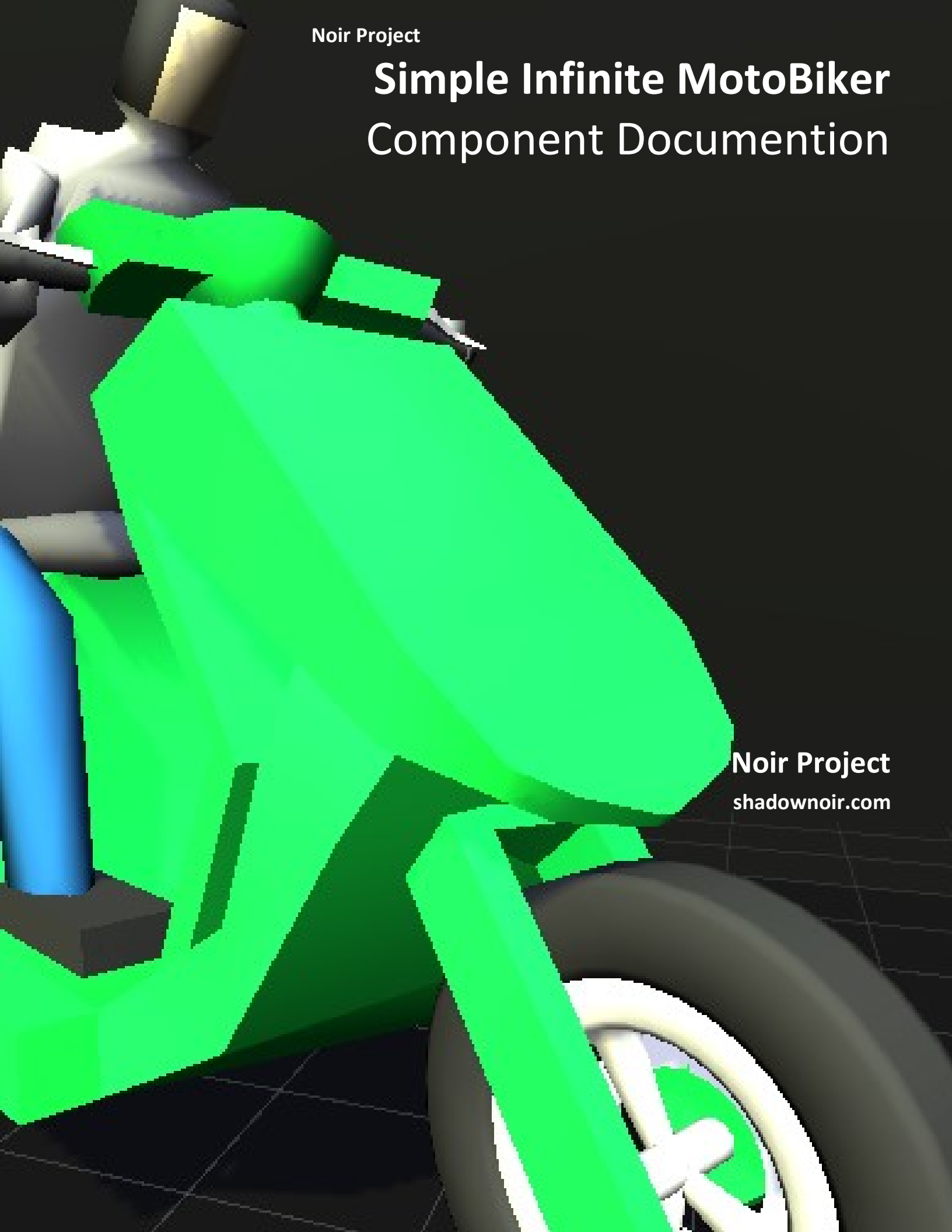
Noir Project

Simple Infinite MotoBiker

Component Documention



Noir Project

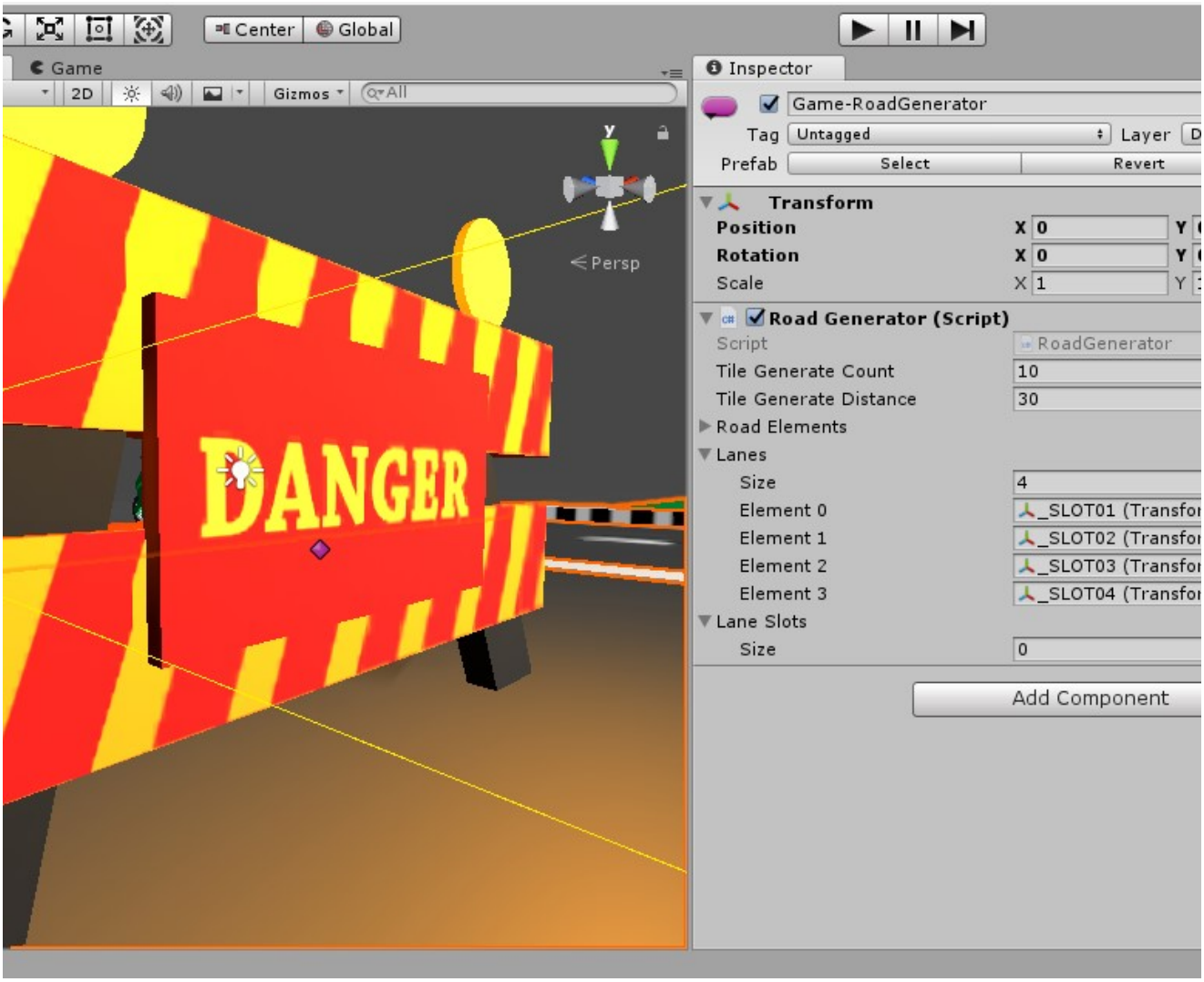
shadownoir.com

Table of Contents

Installation

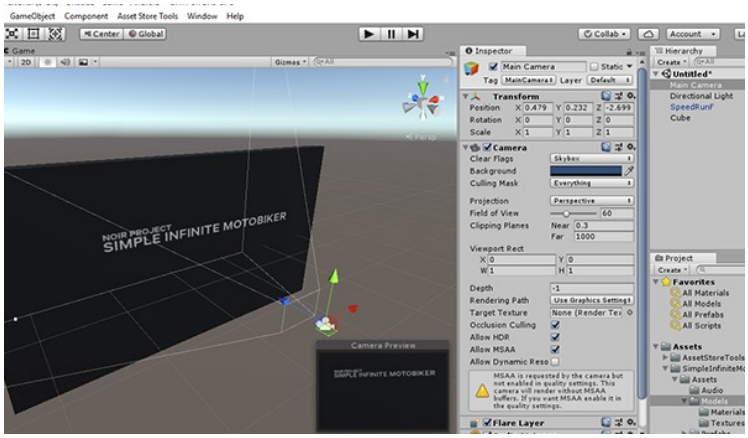
This component has different installation parts such as **menu**, **loading**, **player**, **road generator**, **game over scene**, **spawner** , **spawnables** and etc. also the complete playable scene is included inside scene folder where everything is completely setup.

So I will show you each of them separetly. The sample of each scene is also available in scene folder so you can test and modify the components right now. Just for start playing the game, select **Mainmeu** scene from **Scenes folder** and then press **play** button.



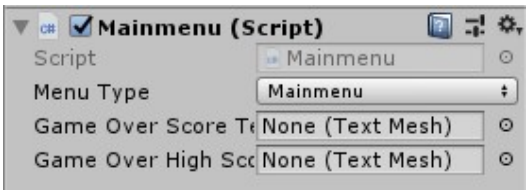
MainMenu Scene Installation

Create an empty scene then decorate the scene with prefabs or custom made meshesand reposition the camera.



Create or select an object and then add **mainmenu** componet on it from:

Noir Project > **Simple Infinite MotoBiker** > **Mainmenu**



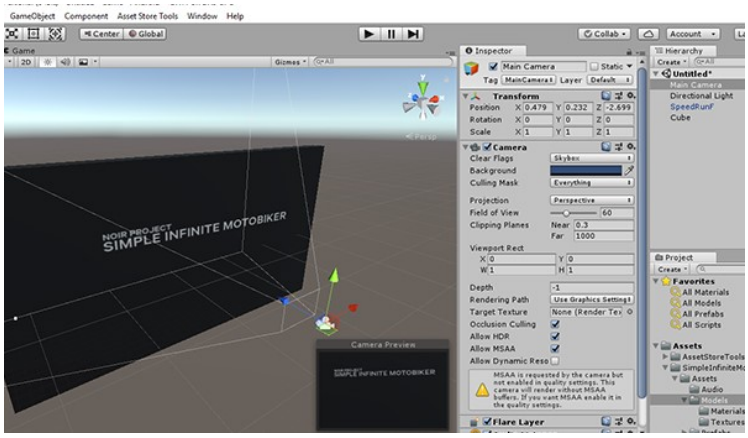
Select **Menu Type** as **Mainmenu**, and leave the others, we talk about them later in other sections.

Now when you play the game and then **tap the screen**, **press LMB on mouse**, or **press space** the next scene with name "**loadScreen**" called, **for changing the scene name** you can **edit mainmenu.cs from scripts folder near line 61**. This scene is a fake loader, we will create a simple fake loader in the next Section.

Everything is setup, now you can play and after tap the next scene is loading. This is our simple single tap to play menu.

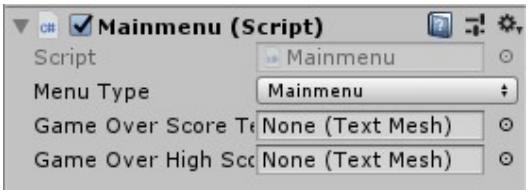
Fake Loader Installation

Fake loader is a scene to just wait for couple of seconds then run the game level, for starting to make fake loader just, create an empty scene then decorate the scene with prefabs or custom made meshes reposition the camera.



Create or select an object and then add **mainmenu** componet on it from:

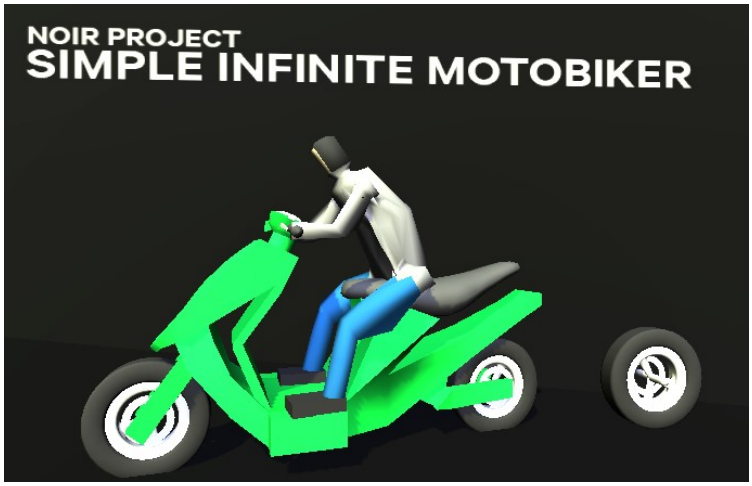
Noir Project > **Simple Infinite MotoBiker** > **Mainmenu**



Select **Menu Type** as **Loading**, and leave the others, we talk about them later in other sections.

Now when you play the game after couple of seconds the next scene with name "**level**" called, **for changing the scene name** and time you wait, you can **edit mainmenu.cs from scripts folder near line 22, 23**. The command `yield return new WaitForSeconds(3.0f);` is the timer, This scene is a fake loader, we will create a simple fake loader in the next Section.

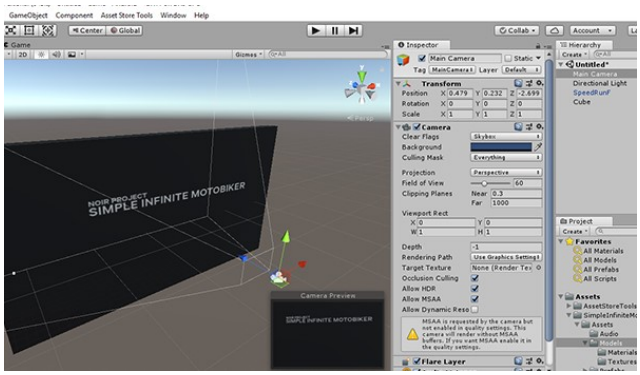
Everything is setup, now you can play and the game level is launched after a couple of seconds.



Gameover Scene Installation

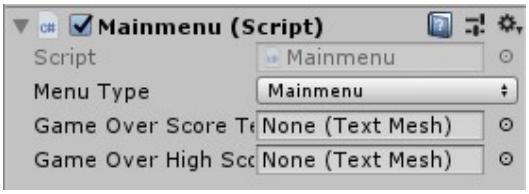
The gameover scene will called after player crash, the scene will show the score and highscore too. The gameover scene is called from **player.cs near line 562 inside DeadBiker()** method.

Create an empty scene, Decorate the scene with prefabs or custom made meshesand place the camera.



Create or select an object and then add **mainmenu** componet on it from:

Noir Project > **Simple Infinite MotoBiker** > **Mainmenu**



Select **Menu Type** as **Game Over**, and then if you would like to show score on scene as TextMesh you can **create a new TextMesh** and **set the position**, then define it as a **GameOverScoreText** to show player score there, you can also do the same for the highscore text.

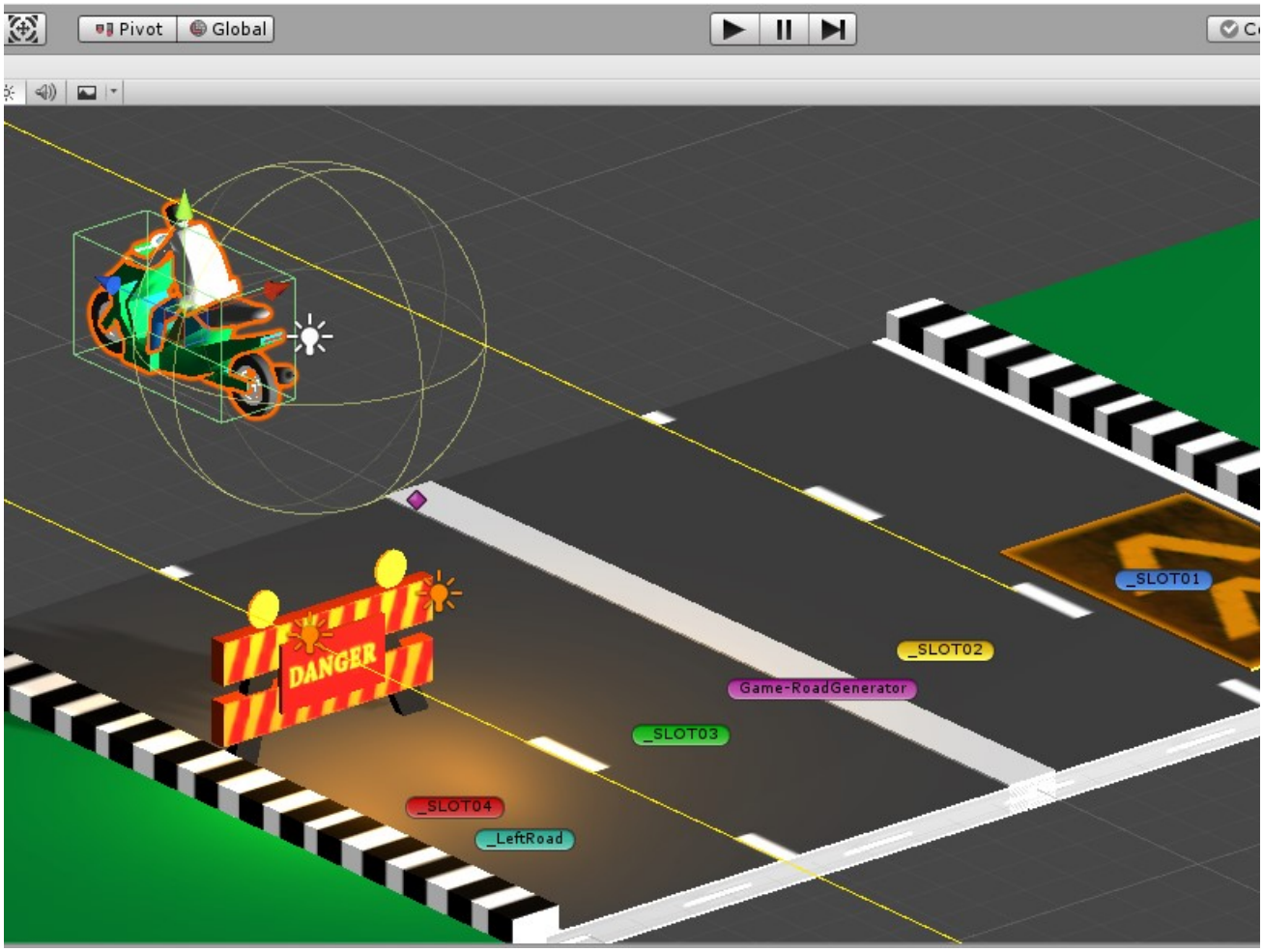
Now when you play the score and highscore will shown on text mesh and when you click on GUI button restart game this will load next scene with name "**MainMenu**", **for changing the scene name** you can **edit mainmenu.cs from scripts folder near line 69**.

Everything is setup.

Game Level Installation

First of all we need to make a road with road generator, constructed via different objects and lenghts. So I will show you how to setup complete game level containig spawners, spawnables, road generator, player, music and etc.

For the beginning we start to make Road Generator. But road generator depend on the road elements so in step one we create a road element then setup the road generator.



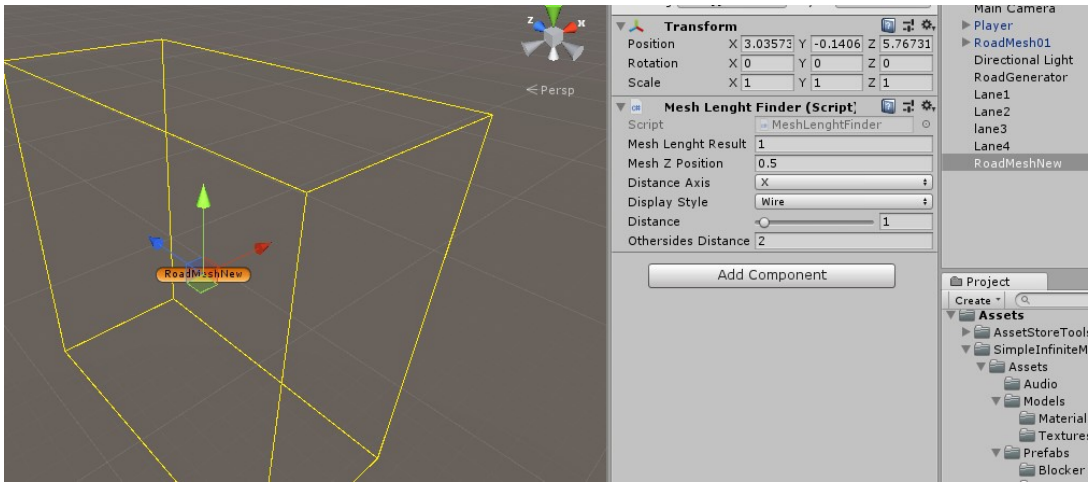
How to create Road Mesh (Road Element)

Road Mesh elements together can be construct your roads, but we need to know how to make one of them. Here is the few steps to create the roadMesh Element.

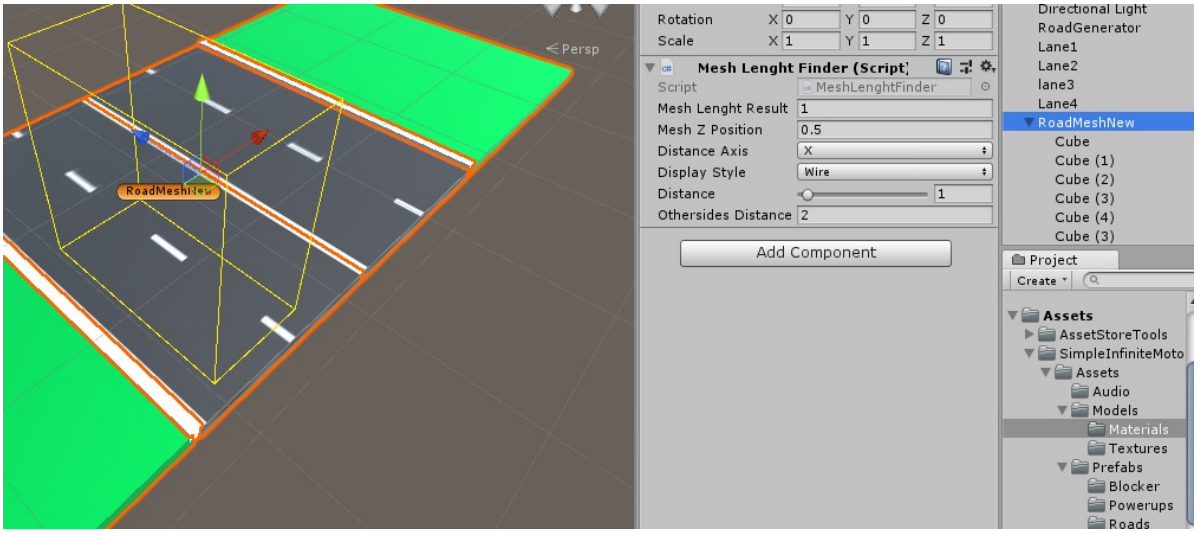
First of all create a new Empty gameObject, I will call that RoadMeshNew

Add MeshLenghtFinder Component on this game object from:

Noir Project > **Tools** > **Mesh Length Finder**

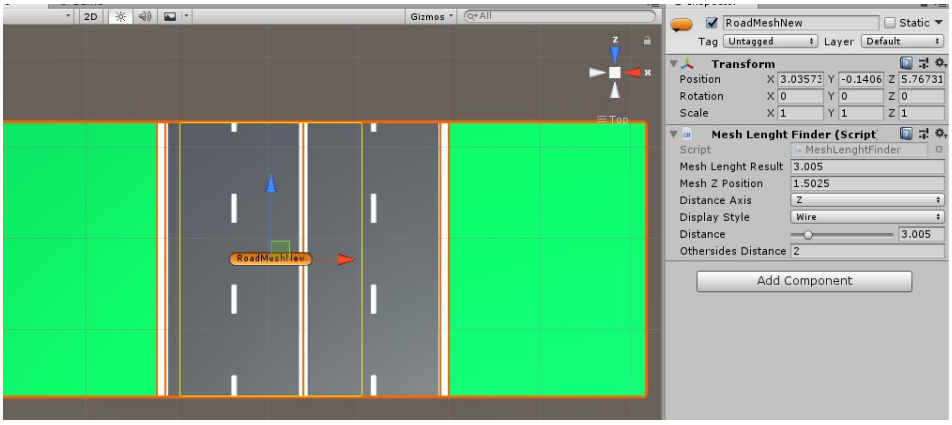


Now use the RoadMeshNew Gameobject as parent and start to draw your road mesh part or import a model and align it inside the RoadMeshNew, please be careful about the Z Axis, it's forward, I make something like this: (the childs are some cubes textured, and on top of them Mesh Length Finder is available. All of the childerens should be place on the center of the parent gameobject. Because the length finder find the spawn length from the center.

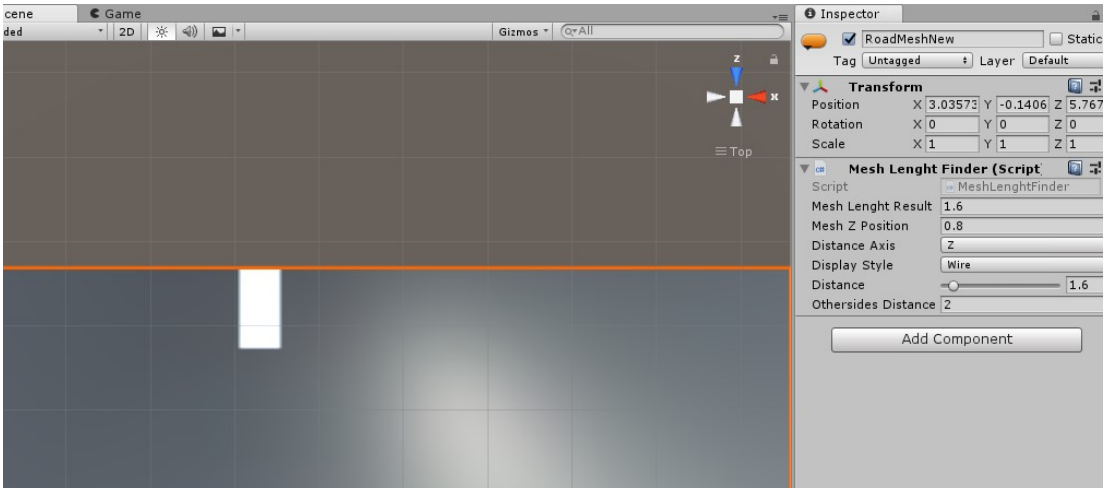


Now Select **Distance Axis Z** because our road goes forward axis, and **Display Style** to **Wire** for **Wire gizmo Cube** also you have option to use **solid gizmos**.

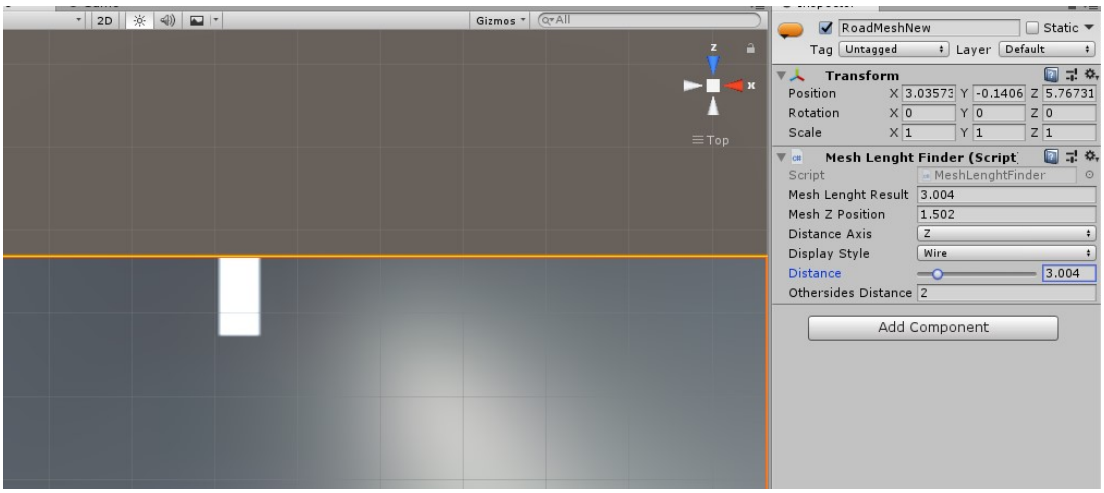
Set the editor camera to **iso mode** and click on **top arrow** to show the road in top view.



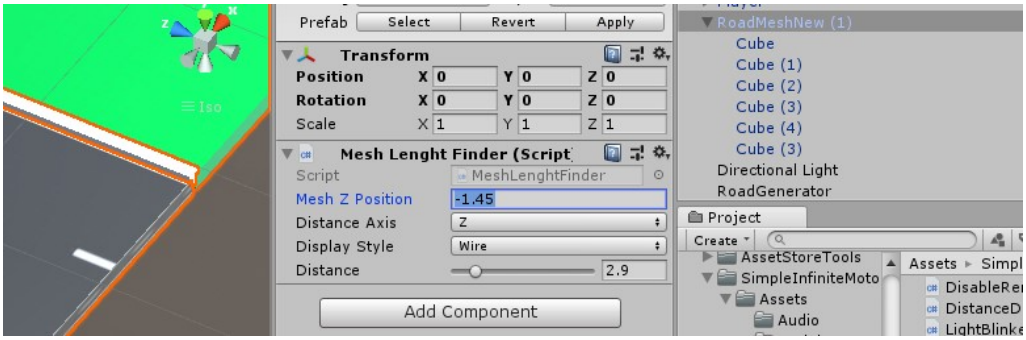
Zoom in to see the edge of the road.



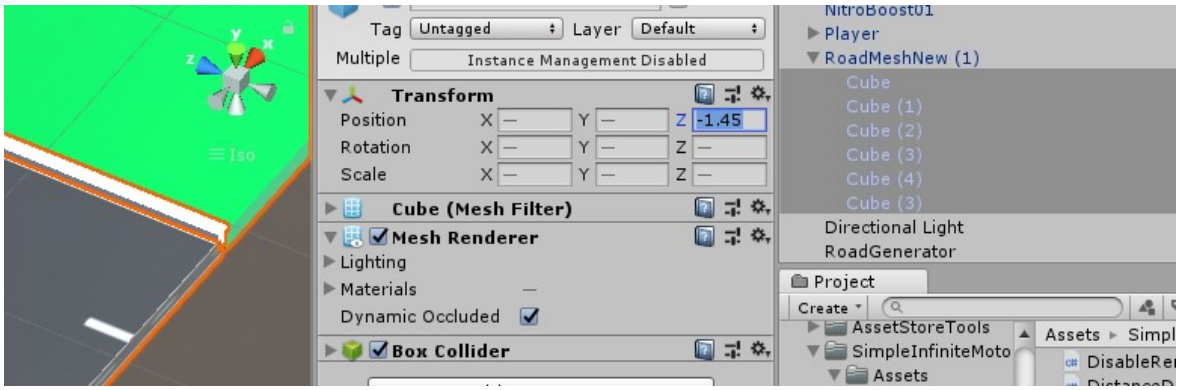
Now change the distance to the yellow line, aligned to the orange select line. As shown in the image, in my case 3.004 is alignment of the road.



Now the mesh length tool is aligned. Now check out the value Mesh Z Position shown.



You should set this value on the transform Z position of all of your road element mesh childerens, as shown above RoadMeshNew (1) have 6 Child mesh under so we need to select them all and add the Mesh Z Position on them, if you don't do that, you cant use different road mesh in road generator and I will make gape between the generated surfaces.



Now the road element is ready. Save it as a prefab by drag and drop the gameobject to the Project window.

How to create Road Generator

Road genrator is a component, that can handle generating road and road lanes, the road can be constructed from array of gameObjects and each element has property such as Transform, Length and Spawn chance.

Also you can use predefined prefab from **Assets** > **Prefabs folder for ease of use** we have a file named **Game-RoadGenerator**, this prefab contain all of the settings predefined and you can put it on the scene and **set** the **position** to **(0,0,0)** and road generator should work.

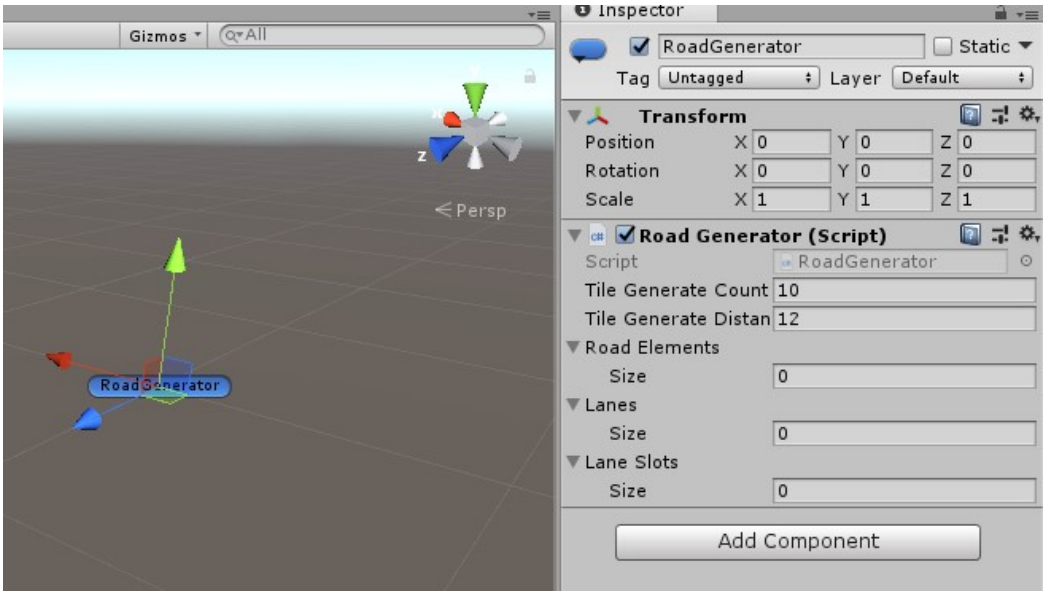
Right now we start to produce a new road generator from scratch.

Few steps to create a new Road Generator

Create a new Scene and then Create a **new Empty GameObject**, set the **gameobject position to 0,0,0**, the **rotation to 0,0,0**, and for the **scale 1,1,1**, and then **name** it something that can make sense such as **RoadGenerator**, etc.

Add new component on the gameobject from

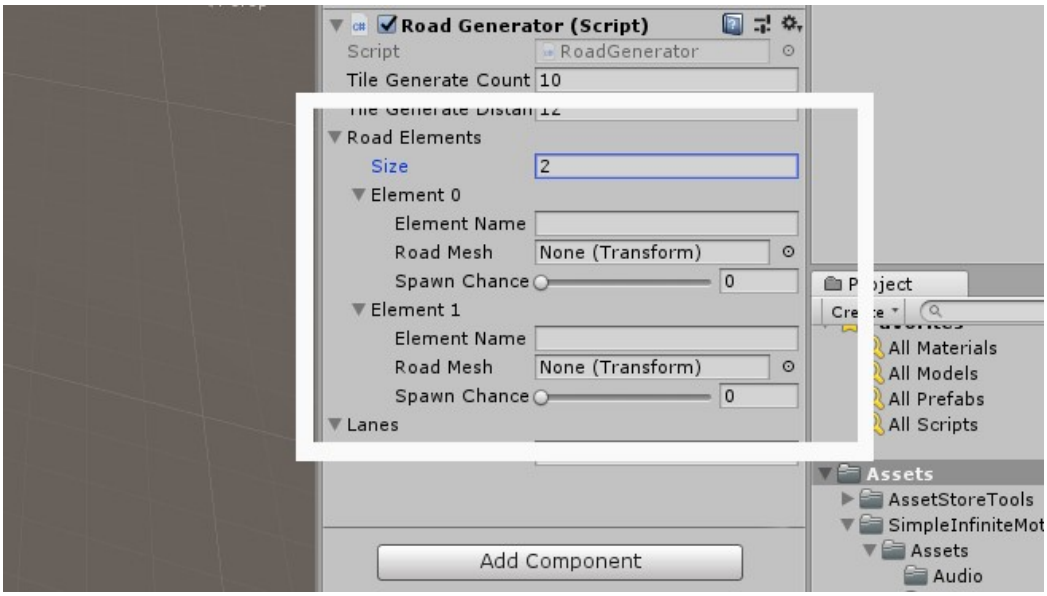
Noir Project > **Simple Infinite MotoBiker** > **Road Generator**



Set the **Tile Generate Count to 20**, this value define **how many tile (roadElement) required** for eah time road generator **constuct a road**.

Set the **Tile Generate Distance to 30**, this value define how much **distance** can be **between** the **player and** the **last roadElement for regenerating** next batch of road.

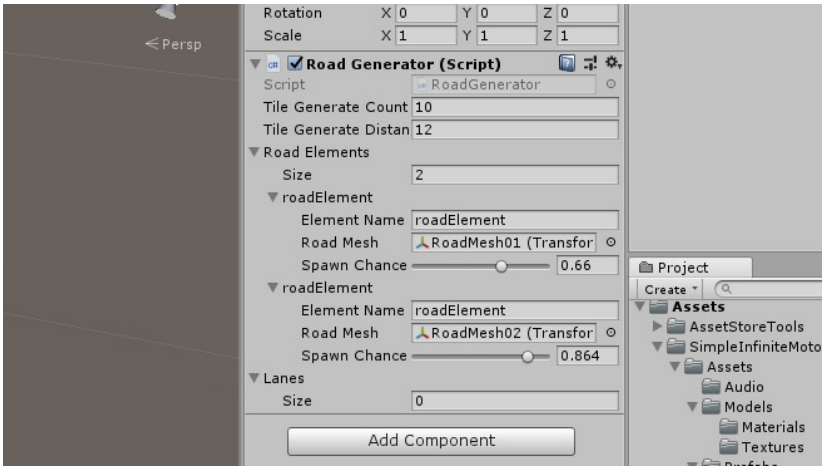
Now define the RoadElements Size to the number of RoadElements you make, for now I will set It to 2.



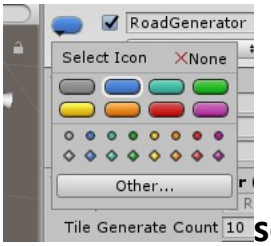
Each element contain 3 definition, **Element name** is **just for identity here** and no more function so define something for yourself or leave it alone.

Road mesh is a transform containing **MeshLenghtFinder** Component on root of it, and it's our Road Element as part of the road, we talk about how to make road element in the previous section. For now you can use some road elements from the prefabs or use your own road elements, there are 4 element and we pick two of them for now, from this location: **Assets** > **Prefabs** > **Roads** > **RoadMesh01 to 04**.

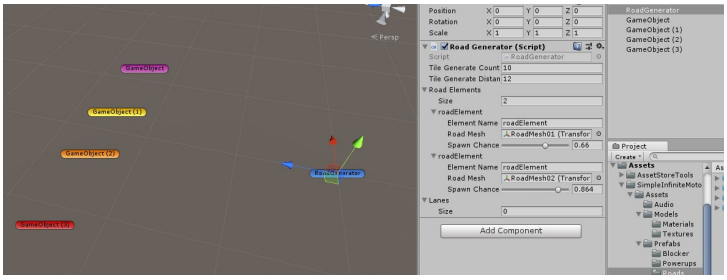
And The **Spawn Chance** is set the chance of spawning of each element. Higher value more chance to spawn. Do not use one it may get all the chances alone.



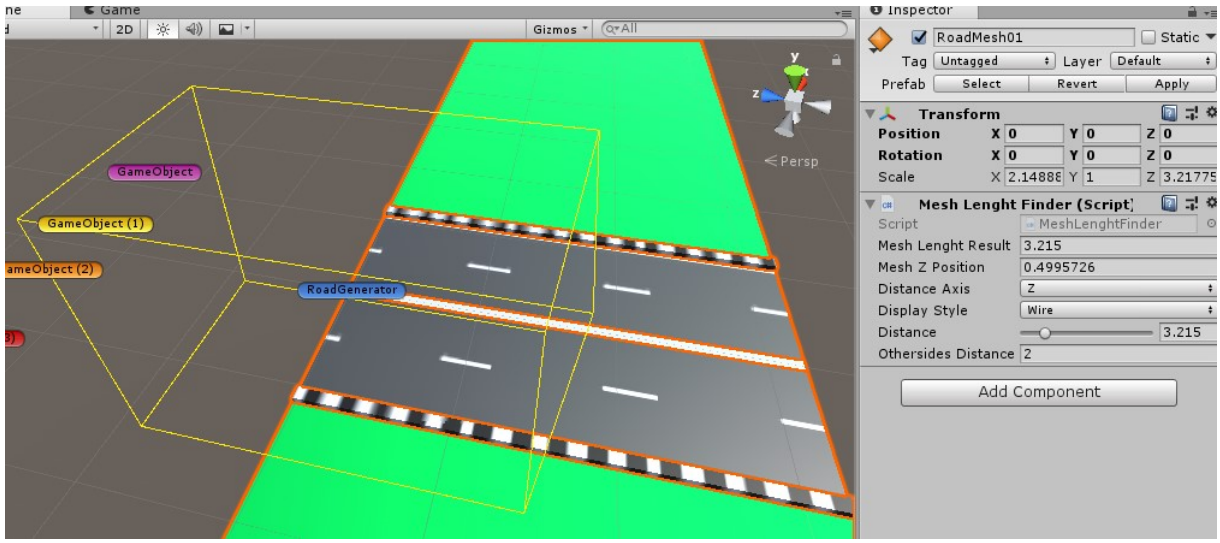
For the elements everything is setup, but for the lanes we need to do something. Create 4 new empty gameObject, each empty gameobject is for each lane. In our roadElements we have 4 lanes so **we need 4 empty game objects**. For setting up the lanes, the best practice for correct positioning of the lanes is first **enabling the icon for each game object** by selecting the icon sign on top left of inspector.



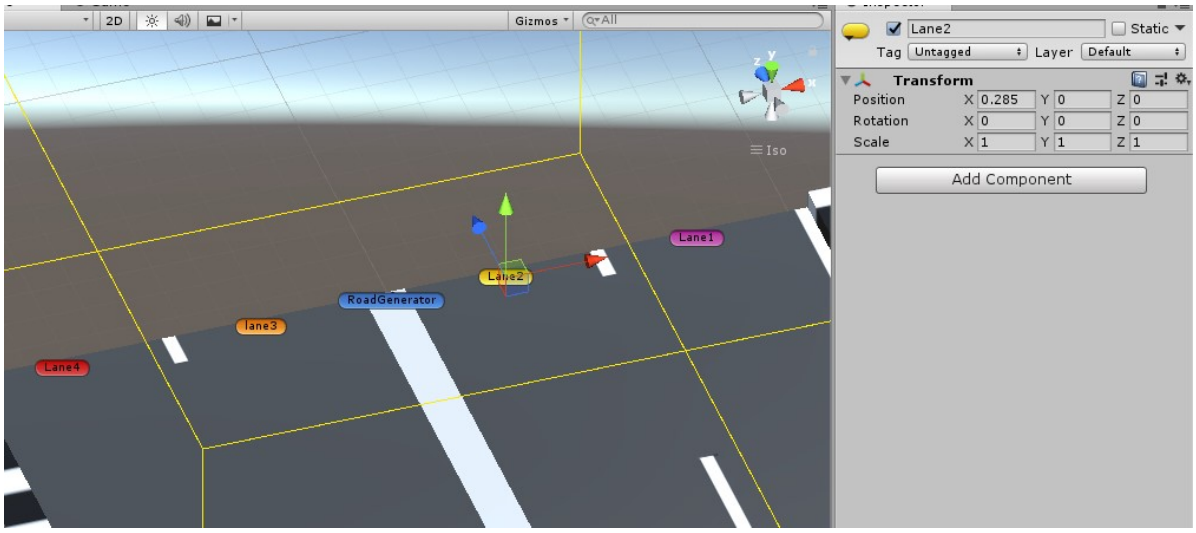
select icon from topleft of inspector view



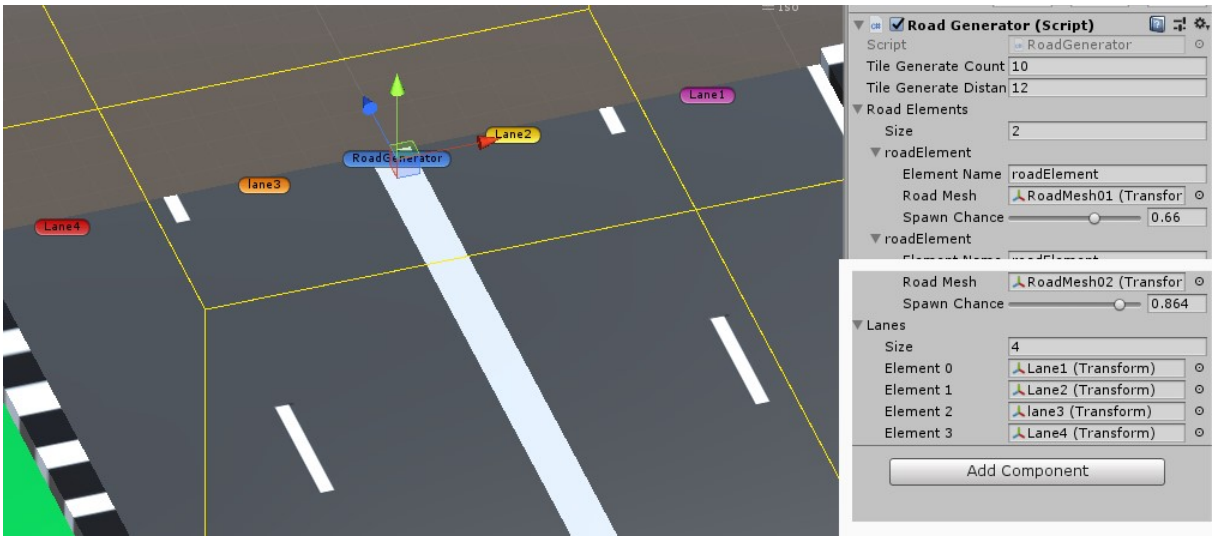
Now drag and drop one of the prefabs from roadElements into the scene, I will pick RoadMesh01 from prefabs folder and then set the position of the roadMesh transform to 0,0,0 (it's mandatory for alignment)



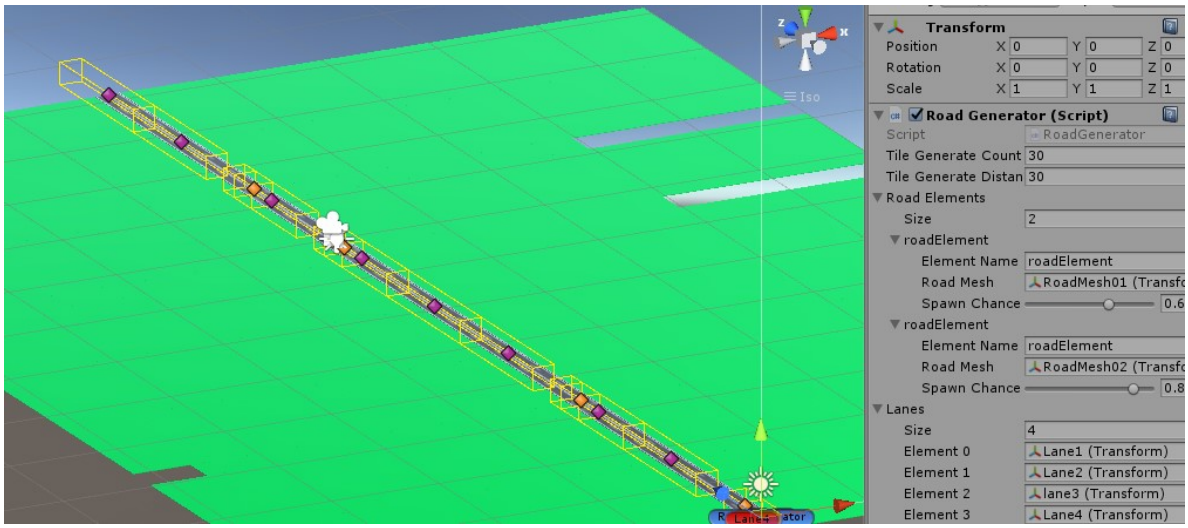
Now place the empty gameobjects on center of the road mesh lanes and then rename them to the desired lane name. the Z and Y position of this game object should be set to 0 and you should just align the X axis to the center of the road lane.



And set the lane game objects inside the road generator, so set the lane size to 4 and then put the lanes inside the definition slots, the order isnt important but when it's ordered we can be better to access.

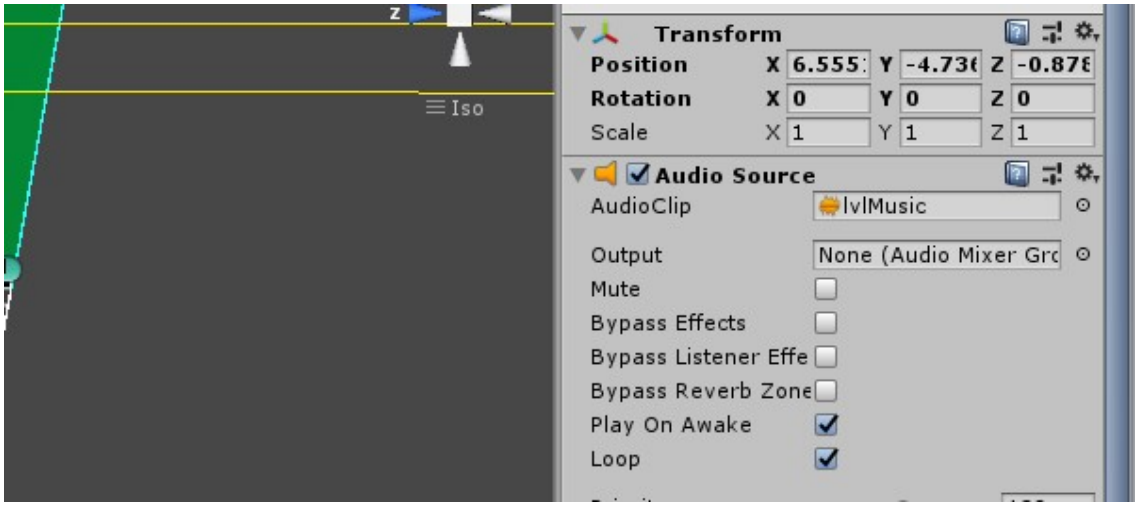


The road generator is setup, now it's require player to generate the road. For testing you can put the player prefab from the prefabs folder to the scene and align it to the road Y axis the play to see the road will generated. Don't worry about the player camera when testing we talk about that later.



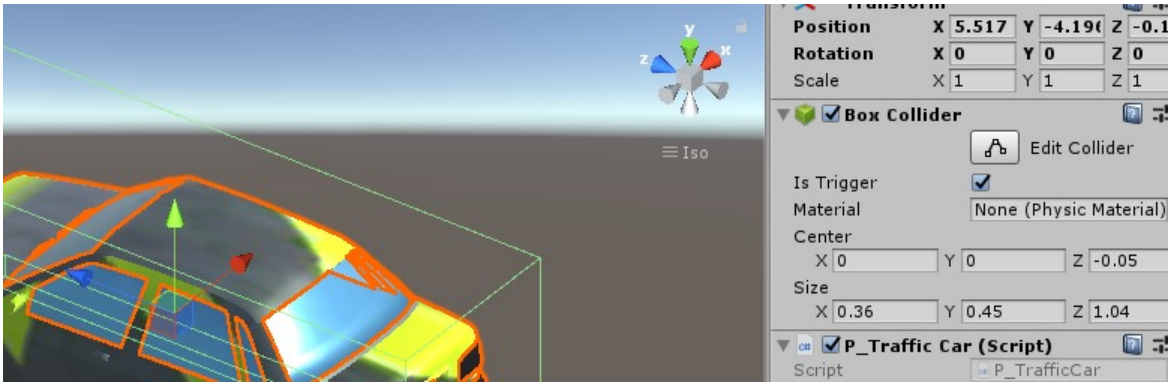
Level Scene Music

It’s simple, just create an empty gameobject inside your level, add an AudioSource component set the cip from available musics, set volume, enable Play on Awale and loop, now it’s ready.



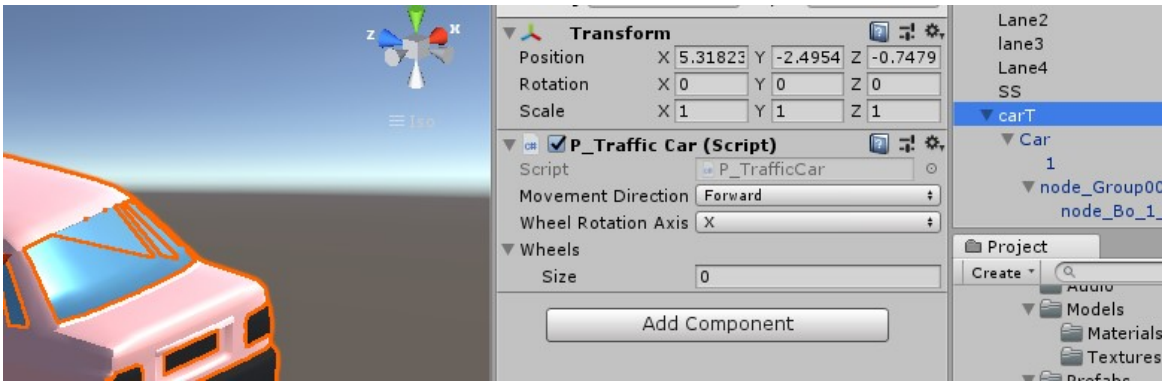
Spawnable: Car

For creating a car spawnable you need a car mesh, create an Empty gameObject then place the car mesh as child of this empty gameobject, then set the position of the car mesh to 0,0,0 in local of the gameobject and then add a **rigidbody on** the **gameobject root**, then put a **Box Collider on** the **gameobject root** and **set** the **position** and **size** and turn on **the Is Trigger**.



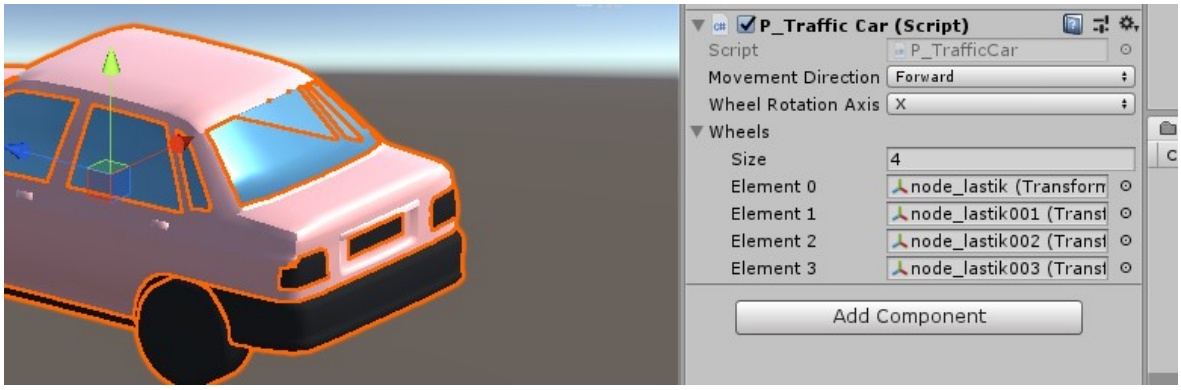
Add P_TrafficCar component on the gameobject from:

Noir Project > **Simple Infinite MotoBiker** > **Spawnables** > **Traffic Car**



Now set the **Movement Direction**, for now in here we use **Forward Axis**.

Then set the wheels and rotation direction, this car have 4 wheels so we need to set the size of the wheels to 4 and then define eachWheel Separatly on the transform. Also we need to set the **wheel rotation direction axis** too. Here for our model we need to use **X axis.**



The car spawnable is ready for use inside the spawner, please create a prefab from this gameobject for further use.

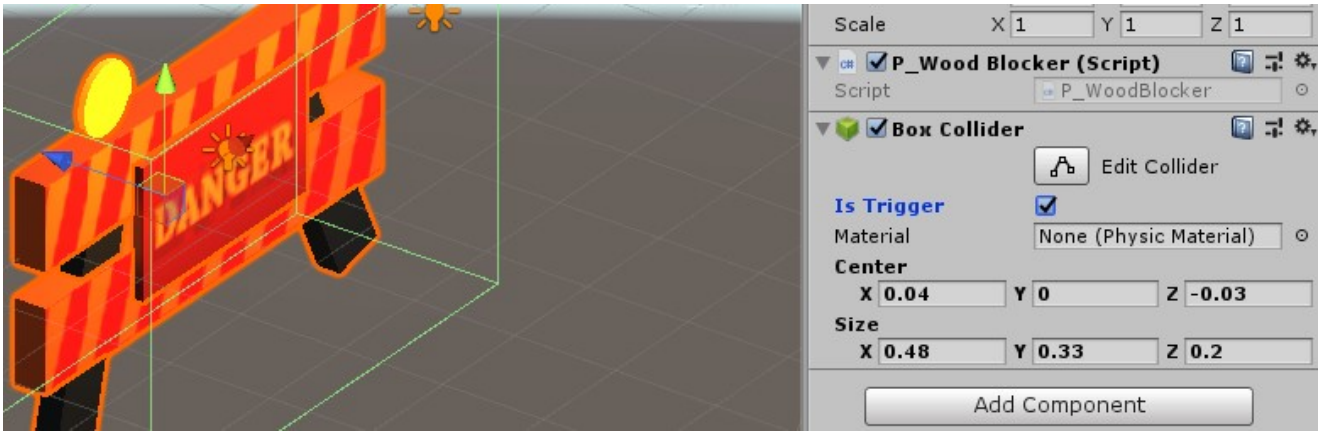
Spawnable: Blocker

The roads blocks are the same as the cars in use but different in properties. Now lets create a new Empty Game object then add P_WoodBlocker component on the gameobject from:

Noir Project > **Simple Infinite MotoBiker** > **Spawnables** > **Wooden road block**

Ok place your road block mesh inside the gameobject, or create it there with the mesh objects then set the transform position to 0,0,0.

Now it's time to set the collider. Place a **Box Collider component** on the empty root gameobject and set the size and position of the collider to cover the blocker, then **enable the Is Trigger value**.



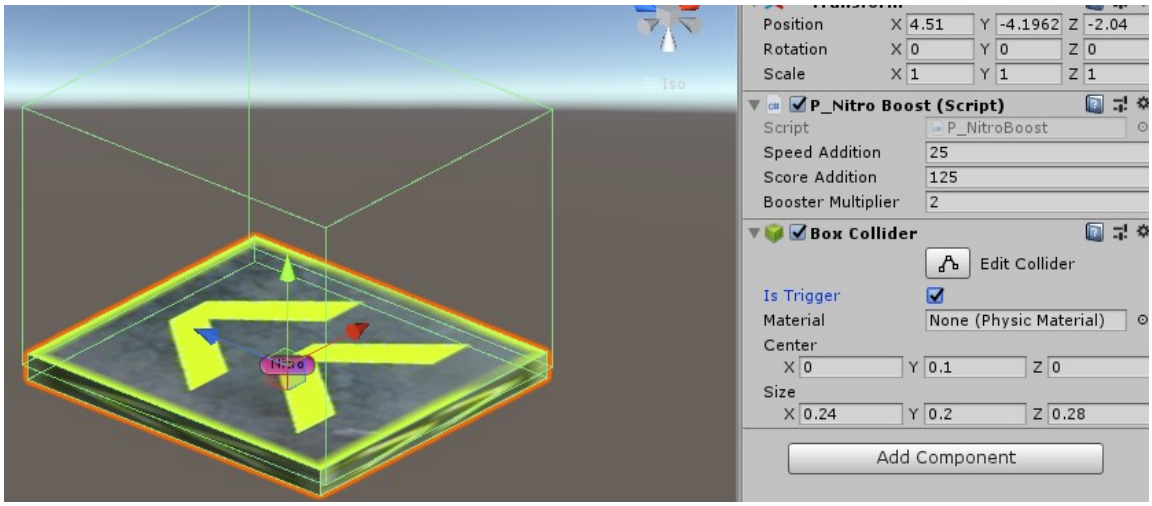
It's setup, now save it as a prefab for further use. You can have collider on the mesh childrens for blocking purpose also you can add the rigidbody too but it should not be important on this item .

Spawnable: Nitro Boost

The nitro boost surface is also the same as the other spawnable items, for creating new Boost Spawnable item, the requirement is an Empty new gameobject. Then add P_NitroBoost component on the gameobject from:

Noir Project > **Simple Infinite MotoBiker** > **Spawnables** > **Nitro Boost**

Now place the nitro mesh inside the gameobject and align it to 0,0,0. Add a **Box Collider** on the gameobject and resize and reposition the collider to cover the area player can move to it, set the Is **Trigger enable**, and trigger is set now.



Now the Settings:

Speed Addition is the value of speed added to the player when cross over the trigger.

Score Addition is the value of how much score added to the player when collider with the nitro trigger.

Booster Multiplier is the amount of how fast nitro booster reach player to the desired speed.

Now the nitro is ready so create a prefab from the gameobject for further use.

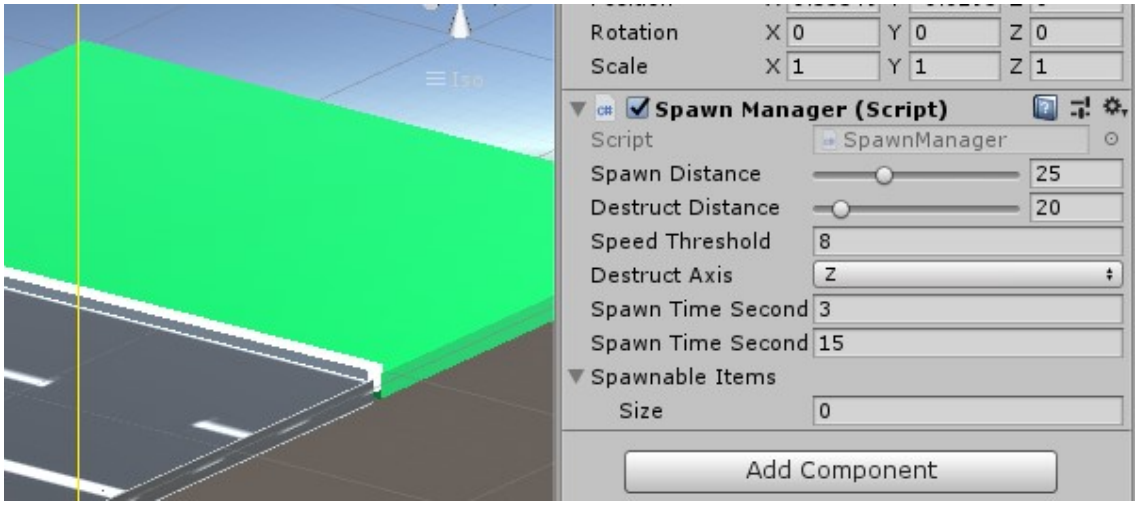
Spawner (Car, Boost, Blocker, etc.)

Spawner spawn the road spawnable items to the road lane, you need to set the spawner up, you can also add one or more spawner to spawn items simultaniously in different time.

For creating spawner we need an Empty GameObject, so create an empty one and call it something that make sense.

Now add the Spawner manager component on the gameobject, the component location is:

Noir Project > **Simple Infinite MotoBiker** > **Spawner**



Spawn distance is set distance to spawn the items from the player. The 25 is good for now.

Destruct distance is the distanc between the item and behind of the player when the player pass over the element for desired distance unit value, the object removed from scene because it's useless. Now we set it 20

Speed threshould is the value for start spawning the items, the value is calculated from the minimum player speed + speed threshould, for example when threshould set to 8 and minimum player speed is 10, the spawner works when player speed is 18+, the value 8 is good for now.

Direction Axis is the axis to calculate the destruct and spawn distance in selected axis. For now we use **Z**

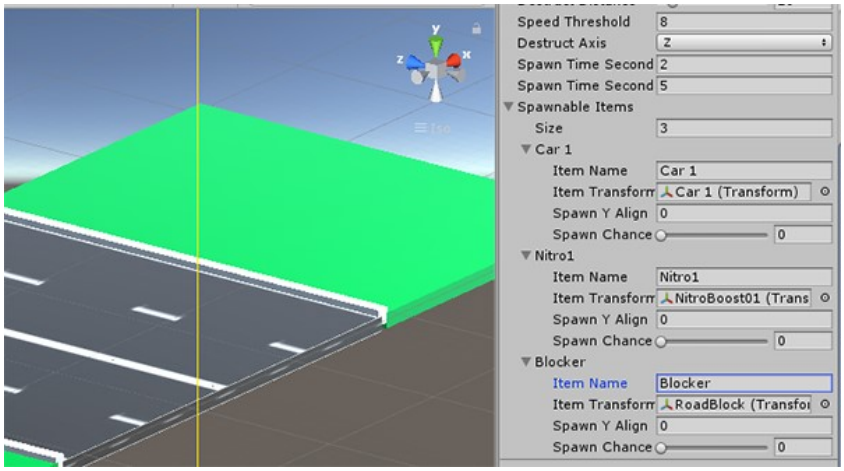
Spawn Time Seconds Min and **Spawn Time Seconds Max** are the value in time seconds for spawning the spawnable items, the value is randomly selected from the min and max value. We set the **min to 2** and the **max to 5** for now.

Spawnable Items are our items to spawn in the prev. section we talk about how to make the spawnables in different kinds, now we can use them in here. Now we use some of our available prefabs such as **RoadBlock, NitroBoost01, Car 1**, all of this prefabs are available at **Assets > Prefabs > [Powerups, Teraffic, Blocker]** folder.

also prefer to spawn cars with another clone of spawner for better performance, for now we use just one spawner for all of the spawnable items.

The **Spawnable Items** have some definnitions, we set the **size of spawnable items to 3** and then **drag and drop** the above **spawnable prefabs to Item transfrom slots**, then set another settings.

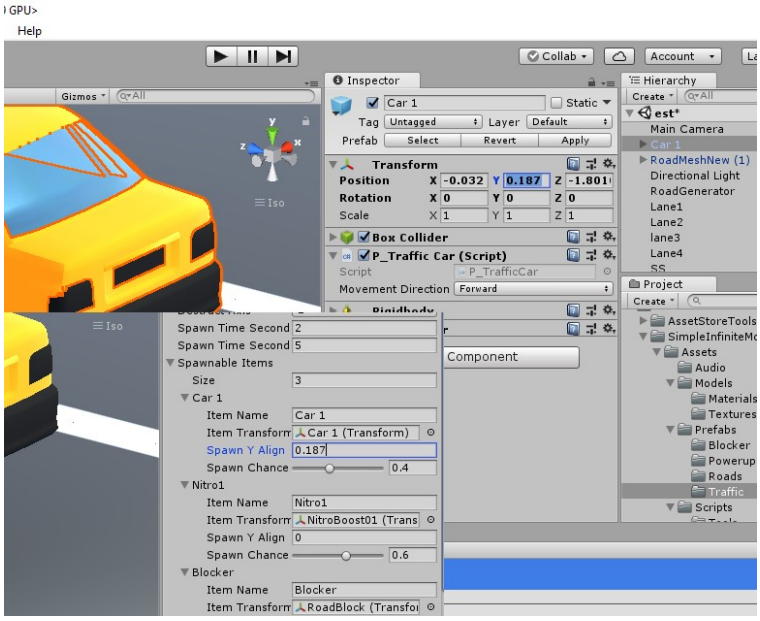
Item name in **Spawnable Items** is for naming and ease of access also the **instantiated mesh** on the scene **also called with this name** with some prefixes.



Item transform is the spawnable mesh you learnt to make in prev. sections. You sould select one of the created or template spawnables and define it in here.

Spawn Chance is the same as the road chances, this will set the **chance of item spawn**. So we set the chances too.

And the key element here is **Spawn Y Align**, this value designed because the road surface is may upper than the **origin Y** or mesh maybe miss aligned or etc. just as we say before we can **put the road mesh on the scene** and then set the **position to 0,0,0** for aligning everything, for knowing the element Y position, just **put the spawnable item on the road** and then **align the Y axis** of the mesh on the road, for example: place **Car 1 prefab on the road** and **align the tires**. Then check the Transform Y position and place the Y value inside the Spawn Y Align.



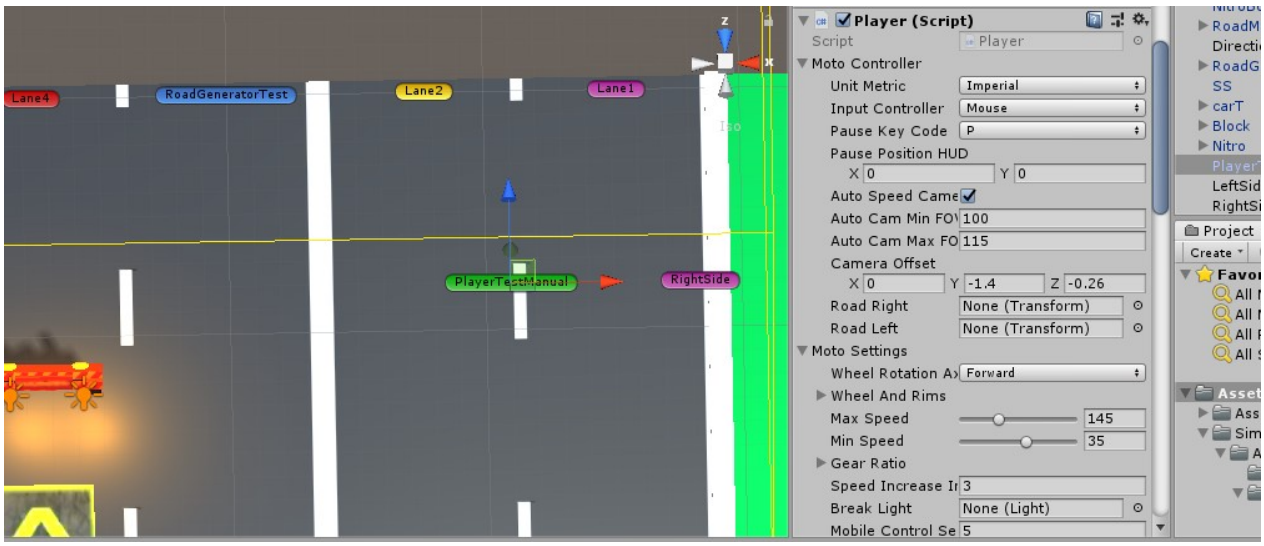
Do the same for the other spawnable items and then everything is ready here.

Setting up the Player

The Gameplay, controls, and almost all of the game mechanics are inside player component. Lets start to **make a new Empty GameObject** on the scene **to start creating the player** and configure the settings.

Add **Player Controller** on the Empty gameObject from:

Noir Project > **Simple Infinite MotoBiker** > **Player Controller**

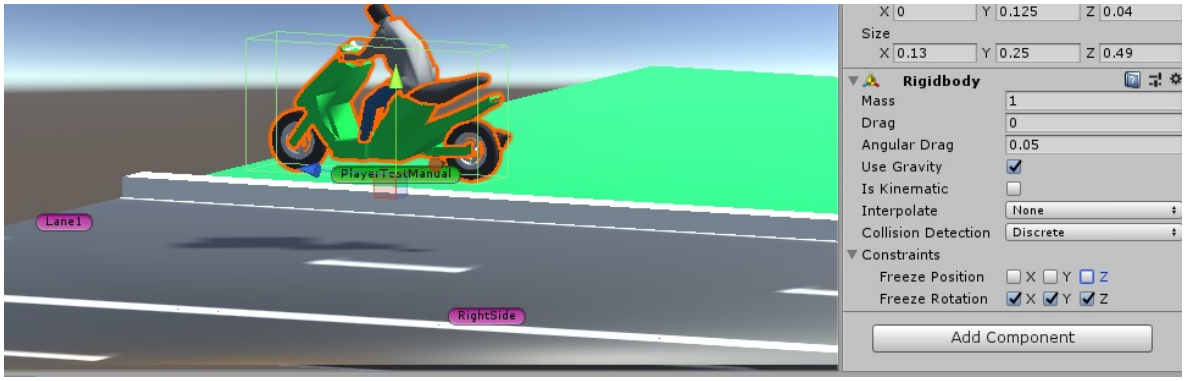


Before we start to configure the player controller, we need to **create 2 new Empty Gameobject** for the road side, the **first gameobject should align on the left side of your road** and **the other one should be aligned on the right side the road,** the position **alignment should be applied on X Axis** and the **Y and Z Axis should be set** to **0**. Name them LeftSide and RightSide (optional).

Define them on the Player Controller component [inside the MotoController].

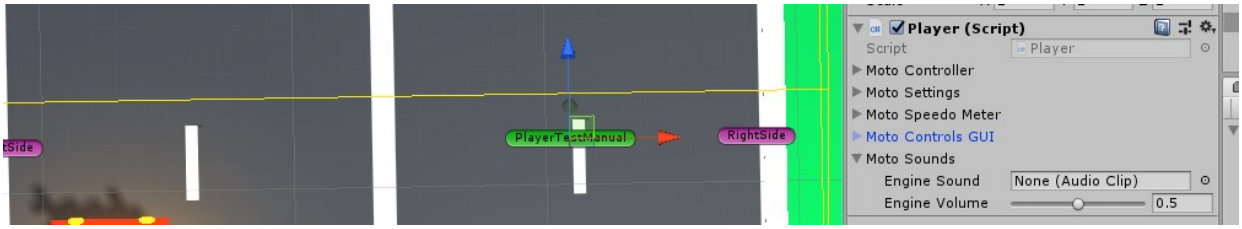
Now the player controller knows the side road and the player cant move over this two indicator. Then add the motor bike 3d model inside the player component you made now, so the player object is parent of the motorbike mesh. After adding the motorbike mesh you need to rotate the motor bike front to align with forward axis (Blue Axis - Z), then create a prefab from your player for further action.

Then add a collider to bound your moto model, and add rigidbody on the player gameobject root. Please set the **Freeze rotation Axis on X, Y and Z enabled**, on the **rigidbody** of player.



Ok initial setup is completed. Now we will setup the player controller.

Take a **close look on the player script**, the player controller have 5 sections.



1. Moto Controller,
2. Moto Settings,
3. Moto Speedo Meter,
4. Moto Controls GUI,
5. Moto Sounds

Player – Moto Controller

Player controller definitions are inside this section, we start to configure from top to bottom.

Unit Metric is for set the the distance calculated by **km/h** or **mph** the settings contain Imperial for Miles and Metric for Kilometer.

Imput controller , when we use Mouse, the game controller is set to mouse mode, so the LMB is for acceleration, RMB for break, P for Pause, and mouse movement is for bike movement, the pedals are dissappeared from the screen too.

Input controller contain to choices, Mouse and Accelerometer. The accelerometer is for controlling the bike with mobile device, the acceleration and break pedal are activated on screen, and the game controller completely change to device mode.

Select the desired controller now. I prefer mouse if you are not already connect any android or ios device. To the unity editor.

Pause Key Code the key code for pause on keyboard can be set.

Pause Position HUD is offset calculated from the top left of the screen, this X, Y values are defines the position of Pause UI when pause called. Now set the X to 10 and the Y to 180.

Auto Speed Camera if you would like, you can make this enable, this feature increase the camera FOV when your player speed up, and the camera can show farther road on the screen, this feature help player to see more of the roads and traffic and the FOV min and max value can be set through the two next values. Now Enable the feature.

Auto cam min FOV is the minimum value of the camera FOV for Auto camera speed, set it to 100 for now.

Auto cam max FOX is the maximum value of the camera FOV for Auto camera speed, set it to 115 for now.

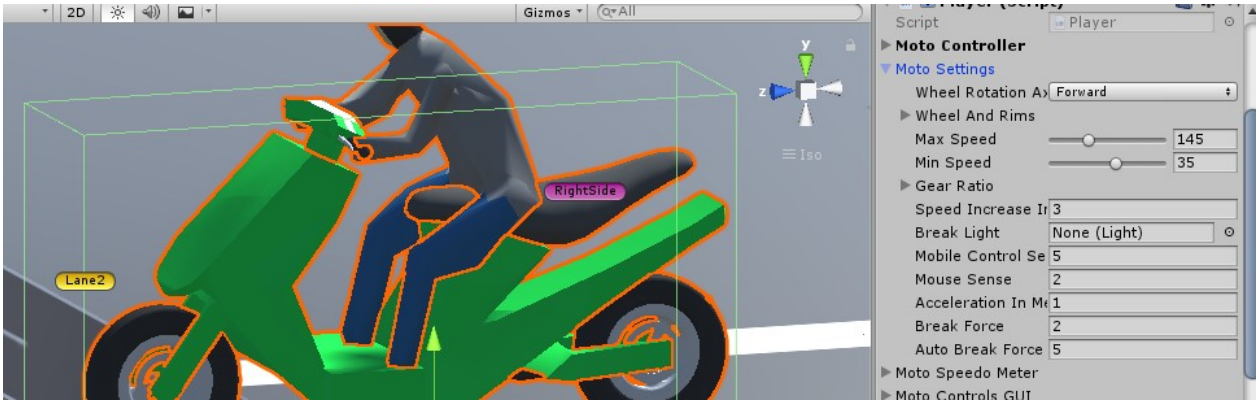
Camera offset is the distance of the camera from the player, set it to (0, -1.4, -0.26)

Camera Rotation offset and this value define the rotation offset of the camera from the current camera rotation, if your main camera id in transform position (0,0,0) and the rotaion of (0,0,0) then set the rotation offset to (60,0,0) for now

Road Left and Road Right we talk about them at the start of the player section, but this transforms require empty gameobjects to defile X axis position of road left and right.

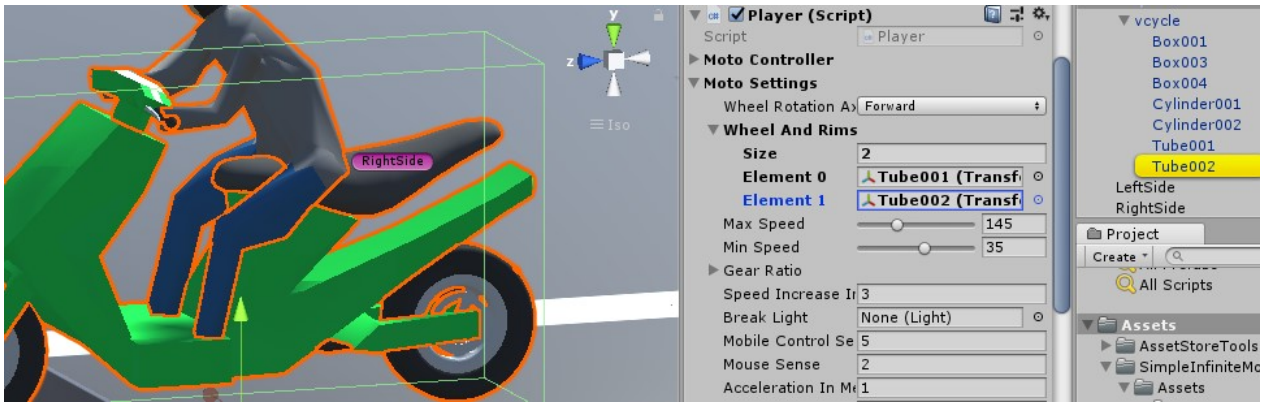
Player – Moto Settings

Player Minimum and maximum speed, Gearbox definitions, Break light, and controller sensitivity, … is here, take look at the moto Settings on player controller component.



Wheel Rotation Axis this will handle which direction the wheels should be rotated, for each model you shoud use one this axis, Forward, Right and up. Now for this model I will set to forward.

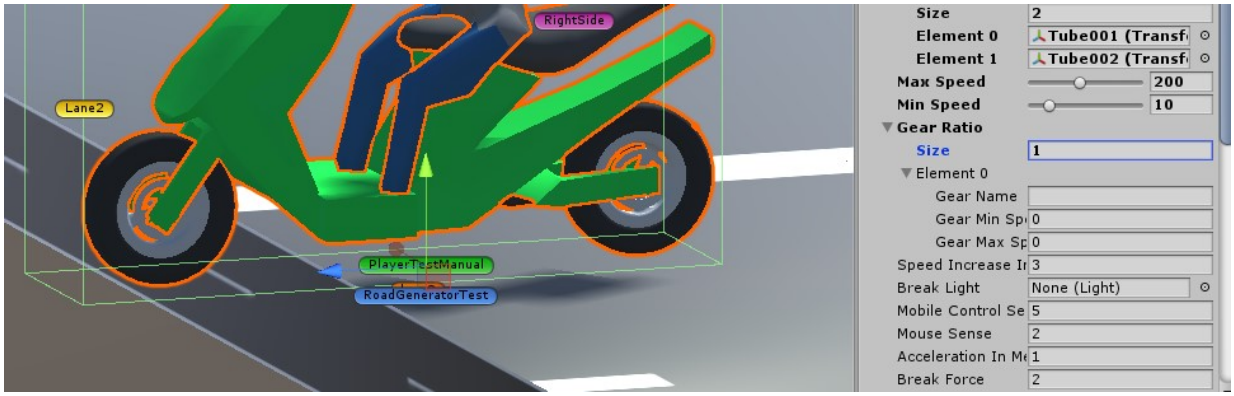
Wheel and Rims is array of wheels, the wheel models should be added separatly in this array. The rotation speed is increased by motor speed. And slow down when it breaks. Because may you want to change 2 wheel motorbike to 4 wheel or 3 wheel cycle, this is an array that can contain any number of wheels. So I will define front and read wheel transform the the Wheel and Rims.



Now it's time to set min and max speed of the player.

Min speed is the minimum speed player can travel, as you know in this kind of games the player always moves so I will define 10 for minimum speed. For moving the player slowly forward.

Max Speed is the Maximum speed of player can be traveled. Now we set the maximum speed to 200 because our speedometer template is now set on 200 for maximum.



Gear Ratio is for gearbox setting, we want to define 4 gear for now, and 1 for Neutral Gear, so we define the Gear Ratio size to 5.

Before we degine the Value, we should know what is these 3 definition each gear element has.

Gear Name is for HUD/GUI of the gear inside the speedometer. You sould name it in single character with the gear names such as N, 1, ,2 ,3 ,4.

Gear Min Speed and Gear Max Speed each gear has minimum and maximum speed, the lowest speed in this gear and the highest speed.

If you would like longer gear change time you can do a trick for lower the gear change speed with this values. For example you can set the max speed of gear 1 to 45 and set the min speed of gear two to 44 then the neutral gear do much longer to change the gear. It's tricky but it works.

Now we configure all gears like table below and the gears are setup.

Gear Name	Min Speed	Max Speed
N	0	1
1	1	40
2	41	95
3	96	143
4	144	200

Speed Increase in Time this value is how much speed increate in delta time when player press the accelate button. The value is currently set to 3.

Break Light If you would like to turn on the break light on rear of your motorcycle when player break, you should create new point light and then place it inside the player gameobject so the light accept the player gameobject as the parent, then assign the point light to this field,, and for break light everything is setup, just don't forgot to make it red and set the desired intensity for the light ;)

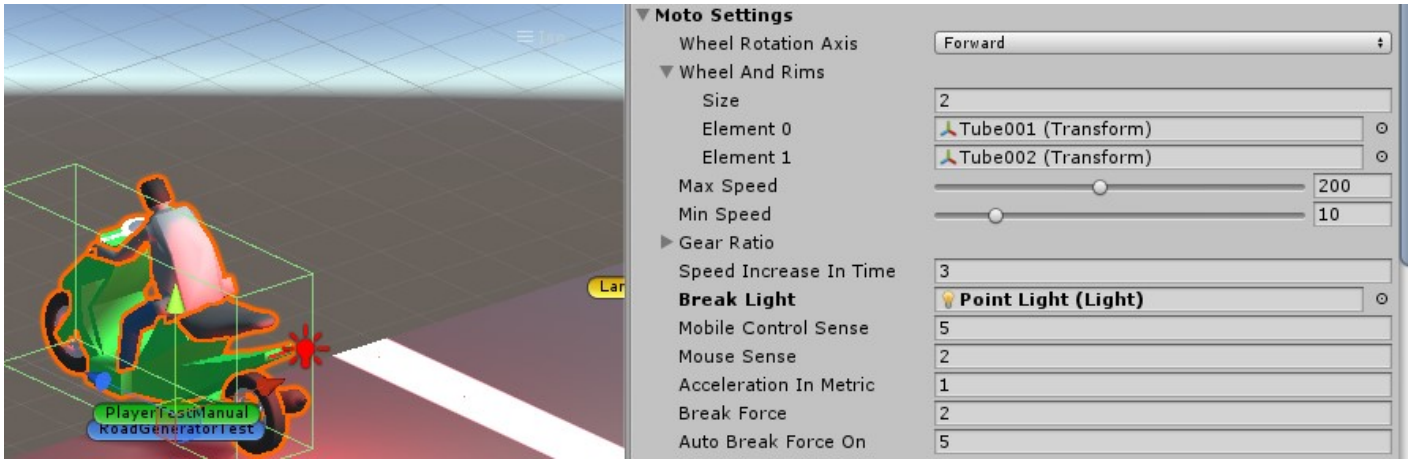
Mobile control sense this is the accelerometer sense on your device or on your mobile device accelerometer, higher value for faster player movement and lower the value for lower player movement sense on mobile devices.

Mouse sense this value is sense of movement with mouse, lower down to slow the movement.

Acceleration in metric this is the like an acceleration value, higher value faster top speed reach.

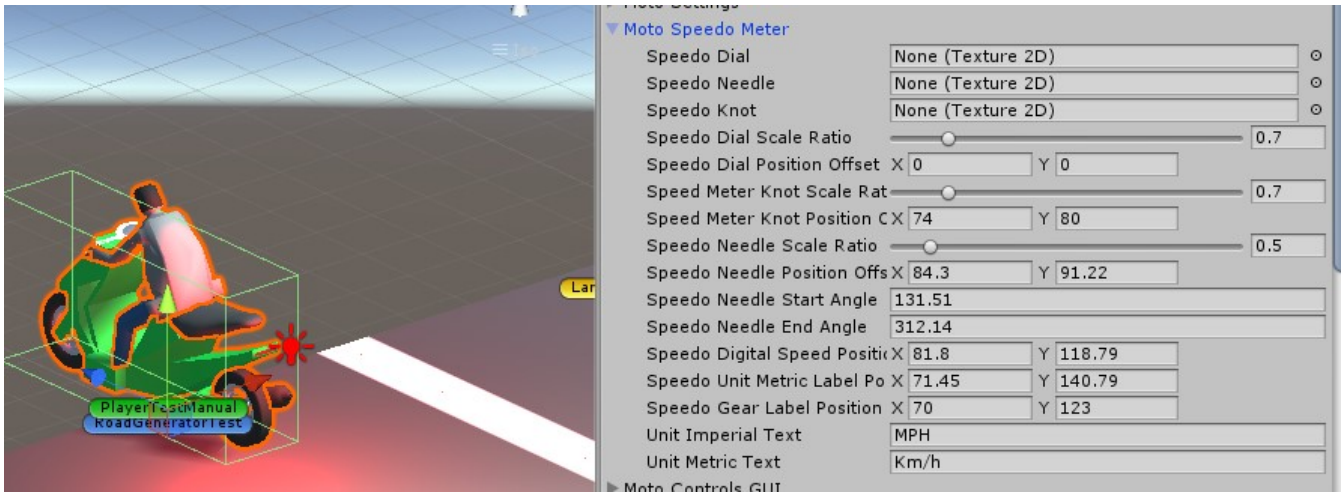
Break Force how much speed reduce when break pressed in delta time.

Auto Break Force On it define the auto break force to break in time, for example when nitro boost slowdown happen when the counter reach this number, for example if set to 12 each 12 deltatime one break force is applied.



Player – Moto Speedo Meter

This section is dedicated to on screen speedometer settings. For the configuration of the speedo meter follow the instruction:



You can make a custom made dial needle and knot and setup it in here, but right now we try to setup the dial with teplate items, the template speedometer files are here:

Assets > UI Folder

Speedo Dial is the background image of the Dial. Set the SpeedoDial file from dial folder now.

Speedo Needle this is speed needle, the structure of image can be read from the needle template file, so we select SpeedoNeedle file from ui folder for now.

Speedo Knot this is a knot for the center of the dial. Use texture from ui folder called Speedoknot.

When you set this textures the default value of the component is completely compatible with these 3 textures rom template, so when you play the game speedo meter now works fine, but now I will tell you what is the next options.

Speedo dial Scale Ratio it is the Dial scale multiplier higher value is for bigger dial image.

Speedo dial Position Offset this is where to draw speedometer dial background.

Speedometer Knot Scale Ratio it is the knot scale multiplier higher value is for bigger knot image.

Speedometer Knot Position Offset this is where to draw speedometer knot background will drawn.

Speedo Needle Scale Ratio it is the speedometer needle scale multiplier higher value for bigger needle.

Speedo Needle Position Offset this is where to draw speedometer speed needle will appeared.

Speedo Needle Start Angle this value defines what angle the speedometer can start, so the 0 for speedo meter needle is this angle degree. Now it will show needle on 0.

Speedo Needle End Angle this value is for maximum degree where needle can goes. The max speed is now 200 in this template so 312.14 is the position for max speed destination of needle.

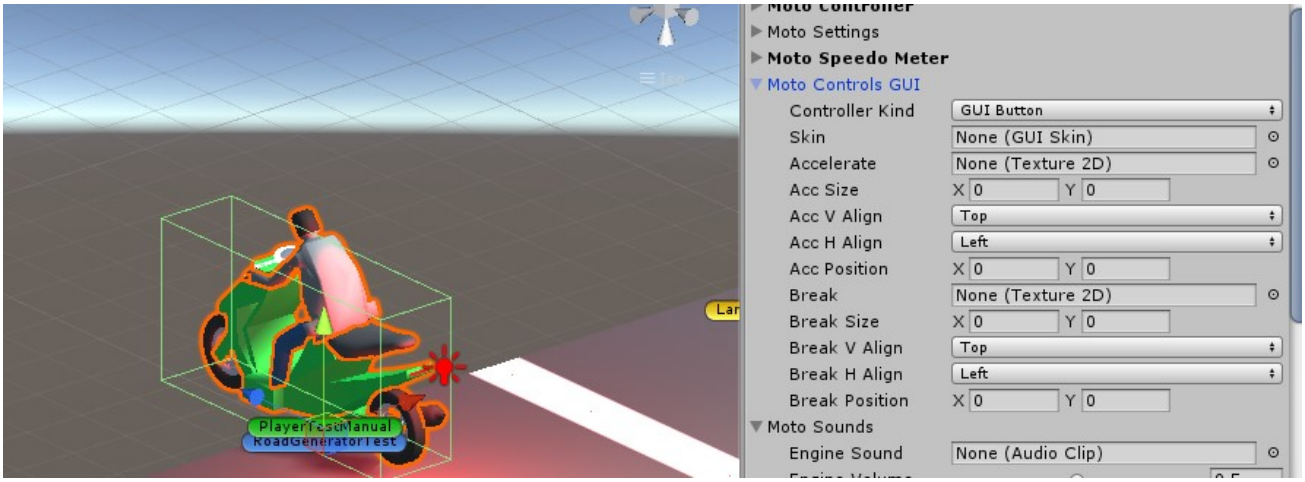
Speedo Digital Speed Position this is where to draw the digital text of speed number on the screen, it's now inside the lcd thing on the speedo background. With the default values.

Speedo Unit Metric Label Position the text for km/h and mph on the speedo meter dial this text will automatically changed by changing the player controller unit metric. And the labels are defined from **Unit Imperial Text** and **Unit Metric Text** here in this section just below Speedo Unit Metric.

Speedo Gear Label Position this value is for position to show the gear name from gear ratio definition current location is set to the location of the nottom left corner of the lcd thing on the speedo meter diag background.

Player – Moto Controls GUI

This section is for define the control pedals and how controller looks. This section is look like this:



First we should select the **controller Kind**. The controller kind has 2 type, the draw of the gui pedals can be selected as GUI Button or Canvas Panel, these are two different draw of the GUI Buttons and I Prefer **Use canvas panel on mobile device** because the touch on canvas panel is much more better.

Skin the gui skin required for the button style and works, the HUD skin file is available inside UI folder.

Accelerate is GUI texture for the Acceleration pedal, throPedal texture file template is available inside UI folder.

Acc Size this is the size of accelerate pedal in pixel on the screen.

Acc V Align this is a setting for aligning the acc pedal in vertical to top or bottom.

Acc H Align this is a setting for aligning the acc pedal in horizontal to left or right of the screen.

Acc Position this is Acceleration pedal position on the screen. Based on the H and V align.

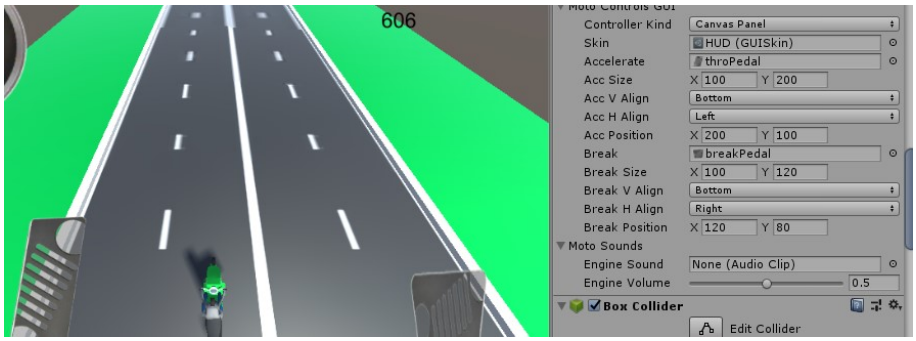
Break is GUI texture for the Break pedal, breakPedal texture file template is available inside UI folder.

Break Size this is the size of break pedal in pixel on the screen.

Break V Align this is a setting for aligning the break pedal in vertical to top or bottom.

Break H Align this is a setting for aligning the break pedal in horizontal to left or right of the screen.

Break Position this is Break pedal position on the screen. Based on the H and V align.



Player – Moto Sounds

This option is for motorcycle engine sound clip and the volume of the engine sound.

Engine Sound is the sound clip slot for define engine sound, the sound automatically pitched when player speed up. The template engine sound is available on **Assets** > **Audio** > **engine** file

Engine Volume this is volume of the playback of motorcycle engine sound.


_SLOT03
Game-RoadGenerator
_SLOT02
_SLOT01
_RightRoad
Shadownoir.com