes in the car spawner:



☑ Car Spawner (Script)

Script CarSpawner

Spawn settings

Left Same Direction □

Right Same Direction ☑

Left same direction means that all cars on the left hand side move in the same direction as the player and right same direction means that all cars on the right move in the same direction. So not checking ‘left same direction’ means all cars on the left drive towards the player. Disabling both options would make the game quite hard, since all cars would drive in the opposite direction of the player.

Conclusion

Thank you again for your support. In this document I tried to explain the Traffic Racing complete kit as clear as possible. I hope you make great games with it. All code has been commented, so that should help for customization.

If you have any questions, you can always contact me via:

T3Dmake@gmail.com

If you like the kit, a review in the asset store would help a lot.