Realistic Car Controller 2.6

by

Package Includes

6 Configurated Vehicle Prefabs for READY TO USE,

Configurated NGUI Controller for Mobile, and Dashboard Prefabs,

4 Demo Scenes for Presenting UI Mobile Controller, NGUI Controller, FPS Enter-Exit, and Desktop Controller,

User Friendly New Editor Script,

All Necessary Scripts,

Engine Sounds, Gear Sounds, Crash Sounds,

And more...

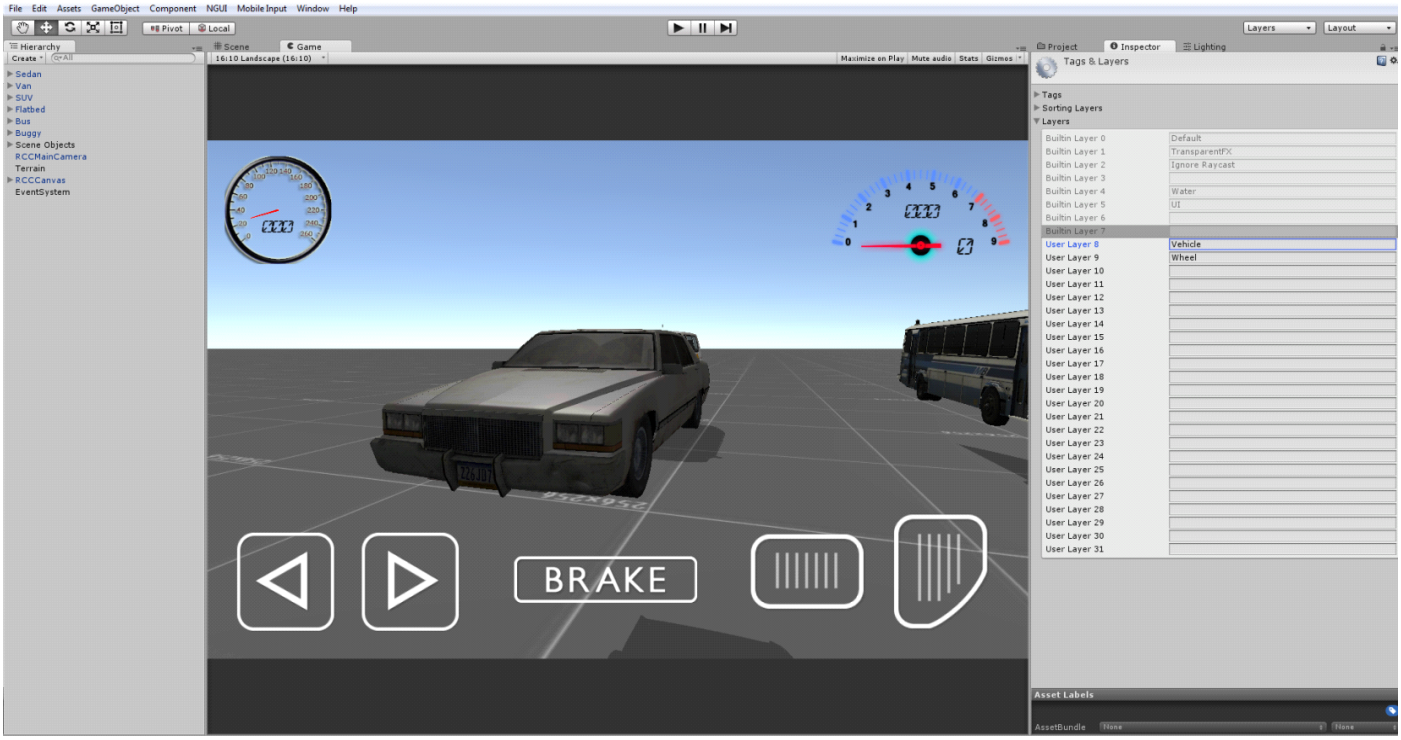
You can find more updated details on http://bugra381.wix.com/bonecrackergames#!realistic-car-controller-documentation/c1z2r

!!!Warning!!!

If you are going to Update your existing Realistic Car Controller in your Project, BACK-UP your whole Project Folder before each Update! Each Update uses different structs, and may overwrite your existing scripts/prefabs, or you may have doubled scripts! Keep in that mind...

First to Do!

Unity can't export Layers. You need to add these necessary Layers in proper order;



(If you will use NGUI, create "NGUI" layer for NGUI gameobjects too)

!!!Advice!!!

Please don't release the Demo Scenes on any Stores. Design and develop your own scenes, then release your game. Stores are going to garbage with these sample scenes. Of course all of our goals are earning money, but please keep this in your mind.

Each vehicle has own RCCCarControllerV2 script. Each vehicle is responsible for own RCCCarControllerV2.

8 Main Categories for Easily and Understandable Creating-Configurating Vehicles.

Wheels, Steering, Mechanic Configuration, Lights, Sounds, Mobile Controller, Dashboard, and Smoke Effects.

(You can zoom in with CTRL + ScrollUp for enlarge PDF pages)

About New Wheel Colliders

As you well know, Unity 5 uses Physx 3.3. They have improved wheelcolliders too. I'll explain new features about this;

Wheel Colliders can actually spin if you apply high torque to them. You can set your force point on Wheel Colliders. Realistic handling. Almost perfect everything. But... There are some bugs about current Wheel Colliders (5.0.0f4).

Wheel Colliders are based on your vehicle rigidbody. E.g. your Wheel Collider's spring force is 35000, and your vehicle mass is 1500. So when you lower your vehicle mass, suspension will loose force and vehicle will hit the ground unrealisticly. This is known bug about new suspension calculation.

Enabling/disabling "isKinematic" bool in Rigidbody, requires re-enable all Wheel Colliders. Otherwise Wheel Colliders won't work after enabling/disabling "isKinematic" bool.

Sometimes Wheel Colliders are lagging and shaking when your Wheel Collider's settings are unproportional, or your Wheel Collider's mass is too heavy/light.

Root of the vehicle scale must be 1, 1, 1. Otherwise Wheel Colliders positions will be changed. Just try it, and you will see your Wheel Colliders positions are changing while scaling root of the vehicle.

I'll update the package with each Unity update.

Creating New Vehicles

Some developers struggling with creating new vehicles. So, i have improved the editor script for simplify creating new vehicles.

!!!WARNING!!!

Script and behavior depends on vehicle X, Y, Z directions. So, your vehicle model and wheel models transform directions ~~should~~ MUST be correct. Just check the demo scenes.

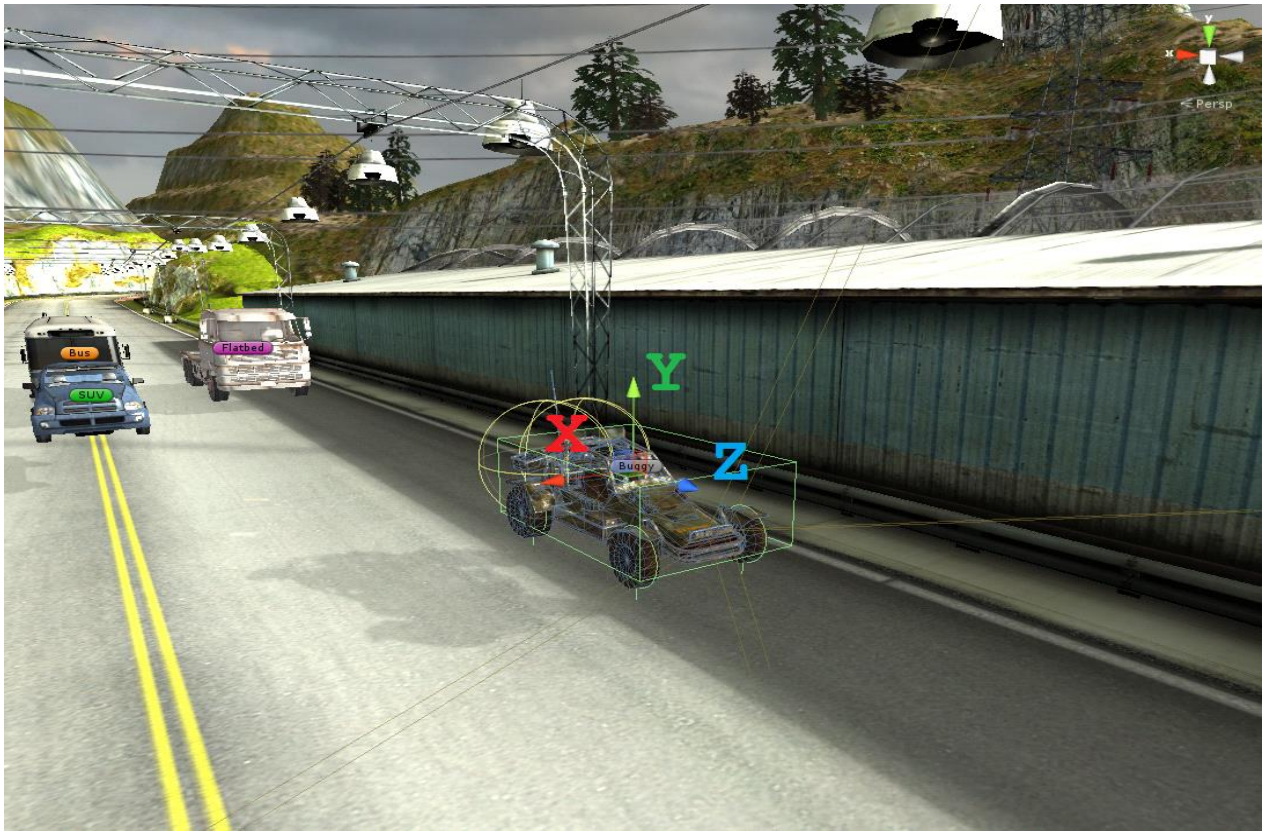
!!!IMPORTANT!!!

Be sure you are in PIVOT and LOCAL mode while checking directions.

X should Right,

Y should Up,

Z should Forward.



Many designers are making models with wrong directions. This is really painful if you don't know how to fix models directions. 2 ways for fixing this;

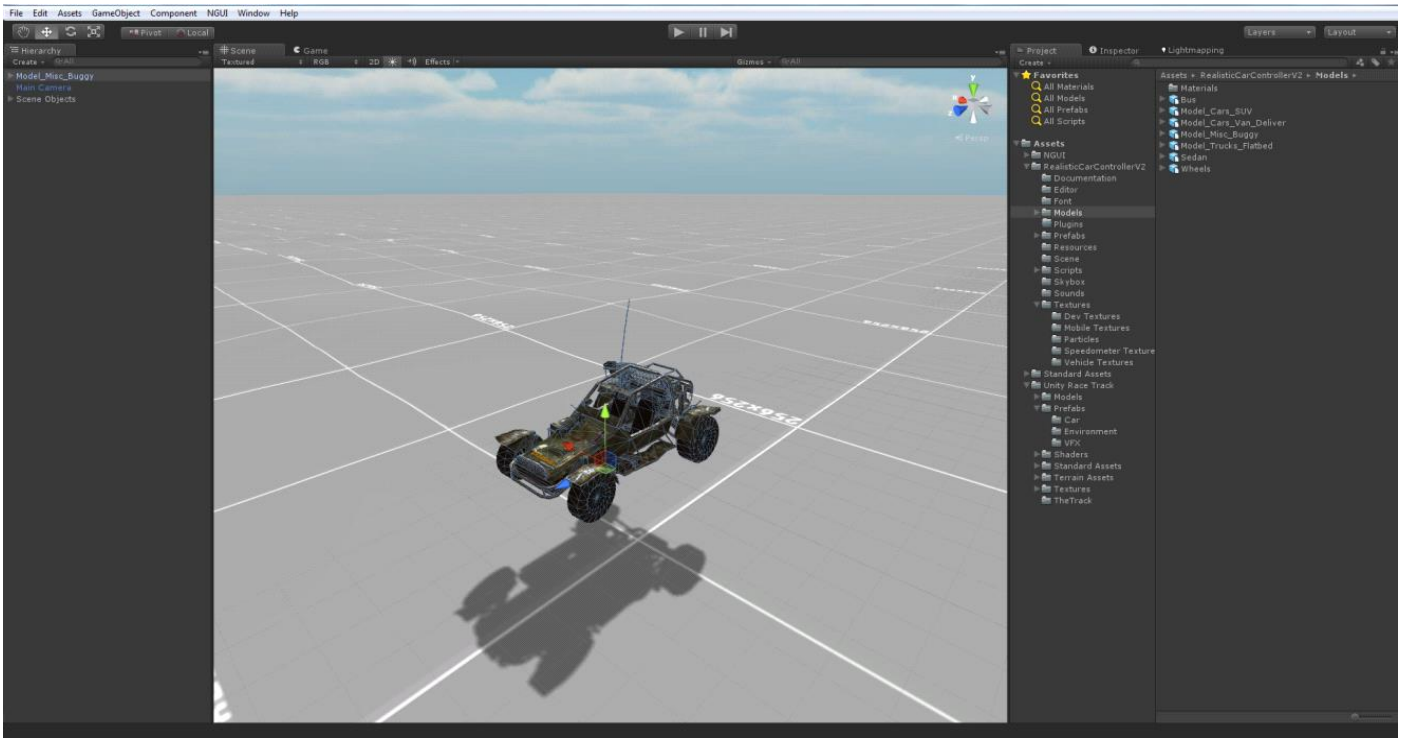
1 – Create a empty Gameobject at the center of the model, and rotate the new Gameobject to correct directions. Then parent your model in to this new Gameobject. (I don't prefer this way. If you want to do clean work, you should do this on your Designing Software).

2 – Fix model's directions and pivot positions inside your Designing Software. Check my Youtube Channel for how to do this in 3ds Max.

So, you have checked your Layers, Inputs, your vehicle and wheel models pivot positions, and their X, Y, Z directions. Everything is OK right? Then...

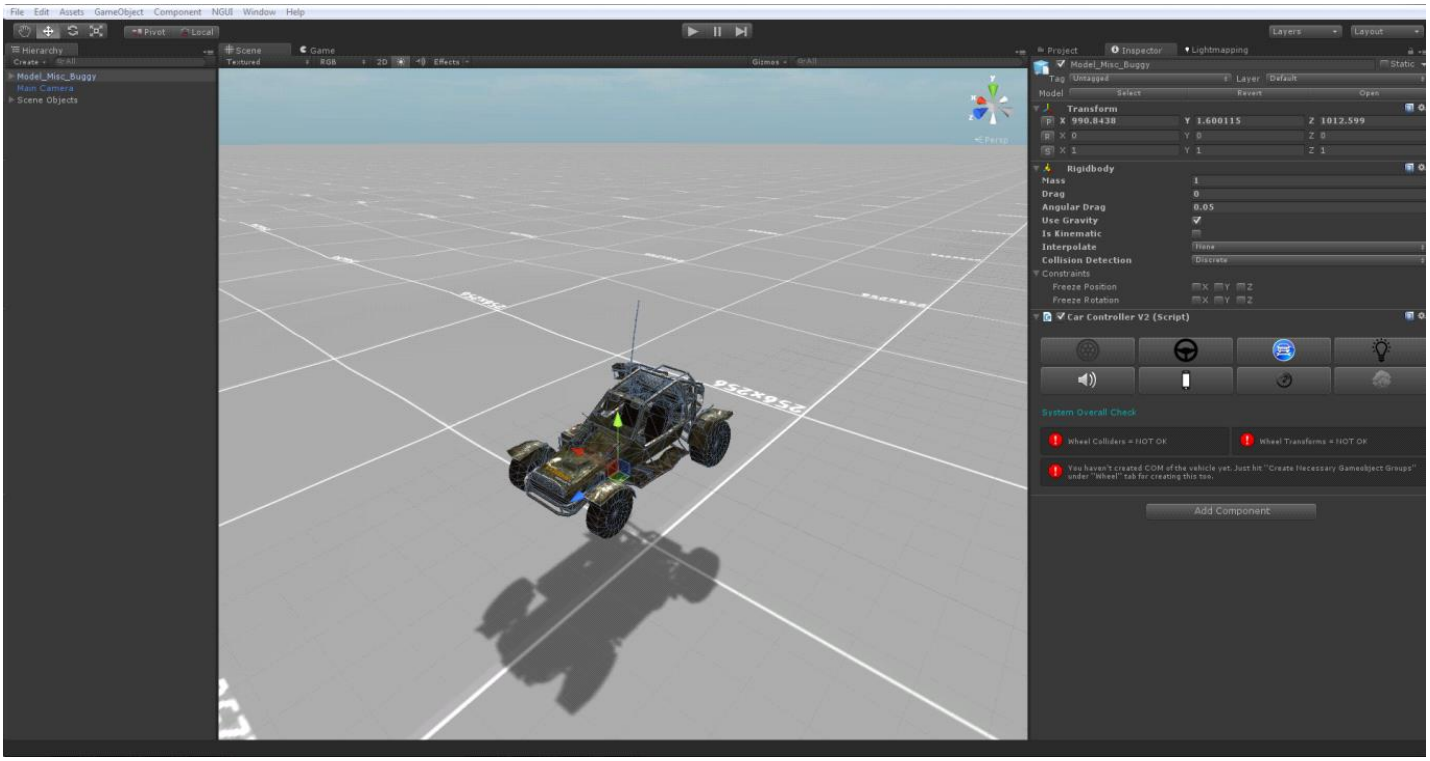
Drag and drop your Vehicle Model to your existing Scene and let's get started;

(Some images were taken in Unity 4, but they are nearly same with in Unity 5)



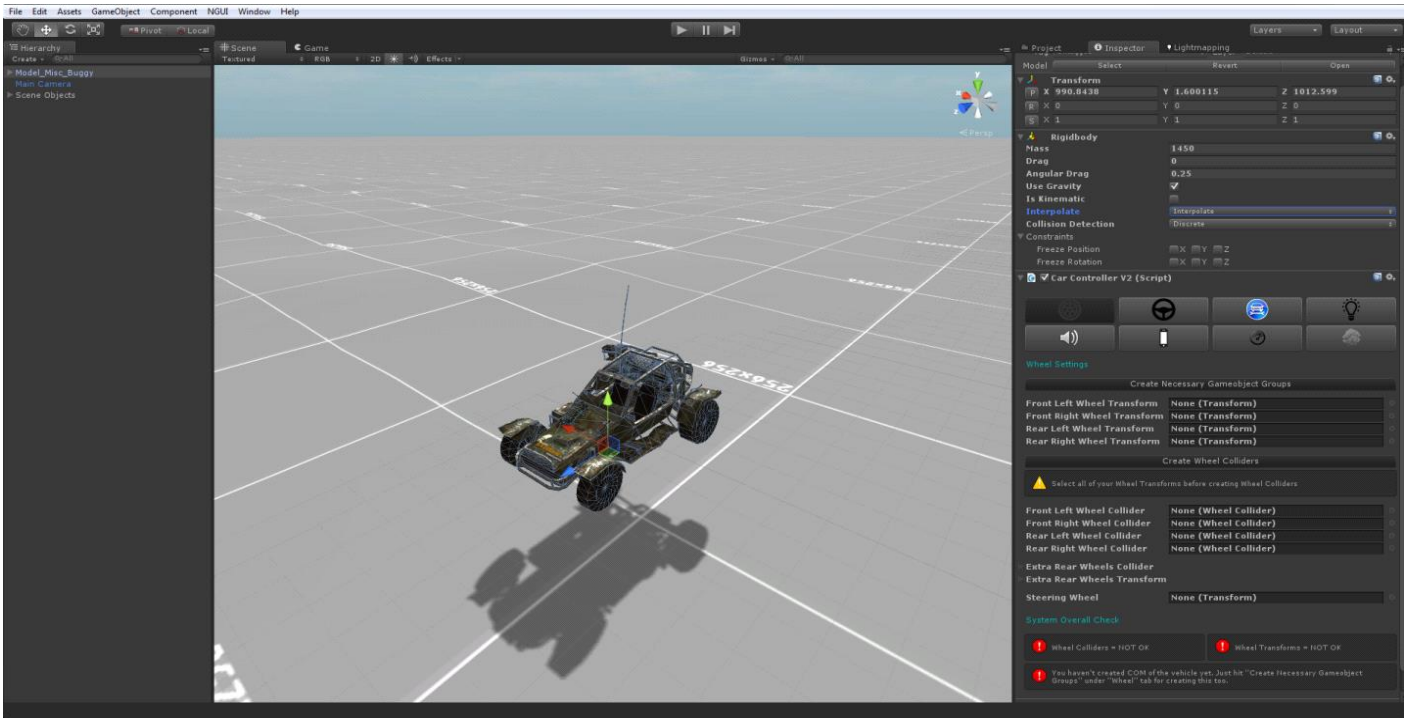
I'm using "Extreme Vehicle Models" for this. This is really great and mobile optimized asset for low price. I highly recommend you to use this amazing asset for your Project.

Now, i have to add RCCCarControllerV2 to root of the vehicle. Just select your root of the vehicle, and click "Add Component". Search for "RCCCarControllerV2", and add.

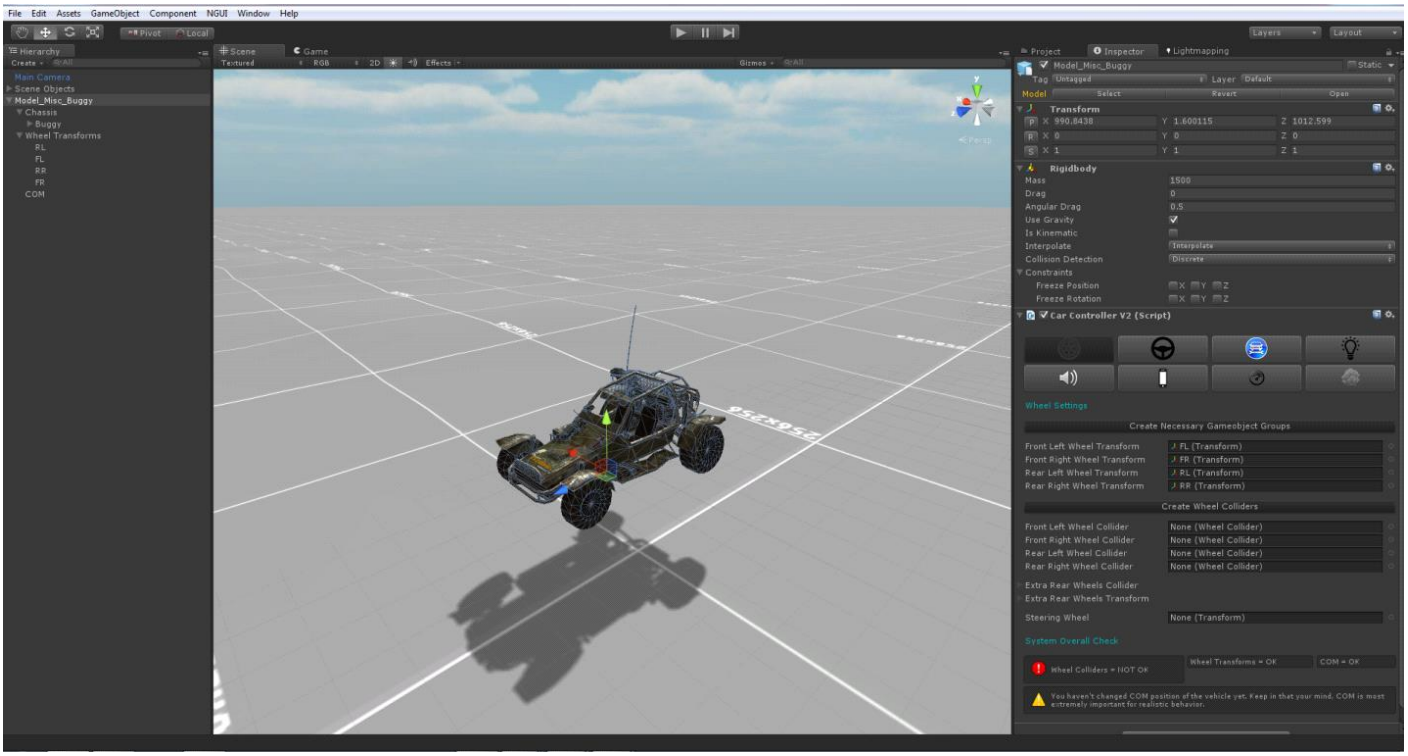


As soon as when i added RCCCarControllerV2 script to my car, Rigidbody component will be added automatically. Set your mass to around 1250-1750 for this type of the vehicles. Interpolate Mode = Interpolate, Angular drag is around 0.05 – 0.15 for medium angular velocity.

After rigidbody settings, click “Wheel” tab in the editor script;



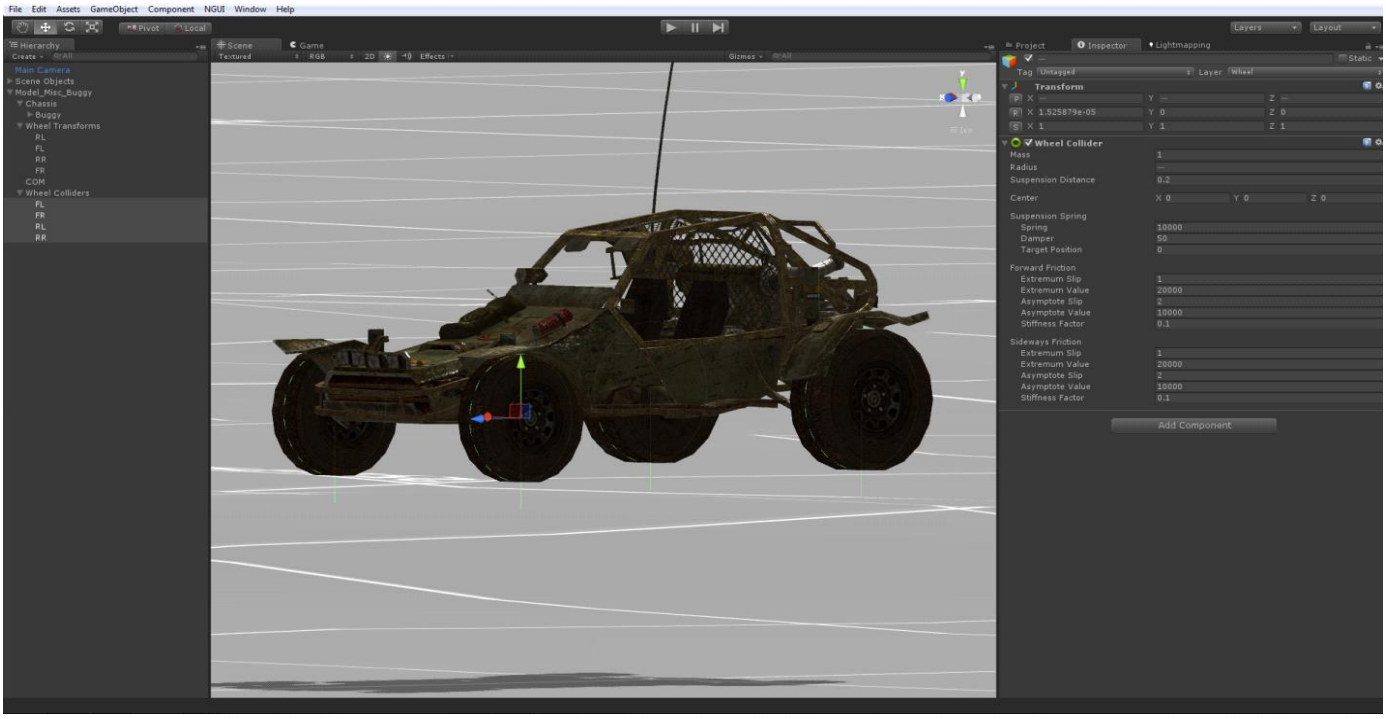
Click “Create Necessary Gameobject Groups”. This will create Chassis, Wheel Transforms, and COM of the vehicle.



After creating these necessary gameobjects, simply drag and drop your model's parts to corresponding property. Your wheel gameobjects to "Wheel Transforms", your chassis to "Chassis". Then select your wheel transforms inside editor script.

After selecting your wheel transforms, hit the "Create Wheel Colliders" for creating Wheel Colliders with proper radius, suspension, damper, and friction curves AUTOMATICALLY*.

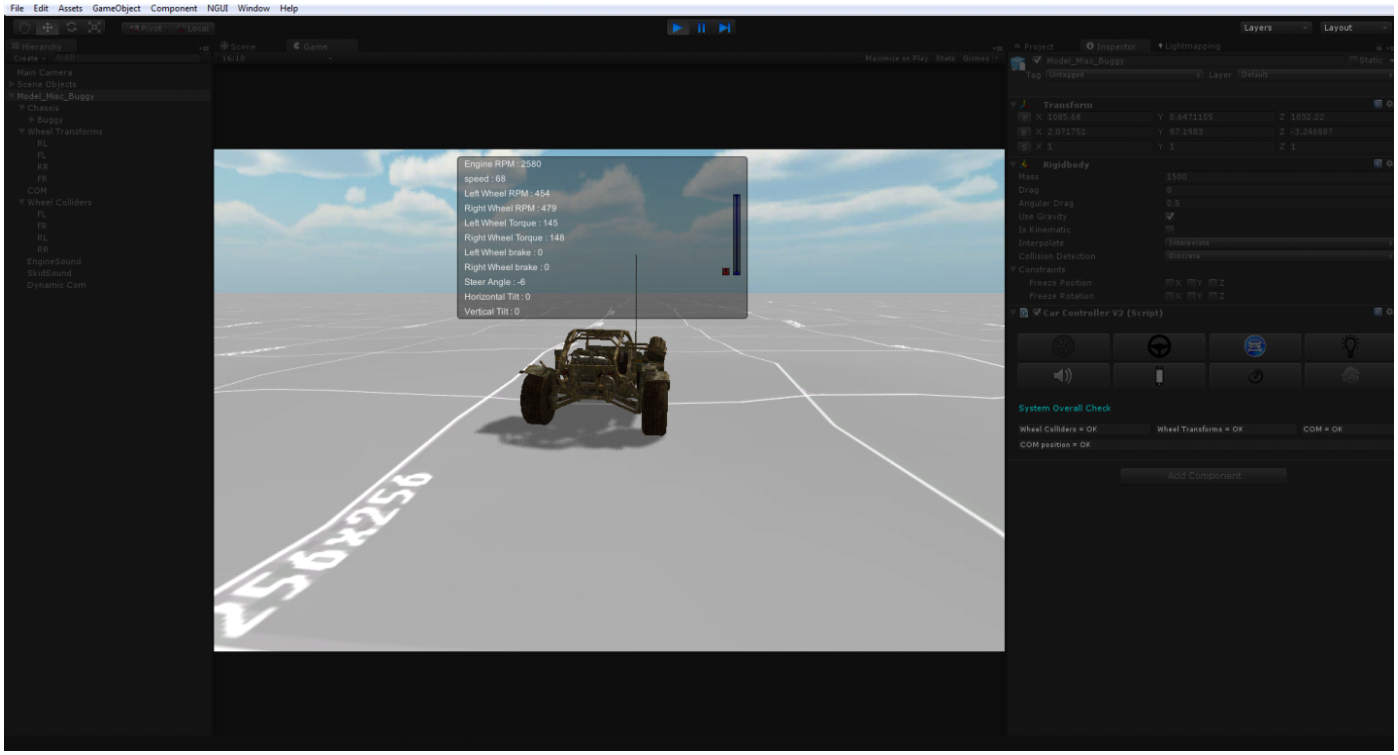
*(If your vehicle model direction is wrong, script will create Wheel Colliders at wrong angles.)



Generated Wheel Colliders settings are fine for 1250-1750 mass vehicles. If you have heavy vehicle such as bus or truck, you must increase Wheel Collider's mass. Your vehicle must have one of any Colliders (Such as Box Colliders, or Mesh Colliders etc...) Otherwise, physics won't work.

After end up with Wheel Transforms and Wheel Colliders Configurations, place your COM to correct place. This is our Center Of Mass. And COM's position is effecting whole behavior. Usually COM of the vehicle is at just below about front seats. Engine and transmission is at front of the vehicle, and they are heavy. At this model, engine and other stuff is placed to back of the vehicle. But i'm not gonna place the COM to back of the car. Why? Because this vehicle is RWD, and can flip back easily when applying high torque to rear wheels. So, i'll just set it to just like this;

That’s what I’m talking about, right? Give it a try!



Runs perfectly after just few clicks.

Default Main Settings are;

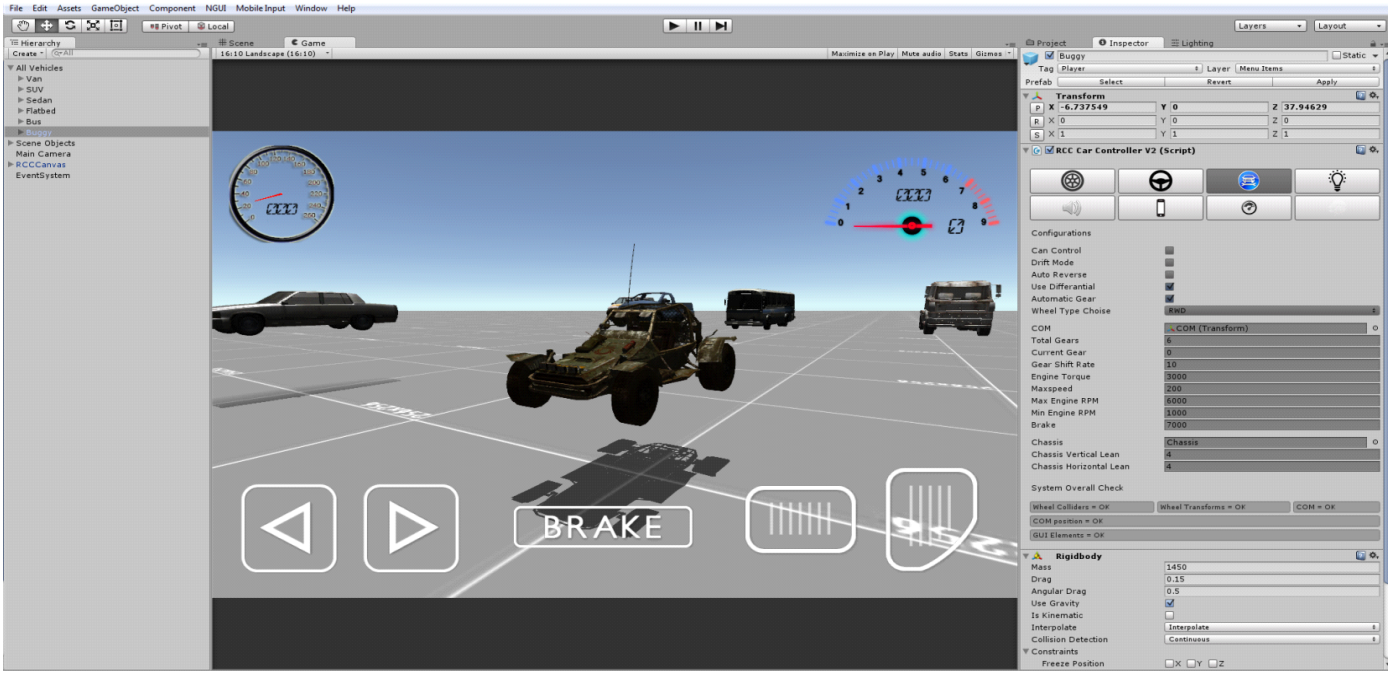
Engine Torque = 2500;

Brake Torque = 4000;

Maximum Speed = 180;

Wheel Type = FWD (I used RWD for this vehicle);

Use Differantial, Automatic Gear, etc...



Configurate your vehicle as you wish. If you want to use Manual Gear, you need to create 2 Inputs. "RCCShiftUp" and "RCCShiftDown". And set which key you want to. You can create new Inputs by Edit Project Settings Input.

Editor script is changing necessary object's Layers to corresponding Layer. Your vehicle Layer is "Vehicle", wheels Layers are "Wheel"

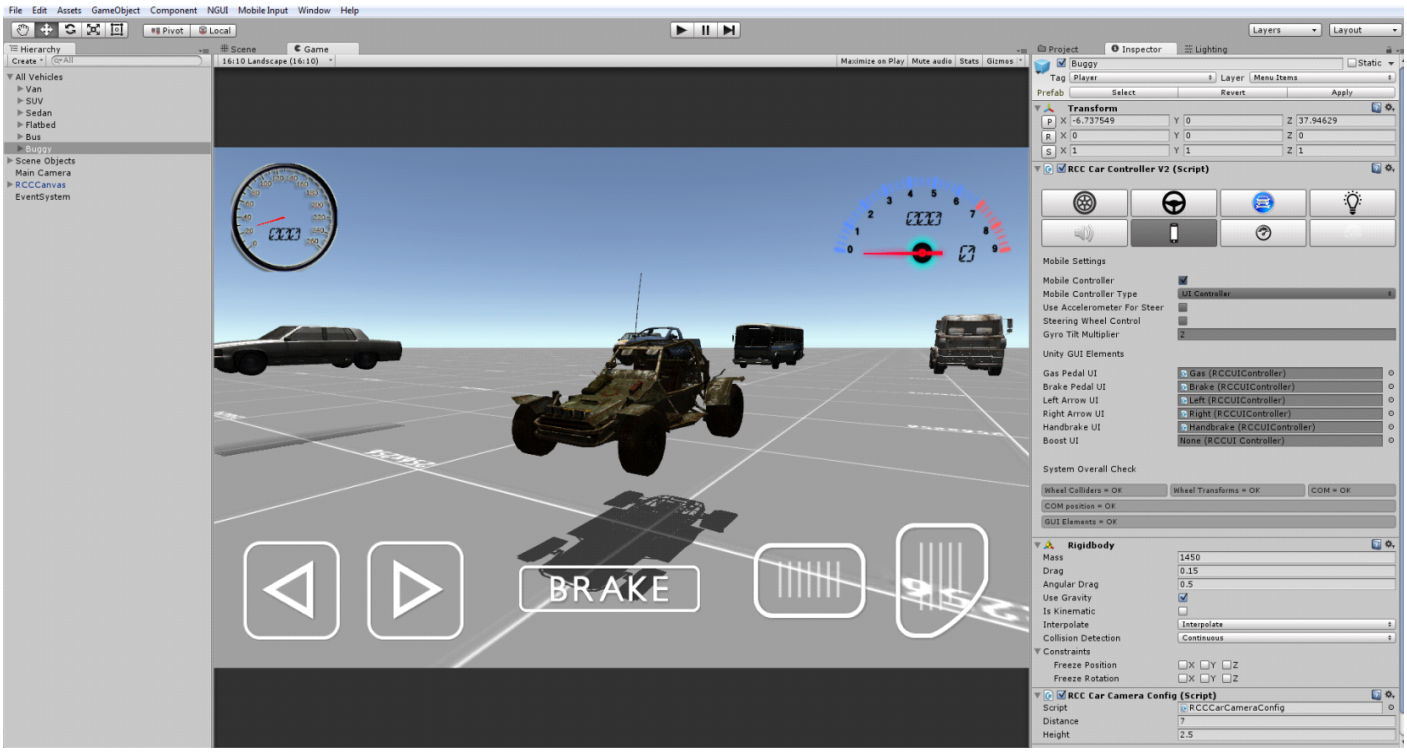
Now let’s make our vehicle can be controller by Mobile Controller!

Just click “Mobile” tab in the editor script and enable “Mobile Controller”.

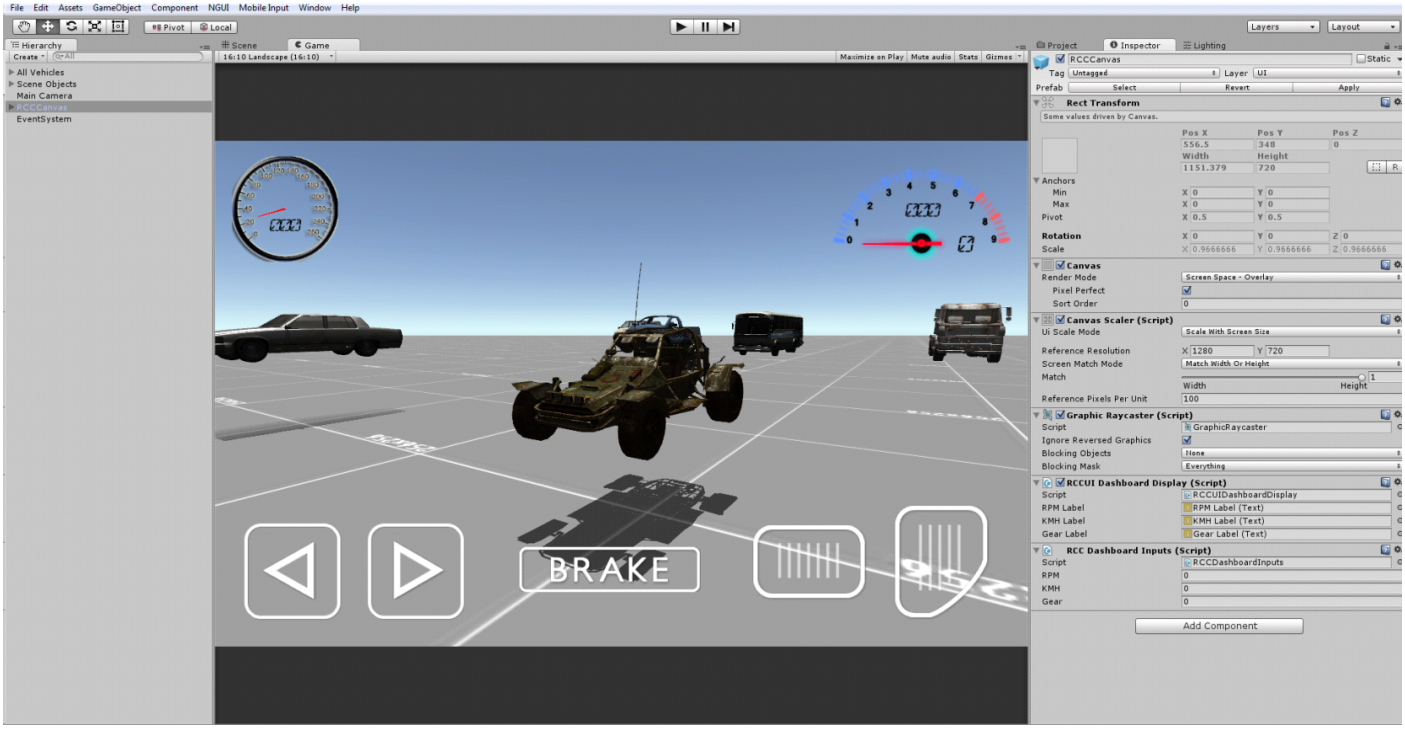
Package supports two type of controller and dashboard display mechanim;

- Unity UI (+4.6),
- NGUI.

MobileController (With Unity UI)



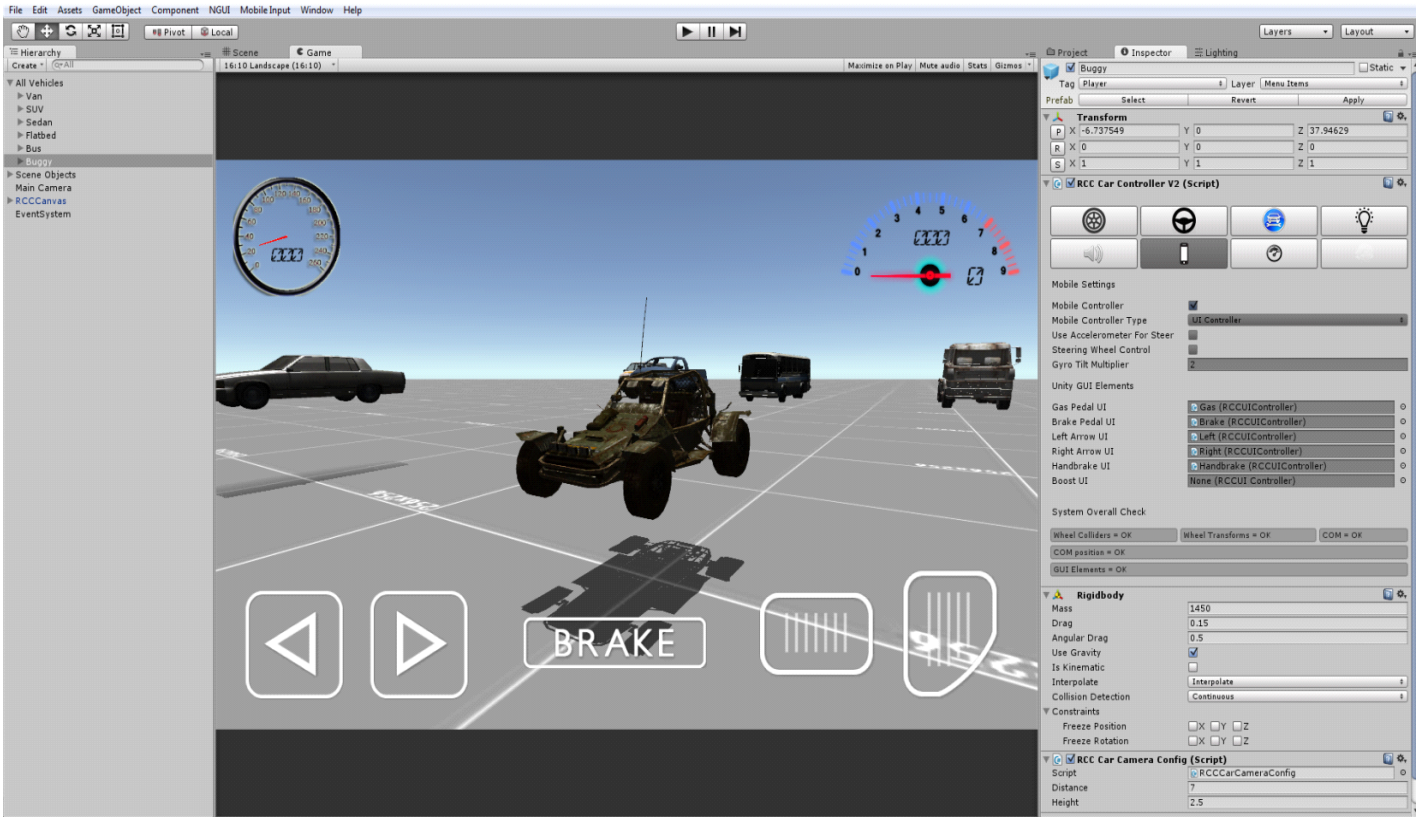
Default Mobile Controller type is “UIController”. You will find “RCCCanvas” prefab under RealisticCarControllerV2/Prefabs. Drag and drop to your Hierarchy.



Each UI controller gameobject has “RCCUIController” script for inputs.

~~(Obsolete)Select all of your UI controller elements inside the editor script, and give it a try.~~

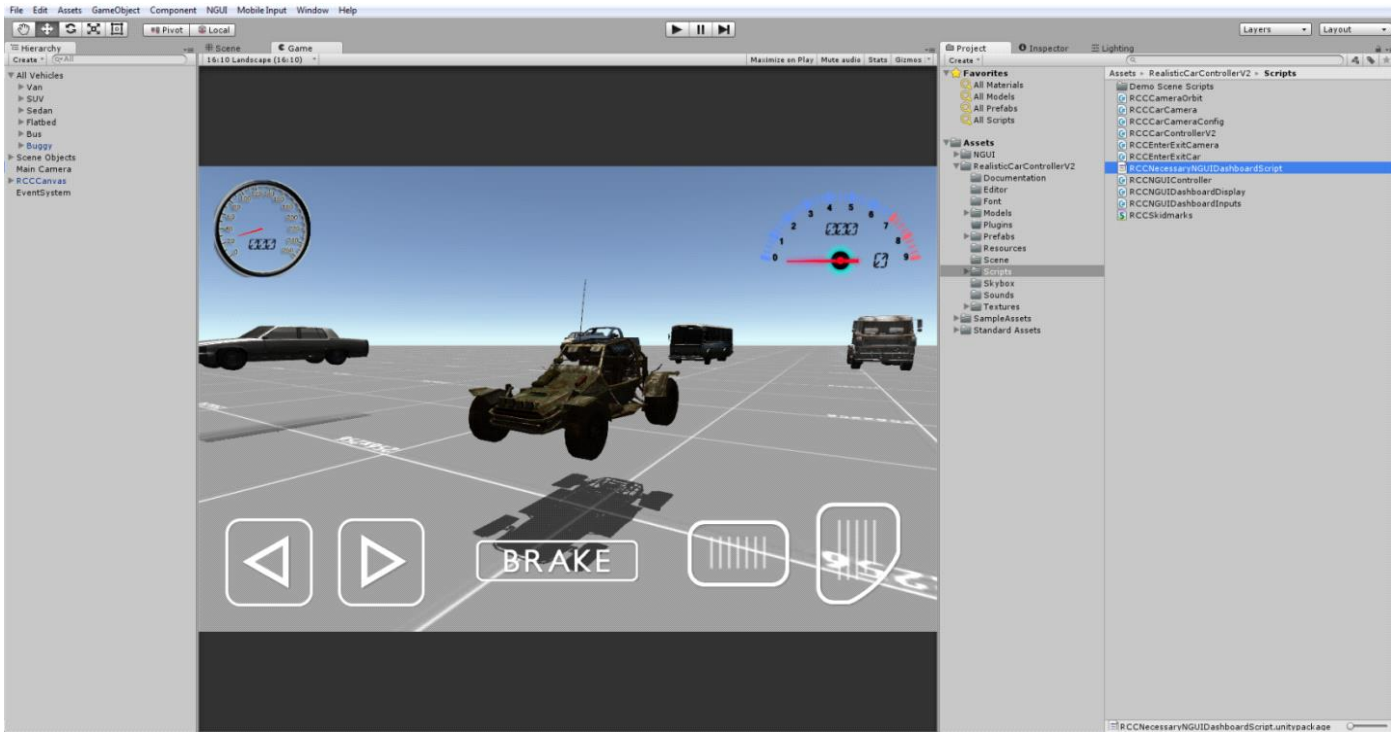
You can enable “Auto Find Mobile Buttons” bool, and let script find them automatically.



Runs perfectly with Unity Remote (Emulating to my android device). I forgot to add boost button, anyway, boost button is optional. Use it or not. Boost button will multiply your engine torque by %50.

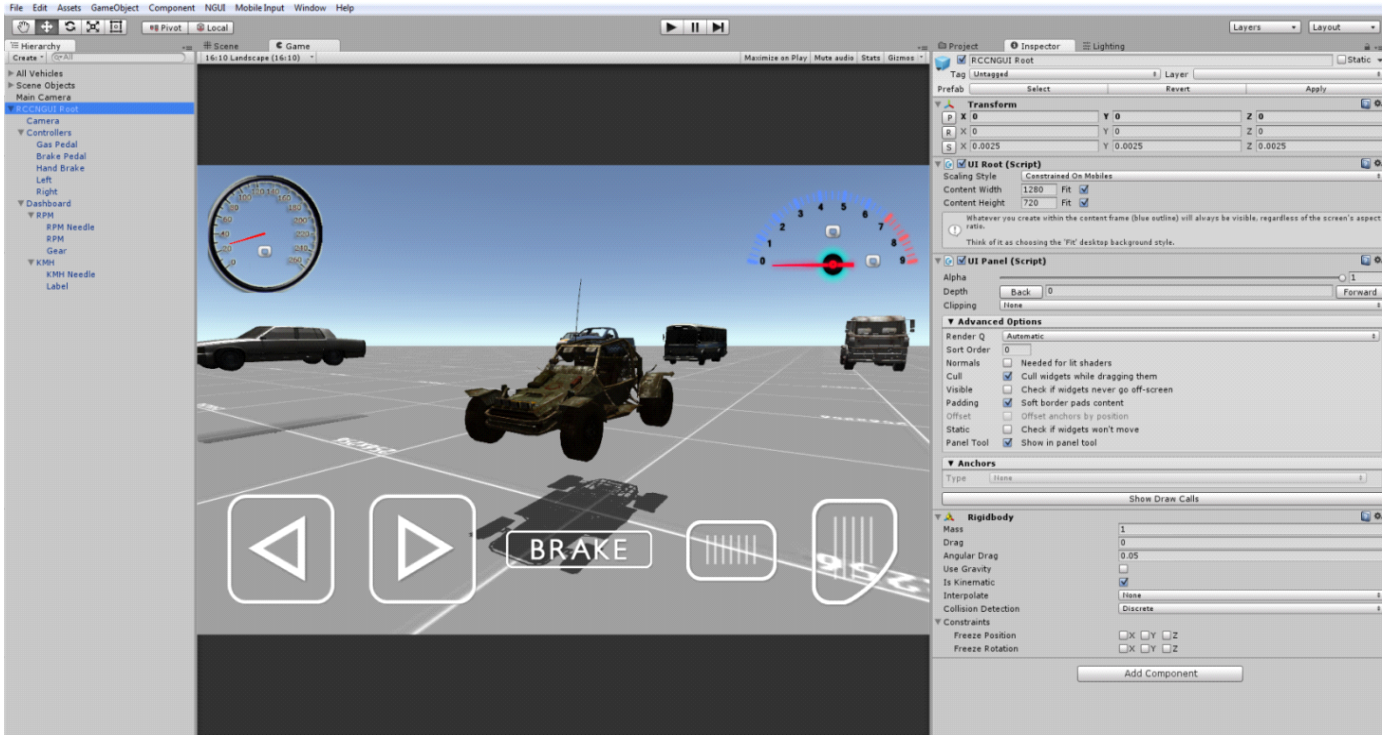
MobileController (With NGUI)

First, import latest NGUI Package to your Project. Then, import "Necessary NGUI Dashboard Scripts"(under RealisticCarControllerV2/Scripts) to your Project;



You will find "RCCNGUIRoot" under RealisticCarControllerV2/Prefabs. Drag and drop the RCCNGUIRoot prefab to your Hierarchy.

This prefab includes NGUI Controller, NGUI Dashboard, and other necessary scripts for Ready to Use. Canvas is designed for all aspect ratios. Controller elements won't disappear on other aspect ratios or screen resolutions.

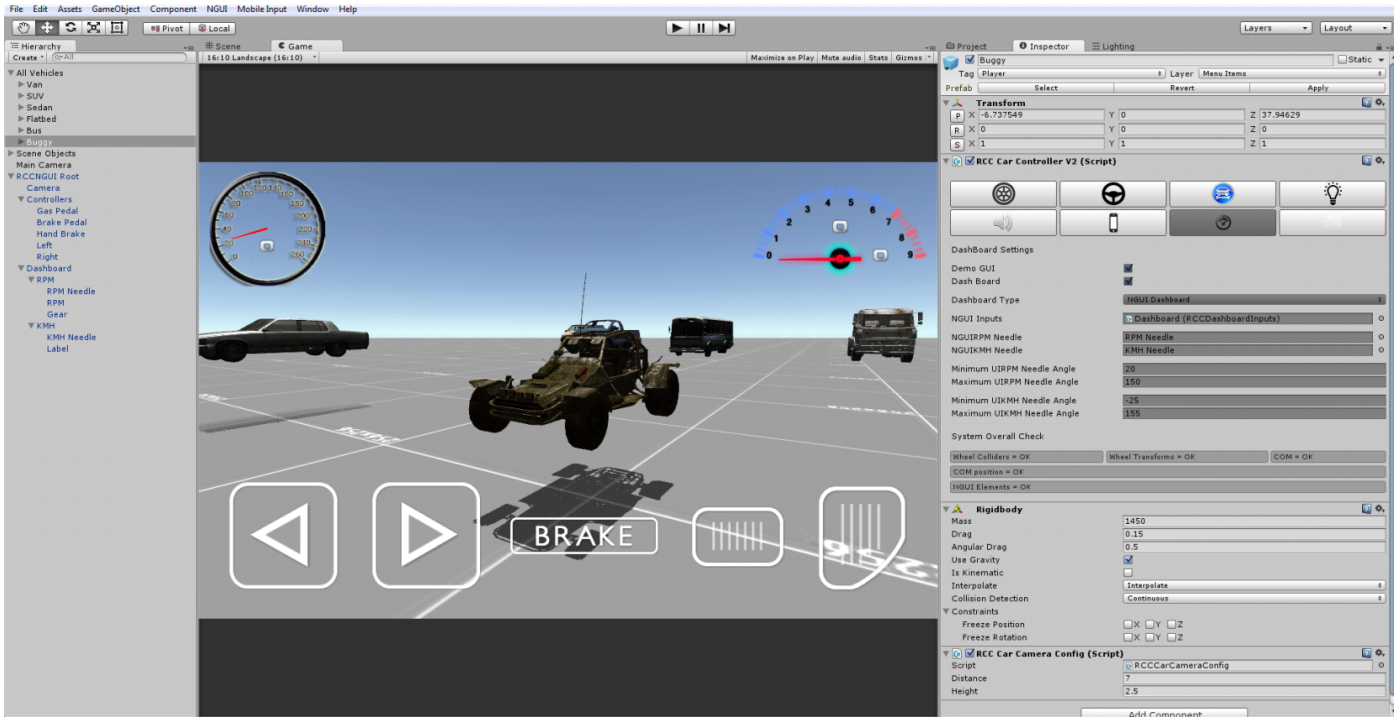


~~(Obsolete)Select all of your NGUI Controller elements inside editor script.~~

You can enable “Auto Find Mobile Buttons” bool, and let script find them automatically.

Dashboard Configuration

Just click the “Dashboard” tab, and enable “Dashboard”.



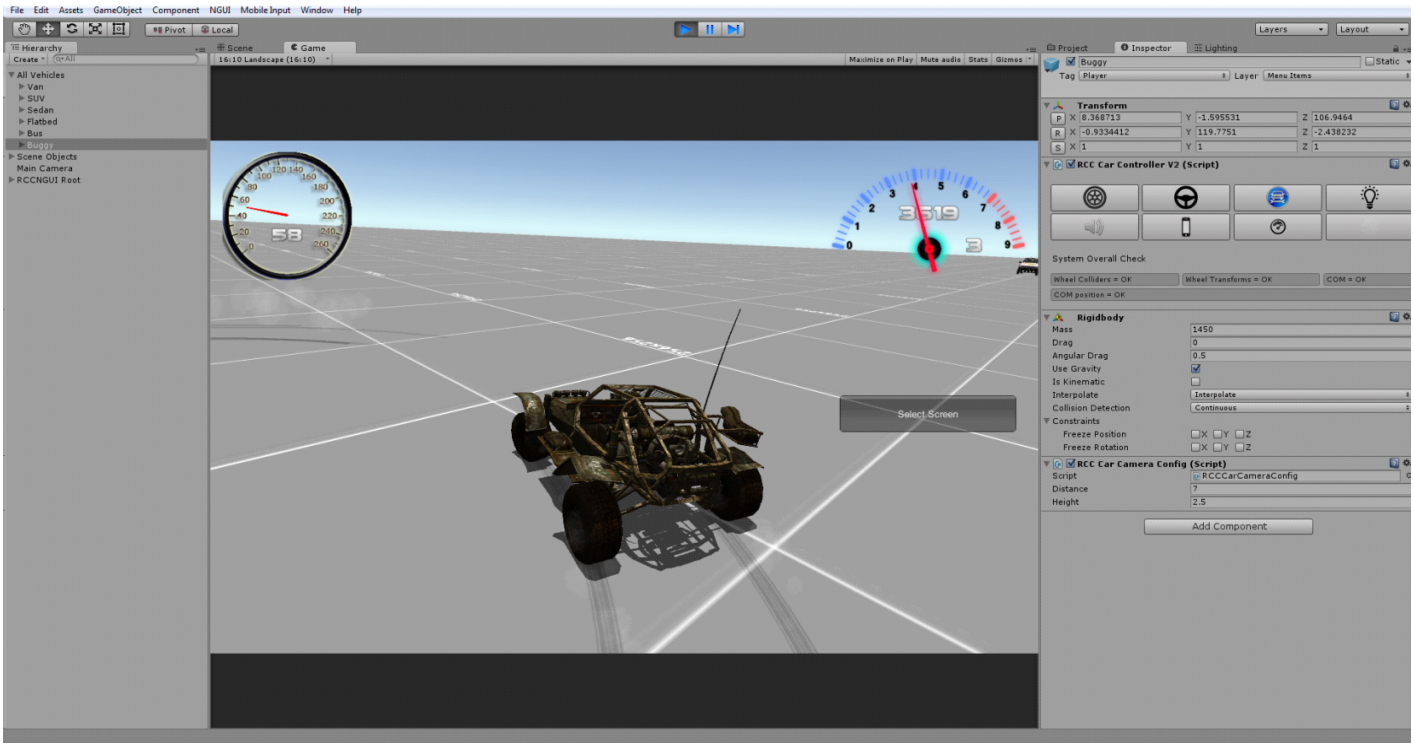
Default Dashboard Type is “Unity Dashboard”. ~~These GUI elements are not UI Textures or Sprites like Mobile Controller Elements. Script is drawing them inside OnGUI() function. So, if you want to use your own sprites/textures, you have to change their pivot positions and rotation angles inside script. If you don’t want to mess with scripting, just use NGUI. NGUI Dashboard configurations are much more easy against writing your own dashboard inside OnGUI().~~

Well, as I said, this method has been removed in this version. UI (+4.6) will be used for future updates of the package. Script will access “RCCDashboardInputs” and “RCCDashboardDisplay” scripts for displaying dashboard. These scripts are attached on;

Unity UI - RCCCanvas root.

NGUI – RCCNGUI Root.

It’s extremely easy to create and customize your own dashboard and controller HUD on Unity UI and NGUI instead writing your own OnGUI() method.



Runs perfectly.

(Ignore wrong vehicle layer. Importing “Sample Assets” from Unity will overwrite your existing Project Settings. That’s why vehicle layer is wrong at the picture :/)

Creating Lights, Sounds, Skidmarks, Smoke Effects

These effects are optional.

Create Point Lights for braking and reverse gear, spot lights for headlights. Place them correctly on your vehicle model. Click "Light" in editor script, and select all of your corresponding light.

Script doesn't Instantiate, Destroy any smoke particles, or any kind of stuff. Just enabling/disabling particle emitters for avoid garbage memory.

You will find "RCCWheelSlip" prefab under Prefabs folder. You can use your own smoke prefab as you wish.

If you want to use exhaust effects, you will find "RCCExhaust" under Prefabs folder. You need to place it to your model correctly. Then select it inside editor script.

You will find "RCCWheelSkidmarks" under Scripts folder. Attach it to your all of your Wheel Colliders. Select your vehicle inside script. And scene must have "RCCSkidmarksManager". You will find the "RCCSkidmarksManager" prefab under Prefabs folder. Drag and drop to your Hierarchy.

As i said, these are optional effects. If you don't want to use them, just leave.

So, how you holding up there? You can ask me anything about my Assets! If you want change minor things in the package, don't waste your time with editing scripts. Just tell me, I'll do my best with no cost. I don't take any Projects right now, I'm on multiple Projects at the same time. Please mail me if you used any of my Assets on your game. I'd like to see it in action!

Made by Buğra Özdoğanlar

BoneCrackerGames@Gmail.com