



Documentation v1.0

Table of contents

Link

Where to begin :

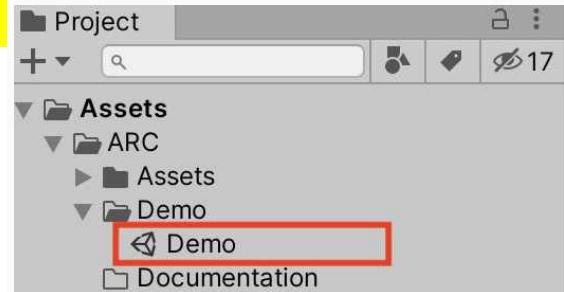
First of all, thank you for purchasing Airplane Race Creator.

Step 1:

Open scene **Demo** to try the demo.
(Project tab : ARC → Demo → Demo)

Default inputs are:

- Turn: Left | Right | Up | Down
- Booster: X
- Break: C
- Power-up: Space
- Camera: N
- Pause: P or Escape
- Validate: return/enter



Step 2:

If you encounter a problem when you import the asset. Example: All the materials are pink read:

Tuto 0: Read First → Asset Import Process
→ Page 12

The tutorial explains how to setup correctly the project.

Then read the next two sections in the documentation to learn the basics of Airplane Race Creator.

Tuto 1: Create your first track → Page 14

Tuto 2: Customize a car → Page 49

Step 3:

Have a look to the complete documentation for advance features.

If you don't find information in the documentation contact me at: targetsoundfx@gmail.com

Table of contents:

Brief description:

Doc Part 1 Tutorials: It explains how to use the asset and the main features of the asset.

Doc Part 2: All features: It contains all the elements relating to the core features of the asset (UI, vehicles, Inputs...)

Doc Part 3: Graphics and post-process effects: It contains information about graphics and post-process effects.

Doc Part 4: FAQ: It contains the most frequently asked questions.

Doc Part 1: Tutorials

Brief description:

By following the tutorials:

-You will discover the basics of how the asset works.

-You will learn the main features of the asset.

IMPORTANT: It is advisable to do the tutorials before you start creating your own game.

Page Link

Tuto 0: Asset Import Process 12

Tuto 1: Create your first track 14

- 1-Create the track path
- 2-Initialize the Start Line
- 3-Set up the scene to test the track
- 4-Test the track
- 5>Create the Minimap
- 6>Create the game area limits
- 7-Add Power-ups to your track
- 8-Add AI booster
- 9-Set up the track in the Track Manager
- 10>Create AI alternative Path

Tuto 2: Customize a car 49

- 1>Create a new vehicle
- 2>Customize a vehicle with your own 3D models
- 3>Allow and set up a vehicle to be used by a player
- 4>Go further

Tuto 3: Replace default font and music 58

- 1-Replace Font (Arial)
- 2-Replace Music

Doc Part 2: All features

Brief description:

It contains all the elements relating to the core features of the asset (UI, vehicles, save system, Inputs...)

Asset content

62

Have an overview of the content available in the asset.

Overview: Starter kit, Prefabs and ready to use contents

Managers

63

Short description of all the most important object in a scene (Main Menu and Track Scene).

Short description of each common module (all scenes)

Short description of each Main Menu scene module

Short description of each Track scene module

Short description of each tool available in the Unity tool bar

Vehicles

84

Overview

Short description of each vehicle module

Vehicle parameters: Basic

Vehicle Parameters: Advanced

How to: Create a new vehicle

How to customize a vehicle with your own 3D models

Set up a vehicle to be usable in-game

How to: Modify Player + AI UI color

How to: Modify Player + AI names used in leaderboard

How to: Modify AI behaviors

How to: (Path) Change the position followed by AI vehicle (Start Line)

Game Mode

103

Information about how to managed Championship, Arcade and Time Trial modes

Track Manager: Set up new track.

Track Manager: Set up Time Trial leaderboard.

Track Manager: Set up coins earned after a race (Arcade, Time Trial).

Championship: Create a new Championship

Championship: Delete a championship

Arcade: Choose the number of vehicles in a race

Test Mode (3 and 5 mode)

Choose the number of lap for a track

Tracks

118

Basic info about track creation is contained in Tuto 1: Create your first track section 1-Create the track path

Overview

Track path: Add a point between two points that had been already created

Track path: Remove a checkpoint

Start line

120

Basic info about Start Line creation is contained in Tuto 1: Create your first track section 2-Initialize the Start Line

Overview

Adjust vehicle positions on the starting grid

Change Start Line Collider size

Change Start Line 3D models

Init Start Line position using track checkpoint 0

Booster (Power-up)

123

Basic info about AI Booster creation is contained in Tuto 1: Create your first track section 9-Add AI booster

Overview

Vehicle Booster Parameters

Tips

AI: Booster parameter depending AI difficulty

Alternative Path

125

Basic info about Alt Path creation is contained in Tuto 1: Create your first track section 10>Create AI alternative Path

Overview

Alt Path priority

Alt Path parameter depending AI difficulty

Remove Alternative Path

Reminder

Game Area Limit

128

Basic info is contained in Tuto 1: Create your first track section 6>Create the game area limits

Overview

Game Area Limit: Move a point

Game Area Limit: Add point

How to: Create a simple limit collider

Reminder

Minimap

131

Basic info is contained in Tuto 1: Create your first track section 5-CREATE the Minimap

Overview

Create the Minimap

Modify the Minimap position, scale and background

Power-ups

134

Basic info about Power-up creation is contained in Tuto 1: Create your first track 7-Add Power-ups to your track

Create a Power-up group that can be used by AIs + PlayerPrefs

Create a Power only used by players

Customize vehicle Power-Ups parameters (missile, mine, machine gun,...)

Customize Power-up object (3D models and sound fx).

Reminder

Camera

138

Overview

How to: Change camera view order

How to: Modify the default camera views

How to: Create a new camera view

Scenes

145

Starter Kit

Scene Initialization (Order of execution Diagram)

Game Data

147

Data loaded when the game is launched (Player progression, Vehicles, coins, championship, ...)

Which object in the hierarchy loads Global Data

How to: Load custom data when game starts

Scripting: Custom method to load data (example)

Scene Initialization

149

Which object initialize the scene

Scripting: Useful methods to Initialize a scene

How to: Initialize a custom module when a scene is loaded

The load scene manager

151

Overview

Scripting: How to: Load a new scene

How to: Call a method when a new scene is loaded

The Scene Step System

154

Overview

How to: Enable/Disable a step

How to: Modify a step

Scripting: Useful methods to use the Scene Step System

Loading screen

159

Overview

How to: Customize loading screen

Scripting: Useful methods to use Loading screen

Canvas Auto Saving

161

Overview

Transition system

162

Overview

How to: Modify a transition

How to: Create a transition

How to: Play a transition

Countdown system

170

Overview

How to: Disable camera movement during countdown

UI Countdown manager (overview)

How to: Delete Countdown UI elements

How to: Add Countdown UI element

Congratulation screen

177

Overview

How to: Modify Congratulation screen

Tutorial Manager

179

Overview

Tutorial Manager Parameters

Audio

181

Overview

Scripting: Load and save AudioMixer Volumes

How to: Integrate a new music, an ambiance or an Sfx

Scripting: Useful methods to Access/ Play/ Stop a music, an ambiance or an Sfx

Spatialized Sounds

How to: Delete/Reset Audio Mixer Data

How to: Set the default Audio Mixer values

How to modify default audio sound fx (engine, power up, damage,...)

Save System

207

Overview

Save System Diagram

How to: Select the save format (PlayerPrefs or .Dat)

How to: Reset Data

Scripting: Save/Load data

Localization and Texts at runtime

211

Create and Edit texts (w_TextCreator window) : Overview

How to: Edit a text

How to: Create a new text

How to: Create a text folder

How to: Add a new language

LanguageManager: Overview

Scripting: Useful methods to access multi-language texts

Text component (CurrentText.cs): Overview

How to: Set a text object using the multi-language system (Hierarchy)

Scripting: Useful methods to display multi-language texts

Player Progression

219

This section describes the system included in the asset to save info likes unlocked vehicles, track unlocked,...

Overview

How to: Save and Load a player progression data

Overview

How to: Reset inputs.

How to: Delete existing input saved data

How to: Create a new input.

How to: Create a boolean parameter (switch button).

How to: Create a float parameter (slider).

Remap Menu: How to set up an input, a bool or a float.

Scripting: Useful methods to access the inputs

How to: Change validation button ID

How to: Change back button ID

How to: Change pause button ID

PauseManager

This section describes the system included in the asset to pause the game.

Overview

How to: Add a custom event when Pause is called

How to: Add a custom event when Unpause is called

How to: Change inputs used for Pause

How to: Choose the pages that trigger the pause

Scripting: Useful methods to access the pause state

UI

This section describes the system included in the asset to customize the UI menus.

Main Menu UI (Diagram)

Race scene UI (Diagram)

Global UI Parameters

How to: Change the main menu scene ID

Overview: The UI Page System

How to: Create a new UI page

How to: Create a button to go from one page to another

How to: Add custom page event during the initialization of a page

Overview: Open/ Close page options

Open a page: Options overview

[Close a page: Options overview](#)

[How to: Add an event when a page is opened](#)

[How to: Create method that can be used as an event when the page is opened](#)

[How to: Add action when a page is closed](#)

[How to: Create method that can be used as an event when the page is closed](#)

[Leaderboard](#)

264

[Overview](#)

[How to: Choose the number of entries saved in the leaderboard.](#)

[Scripting: Access Player name and time](#)

[Scripting: Access Player name and time after Time Trial race](#)

[How to: Customize leaderboard save process.](#)

[How to: Load custom data in the leaderboard](#)

Doc Part 3: Graphics

It contains information about graphics and post-process effects.

Rendering	269
------------------	-----

Lightmaps settings

Reflection Probs

Global Post Effects (camera)

Local Post Effects (camera)

Night

Glow

Optimization	276
---------------------	-----

Occlusion Culling

Environment Prefabs	277
----------------------------	-----

Overview

Tips

Unity Terrain	278
----------------------	-----

Overview

Terrain Setup

Add and connect new terrain

Terrain Tips

Screenshots	283
--------------------	-----

How to create thumbnail and full screen images for track and championship

Shadow Distance	285
------------------------	-----

How to: Edit the shadow distance depending the scene

Doc Part 4: FAQ

It contains the most frequently asked questions.

FAQ	290
------------	-----

Asset Import Process

This section explains how to create a new Unity project to use the Airplane Race Creator asset.

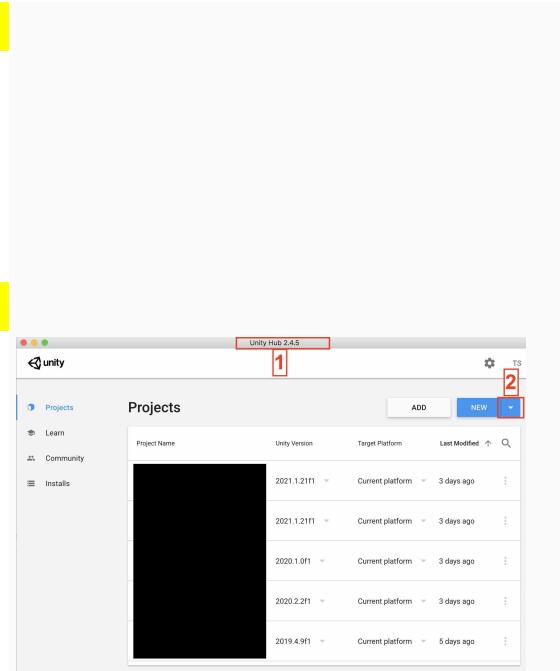
Overview

To set up correctly the project we are going to:

- Create a new empty Unity 3D project.
- Import the URP Package.
- Import the Asset.

Create an empty Unity 3D project

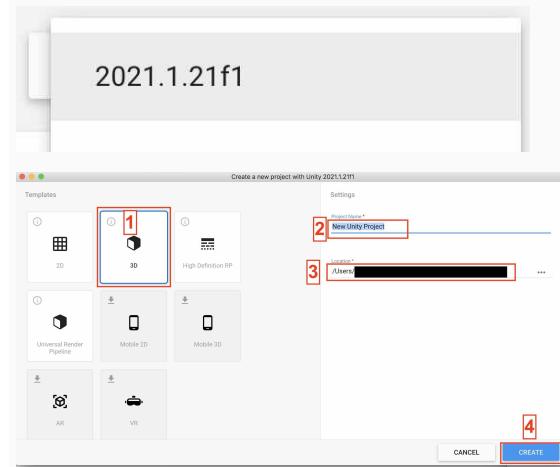
- Open **Unity Hub** (spot 1)
- Press the **arrow** button (spot 2) next to **New** button.



- Select the Unity version **2021.1.21f1** or later.

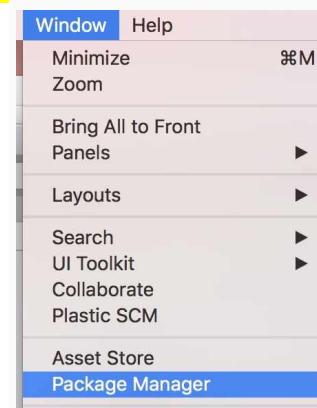
Unity version 2021.1.21f1 is the version I use to test the asset.

- Press **3D** button (spot 1) to choose an empty 3D project.
- Choose the name of your project (spot 2)
- Choose the location of your project (spot 3)
- Press **Create** (spot 4)



Import URP Package

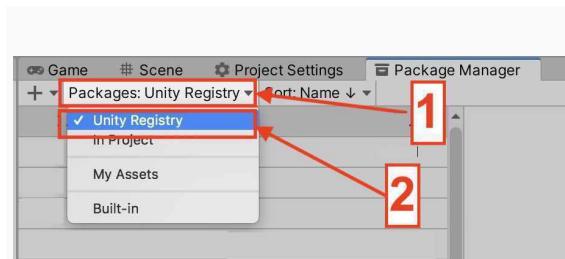
- In Unity go to **Window → Package Manager**



A new window appears.

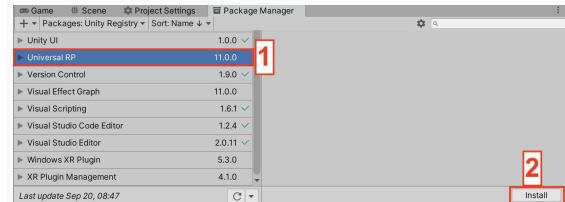
In the dropdown menu (spot 1)

-Select **Unity Registry** (spot 2)



-Select **Universal RP** in the list (spot 1)

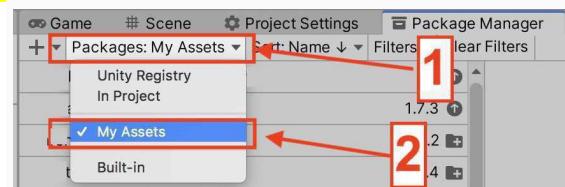
-Press **Install** (spot 2)



Import the asset

In the dropdown menu (spot 1)

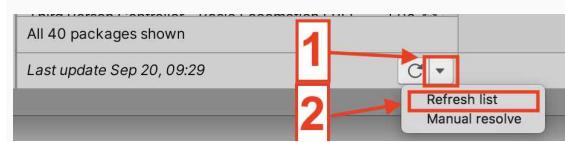
-Select **My Asset** (spot 2)



-On the bottom of Package Manager window press **arrow** button (spot 1).

-Select in the dropdown menu **Refresh list** (spot 2)

-Press **Load** button (spot 1)



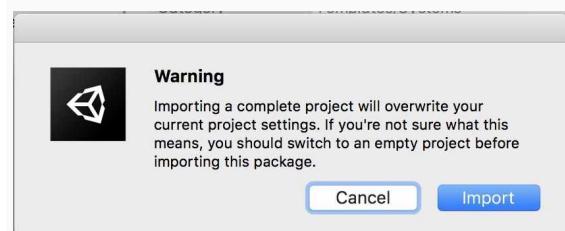
-Select **Airplane Race Creator** in the List (spot 1).

-Press **Import** (spot 2)



When the window named **Warning** appears:

-Press **Import**.



Conclusion

The asset is now setup.

If you have a problem send me an email to help you find a solution.

Tuto 1: Create your first track

This section explains all the steps to create a track.
You will find more advance information about all the features included in the asset in the Doc Part 2: All features (Page 62) .

Tutorials Parts:

1-Create the track path	Link
2-Initialize the Start Line	Link
3-Set up the scene to test the track	Link
4-Test the track	Link
5>Create the Minimap	Link
6>Create the game area limits	Link
7-Add Power-ups to your track	Link
8-Add AI booster	Link
9-Set up the track in the Track Manager	Link
10>Create alternative Path	Link

Note:

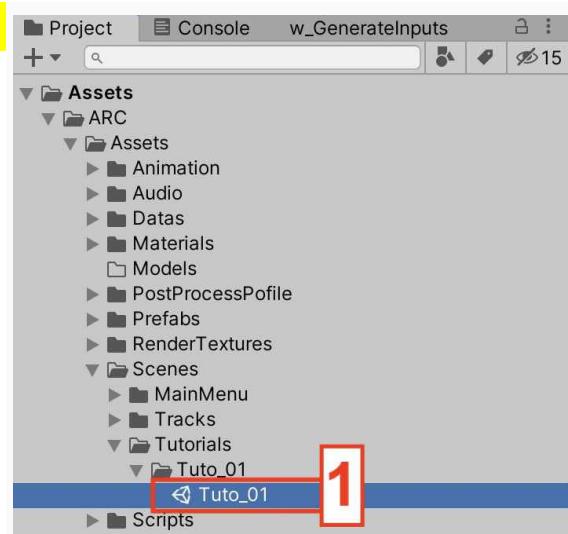
Part 1 to 4 explain how to create a track. It takes about 20 minutes.

Parts 5 to 10 explain all the main features to create tracks. It takes about 45 minutes.

1-Create the track path

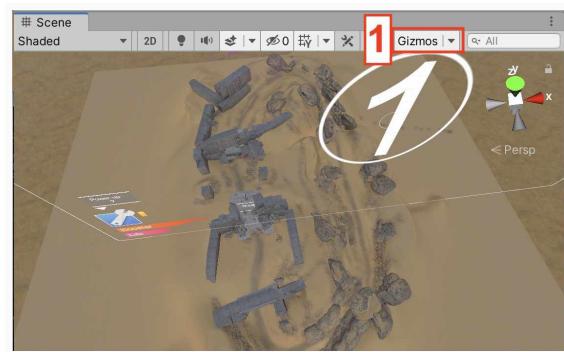
For this tutorial, we are going to use the scene [Tuto_01](#).

-In Project tab open [Tuto_01](#) scene (spot 1)
(Project tab: Assets → Scenes → Tutorials → [Tuto_01](#) → [Tuto_01](#))

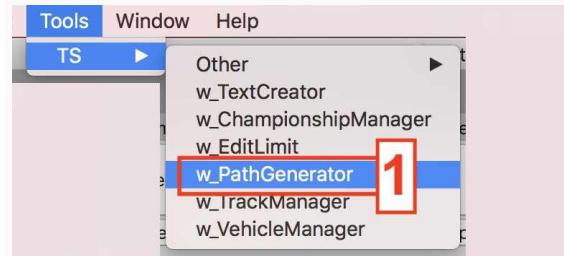


In the **Scene view** press **Gizmo** button (spot 1) to enable gizmos.

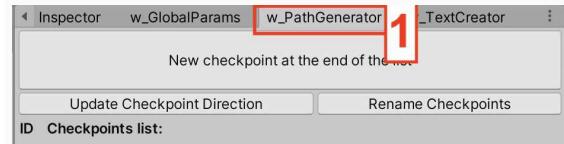
The button should be "White" when gizmos are enabled.



-Open **w_PathGenerator** window (spot 1).
(Tools → TS → w_PathGenerator)

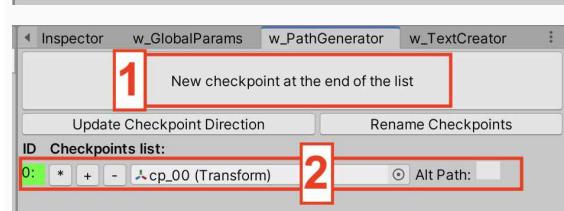


A new window appears. I suggest you to attach the window to the editor to easily use the window (spot 1).



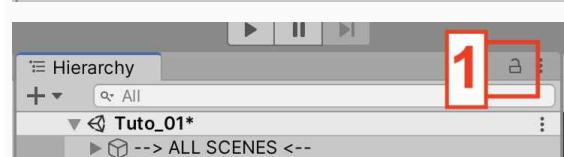
-Press button **New checkpoint at the end of the list** (spot 1).

A new checkpoint is created in the list (spot 2).



In the Hierarchy check if the **padlock icon** is **unlocked** (spot 1).

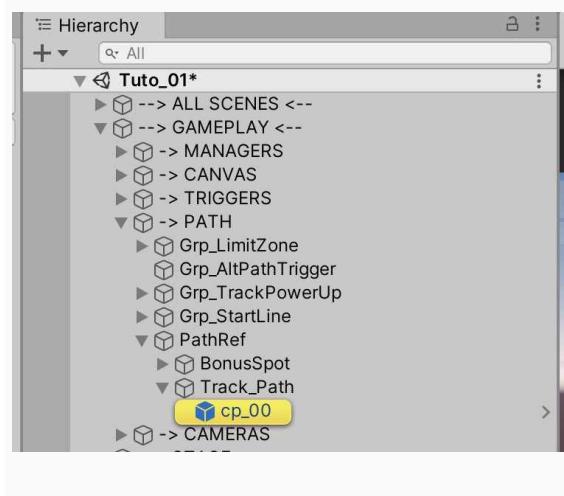
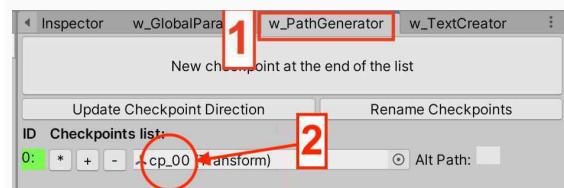
If the padlock is locked, press the padlock to unlock it.



-Go to **w_PathGenerator** window (spot 1).
(Tools → TS → w_PathGenerator)

-Click on the field that contained the first checkpoint (spot 2).

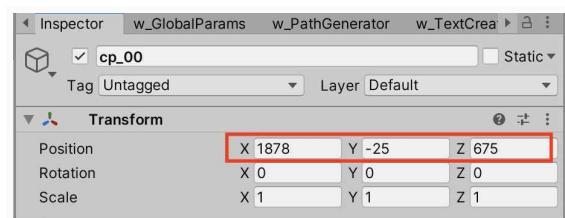
The object is shown in the Hierarchy.



-Select the checkpoint named **cp_00** in the Hierarchy.
(Hierarchy → GAMEPLAY → PATH → PathRef → TrackPath → cp_00)

Now it is possible to move it where you want in the scene.

In our example move the object to:
Position **x = 1878 y = -25 z = 675**

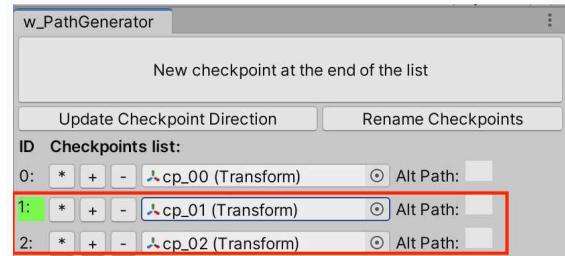


For the next 2 checkpoints use the previous technique:

-In **w_PathGenerator** window press
New checkpoint at the end of the list (spot 1)

-Then move the checkpoint using the following list:

cp_00 Position x = 1878 y = -25 z = 675
cp_01 Position x = 1878 y = -25 z = 613
cp_02 Position x = 1878 y = -25 z = 377



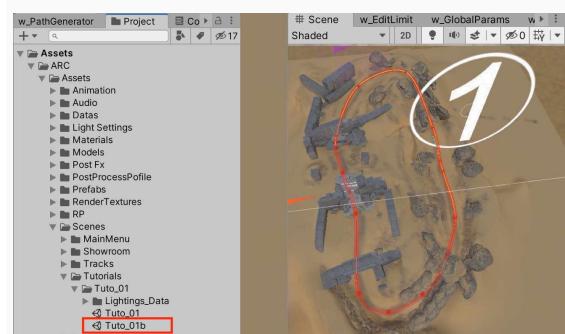
Info:

When you create a track you can move checkpoints directly in scene view.

The demo track contains 13 checkpoints.

To be sure we have the same track for this tutorial open the scene **Tuto_01b**

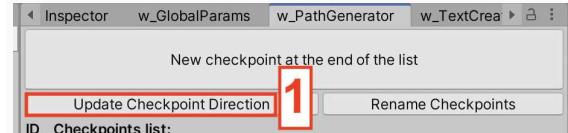
(Project tab: Assets → Scenes → Tutorials → Tuto_01 → Tuto_01b)



VERY IMPORTANT

The next step is important.

-In **w_PathGenerator** window press **Update Checkpoint Direction** button (spot 1).
Each checkpoint is now looking in the direction of the next checkpoint.

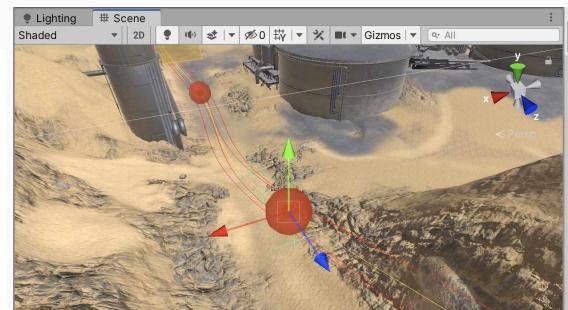


After creating a track or modifying checkpoints position don't forget to update the checkpoint direction. This step is very important because when you create alternative path, power-ups ... the system use the direction of the checkpoint. You may have issues if you don't update the direction of checkpoints.

Reminders

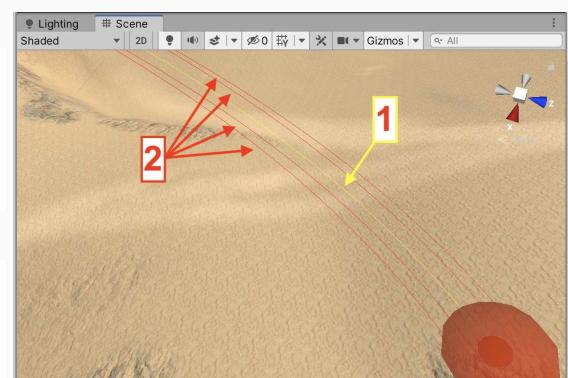
Info:

You can select a checkpoint in the **Scene View** by clicking the red sphere.



Info:

The yellow line corresponds to the path (spot 1).



Info:

AI Vehicles can move between the red limits (spot 2).

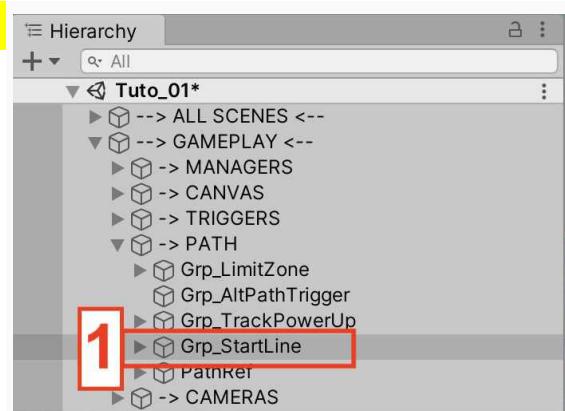
If a red line passes through a stage element (rock, wall,...) move the checkpoint to prevent collision between AI vehicles and stage elements.

Info:

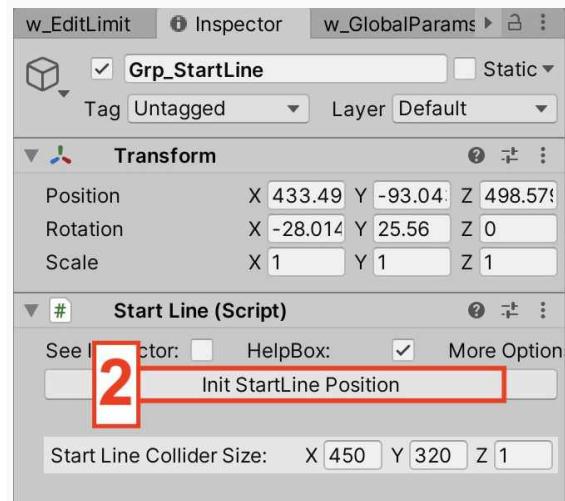
Learn more about Track creation in the Doc Part 2: All features section **Tracks**.

2-Initialize the Start Line

-In the Hierarchy select **Grp_StartLine** (spot 1).
(Hierarchy Tab → GAMEPLAY → PATH → Grp_StartLine)



-In the Inspector press **Init StartLine Position** button (spot 2).

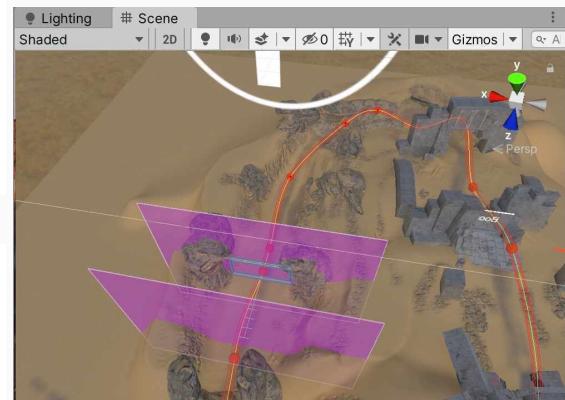


The Start Line is moved and rotate to checkpoint 0 position (cp_00).

The Start Line is represented by 2 purple squares on scene view.

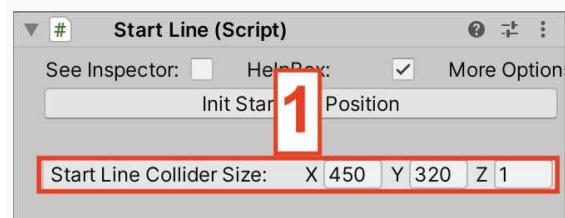
Info:

When the player finishes a lap, the player must pass through the start line.



By default the Start Line Collider Size is set to:
X = 450 Y = 320 Z = 1

It is possible to change the collider size.



Start Line is now set up.

More info in section **Start Line**
→ Page 120

3-Set up the scene to test the track

-Open **w_TrackManager** window (spot 1).
(Tools → TS → w_TrackManager)

A new window appears. I suggest you to attach the window to the editor to easily use the window (spot 1).

-Click on tab **Global Params** (spot 1).

-Set **Current Game Mode** to 3 (spot 2).

Current Mode = 3 is a game mode dedicated to test a race.

-Set **HowManyPlayer** to 1 (spot 3)

*HowManyPlayer = 1 means that there is one Player
(2 means 2 players split screen)*

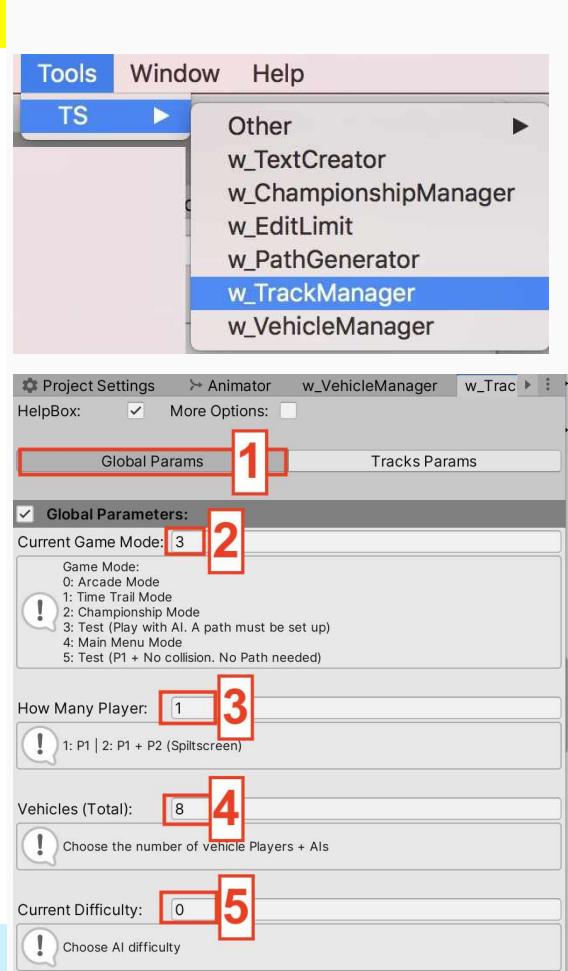
-Set **Vehicles (Total)** to 8 (spot 4).

Vehicles (Total) = 8 means that there are 8 vehicles in the race.

-Set **Current Difficulty** to 0 (spot 5).

Current Difficulty = 0 means that the AIs are in easy mode.

More info in section **Game Mode**
→ Page 103



4-Test the track

-Press **Play** button to start the scene.

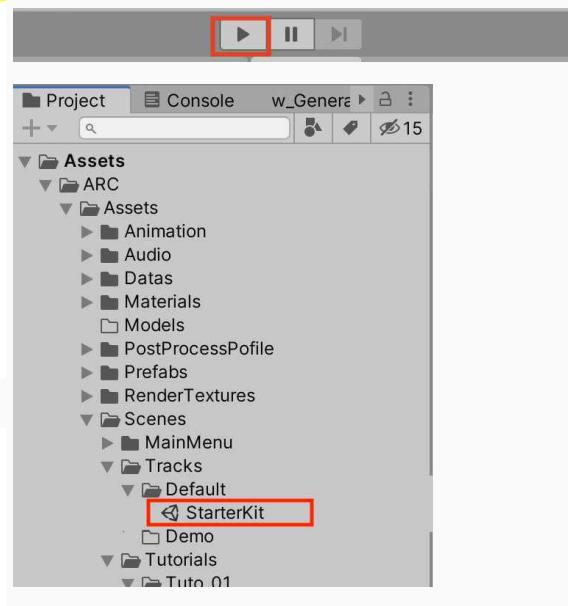
You can play the track. By default there are 2 laps to complete the race.

A scene named **StarterKit** is available to start creating a new track.

(Project tab: Assets → Scenes → Tracks → Default → StarterKit)

Info:

To create a track you can use the starter kit.
Find more info in **Doc Part 2** section **Asset content**
(More info here).

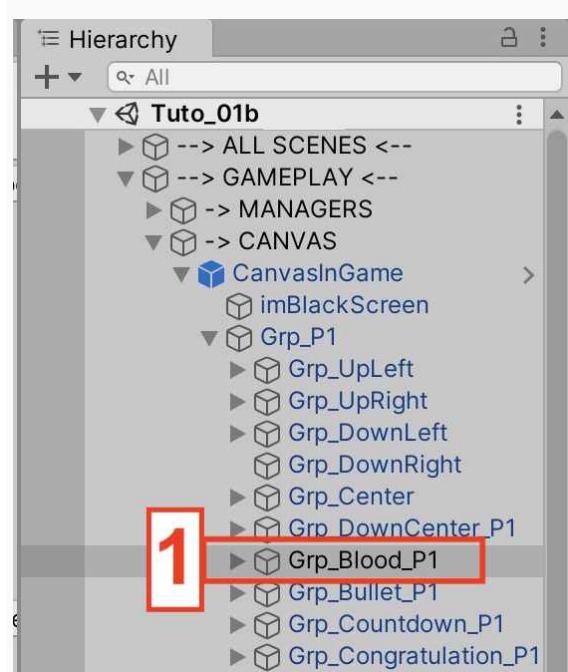


5-Create the mini Minimap

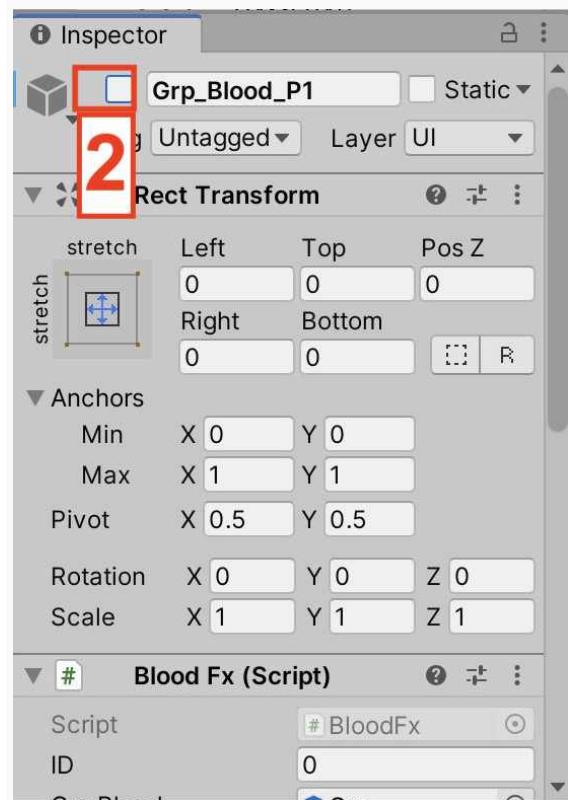
To easily create the minimap we are going to disable the blood effect first.

In the Hierarchy select **Grp_Blood_P1** (spot 1).

(Hierarchy: GAMEPLAY → CANVAS → CanvasInGame → Grp_P1 → Grp_Blood_P1)



-In the Inspector disable the object (spot 2)

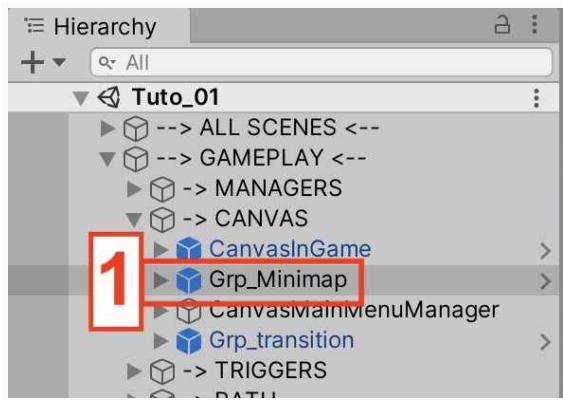


You should have this result.

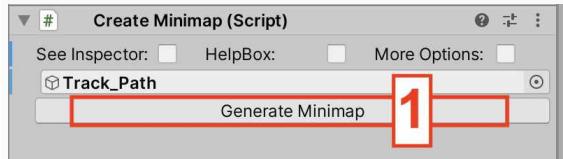


Now we are creating the minimap

In the Hierarchy select **Grp_Minimap** (spot 1).
(Hierarchy: GAMEPLAY → CANVAS → Grp_Minimap)



In the Inspector press **Generate Minimap** button.



The minimap is generated (spot 1). The minimap is visible in the scene view.



The minimap position and scale are not optimized. We are going to change the position and the scale of the minimap

In the Hierarchy select **MinimapCam** (spot 1)
(Hierarchy: GAMEPLAY → CANVAS → Grp_Minimap → MinimapCam)



In the Inspector change the cam position to (spot 1):

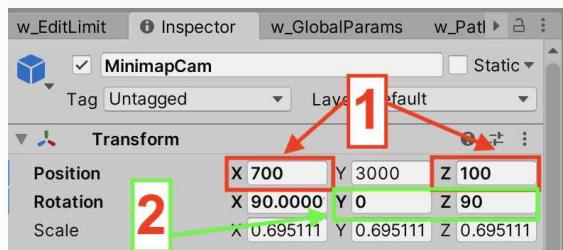
X = 700 Y = 3000 Z = 100

(Note: Do not change Y position)

Change the cam rotation to (spot 2):

X = 90 Y = 0 Z = 90

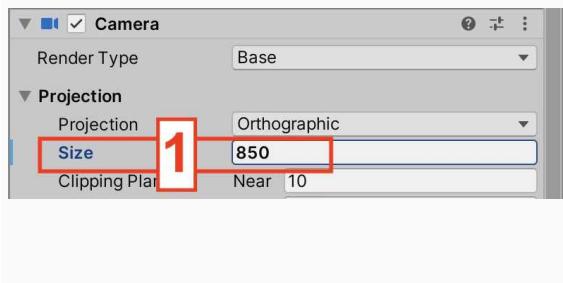
(Note: Do not change X rotation)



You can change the scale of the minimap:

In the Inspector change the Size of camera position:
(spot 1) (Projection -> Size)

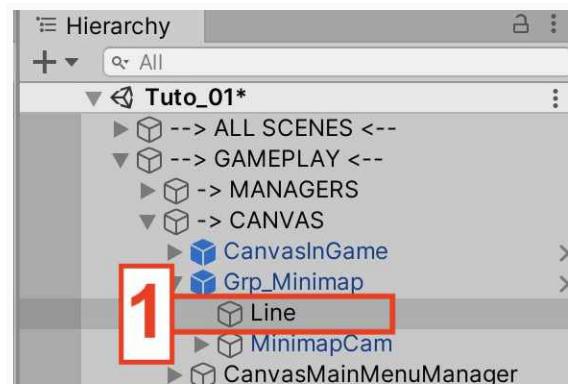
Size = 850



You can change the size of the minimap line

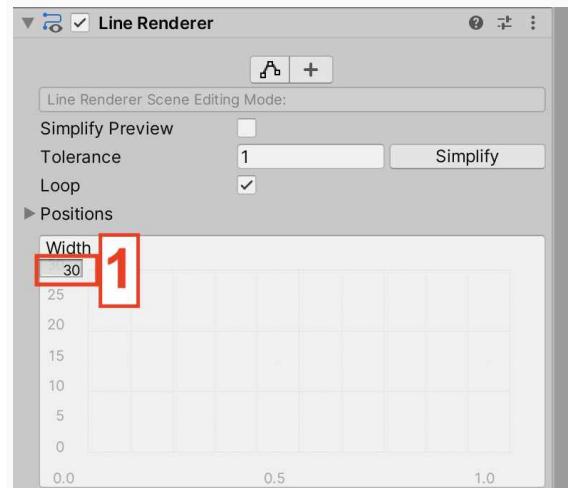
In the Hierarchy select **Line** (spot 1)

(Hierarchy: GAMEPLAY → CANVAS → Grp_Minimap → Line)



In the Inspector change the Width (spot 1):

Width = 30



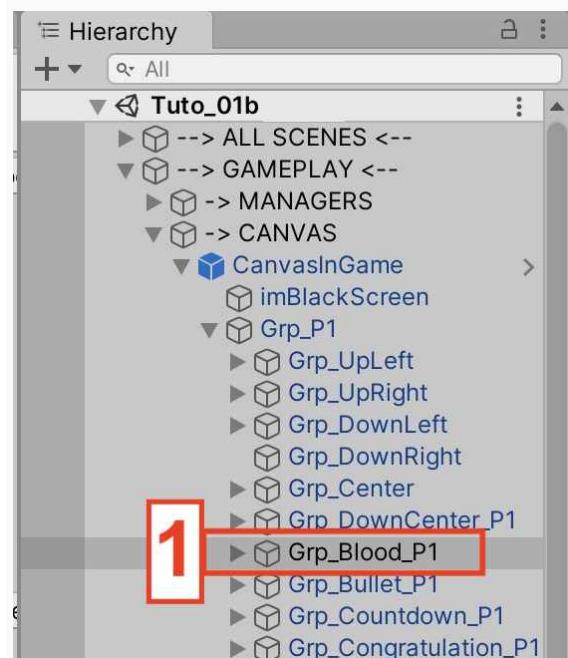
You should have this result (spot 1).



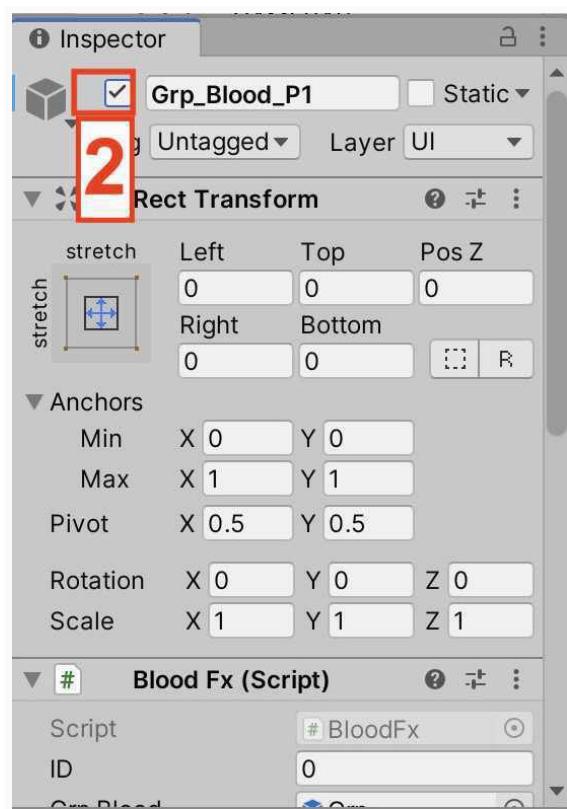
To finish we are going to enable the blood effect.

In the Hierarchy select **Grp_Blood_1** (spot 1).

(Hierarchy: GAMEPLAY → CANVAS → CanvasInGame → Grp_P1 → Grp_Blood_P1)



-In the Inspector enable the object (spot 2)



You should have this result.



More info in section [Minimap](#)
→ Page 131

6-Create the game area limits

The asset has a system to limit the game area for the player.

When the player enters into a LimitZone trigger the vehicle is destroyed.

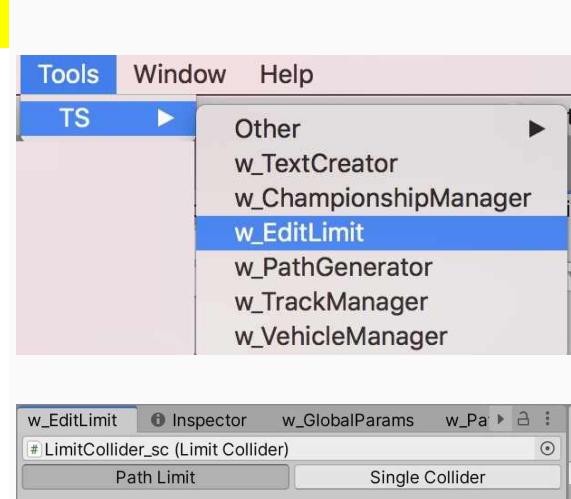
A warning is displayed on screen if the player is too close from the scene limits.

To create the game area limit:

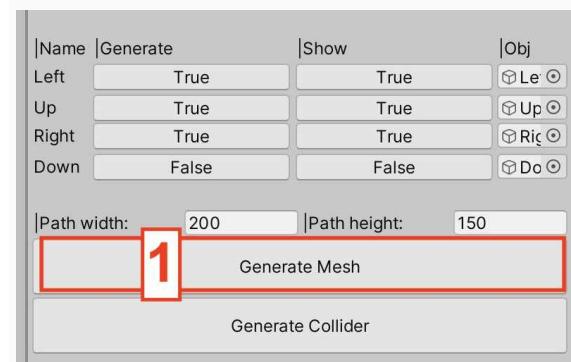
-Open **w_EditLimit** window.

(Tools → TS → w_EditLimit)

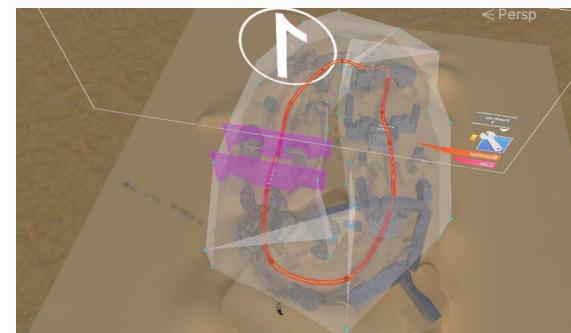
A new window appears. I suggest you to attach the window to the editor to easily use the window.



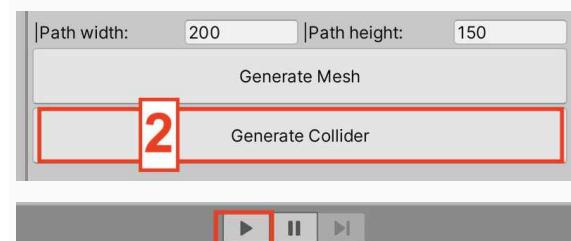
-Press **Generate Mesh** button (spot 1).



A mesh is generated along the path. This mesh is used to create the collider that limit the game area.



-Press **Generate Collider** button (spot 2).



The limit area system is now set up.

Press **Play** button

Then try to go out the limit of the game area to test the system.

More info in section **Game Area**

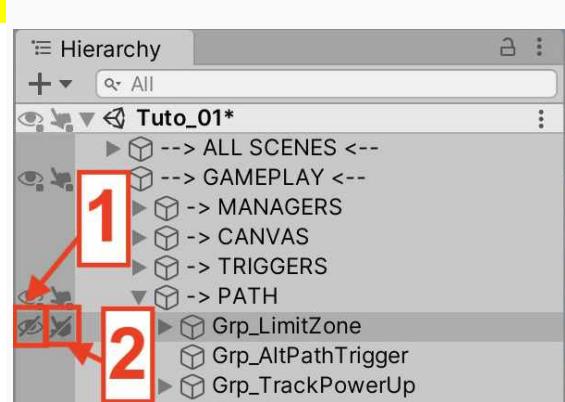
→ Page 128

7-Add Power-ups to your track

To easily create Power-ups we are going to hide and disable the ability to select of the **Grp_LimitZone** object in the scene view.

-In the Hierarchy on the left of **Grp_LimitZone**
(Hierarchy: GAMEPLAY → PATH → Grp_LimitZone)

- Click on the **Eye** icon (spot 1) to hide the **Grp_LimitZone** in the scene view.
- Click on the **Hand** icon (spot 2) to disable the ability to select **Grp_LimitZone** in the scene view.

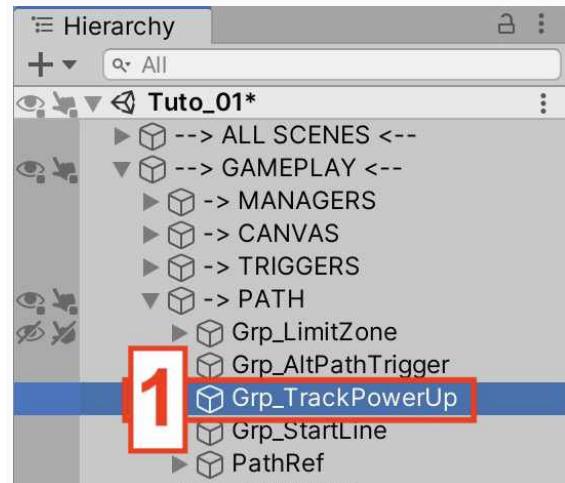


Info:

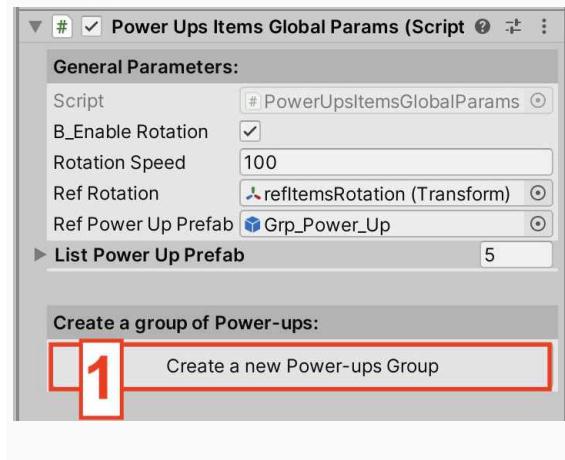
To prevent issues you must not create power-up between the Checkpoint 0 and the checkpoint 1.

Now we are going to create power-ups between checkpoints **cp_01** and **cp_02**

-In the Hierarchy select **Grp_TrackPowerUp** (spot 1)
(Hierarchy: GAMEPLAY → PATH → Grp_TrackPowerUp)

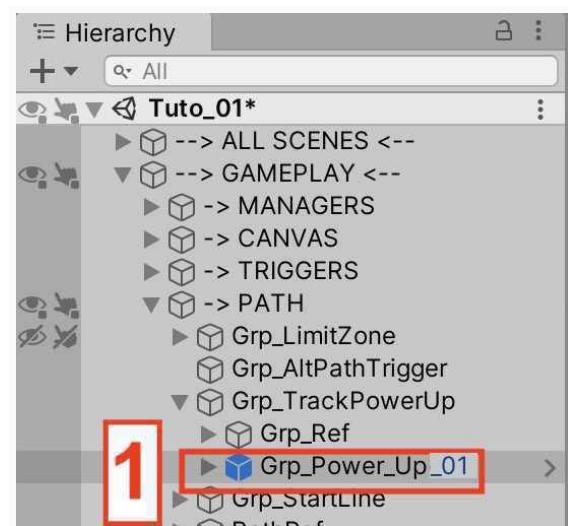


-In the Inspector press
Create a new Power-ups Group button (spot 1).



A new object named `Grp_Power_Up` is created and automatically selected in the Hierarchy.

Rename it `Grp_Power_Up_01` (spot 1).



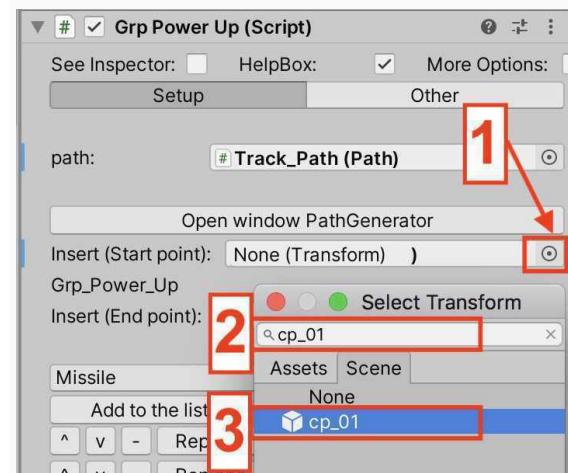
We want powers-ups to be positioned between `cp_01` and `cp_02`. We are going to set up the new created power-ups group.

-In the Inspector press the cercle next to **Insert (Start point)** field (spot 1).

A new window appears.

-In the search field write `cp_01` (spot 2).

-In the list select `cp_01` (spot 3)

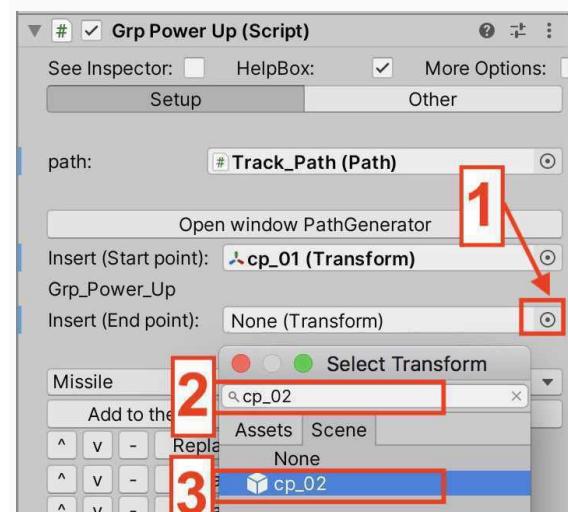


-In the Inspector press the cercle next to **Insert (End point)** field (spot 1).

A new window appears.

-In the search field write `cp_02` (spot 2).

-In the list select `cp_02` (spot 3)

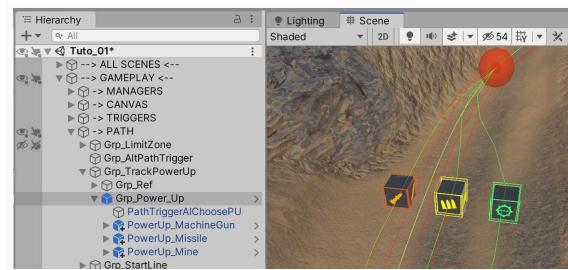


-In the Inspector press **Create Power-ups list** (spot 1).

-A pop-up appears. Press **Yes**.



3 power-ups have been created in the scene.



-In the Hierarchy select **PowerUp_MachineGun** (spot 1)

(Hierarchy: → GAMEPLAY → PATH → Grp_TrackPowerUp → Grp_PowerUp_01 → PowerUP_MMchineGun)

If you move the object in the scene view:
you see a green line. This green line shows the path followed by AI to grab the power-up. The green line must not go through any element of the stage.

-In the Inspector change the transform to:

X = -6 Y = 12 Z = 118

-In the Hierarchy select **PowerUp_Missile** (spot 2)

(Hierarchy: → GAMEPLAY → PATH → Grp_TrackPowerUp → Grp_PowerUp_01 → PowerUP_Missile)

-In the Inspector change the transform to:

X = -21 Y = 2 Z = 118

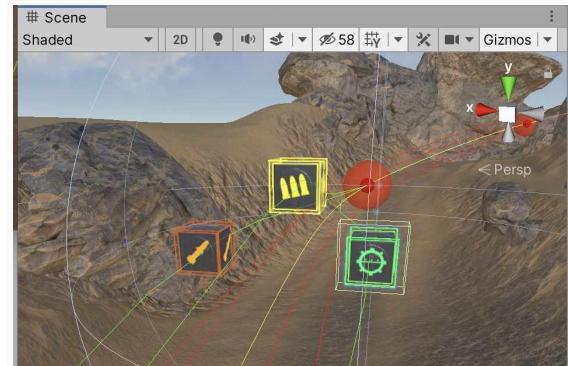
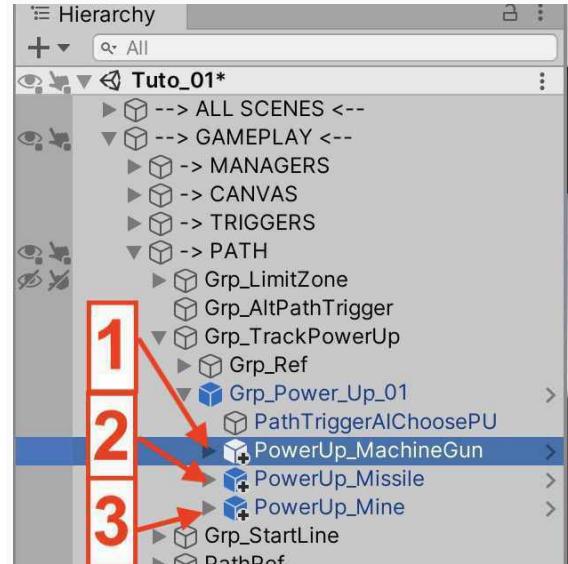
-In the Hierarchy select **PowerUp_Mine** (spot 3)

(Hierarchy: → GAMEPLAY → PATH → Grp_TrackPowerUp → Grp_PowerUp_01 → PowerUP_Mine)

-In the Inspector change the transform to:

X = 5 Y = 2 Z = 118

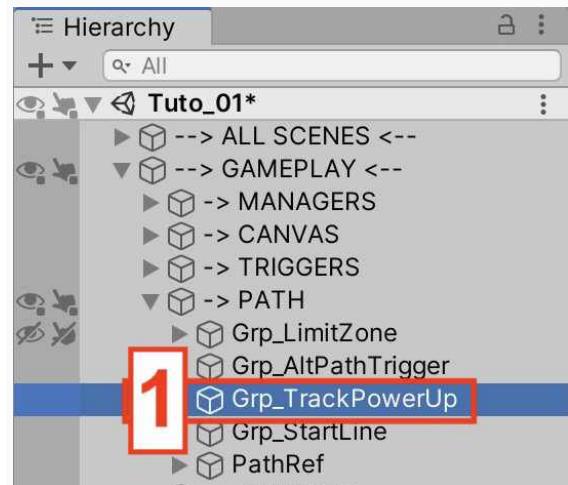
You should have the same result as the picture.



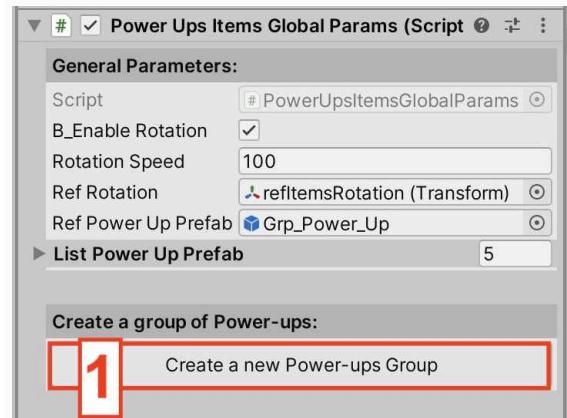
Now we are going to use the same process to create a second power-ups groups between checkpoints **cp_06** and **cp_07**

-In the Hierarchy select **Grp_TrackPowerUp** (spot 1)

(Hierarchy: GAMEPLAY → PATH → Grp_TrackPowerUp)



- In the Inspector press Create a new Power-ups Group button (spot 1).



A new object named Grp_Power_Up is created and automatically selected in the Hierarchy.

- Rename it Grp_Power_Up_02 (spot 1).



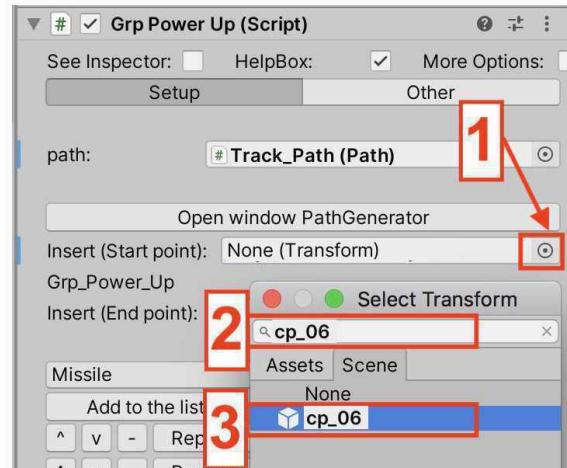
We want powers-ups to be positioned between cp_06 and cp_07. We are going to set up the new created power-ups group.

- In the Inspector press the cercle next to Insert (Start point) field (spot 1).

A new window appears.

- In the search field write cp_06 (spot 2).

- In the list select cp_06 (spot 3)

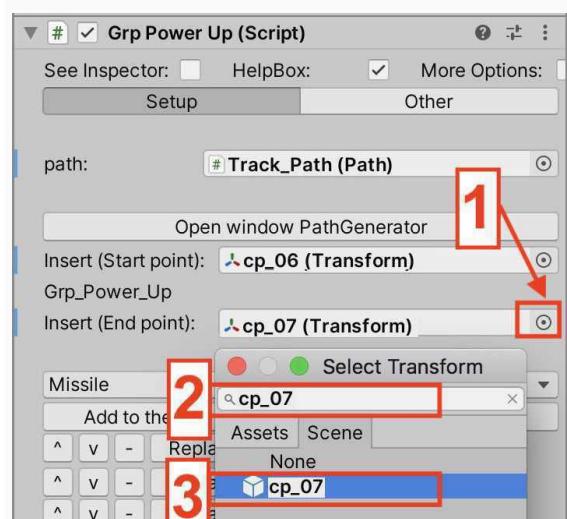


- In the Inspector press the cercle next to Insert (End point) field (spot 1).

A new window appears.

- In the search field write cp_07 (spot 2).

- In the list select cp_07 (spot 3)



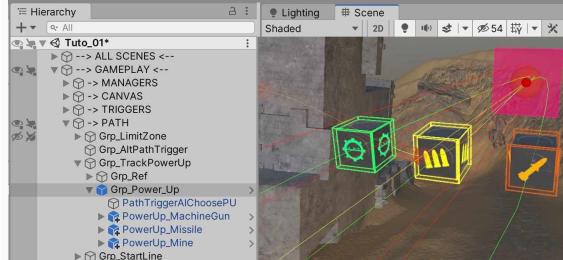
-Press **Create Power-ups list** (spot 1).



-A pop-up appears. Press **Yes**.



3 power-ups have been created in the scene.

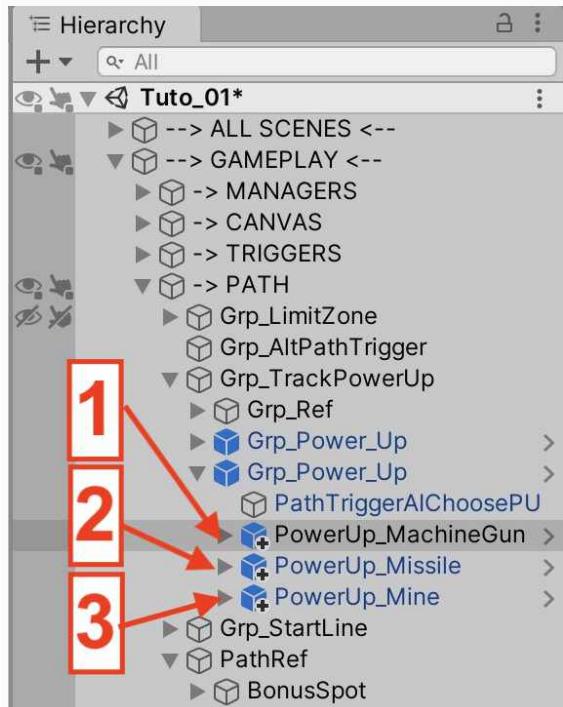


-In the Hierarchy select **PowerUp_MachineGun** (spot 1)

(Hierarchy: → GAMEPLAY → PATH → Grp_TrackPowerUp → Grp_PowerUp_02 → PowerUP_MachineGun)

-In the Inspector change the transform to:

X = -2 Y = 17 Z = 150

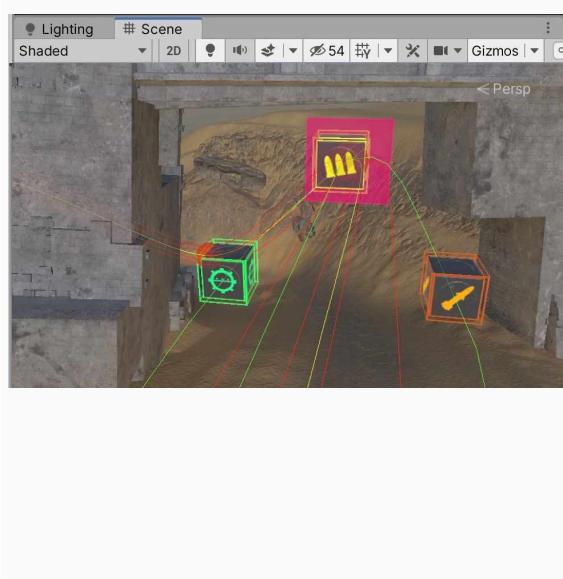


-In the Hierarchy select **PowerUp_Missile** (spot 2)

(Hierarchy: → GAMEPLAY → PATH → Grp_TrackPowerUp → Grp_PowerUp_02 → PowerUP_Missile)

-In the Inspector change the transform to:

X = -18 Y = 0 Z = 150



-In the Hierarchy select **PowerUp_Mine** (spot 3)

(Hierarchy: → GAMEPLAY → PATH → Grp_TrackPowerUp → Grp_PowerUp_02 → PowerUP_Mine)

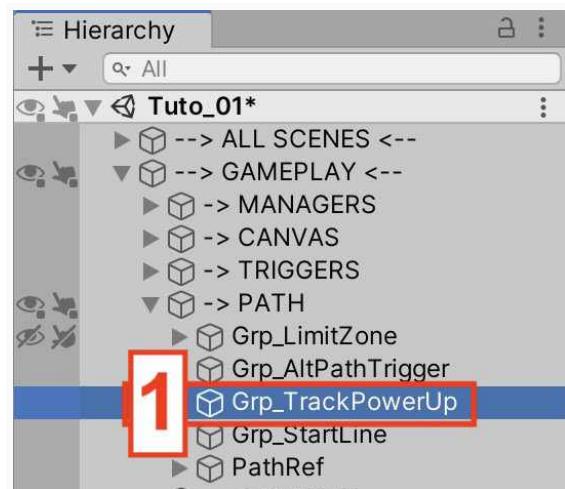
-In the Inspector change the transform to:

X = 14 Y = 0 Z = 150

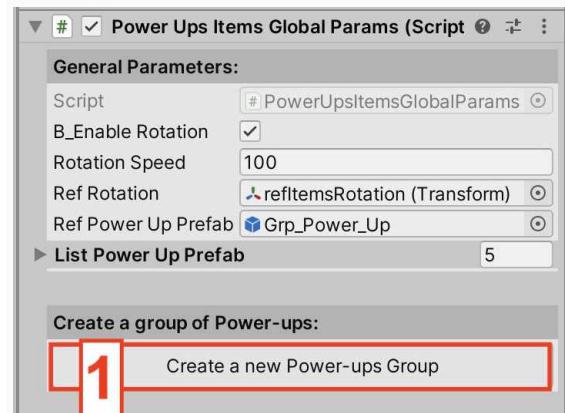
You should have the same result as the picture.

To finish this tutorial part, we are going to create a third power-up group between checkpoint **cp_09** and **cp_10**

-In the Hierarchy select **Grp_TrackPowerUp** (spot 1)
(Hierarchy: GAMEPLAY → PATH → Grp_TrackPowerUp)

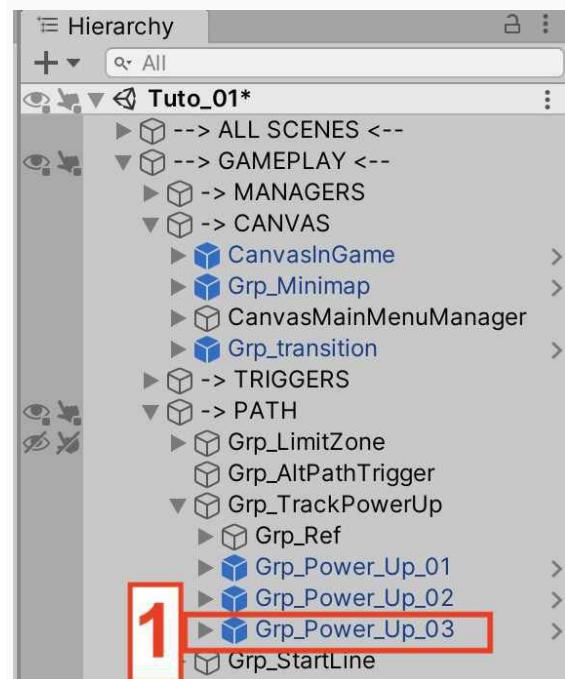


-In the Inspector press
Create a new Power-ups Group button (spot 1).



A new object named **Grp_Power_Up** is created and automatically selected in the Hierarchy.

Rename it **Grp_Power_Up_03** (spot 1).



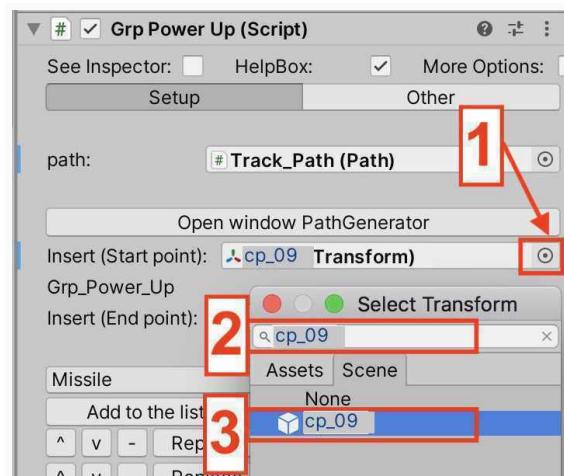
We want powers-ups to be positioned between cp_09 and cp_10. We are going to set up the new created power-ups group.

- In the Inspector press the cercle next to Insert (Start point) field (spot 1).

A new window appears.

- In the search field write cp_09 (spot 2).

- In the list select cp_09 (spot 3)

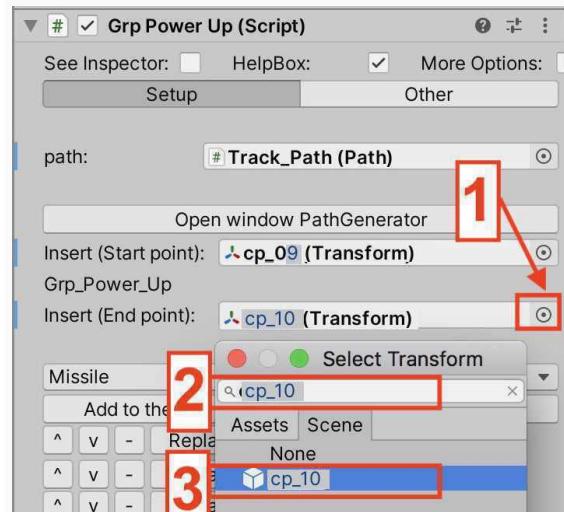


- In the Inspector press the cercle next to Insert (End point) field (spot 1).

A new window appears.

- In the search field write cp_10 (spot 2).

- In the list select cp_10 (spot 3)

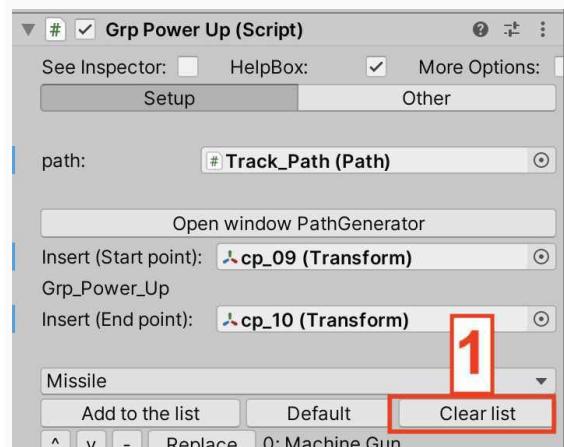


It is possible to choose:

- Which power-ups are created.
- The number of power-Ups created.

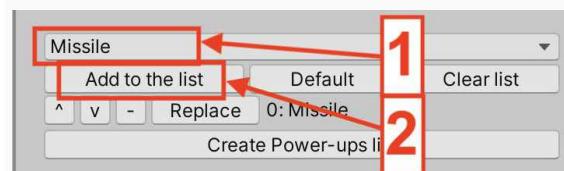
We are going to create a custom list of power-ups.

- Press Clear List (spot 1).



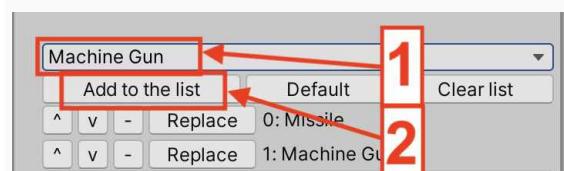
- Select Missile (spot 1).

- Then press Add to the list button (spot 2).

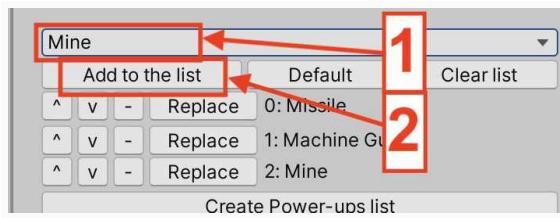


- Select Machine Gun (spot 1).

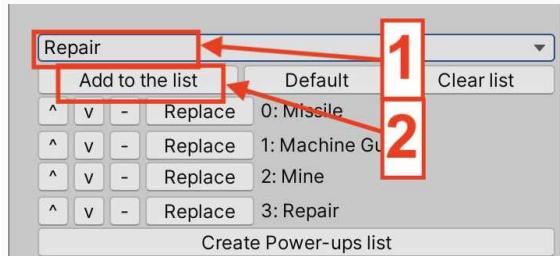
- Then press Add to the list button (spot 2).



- Select **Mine** (spot 1).
- Then press **Add to the list** button (spot 2).



- Select **Repair** (spot 1).
- Then press **Add to the list** button (spot 2).



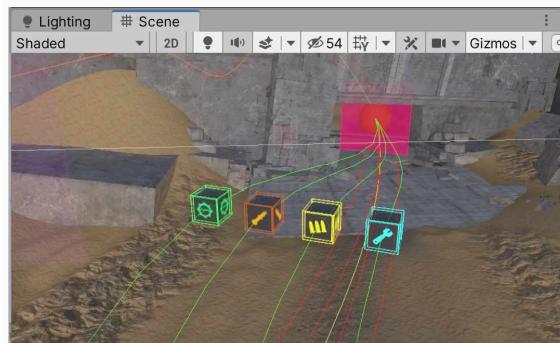
-Press **Create Power-ups list** button.



-A pop-up appears. Press **Yes**.



4 power-ups have been created in the scene.



Power-ups are setup

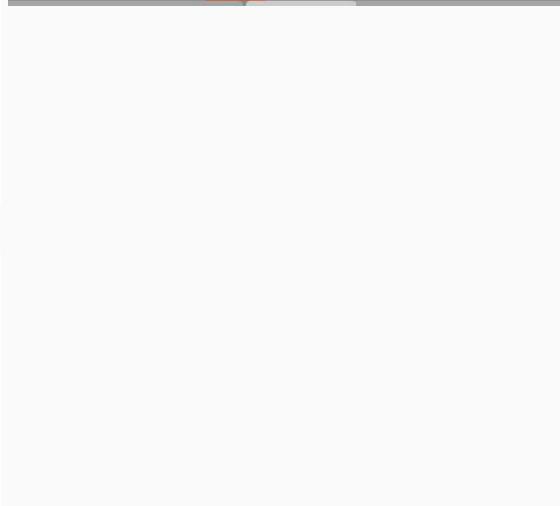


Press **Play** button to test the power-ups.

Conclusion:

Now the scene is set up to be used in the game.

-Press **Ctrl+S** to save the scene.



Reminders

Very Important:

Power-ups groups **MUST** be created between:

-2 checkpoints from the main path.

Or

-2 checkpoints from an alternative path.

(more info about Alternative Path in the part 10-Create alternative Path)

Example:

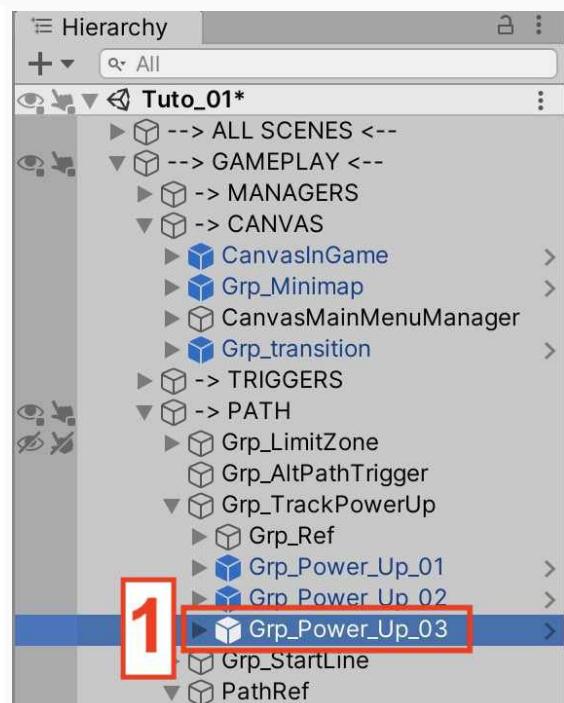
You'll have issues if you create a Power-ups group that starts from the main path and ends on an alternative path.

You'll have issues if you create a Power-ups group that starts from the alternative path and ends on main path.

Note:

If you want to remove a Power-ups group.

-In the Hierarchy, select and delete the power-ups group you do not want to use anymore (spot 1).



More info in section **Power-ups**

→ Page 134

8-Add AI booster

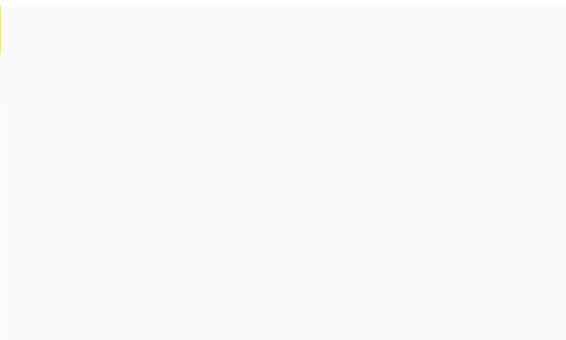
Note:

-AI never uses its booster automatically. AI uses its booster if it enters a special trigger named **Trigger_AIBooster**.

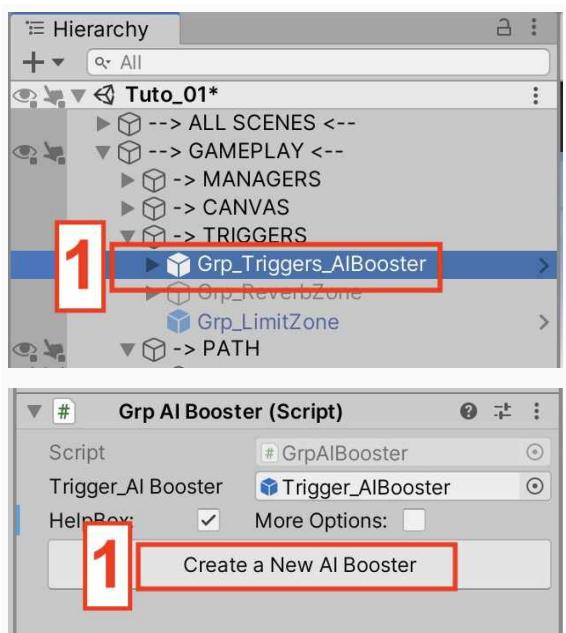
-Adding **AI_Booster** to a track is necessary to have competitive AIs.

-In the Hierarchy select **Grp_Triggers_AIBooster** (spot 1).

(Hierarchy: GAMEPLAY → TRIGGERS → **Grp_Triggers_AIBooster**)

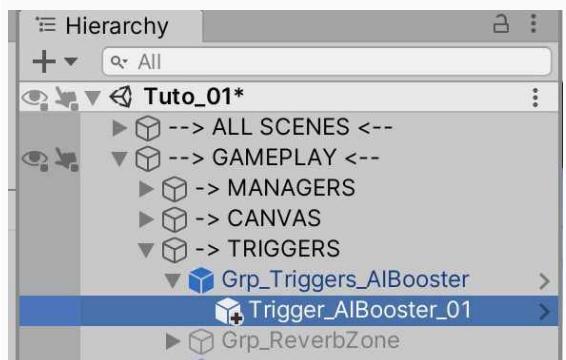


-In the Inspector press **Create a New AI Booster** (spot 1).



A new object named **Trigger_AIBooster** is created and automatically selected in the Hierarchy.

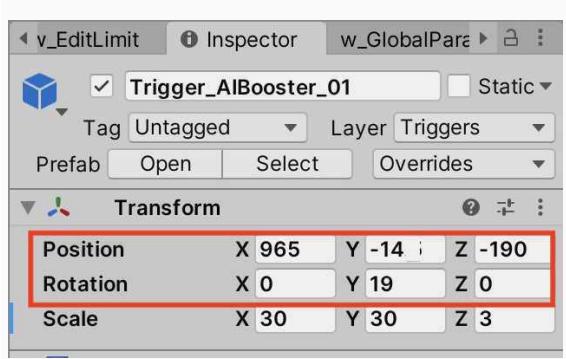
-Rename it **Trigger_AIBooster_01**.



-Change **Trigger_AIBooster** transform to:

Position: X= 965 Y= -14 Z= -190

Rotation: X=0 Y= 19 Z=0



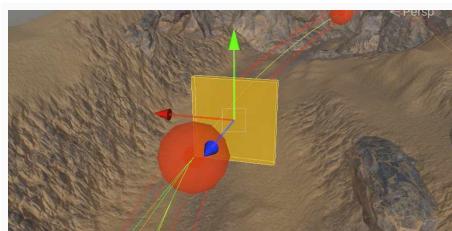
Trigger_AIBooster is represented by a yellow square.

If you don't see the trigger press gizmos buttons in the scene view (spot 1).



On the picture we can see that the **track path** goes through the **Trigger_AIBooster**.

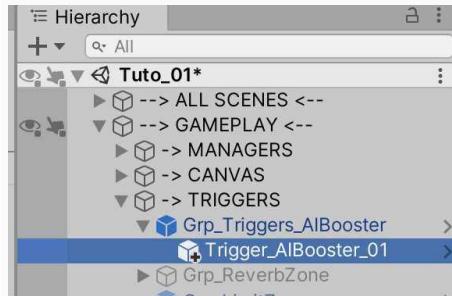
When a vehicle enters the trigger its booster will be enabled.



Now we are going to create a second **Trigger_AIBooster**.

-In the Inspector select **Trigger_AIBooster_01**.

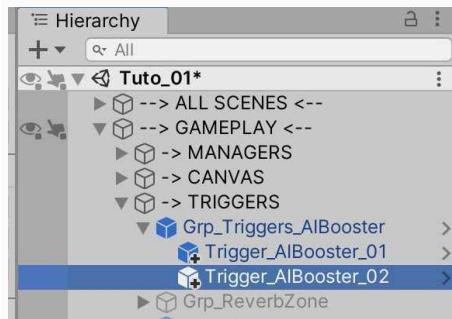
(Hierarchy: GAMEPLAY → TRIGGERS → Grp_Triggers_AIBooster → Trigger_AIBooster_01)



-Duplicate the object by pressing **CTRL + D**

A new object named **Trigger_AIBooster (1)** is created and automatically selected in the Hierarchy.

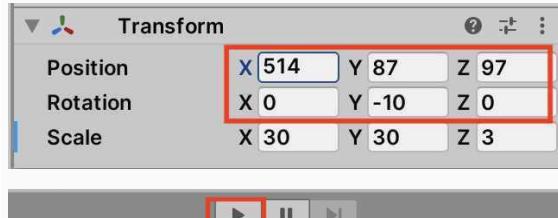
-Rename it **Trigger_AIBooster_02**.



-In the Inspector change **Trigger_AIBooster_02** transform to:

Position: X = 514 Y = 87 Z = 97

Rotation: X = 0 Y = -10 Z = 0



Press **Play** button to test AI booster.

Reminders

1-**AI_Booster** are very important to have competitive AIs and good difficulty balance.

2-You can add as many **AI_Booster** as you want.

Consider that it is good to put a booster every **2 or 3** checkpoints.

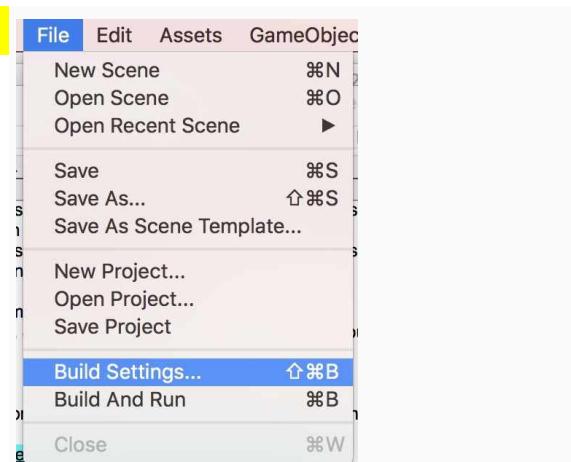
3- **IMPORTANT:** To prevent issues do not put trigger at the same place as an other trigger.

More info in section **AI Booster**
→ Page 123

9-Set up the track in the Track Manager

This section explains how to set up a Track in the Track Manager. The Track Manager allows to set up the track used during the game.

-Go to File → Build Settings

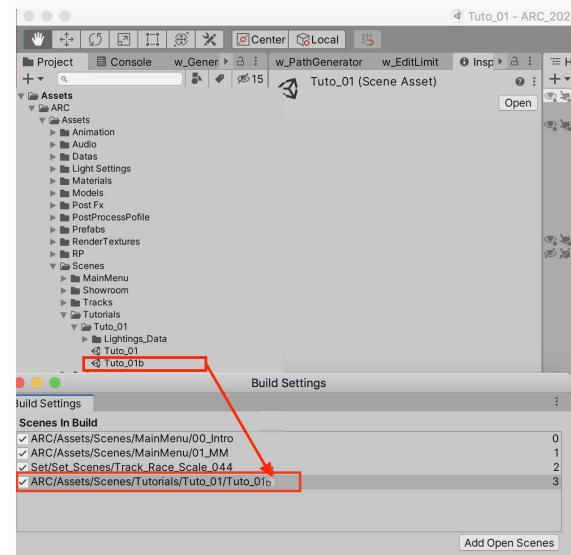


For the example we are adding the scene we have created during this tutorial.

-Drag and drop **Tuto_01b** scene at the end of the list in the **Scenes In Build Box**.

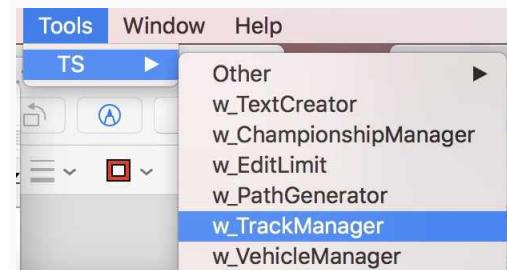
(Project tab: Assets → Scenes → Tutorials → Tuto_01 → Tuto_01b)

Unity is now able to load the scene during the game.



-Go to Tools → TS → w_trackManager

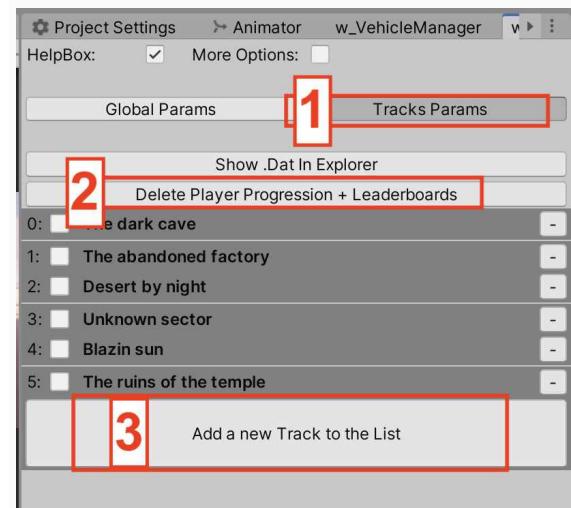
A new window appears. I suggest you to attached the window to the editor.



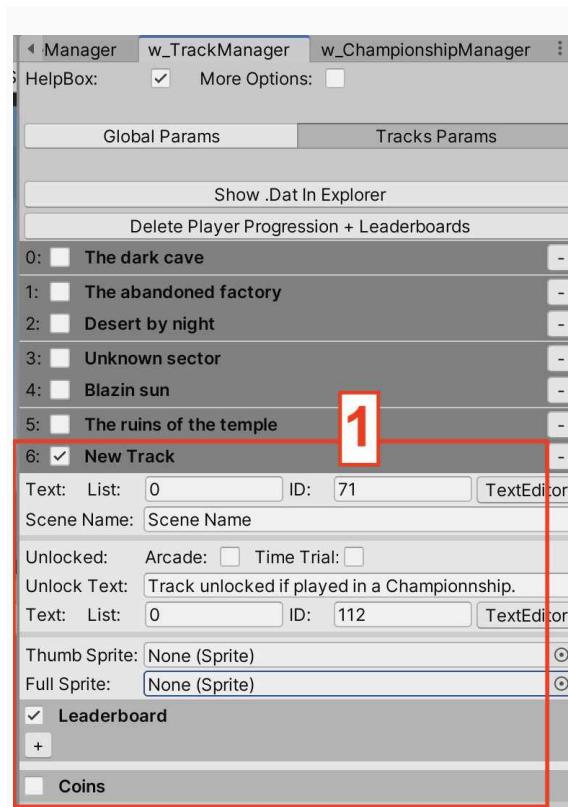
-In **w_TrackManager** click on tab **Tracks Params** (spot 1).

-Press **Delete Player Progression + Leaderboards** button (spot 2).

-Go to the end of **w_TrackManager** tab and press **Add a new Track to the List** button (spot 3).



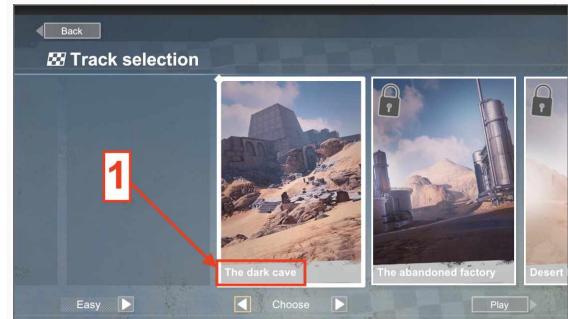
A new slot is created (spot 1).



First we are going to set up the name of the track shown by the player in the main menu.

The name of the track is displayed when the player choose a track in Arcade or Time Trial Mode (spot 1).

The asset allows to display text using a multi-language system.



For this example we are going to use the multi-language text with the ID = 157 already set up in the project. ID: 157 displays the text Track Demo

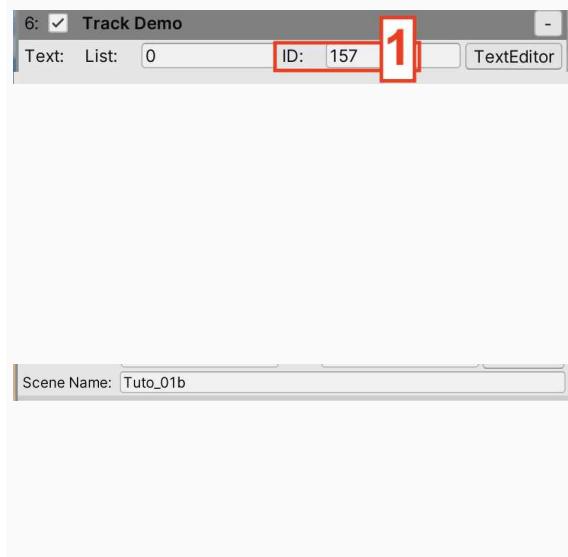
(You can learn more about text and multi-language in Doc Part 2 section Localization and Text at runtime).

-In field **ID** write **157** (spot 1)

To load a scene during the game, the system needs to know the name of the scene saved in Project Folder.

-In field **Scene Name** write the name of the scene use for this track.

For this example write **Tuto_01b**.



It is possible to choose if the track is unlocked in Arcade and Time Trial when the game starts the first time.

For this example we want the track available in both mode:

-Check **Arcade** and **Time Trial** checkbox.

You can change the text displayed in the UI when the track is locked. In the example we do not change this text.

When the player select a track in Arcade or Time mode a thumbnail image is used to represent the track.

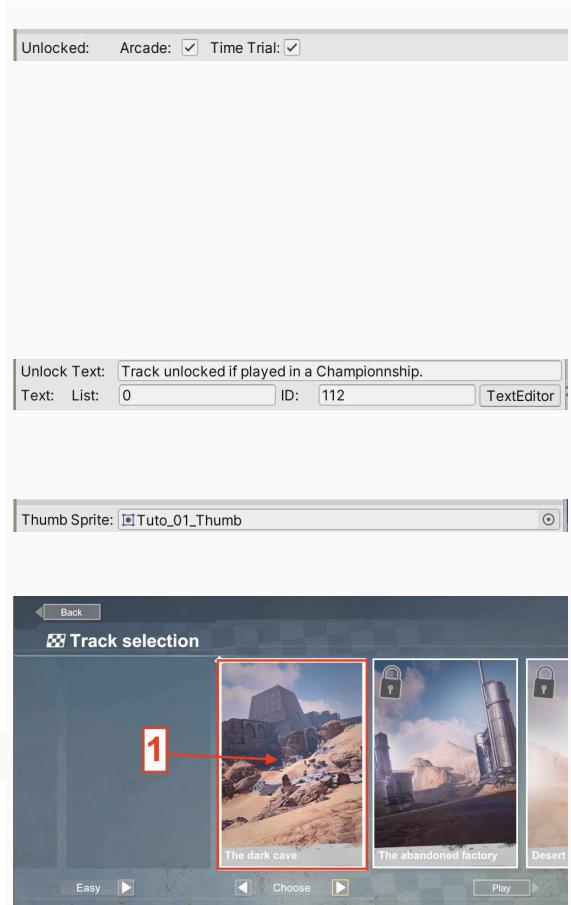
-Click on the **circle** in field **Thumb Sprite**.

For this example choose **Tuto_01_Thumb**.

This picture is displayed when the player choose a race.

Note:

The thumbnails used by default in the asset have a size of 499 x 721

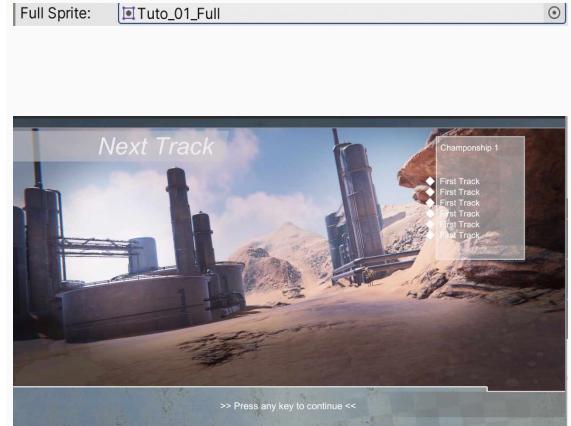


When the player starts a race in Championship a full screen image is used to represent the track.

-Click on the **circle** in field **Full Sprite**.

For this example choose **Tuto_01_Full**.

This picture is displayed when the player starts the race in Championship Mode.



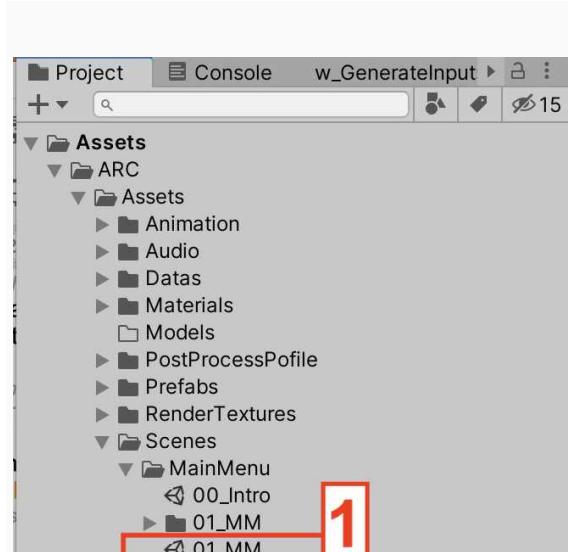
Info:

The full images used by default in the asset have a size of 1920 x 945

-Save the scene (Ctrl+S)

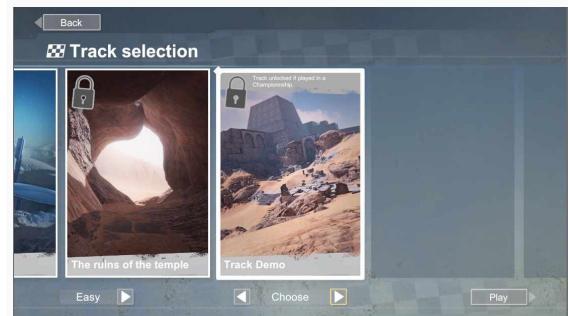
-Then open the Main Menu Scene 01_MM (spot 1).
(Project tab: Assets → Scenes → Main Menu → 01_MM).

(If the pop-up save the scene appears press Yes).



Press Play button.

It is possible to choose the [Track_Demo](#) in Arcade and Time Trial Mode



More info to set up tracks and championship in
[Game Mode](#) section → Page 103

10-Create alternative Path

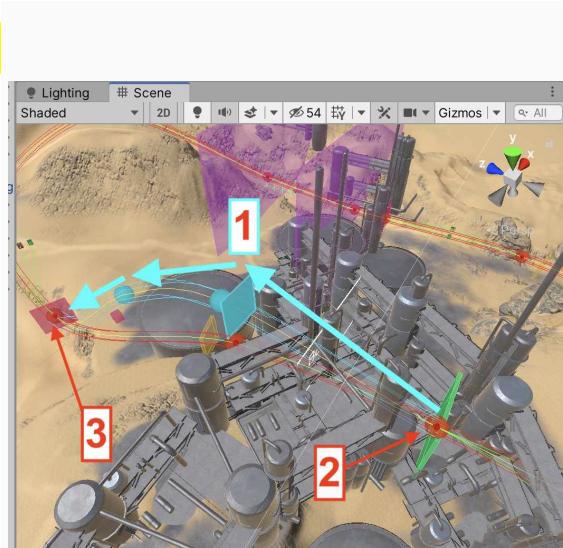
An AI alternative path (spot 1) is a path that:

- Starts on the main track path (spot 2)
- Ends on the main track path (spot 3)

It is useful for making shortcuts or path variations.

In the asset we use the name **Alt Path** to designate an alternative path.

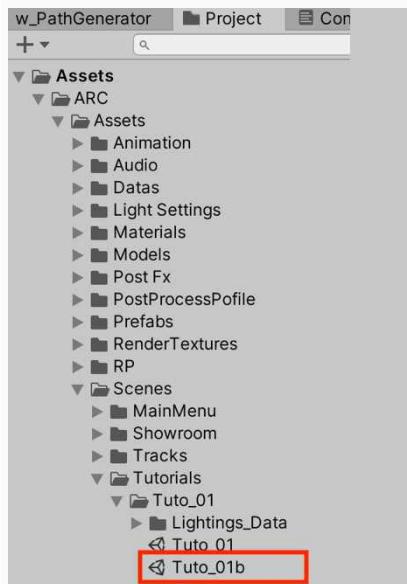
By default Alt Path is represented by blue gizmos.



For this tutorial part we are going to use the scene we have created in the previous part of this tutorial.

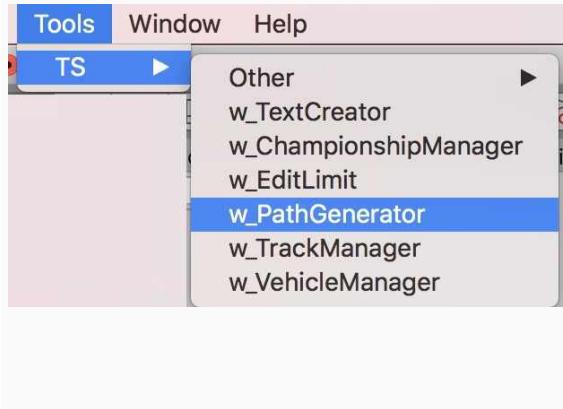
-Open **Tuto_01b** scene (spot 1).

(Project tab: Assets → Scenes → Tutorials → Tuto_01 → Tuto_01b)



As an example we are going to create an Alt Path between checkpoint **cp_07** and **cp_09**.

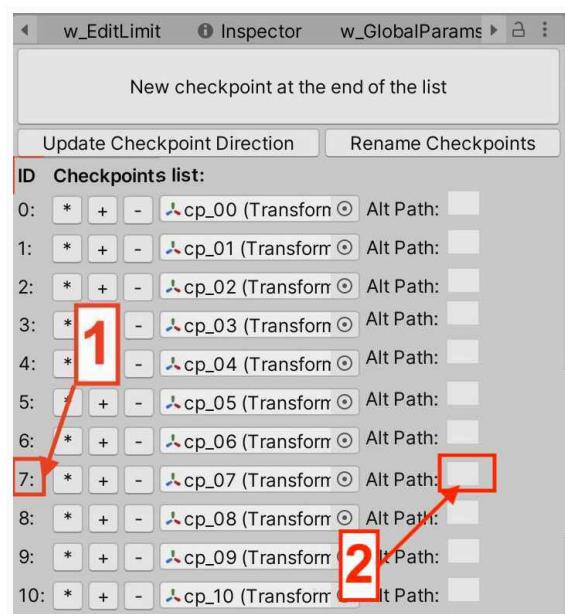
-Go to **Tools → TS → w_PathGenerator**



A new window appears. I suggest you to attached the window to the editor.

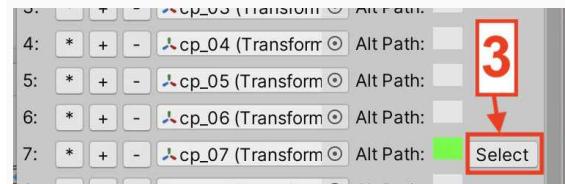
-In **w_PathGenerator** window go to **ID 7** (spot 1)
(checkpoint *cp_07*)

-Press **gray square** next to Alt Path Trigger (spot 2)



The square become green.

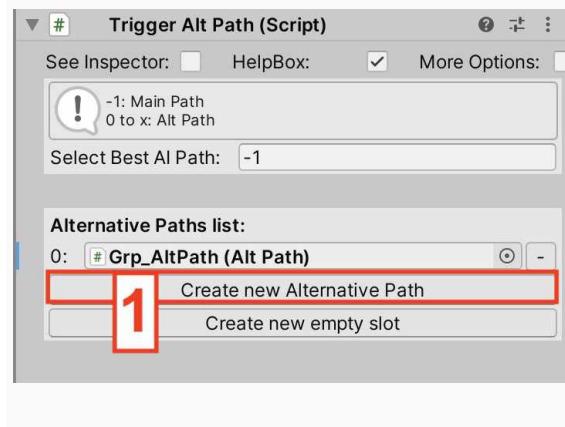
-Press **Select button** (spot 3)



Object **AltPathTrigger** inside *cp_07* object is automatically selected.

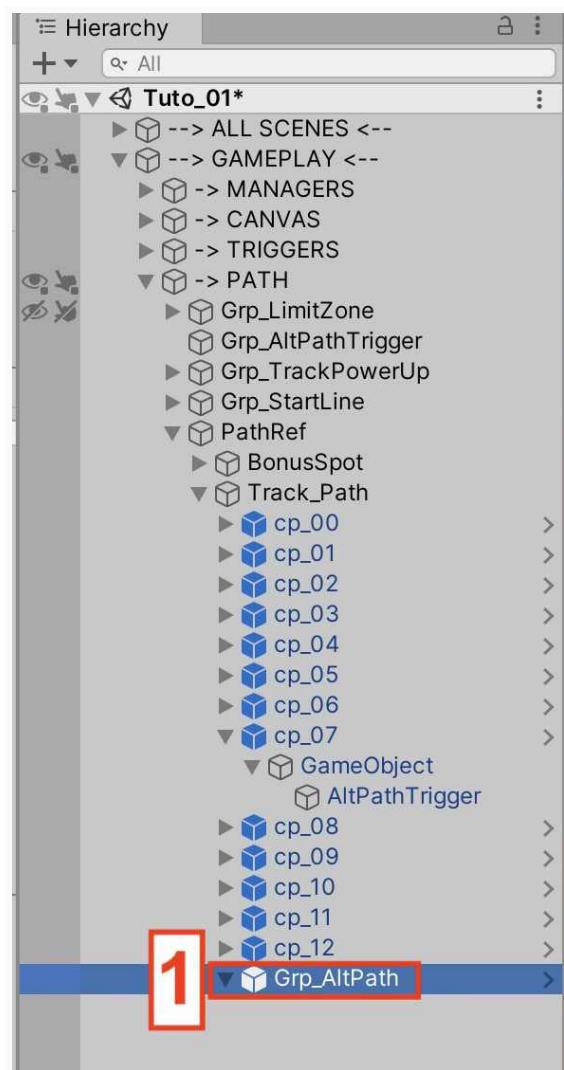
-In the Inspector press **Create new alternate Path** button (spot 1).

A new object named *Grp_AltPath* is created in the Hierarchy.



-In the Hierarchy select **Grp_AltPath** (spot 1).

(Hierarchy: GAMEPLAY → PATH → PathRef → Track_Path → Grp_AltPath)

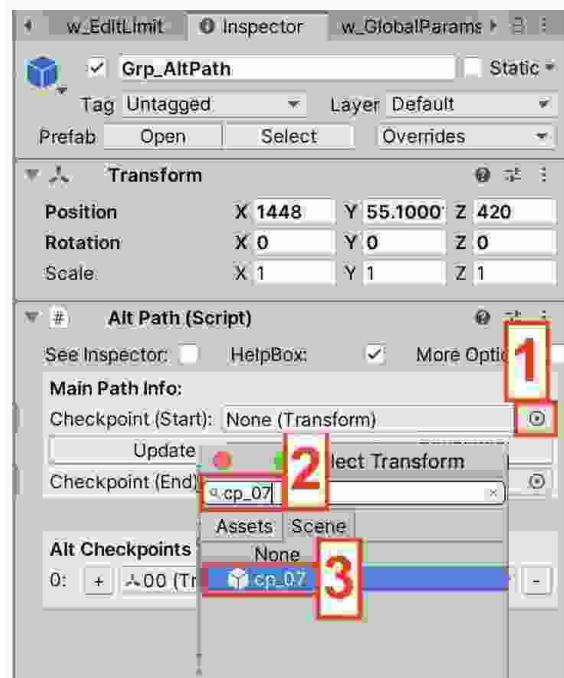


Our Alt Path should start on checkpoint **cp_07** and ends on checkpoint **cp_09**.

-Click on the **circle** next to **Checkpoint (Start)** field (spot 1).

-In the search field write **cp_07** (spot 2).

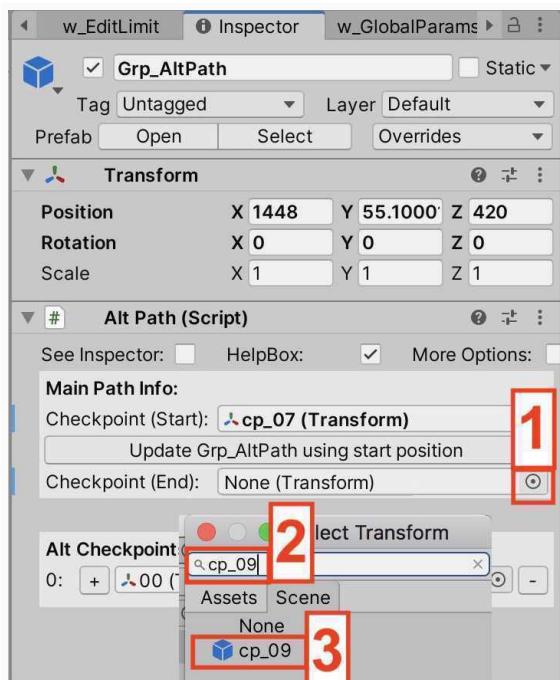
-Select **cp_07** (spot 3).



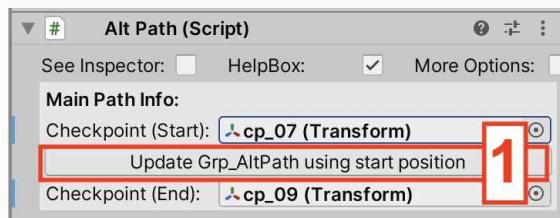
-Click on the **circle** next to **Checkpoint (End)** field (spot 1).

-In the search field write **cp_09** (spot 2).

-Select **cp_09** (spot 3)



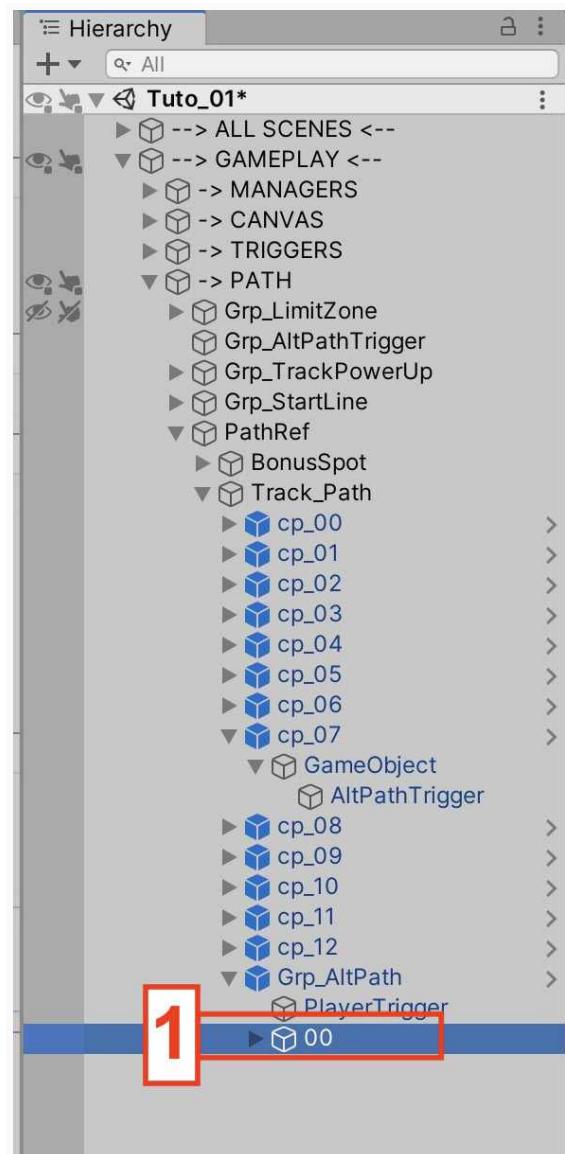
-Press **Update Grp_AltPath using start position**.
(*Grp_AltPath is moved to checkpoint cp_07 position*).



-In the Hierarchy select object named **00** inside

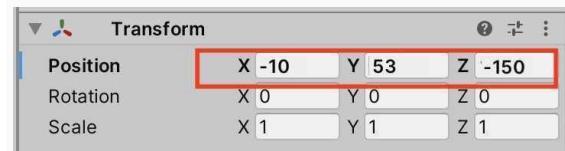
Grp_AltPath (spot 1).

(Hierarchy: GAMEPLAY → PATH → PathRef → Track_Path → Grp_AltPath → 00)

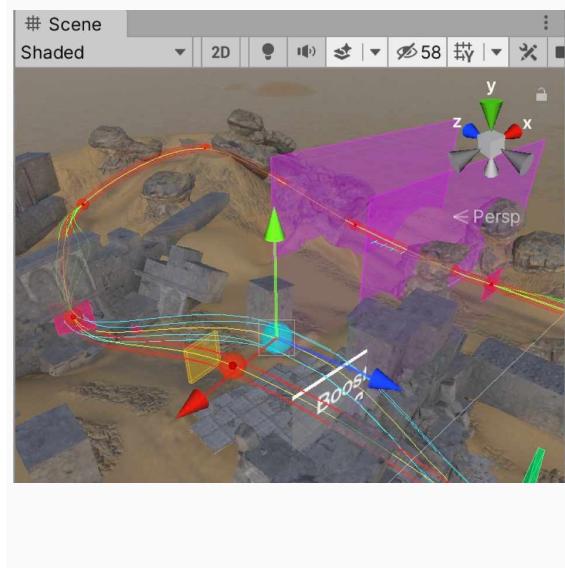


-In the Inspector change transform to:

Position X = -10 Y = 53 Z = -150



Alt Path is represented in blue.



We are going to create a second point for the Alt Path.

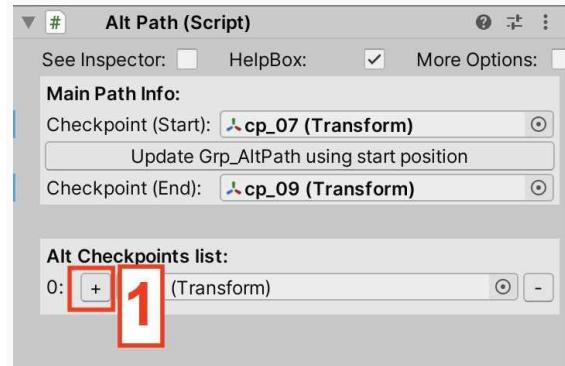
-In the Hierarchy select **Grp_AltPath** (spot 1).

(Hierarchy: GAMEPLAY → PATH → PathRef → Track_Path → Grp_AltPath)



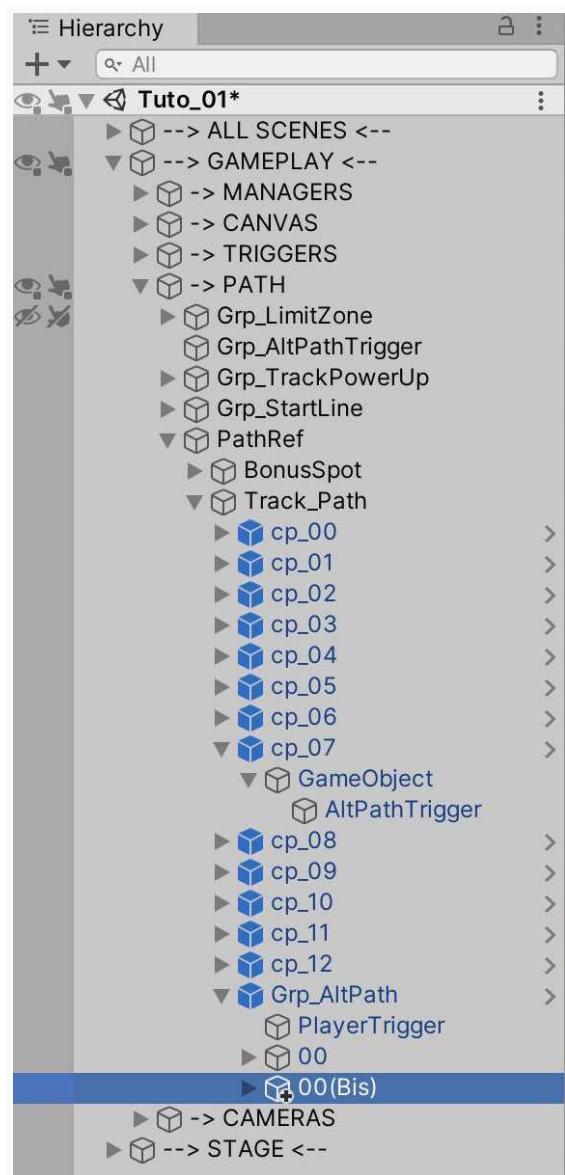
-In the Inspector press + button (spot 1).

A new point is created inside *Grp_AltPath* object named 0(Bis)



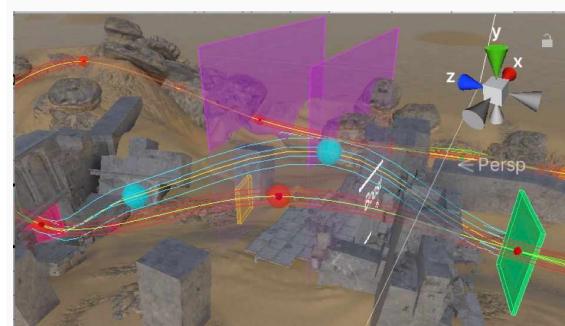
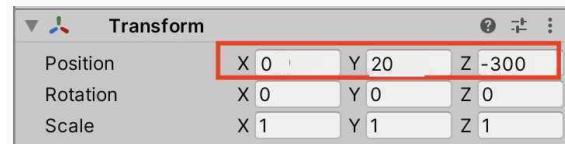
-In the Hierarchy select object named **00(Bis)** inside **Grp_AltPath** (spot 1).

(Hierarchy: GAMEPLAY → PATH → PathRef → Track_Path → Grp_AltPath → 00(Bis))



-In the Inspector change transform to:

Position **X = 0 Y = 20 Z = -300**



Note

You can create as many points as you want.

IMPORTANT:

The setup is finished for AI vehicles.

Now we need to set up the alternative path for Player 1 and Player 2.

During a race, when Player 1 or Player 2 exploded, the system must know if the vehicle is currently following an alternative path or the main path.

If the vehicle is following an alternative path, the vehicle respawned on this alternative path.

If the vehicle is following the main path, the vehicle respawned on the main path.

So, now, we are going to set up the trigger used to indicate that a player is using an alternative path.

Info:

Inside each Alt Path there is an object named **PlayerTrigger** (spot1).

When the player 1 or 2 pass through this object the system know that Player 1 or 2 are using this Alt Path.

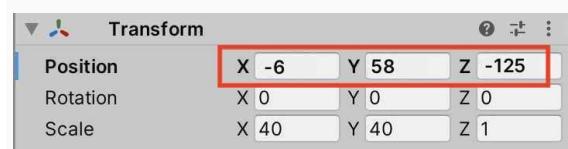
-In the Hierarchy select **PlayerTrigger** object inside **Grp_AltPath** (spot 1).

(Hierarchy: GAMEPLAY → PATH → PathRef → Track_Path → Grp_AltPath → PlayerTrigger)

-In the Inspector change transform to:

Position X = -6 Y = 58 Z = -125

Now when Player 1 or Player 2 enter PlayerTrigger the system knows that the player is following this Alt Path



Reminder

IMPORTANT:

Do not put 2 triggers in the same position

A trigger **must not** touch another trigger.
It will cause issues. Some triggers can be ignored.

Do not create Power-ups group connected to the last point of the Alternative Path.

More info in section **Alt Path**

→ Page 118

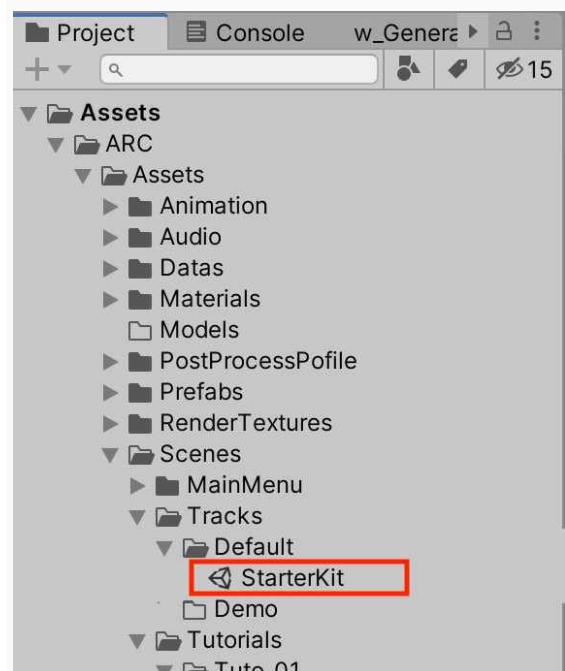
Conclusion:

First tutorial is finished.

Now you are able to create a track.

A scene named **StarterKit** is available to start creating a new track.

(Project tab: Assets → Scenes → Tracks → Default → StarterKit)



The second tutorial explains how to customize your vehicles.

Tuto 2: Customize vehicles

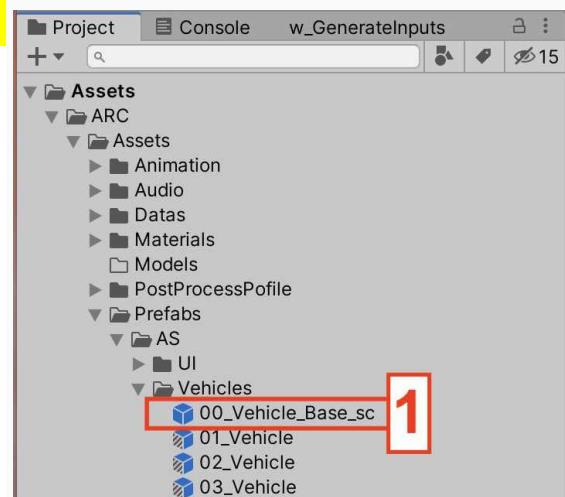
This section contains 4 tutorials that explain how to use and customize vehicles.
You will find more advanced information about vehicle in the Doc Part 2 section Vehicle → Page 84

Tutorials Parts:

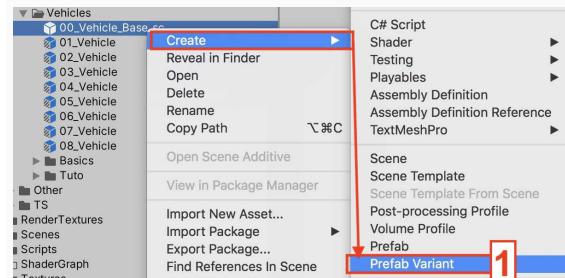
- 1-Create a new vehicle [Link](#)
- 2-Customize a vehicle with your own 3D models [Link](#)
- 3-Allow and set up a vehicle to be used by a player [Link](#)
- 4-Go further [Link](#)

1-Create a new vehicle

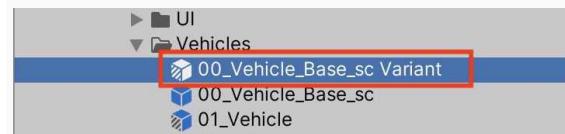
- In Project tab select **00_Vehicle_Base_sc** (spot 1)
(Project tab: Assets → Prefab → AS → Vehicles → **00_Vehicle_Base_sc**)



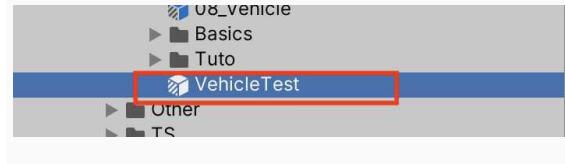
- Right click on **00_Vehicle_Base_sc**.
- In the menu select **Create → Prefab Variant** (spot 1)



A new Prefab is created.



- Rename it. For the tutorial rename the vehicle **VehicleTest**.



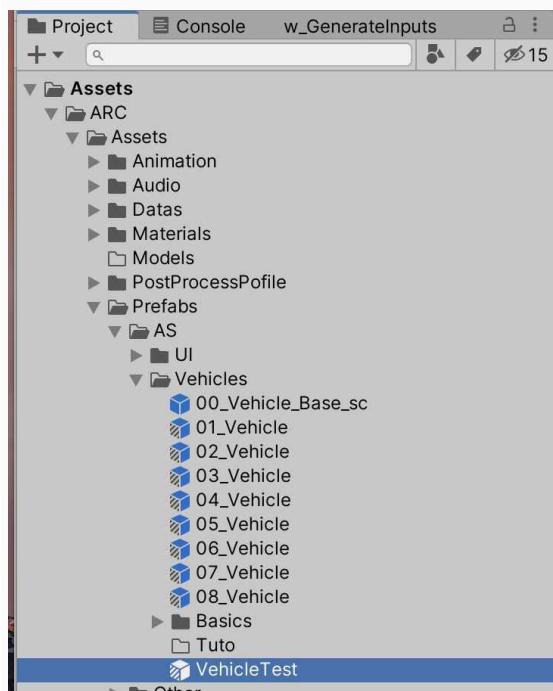
Now the new vehicle can be used as a vehicle in the game.

2-Customize a vehicle with your own 3D models

As an example we are going to modify the vehicle created in the 1st part of the tutorial named [VehicleTest](#).

-In Project Tab select [VehicleTest](#) prefab.

(Project tab: Assets → Prefab → AS → Vehicles → VehicleTest)



-In the Inspector press [Open Prefab](#) button.

-In the Hierarchy open object [Grp_3DModels](#) (spot 1)

(Hierarchy: VehicleTest → Vehicle_sc → Grp_MoveWithWings → Grp_3DModels)

-Disable [Ref](#) objects in (spot 2)

[Body](#) → [Ref](#)

[Propeller_Stop](#) → [Ref](#)

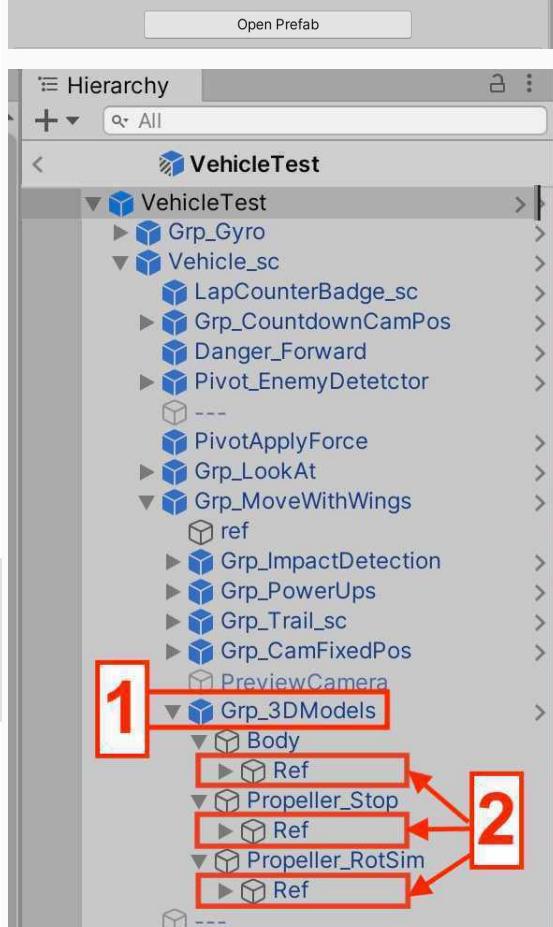
[Propeller_RotSim](#) → [Ref](#)

Now we are ready to add your own 3D models to create your own vehicles.

VERY IMPORTANT:

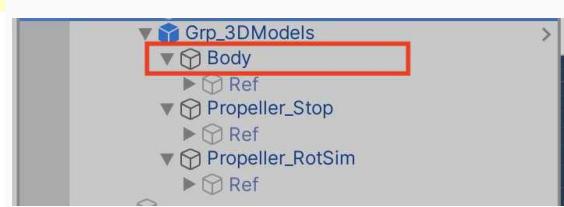
Your 3D models mustn't contain colliders.

Remove all the colliders included with your 3D models.



The Body

-In **Body** object: Add your 3D models that correspond to the vehicle body.



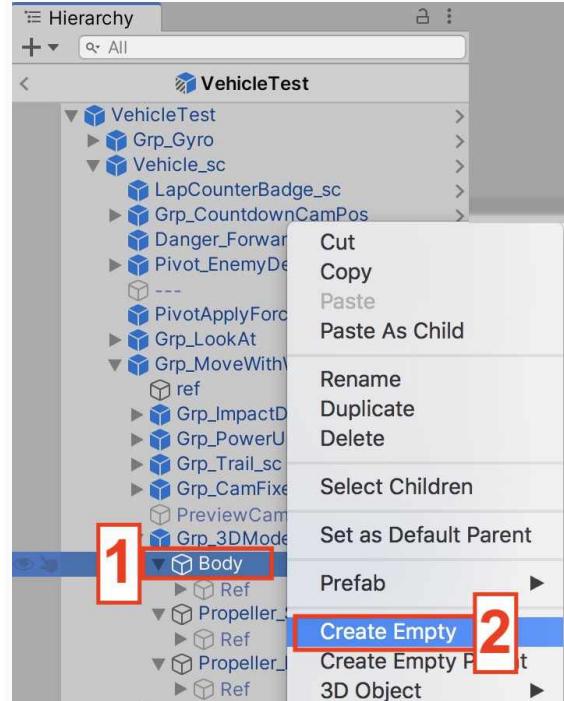
As an example:

We are going to create an empty object inside **Body** object. Then we are going to add our new 3D body model.

-Right clic on **Body** object (spot 1)

(Hierarchy: VehicleTest → Vehicle_sc → Grp_MoveWithWings → Grp_3DModels → Body)

-In the menu choose **Create Empty** (spot 2)



A new empty object is created in the Hierarchy.

-Rename the object:

Grp

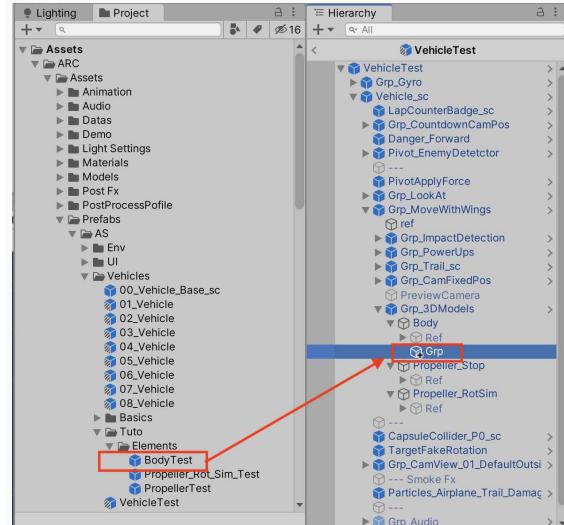


Drag and drop **BodyTest** prefab

(Project tab: Assets → Prefabs → AS → Vehicles → Tuto → Elements → BodyTest)

inside **Grp**

(Hierarchy: VehicleTest → Vehicle_sc → Grp_MoveWithWings → Grp_3DModels → Body → Grp →)

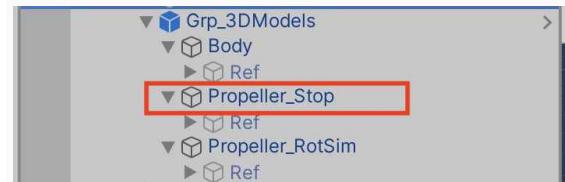


You should have this result.



3D model used when the propeller is stopped

-In **Propeller_Stop** object: Add your propeller 3D models used when the propeller is stopped.



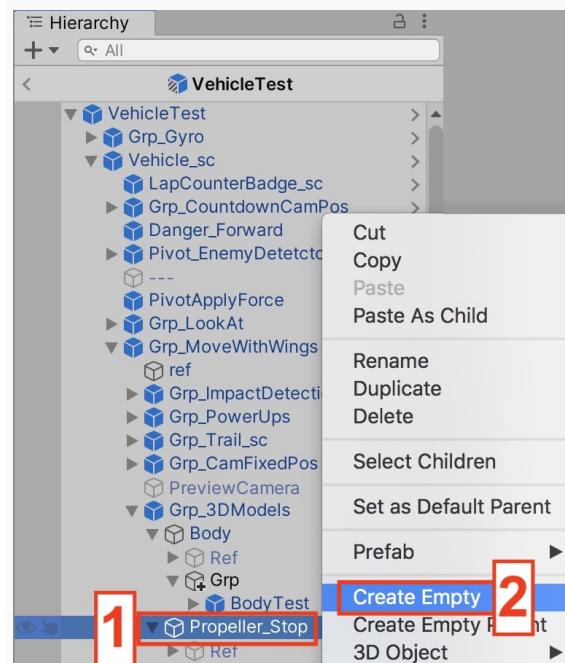
As an example:

We are going to create an empty object inside **Propeller_Stop** object. Then we are going to add our new 3D propeller model.

-Right clic on **Propeller_Stop** object (spot 1)

(Hierarchy: VehicleTest → Vehicle_sc → Grp_MoveWithWings → Grp_3DModels → Propeller_Stop)

-In the menu choose **Create Empty** (spot 2)



A new empty object is created in the Hierarchy.



-Rename the object:

Grp

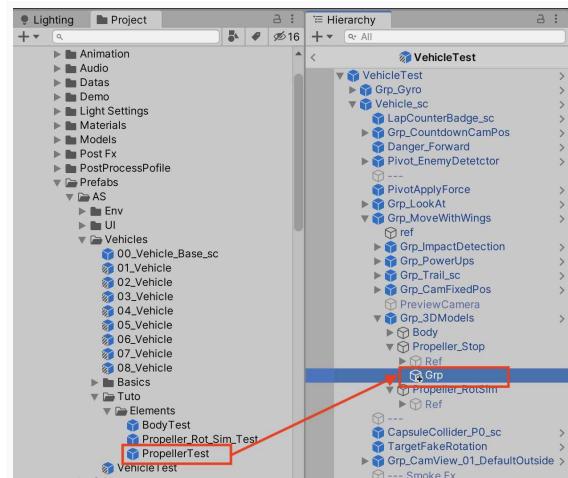


Drag and drop PropellerTest prefab

(Project tab: Assets → Prefabs → AS → Vehicles → Tuto → Elements → PropellerTest)

inside Grp

(Hierarchy: VehicleTest → Vehicle_sc → Grp_MoveWithWings → Grp_3DModels → Propeller_Stop → Grp →)



You should have this result.



3D model used when the propeller is rotating

-In Propeller_RotSim object: Add your propeller 3D models used when the propeller is rotating.

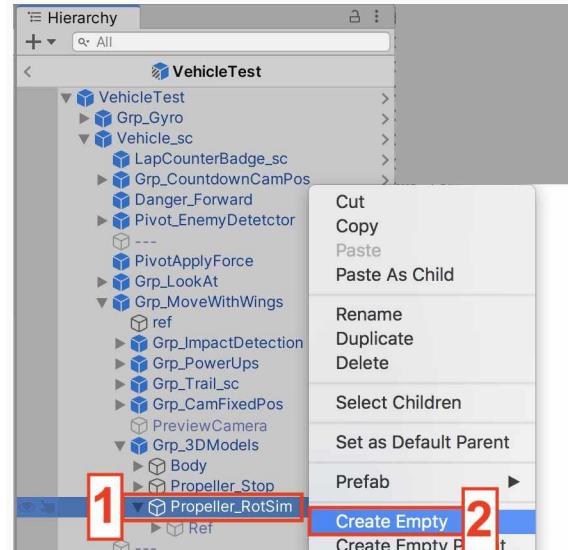
As an example:

We are going to create an empty object inside **Propeller_RotSim** object. Then we are going to add our new 3D propeller model.

-Right clic on Propeller_RotSim object (spot 1)

(Hierarchy: VehicleTest → Vehicle_sc → Grp_MoveWithWings → Grp_3DModels → Propeller_RotSim)

-In the menu choose **Create Empty** (spot 2)



A new empty object is created in the Hierarchy.



-Rename the object:

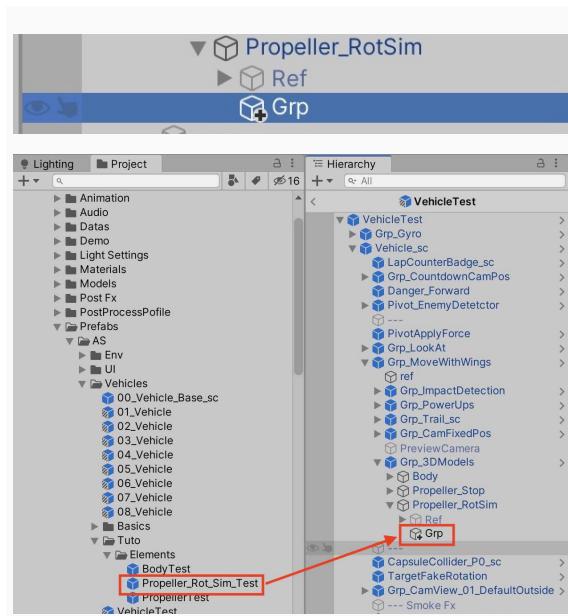
Grp

Drag and drop **Propeller_RotSim_Test** prefab

(Project tab: Assets → Prefabs → AS → Vehicles → Tuto → Elements → Propeller_RotSim_Test)

inside Grp

(Hierarchy: VehicleTest → Vehicle_sc → Grp_MoveWithWings → Grp_3DModels → Propeller_RotSim → Grp →)



You should have this result.



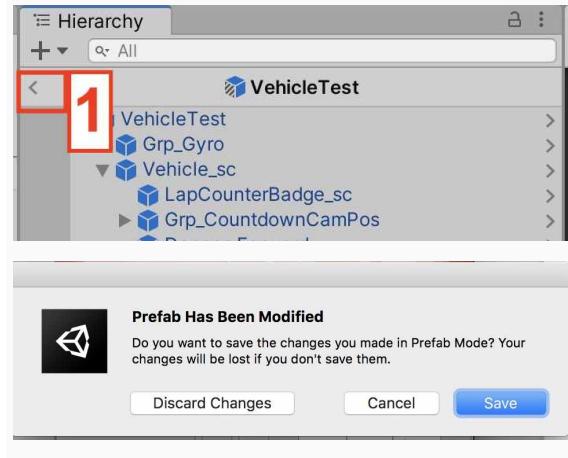
Your vehicle should look like this:



VERY IMPORTANT:

Remember that your 3D models mustn't contain colliders. Remove all the colliders included with your 3D models.

-In the Hierarchy press < to exit the prefab edition.

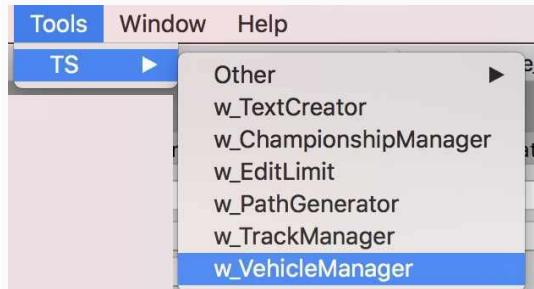


If you have this message. Press **Save**.

3-Allow and set up a vehicle to be used by a player

-Open **w_VehicleManager** window.
(Tools → TS → w_VehicleManager)

A new window appears.



-Press **Vehicles Setup** button.



-Go to the end of the window and press **Add a new vehicle to the list**.

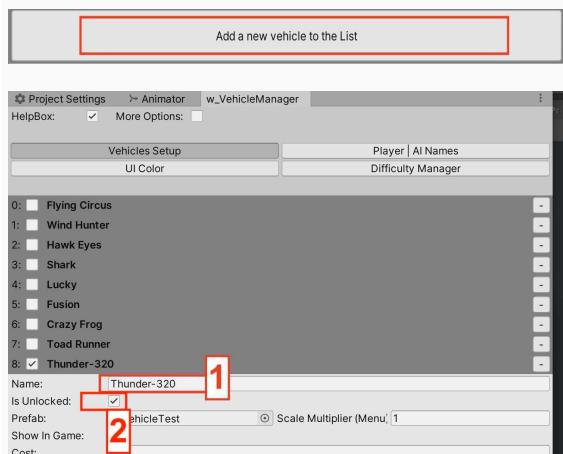
At the end of the list a new section appears.

-Write the vehicle name.

As an example write **Thunder-320** (spot 1)
This name is used when the player choose a vehicle.

-Keep **Is Unlocked** checked (spot 2)

It allows the player to use the vehicle the first time he starts the game.

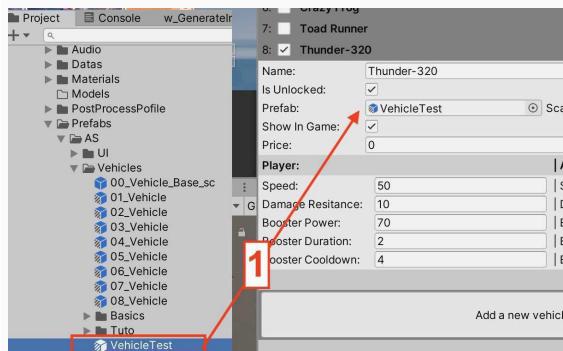


This step explains how to choose the prefab used for this vehicle.

As an example you can drag and drop the vehicle we have created in the first part of the tutorial.

-Drag and drop **VehicleTest** prefab (spot 1).

(Project tab: Assets → Prefab → AS → Vehicles → VehicleTest)



If needed you can change **Scale Multiplier** parameter.

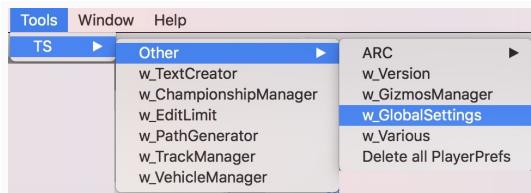
This parameter allows to increase or decrease the size of the vehicle when the vehicle is displayed in the garage or when the player choose the vehicle for a race.

-Keep **Show In Game checked**. If the button is unchecked the vehicle is not displayed in the garage.

If needed it is possible to:

- Choose a price to unlock the vehicle in the garage.
- Customize some vehicle parameters (speed, resistance and power-ups)

-Open **w_GlobalSettings**
(Tools → TS → Other → w_GlobalSettings)

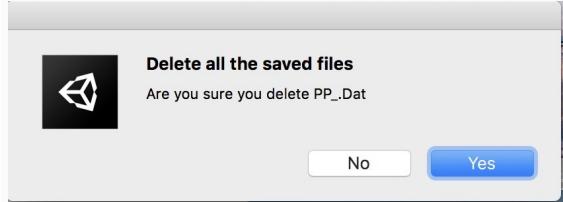


-Press **Player Progression** button to delete the data already saved (spot 1).



A window appears.

-Press **Yes**.

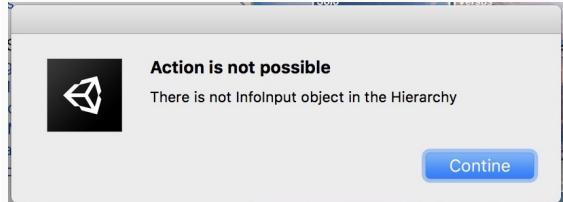


Note:

If a window appears with the text:
There is no InfoInput object in the Hierarchy.

-Open the scene **01_MM**
(Project Tab: Assets → Scenes → Main Menu → 01_MM)

-Press again **Player Progress** button to delete the data already saved



The new vehicle is available for the player.

Now the player is able to choose the new vehicle in the main menu.



4-Go further

If you want to learn more about vehicle customization go to section [Vehicle](#) in the documentation

More info in section [Vehicle](#)→ Page 84

You will find info about:

Overview

Short description of each vehicle module

Vehicle parameters: Basic

Vehicle Parameters: Advanced

How to: Create a new vehicle.

How to customize a vehicle with your own 3D models

Set up a vehicle to be usable in-game

How to: Modify Player + AI UI color

How to: Modify Player + AI names used in
leaderboard

How to: Modify AI behaviors

Tuto 3: Replace default font and music

This section explains how to replace the default Font and default Music included in the asset.

Table of contents:

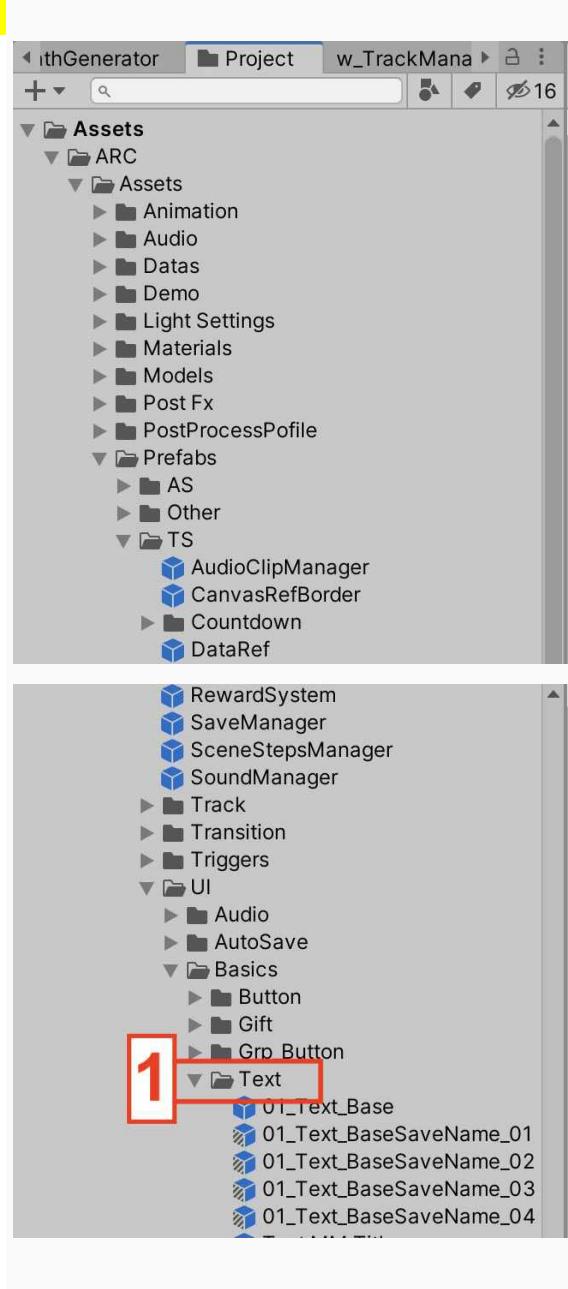
1-Replace Font (Arial) [Link](#)

2-Replace Music [Link](#)

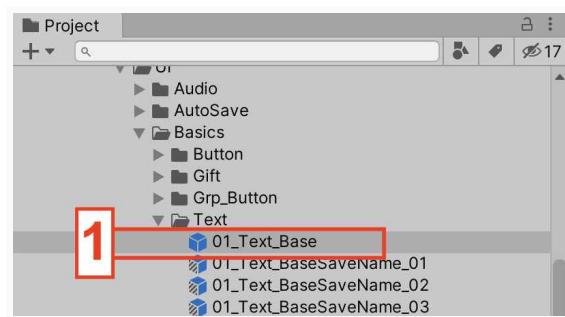
Replace Font (Arial)

-In the Project Tab go to folder **Text** (spot 1)

(Project tab: Assets → Prefabs → TS → UI → Basics → Text)



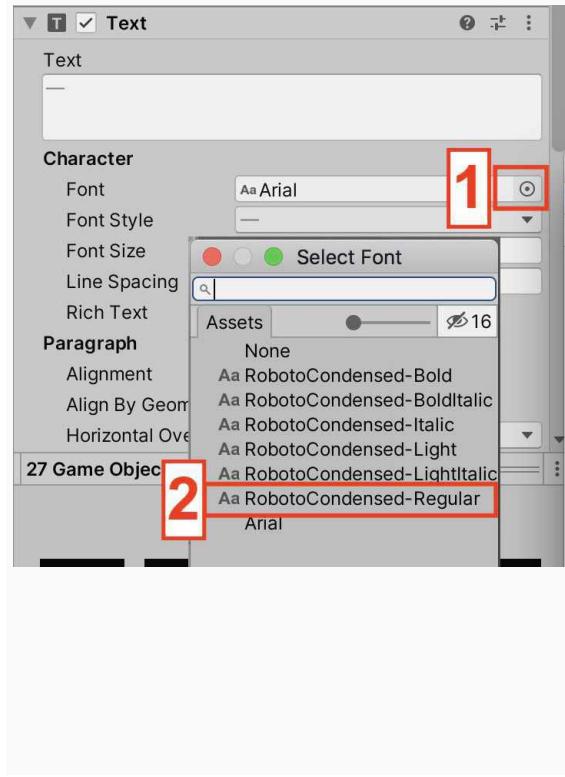
-Select the first Prefab named **01_Text_Base** (spot 1)



-In the Inspector press the **small circle** next to selected font (spot 1)

A new window appears.

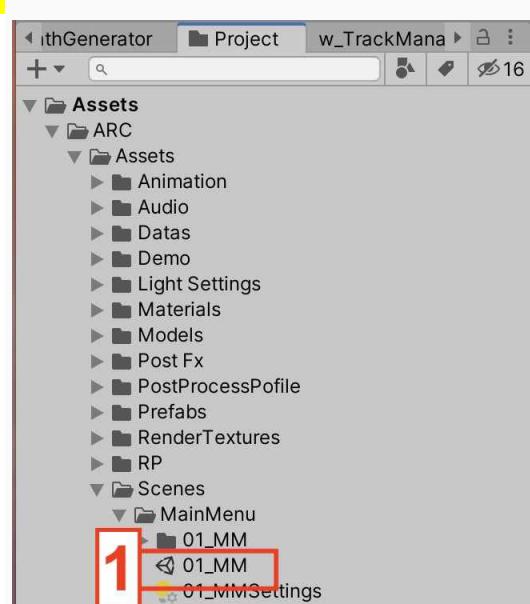
-Select your Font (spot 2)



Replace Music

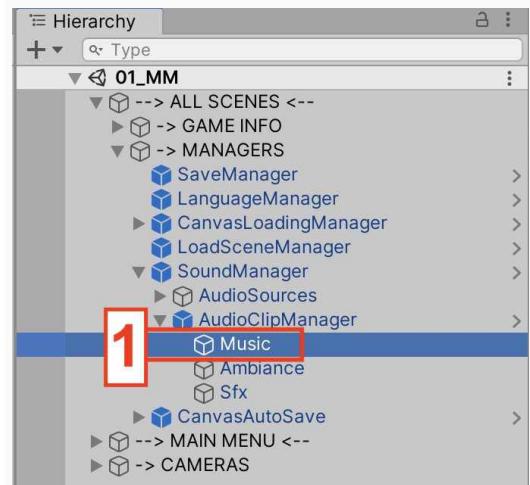
-In the Project tab open the Main Menu scene named **01_MM** (spot 1)

(Project tab: Assets → Scenes → MainMenu → 01_MM)

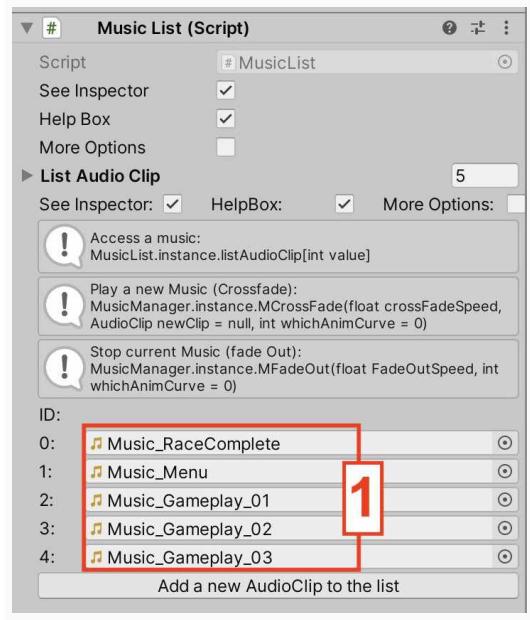


-In the Hierarchy select **Music** (spot 1)

(Hierarchy: ALL SCENES → MANAGERS → SoundManager → AudioClipManager → Music)



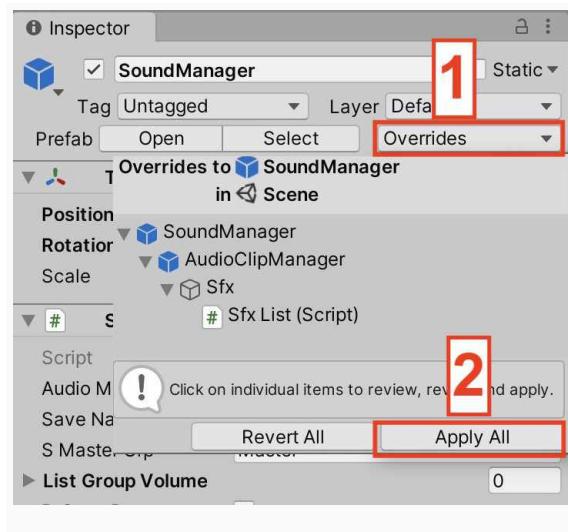
-In the Inspector drag and drop all the music you want to use (spot 1).



-In the Inspector press **overrides** (spot 1)

-Press **Apply All** (spot 2)

Now you can use your musics in all the scenes.



Asset content

To easily create an environment you can use the ready to use prefabs.

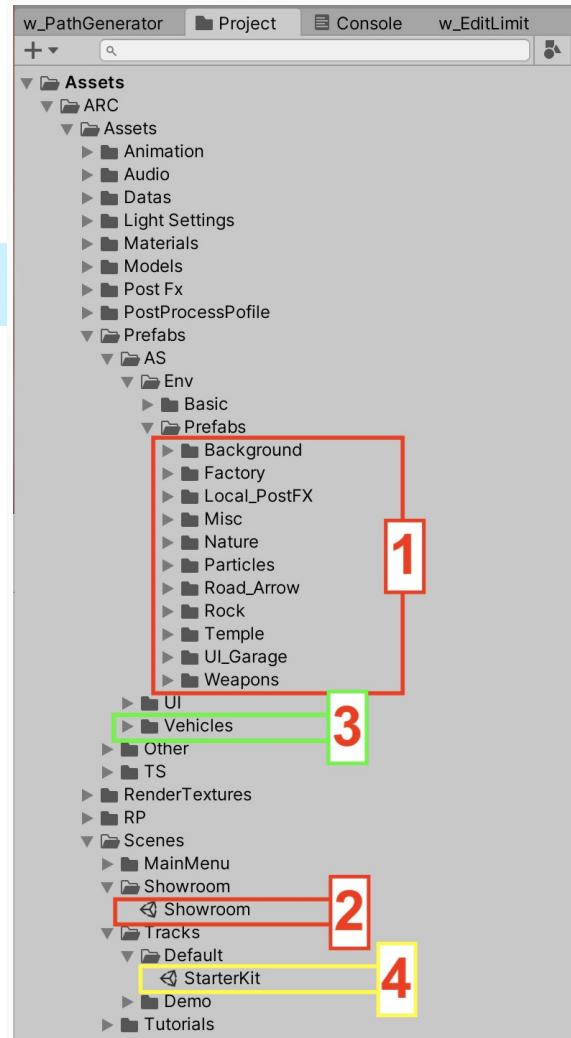
Prefabs are inside folder named **Prefabs** (spot 1).
(Assets → Prefabs → AS → Env → Prefabs →)

More info in section Environment prefab
→ Page 277

You can have an overview of all the environment prefabs in scene **Showroom** (spot 2)
(Assets → Scenes → Showroom → Showroom)

All the vehicles prefabs are inside folder **Vehicles** (spot 3).
(Assets → Prefabs → AS → Vehicles →)

To create a track you can duplicate an existing track or using the **StarterKit** track (spot 4)
(Assets → Scenes → Track → Default → StarterKit)



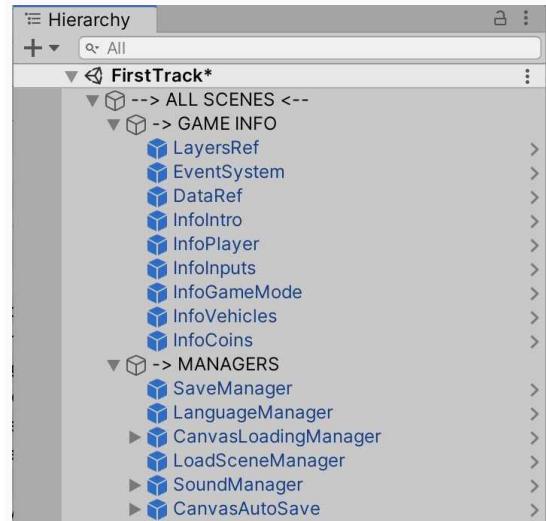
Managers

This section gives a short description of the managers contained in the asset.

Table of contents:

Short description of each common module (all scenes)	Link
Short description of each Main Menu scene module	Link
Short description of each Track scene module	Link
Short description of each tool available in the Unity tool bar	Link

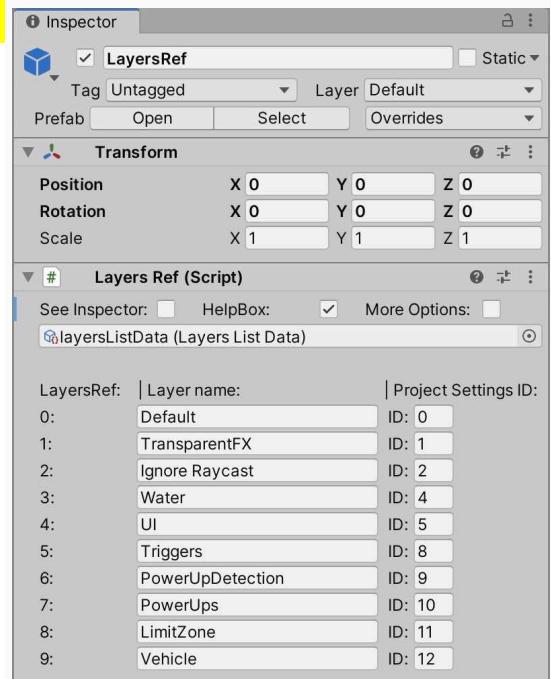
Short description of each common module (all scenes)



LayersRef

Hierarchy: ALL SCENES → GAME INFO → LayerRefs

From any script, [LayersRef.cs](#) allows to access layers set up in [Project Settings → Tags and Layers](#).
(Edit → Project Settings → Tags and Layers)



EventSystem

Hierarchy: ALL SCENES → GAME INFO → EventSystem

This object is used to manage UI input behavior.

EventSystem and StandaloneInputModule are Unity core components.

From any script, TS_EventSystem.cs (spot 1) allows to access EventSystem and StandaloneInputModule components.

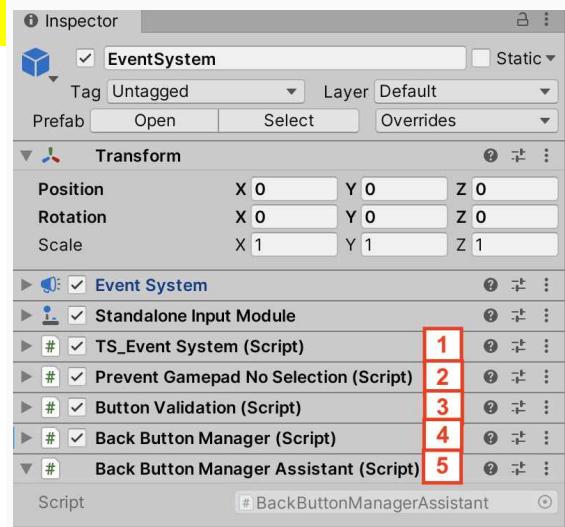
PreventGamepadNoSelection.cs (spot 2) automatically selects a UI button if no UI button is already selected.

ButtonValidation.cs (spot 3) invokes OnClick() when a button is pressed (keyboard or gamepad).

BackButtonManager.cs (spot 4) manages what to do if the player presses back button. It is possible to add condition in the Inspector.

BackButtonManagerAssistant.cs (spot 5) contains methods used in BackButtonManager.cs.

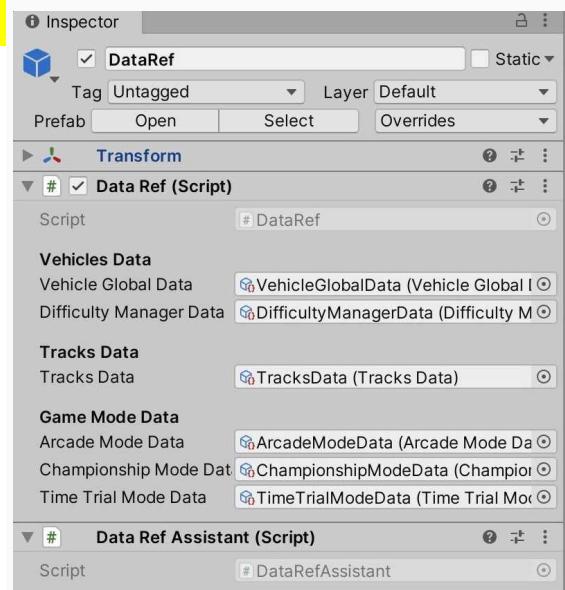
MouseManager.cs contains methods to show and hide the mouse cursor.



DataRef

Hierarchy: ALL SCENES → GAME INFO → DataRef

From any script, this object allows access to scriptable objects that contains game data.



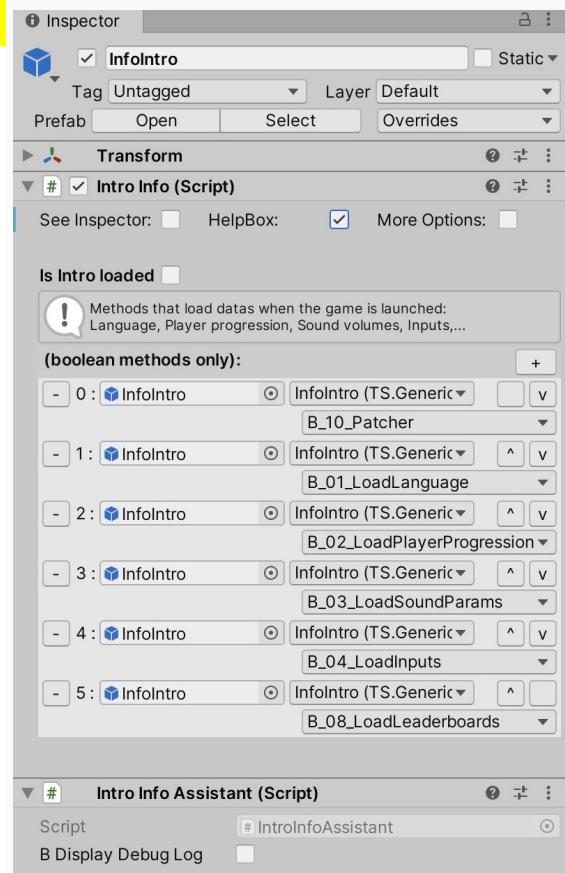
InfoIntro

Hierarchy: ALL SCENES → GAME INFO → InfoIntro

This object allows to load information when game is launched.

It allows to initialize:

- Patcher.
- Language.
- Player progression.
- Sound parameters.
- Inputs.
- Leaderboards.



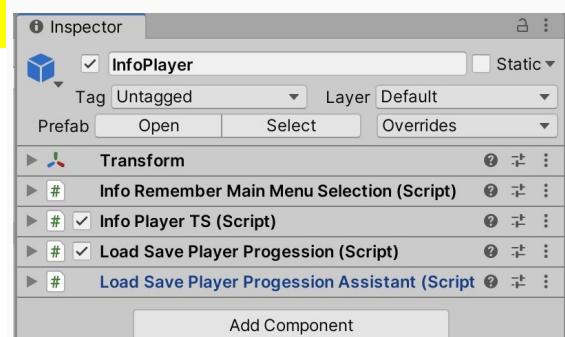
InfoPlayer

Hierarchy: ALL SCENES → GAME INFO → InfoPlayer

From any script, this object allows to access player info.

InfoRememberMainMenuSelection.cs gives access to:

- The number of player (Solo or splitscreen).
- The current game mode.
- The current AI Difficulty.



InfoPlayerTS.cs lets you know if the player is allowed to do something in the menu.

LoadSavePlayerProgression.cs allows to save and load player progression:

- Vehicle info.
- Inputs.
- Game Mode.
- Coins.

...

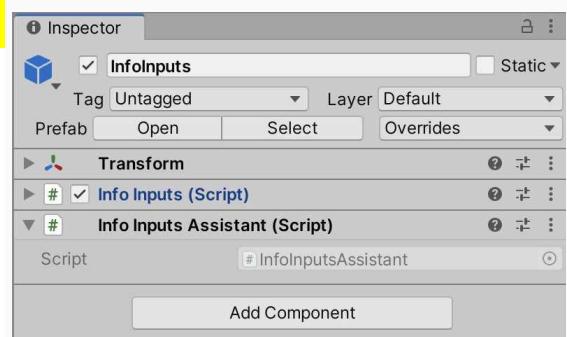
More info in section [Player Progression](#)
→ Page 219

InfoInputs

Hierarchy: ALL SCENES → GAME INFO → InfoInputs

InfoInputs.cs allows to:

- Set up inputs.
- Access Inputs from any script.
- Load/Save inputs.



InfoGameMode

Hierarchy: ALL SCENES → GAME INFO → InfoGameMode

From any script, this object allows to access game modes info.

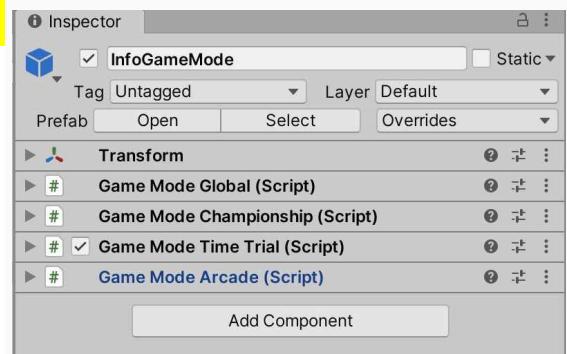
GameModeGlobal.cs allows to:

- Access AI names.
- Access the vehicles selected for the race.

GameModeChampionship.cs allows to access info about the Championship Mode.

GameTimeTrial.cs allows to access info about Time Trial Mode.

GameModeArcade.cs allows to access info about Arcade Mode.



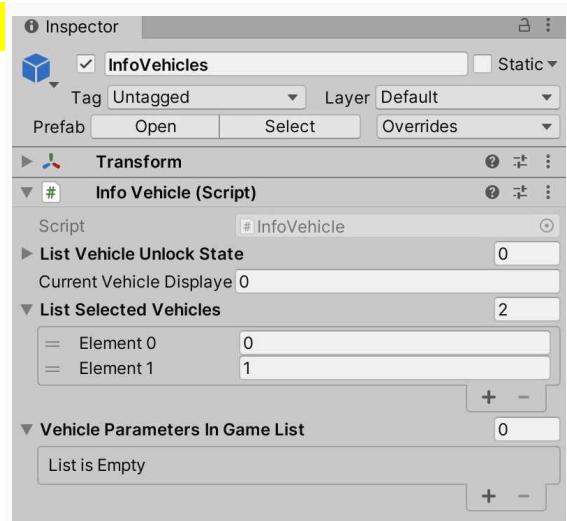
InfoVehicle

Hierarchy: ALL SCENES → GAME INFO → InfoVehicles

From any script, this object allows access to vehicle info.

InfoVehicle.cs allows to:

- Know which vehicle is unlocked / lock.
- Which vehicle is selected by P1 or P2.
- Access each vehicle parameters (VehicleGlobalData).



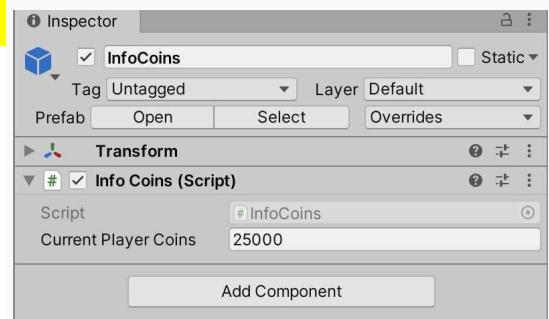
InfoCoins

Hierarchy: ALL SCENES → GAME INFO → InfoCoins

From any script, this object allows to access player coins info.

InfoCoins.cs allows to:

- Access player coins.
- Add/Update coins.

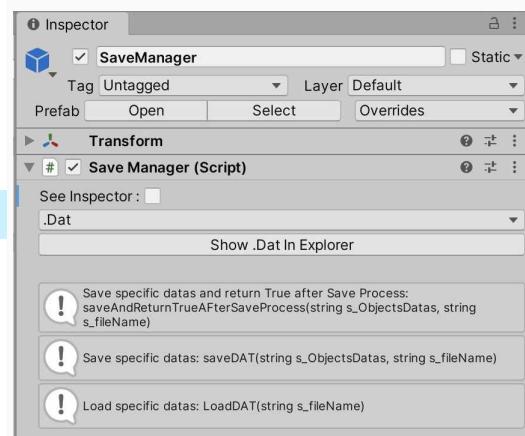


SaveManager

Hierarchy: ALL SCENES → MANAGERS → SaveManager

From any script, this object allows to save/load data.

More info in section [Save System](#) → Page 207



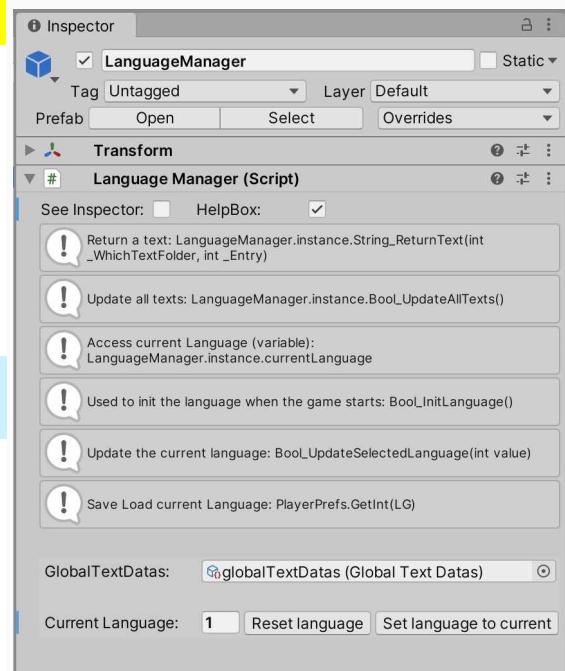
LanguageManager

Hierarchy: ALL SCENES → MANAGERS → LanguageManager

From any script, this object allows to:

- Access multi-language texts.
- Update the language used for texts.
- Initializing language.
- Save/Load the current language.

More info in section [Localization and text](#)
→ Page 211



CanvasLoadingManager

Hierarchy: ALL SCENES → MANAGERS → CanvasLoadingManager

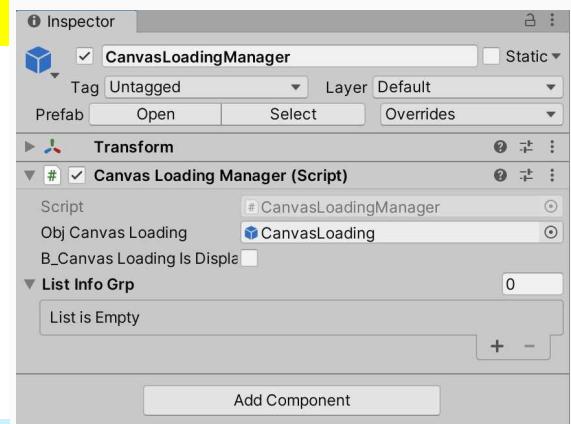
From any script, this object allows to manage the loading screen.

The loading screen is displayed automatically when a scene is loaded.

The loading screen is disabled automatically at the end of the loading process.

More info in section [Loading Screen](#)

→ Page 159



LoadSceneManager

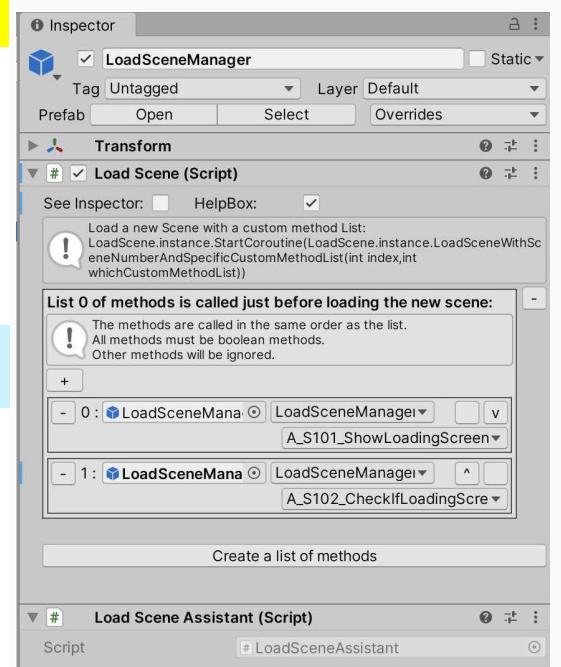
Hierarchy: ALL SCENES → MANAGERS → LoadSceneManager

From any script, this object allows to load a scene.

It is possible to call methods before loading a scene.

More info in section [The Load Scene Manager](#)

→ Page 151



SoundManager

Hierarchy: ALL SCENES → MANAGERS → SoundManager

From any script, this object allows to:

- Save and load Audio volumes (Global, Music, Sfx)
- Access and play music UI Sfx audios.

More info in section [Audio](#)→ Page 181

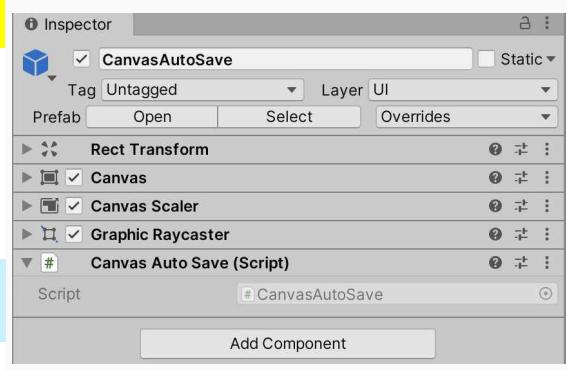


CanvasAutoSave

Hierarchy: ALL SCENES → MANAGERS → CanavsAutoSave

From any script, this object allows to display a UI feedback when the game is saving data.

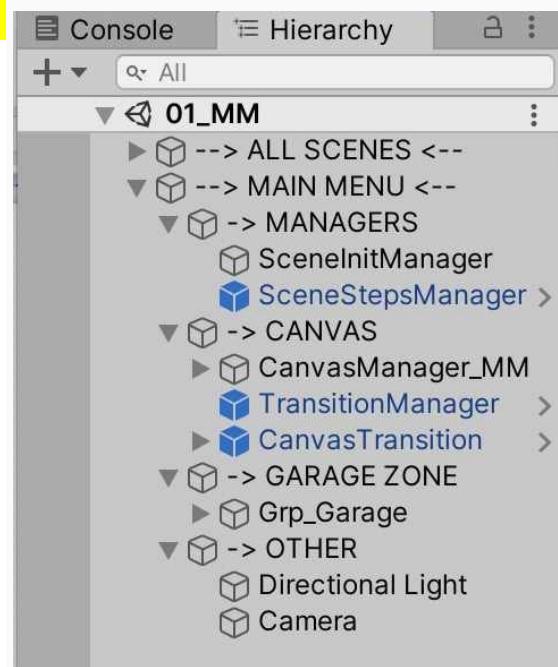
More info in section [Canvas Auto Save](#)
→ Page 161



Short description of each Main Menu modules

This section describes objects specific to the Main Menu. Those objects are contained in Hierarchy: MAIN MENU. Objects inside this group are destroyed when a new scene is loaded.

Overview:

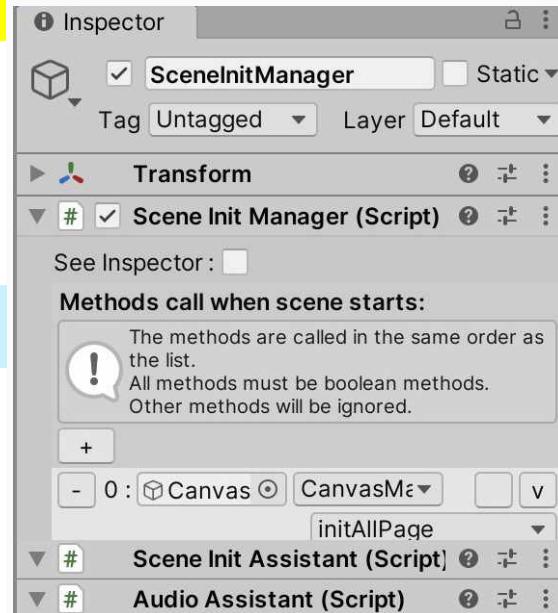


ScenelnitManager

Hierarchy: MAIN MENU → MANAGERS → ScenelnitManager

This object is used to call methods that initialize the scene. This object is called when the scene starts.

More info in section [Scene Initialization](#)
→ Page 149

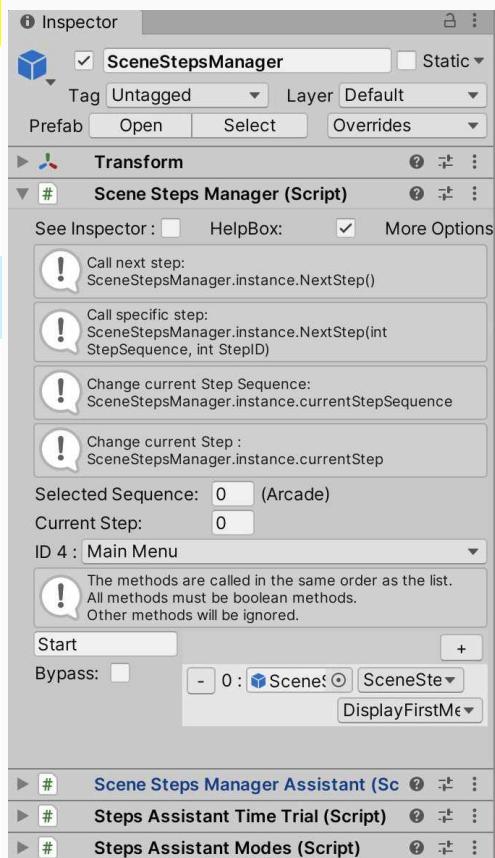


SceneStepsManager

Hierarchy: MAIN MENU → MANAGERS → SceneStepsManager

In the main menu the Scene Step system open the UI menu when the scene starts.

More info in section [The Scene Step system](#)
→ Page 154



CanvasManager_MM

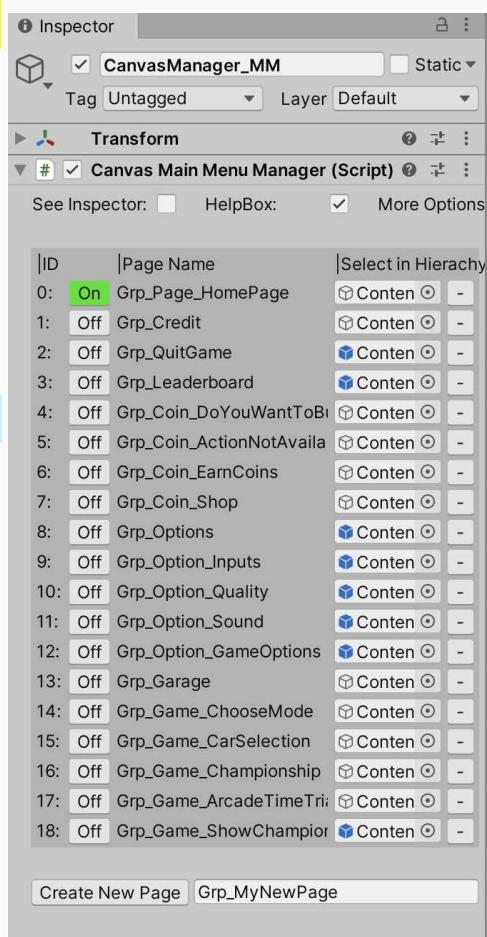
Hierarchy: MAIN MENU → CANVAS → CanvasManager_MM

CanvasManager_MM manages the behavior of the main menu pages.

It allows to:

- Open a page
 - Go back to an other page
 - Manage transition between pages
- ...

More info in section [UI](#) → Page 238



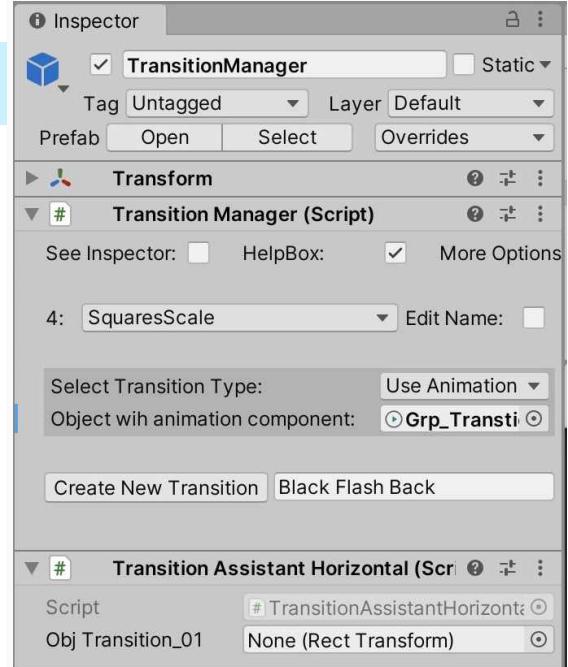
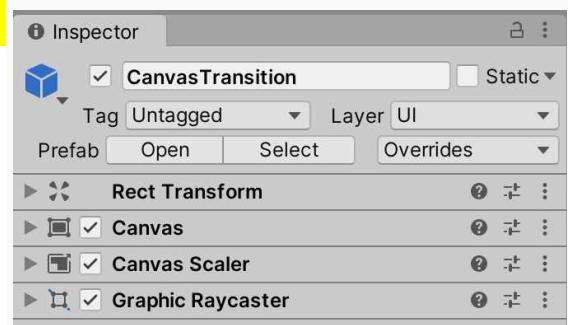
TransitionManager + CanvasTransition

Hierarchy: MAIN MENU → CANVAS → TransitionManager

It is possible to play UI transition. These transitions are managed by **TransitionManager** object.
CanvasTransition is the canvas that display these UI transitions.

More info in section **Transition system**

→ Page 162

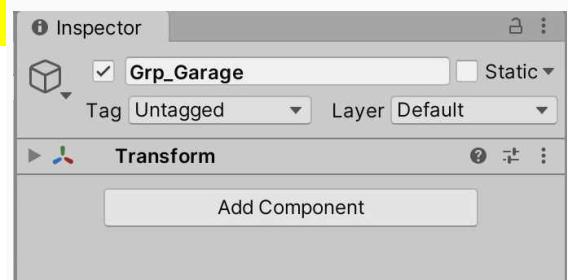


Grp_Garage

Hierarchy: MAIN MENU → GARAGE | SELECTION ZONE → Garage

This object includes all the objects used to display vehicle in the UI.

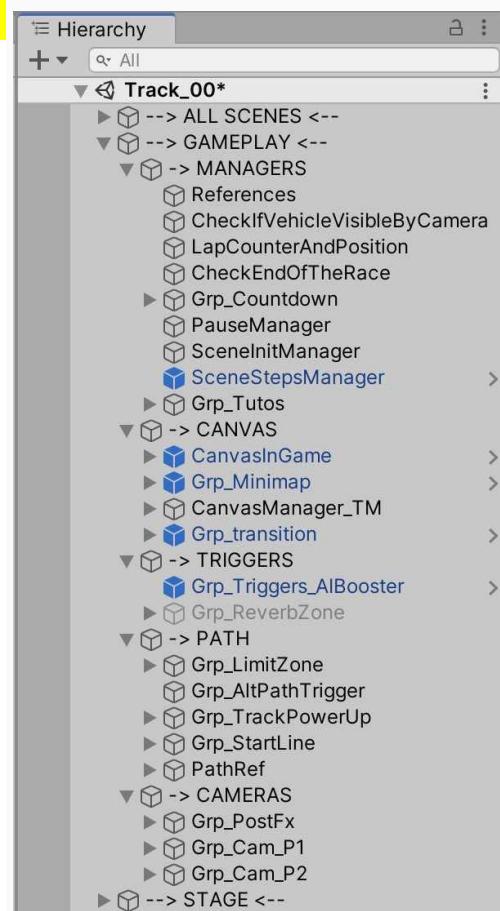
It is used with the Garage and when the player choose his vehicle.



Short description of each Track scene modules

This section describes objects specific to a Track scenes. Those objects are contained in Hierarchy: GAMEPLAY. Objects inside this group are destroyed when a new scene is loaded.

Overview:



References

Hierarchy: GAMEPLAY → MANAGERS → References

From any script, [VehicleRef.cs](#) allows to access:

- The vehicle list.
- The data that describes each car parameter.

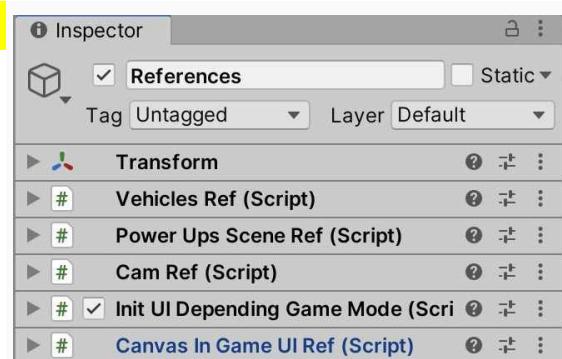
From any script, [PowerUpsSceneRef.cs](#) allows to access UI elements related to power-ups.

From any script, [CamRef.cs](#) allows to access:

- Player cameras.
- Post-effect on camera P1 and P2.

From any script, [CamRef.cs](#) allows to initialize the UI depending the number of Player (1 or 2).

From any script, [CanvasInGameUIRef.cs](#) allows to access UI elements.



CheckIfVehicleVisibleByCamera

Hierarchy: GAMEPLAY → MANAGERS → CheckIfVehicleVisibleByCamera

From any script, [VehicleVisibleByCamList.cs](#) allows to access to:

- The list of the vehicles visible by the camera P1 or P2

From any script, [VehicleFlagManager.cs](#) manages the flags display above vehicles.

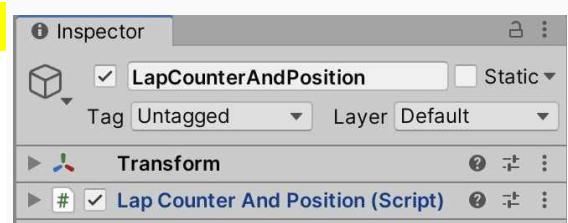


LapCounterAndPosition

Hierarchy: GAMEPLAY → MANAGERS → LapCounterAndPosition

From any script, [LapCounterAndPosition.cs](#) allows to access:

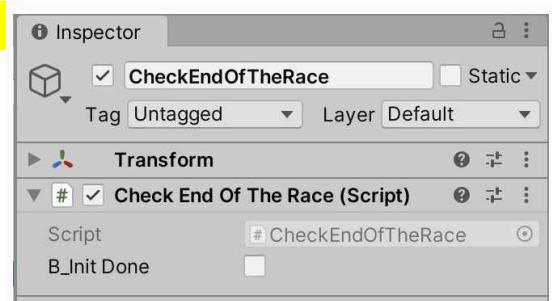
- Info about the race (Time, position, race complete, ...)



CheckEndOfTheRace

Hierarchy: GAMEPLAY → MANAGERS → CheckEdOfTheRace

From any script, [CheckEndOfTheRace.cs](#) checks if the score menu must be displayed when a vehicle finishes the race.

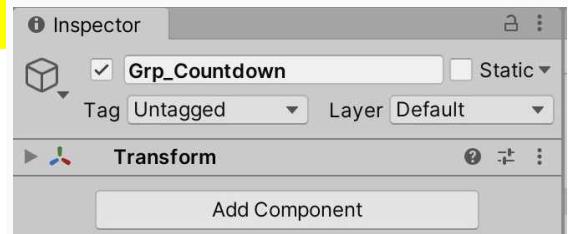


Grp_Countdown

Hierarchy: GAMEPLAY → MANAGERS → Grp_Countdown
[Countdown.cs](#) allows to create countdown animation.

[CamDuringCountdown.cs](#) allows to create camera animation used during countdown.

More info in section [Countdown](#) → Page 170

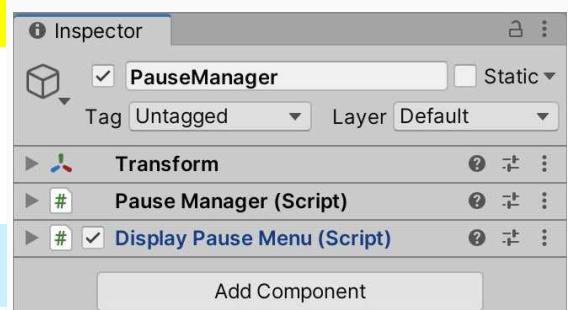


PauseManager

Hierarchy: GAMEPLAY → MANAGERS → PauseManager

[PauseManager.cs](#) allows to manage pause.
It is possible to customize the pause manager

More info in section [PauseManager](#) → Overview
→ Page 235

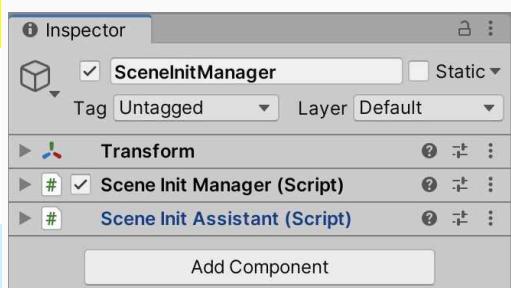


SceneInitManager

Hierarchy: GAMEPLAY → MANAGERS → SceneInitManager

This object is used to call methods that initialize the scene. This object is called when the scene starts.

More info in section [Scene Initialization](#)
→ Page 149

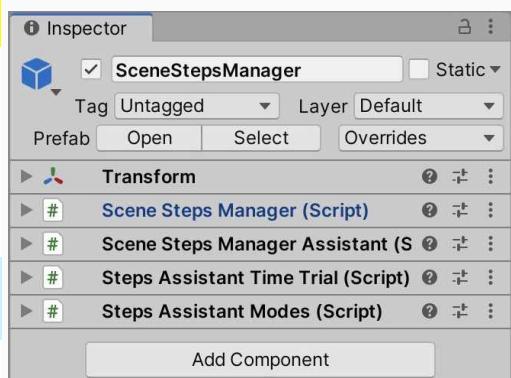


SceneStepsManager

Hierarchy: GAMEPLAY → MANAGERS → SceneStepsManager

In the gameplay the Scene Step system opens the UI menu when the player presses pause button.

More info in section [The Scene Step system](#)
→ Page 154



Tutorial Manager

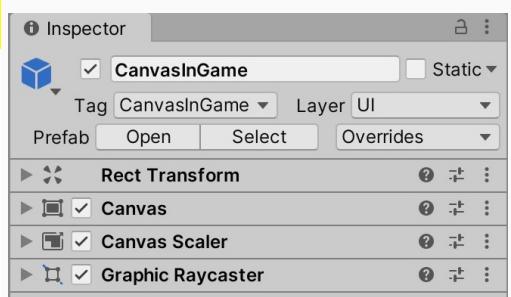
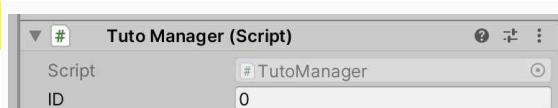
Hierarchy: GAMEPLAY → MANAGERS → Grp_Tutos → TutoManager_01

This object displays popups to explain which buttons are used to trigger booster and Power-Up. Popups are displayed only the first time the player participates in a race.

CanvasInGame

Hierarchy: GAMEPLAY → CANVAS → CanvasInGame

This object contains the UI displayed during the race. Lap, Time, Position, Warning, booster, Power-ups.

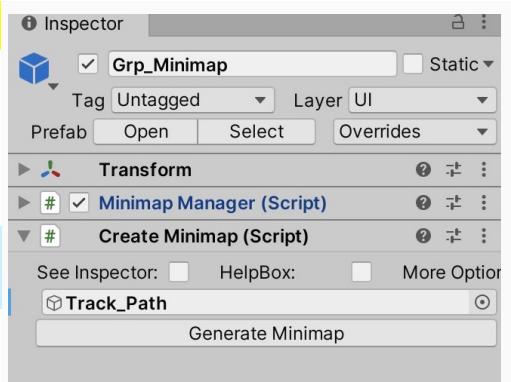


Grp_Minimap

Hierarchy: GAMEPLAY → CANVAS → Grp_Minimap

This object manages the minimap.

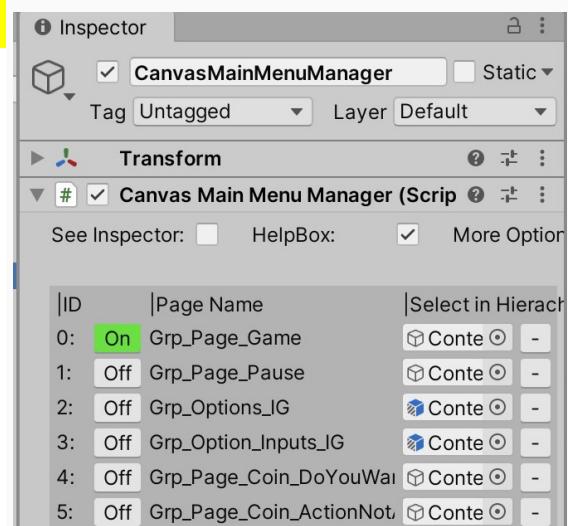
More info in section [Minimap → Overview](#)
→ Page 131



CanvasMainMenuManager

Hierarchy: GAMEPLAY → CANVAS → CanvasMainMenuManager

This object manages the menu displayed when the player press pause

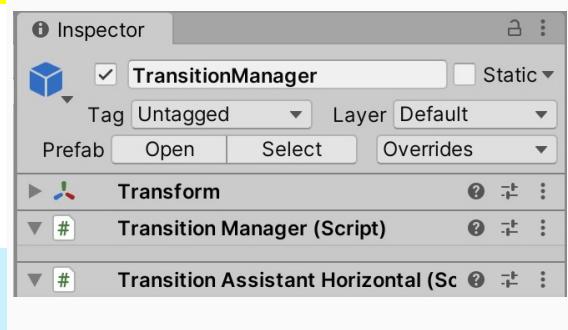


TransitionManager

Hierarchy: GAMEPLAY → CANVAS → transitionManager

It is possible to play UI transition. These transitions are managed by TransitionManager object. CanvasTransition is the canvas that display those UI transitions.

More info in section [Transition system](#)
→ Page 162



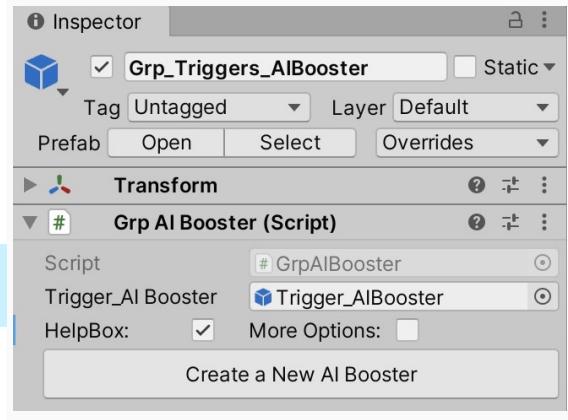
Grp_Triggers_AIBooster

Hierarchy: GAMEPLAY → TRIGGERS → AIBooster

This object allows to create AI Booster.

When an AI go through an AIBooster trigger the vehicle uses automatically its Booster.

More info in section [Tracks → AI Booster](#)
→ Page 123



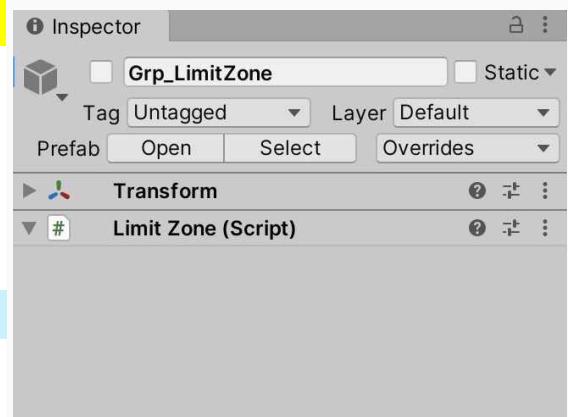
Grp_LimitZone

Hierarchy: GAMEPLAY → PATH → Grp_LimitZone

This object contains the game limit area collider.

When a vehicle go through the LimitZone trigger:
The vehicle exploded.

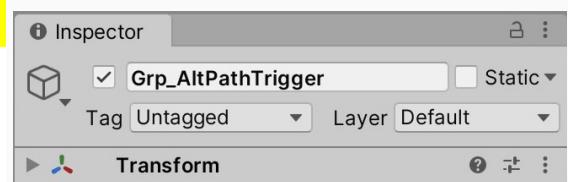
More info in section [Game Limit](#) → Page 128



Grp_AltPathTrigger

Hierarchy: GAMEPLAY → PATH → Grp_AltPathTrigger

During the scene initialization all the Alternative Paths
are placed in this objects. Do not delete it.



Grp_TrackPowerUp

Hierarchy: GAMEPLAY → PATH → Grp_TrackPowerUp

[PowerUpsItemsGlobalParams.cs](#) allows to:

- Create a new group of Power-ups
- Rotate power-ups in the scene if needed.

Hierarchy: GAMEPLAY → PATH → Grp_TrackPowerUp
→ Grp_Ref

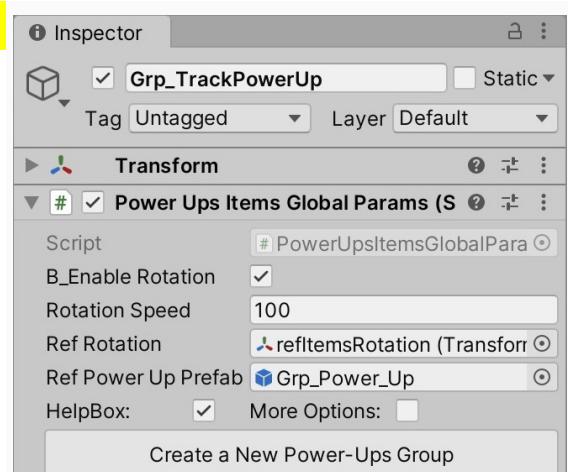
Grp_Ref contains the object used as reference to
rotate all the power-ups in the scene.

Hierarchy: GAMEPLAY → PATH → Grp_TrackPowerUp
→ Grp_Power_Up

Grp_Power_Up object:

- Contains the Power-Up groups.
- Manage the power-ups contained in this group of
power-ups.

During the race the vehicle call the script
[TriggerAILookAtPowerUp](#) to select its Power-Up.



Grp_StartLine

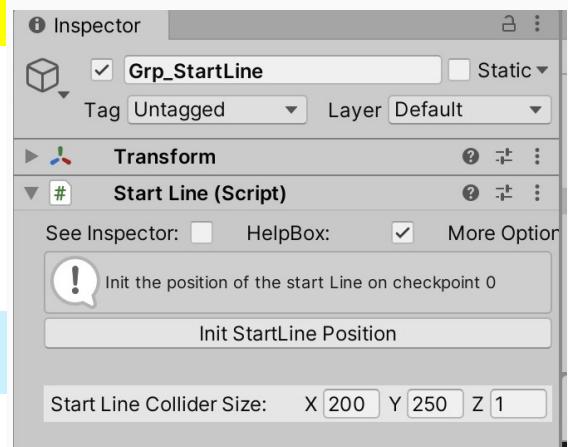
Hierarchy: GAMEPLAY → PATH → Grp_StartLine

StartLine.cs allows to:

- Set up the Start Line.
- Choose the vehicle positions on race grid.
- Manage the Start Line during the race.

More info in section [Start Line](#)

→ Page 120



PathRef

Hierarchy: GAMEPLAY → PATH → PathRef

From any script, [PathRef.cs](#) allows to access:

- The path info.

Hierarchy: GAMEPLAY → PATH → PathRef → BonusSpot

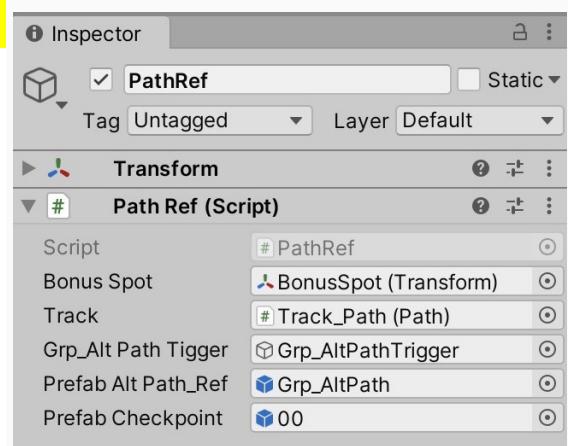
This object is used by script [vehicleAI.cs](#) and [VehiclePathFollow.cs](#).

Hierarchy: GAMEPLAY → PATH → PathRef → Track_Path

[Track_Path](#) object contains all the checkpoints.

The script [Path.cs](#) attached to [Track_Path](#) manages the track path.

More info in section [Tracks](#) → Page 118



Grp_PostFx

Hierarchy: GAMEPLAY → CAMERAS → Grp_PostFx

[Grp_Postfx](#) contains camera Post Fx for Player 1 and Player 2.

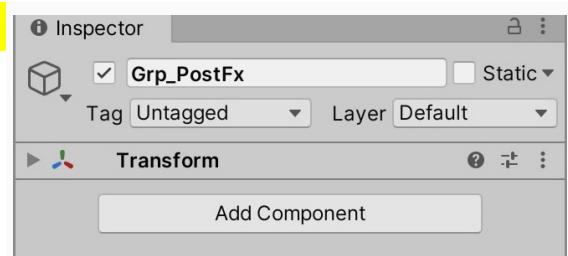
Hierarchy: GAMEPLAY → CAMERAS → Grp_PostFx → Effect_P1

[Effect_P1](#) contains the script [TS_PostProcess.cs](#) which manages Player 1 Post Fx.

Hierarchy: GAMEPLAY → CAMERAS → Grp_PostFx → Effect_P2

[Effect_P2](#) contains the script [TS_PostProcess.cs](#) which manages Player 2 Post Fx.

More info in section [Rendering](#) → Page 269

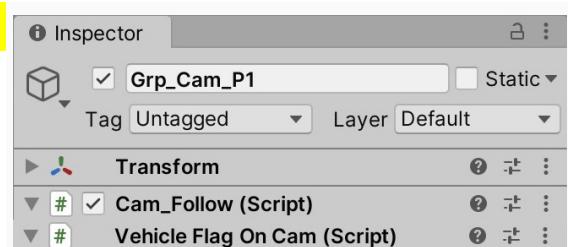


Grp_Cam_P1|P2

Hierarchy: GAMEPLAY → CAMERAS → Grp_CamP1 | Grp_CamP2

CamFollow.cs manages the camera behavior (Player 1 and Player 2)

More info in section [Camera → Overview](#)
→ Page 138



Stage

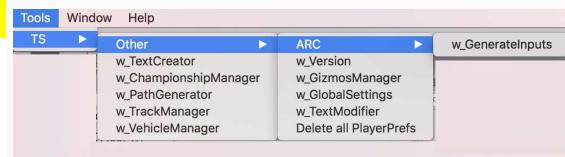
Hierarchy: GAMEPLAY → STAGE

This object contains all the objects used to create the stage (3D models, lights, light probs,...)

Short description of each tool available in the Unity tool bar

This section gives a short description of all the menu available in the Unity Menu bar: **Tools → TS**

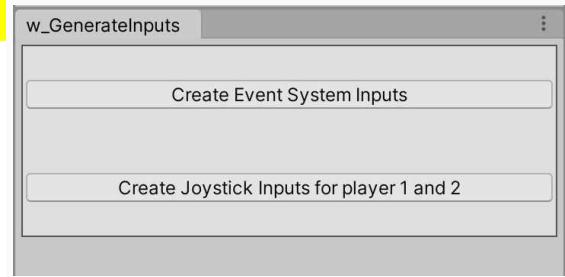
Overview:



w_GenerateInputs

Tools → TS → Other → ARC → w_GenerateInputs

This window allows you to automatically create needed inputs for the asset in **Project Settings → Input Manager**.

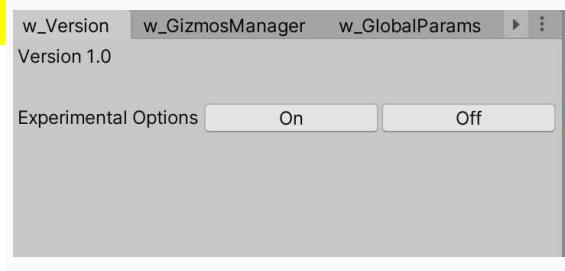


w_Version

Tools → TS → Other → w_Version

Show the current asset version.

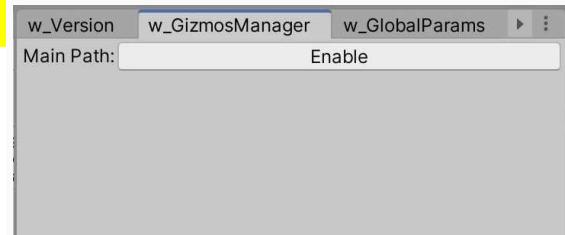
Experimental Options displays, if available, the button **More Options** on some window tabs.
More options is used during the asset development.



w_GizmoManager

Tools → TS → Other → w_GizmoManager

Allows to enable or disable gizmos.
Useful when there are too many gizmos on screen.

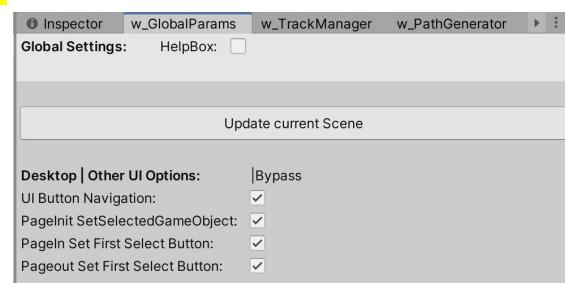


w_GlobalParams

Tools → TS → Other → w_GlobalParams

Access some global asset options.

In the window, press the **HelpBox** button for more information.



Choose the ID corresponding to the main menu scene.



Show .Dat In Explorer:

Allows to open the folder that contains the saved data on the computer.

Delete (data):

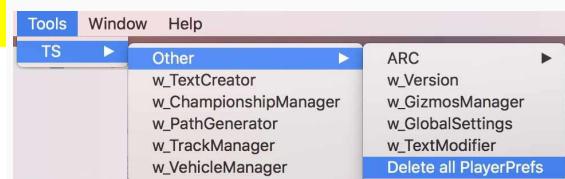
Allows to delete the Inputs, Audio, Player Progression data and leaderboard.



Delete All PlayerPrefs

Tools → TS → Other → Delete All PlayerPrefs

Delete all PlayerPrefs attached to this project.

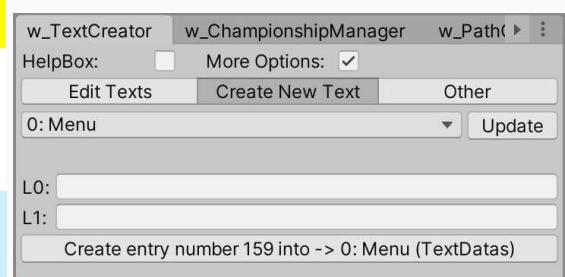


w_TextCreator

Tools → TS → w_TextCreator

Edit and create multi-language texts.

More info in section **Localization and Texts**
→ Page 211



w_ChampionshipManager

Tools → TS → w_ChampionshipManager

Edit and create championships.

More info in section [Game Mode](#)

→ Page 103

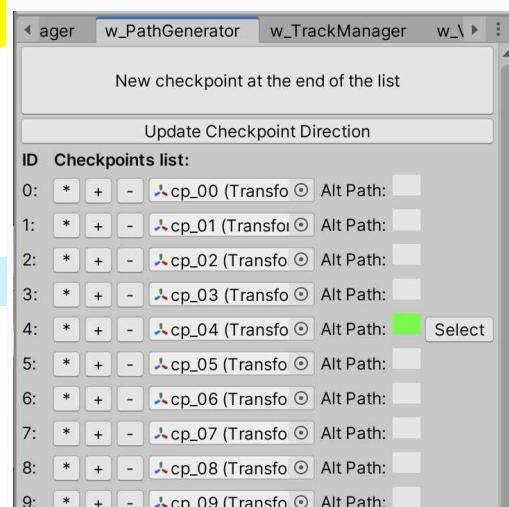


w_PathGenerator

Tools → TS → w_PathGenerator

Edit and created track paths. Only available in a track scene.

More info in section [Tracks](#) → Page 118



w_TrackManager

Tools → TS → w_TrackManager

Edit and create track infomation used by game modes (Arcade, Time Attack, Championship).

Access Arcade and Time Trial track parameters (Ai difficulty, leaderboard,...).

More info in section [Game Modes](#) → Page 103

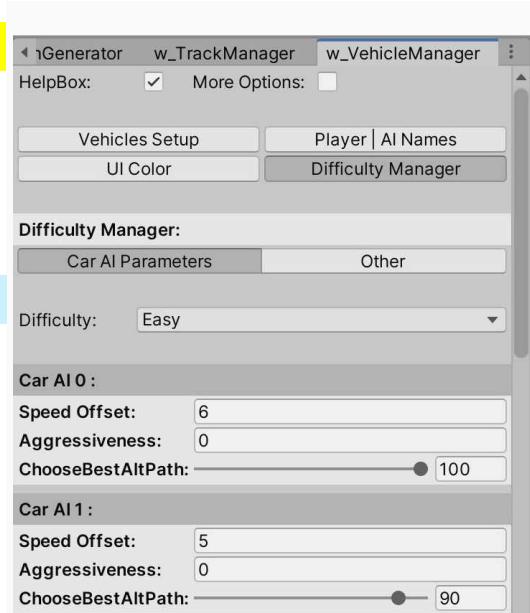


w_VehicleManager

Tools → TS → w_VehicleManager

Edit vehicle information used by game modes (Arcade, Time Attack, Championship) and the garage.

More info in section [Vehicles](#) → Page 84



Vehicles

This section describes **Vehicle** objects that can be found in the Project Tab. (Assets → Prefabs → AS → Vehicles →)

Table of contents:

Overview	Link
Short description of each vehicle module	Link
Vehicle parameters: Basic	Link
Vehicle Parameters: Advanced	Link
How to: Create a new vehicle	Link
How to: Customize a vehicle with your own 3D models	Link
Set up a vehicle to be usable in-game	Link
How to: Modify Player + AI UI color	Link
How to: Modify Player + AI names used in leaderboard	Link
How to: Modify AI behaviors	Link
How to: (Path) Change the position followed by AI vehicle (Start Line)	Link

Overview

Vehicles are instantiated in the scene when needed.

All the vehicles are variants from

00_Vehicle_Base_sc

(Assets → Prefabs → AS → Vehicles → 00_Vehicle_Base)

If you want to make a modification for all the vehicles at the same time edit:

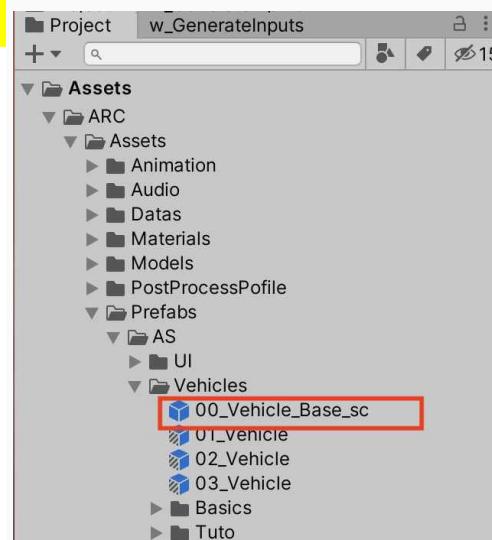
00_Vehicle_Base_sc.

(For example: the vehicle rotation speed)

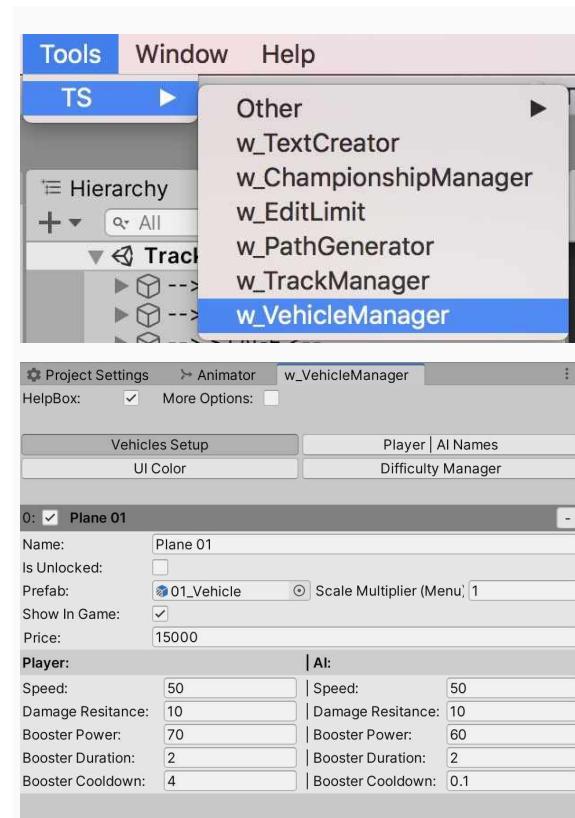
If you want to modify something in a specific vehicle.

Make the modification directly in the variant prefab.

(For example: the vehicle 3D models)



The window **w_VehicleManager**
(Tools → TS → w_VehicleManager)



allows to:

- Set up some vehicle parameters (*Vehicle Setup Tab*).
- Set up player and AI vehicle names (*Player | AI Name Tab*).
- Set up the UI colors used for each Player + AI (*UI Color Tab*).
- Set up the parameters for each AI (*Difficulty Manager Tab*).

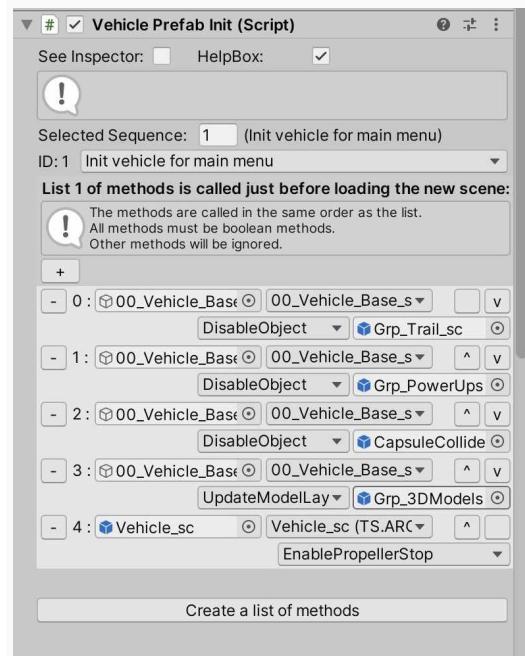
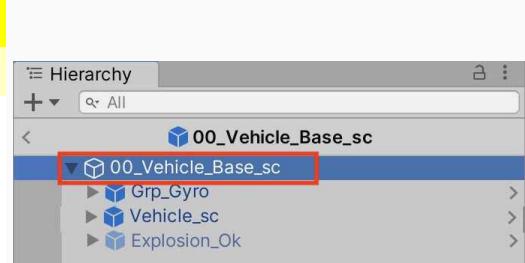
Short description of each vehicle module

00_Vehicle_Base_sc

00_Vehicle_Base_sc contains all the objects needed for the vehicle.

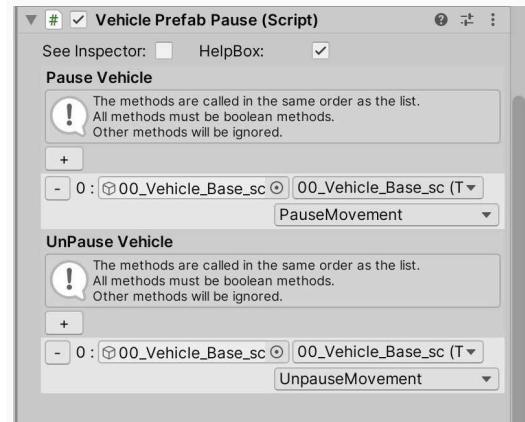
VehiclePrefabInit.cs script is attached to 00_Vehicle_Base_sc.

This script is used to initialize the vehicle according to the situation in which the vehicle is instantiated.



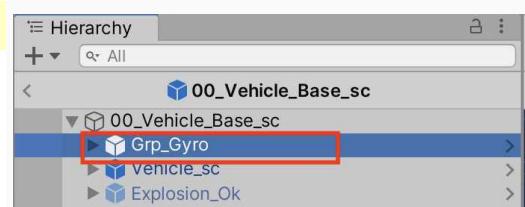
VehiclePrefabPause.cs script is attached to 00_Vehicle_Base_sc.

This script is used to pause | unpause the vehicle during game.



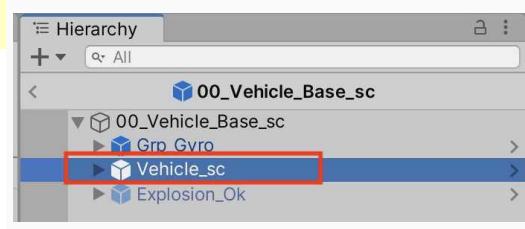
Grp_Gyro

Grp_Gyro is used to manage the vehicle direction.



Vehicle_sc

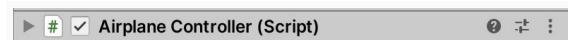
Vehicle_sc contains scripts and objects used to manage the vehicle.



AirplaneController.cs script is attached to Vehicle_sc.

This script is used to:

- Manages player + AI speed and movement.
- Manages some vehicle sounds.
- Gives access to objects that contained vehicle body, propeller stop and propeller rotation rim



CamSystem.cs script is attached to Vehicle_sc.

This script:

- Manages the position of the object followed by the camera. This object moves when inputs (left,right,up,down) are pressed by the player.
- Plays shake effect when the vehicle is destroyed.
- Contains the list CountDownRefPosCamList. This list is used to access camera positions during the countdown.
- Contains the list camPresetList. This list contains the parameters for camera view.



VehiclePathFollow.cs script is attached to Vehicle_sc.



This script:

- Allows AI to follow the path.
- Allows to know the vehicle progression in the race, the number of lap for each vehicle.

VehicleAI.cs script is attached to Vehicle_sc.



This script:

- Manages the AI direction (left,right,up,down),
- Manages AI acceleration | break.

The script works in association with

AirplaneController.cs .

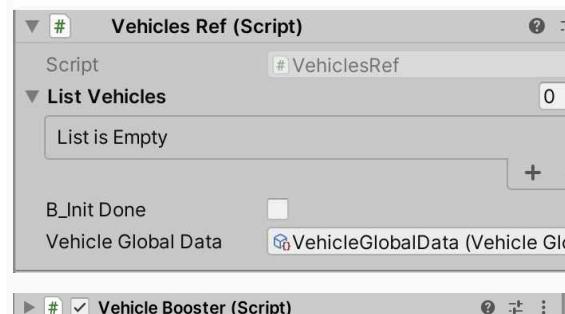
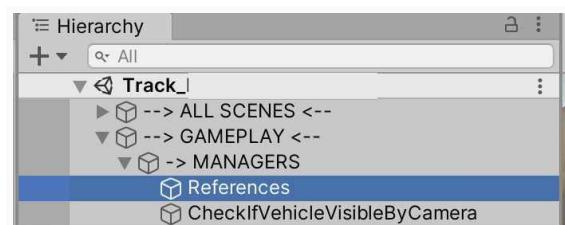
VehicleAI.cs script is attached to Vehicle_sc.



This script gives the number of the player.

This script is accessible for each vehicle in **Reference** object.

VehiclesRef.instance.listVehicles[int the vehicle]



VehicleBooster.cs script is attached to **Vehicle_sc**.

This script:

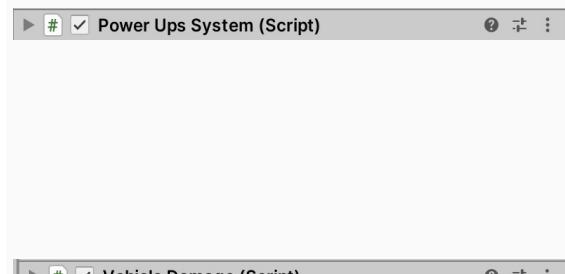
- Manages booster behavior.
- Loads information about booster from VehicleGlobalData (booster duration, cooldown,...)
- Manages the UI booster gauge.
- Allows to use custom methods when the booster is On or Off.

PowerUpsSystem.cs script is attached to **Vehicle_sc**.

This script:

- Manages Power-ups for each vehicle.
- Allows to modify each Power-up type.

VehicleDamage attached to **Vehicle_sc**.



This script:

- Manages the collisions with stage.
- Manages vehicle life.

The script contains 5 delegates:

It is possible to use those delegate to do something when the vehicle is destroyed.

VehicleExplosionAction

Vehicle is destroyed. Life = 0

VehicleRespawnPart1

Transition between the position where vehicle is destroyed and the position where the vehicle is respawned

VehicleRespawnPart2

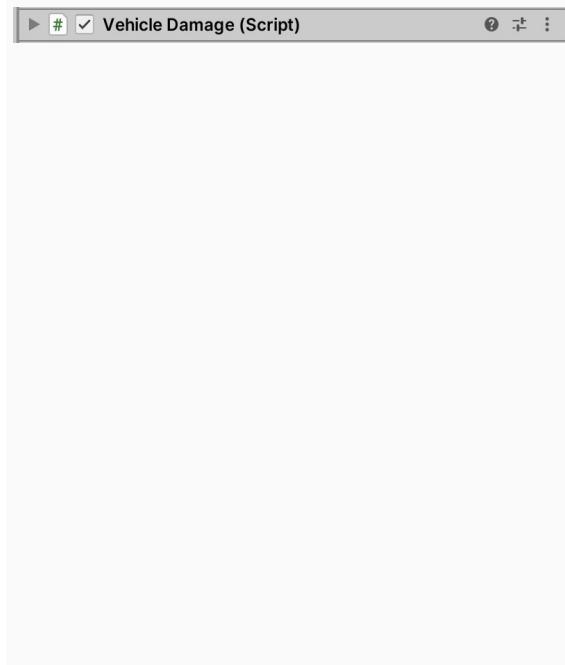
Actions when the vehicle is respawned (init)

VehicleLoseLife

Actions when the player lose life

VehicleWinLife

Actions when the player win life



VehicleInputs attached to **Vehicle_sc**.

This script manages vehicle inputs.

LayerSelector attached to **Vehicle_sc**.

This script allows to choose which layer is used for the vehicle.



RaycastDetectLimitZone attached to **Vehicle_sc**.

This script:

- Detects the stage limits by detecting objects with LimitZone layer.
- Manages the UI feedback to indicate the player is closed to the stage limits.



VehicleInitAudio attached to **Vehicle_sc**.

This script:

- Initializes the vehicle audio volumes.

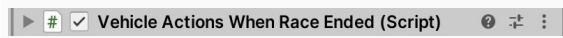
The initialization is called when the vehicles are allowed to move in the scene (method called from the SceneStepsManager).



VehicleActionsWhenRaceEnded attached to **Vehicle_sc**.

This script do actions when the vehicle finishes the race.

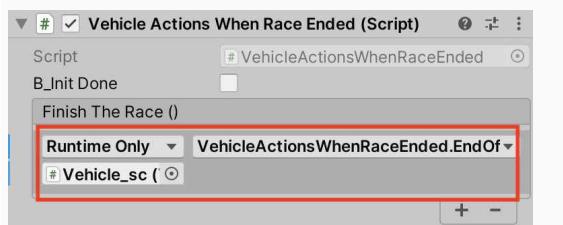
It is possible to add your methods in the UnityEvent in the Inspector.



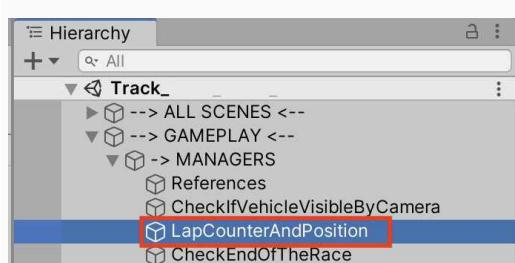
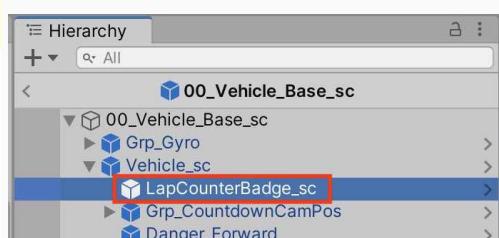
LapCounterBadge_sc

This object:

- Detects the start line.



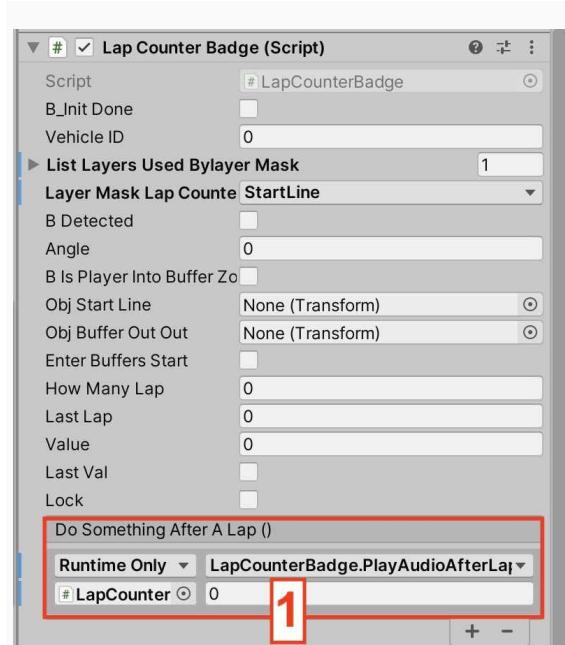
- Calls the method contained in **LapCounterAddPosition** object which counts the laps for each vehicle.



-The script **LapCounterBadge.cs** attached to **LapCounterBadge_sc** allows to call methods when a lap is finished.

By default the a sound is played after a lap (spot 1).

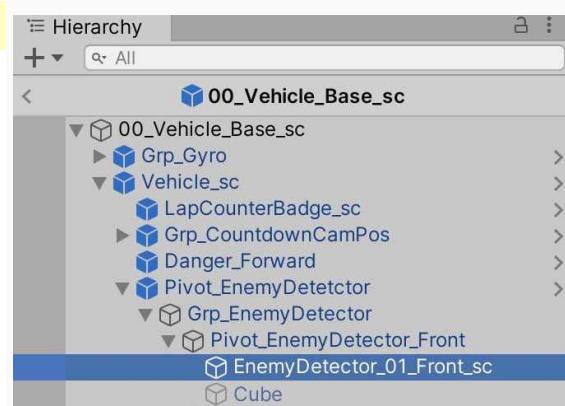
You can add your own methods in this UnityEvent **DoSomethingAfterALap()** (spot 1)



EnemyDetector_01_Front_sc

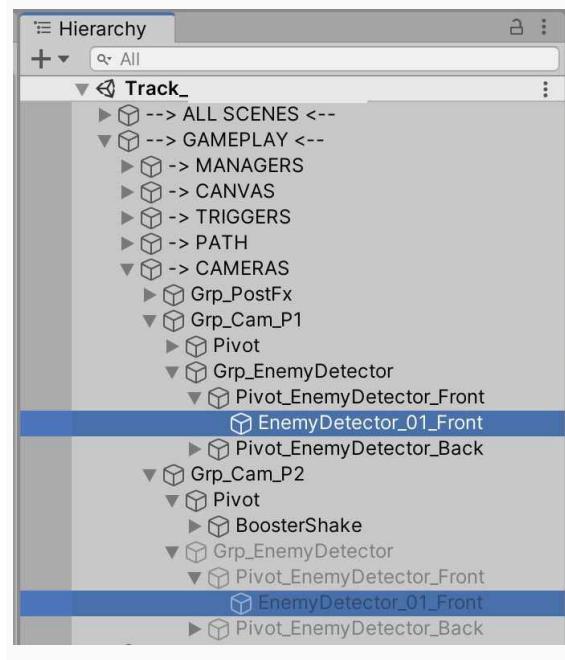
(*Vehicle_sc* → *Pivot_EnemyDetector* → *Grp_EnemyDetector* → *Pivot_EnemyDetector_Front* → *EnemyDetector_01_Front_sc*)

PowerUpsAIDetectVehicle.cs script attached to this object creates a collider area used to detect if an enemy is in front of the vehicle.



For Player 1 and Player 2 the system uses **EnemyDetector_01_Front** which is in their camera.

(*Hierarchy: GAMEPLAY* → *CAMERAS* → *Grp_Cam_P1* → *Grp_EnemyDetector* → *Pivot_EnemyDetector_Front* → *EnemyDetector_01_Front*)



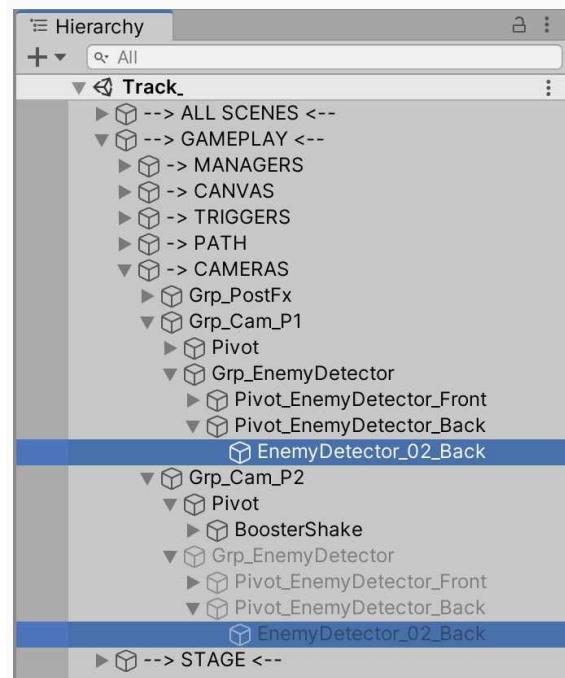
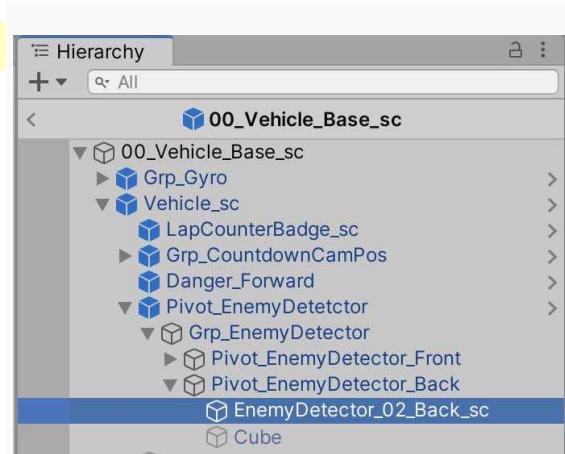
EnemyDetector_02_Back_sc

(Vehicle_sc → Pivot_EnemyDetector → Grp_EnemyDetector → Pivot_EnemyDetector_Back → EnemyDetector_02_Back_sc)

PowerUpsAIDetectVehicle.cs script attached to this object creates a collider area used to detect if an enemy is in behind the vehicle.

For Player 1 and Player 2 the system uses EnemyDetector_02_Front which is in their camera.

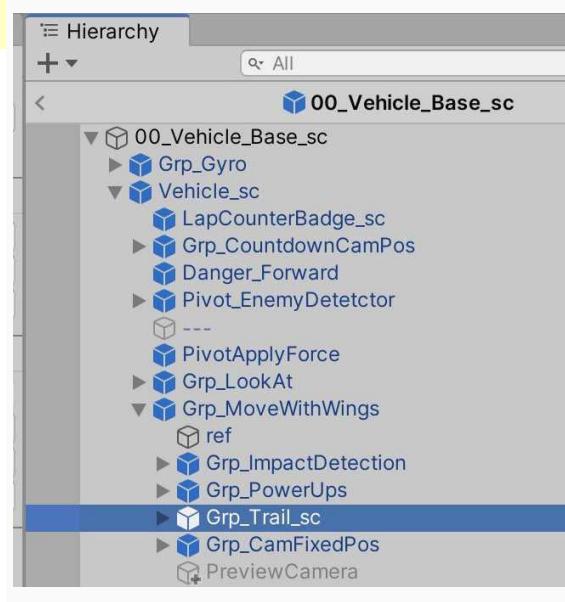
(Hierarchy: GAMEPLAY → CAMERAS → Grp_Cam_P1 → Grp_EnemyDetector → Pivot_EnemyDetector_Back → EnemyDetector_02_Back)



Grp_Trail_sc

(Vehicle_sc → Grp_MoveWithWings → Grp_Trail_sc)

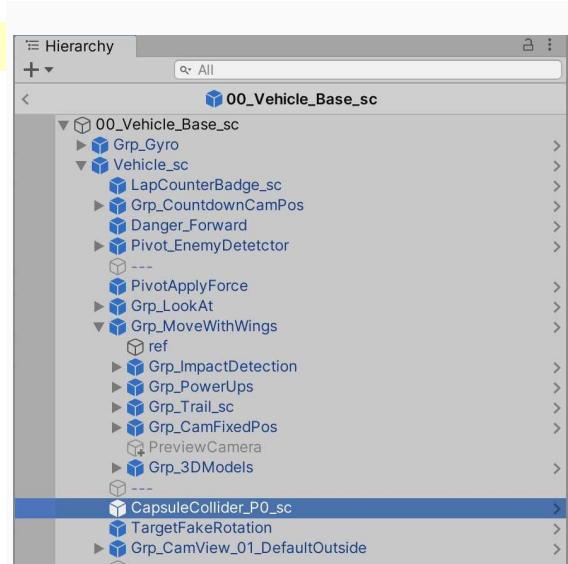
VehicleTrail.cs script attached to this object manage trails behavior when the vehicle is destroyed.



CapsuleCollider_P0_sc

(Vehicle_sc → CapsuleCollider_P0_sc)

CapsuleCollider_P0_sc object is used as the collider used by the vehicle Rigidbody.



Vehicle parameters: Basic

Basic parameters are vehicle parameters set up in **w_VehicleManager** window.
(Tools → TS → w_VehicleManager)

These parameters are used to initialize a vehicle when the vehicle is instantiated in the scene.

-Open **w_VehicleManager** window
(Tools → TS → w_VehicleManager)

-Select tab **Vehicle Setup** (spot 1).

Checkbox (spot 1) allows to display the vehicle parameters.

Name represents the name of the vehicle. This name is used when the player choose a vehicle.

If **Is Unlocked** parameter is checked the vehicle is unlocked the first time the game is launched.

From the Project Tab, drag and drop the prefab corresponding to the vehicle.

If **Show In Game** parameter is checked the vehicle is shown when the player choose a vehicle.

Cost represents the price to buy the vehicle.

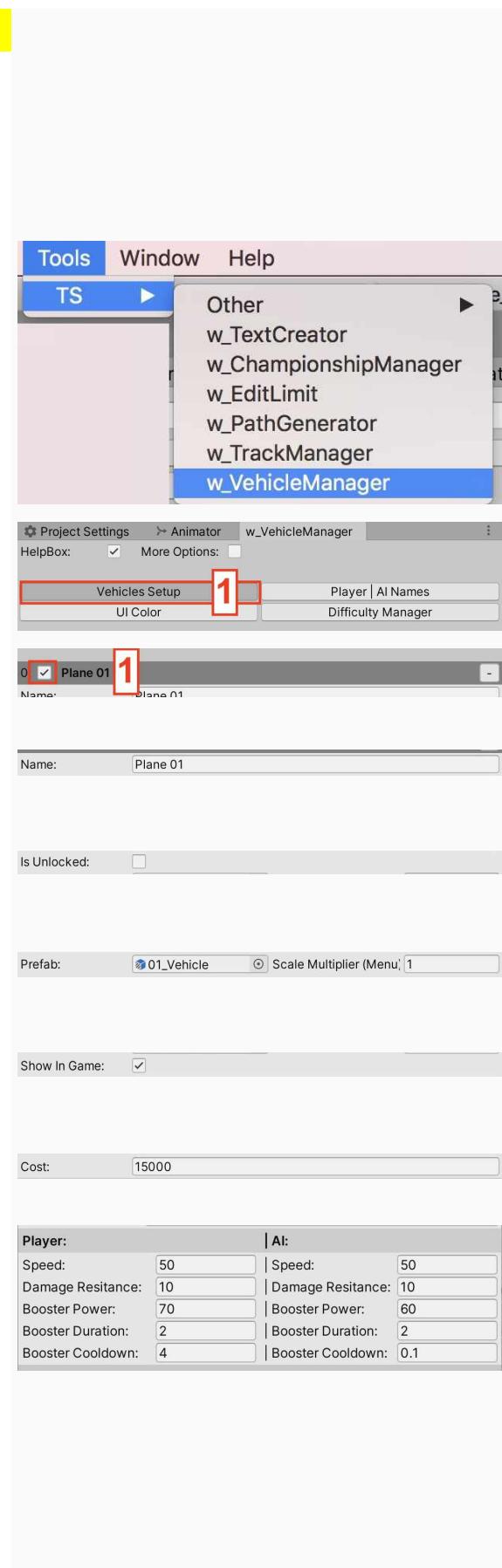
Speed: The vehicle speed when the vehicle accelerate.

Damage Resistance: The vehicle life points.
By default, Missile (-2 points), machine gun (-1 point).

Booster Power: The speed added to the vehicle.

Booster Duration: The duration of the booster.

Booster Cooldown: The duration to reload the booster.

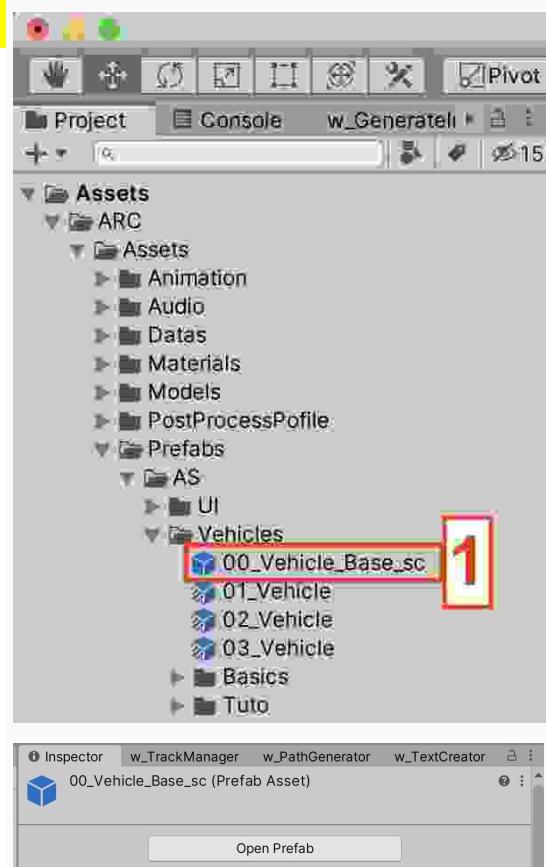


Vehicle Parameters: Advanced

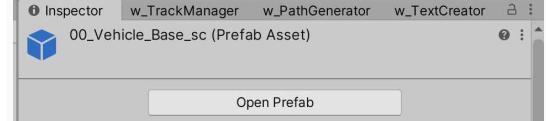
Vehicle Advanced parameters are accessible in the vehicle prefab.

If you want to make the modification for all the prefab at the same time select prefab **00_Vehicle_Base_sc** (spot 1) (*Project Tab: Assets → Prefabs → AS → Vehicles → 00_Vehicle_Base_sc*)

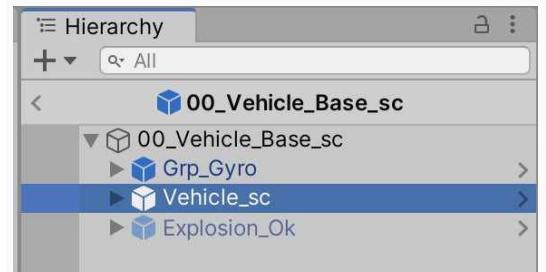
If you want to modify only one vehicle select this vehicle in the Project tab.



-In the Inspector press **Open Prefab**.



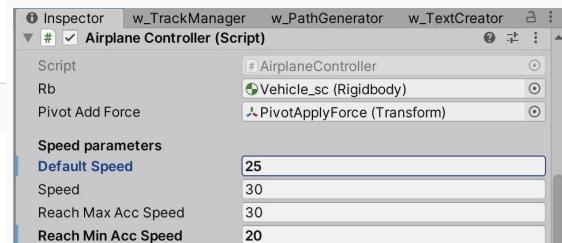
-In the Hierarchy select **Vehicle_sc**.



In the Inspector:

Info:

Vehicle speed is the addition of:
Default speed + Speed + Booster speed



Default Speed: The vehicle speed when acceleration button is not pressed.

Speed: The vehicle acceleration. Managed in **w_VehicleManager** window (*Tools → TS → w_VehicleManager*)

Reach Max Acc Speed: How quick the vehicle reach the max acceleration speed.

Reach Min Acc Speed: How quick the vehicle go back to the default speed.

This section allows to modify vehicle rotation speed:

Left | Right rotation is the addition of:

GyroYRotSpeed
+ GyroYExtraRot
+ GyroYExtraBreakRot

Up | Down rotation is the addition of:

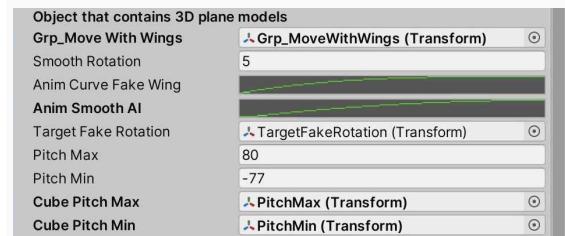
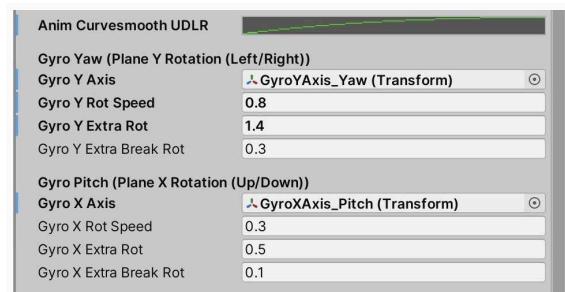
GyroXRotSpeed
+ GyroXExtraRot
+ GyroXExtraBreakRot

It allows to increase rotation when the player keeps the button pressed and when the player presses the button break.

The higher the value, the faster the vehicle turns.

To limit the vehicle pitch change:
Pitch Max and Pitch Min.

This section allows to access the AudioSource used to display vehicle sound fx.



How to: Create a new vehicle

-In Project tab select **00_Vehicle_Base_sc**
(Project tab: Assets → Prefab → AS → Vehicles → 00_Vehicle_Base_sc)

-Right click on the object.

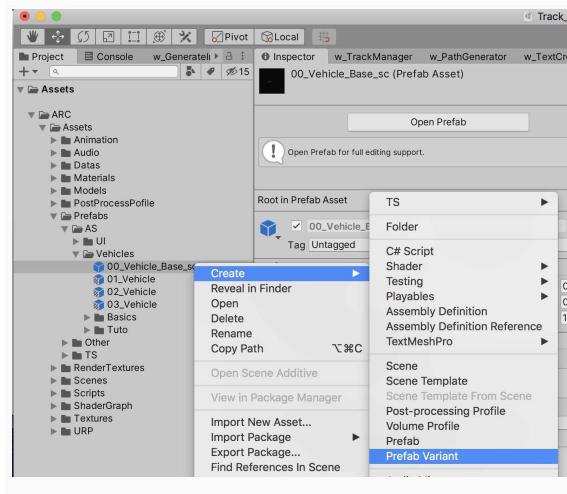
-In the menu select **Create → Prefab Variant**

A new Prefab is created.

-Rename it.

Now the new vehicle can be used as a vehicle in the game.

Note: A prefab variant is a prefab that inherits from another prefab.



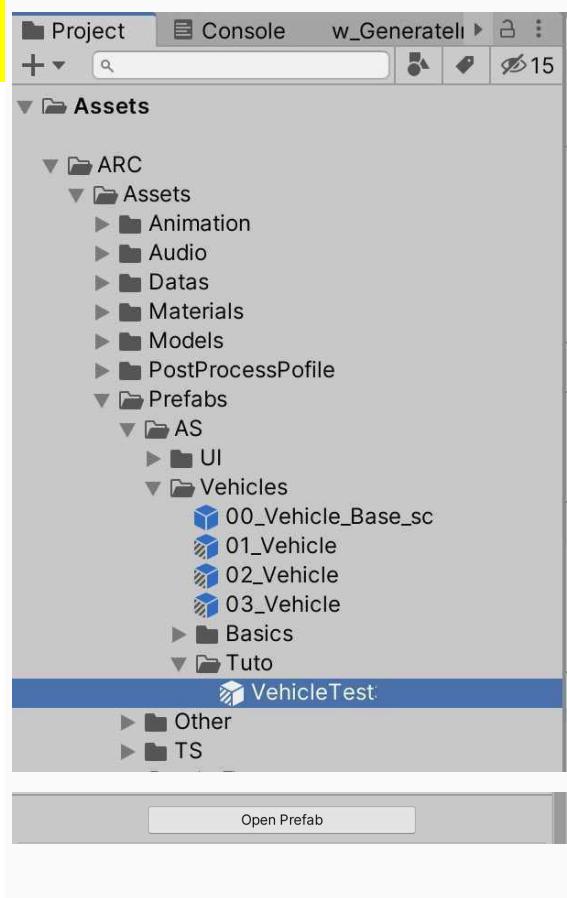
How to: Customize a vehicle with your own 3D models

-In Project tab select the vehicle you want to modify.

-As an example select **VehicleTest**

(Project tab: Assets → Prefab → AS → Vehicles → Tuto → VehicleTest)

-In the Inspector press **Open Prefab** button.



-In the Hierarchy open object **Grp_3DModels**

(Hierarchy → VehicleTest3DModel → Vehicle_sc → Grp_MoveWithWings → Grp_3DModels)

-Disable **Ref** objects in

Body → **Ref**

Propeller_Stop → **Ref**

Propeller_RotSim → **Ref**

-In **Body** object: Add your body 3D models.

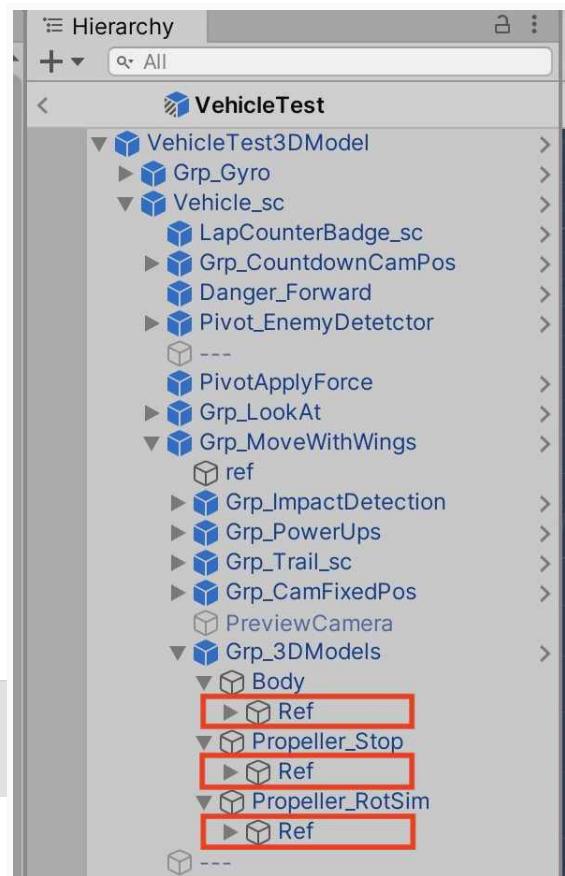
-In **Propeller_Stop** object: Add your propeller 3D models uses when the propeller is stopped.

-In **Propeller_RotSim** object: Add your propeller 3D models uses when the propeller is rotating.

VERY IMPORTANT: Your 3D models mustn't contain colliders. Remove all the colliders included in your 3D models.

-In the Hierarchy press < to exit the prefab edition.

Now your vehicle is ready to be used in the game.



Set up a vehicle to be usable in-game

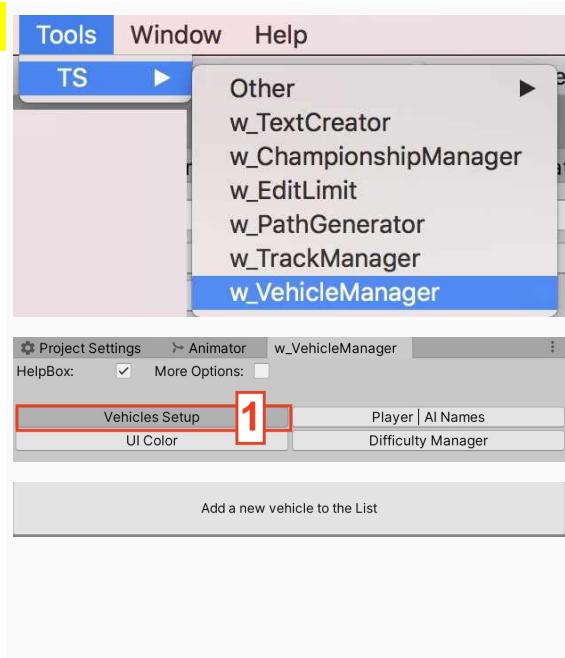
-Open **w_VehicleManager** window.

(Tools -> TS -> w_VehicleManager)

A new window appears.

-Press **Vehicles Setup** button.

-Go to the end of the window and press **Add a new vehicle to the list** button.



At the end of the list a new section appears.

-**Write the vehicle name.** This name is used when the player choose a vehicle.

-**Keep Is Unlocked checked.** It allows the player to use the vehicle the first time is start the game.

-In the empty field drag and drop from the Project tab the prefab of your vehicle (spot 1).

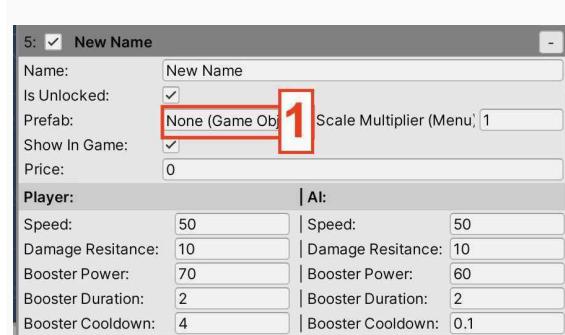
If needed you can change Scale Multiplier parameter.
Change the value if the vehicle is too small or too big when he is displayed in the garage.
By default the value = 1.

-**Keep Show In Game check.** If the box Show In Game is unchecked the vehicle is not displayed in the garage.

If needed it is possible to:

-Choose a price to unlock the vehicle in the garage.
-Customize some vehicle parameters (speed, resistance and power-ups)

Your new vehicle is available for the player.



How to: Modify Player + AI UI color

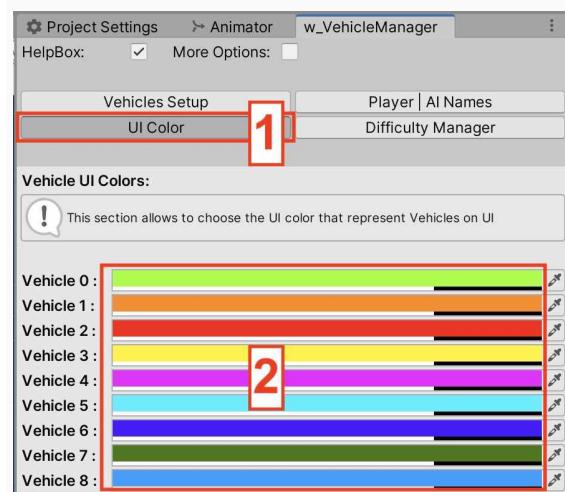
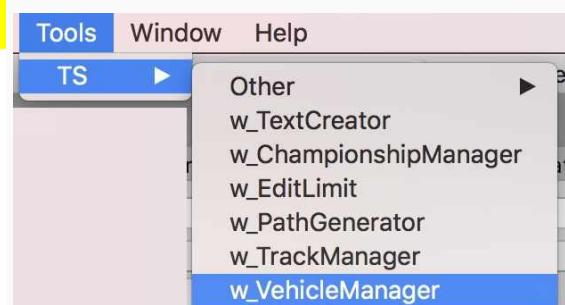
UI colors are used for:

-UI map spots that represent vehicle.
-UI triangle above each vehicle.

-Open w_VehicleManager window.
(Tools → TS → w_VehicleManager)

A new window appears.

-Press Vehicles Setup button (spot 1).
-Change the color for each vehicle (spot 2).



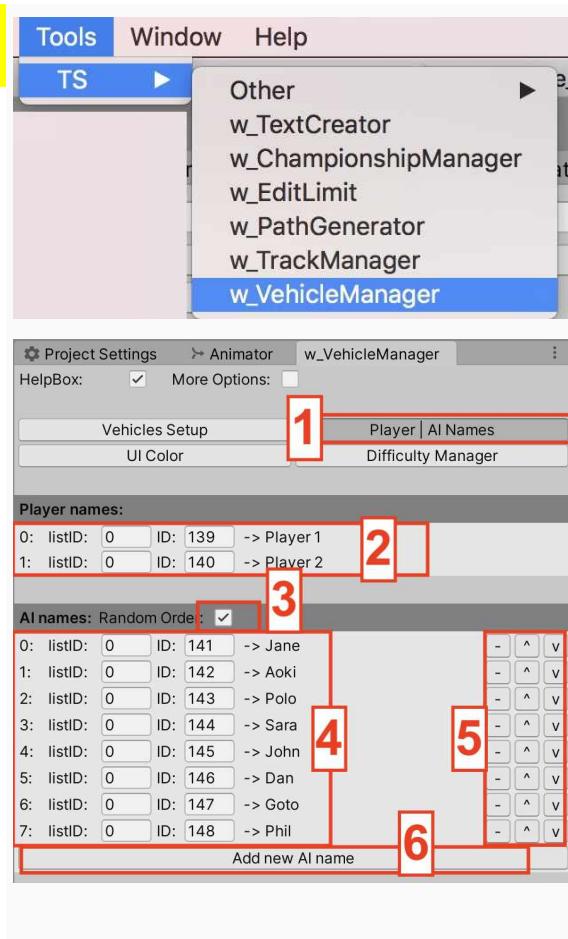
How to: Modify Player + AI names used in leaderboard

It is possible to choose the name of each player.

- Open **w_VehicleManager** window.
(Tools → TS → w_VehicleManager)

A new window appears.

- Press **Player | AI Names** button (spot 1).
- Choose player names (spot 2).
- Random Order checkbox (spot 3) allows to choose if the AI names are randomly selected when the race starts or if the 1st AI has always the 1st AI name, the 2nd AI has always the 2nd AI name ...
- Choose AI names (spot 4).
- Change the name order with **^** and **v** button. Remove a name using **-** button (spot 5).
- Press button **Add new AI name** to add a new AI name at the end of the list (spot 6).



How to: modify AI behaviors

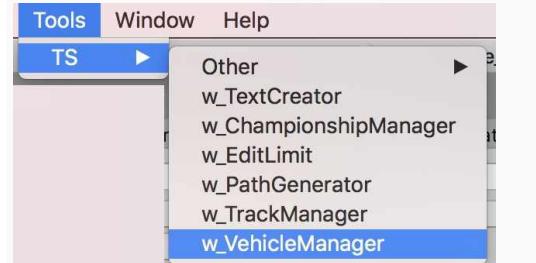
AI behavior is separated into 2 parts:

- The basics AI parameters
- The AI behavior depending the position of the AI vehicle on the grid.

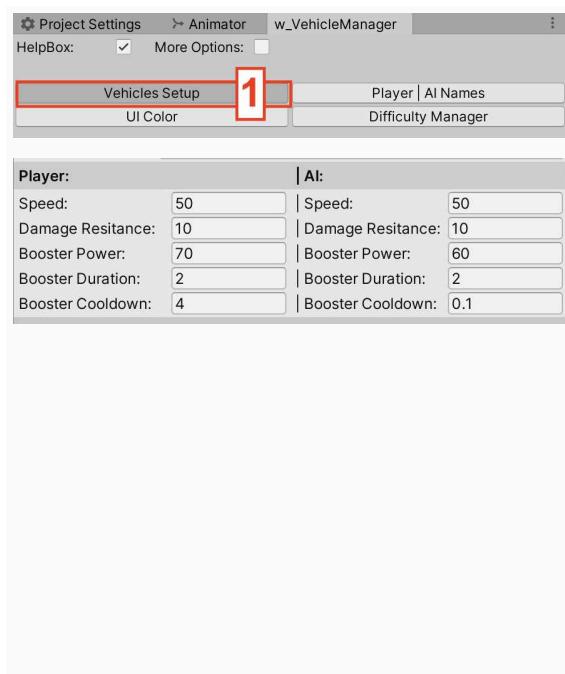
Part 1: The basic AI parameters

The basic AI parameters are vehicle parameters set up in **w_VehicleManager** window. They controls AI speed, damage and booster parameters.
(Tools → TS → w_VehicleManager)

- Open **w_VehicleManager** window
(Tools → TS → w_VehicleManager)



-Select tab **Vehicle Setup** (spot 1).



Speed: The vehicle speed when the vehicle accelerates.

Damage Resistance: The vehicle life points. By default, Missile (-2 points), machine gun (-1 point).

Booster Power: The speed added to the vehicle.

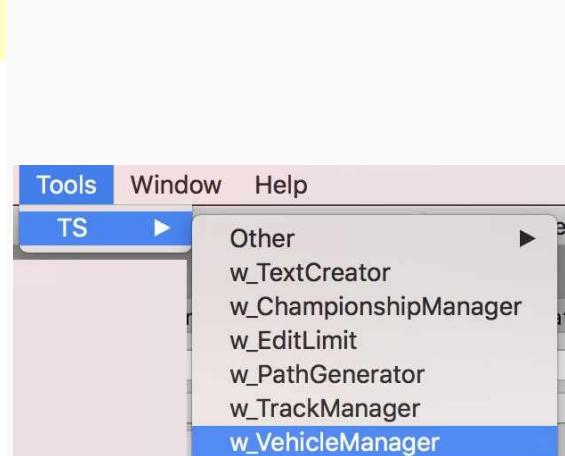
Booster Duration: The duration of the booster.

Booster Cooldown: The duration to reload the booster. *By default the AI Booster cooldown = 0.1. This value ensures that the AI will have its booster usable when the vehicle enters a Booster Trigger.*

Part 2: Behavior depending the position of the AI vehicle on the grid.

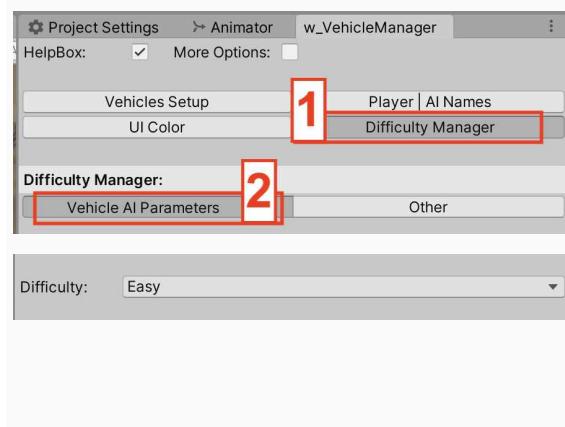
It is possible to customize AI behavior depending his position on the grid.

-Open **w_VehicleManager** window
(Tools → TS → w_VehicleManager)



-Press **Difficulty Manager** button (spot 1).

-Press **Car AI Parameters** (spot 2).



-In the drop down menu select the difficulty you want to change. *By default there are 3 difficulty: Easy, Medium and Hard.*

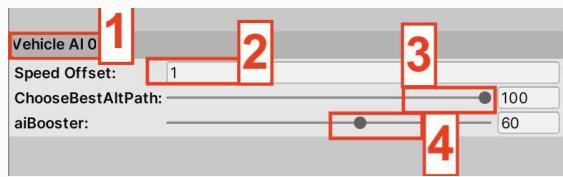
Vehicle AI + number (spot 1) represents a AI vehicle.

Number = 0 represents the 1st vehicle on the grid. Number = 1 represents the 2nd vehicle on the grid ...

Speed Offset (spot 2). This parameter allows to add or remove speed. As an example a vehicle with a speed = 50 and on offset = +1 will have total speed = 51.

ChooseBestAltPath (spot 3). Choose the percentage of change the vehicle choose the best alternative path.

AiBooster (spot 4): Choose the percentage of the boost speed used by the vehicle.



Car AI 1 :
Speed Offset: 0.5
ChooseBestAltPath: 90
aiBooster: 60
Car AI 2 :
Speed Offset: 0
ChooseBestAltPath: 80
aiBooster: 60
Car AI 3 :
Speed Offset: -0.5
ChooseBestAltPath: 70
aiBooster: 60
Car AI 4 :
Speed Offset: -1
ChooseBestAltPath: 60
aiBooster: 60

How to: (Path) Change the position followed by AI vehicle (Start Line)

AI's follow the path during the race.

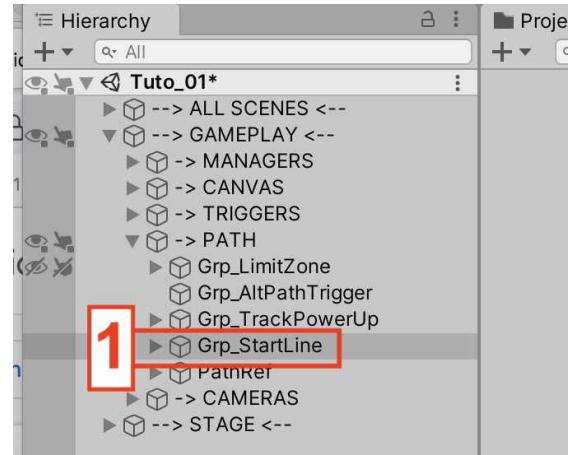
To give the feeling that vehicles do not follow exactly the same path a position offset is applied to each AI.

This offset can be modified:

In a Gameplay scene:

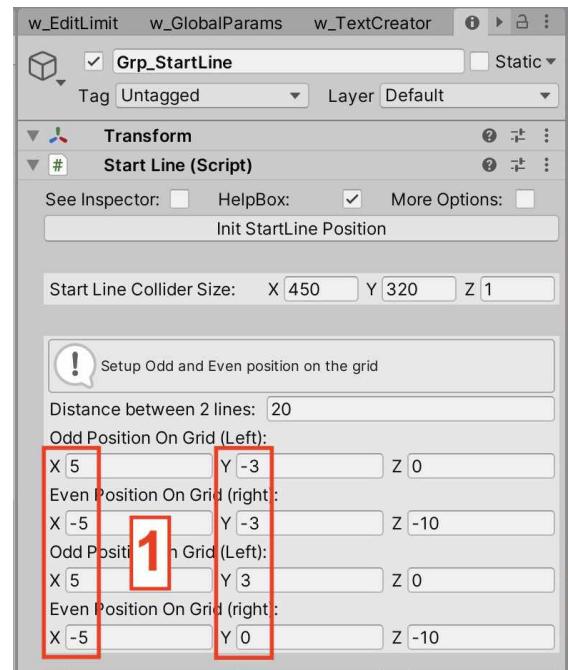
-In the Hierarchy select **Grp_StartLine** (spot 1)

(Hierarchy: GAMEPLAY → PATH → Grp_StartLine)



In the Inspector:

Change X et Y position to give an offset to AI vehicles (spot 1).



Game Mode

This section gives information about how to managed Championship, Arcade and Time Trial modes

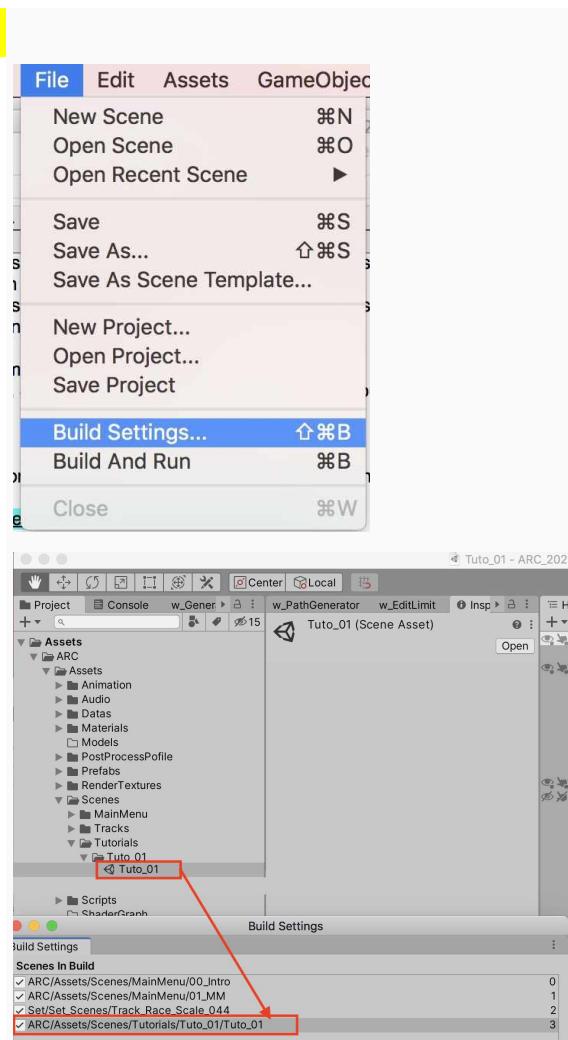
Table of contents:

Track Manager: Set up new track.	Link
Track Manager: Set up Time Trial leadebroard	Link
Track Manager: Set up coins earned after a race (Arcade, Time Trial)	Link
Championship: Create a Championship	Link
Championship: Delete a championship	Link
Arcade: Choose the number of vehicles in a race	Link
Test Mode (3 and 5 mode)	Link
Choose the number of lap for a track	Link

Track Manager: Set up new track

This section explains how to set up a Track in the Track Manager. The Track Manager allows to set up the track used during the game.

-Go to File → Build Settings

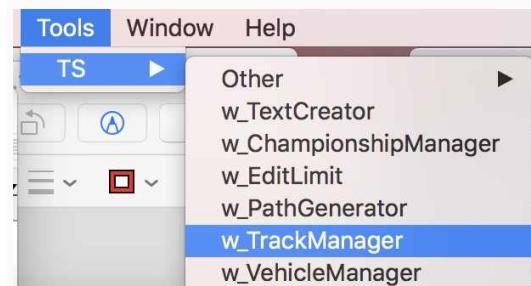


-Drag and drop the scene that contain the track you want to use at the end of the list in the **Scenes In Build Box**.

Unity is now able to load the scene during the game.

-Go to Tools → TS → w_trackManager

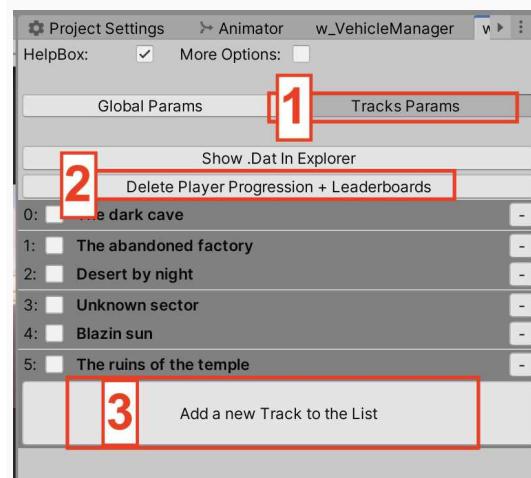
A new window appears. I suggest you to attached the window to the editor.



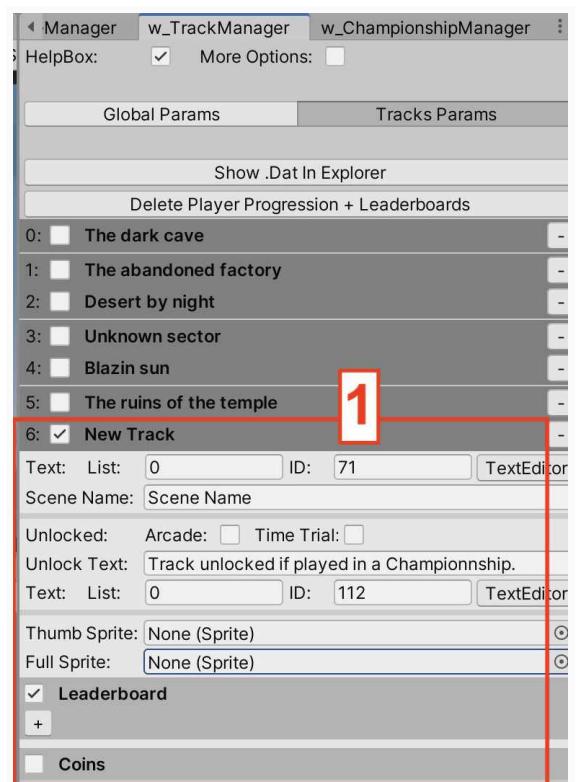
-In w_TrackManager click on tab Tracks Params (spot 1).

-Press Delete Player Progression + Leaderboards button (spot 2).

-Go to the end of the track and press Add a new Track to the List button (spot 3).



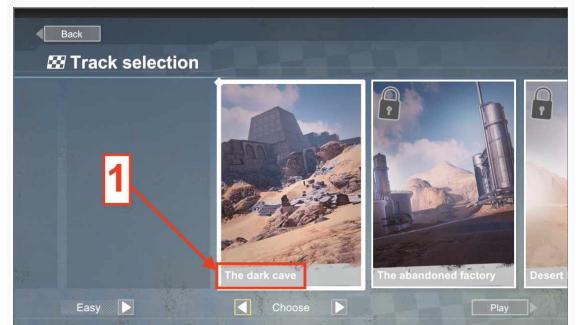
A new slot is created (spot 1).



First we are going to set up the name of the track shown by the player in the main menu.

The name of the track is displayed when the player choose a track in Arcade or Time Trial Mode (spot 1).

The asset allows to display text using a multi-language system.



For this example we are going to use the multi-language text with the ID = 157 already set up in the project. ID: 157 displays the text Track Demo.

More info in section [Localization](#)
→ Page 211

-In field **ID** write **157** (spot 1)

To load a scene during the game, the system needs to know the name of the scene saved in Project Folder.

-In field **Scene Name** write the name of the scene use for this track.



It is possible to choose if the track is unlocked in Arcade and Time Trial when the game starts the first time.

You can change the text displayed in the UI when the track is locked. In the example we do not change this text.

When the player select a track in Arcade or Time mode a thumbnail image is used to represent the track.

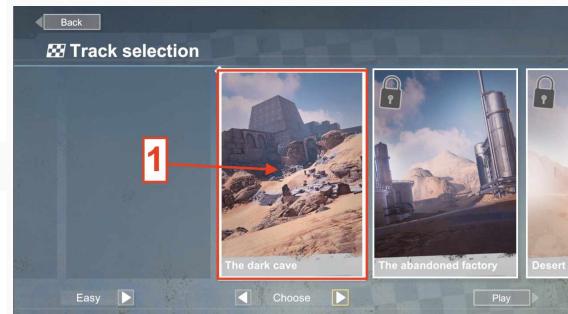
-Click on the **circle** in field **Thumb Sprite**.

Select an image in the list.

This picture is displayed when the player choose a race.

Info:

The thumbnails used by default in the asset have a size of **499 x 721**.



When the player start a race in Championship a full screen image is used to represent the track.

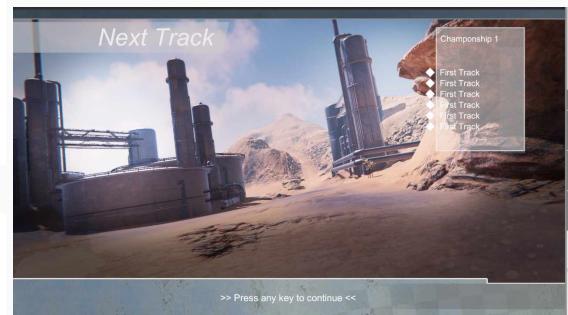
-Click on the **circle** in field **Full Sprite**.

Select an image in the list.

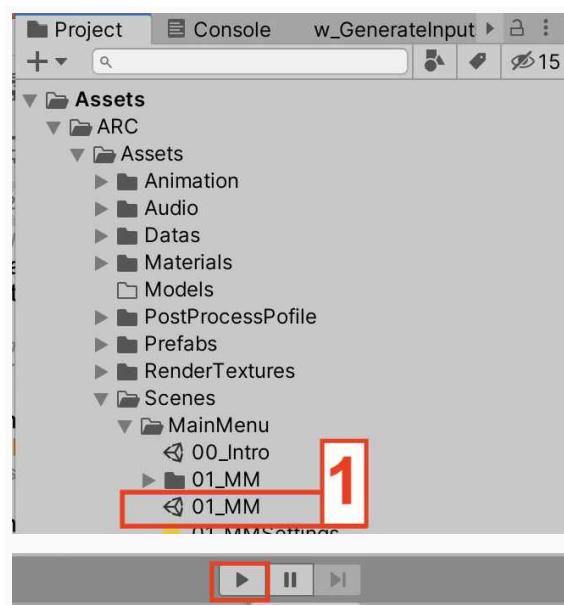
This picture is displayed when the player starts the race in Championship Mode.

Info:

The full images used by default in the asset have a size of **1920 x 945**.

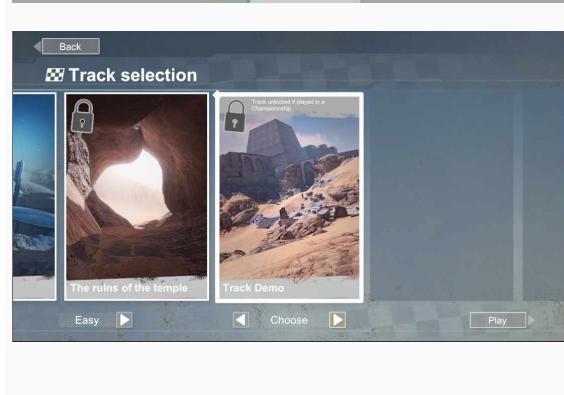


-Open the Main Menu Scene **01_MM** (spot 1).
(Project tab: Assets → Scenes → Main Menu → 01_MM).



Press **Play** button.

It is possible to choose the new track in Arcade and Time Trial Mode



Track Manager: Set up Time Trial leaderboard

This section explains how to set up a default leaderboard.
For information, leaderboard is only used in Time Trial Mode.

First we are going to remove the Player Progression Data.

-Go to Tools → TS → w_trackManager.

A new window appears. I suggest you to attached the window to the editor.

-In w_TrackManager click on tab Tracks Params (spot 1).

-Press Delete Player Progression + Leaderboards button (spot 2).

To open the track to set up:

-Check the box to the left of the track name (spot 1)

-Check the box next to Leaderboard section (spot 1)

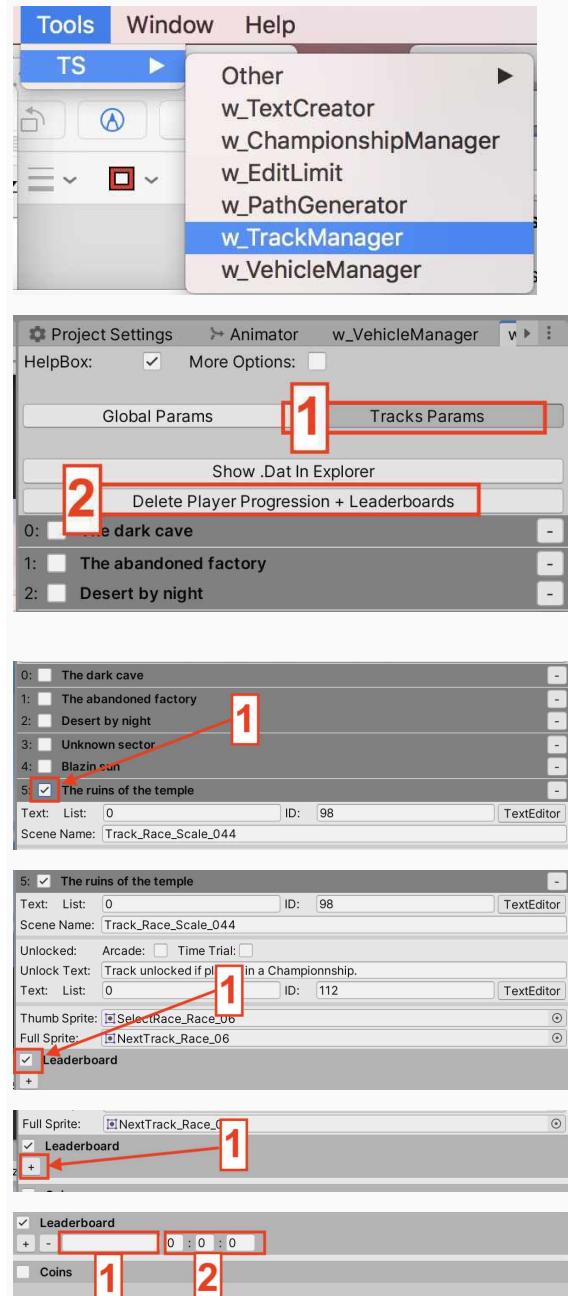
-Press + button to create a new score (spot 1)

-Write a name (spot 1) and a score (spot 2)

Repeat the process for the other scores.

Note:

Scores should be recorded from best to worst
The first score in the list is the best score.
The last score in the list is the worst score.



Track Manager: Set up coins earned after a race (Arcade, Time Trial)

This section explains how to set up the rewards earned by the player after an Arcade or a Time Trial race.

Depending his position at the end of the race the player earned coins.

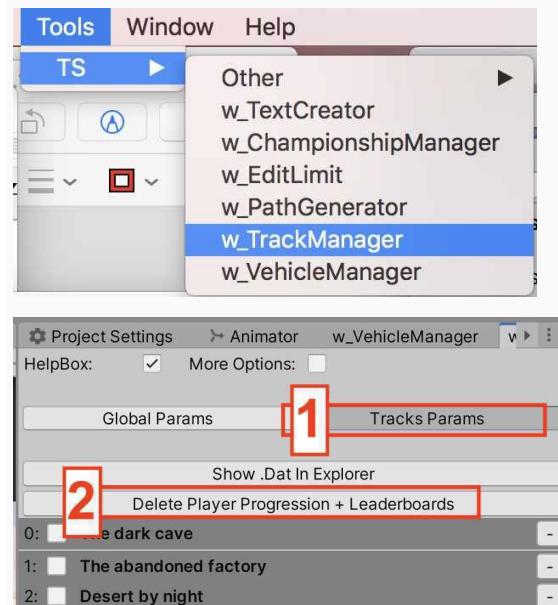
First we are going to remove the Player Progression Data.

-Go to Tools → TS → w_trackManager

A new window appears. I suggest you to attached the window to the editor.

-In w_TrackManager click on tab Tracks Params (spot 1).

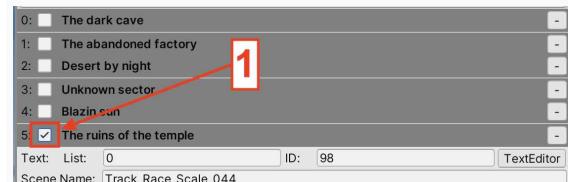
-Press Delete Player Progression + Leaderboards button (spot 2).



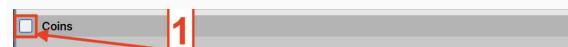
First possibility to create coins values

To open the track to set up:

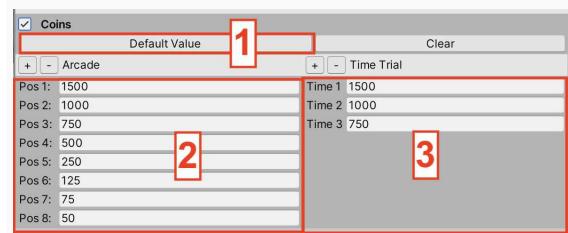
-Check the box to the left of the track name (spot 1)



-Check the box next to Coins section (spot 1)



-Press Default value button (spot 1)
to automatically create a list of coins earned by the player at the end of the race (spot 2 and 3)



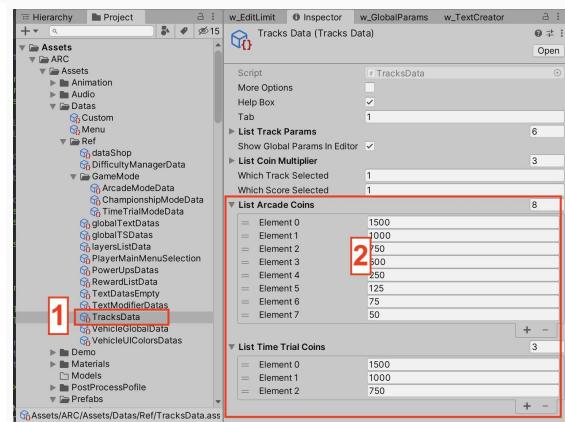
Note:

It is possible to change the default value.

In Project tab select **TracksData** (spot 1).

(Project tab: Assets → Datas → Ref → TrackData)

In the Inspector change the values inside
List Arcade Coins and **List Time Trial Coins** (spot 2).



Second possibility to create coins values

-Press + button next to Arcade create a new coin value (spot 1)

-Choose the value (spot 2)

It is the same process for Time Trial (spot 1 and 2)



Championship: Create a Championship

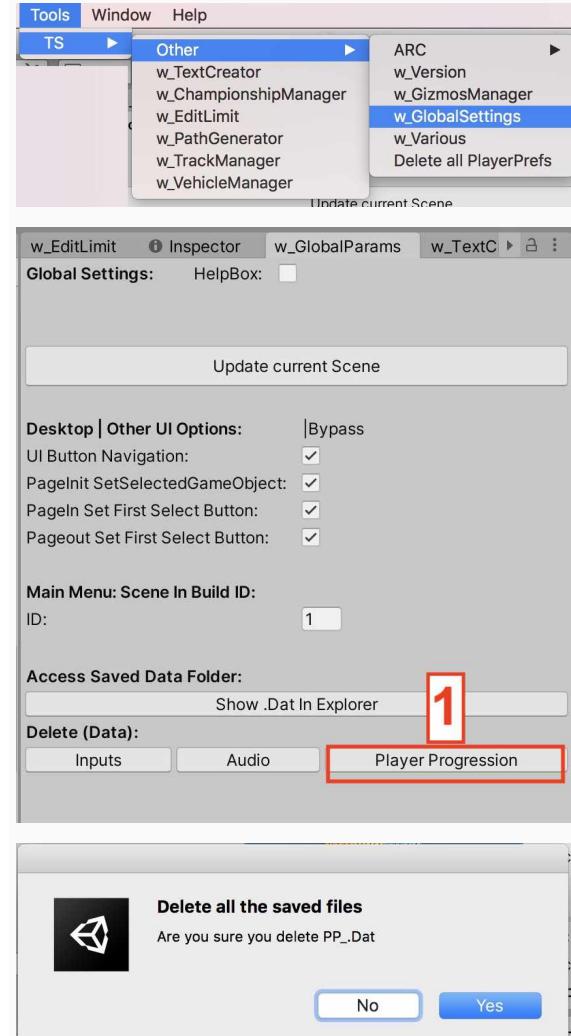
This section explains how to set up a new championship.

First we are going to remove the Player Progression Data.

-Go to Tools → TS → w_GlobalSettings

A new window appears.

-Press Player Progression button (spot 1) to delete the player progression data.



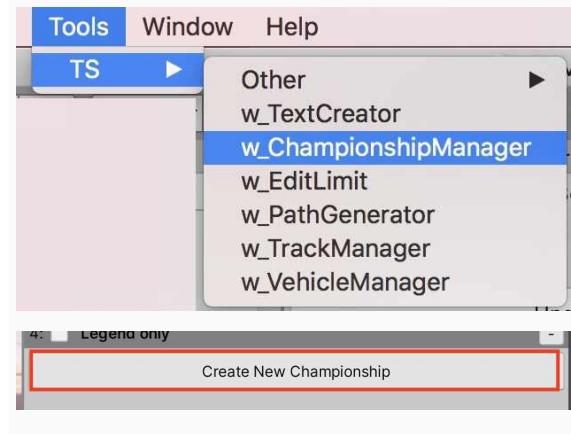
-Press Yes.

Now we are going to create the new Championship.

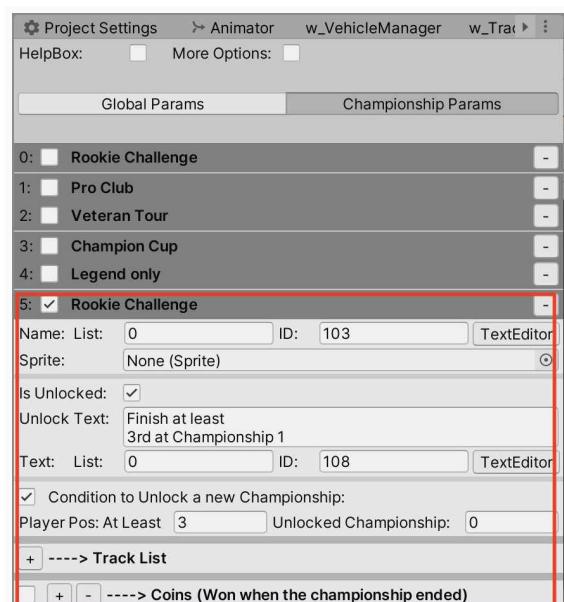
-Go to Tools → TS → w_ChampionshipManager

A new window appears. I suggest you to attached the window to the editor.

-At the end of the window press Create New Championship button

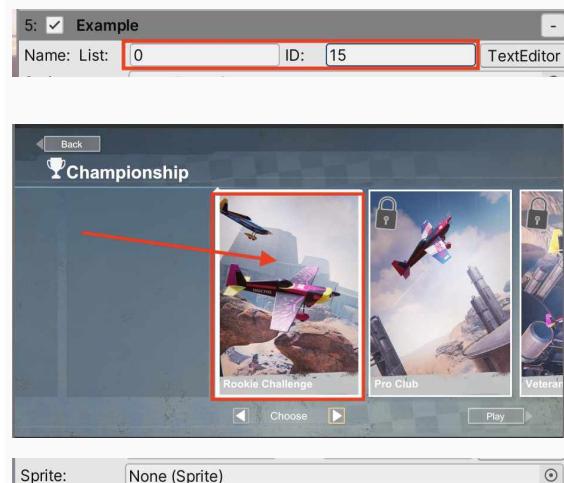


A new slot is created at the end of the list.



-Select a multi-language text ID for the championship name.

-Select an image to represent the championship in the Main Menu section select your championship.



Info:

Make images with a size of 499 x 721.

-Choose if the championship is unlocked the first time the player launches the game.

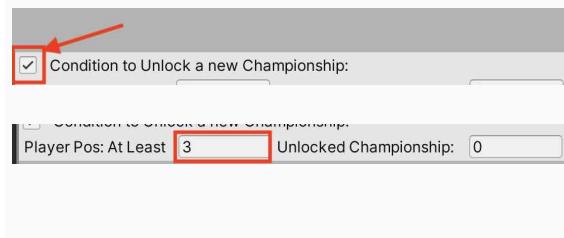
-Select a multi-language text ID that give information about how to unlock the championship.



When the championship is finished it is possible to choose the condition to unlock a new championship

-Check the box
Condition to Unlock a new Championship

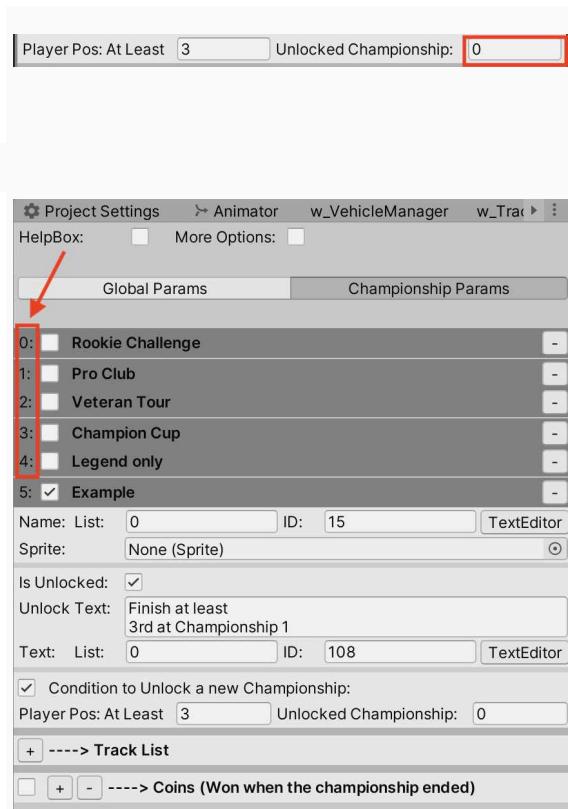
-Choose the minimum player position at the end of the championship to unlock the next championship.



-Choose the championship to unlock if the player complete the condition to unlock the championship.

Info:

You will find the championship **ID** on the left of championship name.



The next section allows to set up the track used in the Championship.

It is only possible to use track already set up in the w_TrackManager window.
(Tools → TS → w_TrackManager)

-Press **+** button next to Track List text

A new track is added to the list.

-Press **checkbox** (spot 1)
to show all the track parameters (spot 2).

-Write the track **ID** of the track you want to use.

You find the Track ID in w_TrackManager window.
(Tools → TS → w_TrackManager)

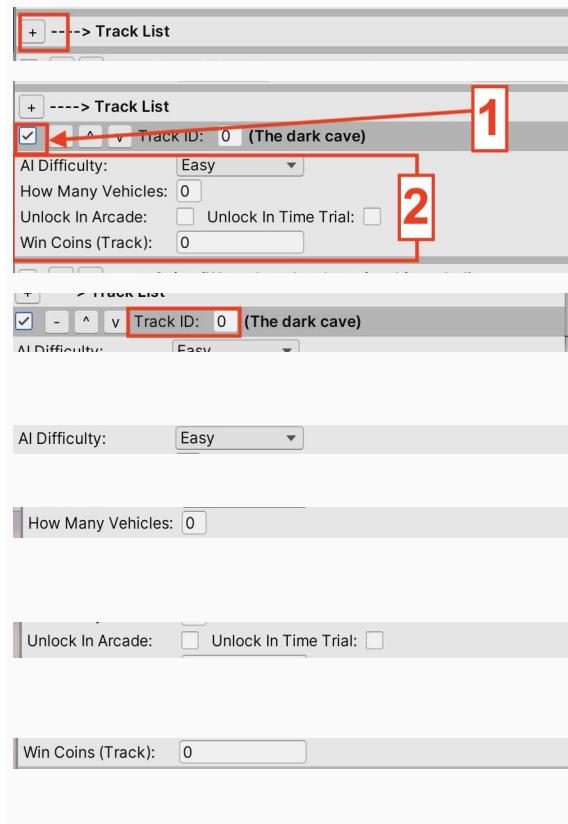
-Choose AI difficulty for this track.

-Choose the number of vehicle in this race.

-Write 8 to have 8 vehicles in the race (Players + Al's)

-Choose if, after playing the track,
the track is unlocked in Arcade and Time Trial Mode.

-Choose the number of coins eared by the player at
the end of the track.



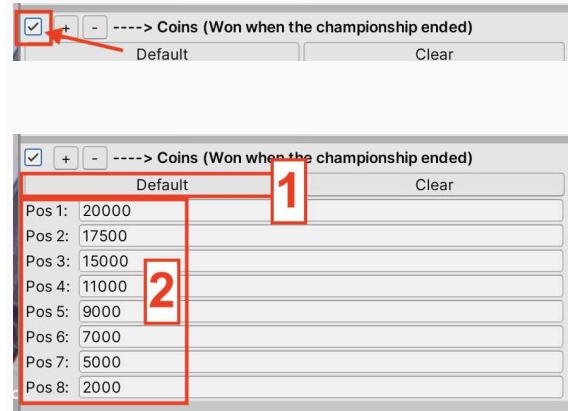
Next section allows to choose the coins earned by the player at the end of the championship depending his final position on the championship.

First possibility to create coins values

To open the track to set up:

-Check the box to the left of coins section

-Press **Default value** button (spot 1)
to automatically create a list of coins earned by the player at the end of the race (spot 2 and 3)

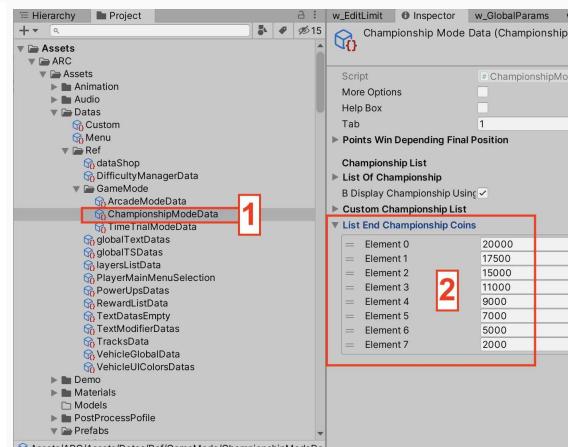


Note:

It is possible to change the default value.

In Project tab select **ChampionshipModeData** (spot 1)
(Project Tab: Assets → Datas → ChampionshipModeData → TrackData)

In the Inspector change the values inside **List End Championship Coins** (spot 2)



Second possiblty to create coins values

-Press **+** buton next to Arcade create a new coin value (spot 1)

-Choose the value (spot 2)

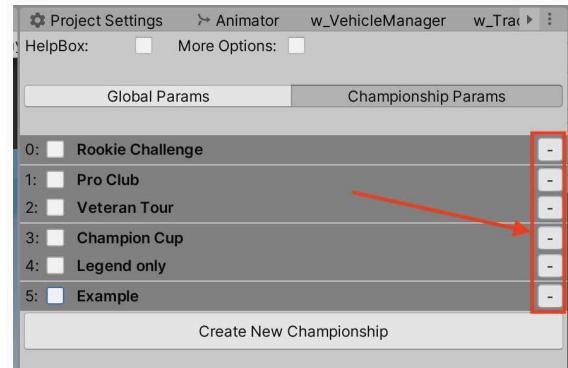
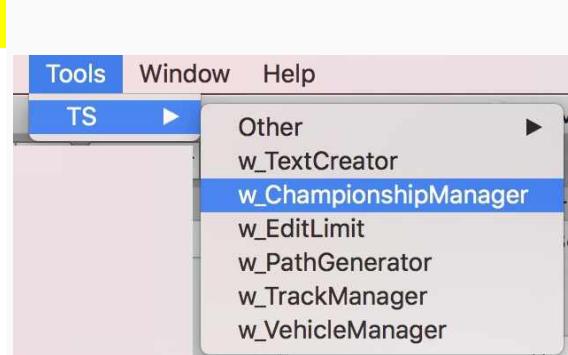


Championship: Delete a championship

-Go to Tools → TS → w_ChampionshipManager

A new window appears. I suggest you to attached the window to the editor.

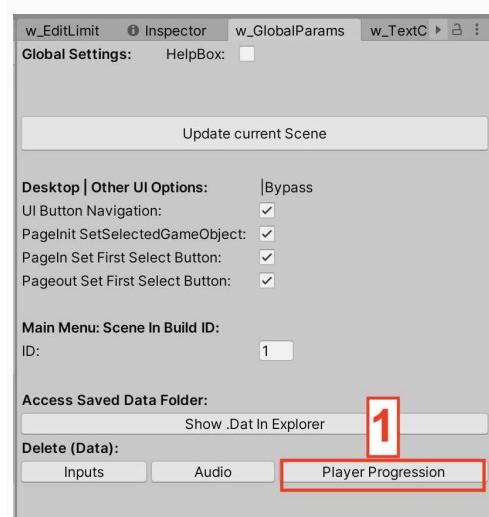
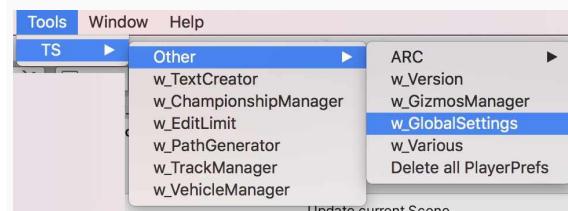
-Press **-** button on the right of the championship you want to delete.



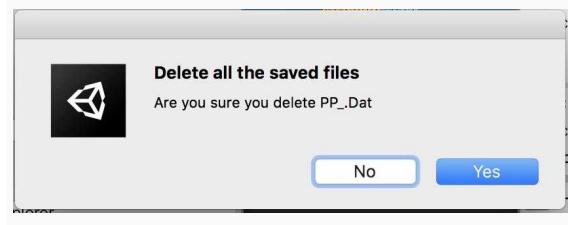
-Go to Tools → TS → w_GlobalSettings

A new window appears.

-Press **Player Progression** button (spot 1) to delete the player progression data.



-Press Yes.



Arcade: Choose the number of vehicles in a race

It is possible to choose the number of vehicles in a race.

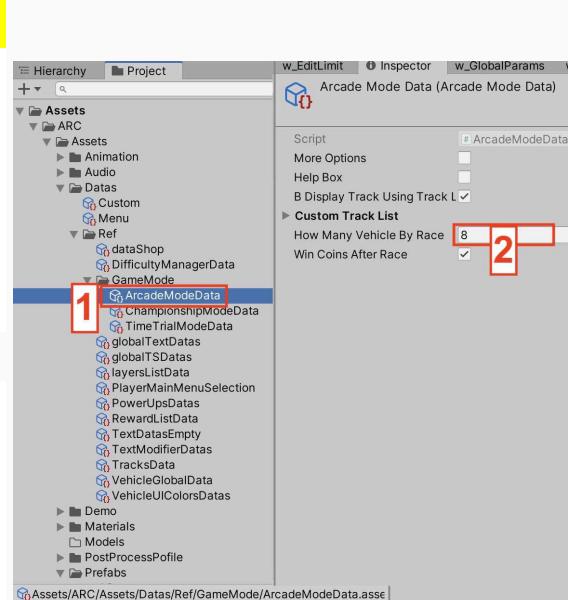
It is the same number of vehicles for all the race.

By default there are 8 vehicles by race in Arcade Mode.

To change the number of vehicles in Arcade mode:

In Project tab select **ArcadeModeData** (spot 1).
(Assets → Datas → Ref → GameMode → ArcadeModeData)

In the Inspector, change the parameter
How Many Vehicles By Race (spot 2).



Test Mode (3 and 5 mode)

To test a track you can use the Game Mode 3 or 5.

Mode 3 allows you to play with AIs and allows to choose the number of players and the AI difficulty.

Mode 5 removes all the collisions with the vehicle and the stage environment. There is only the Player 1 vehicle.

To select a Game Mode:

-Go to **Tools → TS → w_TrackManager**.

-Press **Global Params** button (spot 1).

For Game Mode 3:

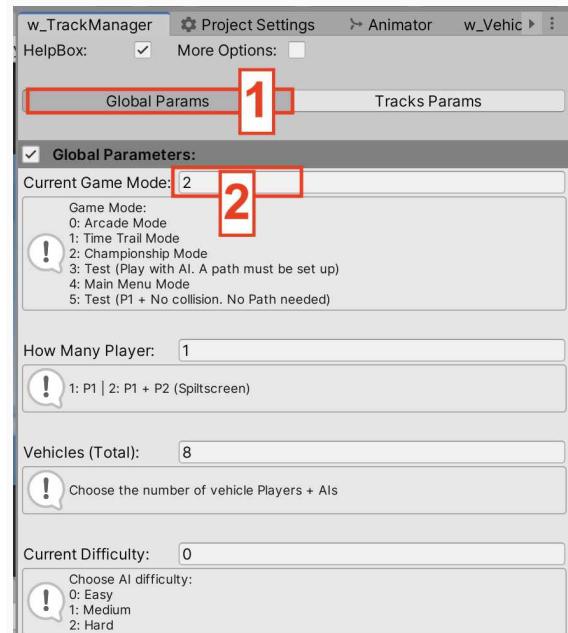
-Write **3** in **Current Game Mode** (spot 2)

You can also choose:

- The number of Real Player
- The number of AI
- The AI difficulty

For Game Mode 5:

-Write **5** in **Current Game Mode** (spot 2)



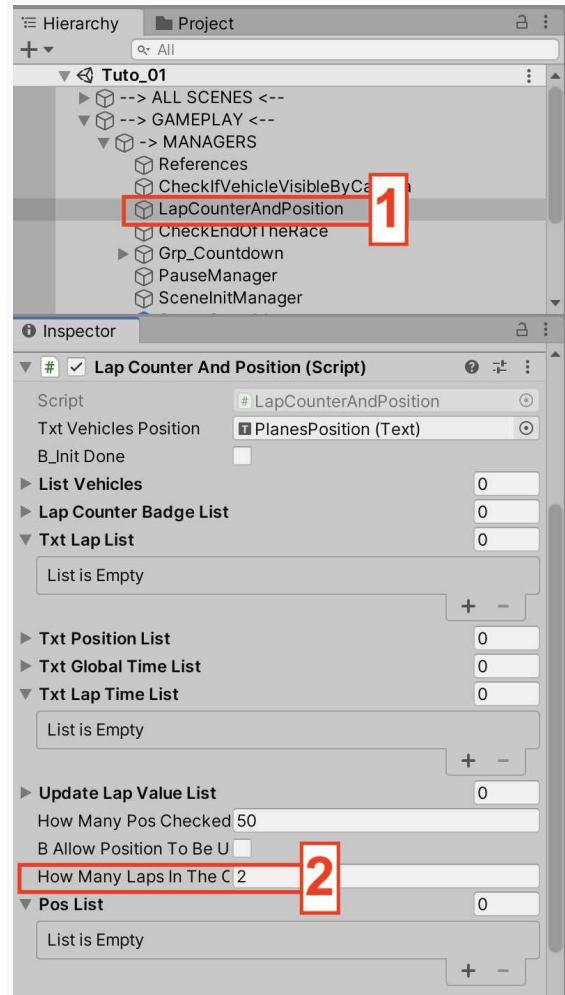
Choose the number of lap for a track

The number of lap for a track is set up directly in the track scene.

-Open your track scene.

-In the Hierarchy select **LapCounterAndPosition** (spot 1).

In the Inspector, inside **LapCounterAndPosition** script change the **How Many Lap In The Current Race** (spot 2)



Tracks

This section describes track creation.

Table of contents:

Overview	Link
Track path: Add a point between two points that had been already created	Link
Track path: Remove a checkpoint	Link

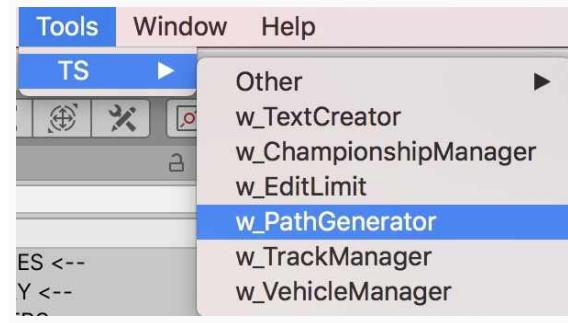
Overview

Basic info about track creation is contained in
Tuto 1: Create your first track
section 1-Create the track path

More info in section [Create your first track](#)
→ Page 14

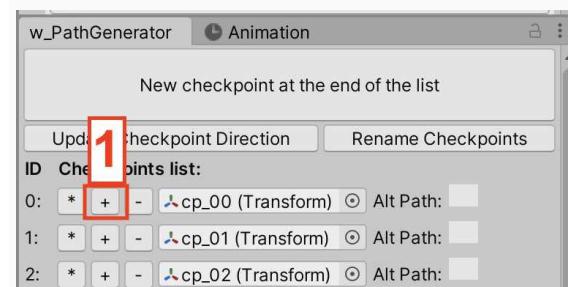
Track path: Add a point between two points that had been already created

-Open [w_PathGenerator](#)
(Tools → TS → w_PathGenerator)

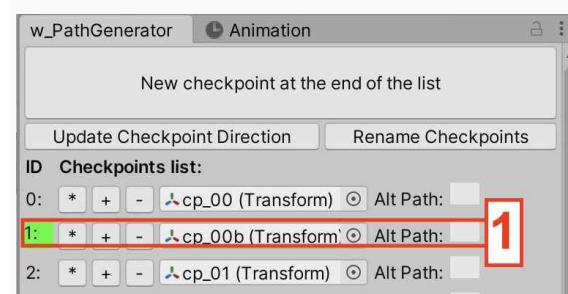


As an example we are going to create a new checkpoint between checkpoint 0 and checkpoint 1.

-Press + button next to cp_00 (spot 1).

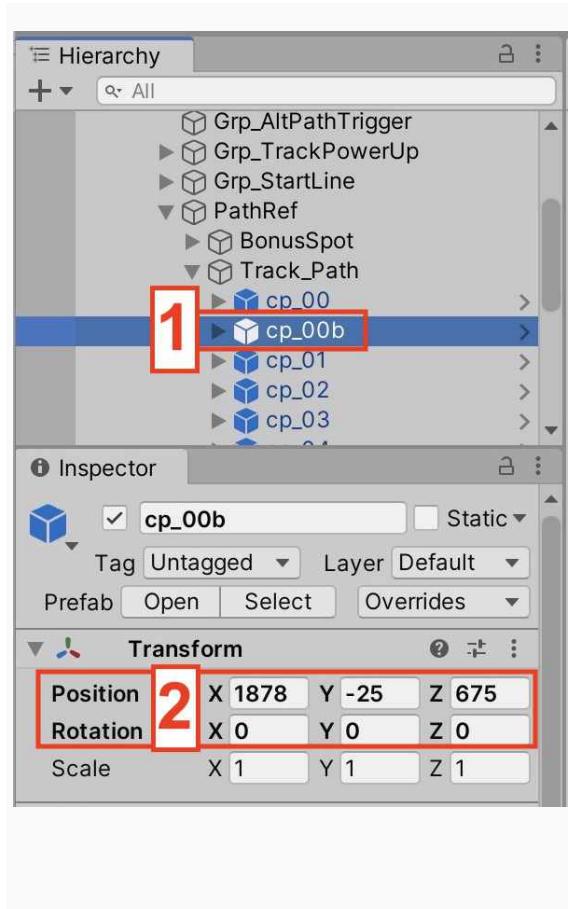


A new checkpoint is added between checkpoint 0 and checkpoint 1 (spot 1).



-In the Hierarchy select the new checkpoint named cp_00b (spot 1)

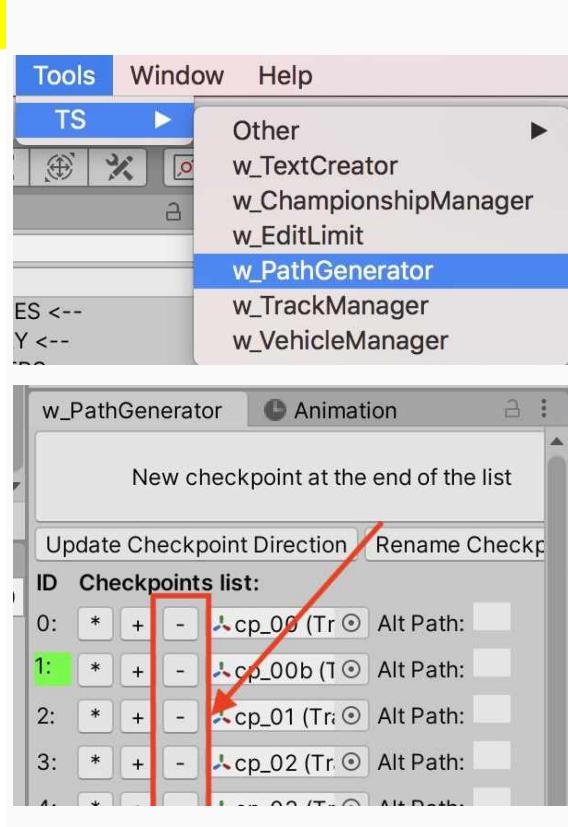
-Move the checkpoint in your scene or change its transform position (spot 2).



Track path: Remove a checkpoint

-Open w_PathGenerator
(Tools → TS → w_PathGenerator)

-Press - button next to the checkpoint you want to remove.



Start Line

This section describes the Start Line system. This system is used to detect when a vehicle finishes a lap. Basic info about Start Line creation are contained in Tuto 1: Create your first track section 2-Initialize the Start Line

Table of contents:

Overview	Link
Adjust vehicle positions on the starting grid	Link
Change Start Line Collider size	Link
Change Start Line 3D models	Link
Init Start Line position using track checkpoint 0	Link

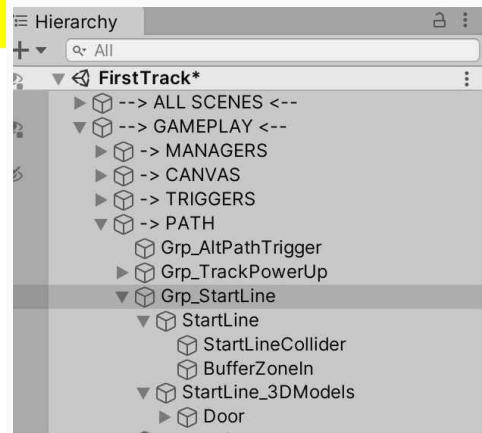
Overview

In the editor **Grp_StartLine** object allows to:

-Init the Start Line position using track checkpoint 0.

-Adjust vehicle positions on the starting grid.

-Adjust the size of the collider which detects if a vehicle passes through the start line.

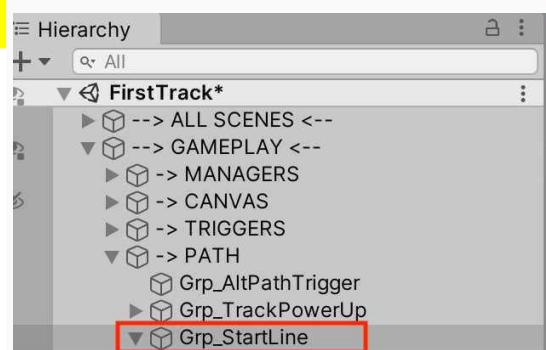


Adjust vehicle positions on the starting grid

(This section explains how to modify the vehicle positions on the starting grid)

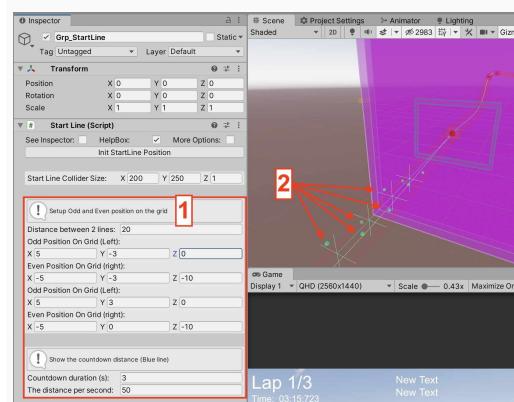
-In the Hierarchy select **Grp_StartLine**

(Hierarchy: GAMEPLAY → PATH → Grp_StartLine)



In the Inspector, section shown on picture (spot 1), allows to change the vehicle position on the starting grid.

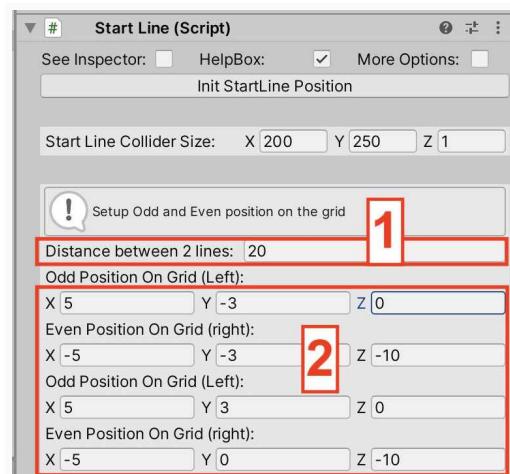
The position of each vehicle is represented by a green spot on Scene Tab (spot 2).



Distance between 2 lines represents the distance between 2 lines on the grid (spot 1).

The next 4 sections (spot 2) represent the vehicles positions on the starting grid.

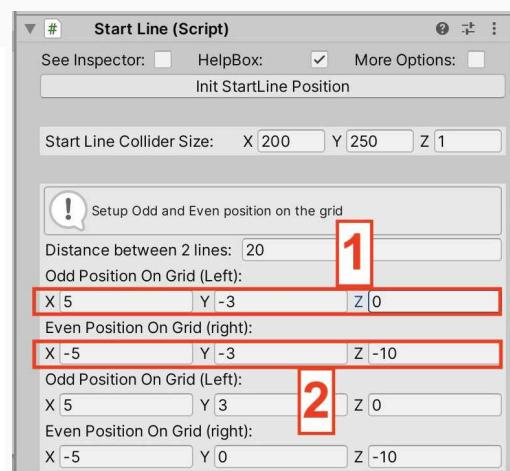
-In the Inspector change the values (spot 2) to change the vehicle position on the starting grid.



Note:

The 5th vehicle uses the position (spot 1)
The 6th vehicle uses the position (spot 2)

...



It is possible to modify the length of the starting grid:

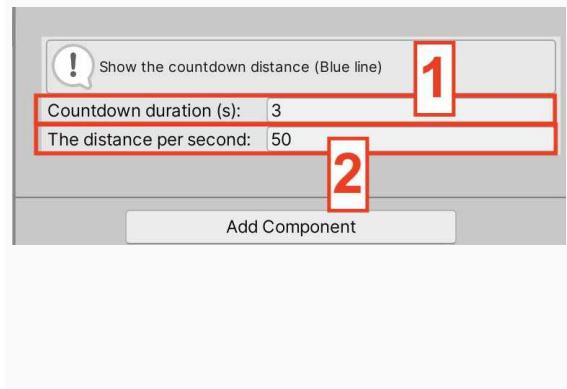
-Choose the countdown duration (spot 1).

(By default the countdown duration is equal to 3s.)

(This value doesn't affect the countdown animation. More info about how to modify the countdown animation in [Doc Part 2](#) section [Countdown System](#))

-Change the distance travelled by the vehicle in 1s (spot 2).

(By default the value = 50 because the default vehicle speed = 50. This allows the Player 1 to be on start line when the countdown is over.)



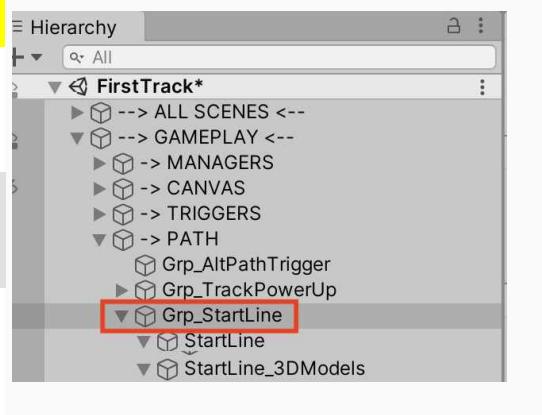
Change Start Line Collider size

When the player completes a lap he must go through the start line. It is possible to change the start line collider size.

IMPORTANT: Choose a starting line size that ensures the player will pass through the starting line at the end of each round.

-In the Hierarchy select **Grp_StartLine**

(Hierarchy: GAMEPLAY → PATH → Grp_StartLine)

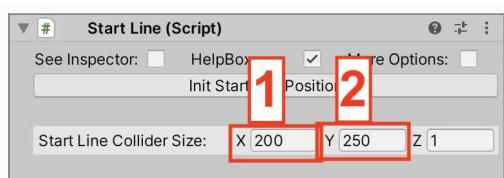


In the inspector change Start Line Collider Size

X (spot 1)

Y (spot 2)

(Note: Don't change Z size).



Change Start Line 3D models

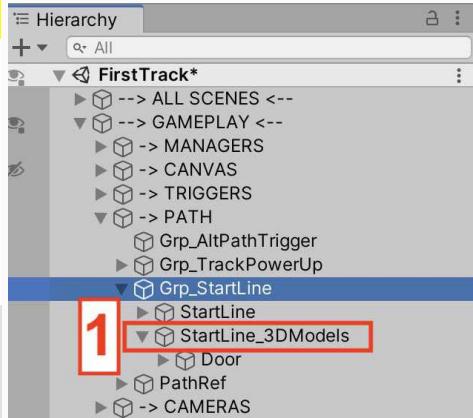
To change the Start Line 3D models:

-Remove objects inside StartLine_3DModels object.
(Hierarchy: GAMEPLAY → PATH → Grp_StartLine → StartLine_3DModels)

-Add your own 3D models in this folder (spot 1).

IMPORTANT:

Remove all the colliders attached to your 3D models.

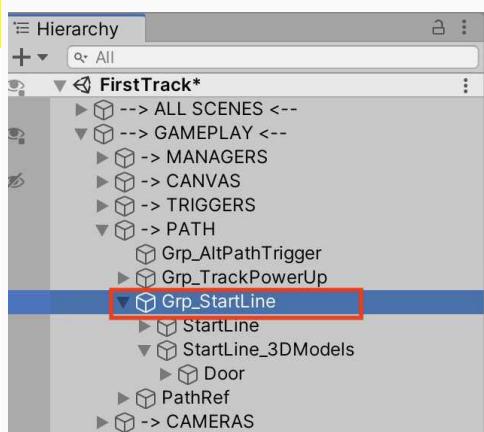


Init Start Line position using track checkpoint 0

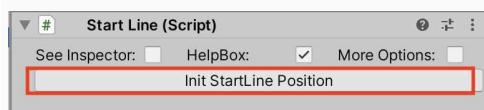
The start line must be on the same position as the first checkpoint of the track.

-In the Hierarchy select Grp_StartLine

(Hierarchy: GAMEPLAY → PATH → Grp_StartLine)



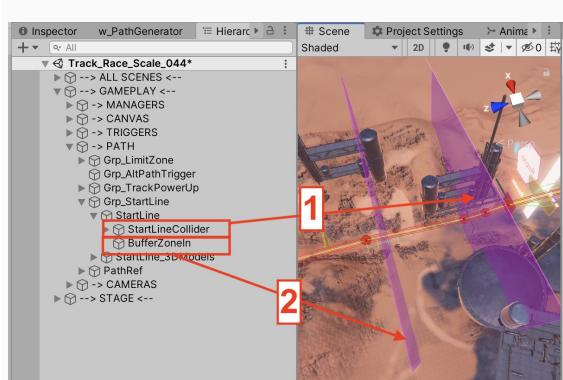
-In the Inspector, press Init StartLine Position to initialize the start line.



Info: When Init StartLine Position button is pressed StartLineCollider and BufferZoneIn are moved.

-StartLineCollider object is moved in the same position as the first checkpoint of the track. This object is used to detect when a vehicle complete a lap.

-BufferZoneIn object is moved next to the last checkpoint. This object is used to prevent bug when the vehicle is destroyed close to the first checkpoint.



Booster (Power-up)

This section describes Boosters.

Table of contents:

Overview	Link
Vehicle Booster Parameters	Link
Tips	Link
AI: Booster parameter depending AI difficulty	Link

Overview

Basic info about AI Booster creation are contained in
Tuto 1: Create your first track
section 8-Add AI booster

More info in section → Page 14

Vehicle Booster Parameters

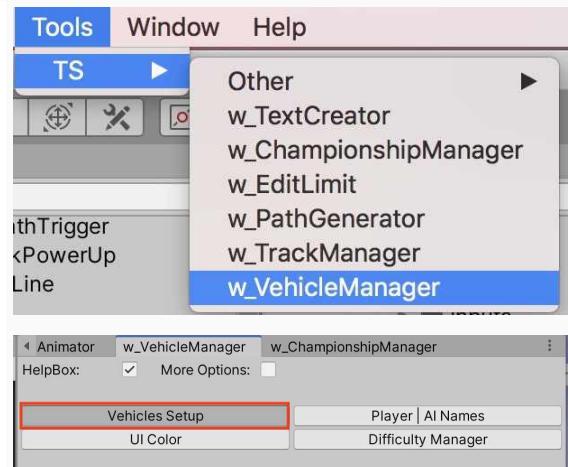
It is possible to choose booster values for each vehicle.

Values can be different for Player and AI.

Booster Parameters:

To access Booster parameters:

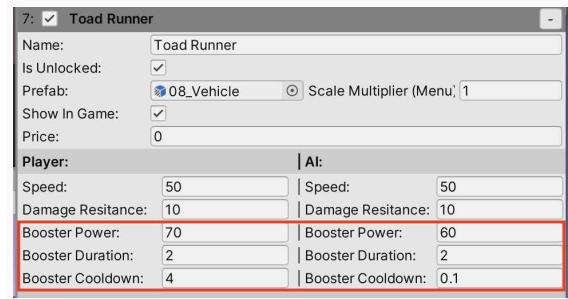
-Go to **Tools** → **TS** → **w_VehicleManager**



A new window appears.

-Press **Vehicle Setup** button.

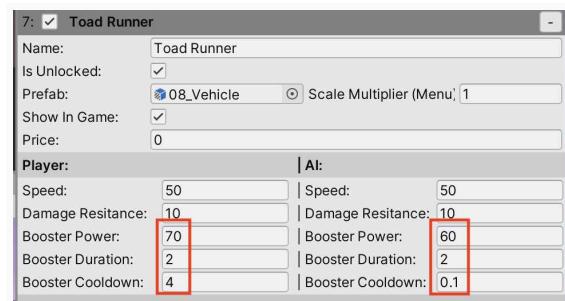
Inside each vehicle you can setup the booster parameters.



Booster Power: Speed offset is applied to the vehicle when Booster is activated.

Booster Duration: The higher the value, the longer the booster is active.

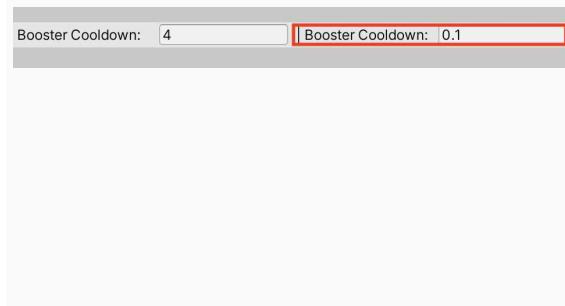
Booster cooldown: the duration to reload the booster.



Tips:

Booster Cooldown for AI is very short (0.1).

With this value,
in most case, AI booster is full when the booster is activated.



AI: Booster parameter depending AI difficulty

Depending the AI difficulty:

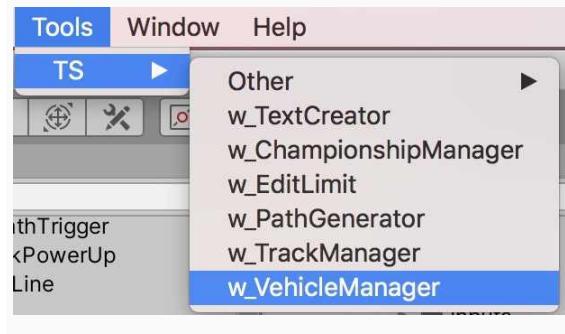
It is possible to choose the booster applied to each AI vehicle.

To access Booster parameter:

-Go to **Tools → TS → w_VehicleManager**

A new window appears.

-Press **Difficulty Manager** button.

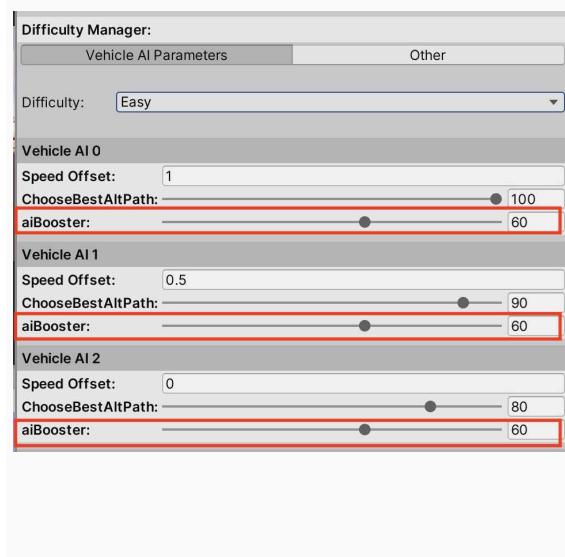


For each Ai vehicle you can choose a percentage (0-100) of booster applied to this AI vehicle.

For example if the vehicle has a
Booster Power value = 60

And if the aiBooster percentage = 50

The Booster Power applied to the vehicle during the race will be $((60 * 50) / 100) = 30$.



You can edit the value for AI vehicle in each difficulty (easy, medium, hard).

Alt Path

This section describes Alternative Path.

Table of contents:

Overview	Link
Alt Path priority	Link
Alt Path parameter depending AI difficulty	Link
Remove Alternative Path	Link
Reminder	Link

Overview

Basic info about Alt Path creation are contained in
Tuto 1: Create your first track
section 10-Create AI alternative Path

More info in section → Page 14

Alt Path priority

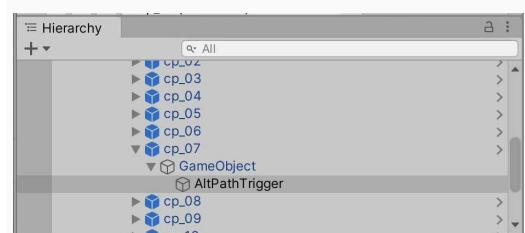
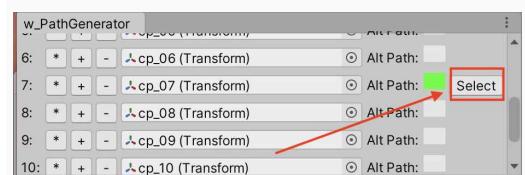
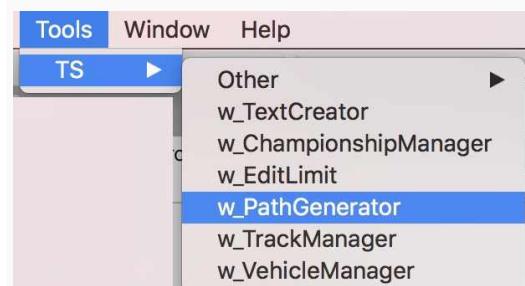
When Alt Path are used it is possible to choose the path considered to be the best path by the AI vehicles.

This section explains how to select this best path.

-Go to [Tools](#) → [TS](#) → [w_PathGenerator](#)

-In [w_PathGenerator](#) press [Select](#) to select the Alt Path group you want to set up.

For example in the picture on the right if Select spot 1 is pressed:
We are going to set up alt paths starting from checkpoint 7
(cp_07)

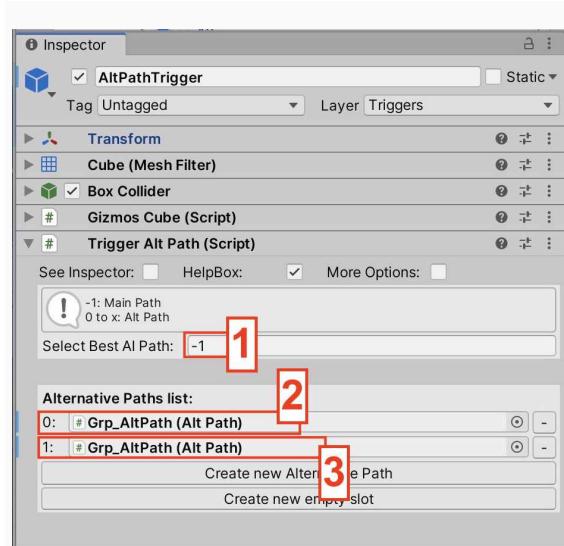


-In the field **Select Best Alt Path** (spot 1):

Choose the value that corresponds to the path you want the AI to consider as the best path.

The value **-1** corresponds to the main
 The value **0** corresponds to the Alt Path with the ID = 0 (spot 2)
 The value **1** corresponds to the Alt Path with the ID = 1 (spot 3)

...



Alt Path parameter depending AI difficulty

Depending the AI difficulty:

It is possible to choose the percentage of chance that the AI uses the best path.

To access best Path percentage parameter:

-Go to **Tools → TS → w_VehicleManager**.

A new window appears.

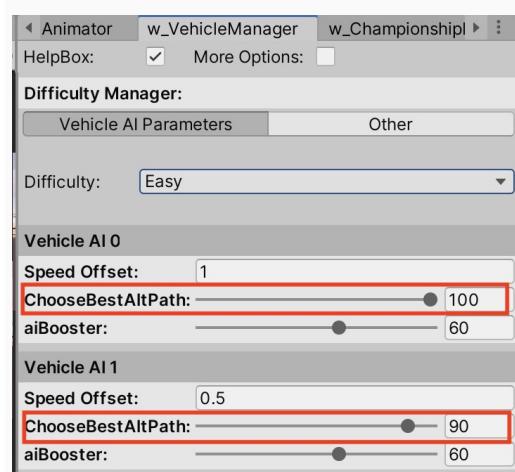
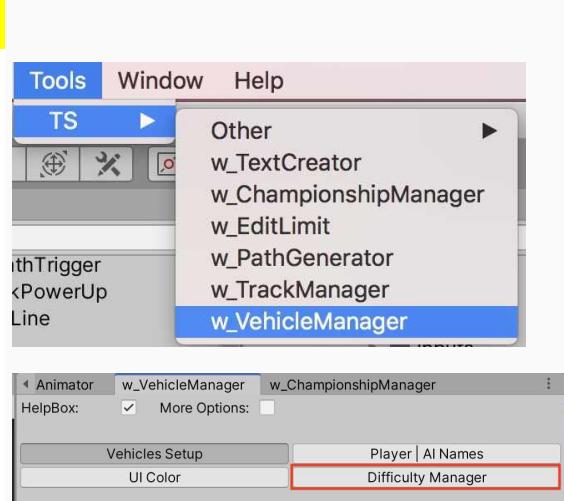
-Press **Difficulty Manager** button.

For each AI vehicle you can choose a percentage (0-100).

For example if the vehicle has a ChooseBestAltPath = 60.

The chance the AI uses the best path will be 60%.

If the AI doesn't choose the best path it chooses a randomly a path.

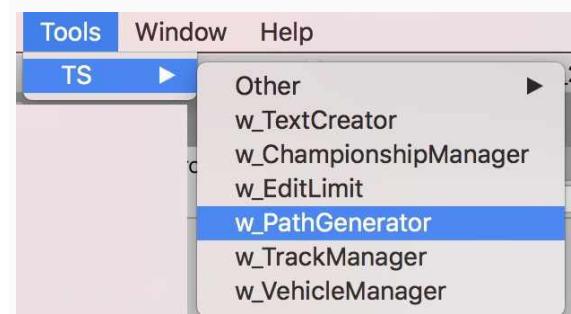


You can edit the value for AI vehicle in each difficulty (easy, medium, hard).

Remove Alternative Path

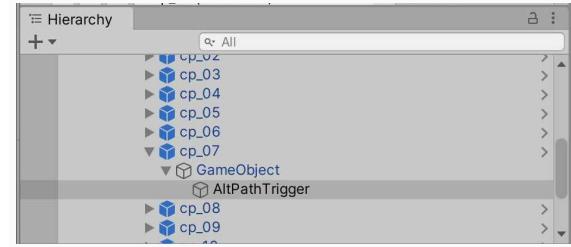
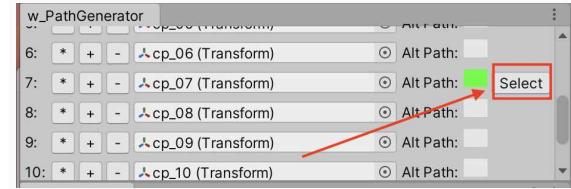
This section explains how to remove an Alt Path.

-Go to Tools → TS → w_PathGenerator

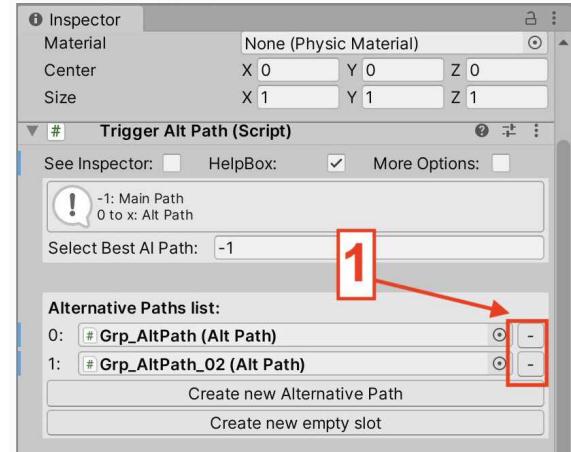


-In w_PathGenerator press Select to select the Alt Path group you want to modify.

For example in the picture on the right if Select spot 1 is pressed: We are going to modify alt paths starting from checkpoint 7 (cp_07)



-Press - button next to the Alt path you want to remove.



Reminder

An AI alternative path is a path that:

- Starts on the main track path
- Ends on the main track path

-Do not create Power-ups group connected to the last point of the Alternative Path.

-A trigger **must not** touch another trigger.
It will cause issues. Some triggers can be ignored.

-Do not put 2 triggers in the same position.

Game Area Limit

This section describes the system to create the game area limitation.

Table of contents:

Overview	Link
Game Area Limit: Move a point	Link
Game Area Limit: Add point	Link
How to: Create a simple limit collider	Link
Reminder	Link

Overview

Basic info are contained in
Tuto 1: Create your first track
section 6-Create the game area limits

More info in section → Page 14

Game Area Limit: Move point

It is possible to move an existing point in the game area limit.

To edit a point position the game area limit must be visible in the scene view (spot 1).

If the game area limit is not visible:
Disable the eye and the hand next to Grp_LimitZone object
(spot 2).

-Go to **Tools** → **TS** → **w_EditLimit**

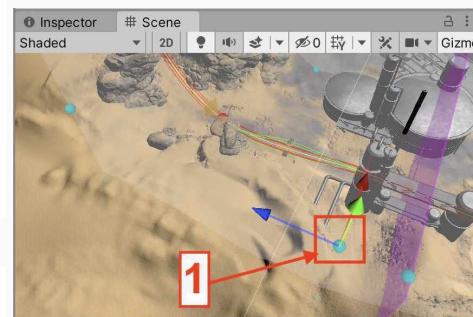
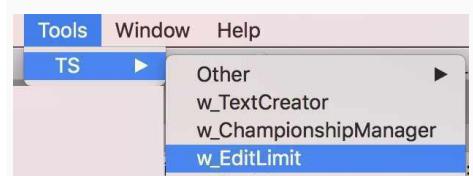
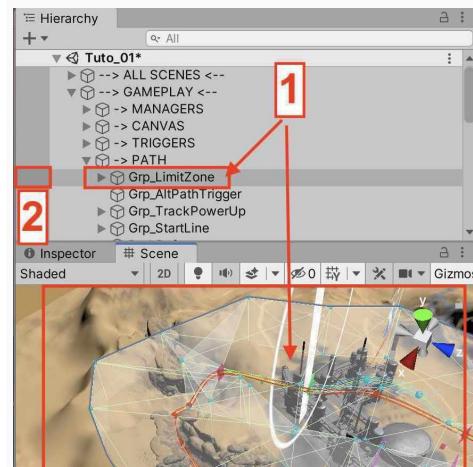
A new window appears.

-Select a spot (blue sphere) on the game area limit (spot 1).

-Now you can move the new spot.

Info:

It is possible to select and move multiple points at the same time.



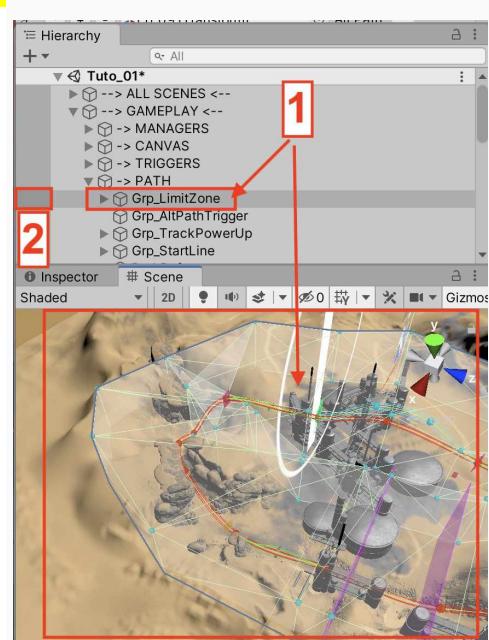
Game Area Limit: Add point

It is possible to add point between two existing points in the game area limit.

This section explains how to add a point between two existing points in the game area limit.

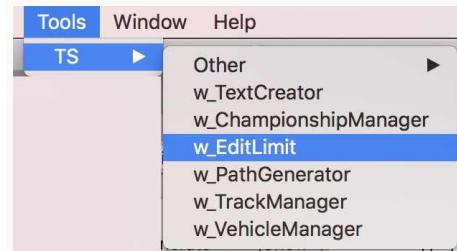
Game area limit must be visible in the scene view (spot 1)

If the game area limit is not visible:
Disable the eye and the hand next to Grp_LimitZone object (spot 2).



-Go to Tools → TS → w_EditLimit

A new window appears.

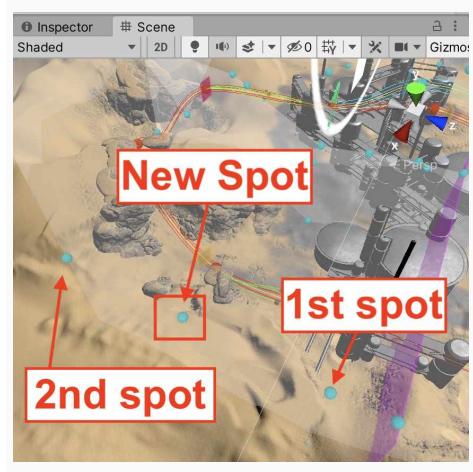
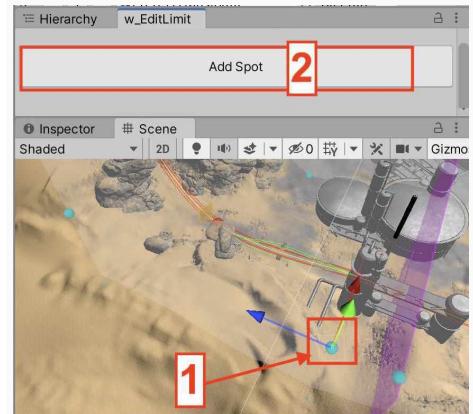


Select a spot (blue sphere) on the game area limit (spot 1).

-In w_EditLimit window press Add Spot (spot 2).

A new spot is created between the 1st spot and second spot.

You can select and move the new spot.



How to: Create a simple limit collider

If you don't want to use the default game area limit you can create your own limits.

This section explains how to create a box object that can be used as a limit collider.

-Go to **Tools → TS → w_EditLimit**

A new window appears.

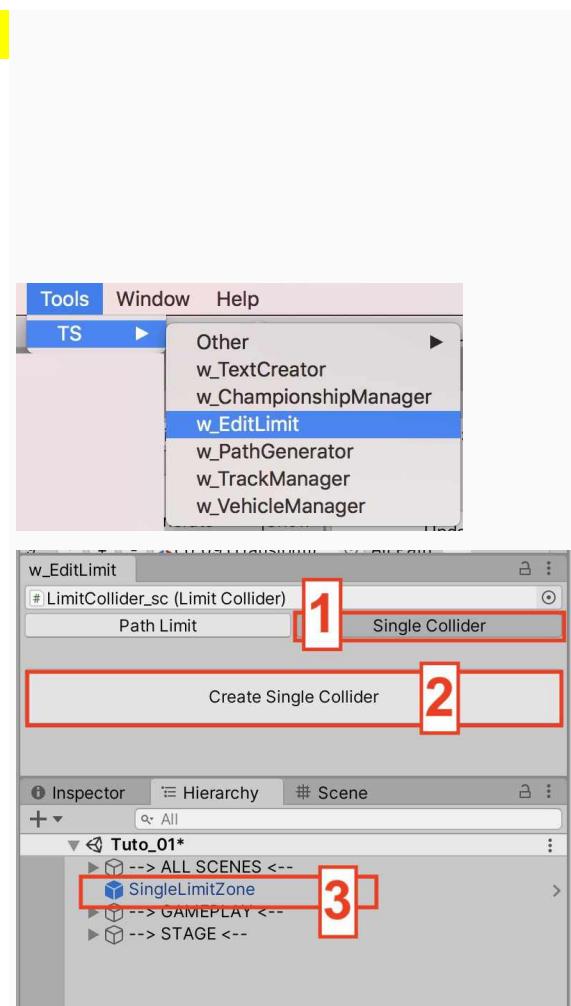
-Press **Single Collider** button (spot 1)

-Press **Create Single Collider** button (spot 2).

A new object is created in the Hierarchy.

Move this object where you want in the scene.

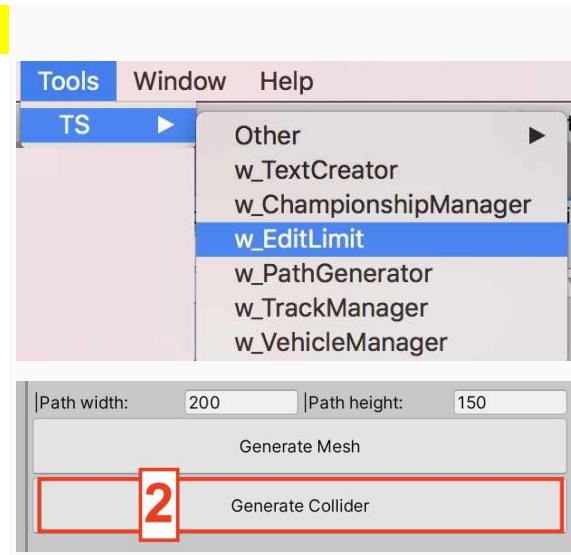
When a vehicle goes through this object, the vehicle will be destroyed.



Reminder

-To modify the Game Area Limit you **MUST** open the **w_EditLimit** window
(*Tools → TS → w_EditLimit*)

-You **MUST** press **Generate Collider** button to update the Area Limit collider.



Minimap

This section describes **Grp_Minimap** object that can be found in the Hierarchy.

(Hierarchy: GAMEPLAY → CANVAS → Grp_Minimap)

During the game the object is used to display the map of the track and the position of each vehicle.

Table of contents:

Overview	Link
Create the Minimap	Link
Modify the Minimap position, scale and background	Link

Overview

In the editor:

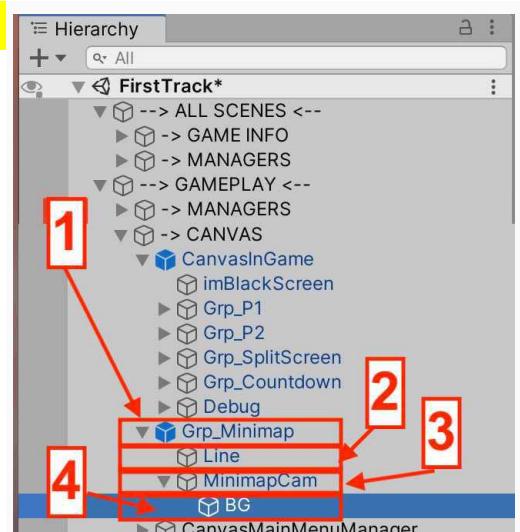
-**Grp_Minimap** object allows to create the Minimap using track checkpoints (spot 1).

Ingame:

-**Line** object is used to display the track path using a line renderer (spot 2).

-**MinimapCam** is a camera used to generate a RenderTexture. This RenderTexture is displayed in **CanvasInGame** object. (spot 3).

-**BG** is an object to display a background (spot 4). It is possible to disable this object.

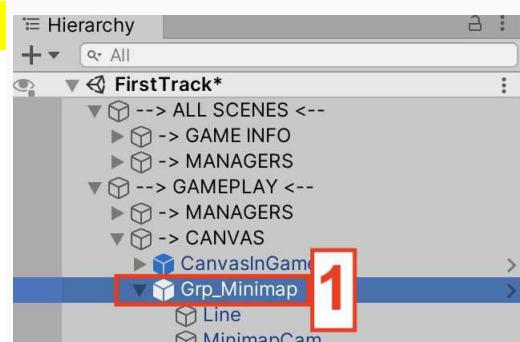


Create the Minimap

(This section explains how to generate the minimap)

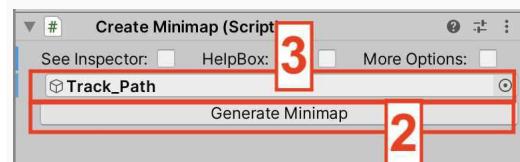
-In the Hierarchy select **Grp_Minimap** (spot 1).

(Hierarchy: GAMEPLAY → CANVAS → Grp_Minimap)

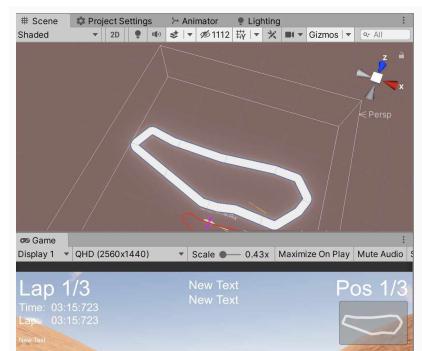


-In the Inspector press **Generate Minimap** button (spot 2).

Note: If it doesn't work check if the track is connected to the slot (spot 3).



(The track minimap is generated in the Scene Tab and shown in Game View)



Modify the Minimap position, scale and background

To change the position of the minimap:

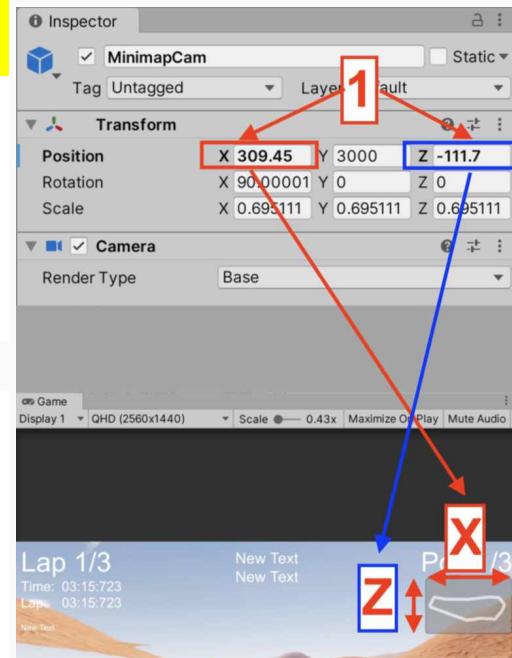
In the Hierarchy select **MinimapCam**

(Hierarchy: GAMEPLAY → CANVAS → Grp_Minimap → MinimapCam)

In the Inspector change the cam transform position (spot 1).

Note:

Modify X an Z axis only.



To change the scale of the minimap:

In the Hierarchy select **MinimapCam**

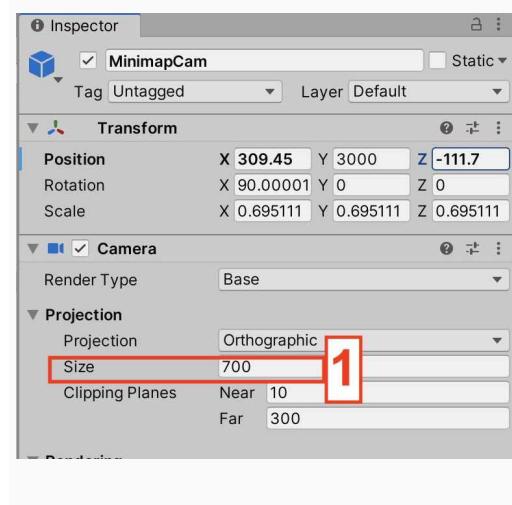
(Hierarchy: GAMEPLAY → CANVAS → Grp_Minimap → MinimapCam)

In the Inspector change the **Size** of camera (spot 1)

(Inspector → Projection -> Size).

Increase Size value to zoom out.

Decrease Size value to zoom in.



Optional:

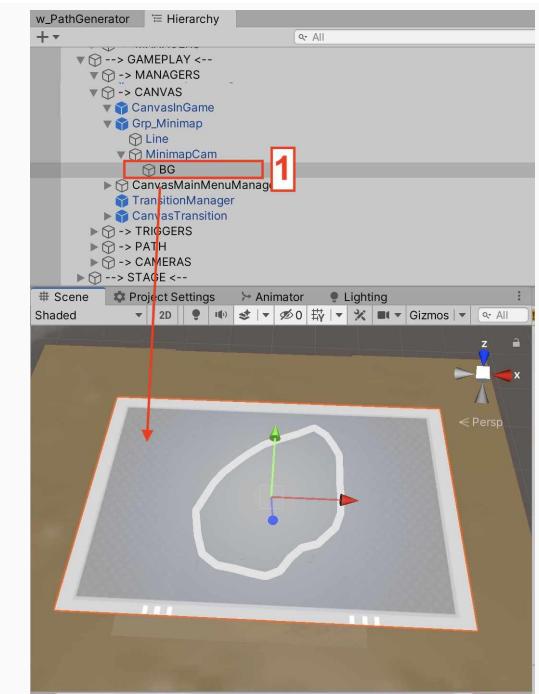
The **BG** object (spot1)

(Hierarchy: GAMEPLAY → CANVAS → Grp_Minimap → MinimapCam → BG)

is used as a background for the minimap.

If needed:

In the Inspector you can change its size and replace the image.

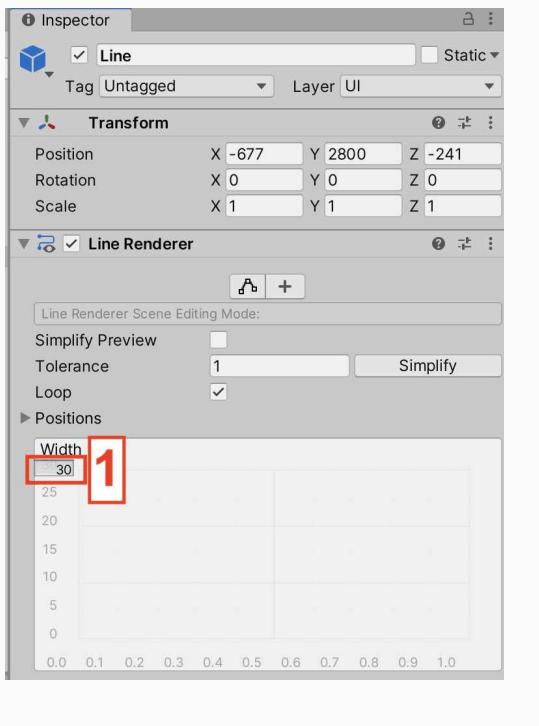


To change the size of the minimap line:

In the Hierarchy select **Line**

(Hierarchy: GAMEPLAY → CANVAS → Grp_Minimap → Line)

In the Inspector change **Width** (spot 1)



Power-ups

This section describes the power-ups system included in the asset.

Table of contents:

Create a Power-up group that can be used by AIs + PlayerPrefs	Link
Create a Power-up only used by players	Link
Customize vehicle Power-ups parameters (missile, mine, machine gun,...)	Link
Customize Power-up object (3D models and sound fx).	Link
Reminder	Link

Create a power-up group that can be used by AIs + PlayerPrefs

Basic info are contained in

Tuto 1: Create your first track

section 7-Add Power-ups to your track

More info in section → Page 14

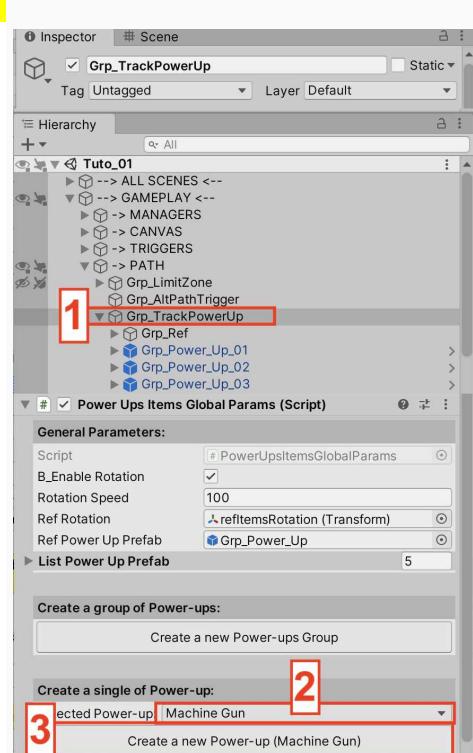
Create a Power-up only used by players (P1 or P2)

-In the Hierarchy select **Grp_TrackPowerUp** (spot 1)

(Hierarchy: GAMEEPLAY → PATH → Grp_TrackPowerUp)

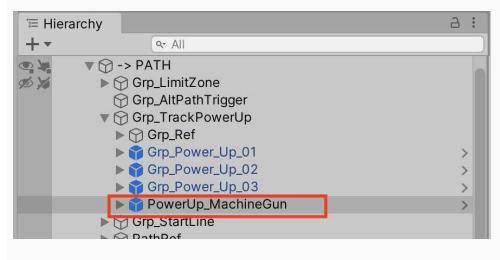
In the Inspector select the power-up in the dropdown menu (spot 2)

-Press button **Create a new Power-up** (spot 3)



A new power-up is created and automatically selected in the Hierarchy.

-Move the object where you want in the scene.

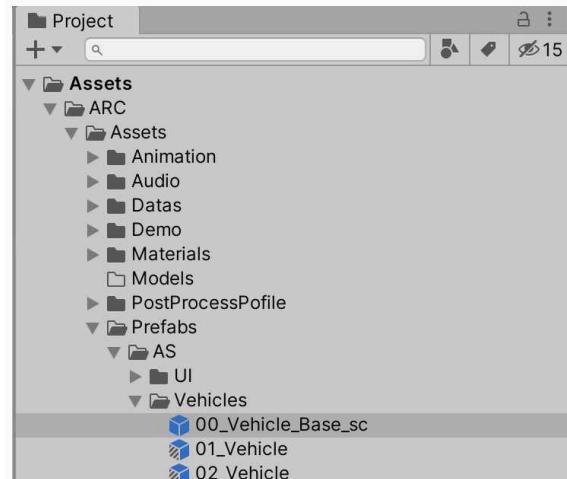


Customize vehicle Power-ups parameters (missile, mine, machine gun,...)

It is possible to customize Power-ups parameters.

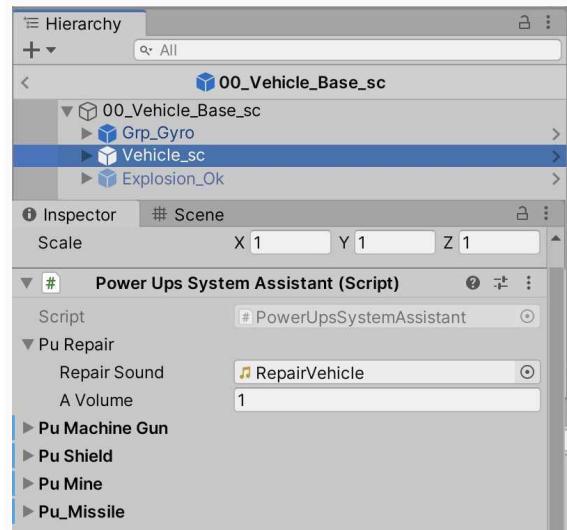
To change the parameters for all the vehicles at the same time open the prefab **00_Vehicle_Base_sc**

(Project Tab: Assets → Prefabs → AS → Vehicles → 00_Vehicle_Base)



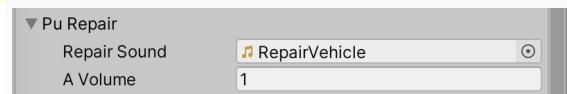
To access power-ups parameters:

- Open the vehicle prefab (**00_Vehicle_Base_sc**)
- Select **Vehicle_sc**
- Go to the script **PowerUpsSystemAssistant**



Repair

It is possible to change the volume and the sound played when the Repair power-up is enabled.



Machine Gun

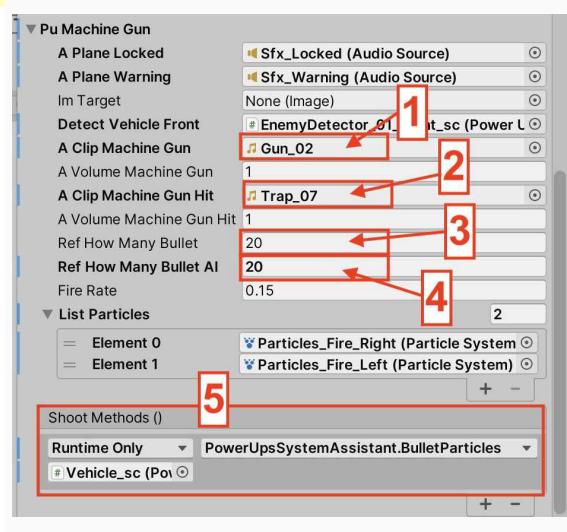
AClipMachine (spot 1): Sound plays when bullets are fired.

AclipMachinGunHit (spot 2): Sound plays when vehicle is hit by bullet.

RefHowManyBullet (spot 3): the number of bullets for player 1 and 2.

RefHowManyBulletAI (spot 4): the number of bullets for AIs.

Shoot Methods (spot 5): Custom methods called when the vehicle uses Machine Gun.



Shield

Grp_Shield: Object enabled when the power-up is enabled.

By default the object is in: [Vehicle_sc](#) → [Grp_MoveWithWings](#) → [Grp-PowerUps](#) → [Grp_ShieldForceField](#)

You can change this object to create your custom shield.



Shield Sound: Sound plays when Shield enabled.

Shield Duration: How long is the vehicle invulnerability

Mine

Mine Sound: Sound plays when Mine is enabled.

HowManyMine: The number of mines that can be launched by the player.

HowManyMineAI: The number mines that can be launched by AI.

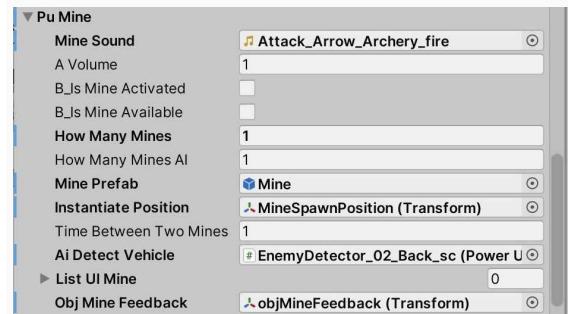
MinePrefab: It is the prefab instantiated when the vehicle activate a mine.

It is possible to customize this prefab with your own 3D models.

ObjMineFeedback: This object is enabled when the vehicle activate a mine.

By default the object is in: [Vehicle_sc](#) → [Grp_MoveWithWings](#) → [Grp-PowerUps](#) → [Grp_MineFeedback](#) → [Feedback](#) → [ObjMineFeedback](#)

You can change this object to create your custom feedback



Missile

Missile Sound: Sound plays when Missile is enabled.

HowManyMissile: The number of missile launched when the vehicle activate the power-up.

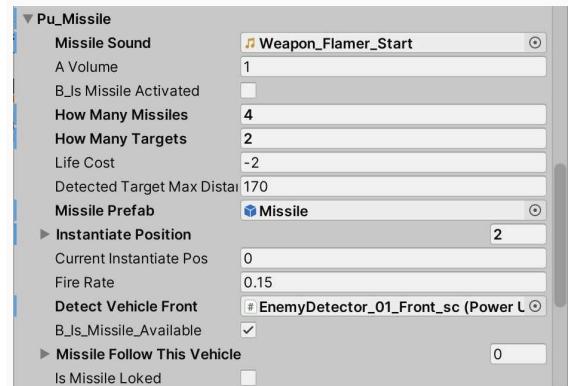
HowManyTarget: By default the vehicle can't touch more than 2 differents vehicles with the missile.

LifeCost: The number of life points losed by vehicle when a missile touch it.

DetectedTargetMaxDistance: The vehicle can't detect enemies far from this distance.

Missile Prefab: It is the prefab instantiated when the vehicle activate a missile.

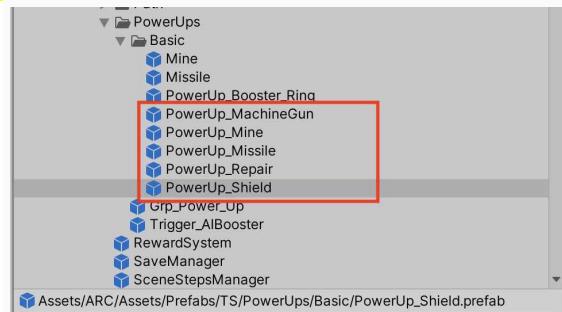
It is possible to customize this prefab with your own 3D models.



Customize Power-up object (3D models and sound fx)

Power-ups are in:

Project Tab: Assets → Prefabs → TS → PowerUps → Basic



Change the 3D Model

-Select a Power-up prefab.

(Project Tab: Assets → Prefabs → TS → PowerUps → Basic)

-Open the Prefab.

-Change objects inside **Grp_Item**



Change the Sound Fx played when Power-up is caughted by a vehicle.

-Select a Power-up prefab.

(Project Tab: Assets → Prefabs → TS → PowerUps → Basic)

-Open the Prefab.

-Select the first object in the Hierarchy.

-In the Inspector change the audioClip .



Reminder

-Do not create Power-ups group connected to the last point of the Alternative Path.

Camera

This section describes the camera system used to follow players 1 and 2 during a race.

Table of contents:

Overview	Link
How to: Change camera view order	Link
How to: Modify the default camera views	Link
How to: Create a new camera view	Link

Overview

Camera system is managed by different modules:

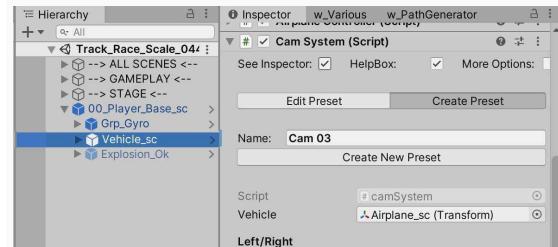
Each vehicle has a script **CamSystem.cs** attached to it. (*00_Player_Base_sc* → *Vehicle_sc*)

This script allows to:

-Managed the behavior of the object followed by the camera.

-Managed camera behavior when the vehicle is destroyed.

-Edit/Create camera view presets.



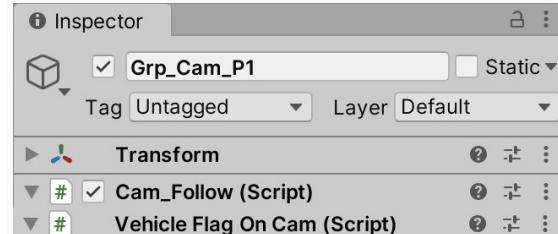
In each gameplay scene:

-The script **CamFollow.cs**

(*Hierarchy: GAMEPLAY* → *CAMERAS* → *Grp_CamP1*)

(*Hierarchy: GAMAPLAY* → *CAMERAS* → *Grp_CamP2*)

manages the camera behavior for Player 1 and Player 2.



Camera is render on LateUpdate().

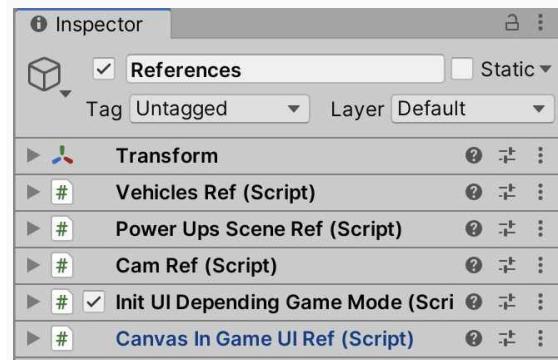
-The script **CamRef.cs**

(*Hierarchy: GAMPLAY* → *MANAGERS* → *References*)

allows to access:

-Player cameras

-Post-effect on camera P1 and P2



-The script **VehicleVisibleByCamList.cs**

Hierarchy: GAMEPLAY → MANAGERS → *CheckIfVehicleVisibleByCamera*

Allows to access:

-To the list of the vehicles visible by the camera P1 or P2

Grp_PostFx object contains camera Post Fx for Player 1 and Player 2.

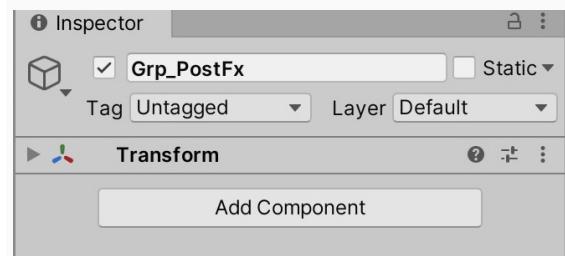
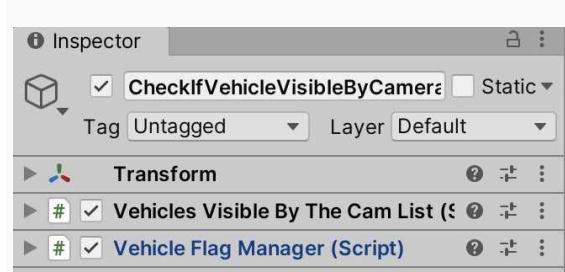
Hierarchy: GAMEPLAY → CAMERAS → *Grp_PostFx*

Effect_P1 contains the script **TS_PostProcess.cs** which manages Player 1 Post Fx.

Hierarchy: GAMEPLAY → CAMERAS → *Grp_PostFx* → *Effect_P1*

Effect_P2 contains the script **TS_PostProcess.cs** which manages Player 2 Post Fx.

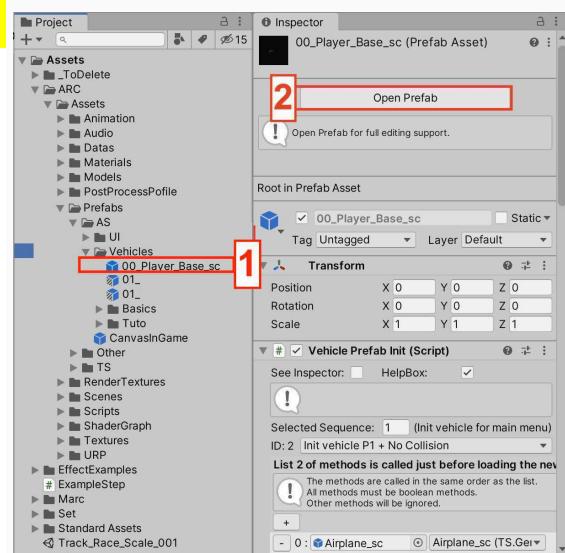
Hierarchy: GAMEPLAY → CAMERAS → *Grp_PostFx* → *Effect_P2*



How to: Change camera view order

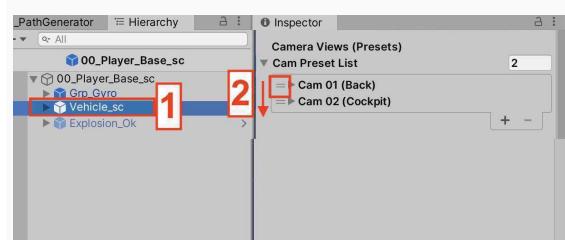
-In Project Tab select **00_Player_Base_sc** prefab (spot 1) (Project tab: → Assets → Prefab → AS → Vehicles → *00_Player_Base_sc* prefab)

-In the Inspector press **Open Prefab** button (spot 2)



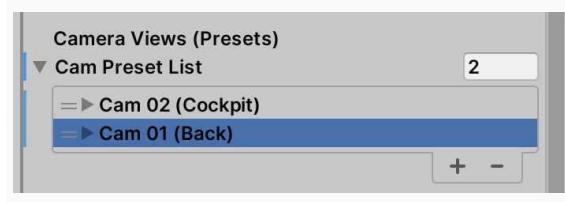
-In the Hierarchy select **Vehicle_sc** (spot 1)

-In the Inspector hold left click on **2 lines** icon (spot 2) and drag the icon at the end of the list.



Now the cameras order has changed.

Cam 02 is displayed when the game starts. Cam 01 is displayed when the player presses the button to switch cameras.



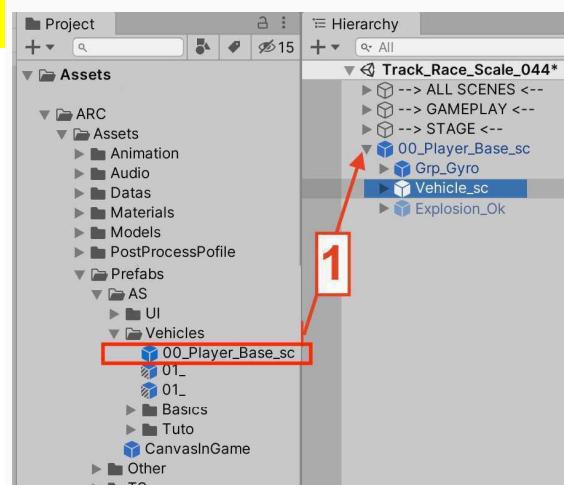
How to: Modify the default camera views

This section explains how to customize the 2 default camera views named:

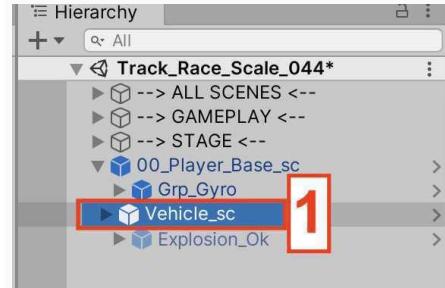
- Cam 01 (Back)
- Cam 02 (Cockpit)

-Open a gameplay scene.

-In Project Tab drag and drop **00_Player_Base_sc** prefab (spot 1) (Project tab: Assets → Prefabs → AS → Vehicles → **00_Player_Base_sc** prefab)



-In the Hierarchy select **Vehicle_sc** (spot 1)
(Hierarchy: **00_Player_Base_sc** → **Vehicle_sc**),



-In the Inspector press **Edit Preset** button (spot 1).

-For this example select in the dropdown menu **Cam 01 (Back)** (spot 2)

Change camera parameters (spot 3):

Break position:

It is the distance from the camera to the camera target when the player press break button.

Acceleration pos:

It is the distance from the camera to the camera target when the player doesn't press break button.

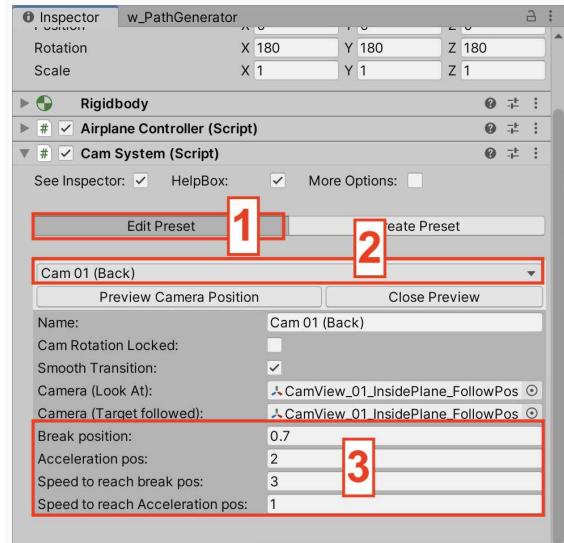
Speed to reach break pos:

It is the speed to move the camera to the break position.

Speed to reach Acceleration pos:

It is the speed to move the camera to the acceleration position.

IMPORTANT: Don't change the other parameters. If you want to create your own camera view go to section [How to: Create a new Camera view.](#)

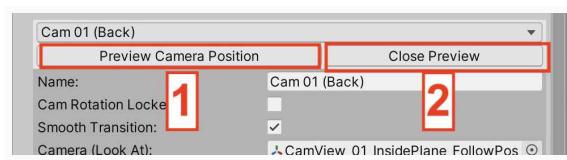


To visualize your modifications press button **Preview Camera Position** (spot 1).

If you don't see the camera position on your scene view:

Press **Close Preview** button (spot 2)

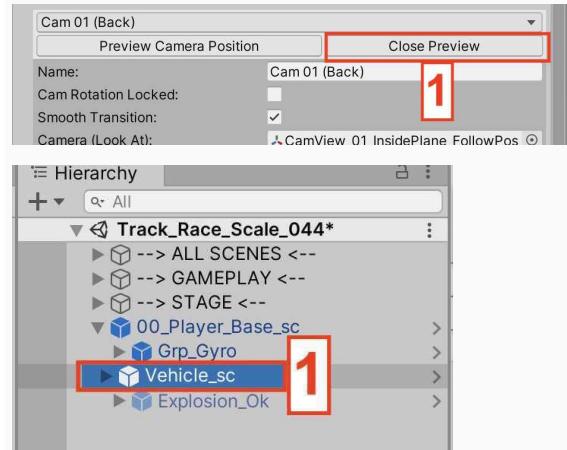
Then press **Preview Camera Position** button (spot 1)



When you have finished your modification:

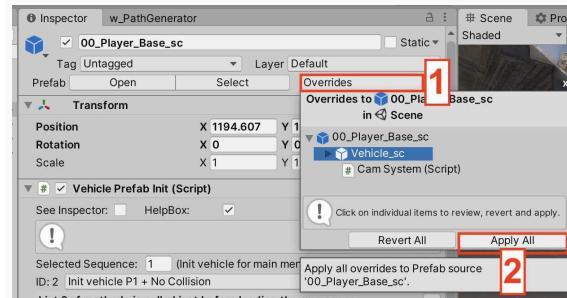
Press **Close Preview** button (spot 1)

-In the Hierarchy select **Vehicle_sc** (spot 1)
(Hierarchy: *00_Player_Base_sc* → *Vehicle_sc*),



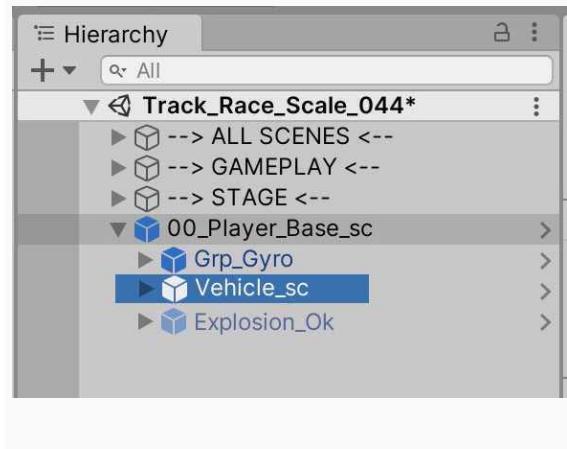
-In the Inspector press **Overrides** (spot 1)

-Then Press **Apply All** (spot 2)



-Delete **00_Player_Base_sc** (spot 1) from the Hierarchy.

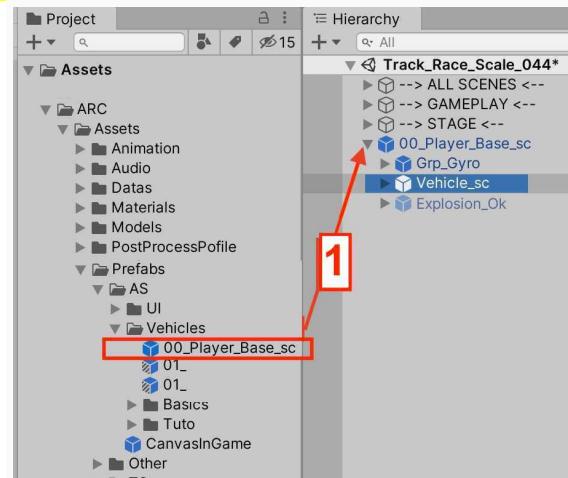
(Hierarchy: *00_Player_Base_sc* → *00_Player_Base_sc*),



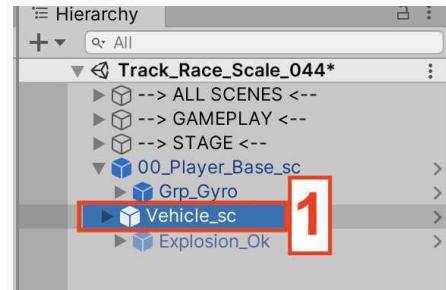
How to: Create a new camera view

-Open a gameplay scene.

-In Project Tab drag and drop **00_Player_Base_sc** prefab (spot 1) (Project tab: Assets → Prefabs → AS → Vehicles → **00_Player_Base_sc**)



-In the Hierarchy select **Vehicle_sc** (spot 1)
(Hierarchy: **00_Player_Base_sc** → **Vehicle_sc**),

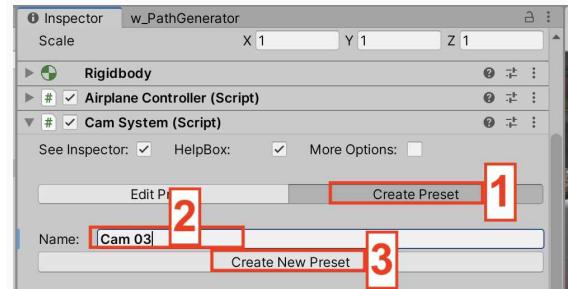


-In the Inspector press **Create Preset** tab (spot 1).

-Choose and write a name for your new camera
(spot 2)

-Press button **Create New Preset** button (spot 3)

(The new preset is automatically selected in Edit Preset tab)

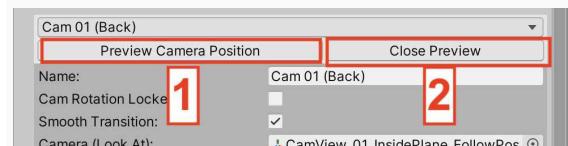


To vizualize your modifications press button **Preview Camera Position** (spot 1).

If you don't see the camera position on your scene view:

Press **Close Preview** button (spot 2)

Then press **Preview Camera Position** button (spot 1)



Info:

The camera is not updated automatically. You must press **Preview Camera Position** to show the modifications.

Here a description of each parameter:

(As example you can have a look to Cam 01 and Cam 02 presets to see how the camera views are setup)

Cam Rotation Locked:

(Checked) The camera use the Camera (look at) object transform rotation.

(Unchecked) The camera look at the Camera (look at).

Smooth Transition:

(Checked) Smoother transition with the previous camera.

Camera (Look at):

The camera looks at this object.

Camera target followed:

The camera follow the position of this object.

Break position

It is the distance from the camera to the camera target when the player press break button.

Acceleration pos:

It is the distance from the camera to the camera target when the player doesn't press break button.

Speed to reach break pos :

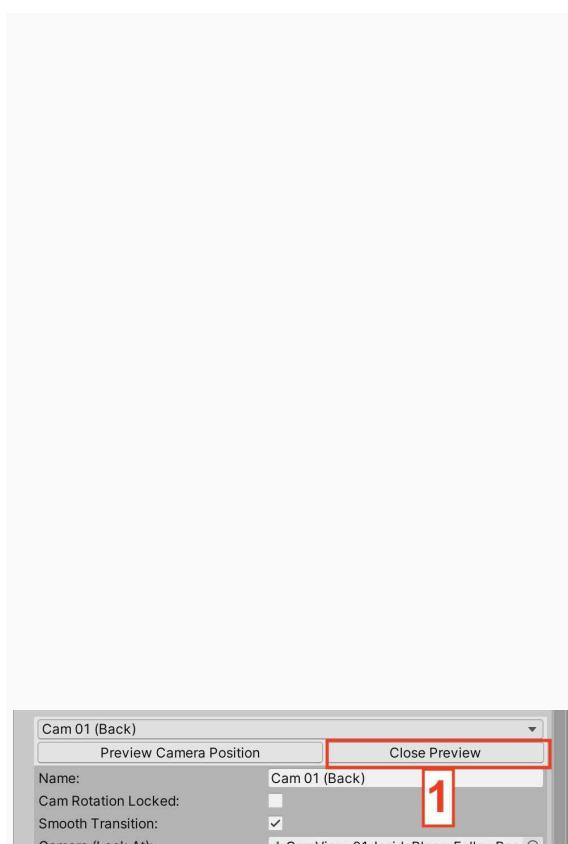
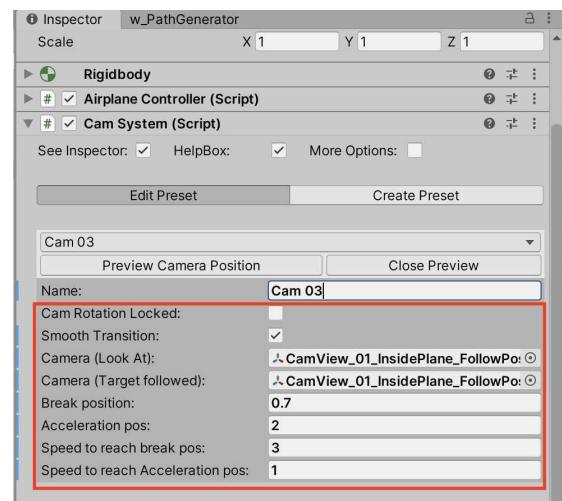
It is the speed to move the camera to the break position.

Speed to reach Acceleration pos:

It is the speed to move the camera to the acceleration position.

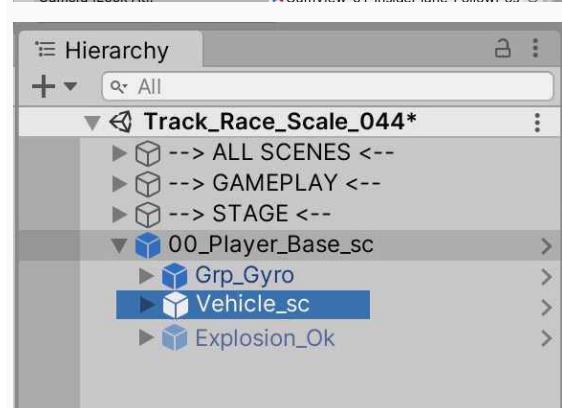
When you have finished your modification:

Press **Close Preview** button (spot 1)



-In the Hierarchy select **Vehicle_sc** (spot 1)

(Hierarchy: 00_Player_Base_sc → Vehicle_sc),

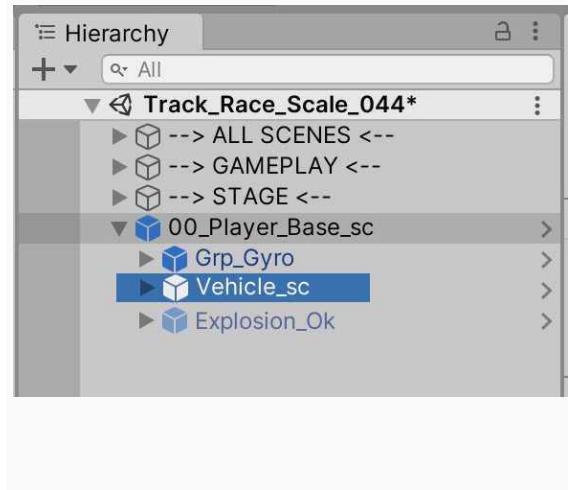
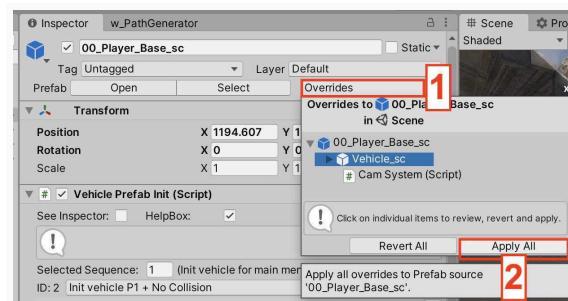


-In the Inspector press **Overrides** (spot 1)

-Then press **Apply All** (spot 2)

-Delete **00_Player_Base_sc** (spot 1) from the Hierarchy.

(Hierarchy: **00_Player_Base_sc**)



Scenes

This section gives information about the stater kit and the how a scene is initialized.

Table of contents:

Starter Kit

[Link](#)

Scene Initialization (Order of execution Diagram)

[Link](#)

Starter Kit

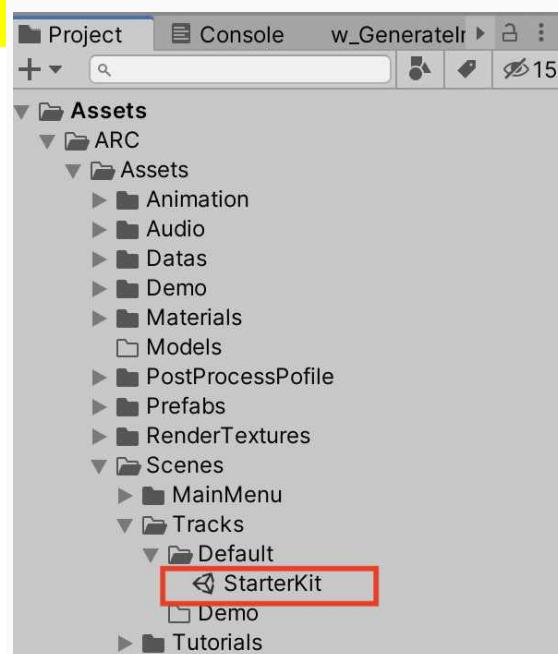
When you want to create a new track you can use the Starter kit scene.

This scene contains all the needed elements to create a track (Managers, lightings,...).

- Duplicate the scene.
- Rename it.
- Open it.
- Create your track.

You will find the Start kit in folder:

Project tab: Assets → Scenes → Tracks → Default → StarterKit

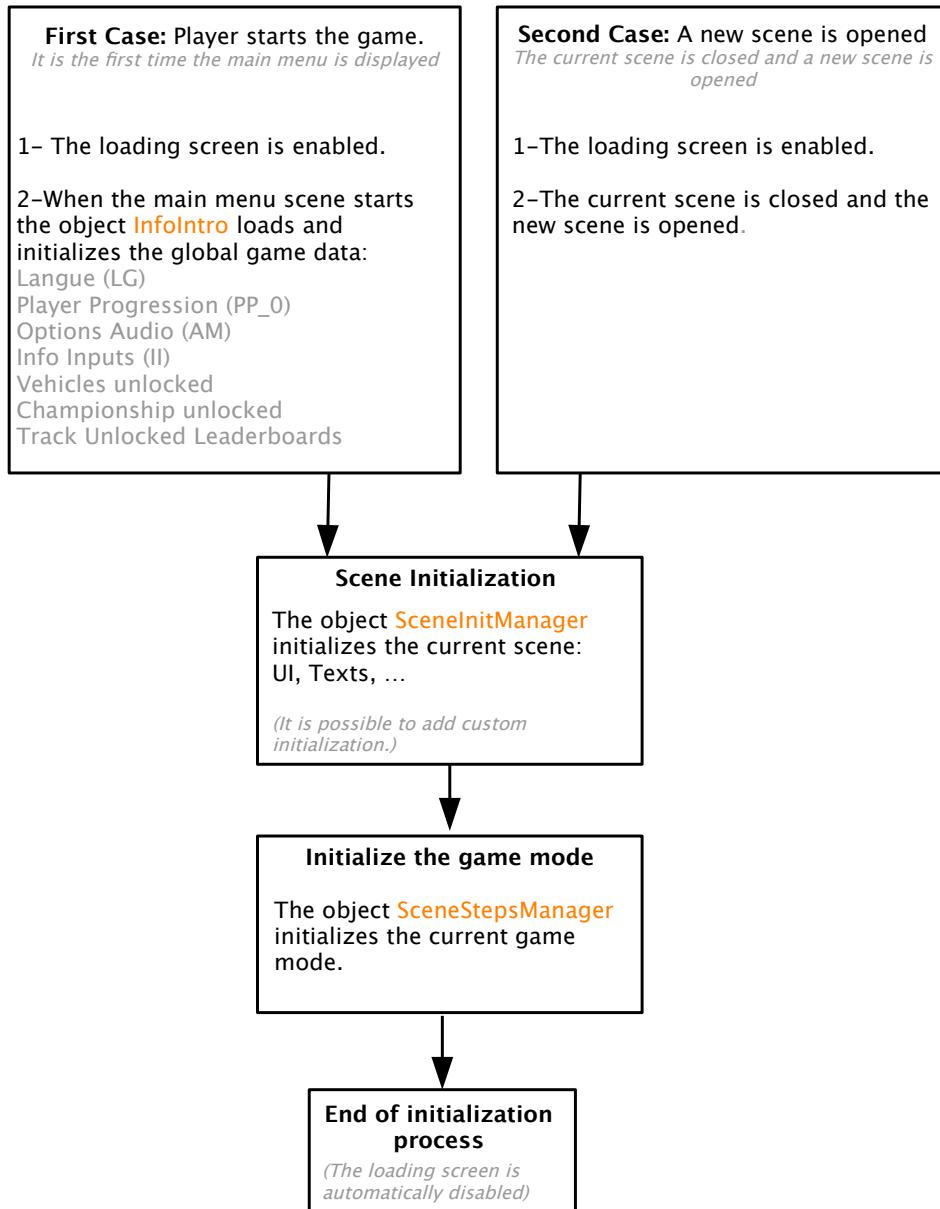


Info:

When you want to create a new track it is also possible to:

- Duplicate a track you like.
- Rename it.
- Create you track.

Scene Initialization (Order of execution Diagram)



Global Game Data

The global data are used to save the player progression (Player progression, Vehicles, coins, championship, ...)

Table of contents:

- | | |
|---|----------------------|
| Which object in the hierarchy loads Global Data | Link |
| How to: Load custom data when game starts | Link |
| Scripting: Custom method to load data (example) | Link |

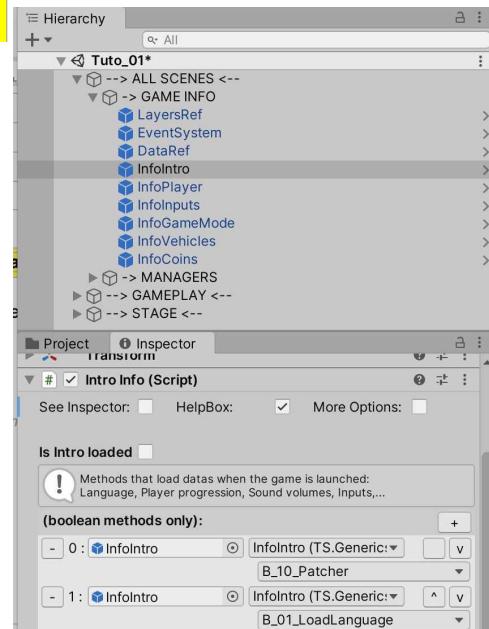
Which object in the hierarchy loads Global Data:

InfoIntro is used to load global game data when the player starts the game.

(Hierarchy: ALL SCENES → GAME INFO → InfoIntro)

By default Patcher, Load Language, Player Progression, Sound Params, Inputs and leaderboards are loaded and initialized.

InfoIntro is also used to know if it is the first time the Main Menu has been opened.



How to: Load custom data when game starts

This section explains how to load your custom data when the player starts the game.

-In the Hierarchy select **InfoIntro** (spot 1)
(Hierarchy: ALL SCENES → GAME INFO → InfoIntro)

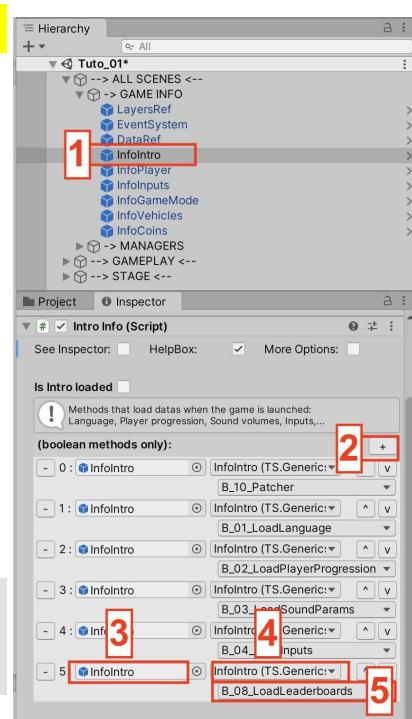
-In the Inspector press + button (spot 2)

-Drag and drop an object from the hierarchy that contains the script that need to be use (spot 3).

-Select the script that need to be use (spot 4).

-Select your boolean method (spot 5).

IMPORTANT: Only boolean method can be used.
Other methods are ignored.
Find an example of boolean in the next section.

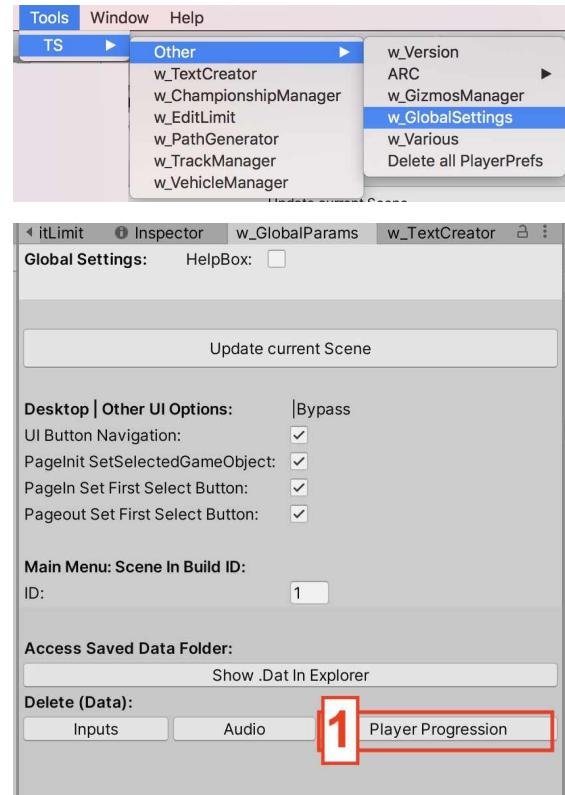


VERY IMPORTANT

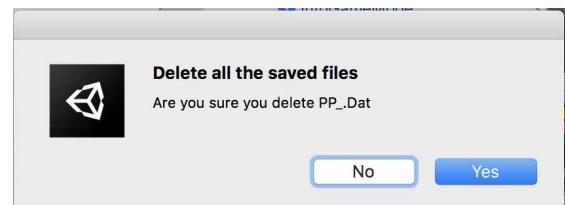
Next step is very important. After adding new element in the Global Game Data it is needed to remove existing data.

-Go to Tools → TS → Other → w_GlobalSettings

-Press Player Progression button (spot 1).



-Press Yes.



Scripting: Custom method to load data (example)

Here an example of custom boolean to load custom data:

Find this piece of code in the script:
InfoIntroAssistant.cs attached to **InfoIntro**
(Hierarchy: ALL SCENES → InfoIntro)

```
298           //--> Example Load Custom Datas
299           public bool B_09_LoadCustomData()
300           {
301               #region
302               // Code: Load custom data
303               return true;
304               #endregion
305           }
306           }
307       }
308   }
309 }
```

Scene Initialization

This section gives information about the scene initialization system.

Table of contents:

- | | |
|---|----------------------|
| Which object initialize the scene | Link |
| Scripting: Useful methods to Initialize a scene | Link |
| How to: Initialize a custom module when a scene is loaded | Link |

Which object initialize the scene

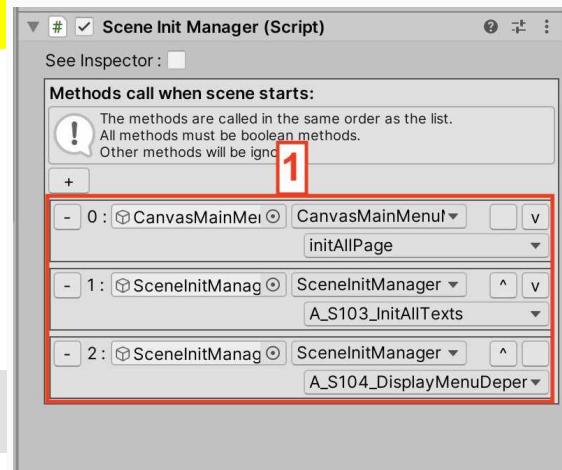
This object **ScenelnitManager** is used to initialize a scene.

In the Main Menu scene (Hierarchy: MAIN MENU → MANAGERS → ScenelnitManager)

In a gameplay scene (Hierarchy: GAMEPLAY → MANAGERS → ScenelnitManager)

There is a **ScenelnitManager** object in each scene.

IMPORTANT: There must **only** have one objet ScenelnitManager in a scene.



When a new scene is loaded:

-The object **ScenelnitManager** is called to initialize the current scene.

The methods contained in the list (spot 1) are called one by one.

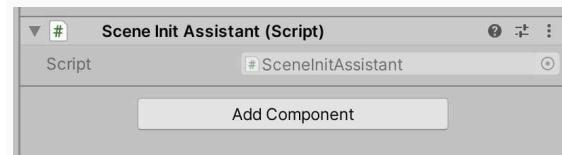
By default those methods initialize different parts of the scene:

- Initialize all the menu page
- Update UI Text depending the selected language
- Call the object that manage game mode

IMPORTANT: Only boolean method can be used in the list. Other methods are ignored.

This script **SceneInitAssistant.cs** attached to object **ScenelnitManager** contains:

- Custom boolean methods called by script **ScenelnitManager.cs**.



Scripting: Useful methods to Initialize a scene

To initialize manually the scene call:

```
StartCoroutine(  
    SceneInitManager.instance.CallAllTheMethodsOneByOne());
```

To check if the initialization is finished:

Check if:

```
SceneInitManager.instance.IsInitDone = true
```

How to: Initialize a custom module when a scene is loaded

During the initialization of the scene the script **SceneInitManager.cs** call one by one a list of methods.

Those methods are used to initialize the different scene modules.

It is possible to add a method to the list to initialize a custom module.

This section explains how to add a custom method to the list.

-In the Hierarchy select **SceneInitManager** (spot 1)

In the Main Menu scene (Hierarchy: MAIN MENU → MANAGERS → SceneInitManager)

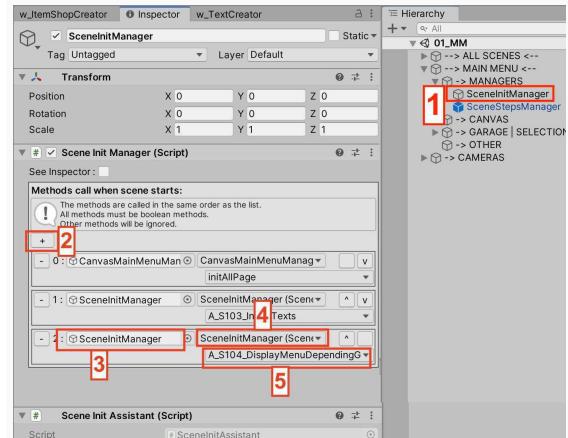
In a gameplay scene (Hierarchy: GAMEPLAY → MANAGERS → SceneInitManager)

-In the Inspector press + button (spot 2)

-Drag and drop an object from the hierarchy that contains the script that need to be use (spot 3).

-Select the script that need to be use (spot 4).

-Select the boolean method that need to be use (spot 5).



IMPORTANT: Only boolean method can be used.
Other methods are ignored.

The load scene manager

This section describes **LoadSceneManager** object that can be found in the Hierarchy.
(Hierarchy: ALL SCENES → MANAGERS → LoadSceneManager)

Table of contents:

Overview	Link
Scripting: How to: Load a new scene	Link
How to: Call a method when a new scene is loaded	Link

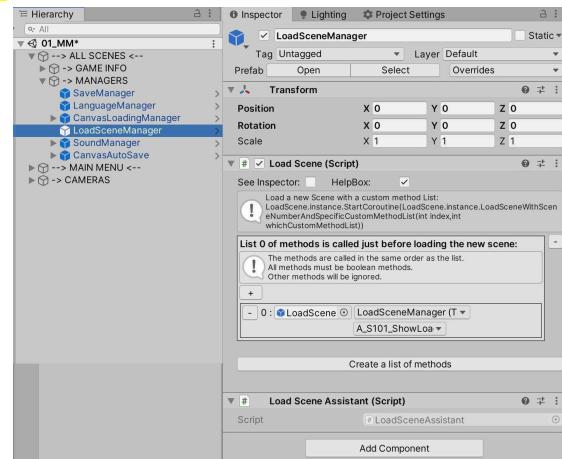
Overview

LoadSceneManager object managed the process to load a new scene.

(Hierarchy: ALL SCENES → MANAGERS → LoadSceneManager)

LoadScene.cs script attached to **LoadSceneManager** allows to:

- 1-Close the current scene and open a new scene.
- 2-Call methods just before loading the scene.
- 3-Load the new scene



IMPORTANT:

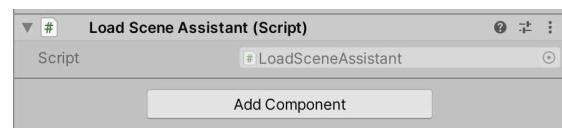
Methods are called using the same order as the list.
Methods **MUST** be boolean methods.

The object **is not destroyed** when a new scene is loaded.

You must have **only 1 LoadSceneManager** in your scene.

It is possible to override the prefab to apply the modification in all scenes.

This script **LoadSceneManagerAssistant.cs** attached to object **LoadSceneManager** contains:
-Custom methods called by script **LoadScene.cs**.
(*LoadScene.cs* is attached to **LoadSceneManager**)



Scripting: How to: Load a new scene

Call the method:

```
StartCoroutine(  
    LoadScene.instance.  
    LoadSceneWithSceneNumberAndSpecificCustomMethodList(  
        int index,  
        int whichCustomMethodList))
```

Index (int): represents scene number in the build
Settings ([Files → Build Settings](#))

whichCustomMethodList (int): represents the list of
custom methods called before loading the scenes.

Info: By default List 0 is called.

Scripting: Know if the loading is finished.

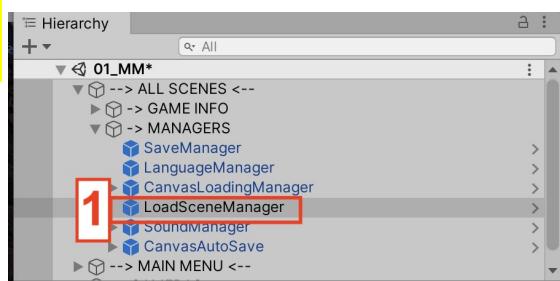
Inside [LoadSceneManager.cs](#) the boolean
b_IsLoadingFinished is turned to **true** after the loading
process.

It is possible to check if the loading process is finished
with:

```
LoadScene.instance.IsLoadingFinished
```

How to: Call a method when a new scene is loaded

-In the **Hierarchy** select **LoadSceneManager** (spot 1).
(Hierarchy: ALL SCENES → MANAGERS → LoadSceneManager)



-Press **+** to add a new empty slot (spot 1)

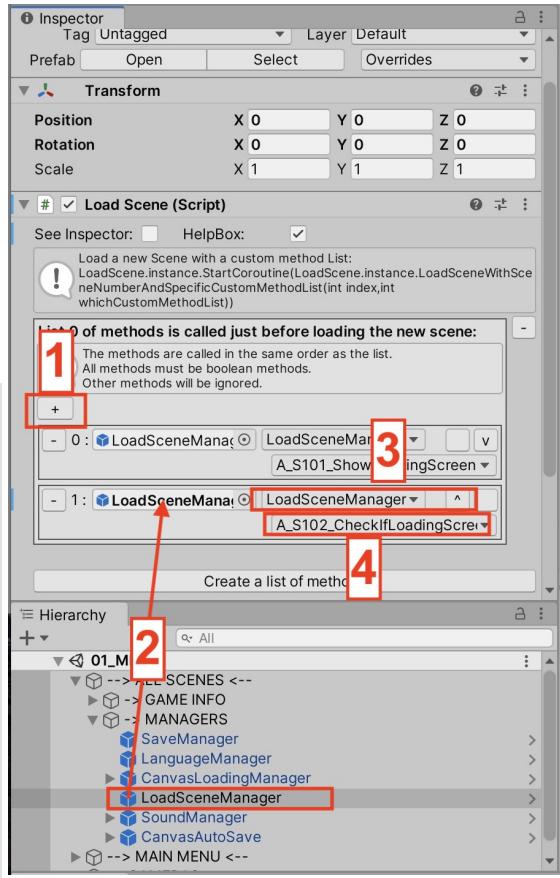
-Drag and drop the object that contains the boolean method (spot 2)

-Select the script (spot 3)
and the method (spot 4).

IMPORTANT: Methods must be attached to an object that is not destroyed when a scene is loaded.

I suggest you:

- To add a script to object **LoadSceneManager**
- To write and call your methods from this new attached script.



The Scene Step Manager

This section describes **SceneStepManager** object that can be found in the Hierarchy.
(Hierarchy: GAMEPLAY → MANAGERS → SceneStepsManager)

Table of content:

Overview	Link
How to: Enable/Disable a step	Link
Modify a Step	Link
Scripting: Useful methods to use the Scene Step System	Link

Overview

The **SceneStepManager** is a system to initialize the different steps during a race.

As an example, when a gameplay scene starts we want to:

- Step 0: Prepare the scene to launch the race.
- Step 1: Start the race.
- Step 2: Display the result screen after the race.

The **SceneStepManager** allows to create those initialization steps for each Game Mode.

*The **SceneStepManager** is not managed automatically. Each step must be called with a script.*

By default when the scene is initialized the **SceneInitManager** is called to initialize the scene for the selected game mode.

When the countdown ended the **SceneStepManager** is called to Init and Start the race.

When the player complete the race the **SceneStepManager** is called to displayed the result page.

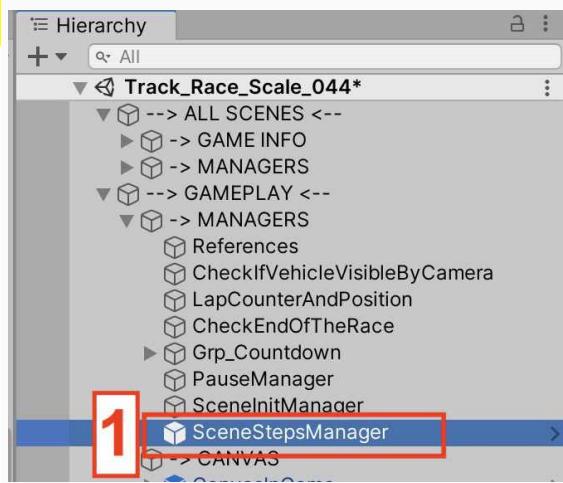
How to: Enable/Disable a step

Some steps can be disabled. By default camera views displayed during countdown can be disabled.

As an example we are going to disable the camera views displayed during the countdown in Time Trial Game Mode.

-Open a gameplay scene.

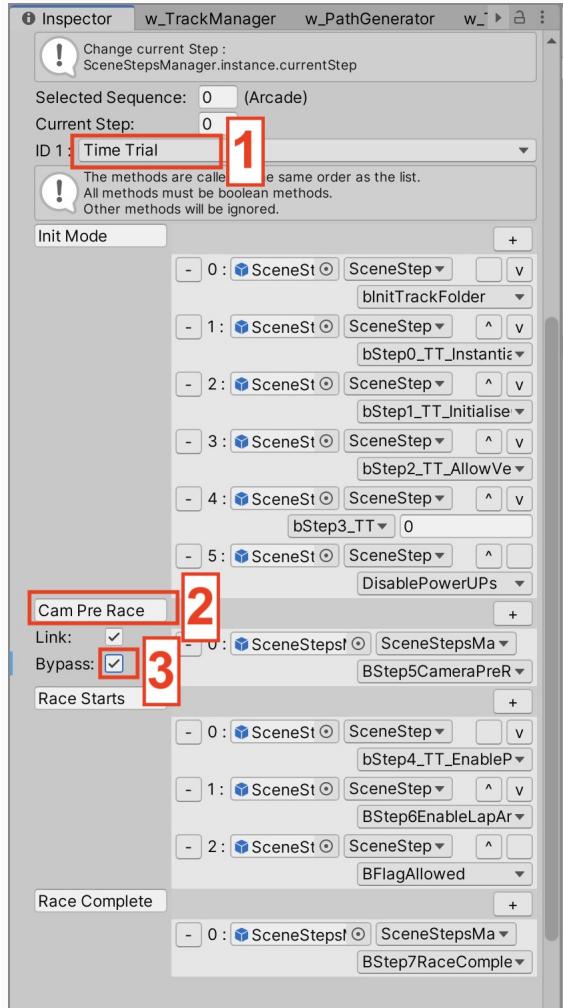
-In the Hierarchy select **SceneStepsManager** (spot 1)
(Hierarchy: GAMEPLAY → MANAGERS → SceneStepsManager)



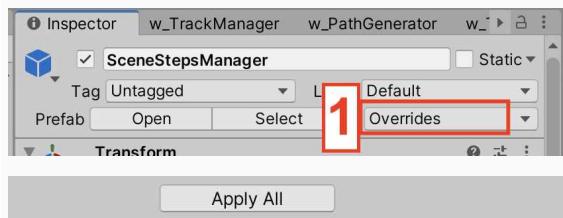
-In the Inspector select **Time Trial** in the dropdown menu (spot 1).

-Go to step **Cam Pre Race** (spot 2)

-Check box **Bypass** (spot 3)



-On the top of the Inspector press **Overrides** button (spot 1)



-Press **Apply All**.

Now the player starts a race in Time Trial mode camera views displayed during countdown are disabled. It is applied for all the scenes.

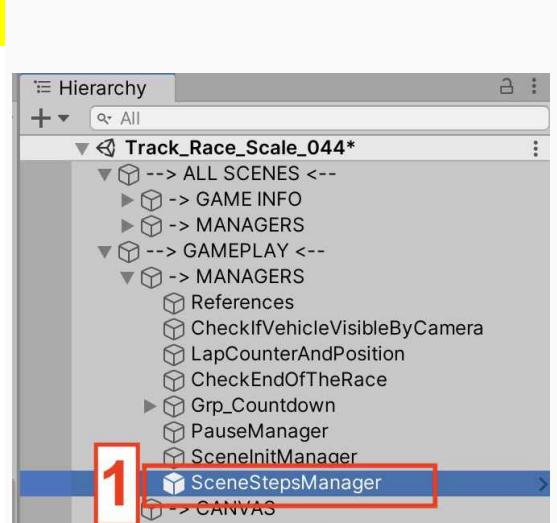
Modify a Step

It is possible to modify or use your own step. Methods called in a step must be boolean methods.

As an example we are going to remove the camera views displayed during the countdown in Time Trial Game Mode and replace place this step by a text displayed in the console tab.

-Open a gameplay scene.

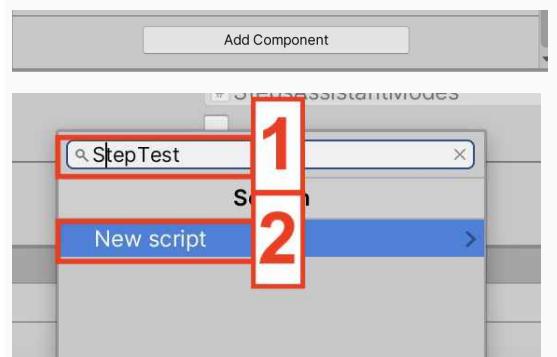
-In the Hierarchy select **SceneStepsManager** (spot 1)
(Hierarchy: GAMEPLAY → MANAGERS → SceneStepsManager)



-Press **Add Component** at the end of the Inspector scroll view.

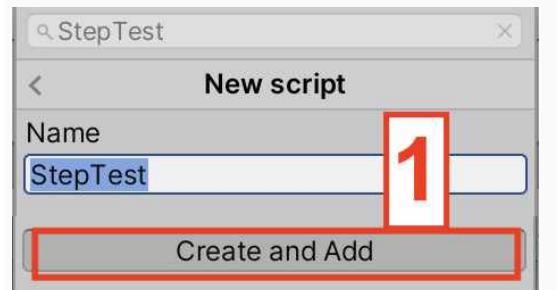
-Write **StepTest** in the search field. (spot 1)

-Press **New script** (spot 2).



-Press **Create and Add** (spot 1).

Wait until Unity compiles C# scripts



Double click on the script file to open the script in your script editor (spot 1).

Replace all the script with this:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class StepTest : MonoBehaviour
{
    public bool DisplayText()
    {
        Debug.Log("Custom Step");
        return true;
    }
}
```

A screenshot of the Unity Script Editor window. The code editor shows a single file named "Step Test (Script)". The code is as follows:

```
No selection
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class StepTest : MonoBehaviour
6  {
7      public bool DisplayText()
8      {
9          Debug.Log("Custom Step");
10         return true;
11     }
12 }
13
```

A red box highlights the entire code area, and a large white number "1" is overlaid on it.

IMPORTANT:

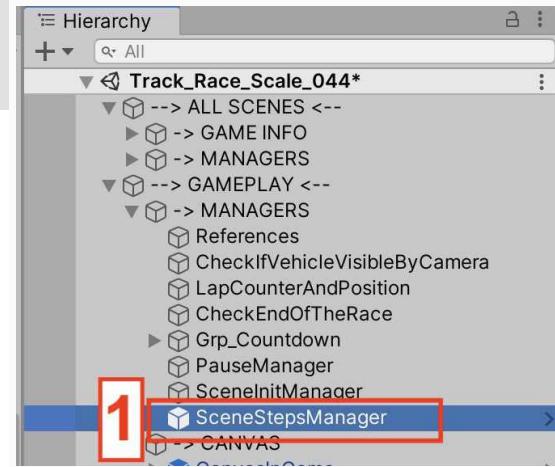
All the methods used in the **SceneStepsManager** must be boolean methods.

-Save the script (Ctrl+S)

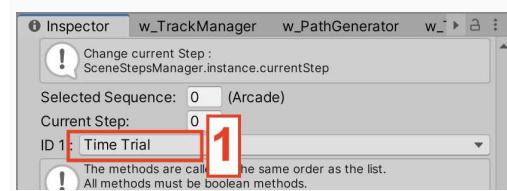
Return in Unity.

Wait until Unity compiles C# scripts

-In the Hierarchy select **SceneStepsManager** (spot 1)
(Hierarchy: GAMEPLAY → MANAGERS → SceneStepsManager)



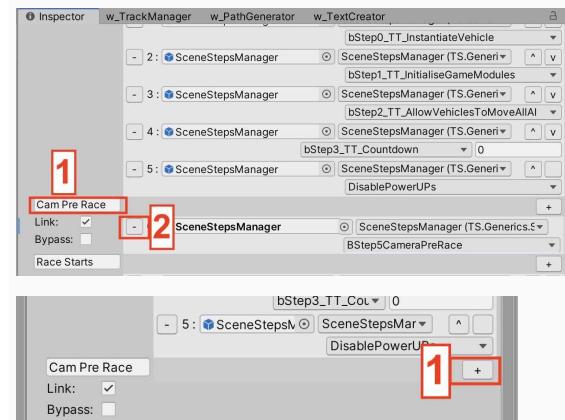
-In the Inspector select **Time Trial** in the dropdown menu (spot 1).



-Go to step **Cam Pre Race** (spot 1)

-Press **-** button (spot 2).

It removes the methods that displays camera views during the countdown.



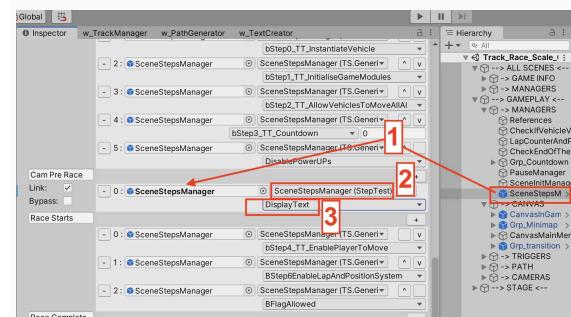
-Press **+** Button (spot 1)



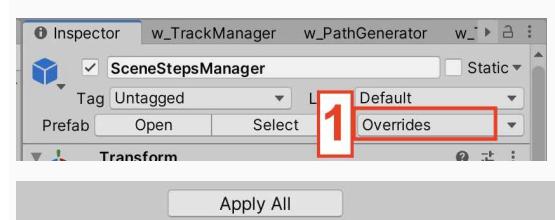
-Drag and drop **SceneStepsManager** in the empty field (spot 1).

-Select in the dropdown menu the script **StepTest** (spot 2)

-Select the method **DisplayText()** (spot 3).

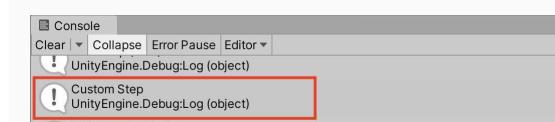


-On the top of the Inspector press **Overrides** button (spot 1)



-Press **Apply All**.

Now the player starts a race in Time Trial mode the text CurrentStep is displayed in the console tab.



Scripting: Useful methods to use the Scene Step System

Call the next step:

```
SceneStepsManager.instance.NextStep()
```

Call specific step:

```
SceneStepsManager.instance.NextStep(int StepSequence, int StepID)
```

Change current Step Sequence:

```
SceneStepsManager.instance.currentStepSequence
```

Change current Step:

```
SceneStepsManager.instance.currentStep
```

Loading screen

This section describes **CanvasLoadingManager** object that can be found in the Hierarchy.
(Hierarchy: ALL SCENES → MANAGERS → CanvasLoadingManager)

Table of contents:

Overview	Link
How to: Customize the loading screen	Link
Scripting	Link

Overview

The loading screen is called when a new scene is loaded (spot 1).

The Loading screen is automatically disabled after the initialization of a new scene.

-Loading screen is managed by the object **CanvasLoadingManager** (spot 2).

-Loading screen graphics are contained in object **CanvasLoading** inside **Grp_Page_Loading** (spot 3).

Note:

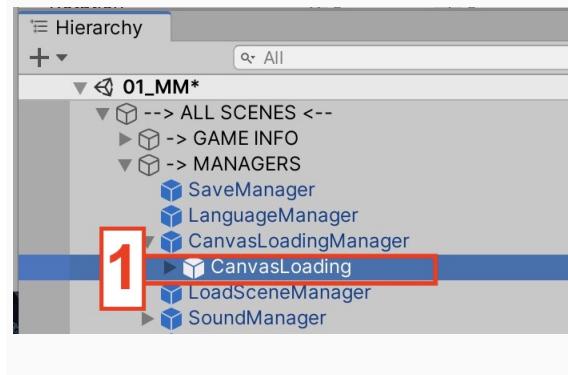
The objects **CanvasLoading** and **CanvasLoadingManager** are not destroyed when a new scene is loaded.



How to: Customize the loading screen

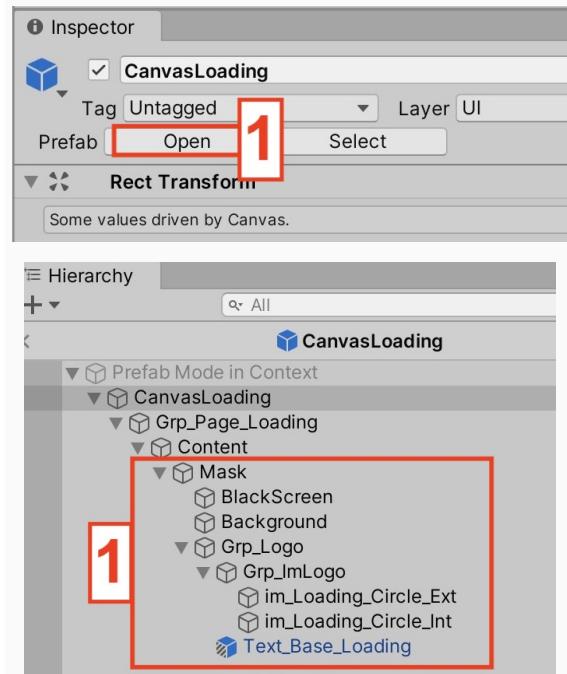
-In the Hierarchy select **CanvasLoading** (spot 1).

(Hierarchy: ALL SCENES → MANAGERS → CanvasLoadingManager → CanvasLoading)



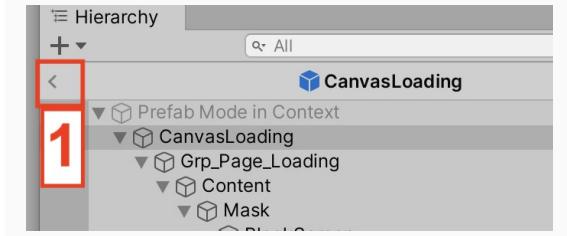
To edit the **CanvasLoading** prefab:

-In the Inspector press **Open** next to Prefab (spot 1)



When you have finished the modification:

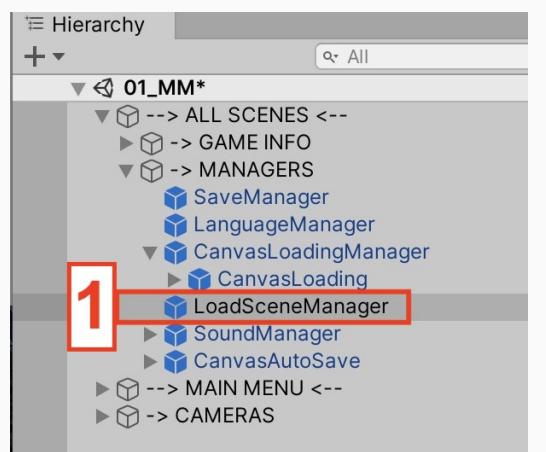
-In the Hierarchy press < button (spot 1) to close the prefab.



Scripting

By default, when a new scene is loaded:

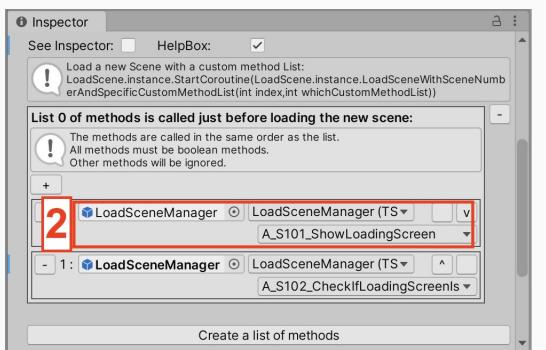
The object **LoadSceneManager** (spot 1)
calls the method **A_S101_ShowLoadingScreen()**
(spot 2).



By default the loading screen is automatically disabled after the scene initialization:

The Coroutine called is:

```
//-> Close Canvas Loading  
StartCoroutine(CanvasLoadingManager.instance.CloseCanvasLoading());
```



Canvas AutoSave

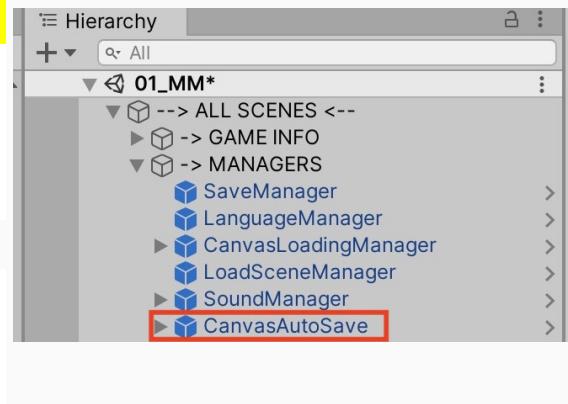
This section describes **CanvasAutoSave** object that can be found in the Hierarchy.
(Hierarchy: ALL SCENES → MANAGERS → CanvasAutoSave)
This object is used as a player feedback to indicate that the game is saving data.

Overview

By default **CanvasAutoSave** is displayed on screen:
-After the player wins rewards (Race Result page).
-After the player saves his name at the end of a Time Trial race.

Info:

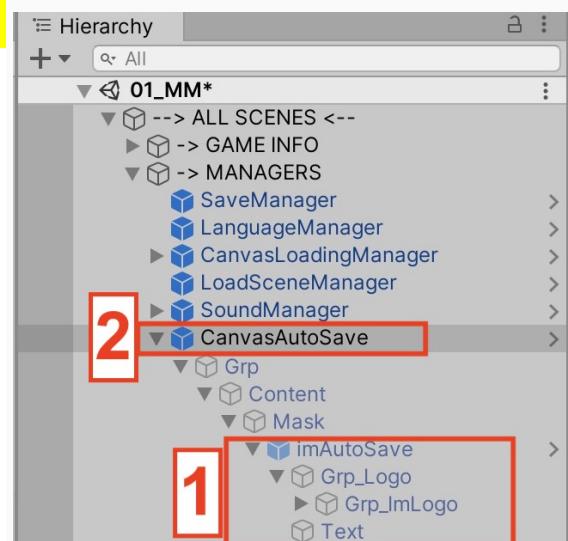
- 1- This object must be enabled or disabled manually.
- 2- This object is not destroyed when a new scene is loaded.



Modify Canvas AutoSave

- Add or Change objects included in **Mask** object (spot 1).

After the modifications:
Select **CanvasAutoSave** (spot 2).
In the Inspector press **Overrides** button and
press **Apply All** button to save the modification in all
scenes.



Scripting:

From any script:

To display the canvas call:
`CanvasAutoSave.instance.transform.GetChild(0).gameObject.SetActive(true);`

To disable the canvas call:
`CanvasAutoSave.instance.transform.GetChild(0).gameObject.SetActive(false);`

Transition System

Brief description:

This section describes the system to create transition

This system is managed into **Grp_Transition** object that can be found in:

(Main Menu scene) Hierarchy: MAIN MENU → CANVAS → Grp_Transition

(Gameplay scene) Hierarchy: GAMEPLAY → CANVAS → Grp_Transition

Table of content:

Overview	Link
How to: Modify a transition	Link
How to: Create a transition	Link
How to: Play a transition	Link

Overview

The transition system is separated into 2 parts:

-TransitionManager (spot 1):

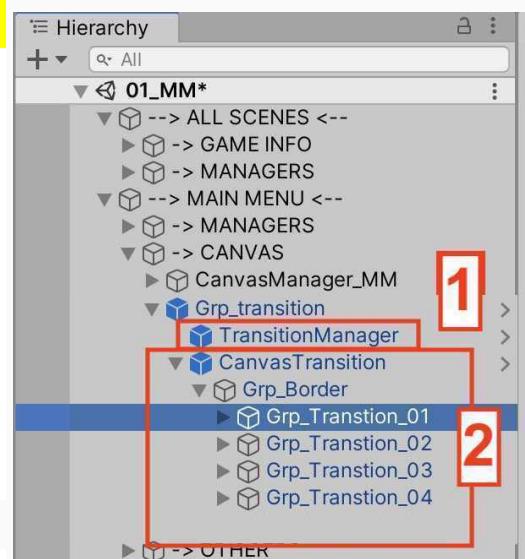
This object allows you to edit / create new transitions.
At runtime, the script **TransitionManager.cs** attached to the object is used to play transitions.

-CanvasTransition (spot 2):

This object contains the animations used for each transition.

Info:

The transition object is destroyed when a new scene is loaded. It allows you to create special transition in each scene.



How to: Modify a transition

To access existing transitions:

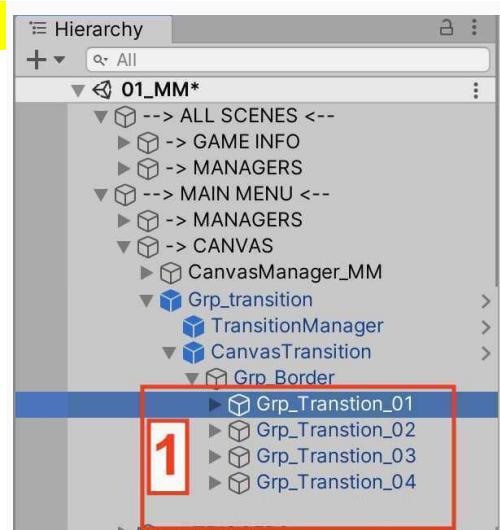
In the **Main Menu** Scene (spot 1):

-In the Hierarchy go to **MAIN MENU** → **CANVAS** → **Grp_Transition** → **Canvas Transition**
→ **Grp_Border** →

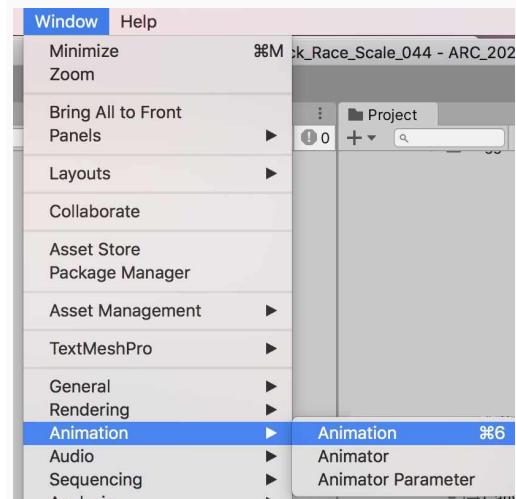
In a **Gameplay** Scene:

-In the Hierarchy go to **GAMEPLAY** → **CANVAS** → **Grp_Transition** → **Canvas Transition**
→ **Grp_Border** →

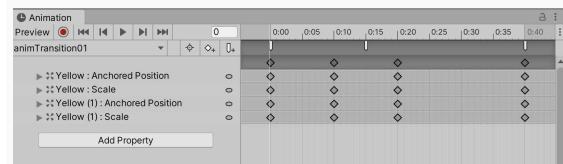
-As an example select **Grp_Transition_01** (spot 1)



-Open the **Animation** window
(**Window** → **Animation** → **Animation**)



Now in the **animation window** you can modify the animation.



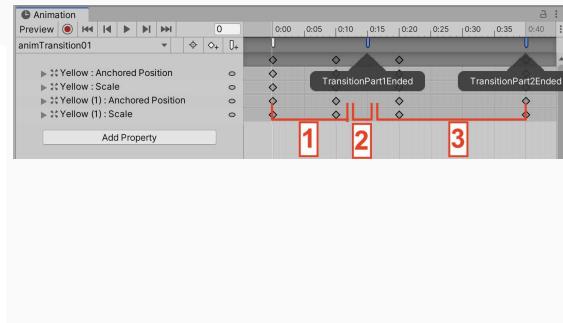
Info:

The animation is separated into 3 parts.

1st part (spot 1): The animation fade out the screen.

2nd part (spot 2): The event **TransitionPart1Ended** is called. When this event is called the transition is paused until the new UI page is displayed on screen.

3rd part (spot 3): The animation fade in the screen.

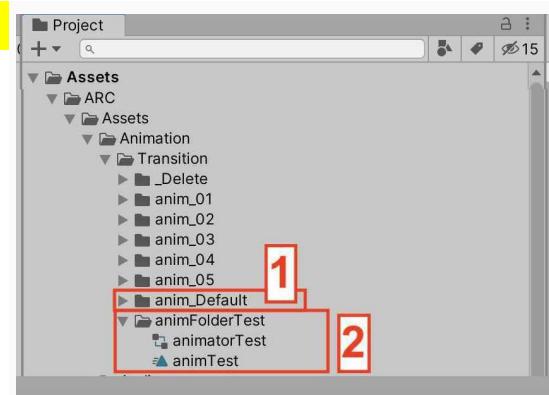


How to: Create a transition

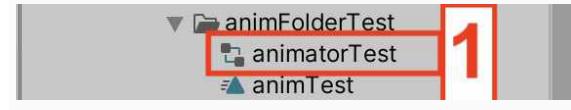
As an example we are going to create a transition in the main menu when the page Choose Mode is displayed.

-In Project Tab select the folder **anim_Default** (spot 1)
(Assets → Animation → Transition)

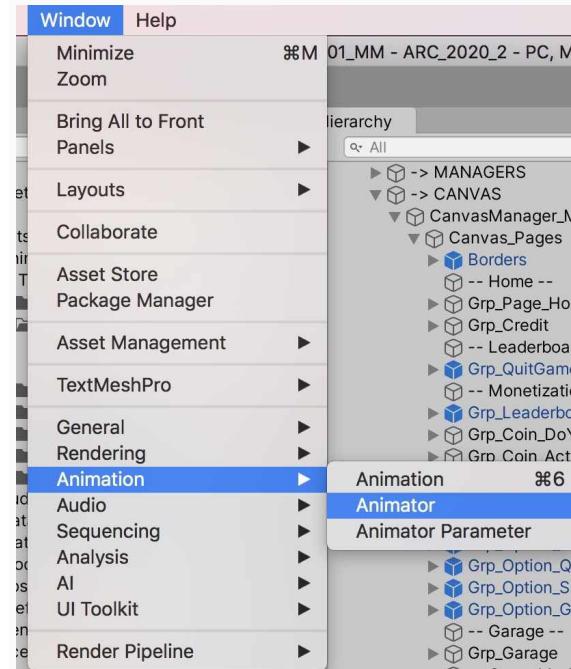
-Duplicate folder **anim_Default** and rename (spot 2):
The folder: **animFolderTest**
The animation: **animTest**
The Animator: **animatorTest**



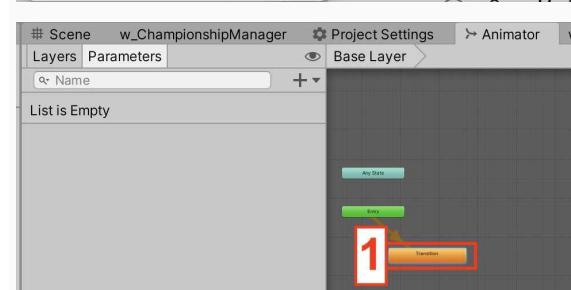
-In Project Tab select **animatorTest** (spot 1).



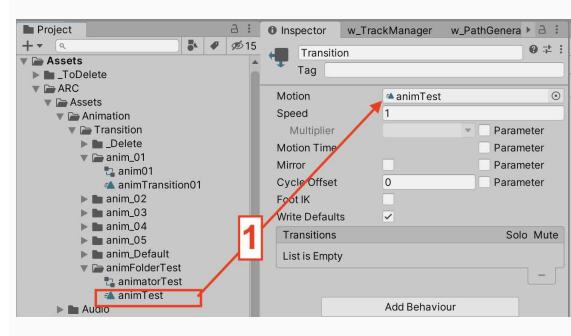
-Open **Animator window** (Window → Animation → Animator)



-In **Animator Window** click on **Transition** button (spot 1).



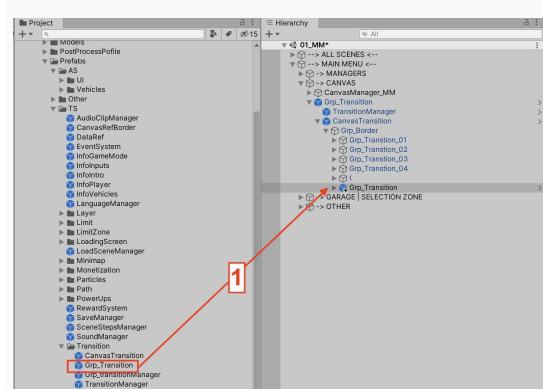
-Drag and drop **animTest** to the **motion field** (spot 2).



-In Project Tab open the **Main Menu** scene.
(Assets → Scenes → MainMenu → 01_MM)

-From Project tab:
Drag and drop **Grp_Transition**
(Assets → Prefabs → TS → Transition → Grp_Transition)

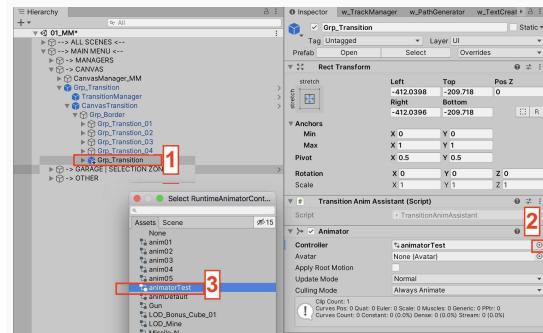
at the end of **Grp_Border** object in the Hierarchy
(spot 1)



-In the Hierarchy select **Grp_Transition** (spot 1)

-In the Inspector press the **circle** (spot 2)

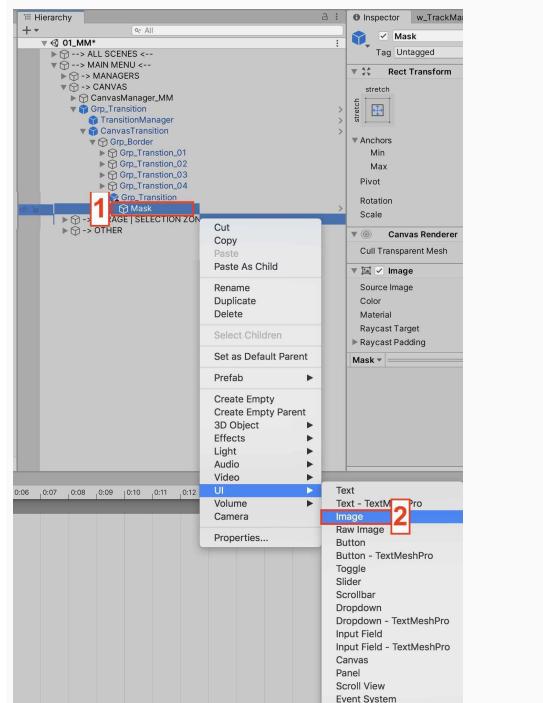
-In the list select **animatorTest** (spot 3).



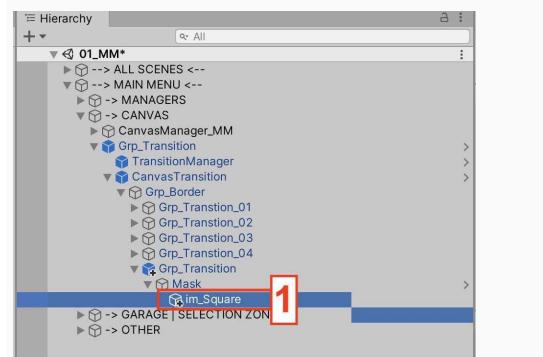
Now we are going to create the animation itself.

-Select **Mask** object (spot 1).

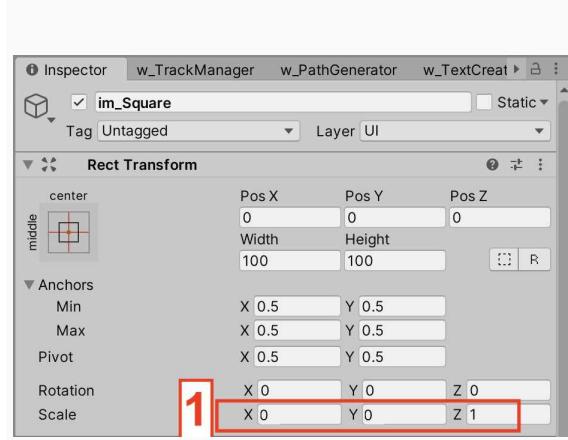
-Right click on the object and go to **UI → Image**.



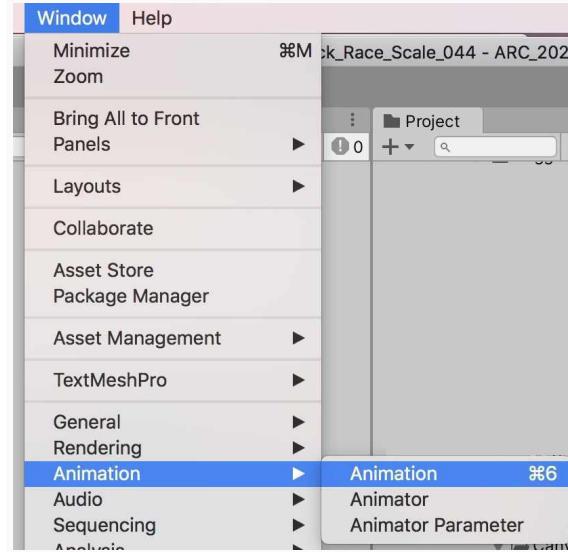
-Rename the UI image **im_Square** (spot 1).



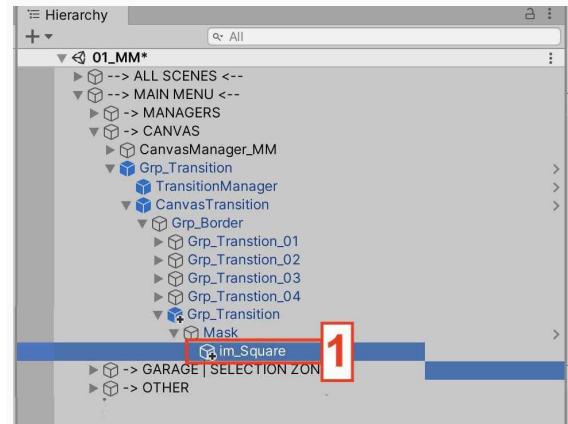
-In the Inspector change the scale to (spot 1):
 X = 0 | Y = 0 | Z = 1



-Open the Animation window
 (Window → Animation → Animation)

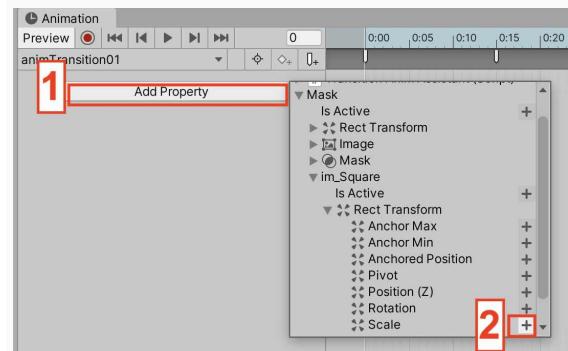


-In the Hierarchy select im_Square.
 (MAIN MENU → CANVAS → Grp_Transition → CanvasTransition
 → Grp_Border → Grp_Transition → Mask → im_Square)

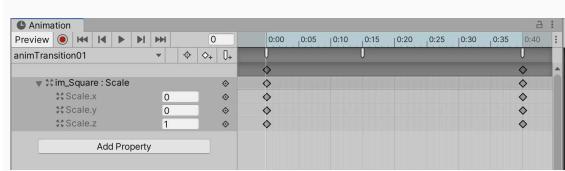


-In the Animation Window press Add Property
 (spot 1).

-Go to Mask → im_Square → Rect Transform → Scale (spot 2)



The new property is created in the Animator Window.



-Click on frame 15 (spot 1).

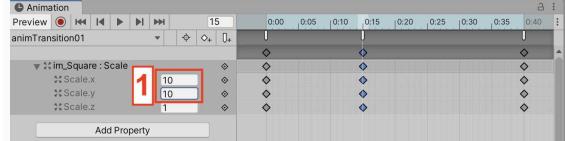
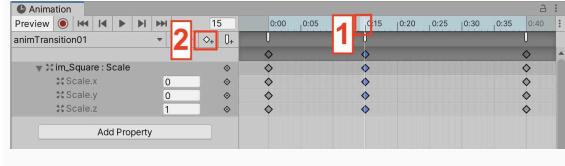
-Press Add Keyframe button (spot 2).

A new keyframe is created on frame 15.

-Change the values (spot 1):

Scale X to 10

Scale Y to 10



Now we are adding the new animation to the Transition system.

-In the Hierarchy select **TransitionManager** (spot 1).

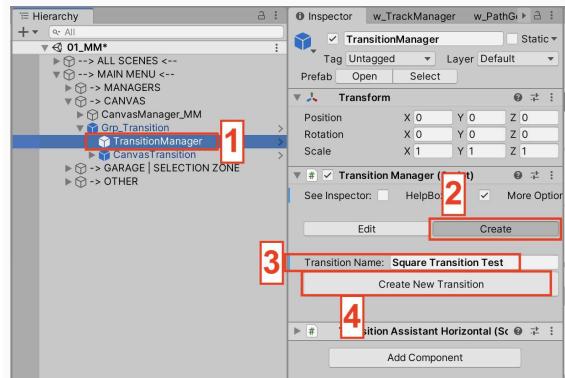
(MAIN MENU → CANVAS → Grp_Transition → TransitionManager)

-In the Inspector press **Create** button (spot 2).

-For this example choose the name

Square Transition Test for the new transition name (spot 3).

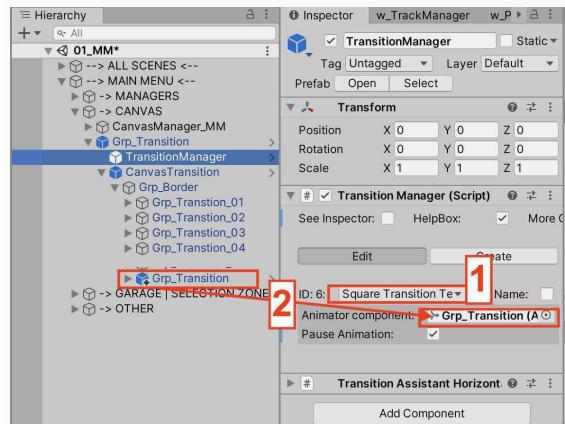
-Press **Create New Transition** button to create the new transition (spot 4).



The transition is automatically selected (spot 1)

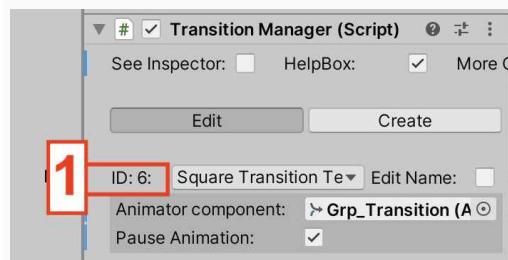
-Drag and drop **Grp_Transition** object in the empty field (spot 2).

Now it is possible to use the transition in the game.



Now we are using the new transition when the player opens the UI page to choose the game mode.

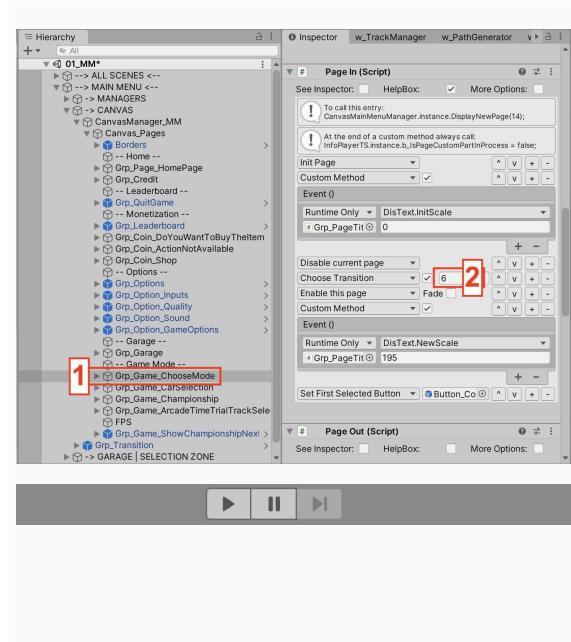
Our transition as the **ID = 6** (spot 1)



-In the Hierarchy select **Grp_Game_ChOOSEMode** (spot 1).

(MAIN MENU → CANVAS → Grp_CanvasManager_MM → Canvas_Page → Grp_Game_CHOOSEMode)

-In the Inspector go to **Page_In (Script)** and change Choose Transition to 6 (spot 2).



-Press **Play** button to start the game.

Now the new transition is played when the page CHOOSE Mode is opened.

Info:

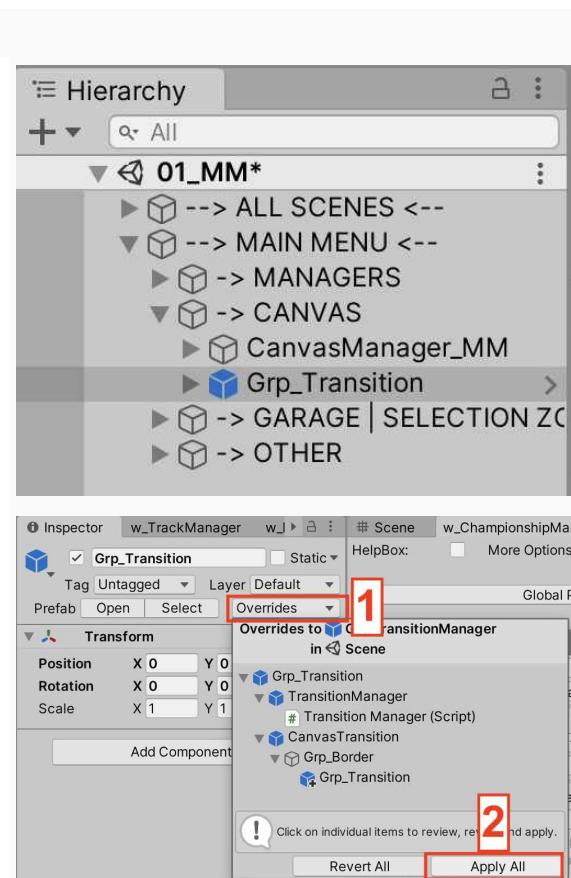
If you want to add this new transition in all scenes:

-In the Hierarchy select **Grp_Transition**.

(MAIN MENU → CANVAS → Grp_Transition)

-In the Inspector press **Overrides** button (spot 1).

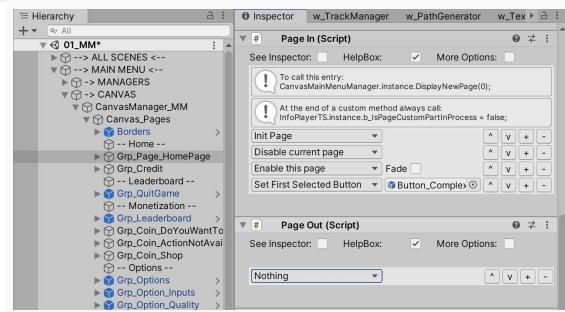
-Press **Apply All** (spot 2).



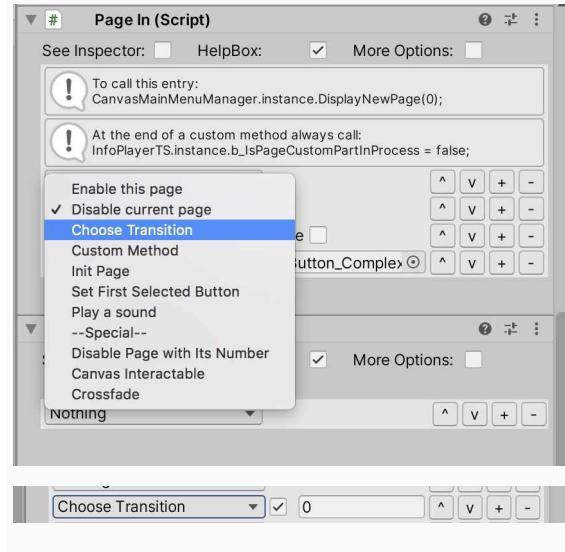
How to: Play a transition

Case 1: Play transition when a UI page is opened or closed

In each UI Page, it is possible to play a transition using **Page In** or **Page Out** script.



In the dropdown menu select **Choose Transition**.



Choose the **value** corresponding to the **ID** of the transition you want to play.

Case 2: Play transition (scripting)

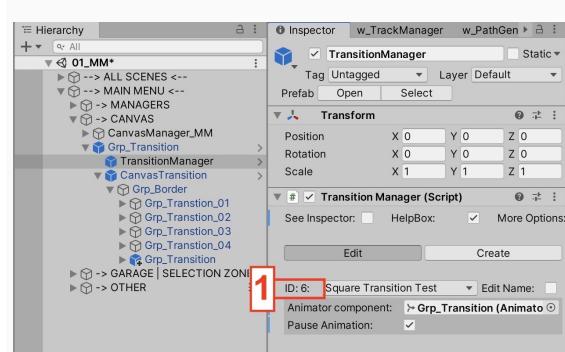
From any script call:

```
StartCoroutine(TransitionManager.instance.Transition(value, true));
```

Value corresponds to your the Transition ID.

To find the transition ID:

In the Hierarchy select **TransitionManager** (spot 1).
(MAIN MENU → CANVAS → Grp_Transition → TransitionManager)



Countdown system

This section describes **Grp_Countdown** object. This object is only available in gameplay scenes.
(Hierarchy: GAMEPLAY → MANAGERS → Grp_Countdown)
During the game this group of objects managed the countdown before the race starts.

Table of contents:

Overview	Link
How to: Disable camera movement during countdown	Link
UI Countdown manager (overview)	Link
How to: Delete Countdown UI elements	Link
How to: Add Countdown UI elements	Link

Overview

-**Countdown** managed the UI elements displayed during the countdown (spot 1).
(Hierarchy: GAMEPLAY → MANAGERS → Grp_Countdown → Countdown)

-**CamDuringCountdown** managed the position of the player camera during the countdown (spot 2).
(Hierarchy: GAMEPLAY → MANAGERS → Grp_Countdown → CamDuringCountdown)

By default, countdown is called by **SceneStepManager** object (spot 3) in mode:

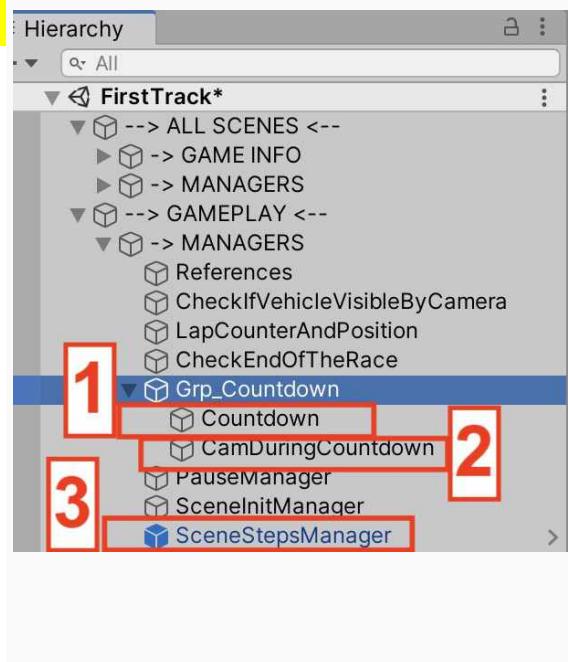
(Hierarchy: GAMEPLAY → MANAGERS → SceneStepsManager)

0: Arcade

1: Time Trial

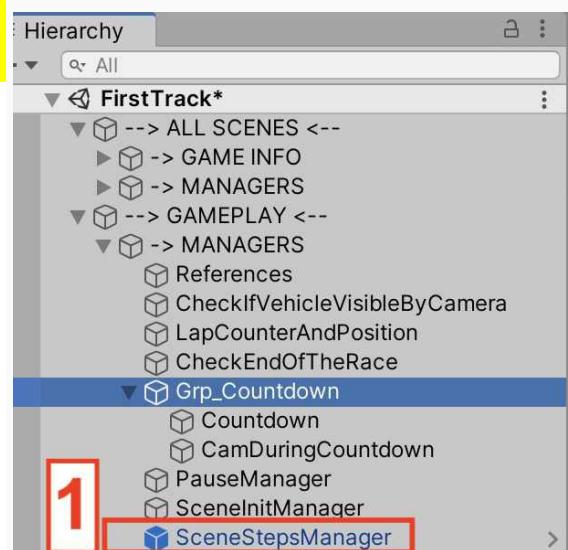
2: Championship

3: Test Track P1 + AI



How to: Disable camera movement during countdown

In the Hierarchy select **SceneStepsManager** (spot 1)
(Hierarchy: GAMEPLAY → MANAGERS → SceneStepsManager)

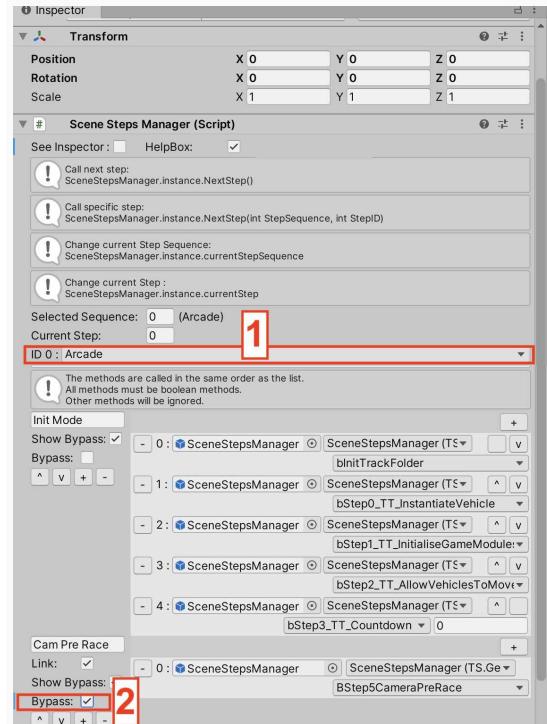


(As an example we are going to disable camera movement in Arcade Mode.)

-In the Inspector select **Arcade mode** (spot 1)

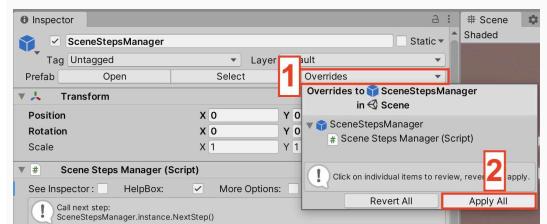
-In section **Cam Pre Race** check **Bypass** button (spot 2)

*(Now the camera remains fixed behind the player.
 Repeat the process for any mode you want to disable the camera movement during the countdown)*



-In the inspector press **Overrides** button (spot 1).
 -Press **Apply All** button (spot 2).

(The modification is applied in all scenes)



UI Countdown manager (overview)

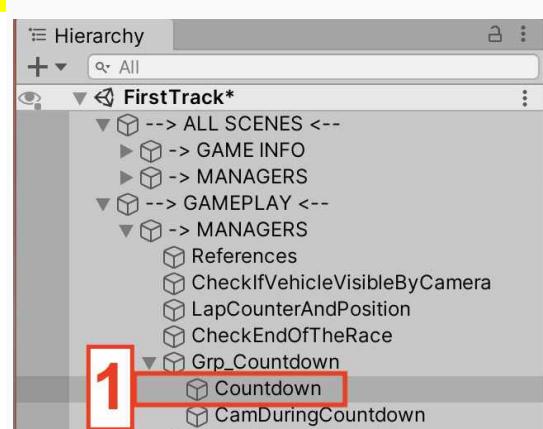
Countdown managed the UI elements displayed during the countdown (spot 1).

(Hierarchy: GAMEPLAY → MANAGERS → Grp_Countdown → Countdown)

By default the countdown displays: 3, 2 , 1, Go

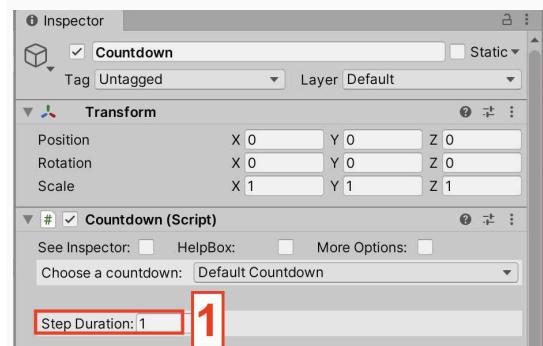
It is possible to modify and create your own countdown. In this section we are going to:

- See the different countdown manager sections.
- See how to delete or add UI elements used during the countdown.



The 1st section (step 1) allows to choose the duration between 2 countdown steps.

By default Step Duration = 1 second.



The 2nd section allows to call events for each step.

By default

Step 0 call methods to:

For Player 1:

- Display the text 3 (spot 1a)
- Start a sprite animation (spot 1b)

For Player 2:

- Display the text 3 (spot 2a)
- Start a sprite animation (spot 2b)

Step 01 call methods to:

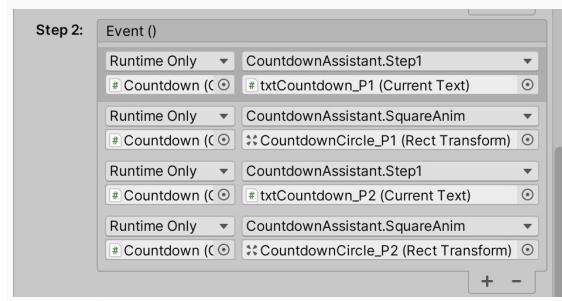
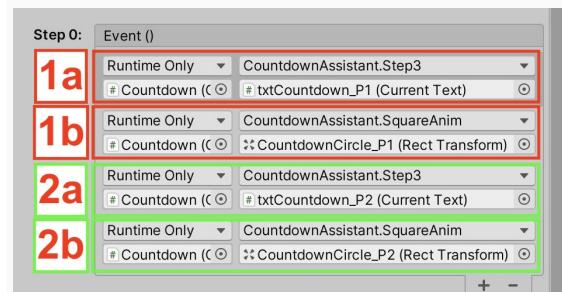
For Player 1:

- Display the text 2.
- Start a sprite animation.

For Player 2:

- Display the text 2.
- Start a sprite animation.

...



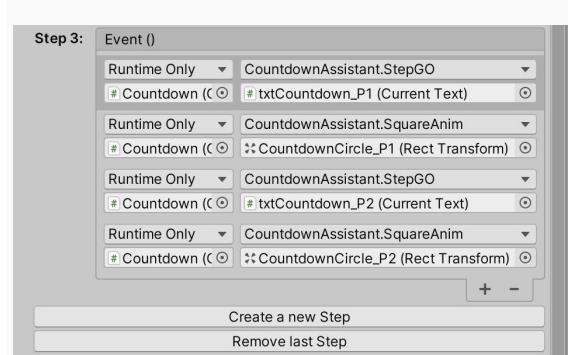
Step 3 call methods to:

For Player 1:

- Display the text GO.
- Start a sprite animation.

For Player 2:

- Display the text GO.
- Start a sprite animation.



The 3rd section allows to call methods when the countdown ended.

By default, when the countdown ended, a method is called to start the next step in the SceneStepManager.

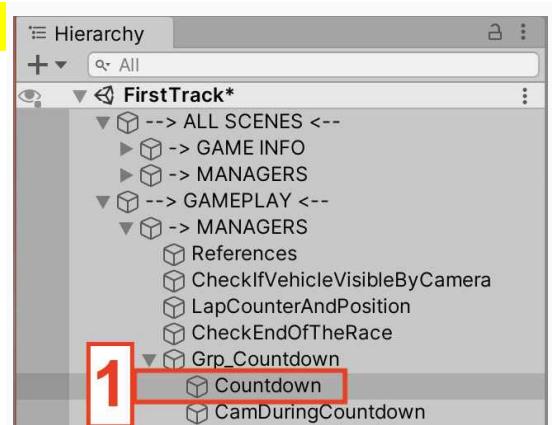


How to: Delete countdown UI elements

As an example we are going to remove the sprite animation during the first step

In the Hierarchy select **Countdown** (spot 1)

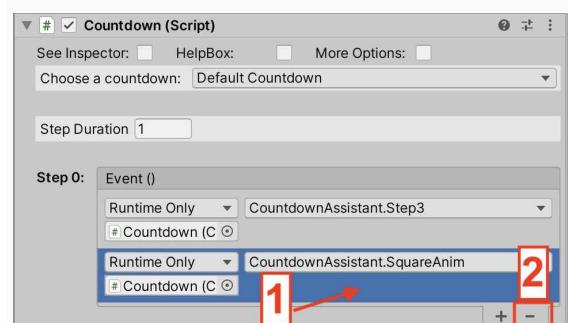
(Hierarchy: GAMEPLAY → MANAGERS → Grp_Countdown → Countdown)



-In the Inspector click on method **CountdownAssistant.SquareAnim** (spot 1)

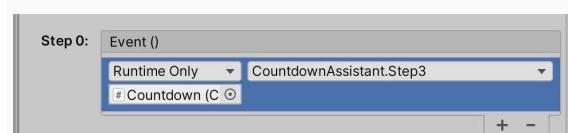
The section turns blue.

-Press **-** button to remove the method (spot 2).



The method is removed.

Now when you play the game using mode Arcade, Time Trial or Championship the sprite animation is not displayed during the first step.



How to: Add Countdown UI elements

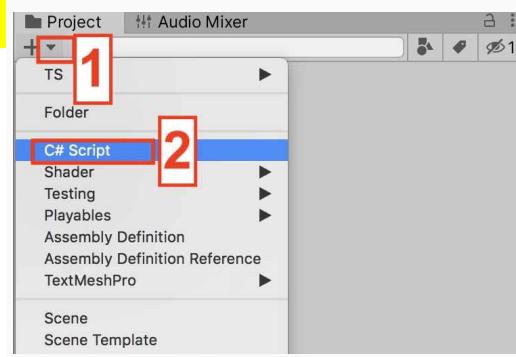
As an example we are going to display on screen the text **Let's Go** instead of **Go**

-We are going to create a method that displays Let's Go on screen.

-Then we will use this method on the last step of the countdown.

-In Project Tab press the small triangle (spot 1)

-Select **C# Script** in the dropdown menu (spot 2).



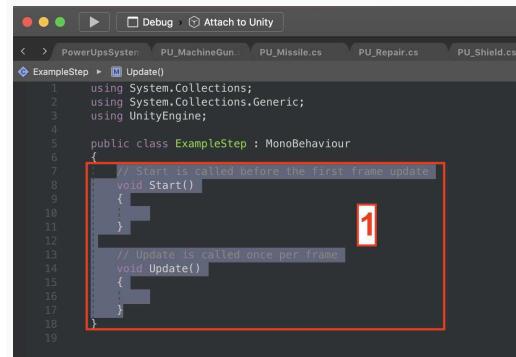
A new script is created.

-For this example rename it **ExampleStep**.

-Double click on the file to open it in your script editor.



-Remove **Start()** and **Update()** methods. We don't need it (spot 1).



Line 6 copy and paste (spot1):

```
//-> Access script with namespace  
using TS.Generics;
```



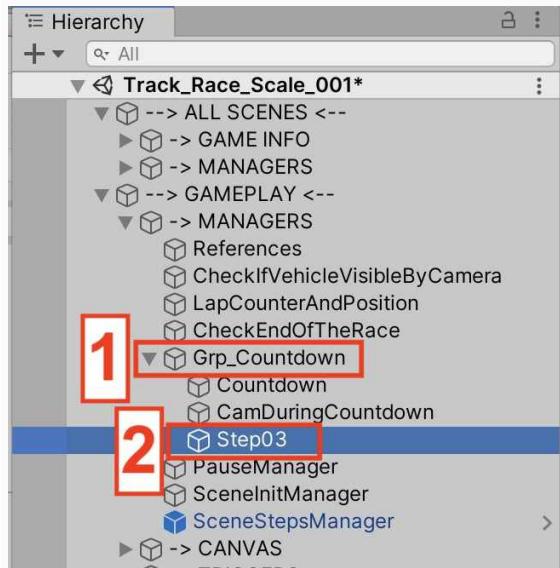
Line 10 copy and paste (spot 2):

```
public void StepThree(){  
    //-> Access The text object used to display the countdown  
    CurrentText objText =  
    CanvasInGameUIRef.instance.listPlayerUIElements[2].listTexts[0]  
;  
    //-> Use the method DisplayTextComponent contained in  
    CurrentText.cs to display the text  
    objText.DisplayTextComponent(objText.gameObject,  
    LanguageManager.instance.String_ReturnText(0, 166));}
```



-Go back to Unity.

-In the Hierarchy create inside **Grp_Countdown** (spot 1) (*Hierarchy: GAMEPLAY → Grp_Countdown*) an **Empty GameObject** and rename it **Step03** (spot 2)



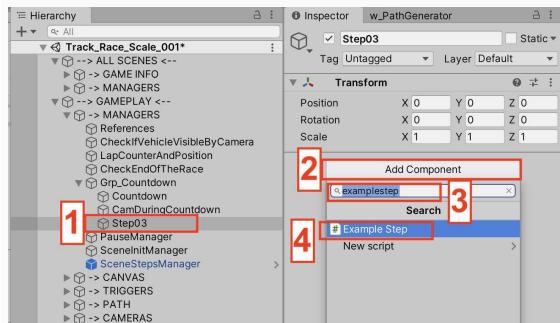
In the Hierarchy select **Step03** object (spot 1).

(*Hierarchy: GAMEPLAY → MANAGERS → Grp_Countdown → Step03*)

In the Inspector press **Add Component** button (spot 2)

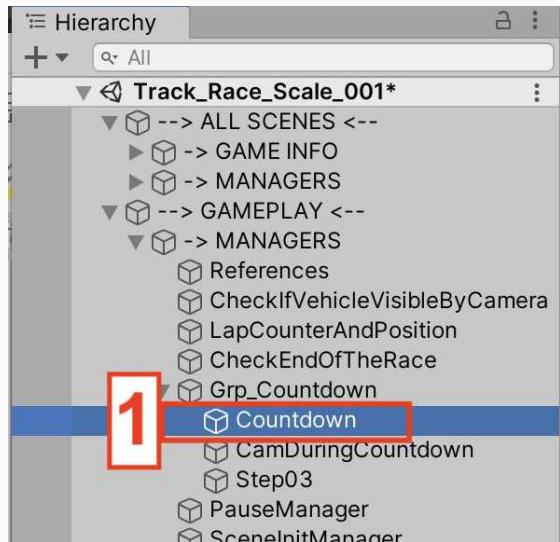
-In the search field write **ExampleStep** (spot 3)

-Select the script in the list (spot 4)



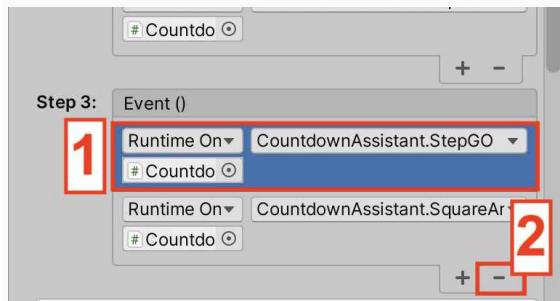
-In the Hierarchy select **Countdown** (spot 1).

(*Hierarchy: GAMEPLAY → MANAGERS → Grp_Countdown → Countdown*)



-In the Inspector select method **StepGO** (spot 1)

-Press **-** button (spot 2) to remove the method.

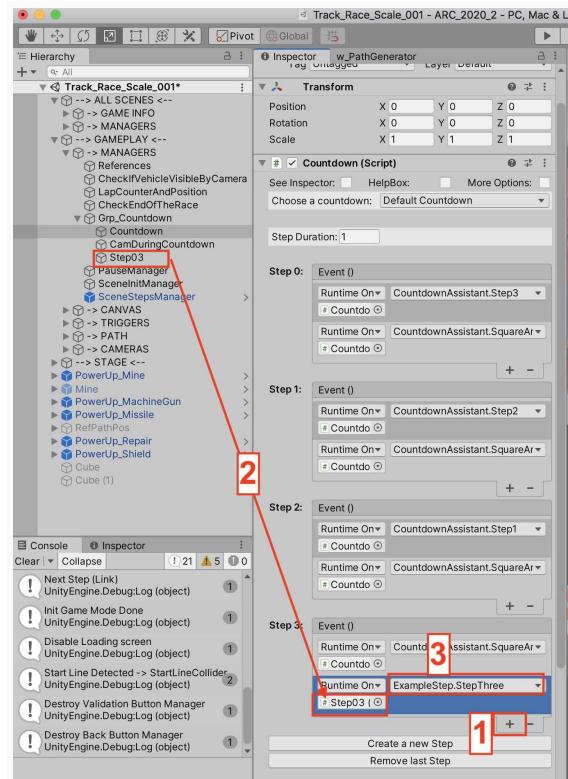


-Press + button (spot 1) to add a new method in step 3.

-Drag and drop Step03 object in the empty field (spot 2)

-In the dropdown menu select the script ExampleStep then the method StepThree (spot 3).

Now the countdown displays *Let's Go* on Step 3.



Congratulation screen

This section describes the system to display congratulation at the end of the race.

Table of contents:

Overview

[Link](#)

How to: Modify Congratulation screen

[Link](#)

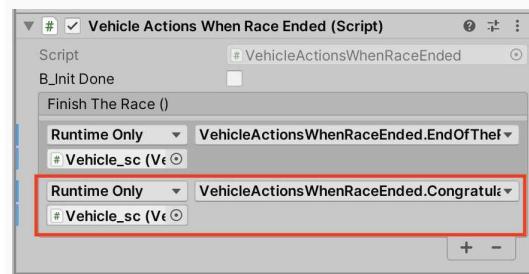
Overview

By default, when a player finishes a race the congratulation screen is displayed.



The congratulation screen is called by **VehicleActionsWhenRaceEnded** script attached to each vehicle.

(00_Vehicle_Base_sc → Vehicle_sc)



Congratulation sequence is managed by:

-**Congratulation.cs** on **Congratulation_P1_sc** object for Player 1.

(GAMEPLAY → CANVAS → CanvasInGame → Grp_P1 → Grp_Congratulation_P1 → Congratulation_P1_sc)

-**Congratulation.cs** on **Congratulation_P2_sc** object for Player 2.

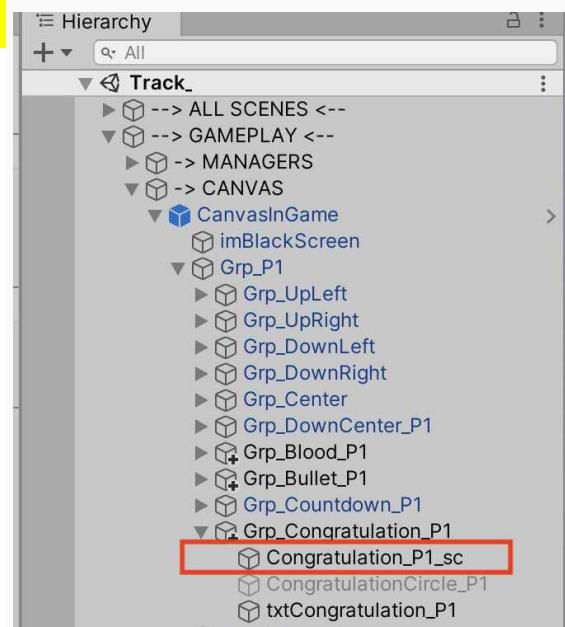
(GAMEPLAY → CANVAS → CanvasInGame → Grp_P2 → Grp_Congratulation_P2 → Congratulation_P2_sc)



How to: Modify Congratulation screen

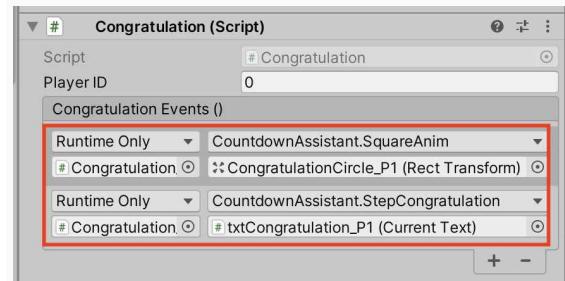
This section explains how to modify the default congratulation screen.

- In the Inspector select **Congratulation_P1_sc** object.
(GAMEPLAY → CANVAS → CanvasInGame → Grp_P1 → Grp_Congratulation_P1 → Congratulation_P1_sc)



- In the Inspector removes methods inside **Congratulation Events()**.

- Add your own methods that displays the congratulation screen.



IMPORTANT:

It is necessary to do the modification for the second player too.

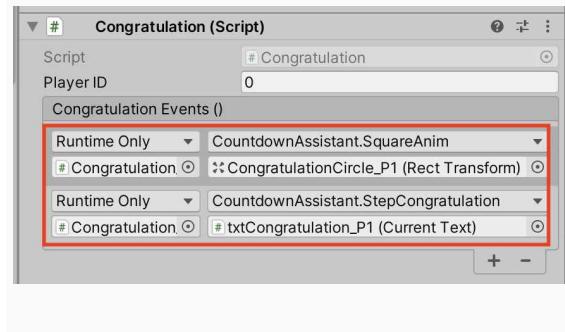
This section explains how to modify the default congratulation screen.

- In the Inspector select **Congratulation_P2_sc** object.
(GAMEPLAY → CANVAS → CanvasInGame → Grp_P2 → Grp_Congratulation_P2 → Congratulation_P2_sc)



- In the Inspector removes methods inside **Congratulation Events()**.

- Add your own methods that displays the congratulation screen.



Tutorial Manager

This section describes **TutoManager_01** object that can be found in the Hierarchy in each gameplay scene.

(Hierarchy: GAMEPLAY → MANAGERS → Grp_Tutos → TutoManager_01)

This object displays popups to explain which buttons are used to trigger booster and Power-Up. Popups are displayed only when the player participates to his first race.

Table of contents:

Overview

[Link](#)

Tutorial Manager Parameters

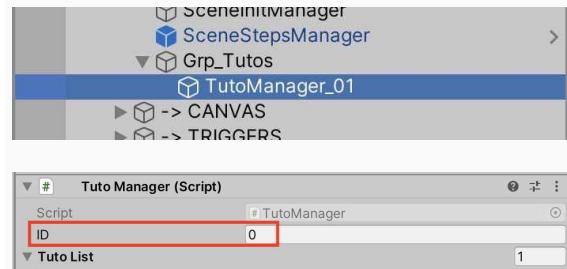
[Link](#)

Overview

The tutorial is managed by **TutoManager_01** with **TutoManager.cs** script attached to it.

(Hierarchy: GAMEPLAY → MANAGERS → Grp_Tutos → TutoManager_01)

To allow the possibility to create more than one tutorial, each tutorial must have a unique **ID**.



The tutorial is initialized by the **SceneStepsManager**



The Tutorial is initialized by the method **bInitTuto**.



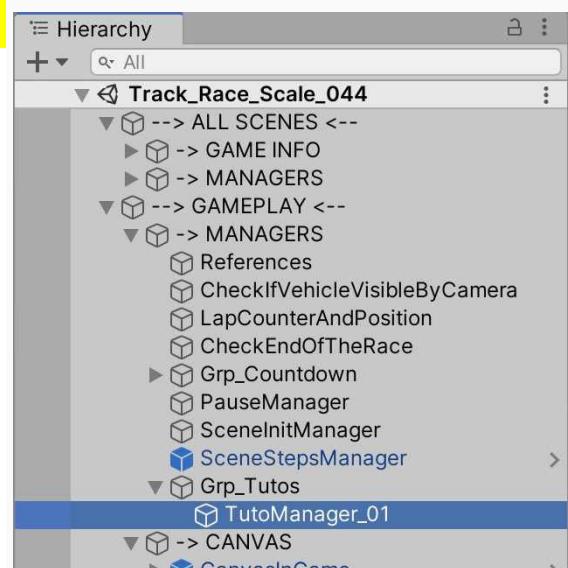
The parameter corresponds to the tutorial **ID**.

The tutorial is used in Mode:
Arcade, Time Trial, Championship, Test Track P1 +
AIs

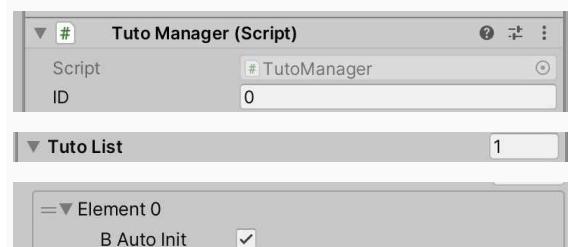
Tutorial Manager Parameters

The tutorial is managed by **TutoManager_01** with **TutoManager.cs** script attached to it.

(Hierarchy: GAMEPLAY → MANAGERS → Grp_Tutos → TutoManager_01)



Each Tutorial must have a unique **ID**.

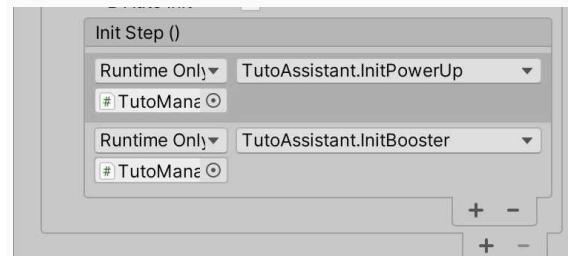


BAutoInit:

True: The methods are automatically initialized when the tutorial starts.

False: The methods to initialized the tutorial must be called manually. It is usefull if there is more than one step in the tutorial.

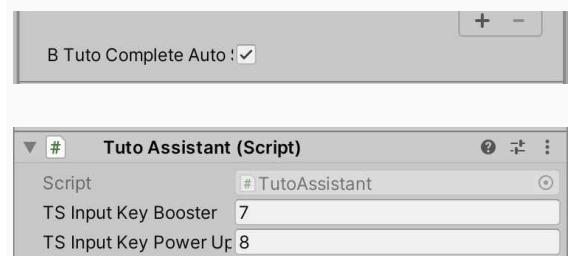
Drag and drop methods to initialize the tutorial.



BtutoCompleteAuto:

True: Automatically remember that the tutorial has been complete.

TutoAssistant.cs contains the methods to managed the tutorial behavior.



Info:

By default, Booster input is displayed on screen when booster is available. The input is disabled when the player presses 4 times the button.

Power-up input is displayed on screen when a power-up is selected by the player (4 times).

Audio

This section describes how audio is managed in the asset. Audio is managed in the Hierarchy by the object **SoundManager**. Volume can be modify by the player in the menu page **Grp_Option_Sound**. This menu page is managed in the Hierarchy by the object **SoundUIManager**.

Table of contents:

Overview	Link
Scripting: Load and Save AudioMixer Volumes	Link
How to: Integrate a music, an ambiance or Sfx	Link
Scripting: Useful methods to Access/ Play/ Stop a music, an ambiance or Sfx	Link
Spatialized Sounds	Link
How to: Delete/Reset Audio Mixer Data	Link
How to: Set the default Audio Mixer values	Link
How to: Modify default audio sound fx (<i>Engine, Booster, Hit, Warning, Power up, ...</i>)	Link

Overview

-In the Hierarchy the object **SoundManager** allows to manage non spatialized sound (Music, Ambiance (global) and UI Sfx) (spot 1):
(Hierarchy: ALL SCENES → MANAGERS → SoundManager)

-In the Hierarchy the object **AudioSources** contains all the AudioSources used to play Music, Ambiance and UI Sfx (spot 2). (Hierarchy: ALL SCENES → MANAGERS → SoundManager → AudioSources)

-In the Hierarchy the object **AudioClipManager** contains the lists of music, ambiance and UI Sfx used in the project (spot 3). (Hierarchy: ALL SCENES → MANAGERS → SoundManager → AudioClipManager)



-In the Hierarchy the object **SoundUIManager** allows the player to change volume in the game options.

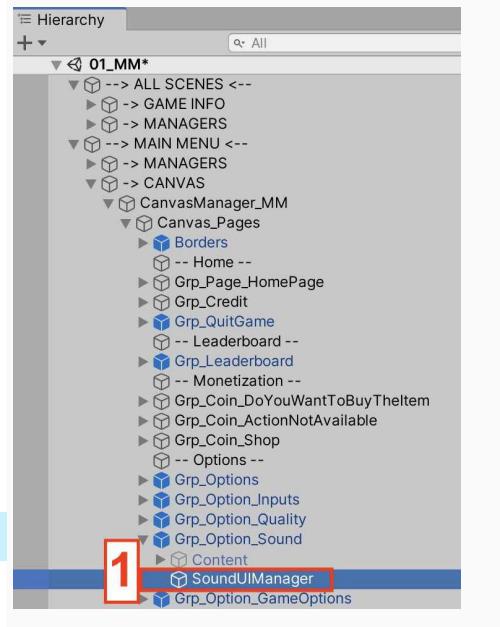
In a gameplay scene:

(Hierarchy: GAMEPLAY → CANVAS → CanvasManager_TM → Canvas_Pages → Grp_Option_Sound_IG → SoundUIManager)

In the Main Menu scene (01_MM):

(Hierarchy: MAIN MENU → CANVAS → CanvasManager_MM → Canvas_Pages → Grp_Option_Sound → SoundUIManager)

More info about spatialized sound → [Link](#)



Scripting: Load and Save AudioMixer Volumes

The object **SoundManager** allows to load and save volumes (Master, Music, Ambiance, Sfx).

By default:

- Volumes are loaded when the game starts.
- Volumes are saved if the player change a volume in the menu page [Grp_Option_Sound](#).

Info:

The object **SoundManager** is not destroyed when a scene is loaded. It is possible to call the SoundManager from any script.

Load AudioMixer volumes:

```
SoundManager.instance.Bool_LoadMixerValues()
```

Save AudioMixer volumes:

```
SoundManager.instance.Bool_SaveMixerValues()
```



How to: Integrate a music, an ambiance or Sfx

The next 3 sections explain how to set a sound that can be used at runtime. Read the next chapter if you want to know how to access audio by script.

Table of contents:

How to: Integrate a new music	Link
How to: Integrate a new ambiance	Link
How to: Integrate a new Sfx	Link

How to: Integrate a new music

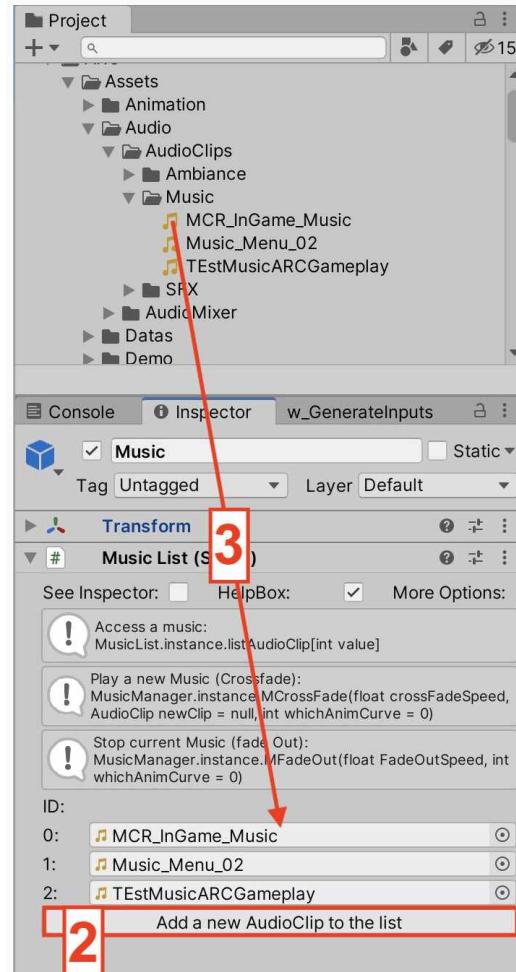
-In the Hierarchy select Music (spot 1).

(Hierarchy: ALL SCENES → MANAGERS → SoundManager → AudioClipManager → Music)



-In the Inspector press the button Add new AudioClip to the list (spot 2).

-Drag and drop a music file in the empty field (spot 3)



How to: Integrate a new ambiance

-In the Hierarchy select **Ambiance** (spot 1).

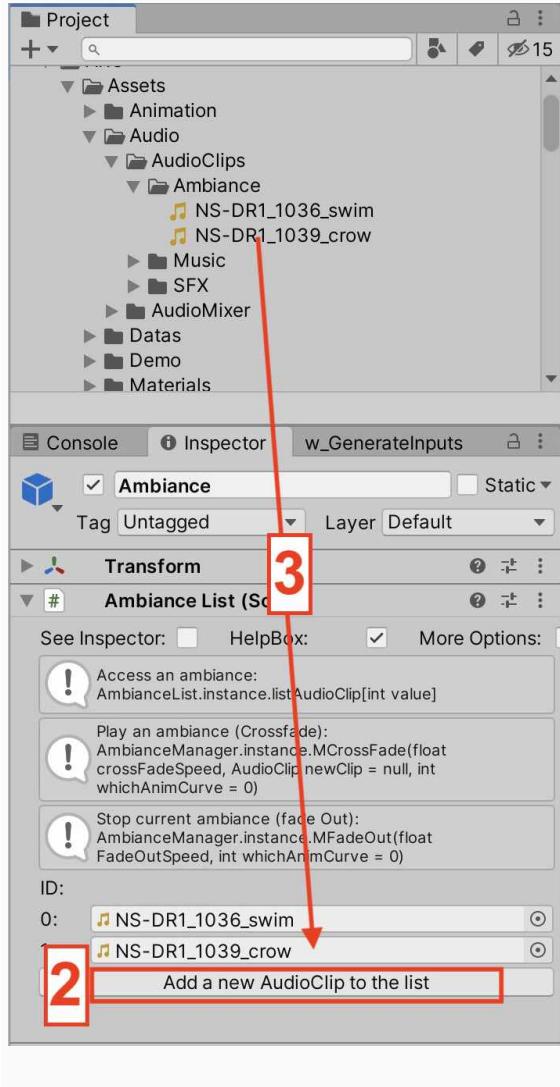
(Hierarchy: ALL SCENES → MANAGERS → SoundManager → AudioClipManager → Ambiance)



-In the Inspector press the button

Add new AudioClip to the list (spot 2).

-Drag and drop an ambiance file in the empty field
(spot 3)



How to: Integrate a new Sfx

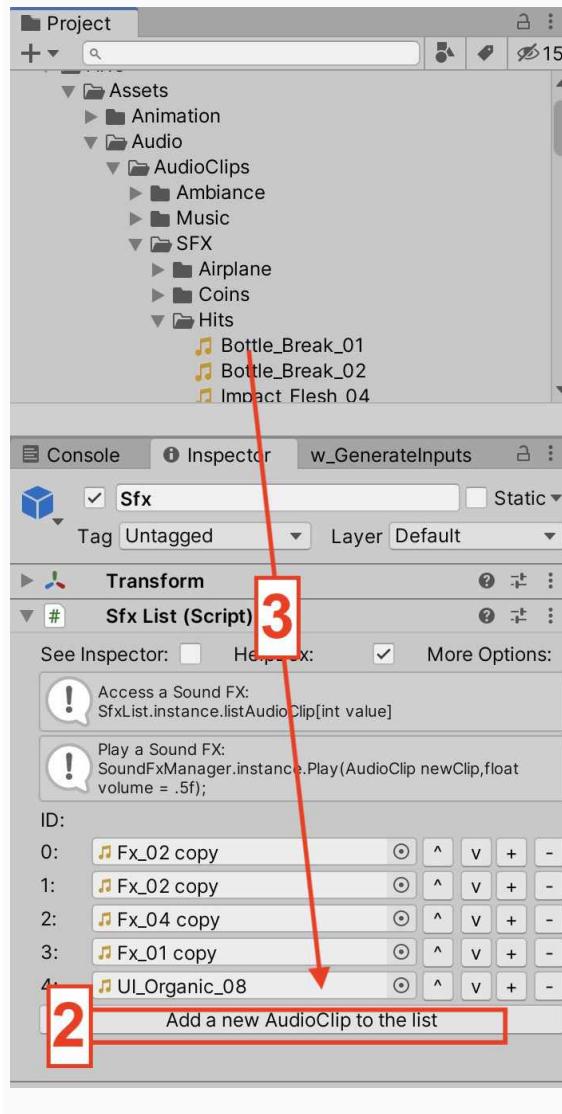
-In the Hierarchy select Sfx (spot 1).

(Hierarchy: ALL SCENES → MANAGERS → SoundManager → AudioClipManager → Sfx)



-In the Inspector press the button
Add new AudioClip to the list (spot 2).

-Drag and drop Sfx file in the empty field
(spot 3)



Scripting: Useful methods to Access/ Play/ Stop a music, an ambiance or Sfx

The next 3 sections explain how to access a sound at runtime.

How to: Access, play or stop a music

Access a music:

```
MusicManager.instance.listAudioClip[int value]
```

Play a new Music (Crossfade):

```
MusicManager.instance.MCrossFade(  
float crossFadeSpeed,  
AudioClip newClip = null,  
int whichAnimCurve = 0)
```

Stop current Music (fade Out):

```
MusicManager.instance.MFadeOut(  
float FadeOutSpeed,  
int whichAnimCurve = 0)
```

How to: Access, play or stop an ambiance

Access an ambiance:

```
AmbianceList.instance.listAudioClip[int value]
```

Play an ambiance (Crossfade):

```
AmbianceManager.instance.MCrossFade(  
float crossFadeSpeed,  
AudioClip newClip = null,  
int whichAnimCurve = 0)
```

Stop current ambiance (fade Out):

```
AmbianceManager.instance.MFadeOut(  
float FadeOutSpeed,  
int whichAnimCurve = 0)
```

How to: Access or play Sfx

Access a Sound FX:

```
SfxList.instance.listAudioClip[int value]
```

Play a Sound FX:

```
SoundFxManager.instance.Play( AudioClip newClip,  
float volume = .5f);
```

Spatialized Sounds

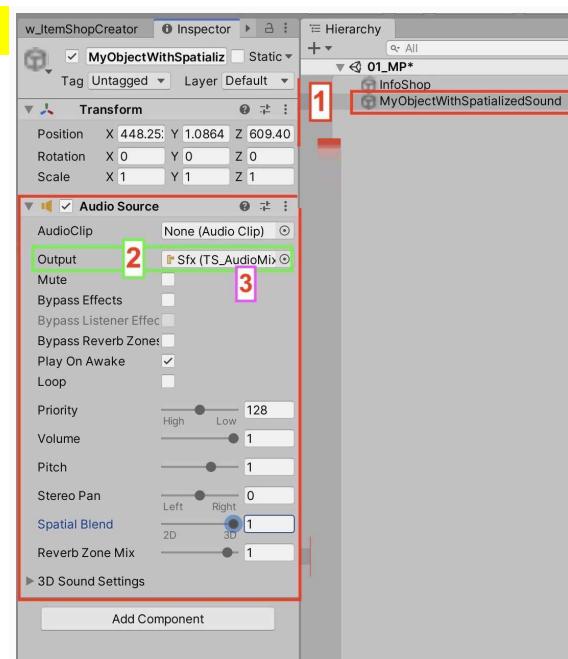
Some sounds needs to be spatialized
(example: the motor engine, weapon sounds...).

In that case, the **AudioSource** component used for this sound must be attached to the object that needs to be spatialized (spot 1).

It is possible to choose the audio output for each spatialized sound.

-Select the object with the **AudioSource** component attached to it (spot 2).

-Select the needed output bus (spot3).



How to: Delete/Reset Audio Mixer Data

-In the Hierarchy select **SoundManager** (spot 1).
(Hierarchy: ALL SCENES → MANAGERS → SoundManager)

-In the Inspector press the button
 Delete Audio Mixer Saved Data (spot 2).

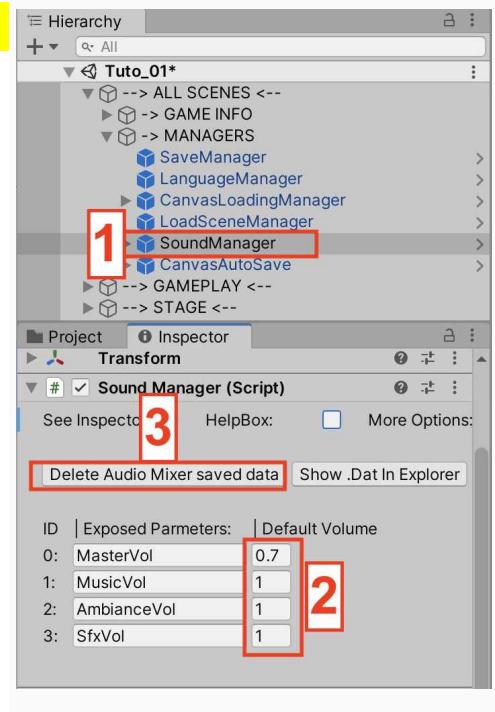


How to: Set the default Audio Mixer values

-In the Hierarchy select **SoundManager** (spot 1).
(Hierarchy → Sound Manager)

-In the Inspector Change the default values for each volume part (spot 2).
(Values **MUST** be between 0 and 1)

-Press the button **Delete Audio Mixer Saved Data** (spot 3).



How to: Modify default audio sound fx (engine, power up, damage,...)

This section describes how to access and modify default audio sound fx.

Table of contents:

Vehicle engine sounds	Link
Vehicle Booster	Link
Vehicle explosion	Link
Vehicle Hit sounds	Link
Warning sound	Link
Locked sound	Link
Power-ups	Link
Lap Sound	Link
Countdown	Link
Congratulation	Link

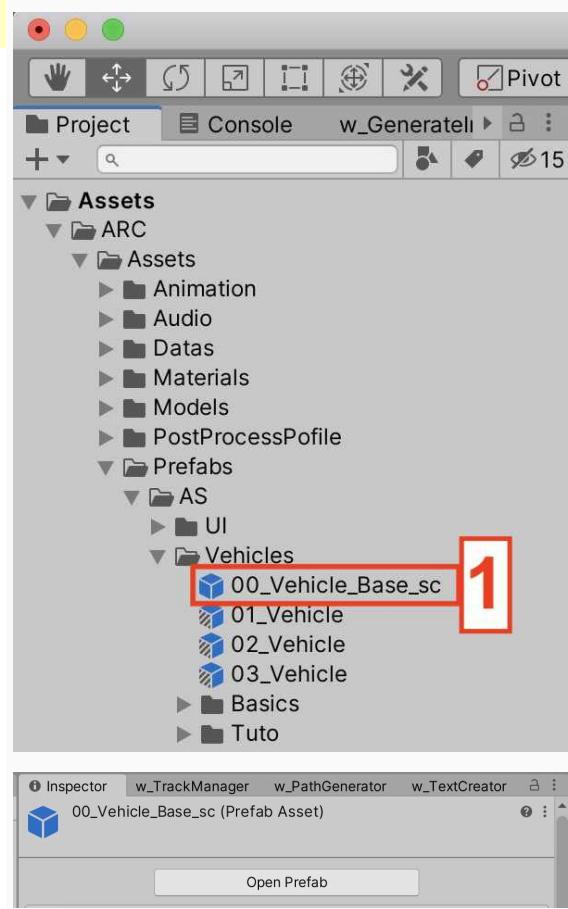
Vehicle engine sounds

If you want to make the modification for all the prefab at the same time select prefab

00_Vehicle_Base_sc (spot 1)

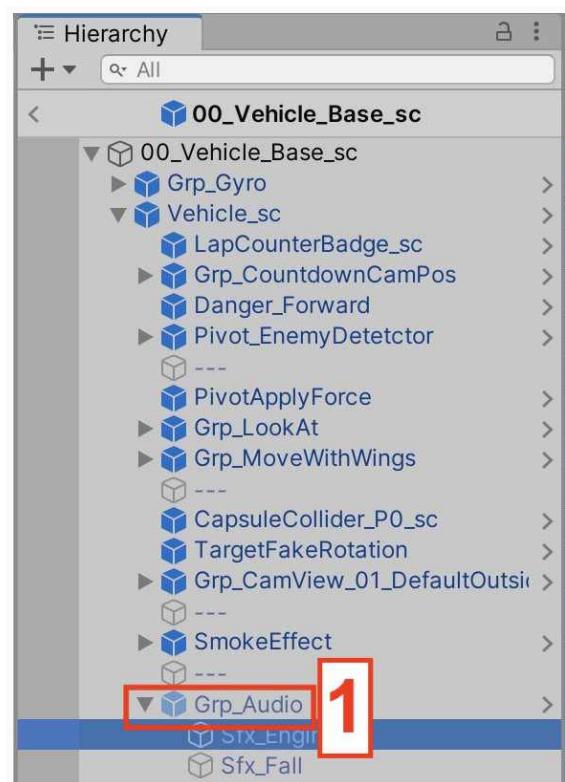
(Project Tab: Assets → Prefabs → AS → Vehicles → 00_Vehicle_Base_sc)

If you want to modify only one vehicle select the vehicle in the Project tab.



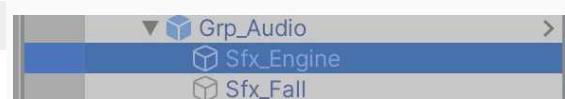
-In the Inspector press **Open Prefab** button.

The engine sounds are in folder **Grp_Audio** (spot 1).
(*Vehicle_sc* → *Grp_Audio*)

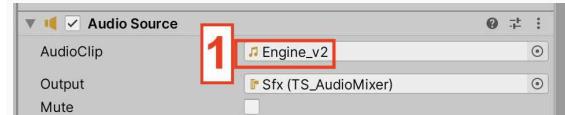


Engine:

Select **Sfx_Engine** object.
(*Vehicle_sc* → *Grp_Audio* → *Sfx_Engine*)

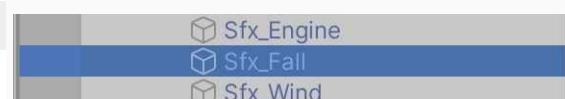


-In the Inspector you can change the **audioclip** (spot 1).



Fall:

Select **Sfx_Fall** object.
(*Vehicle_sc* → *Grp_Audio* → *Sfx_Fall*)



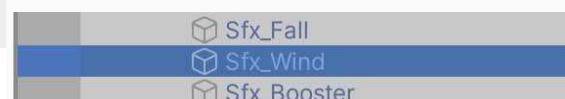
-In the Inspector you can change the **audioclip** (spot 1).



Wind:

Select **Sfx_Wind** object.
(*Vehicle_sc* → *Grp_Audio* → *Sfx_Wind*)

In the Inspector you can change the **audioclip** (spot 1).

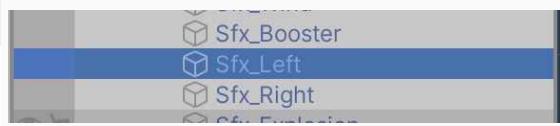


Turn Left:

Select **Sfx_Left** object.

(*Vehicle_sc* → *Grp_Audio* → *Sfx_Left*)

In the Inspector you can change the **audioclip** (spot 1).



Turn Right:

Select **Sfx_Right** object.

(*Vehicle_sc* → *Grp_Audio* → *Sfx_Right*)

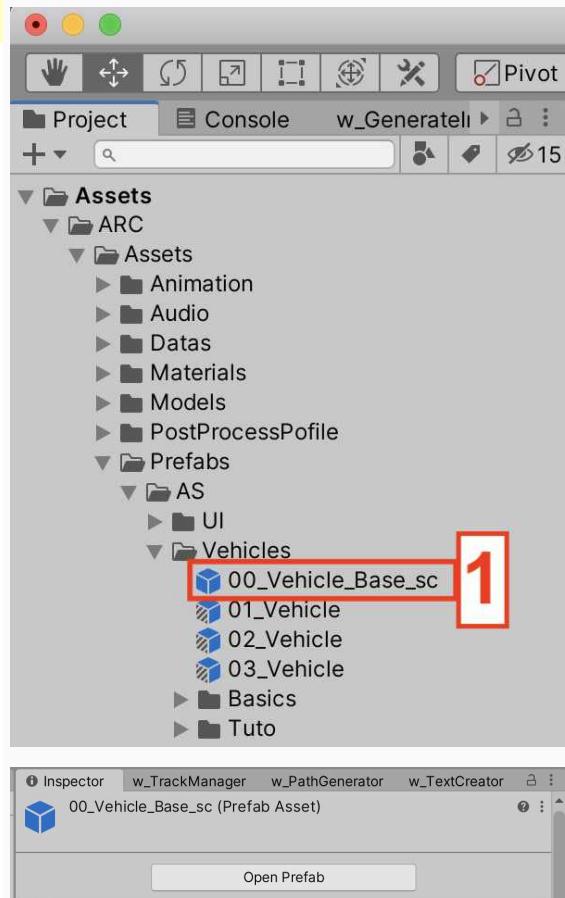
In the Inspector you can change the **audioclip** (spot 1).



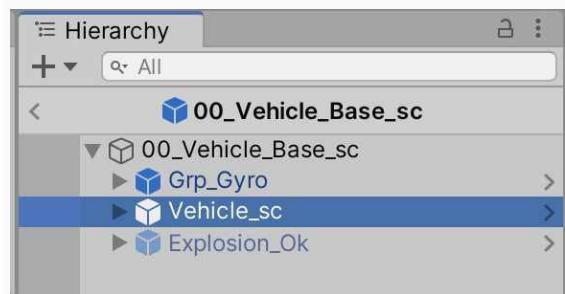
Booster

If you want to make the modification for all the prefab at the same time select prefab **00_Vehicle_Base_sc** (spot 1) (Project Tab: Assets → Prefabs → AS → Vehicles → **00_Vehicle_Base_sc**)

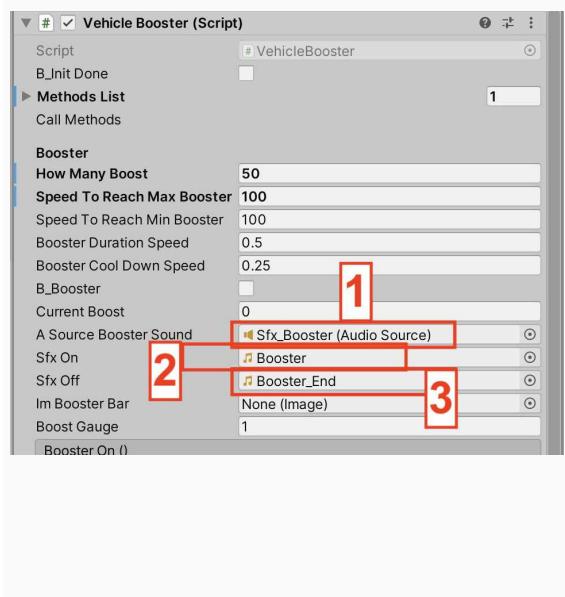
If you want to modify only one vehicle select the vehicle in the Project tab.



-In the Inspector press **Open Prefab** button.



-In the Hierarchy select **Vehicle_sc**.



In the Inspector:

A Source Booster Sound (spot 1) gives you access to the AudioSource used to play the booster sound.

The booster sound is separated into 2 sounds:

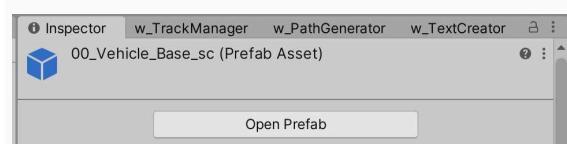
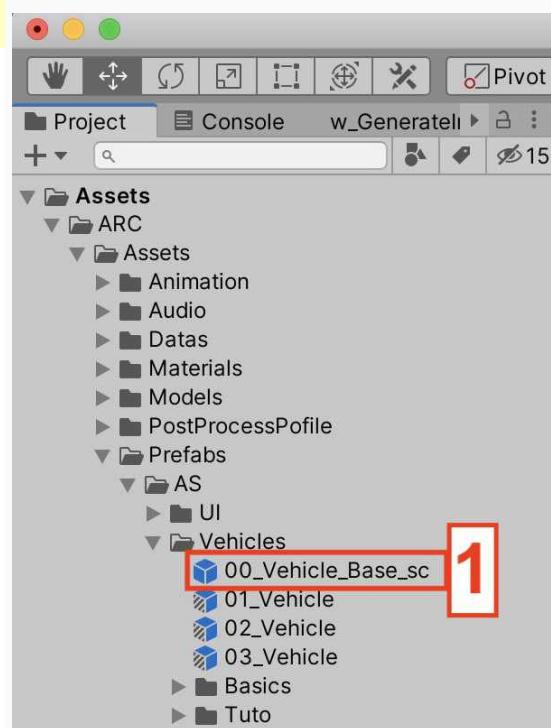
Sfx On (spot 2): This sound is played when the player presses the booster button.

Sfx Off (spot 3): This sound is played when the player releases the booster button.

Vehicle explosion

If you want to make the modification for all the prefab at the same time select prefab **00_Vehicle_Base_sc** (spot 1) (Project Tab: Assets → Prefabs → AS → Vehicles → **00_Vehicle_Base_sc**)

If you want to modify only one vehicle select the vehicle in the Project tab.

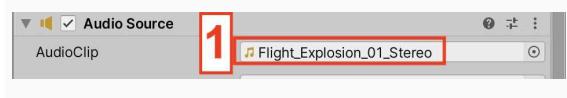


The explosion sound is in folder **Grp_Audio** (spot 1).
(*Vehicle_sc* → *Grp_Audio*)



Select **Sfx_Explosion** object.
(*Vehicle_sc* → *Grp_Audio* → *Sfx_Explosion*)

-In the Inspector you can change the **audioclip** (spot 1).



Vehicle Hit sounds

Hit sounds are sounds when the player is hit but he doesn't exploded.

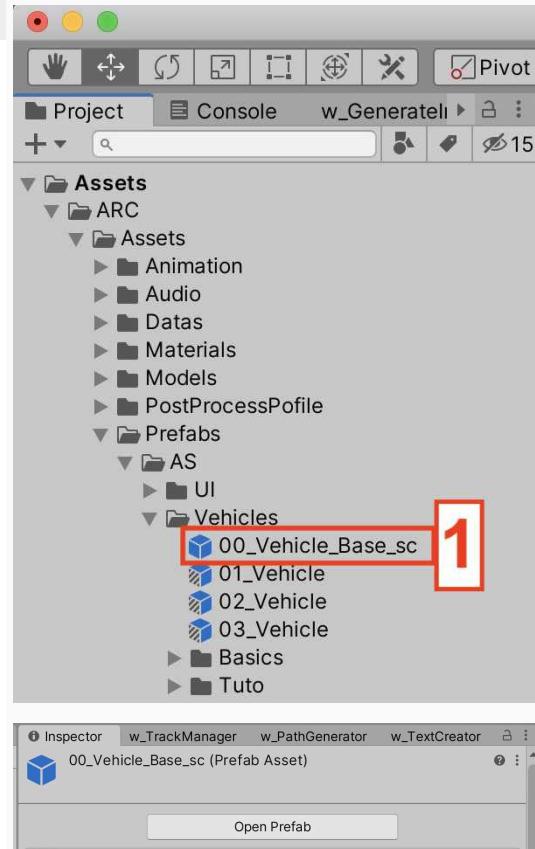
By default hit sounds are:

- The player is touched by a missile
- The player is touched by a machine gun bullet.

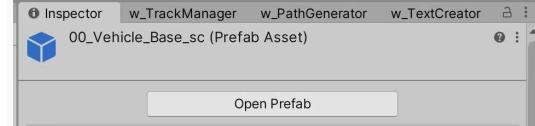
Missile Hit Sound

If you want to make the modification for all the prefab at the same time select prefab **00_Vehicle_Base_sc** (spot 1) (Project Tab: Assets → Prefabs → AS → Vehicles → **00_Vehicle_Base_sc**)

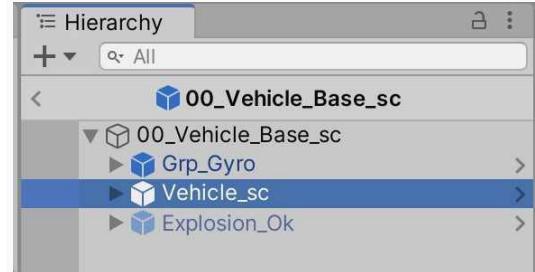
If you want to modify only one vehicle select this vehicle in the Project tab.



-In the Inspector press **Open Prefab** button.



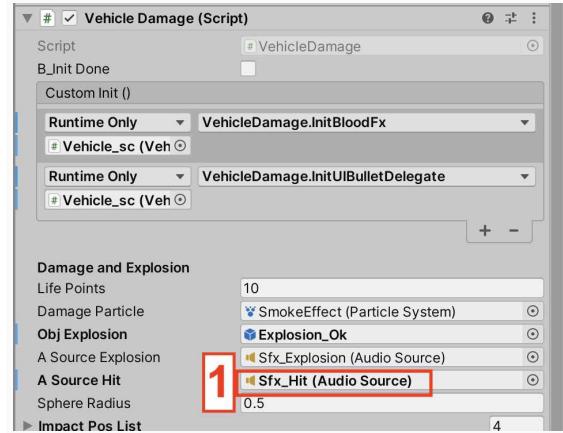
-In the Hierarchy select **Vehicle_sc**.



In the Inspector:

-Go to **Vehicle Damage** component.

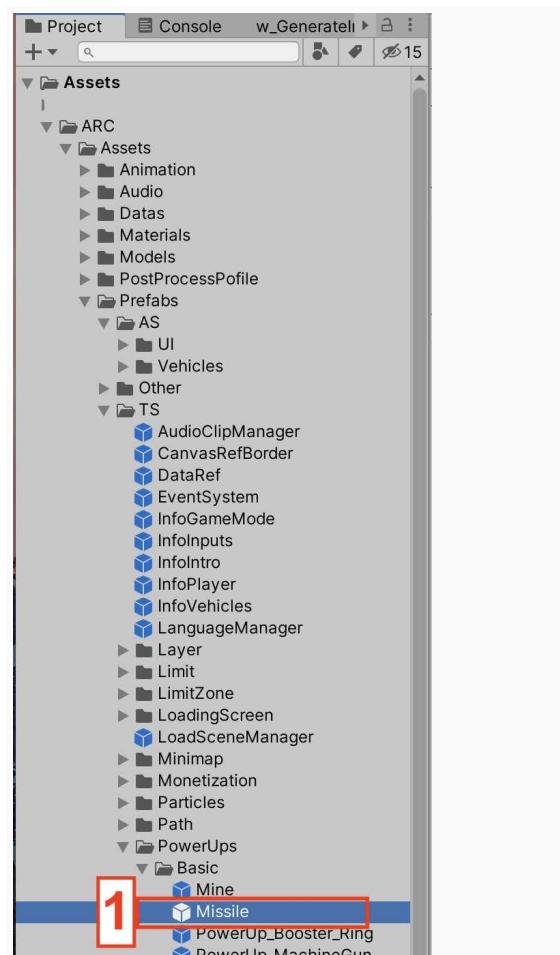
-A **Source Hit** (spot 1) gives you access to the Audiosource used to play the hit sounds.



To access the sound used when player is hit by a missile:

In Project Tab select **Missile** prefab (spot 1).

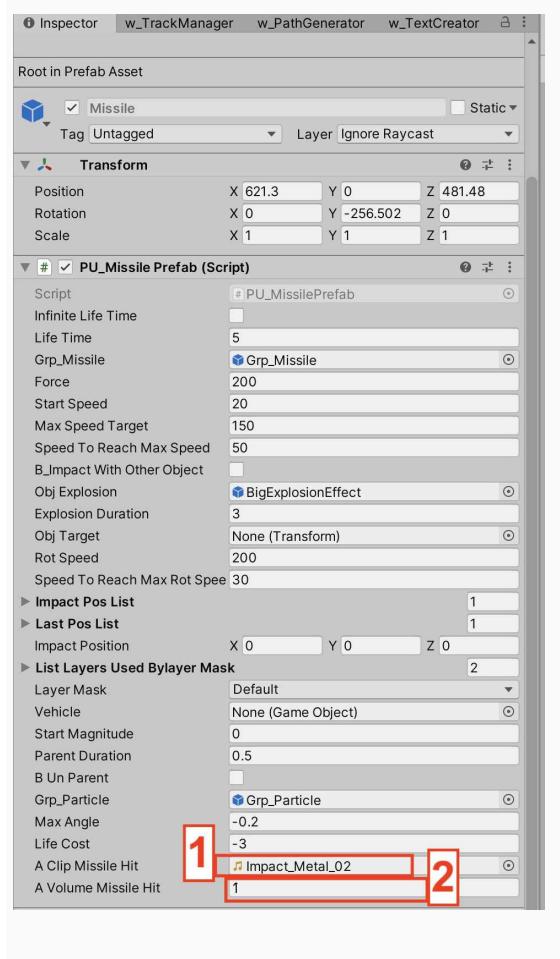
(Project Tab: Assets → Prefabs → TS → PowerUps → basic → Missile).



-In the Inspector:

-**Clip Missile Hit** (spot 1) corresponds to the sound played when the player is hit by a missile.

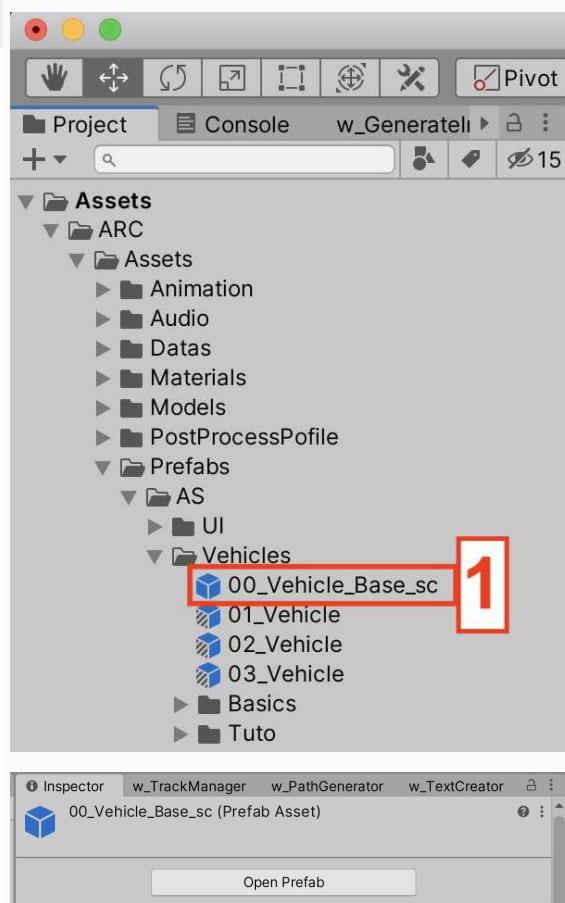
-**Volume Missile Hit** (spot 2) corresponds to the volume used to play the sound played the player is hit by a missile.



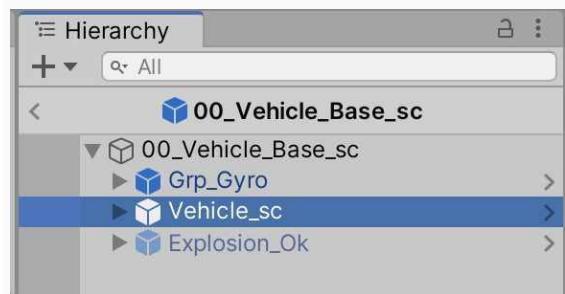
Machine Gun Hit:

If you want to make the modification for all the prefab at the same time select prefab **00_Vehicle_Base_sc** (spot 1) (Project Tab: Assets → Prefabs → AS → Vehicles → **00_Vehicle_Base_sc**)

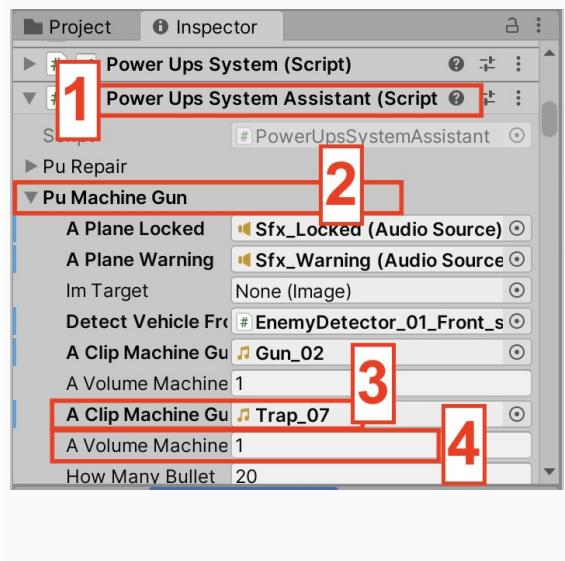
If you want to modify only one vehicle select the vehicle in the Project tab.



-In the Inspector press **Open Prefab** button.



-In the Hierarchy select **Vehicle_sc**.



In the Inspector:

-Go to script **Power Ups System Assistant** (spot 1).

-Go to section **PU Machine Gun** (spot 2)

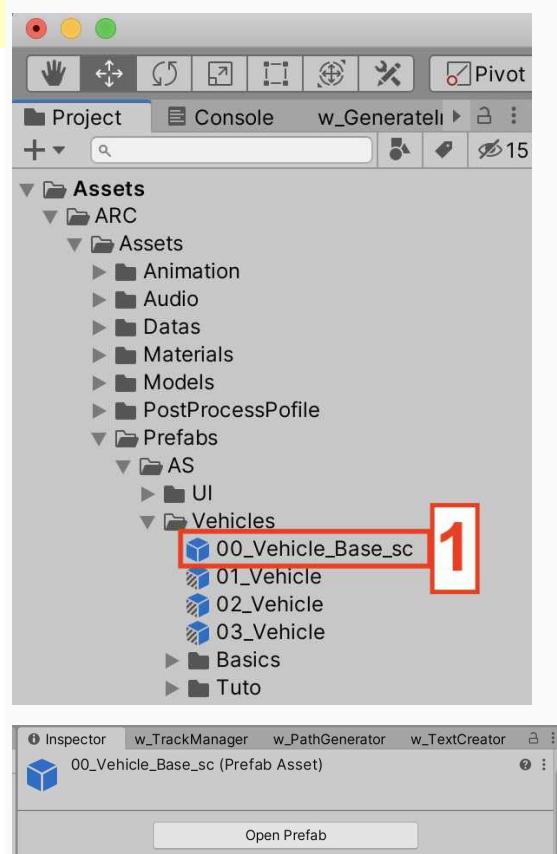
Clip Machine Gun Hit (spot 3) is the sound used when the player is hit by a machine gun bullet.

Volume Machine Gun Hit (spot 4) is the volume uses to play the sound.

Warning sound

If you want to make the modification for all the prefab at the same time select prefab **00_Vehicle_Base_sc** (spot 1) (Project Tab: Assets → Prefabs → AS → Vehicles → **00_Vehicle_Base_sc**)

If you want to modify only one vehicle select the vehicle in the Project tab.



-In the Inspector press **Open Prefab** button.

The engine sound fx are in folder **Grp_Audio** (spot 1). (**Vehicle_sc** → **Grp_Audio**)



Select **Sfx_Warning** object.

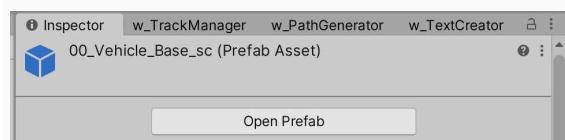
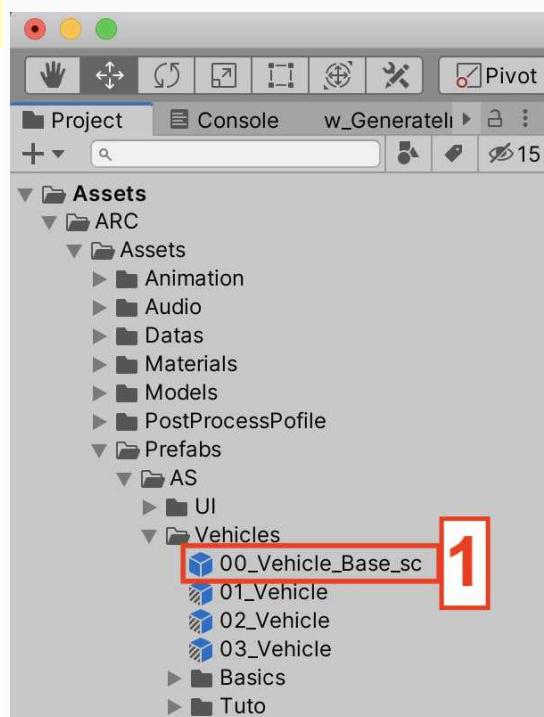
(**Vehicle_sc** → **Grp_Audio** → **Sfx_Warning**)

-In the Inspector you can change the **audioclip** (spot 1).

Locked sound

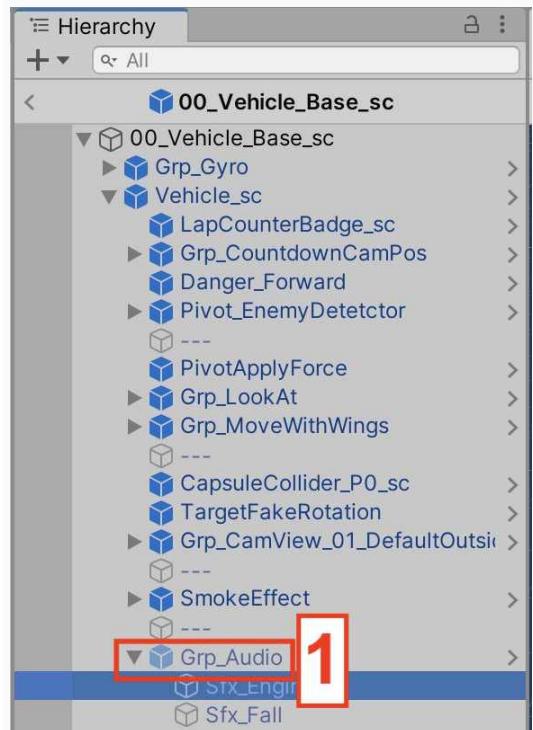
If you want to make the modification for all the prefab at the same time select prefab **00_Vehicle_Base_sc** (spot 1) (Project Tab: Assets → Prefabs → AS → Vehicles → **00_Vehicle_Base_sc**)

If you want to modify only one vehicle select the vehicle in the Project tab.



-In the Inspector press **Open Prefab** button.

The engine sound fx are in folder **Grp_Audio** (spot 1). (**Vehicle_sc** → **Grp_Audio**)



Select **Sfx_Locked** object.

(**Vehicle_sc** → **Grp_Audio** → **Sfx_Locked**)

-In the Inspector you can change the **audioclip** (spot 1).



Power-ups

Grab a power-up

In Project Tab select a power-up prefab (spot 1).

(Project Tab: Assets → Prefabs → TS → PowerUps → basic →).

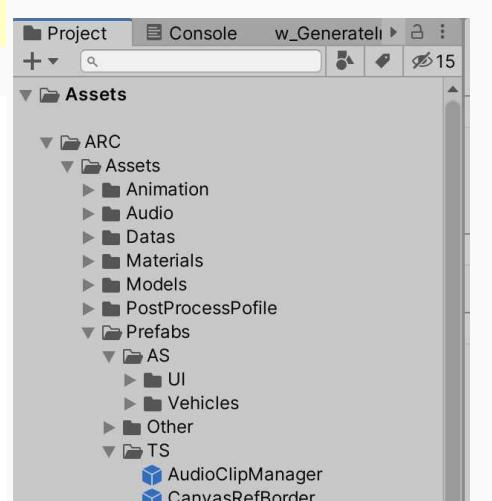
PowerUp_MachineGun

PowerUp_Mine

PowerUp_Missile

PowerUp_Repair

PowerUp_Shield



In the Inspector it is possible to access the sound used when the power-up is grabbed by a vehicle (spot 1).



Note:

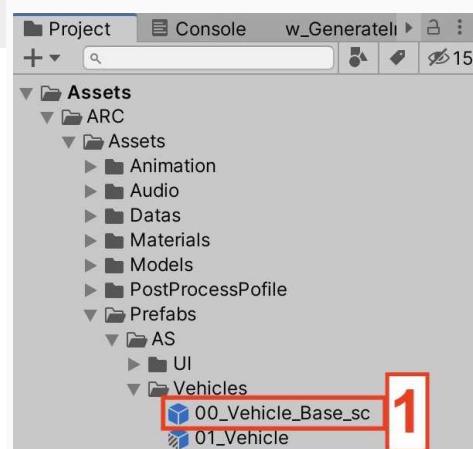
It is possible to choose a different sound for each power-up type.



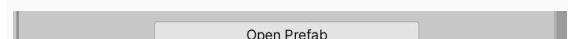
Repair

If you want to make the modification for all the prefab at the same time select prefab 00_Vehicle_Base_sc (spot 1) (Project Tab: Assets → Prefabs → AS → Vehicles → 00_Vehicle_Base_sc)

If you want to modify only one vehicle select this vehicle in the Project tab.



-In the Inspector press Open Prefab button.



-In the Inspector go to script
Power Ups System Assistant



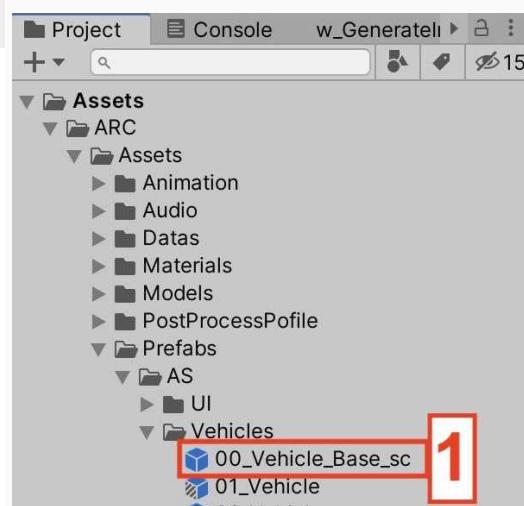
In the Inspector it is possible to access the sound and the volume used when the player uses the power-up (spot 1).



Machine Gun

If you want to make the modification for all the prefab at the same time select prefab **00_Vehicle_Base_sc** (spot 1) (Project Tab: Assets → Prefabs → AS → Vehicles → **00_Vehicle_Base_sc**)

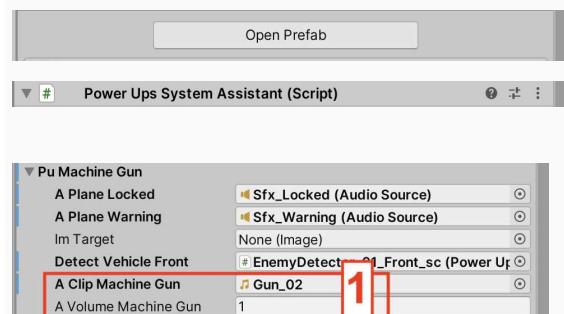
If you want to modify only one vehicle select the vehicle in the Project tab.



-In the Inspector press **Open Prefab** button.

-In the Inspector go to script **Power Ups System Assistant**

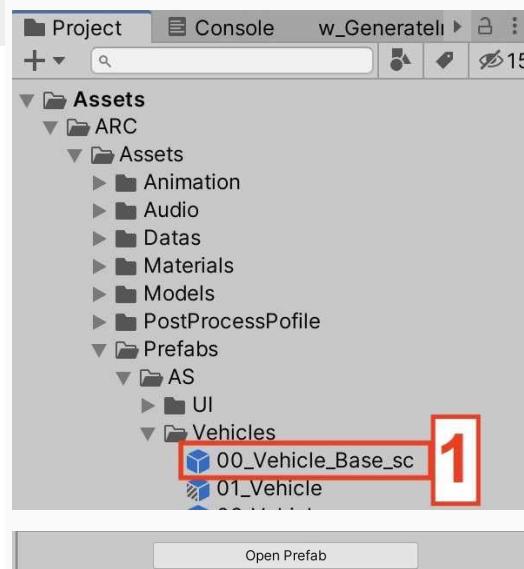
In the Inspector go to **PU Machine Gun** section to access the sound and the volume used when the player uses the power-up (spot 1).



Shield

If you want to make the modification for all the prefab at the same time select prefab **00_Vehicle_Base_sc** (spot 1) (Project Tab: Assets → Prefabs → AS → Vehicles → **00_Vehicle_Base_sc**)

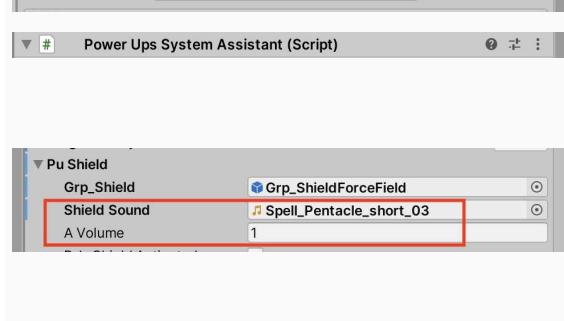
If you want to modify only one vehicle select this vehicle in the Project tab.



-In the Inspector press **Open Prefab** button.

-In the Inspector go to script **Power Ups System Assistant**

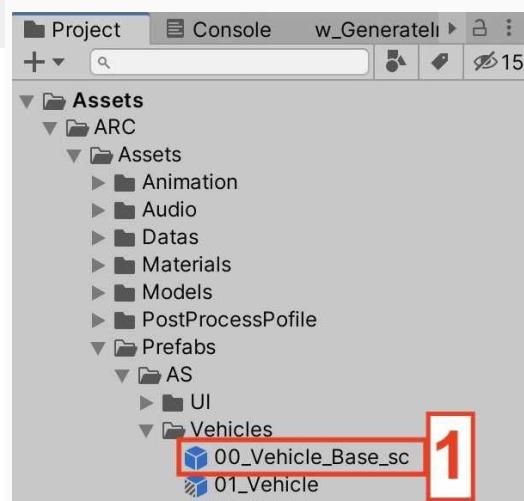
In the Inspector go to **PU Shield** section to access the sound and the volume used when the player uses the power-up.



Mine

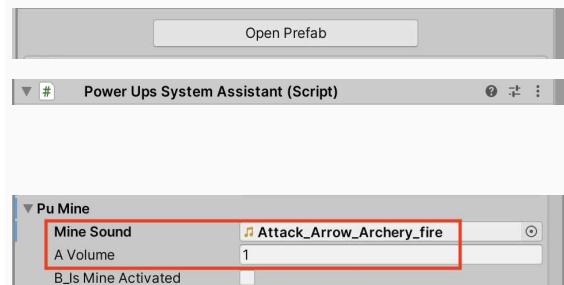
If you want to make the modification for all the prefab at the same time select prefab **00_Vehicle_Base_sc** (spot 1) (*Project Tab: Assets → Prefabs → AS → Vehicles → 00_Vehicle_Base_sc*)

If you want to modify only one vehicle select the vehicle in the Project tab.



-In the Inspector press **Open Prefab** button.

-In the Inspector go to script **Power Ups System Assistant**

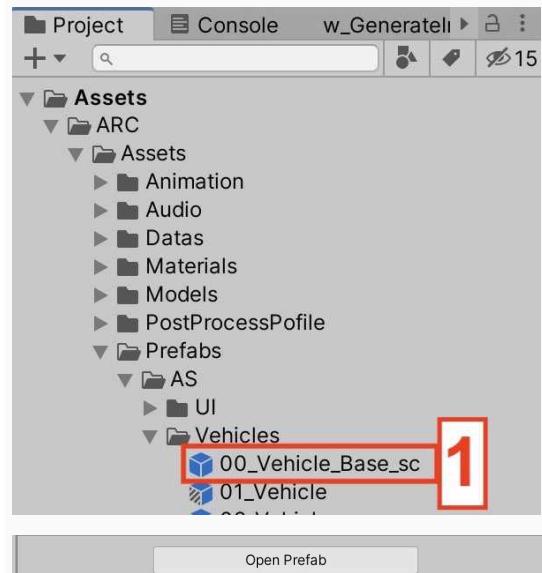


In the Inspector go to **PU Mine** section to access the sound and the volume used when the player uses the power-up.

Missile

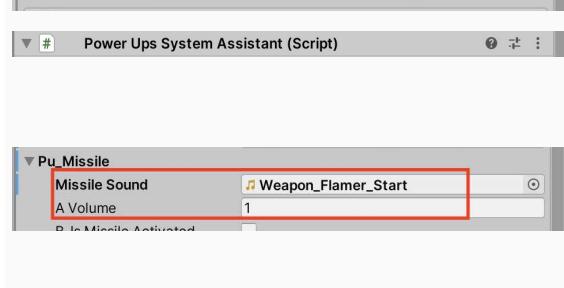
If you want to make the modification for all the prefab at the same time select prefab **00_Vehicle_Base_sc** (spot 1) (*Project Tab: Assets → Prefabs → AS → Vehicles → 00_Vehicle_Base_sc*)

If you want to modify only one vehicle select the vehicle in the Project tab.



-In the Inspector press **Open Prefab** button.

-In the Inspector go to script **Power Ups System Assistant**

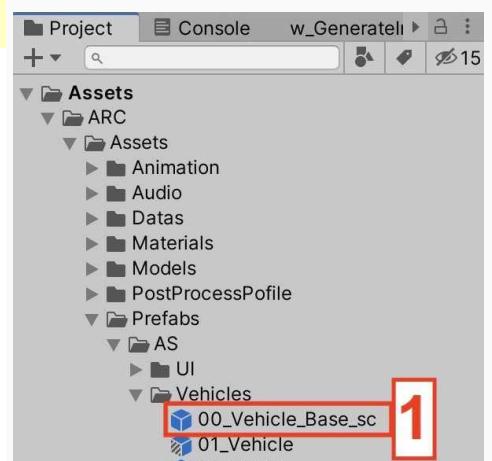


In the Inspector go to **PU Missile** section to access the sound and the volume used when the player uses the power-up

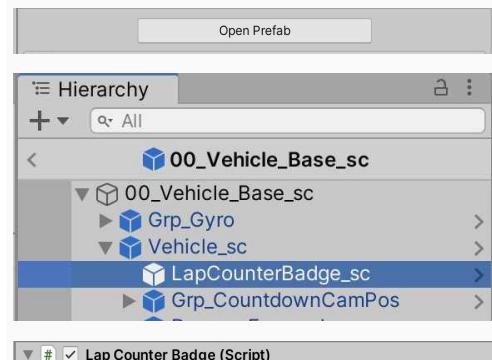
Lap Sound

If you want to make the modification for all the prefab at the same time select prefab **00_Vehicle_Base_sc** (spot 1) (Project Tab: Assets → Prefabs → AS → Vehicles → **00_Vehicle_Base_sc**)

If you want to modify only one vehicle select the vehicle in the Project tab.



-In the Inspector press **Open Prefab** button.



-In the Inspector go to script **Lap Counter Badge**

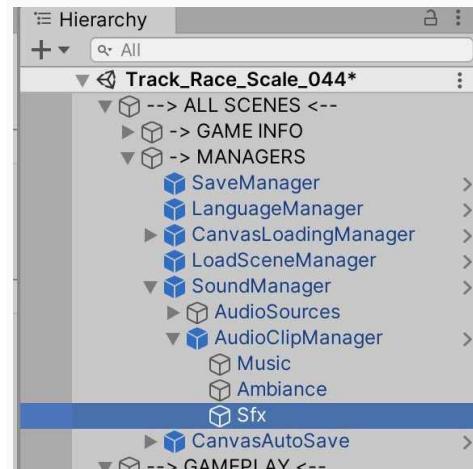


-In section **Do Something After A Lap()** a sound is played.

You can choose the **ID** of the sound you want to play (spot 1).

The ID corresponds to sounds included in the **SfxList** script

In the Hierarchy select **sfx** object
(**ALL SCENES** → **MANAGERS** → **SoundManager** → **AudioClipManager** → **Sfx**)



In the Inspector it is possible to access to the sound list.

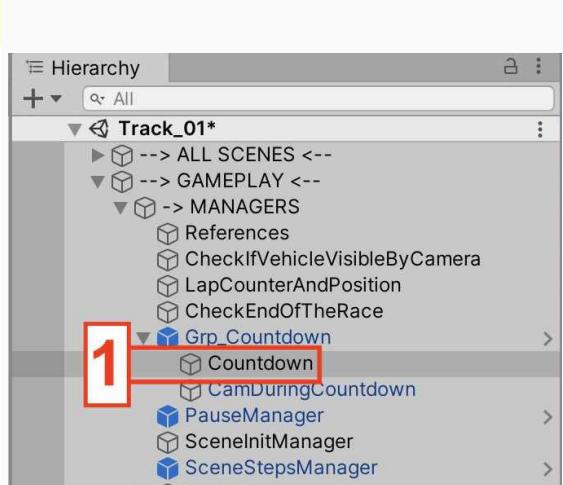
Each sound has a unique ID (spot 1).



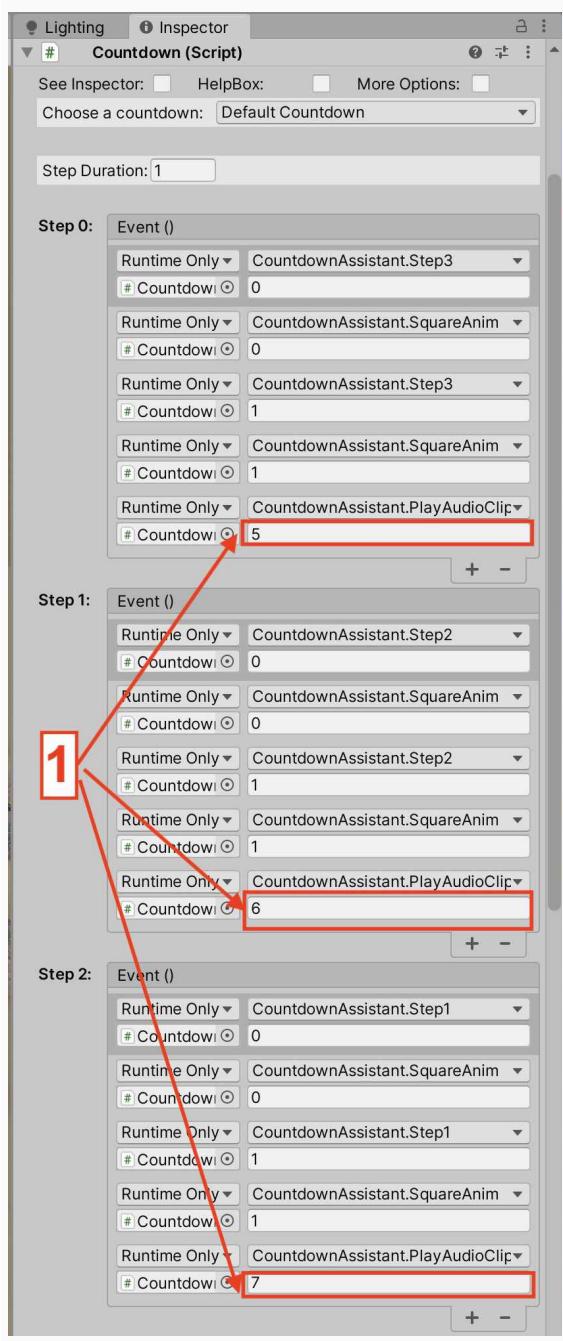
Countdown

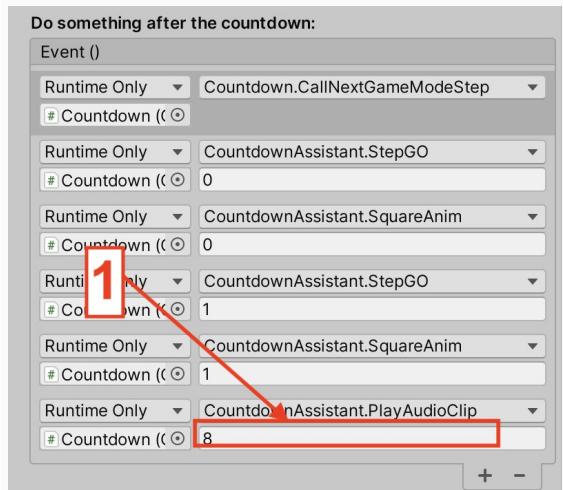
-Open a track scene.

-In the Hierarchy select **Countdown** object (spot 1)
(Hierarchy: GAMEPLAY → MANAGERS → Grp_Countdown → Countdown)



For each countdown step change the ID to select the audioclip.

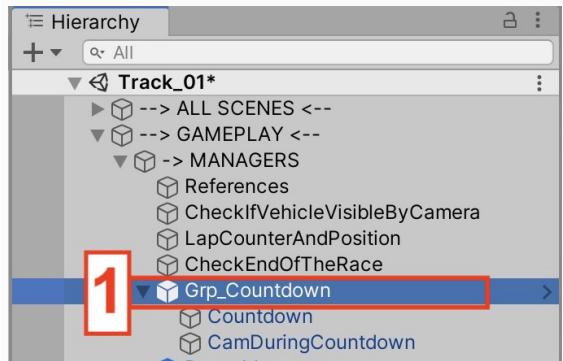




-In the Hierarchy select **Grp_Countdown**

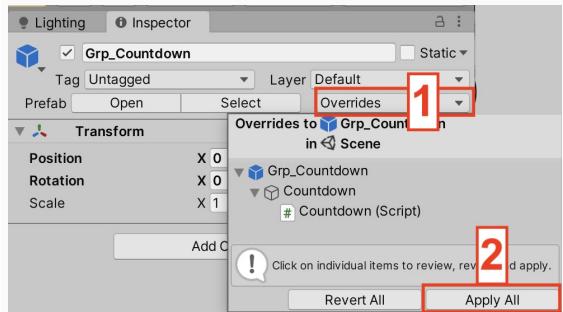
object (spot 1)

(Hierarchy: GAMEPLAY → MANAGERS → Grp_Countdown)



-In the Inspector press **Overrides** button (spot 1).

-Then press **Apply All** button (spot 2).

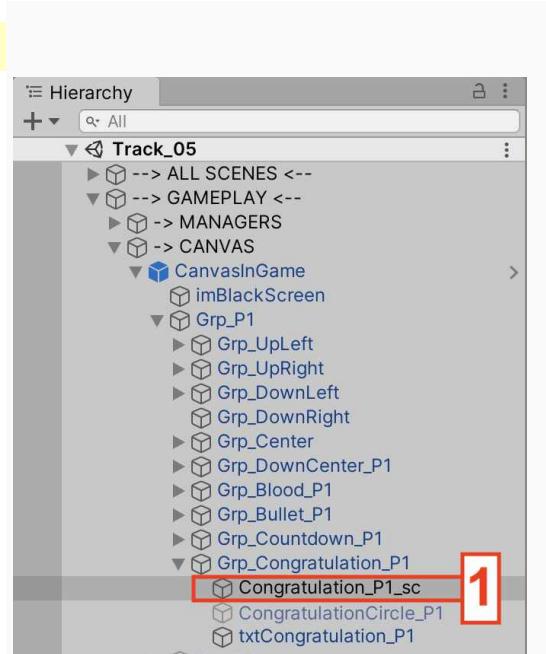


Congratulation

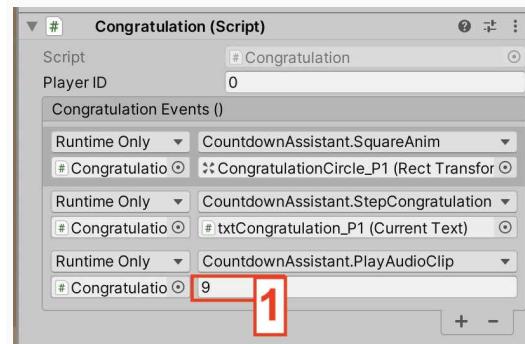
-Open a track scene.

-In the Hierarchy select **Congratulation_P1_sc** object (spot 1)

(Hierarchy: GAMEPLAY → CANVAS → CanvasInGame → Grp_P1 → Grp_Congratulation_P1 → Congratulation_P1_sc)



-In the Inspector change the AudioClip ID (spot 1)

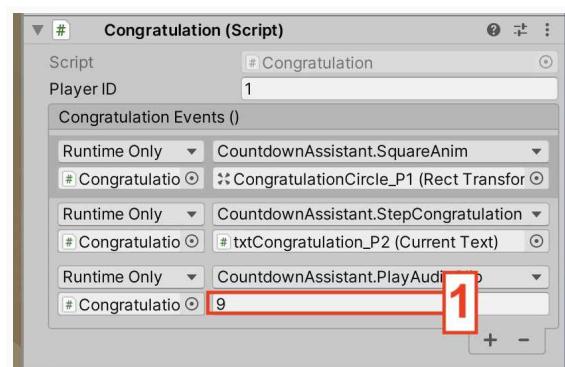


-In the Hierarchy select **Congratulation_P2_sc** object (spot 1)

(Hierarchy: GAMEPLAY → CANVAS → CanvasInGame → Grp_P2 → Grp_Congratulation_P2 → Congratulation_P2_sc)

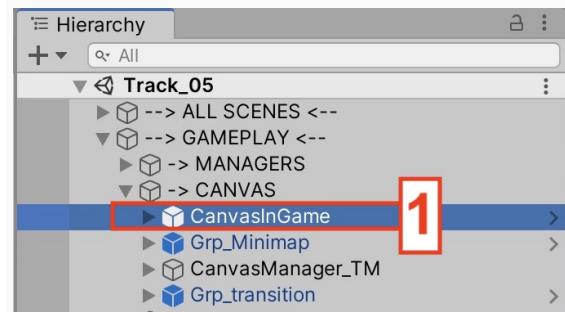


-In the Inspector change the AudioClip ID (spot 1)

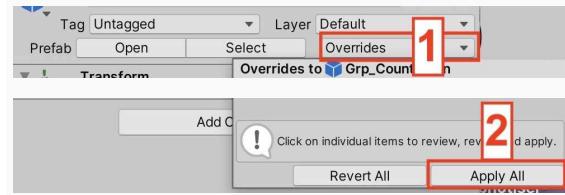


-In the Hierarchy select **CanvasInGame** object (spot 1)

(Hierarchy: GAMEPLAY → CANVAS → CanvasInGame)



-In the Inspector press **Overrides** button (spot 1).



-Then press **Apply All** button (spot 2).



Save System

This section describes the save system included in the asset.

The save system allows to:

- Save data as a string.
- Load data as a string.

Table of contents:

Overview	Link
Save System Diagram	Link
How to: Select the save format (PlayerPrefs or .Dat)	Link
How to: Reset Data	Link
Scripting: Save/Load data	Link

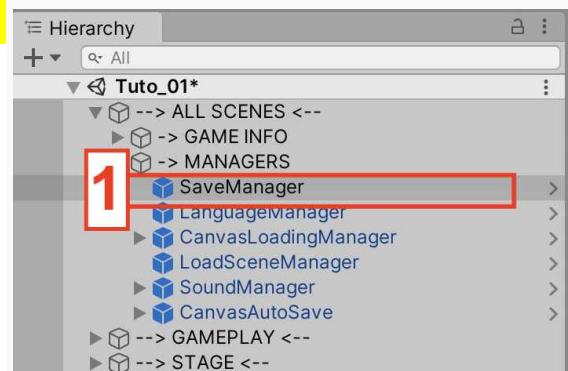
Overview

The save system is managed in the Hierarchy by **SaveManager** object (spot 1).

(Hierarchy: ALL SCENES → MANAGERS → SaveManager)

This object allows to save and load data in the form of a text (string).

Data can be saved as Unity PlayerPrefs or .Dat.



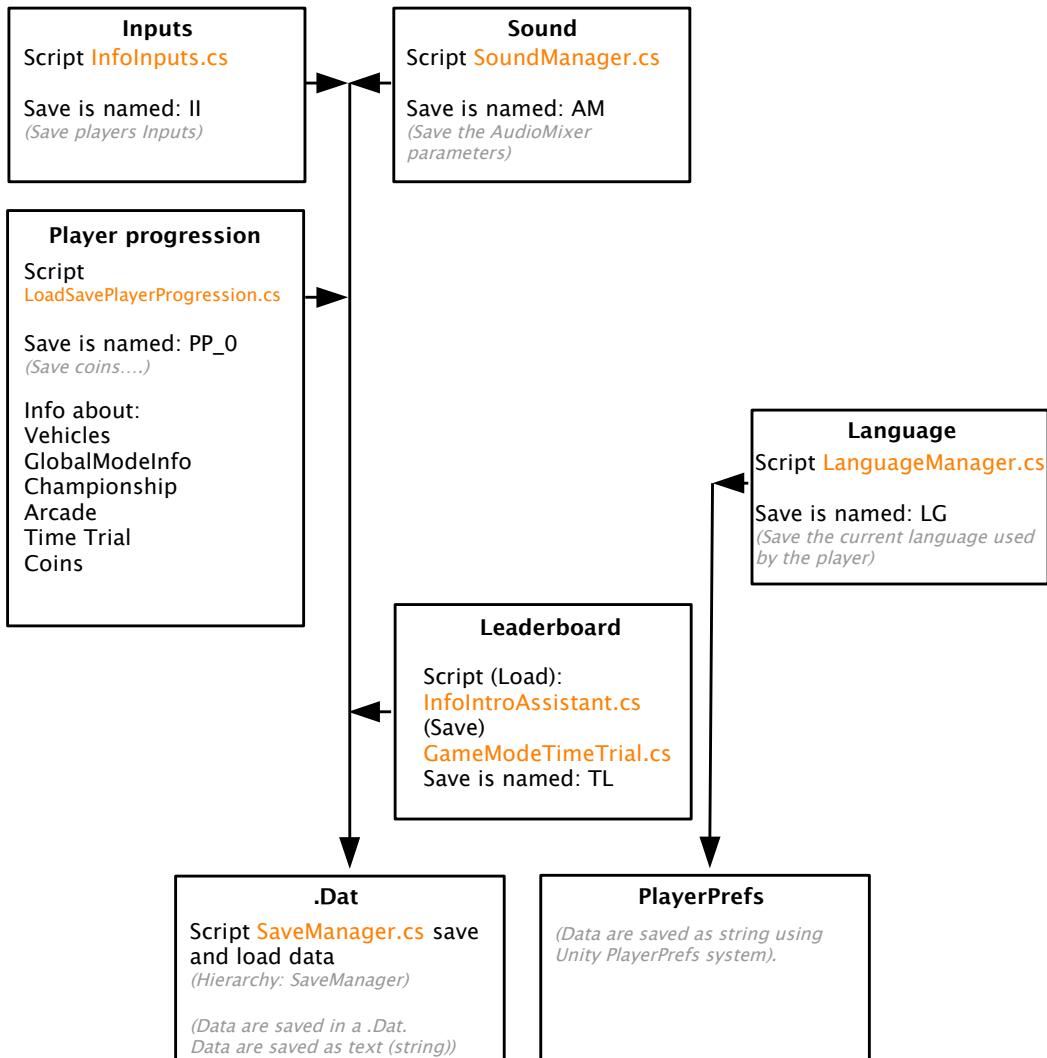
Info:

There is only 1 **SaveManager** by scene.

SaveManager is not destroyed when a new scene is loaded.

It is possible to access **SaveManager** from any script.

Save System Diagram



Data are called in `InfoIntro` when the game is launched:
Input, sound, player progression, Language, Leaderboards

How to: Select the save format

By default .Dat format is selected.

.Dat allows bigger save size than PlayerPrefs.

The only case you must choose PlayerPrefs as the save format is for WebGL build.

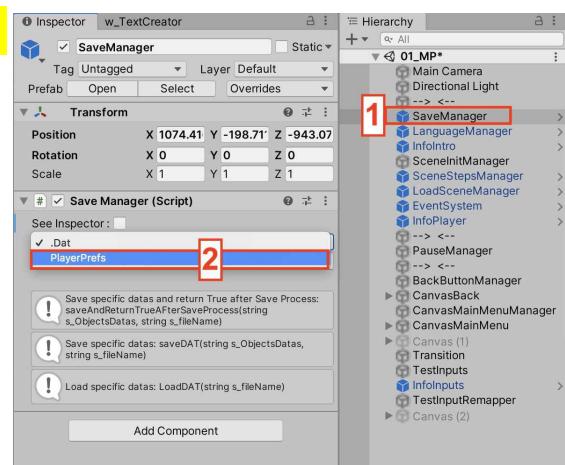
By default data are saved in .Dat file.

It is possible to choose **PlayerPrefs** as the default save format.

-In the Hierarchy select **SaveManager** (spot 1).

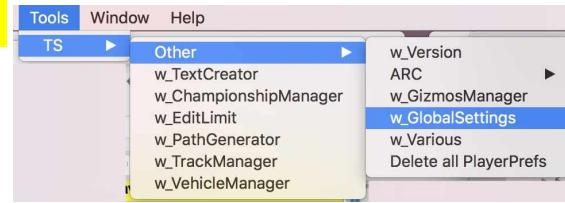
-In the Inspector select **PlayerPrefs** in the dropdown menu (spot 2)

Now the data are save and load as PlayerPrefs



How to: Reset data (.Dat)

-Go to Tools → TS → Other → w_GlobalSettings



In the new window press Show .Dat in Explorer



A new folder appears.

-Delete the file included in the folder.

How to: Reset data (PlayerPrefs)

Go to Tools → TS → Other -> Delete all PlayerPrefs



Scripting

How to: Save data

Call the method:

```
SaveManager.instance.saveDAT(  
    string s_ObjectsDatas, string s_fileName)
```

s_ObjectsDatas is the text to save.
s_fileName is the name of the save.

How to: Load data

Call the method:

```
SaveManager.instance.LoadDAT(string s_fileName)
```

s_fileName is the name of the save.
The method return a string.

Localization and texts at runtime

This section describes the system to create, edit and display texts in the asset.

Table of contents:

Create and Edit texts (w_TextCreator window) : Overview	Link
How to: Edit a text	Link
How to: Create a new text	Link
How to: Create a new text folder	Link
How to: Add a new language	Link
LanguageManager: Overview	Link
Scripting	Link
Text component (CurrentText.cs): Overview	Link
How to: Set up a text using the Multi-language system	Link
Scripting	Link

Create and Edit texts (w_TextCreator window) :

Overview

The asset has a system to manage multi-language texts.

The asset allows to edit and create multi-language texts with the window **w_TextCreator**.

To access this window:

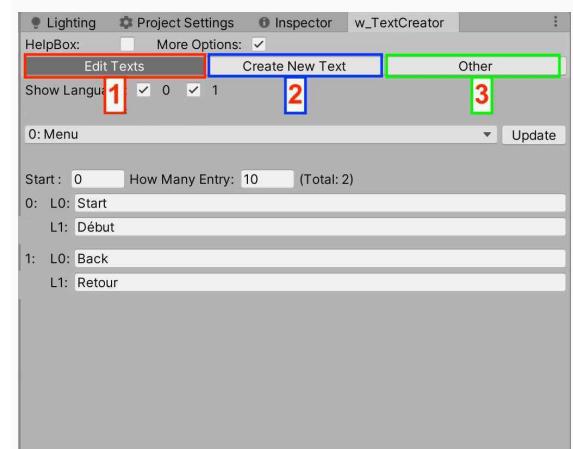
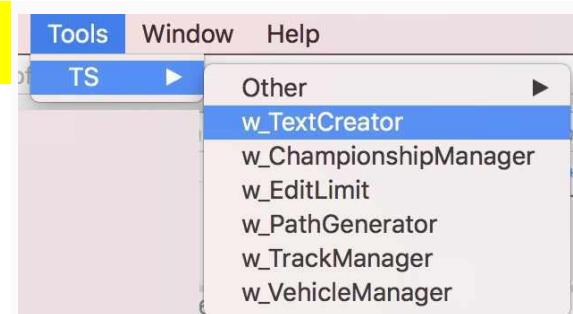
-Go to **Tools** → **TS** → **w_TextCreator**

w_TextCreator window allows to:

-Edit text (spot 1).

-Create new text (spot 2).

-Add new language and create new text folder (spot 3).



Info:

To prevent bugs:

- It is not possible to delete an existing text folder.
- It is not possible to delete an existing text.
- It is not possible to delete an existing language.

IMPORTANT:

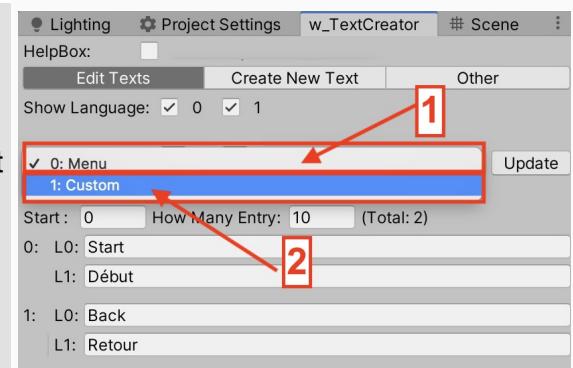
Folders make it easier to find text.

By default there are 2 folders:

Menu: The folder contains all the texts used by default in the asset.

- You can modify the text contained in this folder.
- You cannot create new text in this folder.

Custom: By default this folder is empty. You can use it to add your own text.



If you use a lot of text it is possible to create a new folder to facilitate the management of the text

→ [Link](#)

How to: Edit a text

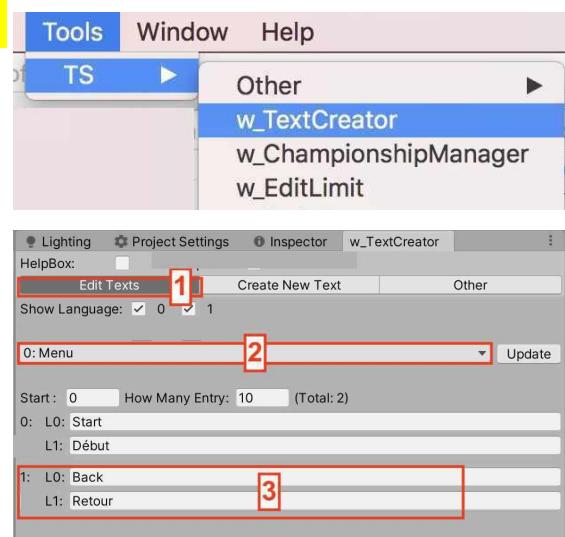
-Go to Tools → TS → w_TextCreator

-Press button **Edit Text** to enter into this section (spot 1).

-Select into the dropdown menu the text folder that contains the text to edit (spot 2).

By default there are 2 text folders (*Menu* and *custom*)

-Modify the text (spot 3).



More Info:

-It is possible to change the number of text displayed in the list by changing **Start** (spot 1)

(For example if *Start* = 2 the first text displayed in the list will be text 2)

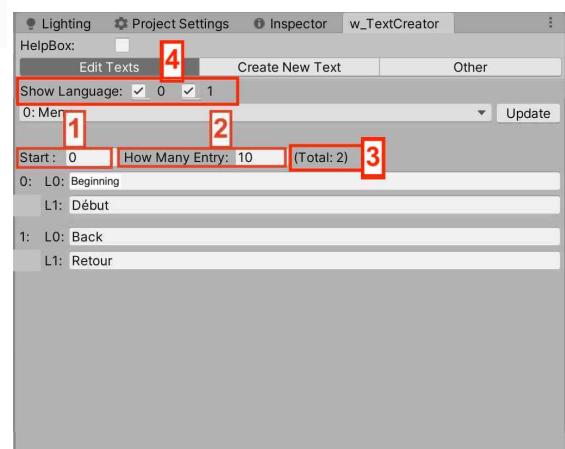
and

How many entry (spot 2)

(For example if *How many entry* = 10 the list will displayed 10 texts)

The number of text in a folder is displayed spot 3.

It is possible to **show** or **hide** a language by pressing checkbox next to **Show Language** (spot 4)



How to: Create a new text

-Go to: Tools → TS → w_TextCreator

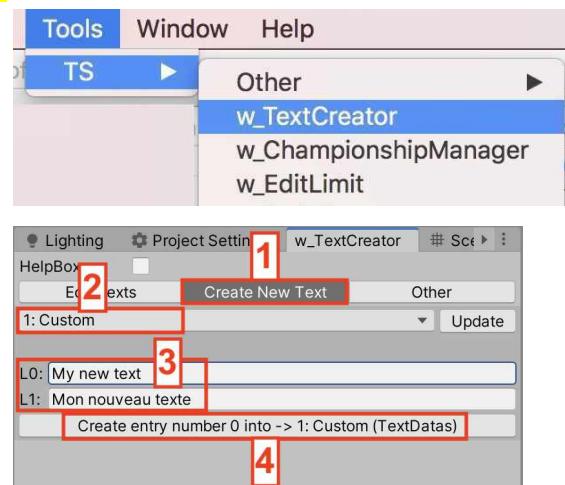
-Select tab **Create new text** (spot 1).

-Select the text folder where the new text will be saved (spot 2).

-Write texts for each language (spot 3)

-Press button **Create entry** (spot 4).

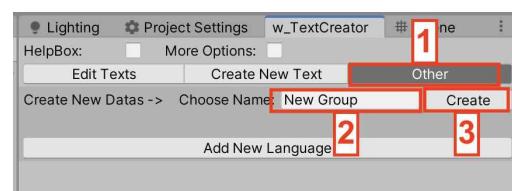
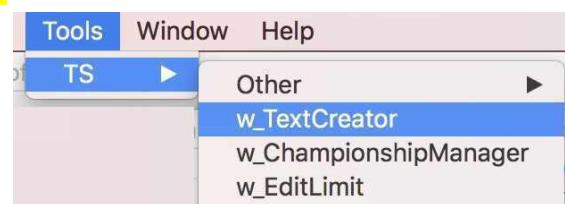
The new entry is created.



How to: Create a new text folder

- Go to: Tools → TS → w_TextCreator
- Select tab Other (spot 1).
- Write the name of the new text folder (spot 2).
- Press button Create (spot 3).

The new text folder is created.



How to: Add a new language

IMPORTANT:

Make a copy of your project before doing the next steps.

-Go to: Tools → TS → w_TextCreator

-Select tab Other (spot 1).

-Press button Add New Language (spot 3).

The new language is added.

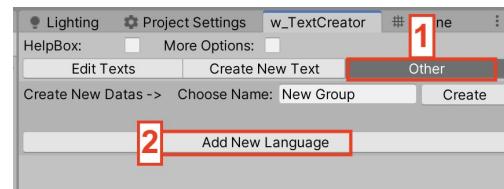
A new language is automatically added to the existing texts.



-Press button Add New Language (spot 3).

The new language is added.

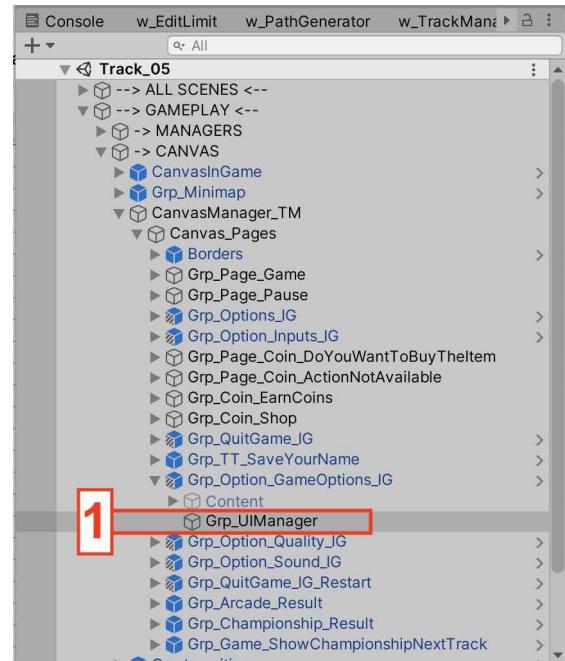
A new language is automatically added to the existing texts.



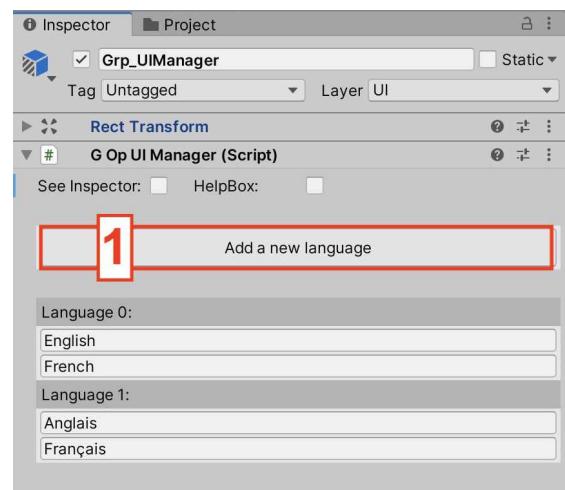
-Open one of your track scene.

-In the Hierarchy select Grp_UIManager (spot 1)

(Hierarchy: GAMEPLAY → CANVAS → CanvasManager_TM → CanvasPages → Grp_Option_GameOptions_IG → Grp_UIManager)

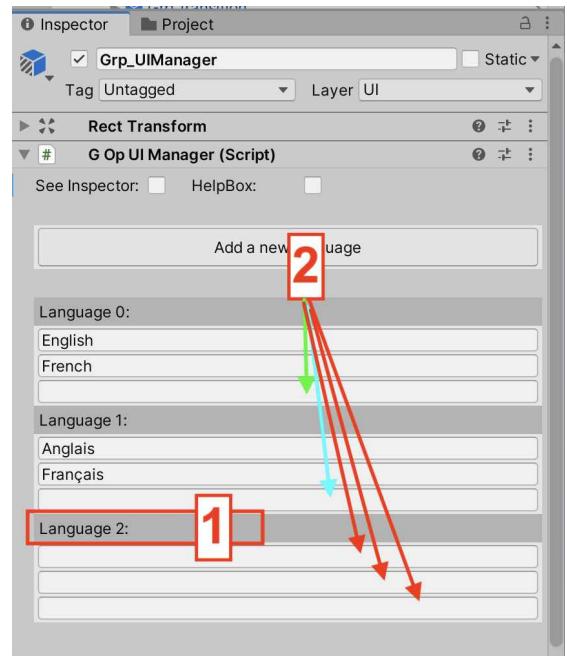


-In the Inspector press button Add a new language (spot 1)



A new language is added (spot 1)

-Write the name of your languages (spot 2)



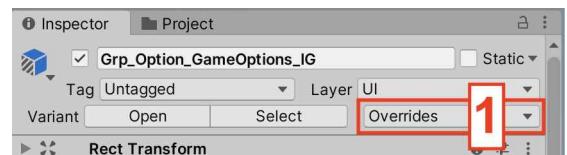
-In the Hierarchy select

Grp_Option_GameOptions_IG (spot 1)

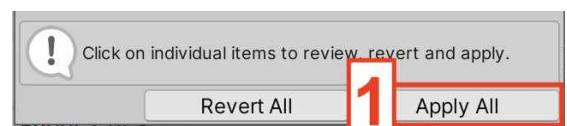
(Hierarchy: GAMEPLAY → CANVAS → CanvasManager_TM → CanvasPages → Grp_Option_GameOptions_IG)



-In the Inspector press **Overrides** button (spot 1)



-Press **Apply All** button (spot 1)

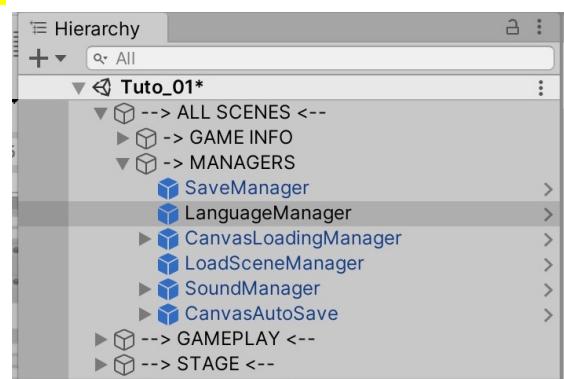


LanguageManager: Overview

Hierarchy: ALL SCENES → MANAGERS → LanguageManager

LanguageManager allows to:

- Access multi-language text from any script.
- Update current language in the current scene.
- Initialize texts when a scene starts.
- Save and Load the current language.



There is only one LanguageManager by scene.

LanguageManager is not destroyed when a new scene is loaded.

Scripting

Return a text:

```
LanguageManager.instance.String_ReturnText(int  
_WhichTextFolder, int _Entry)"
```

Update all texts (in the current scene):

```
LanguageManager.instance.Bool_UpdateAllTexts()
```

Access current Language (variable):

```
LanguageManager.instance.currentLanguage
```

Used to init the language when the game starts:

```
Bool_InitLanguage()
```

Update the current language:

```
Bool_UpdateSelectedLanguage(int value)
```

Save current Language:

```
PlayerPrefs.SetInt(LG)
```

Load current Language:

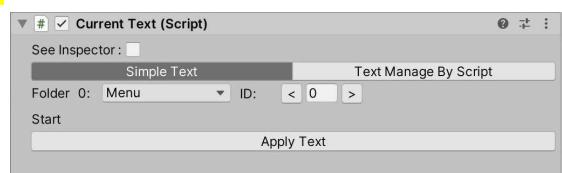
```
PlayerPrefs.GetInt(LG)
```

Text component (CurrentText.cs): Overview

CurrentText.cs is attached to every UI text.

This script allows to display a text using the Multi-language system included in the asset.

This script works the same with the default Unity Text component and Text Mesh Pro component.

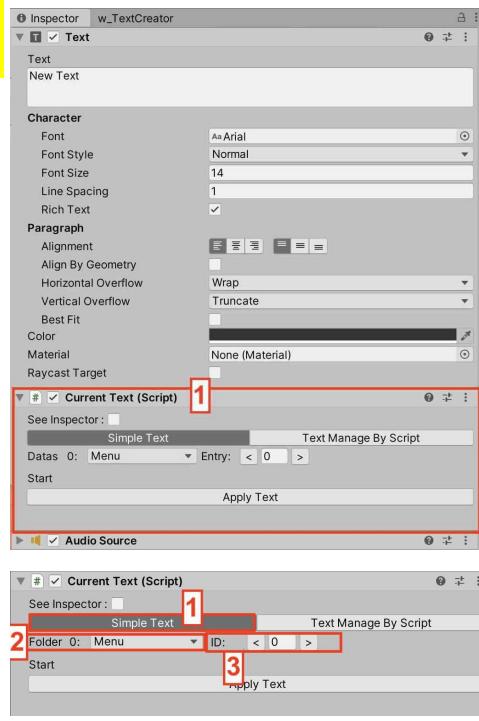


How to: Setup a text using the Multi-language system

-Select a **text object** in the hierarchy that contains the script **CurrentText** (spot 1).

Or

-Select a **text object** in the hierarchy and add the script **CurrentText** to the object (spot 1).



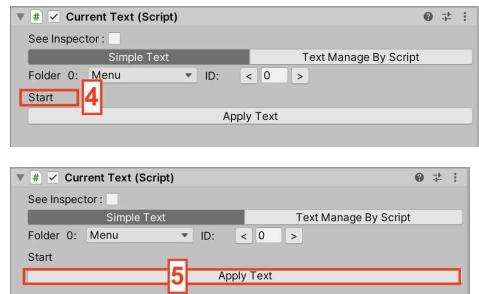
-Press **Simple Text** (spot 1)

-Select the **Folder** that contains the text (spot 2).

-Select the **text ID** (spot 3)

The selected text is displayed (spot 4)

-Press **Apply Text** to apply the text in scene view (spot 5)



Scripting

New multi-language text

NewTextWithSpecificID(int newID, int textFolder)

New multi-language text displayed letter by letter (To use audio add the AudioSource component)

NewTextWithSpecificID_LetterByLetter(int newID, int textFolder, AudioClip aClip = null, bool b_IsSentenceInProcess = true).

Player Progression

This section describes the system to save and load player progression information.

Table of contents:

Overview

[Link](#)

How to: Save and Load a player progression data

[Link](#)

Overview

Player progression is managed in the hierarchy by the object **InfoPlayer**

(Hierarchy: ALL SCENES → GAME INFO → InfoPlayer)

The object **InfoPlayer** contained the script **LoadSavePlayerProgression.cs** that allows to save and load the player progression.

By default Player Progression load and save: vehicle info, Game Mode info (championship, Arcade, Time Trial), coins.

When the game is saved the methods contained in **Method to save Player progression** are called (spot 1).

Each method save a player parameter.
(example: the number of coins available to the player)

When the scene is initialized the methods contained in **Method to load Player progression** are called (spot 2):

Each method load a player parameter.
(example: the number of coins available to the player)

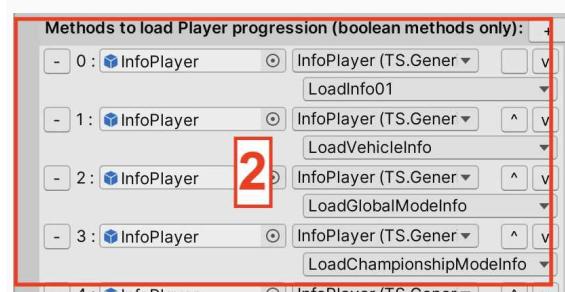
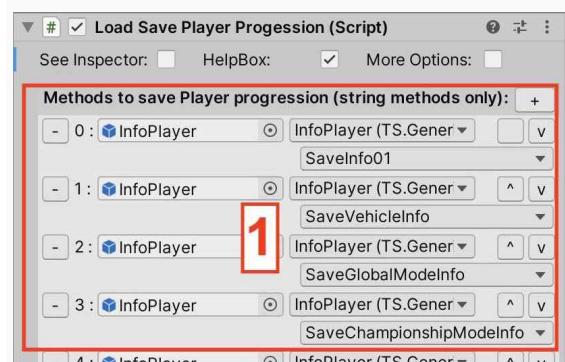
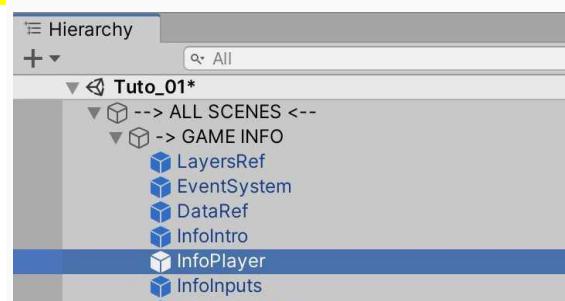
IMPORTANT:

The player progression parameters **MUST** be in the same order in the save and the load section.

Example:

The method to save the number of coins is on the last position of the save list.

The method to load the number of coins **MUST** be on the last position of the load list.



How to: Save and load custom player data

This section gives you an example to save and load a custom player data.

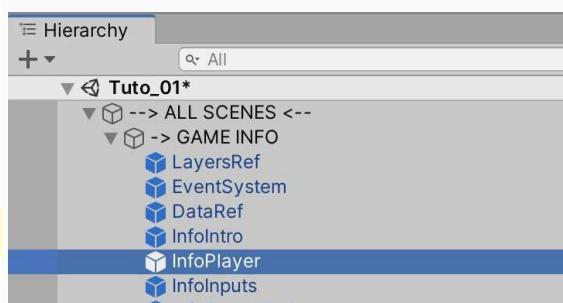
As an example we are going to:

- Save the current month
- Load the current month

Save a player progression parameter

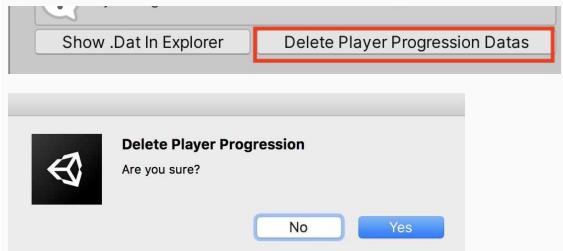
-In the Hierarchy select **InfoPlayer**.

(Hierarchy: ALL SCENES → GAME INFO → InfoPlayer)



-In the Inspector press button

Delete Player Progression Data.



-Press **Yes**

-Press **+** button (spot 1)

to add a new method that save the new player progression parameter.

-Drag and drop the object from the Hierarchy that contains the method to use in the empty field.
In this example the script is attached to **InfoPlayer** (spot 2).

Tips: InfoPlayer is not destroyed when a new scene is loaded. So I suggest you to attached the script to **InfoPlayer** to be sure that the script you want to use is always available in the scene.

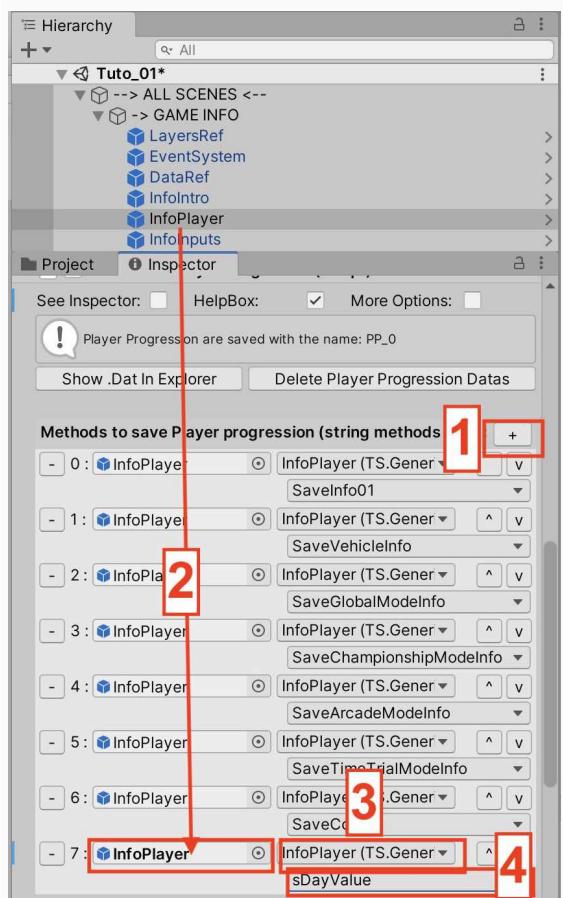
-Select the script that contains the method to use.
In this example choose

LoadSavePlayerProgressionAssistant (spot 3).

-Select the method to use.

In this example choose **sDayValue**

The method return the month of the year as a string.
(spot 4)



VERY IMPORTANT

You **MUST** use a method that return a string.

Load a player progression parameter

-In the Hierarchy select **InfoPlayer**.

(Hierarchy: ALL SCENES → GAME INFO → InfoPlayer)

-Press + button (spot 1)
to add a new method that load the new player progression parameter.

-Drag and drop the object from the Hierarchy that contains the method to use in the empty field.
In this example the script is attached to InfoPlayer (spot 2).

Tips: InfoPlayer is not destroyed when a new scene is loaded. So I suggest you to attached the script to InfoPlayer to be sure that the script you want to use is always available in the scene.

-Select the script that contains the method to use.
In this example choose **LoadSavePlayerProgressionAssistant** (spot 3).

-Select The method to use.
In this example choose **bUpdateMonthValue** (spot 4)

VERY IMPORTANT

A boolean method that return true at the end **MUST** be used.

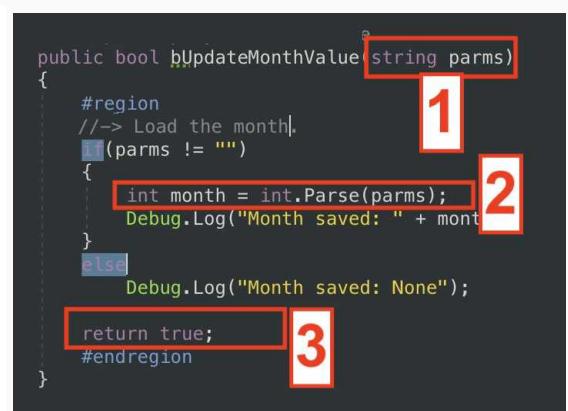
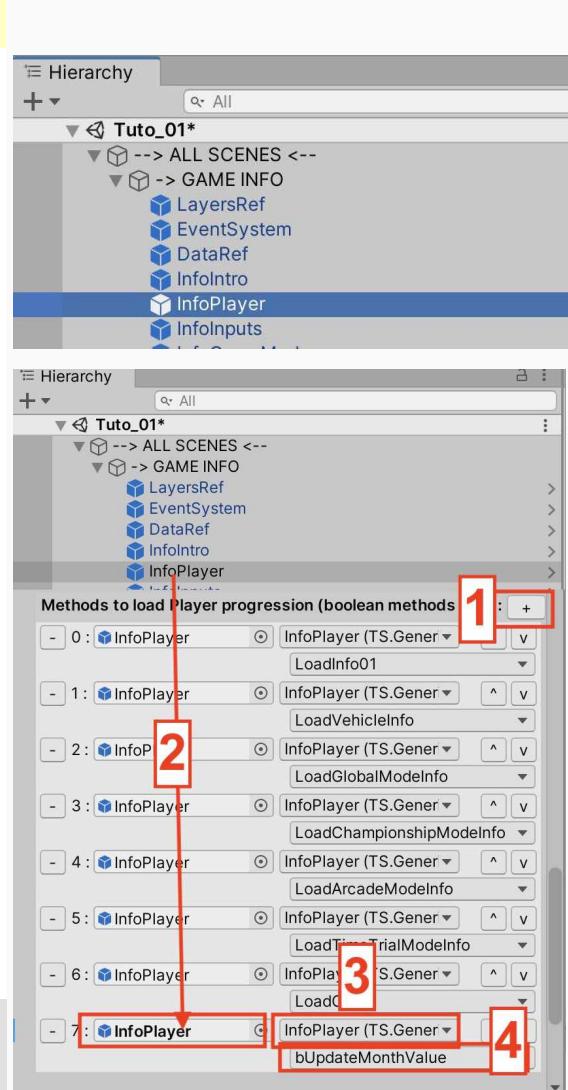
Info:

The method get the parameter with its argument (spot 1)

The parameter is converted to an int (spot 2)

If a month has been saved, the month is displayed in the console tab (spot 3)

At the end the method return true (spot 4)



Conclusion:

Now when the game is saved: The month is saved.

When the game is loaded: The month is displayed in the console tab.

Inputs

This section describes the system that manage inputs in the asset.

Table of content:

Overview	Link
How to: Reset inputs	Link
How to: Delete existing input saved data	Link
How to: Create a new input	Link
How to: Create a boolean parameter (switch button)	Link
How to: Create a float parameter (slider)	Link
Remap Menu: How to set up an input, a bool or a float	Link
Scripting: Useful methods to access the inputs	Link
How to: Change validation button ID	Link
How to: Change back button ID	Link
How to: Change pause button ID	Link

Overview

Inputs are separated into two parts:

-InfoInputs (spot 1)

(Hierarchy: ALL SCENES → GAME INFO → InfoInputs)

This object manages the inputs in the game.

It contains:

Load/Save inputs methods.

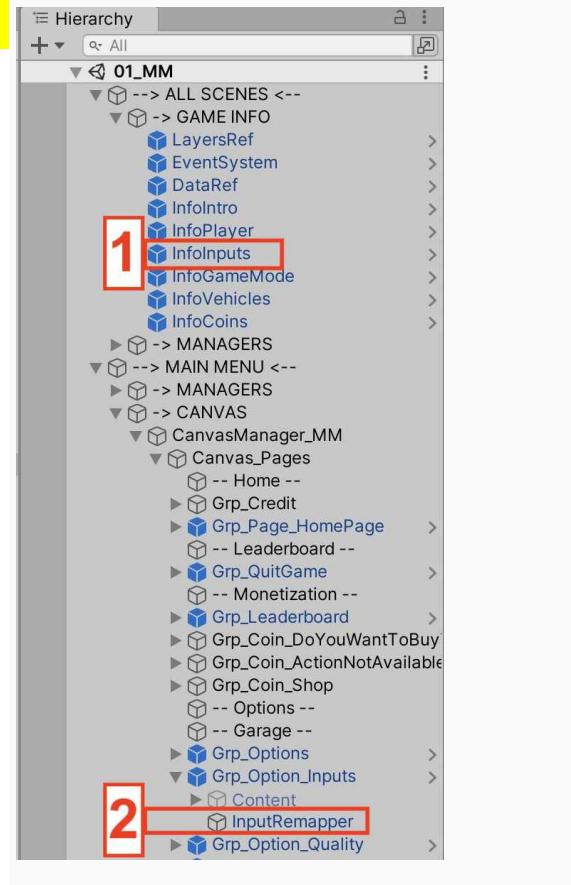
Methods to access inputs from any scripts.

-InputRemapper (spot 2).

In the Main Menu scene: (Hierarchy: MAIN MENU → CANVAS → CanvasManager_MM → Canvas Pages → Grp_Option_Inputs → InputRemapper))

In a Gameplay scene: (Hierarchy: GAMEPLAY → CANVAS → CanvasManager_TM → Canvas Pages → Grp_Option_Inputs_IG → InputRemapper))

It manages the input remapper in the Input menu page.



How to: Reset inputs

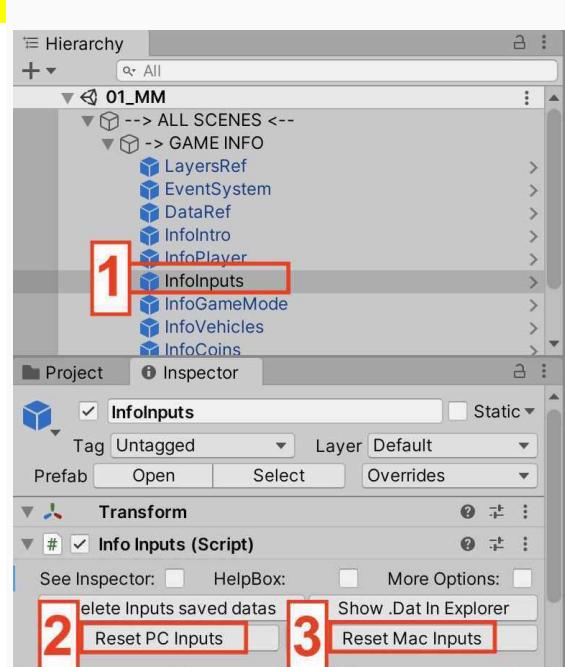
-In the Hierarchy select **InfoInput** (spot 1).

Press:

-Button **Reset PC** (spot 2)

-Button **Reset Mac** (spot 3)

Its resets all players inputs at the same time.

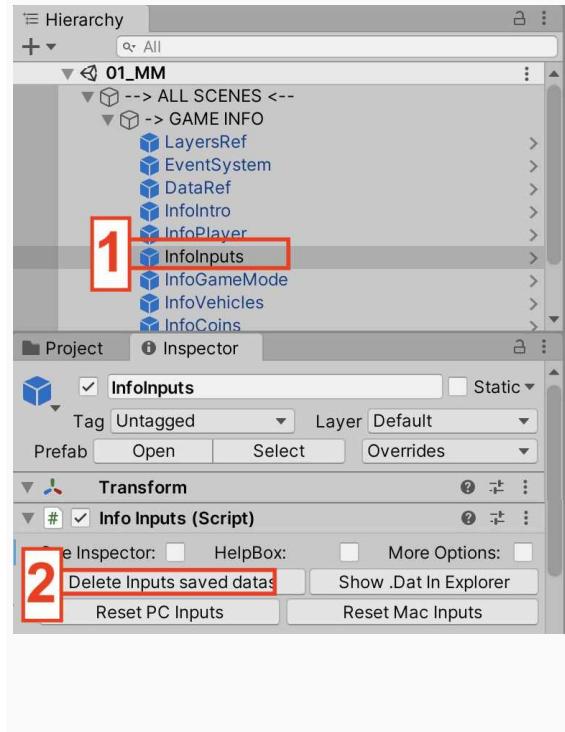


How to: Delete existing input saved data

-In the Hierarchy select **InfoInput** (spot 1).

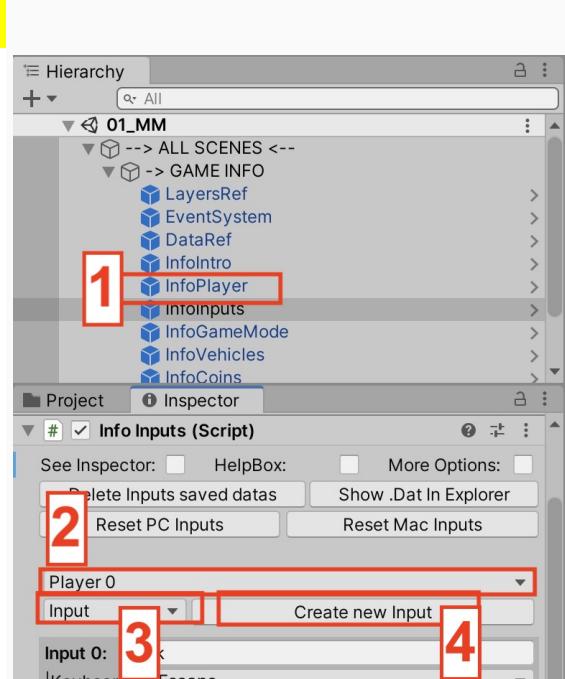
Press:

-Button **Delete Inputs saved datas** (spot 2).

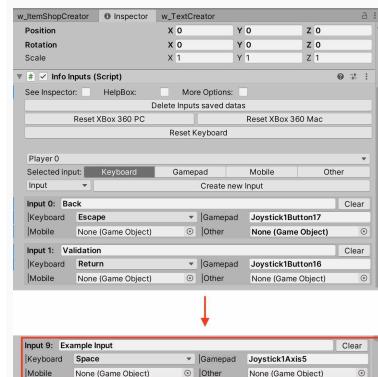


How to: Create a new input

- In the Hierarchy select **InfoInput** (spot 1).
- In the Inspector select the **player 0** in the dropdown menu (spot 2).
- Select **Input** in the dropdown menu (spot 3).
- Press button **Create new Input** (spot 4).



A new input is created at the end of the list.



- Write a name for the new input (spot 1).

This name is only use in the editor (not at runtime).

- With the dropdown menu select the button used by the keyboard for this input (spot 2).

- Write the name of the gamepad input used for this input (spot 3).

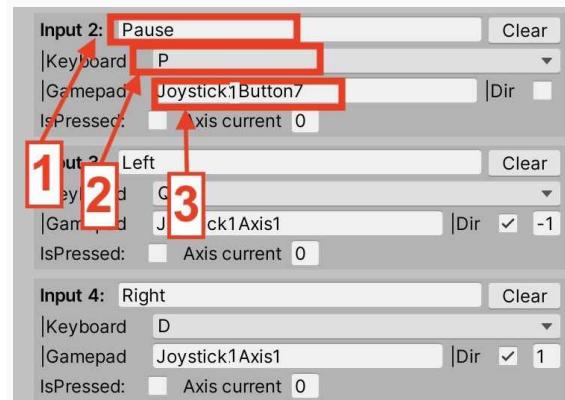
Examples:

To use the Player 1 Axis 0 write: Joystick1Axis0

To use the Player 1 Button 0 write: Joystick1Button0

You can find all the input available here:

Edit → PlayerSettings → Input Manager



IMPORTANT

The input must exist into the Unity InputManager. If the input doesn't exist you will have error when the scene starts.

-If the input corresponds to a button:

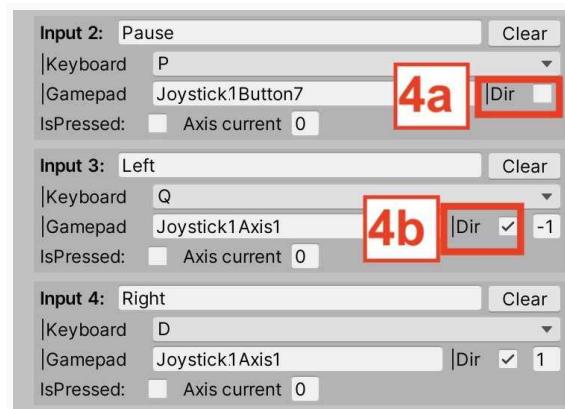
(For example Pause Joystick1Button7)

Uncheck box Dir (spot 4a)

-If the input corresponds to an axis:

(For example Left with Joystick1Axis1)

Check box Dir (spot 4b)



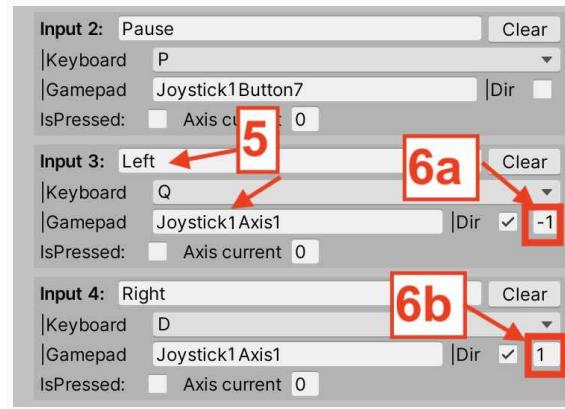
-Value spot 6a and 6b correspond to values returned by the input when the axis is used by the player.

For example: When the player wants to turn left he moves the axis Joystick1Axis1 on the left (spot 5). The value returned by the Unity Input system is -1.

When the player wants to turn right he moves the axis Joystick1Axis1 on the right.

The value returned by the Unity Input system is 1.

Choose 1 or -1 depending the value returned by your axis.



VERY IMPORTANT:

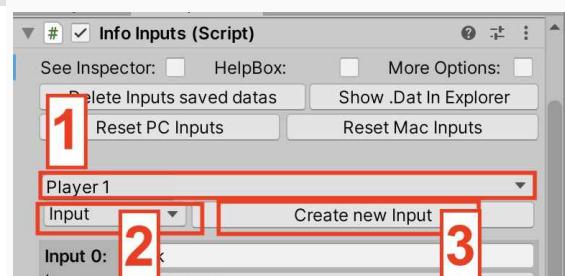
The process to create an input for player 0 is finished. Now you **MUST** create the same input for player 1.

If the same input is not created for player 1 you will have issues at runtime.

-In the Inspector select **player 1** in the dropdown menu (spot 1).

-Select **Input** in the dropdown menu (spot 2).

-Press button **Create new Input** (spot 3).



A new input is created at the end of the list.

-Write a name for the new input (spot 1).

This name is only use in the editor (not at runtime).

-With the dropdown menu select the button used by the keyboard for this input (spot 2).

-Write the name of the gamepad input used for this input (spot 3).

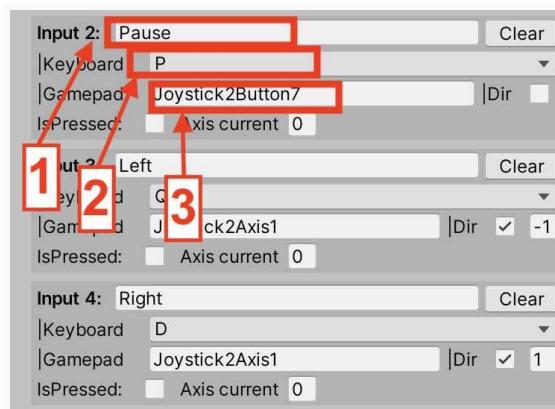
Examples:

To use the Player 2 Axis 0 write: Joystick2Axis0

To use the Player 2 Button 0 write: Joystick2Button0

You can find all the input available here:

Edit → PlayerSettings → Input Manager



IMPORTANT

The input must exist into the Unity InputManager. If the input doesn't exist you will have error when the scene starts.

-If the input corresponds to a button:

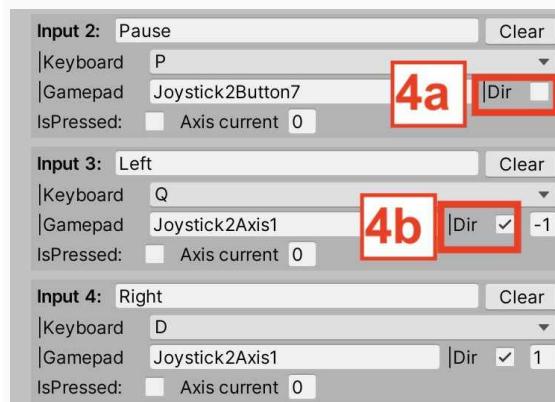
(For example Pause Joystick2Button7)

Uncheck box Dir (spot 4a)

-If the input corresponds to an axis:

(For example Left with Joystick2Axis1)

Check box Dir (spot 4b)



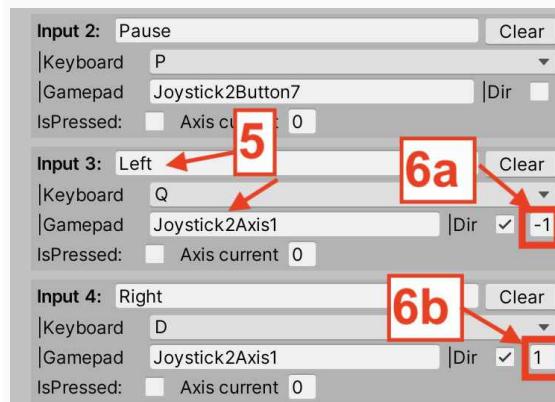
-Value spot 6a and 6b corresponds to value returned by the input when the axis is used by the player.

For example: When the player wants to turn left he moves the axis Joystick2Axis1 on the left (spot 5). The value returned by the Unity Input system is -1.

When the player wants to turn right he moves the axis Joystick2Axis1 on the right.

The value returned by the Unity Input system is 1.

Choose 1 or -1 depending the value returned by your axis.



The new input is now ready to use.

More info about: How to setup an input in the Input Menu page → Page [Link](#)

More info about: How to use an input with script
→ Page [Link](#)

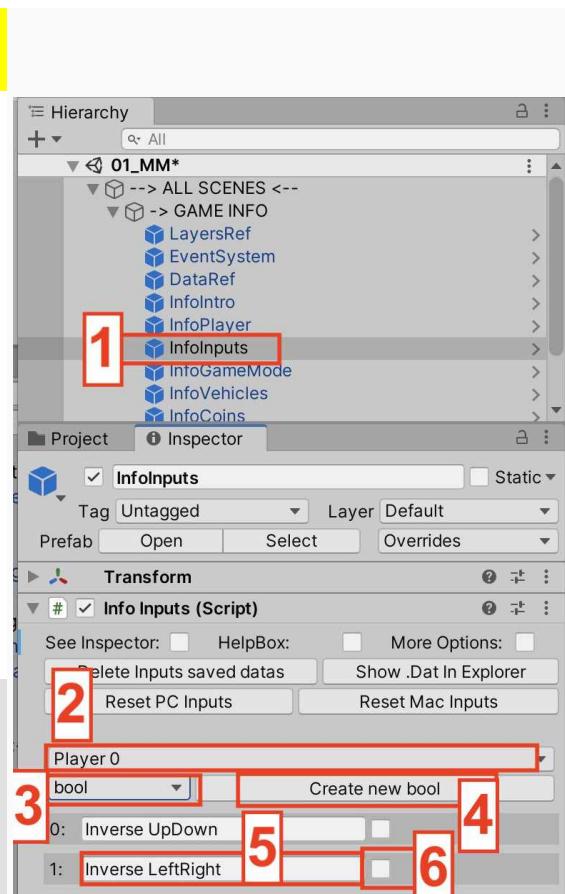
How to: Create a boolean parameter (switch button)

- In the Hierarchy select **InfoInput** (spot 1).
- In the Inspector select the **player 0** in the dropdown menu (spot 2).
- Select **bool** in the dropdown menu (spot 3).
- Press button **Create new bool** (spot 4).
- Rename the bool (spot 5). The name is only used in the editor not at runtime.
- Check the box (spot 6) if the boolean must be enabled by default.

VERY IMPORTANT

The process to create a bool for **player 0** is finished. Now you **MUST** create the same bool for **player 1**.

If the same bool is not created for player 1 you will have issues at runtime.

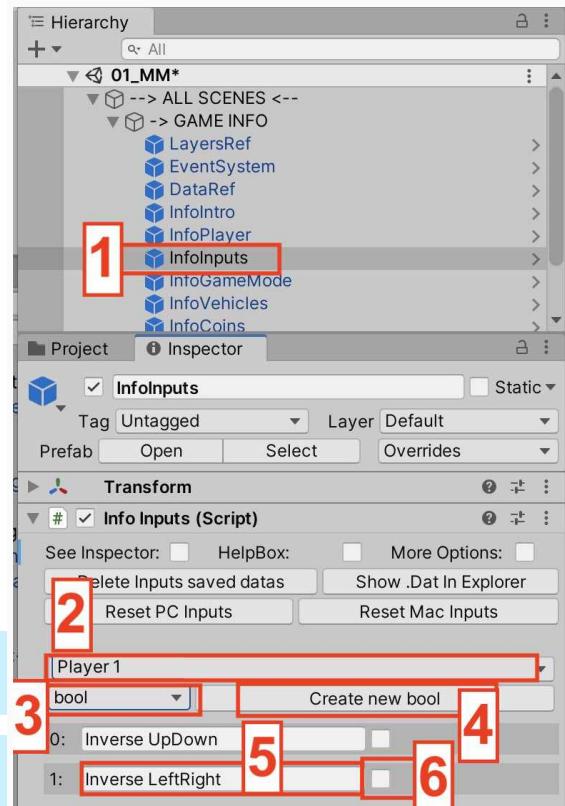


- In the Hierarchy select **InfoInput** (spot 1).
- In the Inspector select the **player 1** in the dropdown menu (spot 2).
- Select **bool** in the dropdown menu (spot 3).
- Press button **Create new bool** (spot 4).
- Rename the bool (spot 5). The name is only used in the editor not at runtime.
- Check the box (spot 6) if the boolean must be enabled by default.

The new input is now ready to use.

More info about: How to setup an input in the Input Menu page → Page [Link](#)

More info about: How to use an input with script → Page [Link](#)



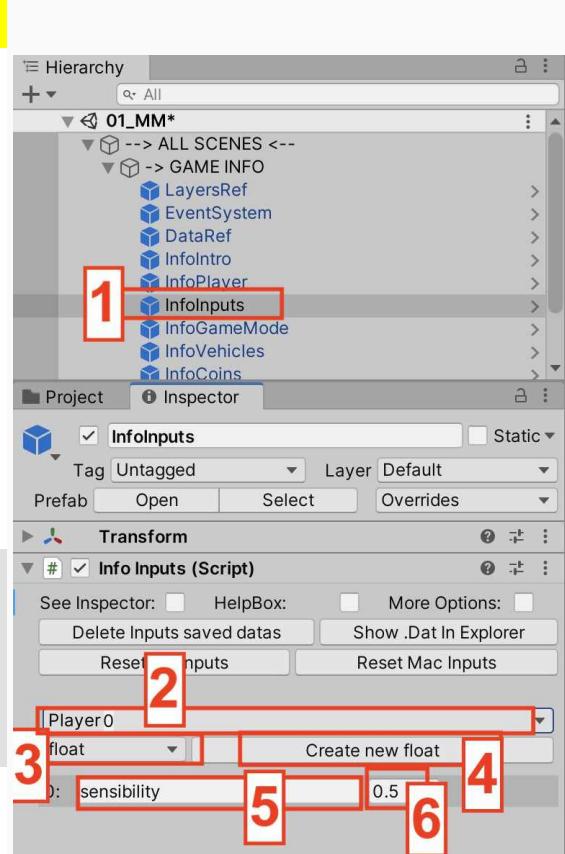
How to: Create a float parameter (slider)

- In the Hierarchy select **InfoInput** (spot 1).
- In the Inspector select the **player 0** in the dropdown menu (spot 2).
- Select **float** in the dropdown menu (spot 3).
- Press button **Create new float** (spot 4)
- Rename the float (spot 5). The name is only used in the editor not at runtime.
- Choose the default value for the float (spot 6).

VERY IMPORTANT

The process to create a float for **player 0** is finished. Now you **MUST** create the same float for **player 1**.

If the same float is not created for player 1 you will have issues at runtime.

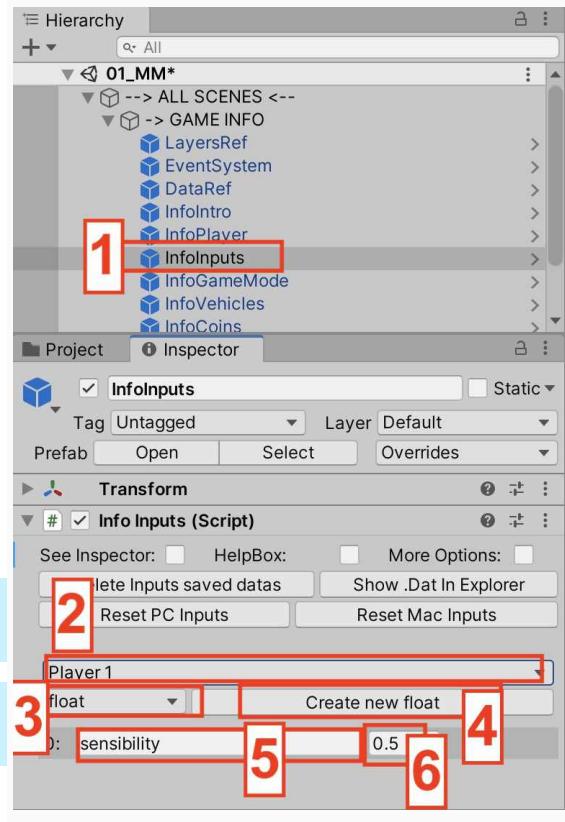


- In the Hierarchy select **InfoInput** (spot 1).
- In the Inspector select the **player 1** in the dropdown menu (spot 2).
- Select **float** in the dropdown menu (spot 3).
- Press button **Create new float** (spot 4).
- Rename the float (spot 5). The name is only used in the editor not at runtime.
- Choose the default value for the float (spot 6).

The new input is now ready to use.

More info about: How to setup an input in the Input Menu page → Page [Link](#)

More info about: How to use an input with script
→ Page [Link](#)



Remap Menu: How to set up an input, a bool or a float

As an example we are going to create a new button in the remap menu. This button will allow to remap the PowerUp button.

The process is the same for a bool or a float value.



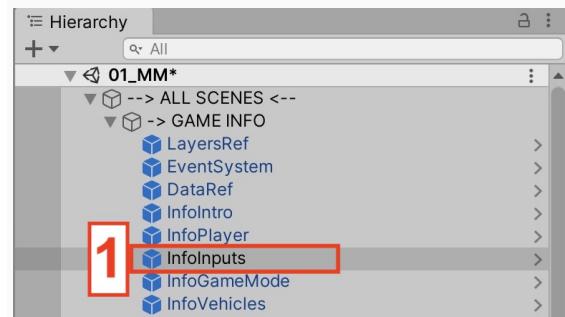
-In Project Tab open the **Main Menu** scene.

(Assets → Scenes → MainMenu → 01_MM)

First of all we need to know the ID of our input.

-In the Hierarchy select **InfoInputs** (spot 1).

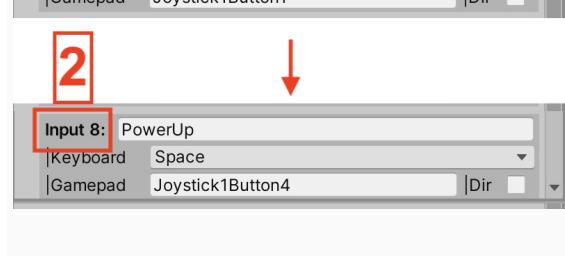
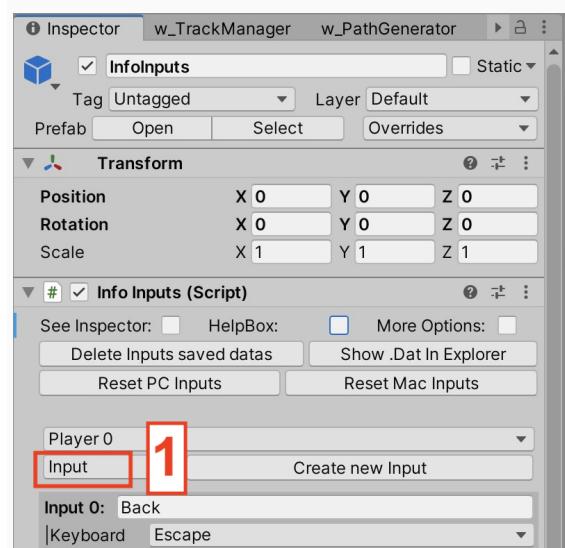
(Hierarchy: ALL SCENES → GAME INFO → InfoInputs)



-In the Inspector select **Input** in the dropdown menu (spot 1)

-In the list **Power Up** is the **Input 8** (spot 2).

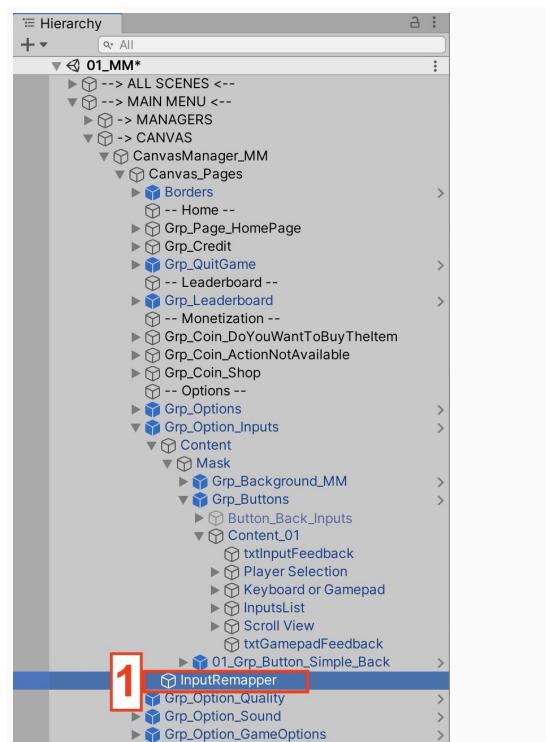
The **ID = 8**.



We are going to create the keyboard button.

-In the Hierarchy select **InputRemapper** (spot 1).

(Hierarchy: MAIN MENU → CANVAS → CanvasManager_MM → CanvasPages → Grp_Option_Inputs → InputRemapper)



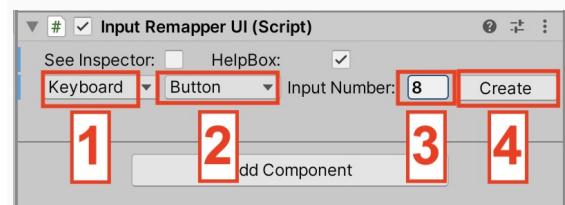
-In the Inspector select:

Keyboard (spot 1)

Button (spot 2)

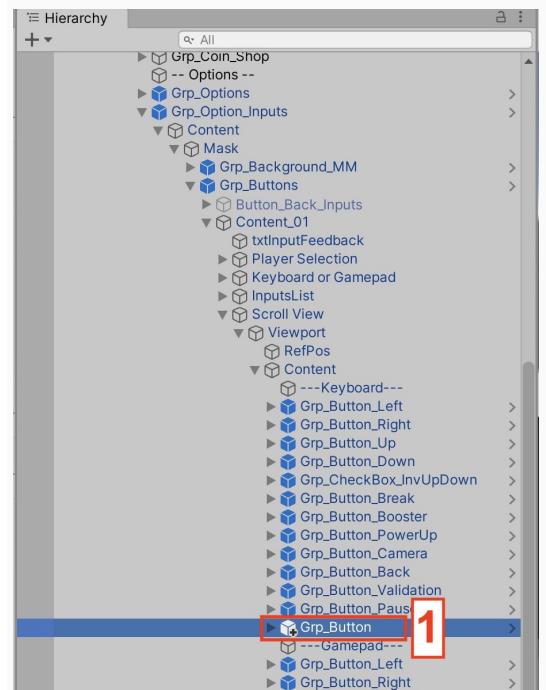
Input Number = 8 (spot 3)

-Press **Create** button (spot 4)



The button is created in the Hierarchy (spot 1).

(Hierarchy: MAIN MENU → CANVAS → CanvasManager_MM → CanvasPages → Grp_Option_Inputs → Content → Mask → Grp_Buttons → Content_01 → ScrollView → Viewport → Content → Grp_Button)



We are going to create the Gamepad button.

-In the Hierarchy select **InputRemapper** (spot 1).

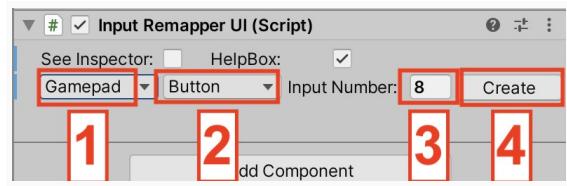
(Hierarchy: MAIN MENU → CANVAS → CanvasManager_MM → CanvasPages → Grp_Option_Inputs → InputRemapper)

-In the Inspector select:
Gamepad (spot 1)
Button (spot 2)
Input Number = 8 (spot 3)

-Press **Create** button (spot 4)

The button is created in the Hierarchy (spot 1).

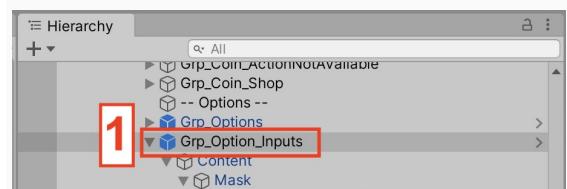
(Hierarchy: MAIN MENU → CANVAS → CanvasManager_MM → CanvasPages → Grp_Option_Inputs → Content → Mask → Grp_Buttons → Content_01 → ScrollView → Viewport → Content → Grp_Button)



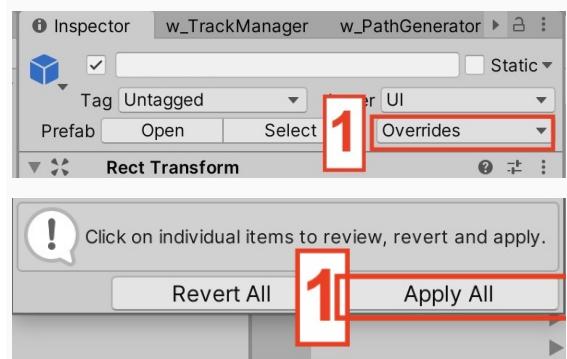
The 2 buttons are added inside a prefab so it is not possible to move them in the Hierarchy before overriding the prefab.

-In the Hierarchy select **Grp_Option_Inputs** (spot 1).
(Hierarchy: MAIN MENU → CANVAS → CanvasManager_MM → CanvasPages → Grp_Option_Inputs)

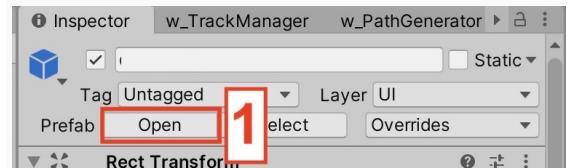
-In the Inspector press **Overrides** button (spot 1).



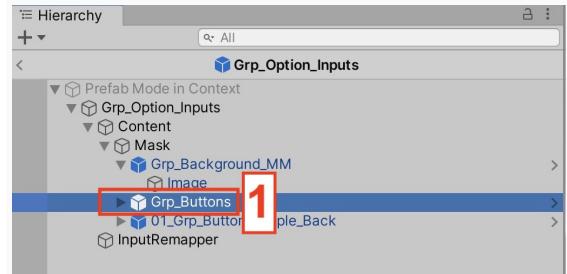
-Press **Apply All** (spot 1)



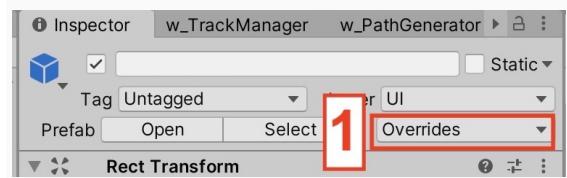
-Press **Open** button to open the prefab (spot 1)



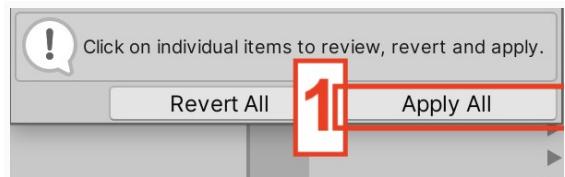
-In the Hierarchy select **Grp_Buttons** (spot 1)



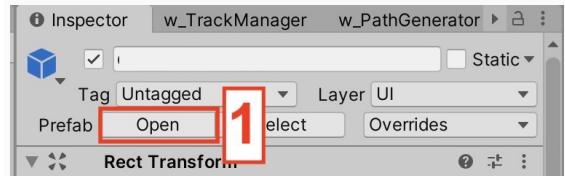
-In the Inspector press **Overrides** button (spot 1).



-Press **Apply All** (spot 1)



-Press **Open** button to open the prefab

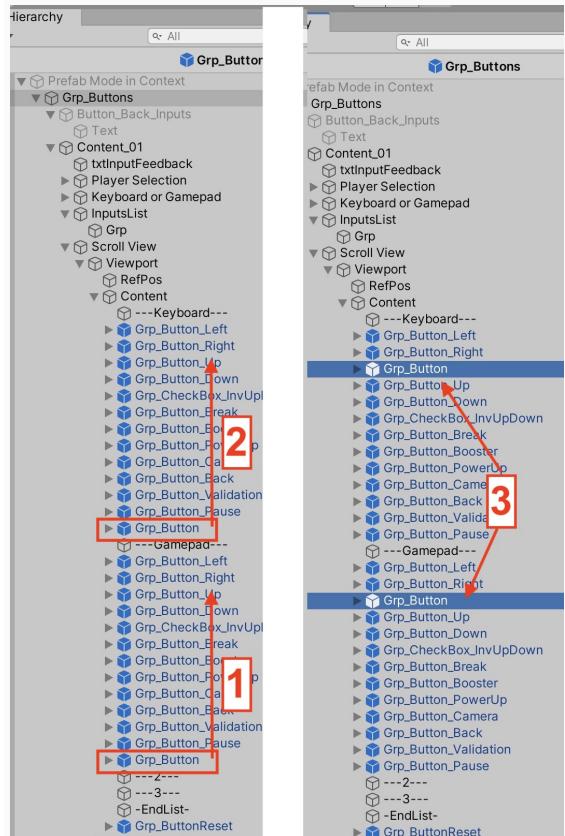


Now we can move our buttons in the list. We are going to move them after Left Input.

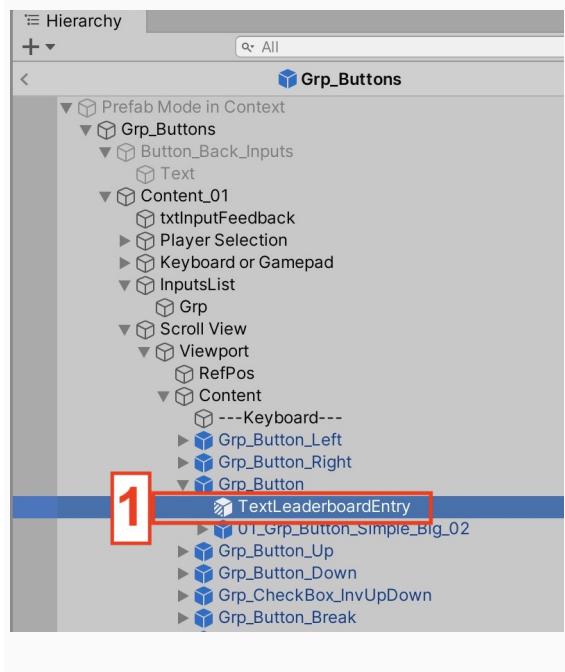
Drag and drop the **1st** button to **spot 1** position.

Drag and drop the **2nd** button to **spot 2** position.

Result spot 3.



-Inside the **1st** button select **TextLeaderboardEntry** object (spot 1)



-In the Inspector go to Current Text script.

-Choose ID = 10 (spot 1)

-Press Apply (spot 2)

-Inside the 2nd button select TextLeaderboardEntry object (spot 1)

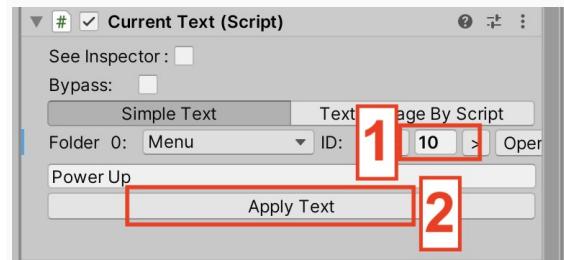
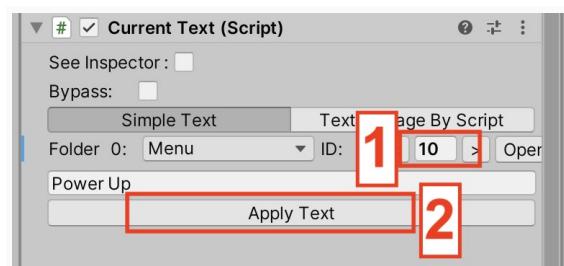
-In the Inspector go to Current Text script.

-Choose ID = 10 (spot 1)

-Press Apply (spot 2)

Now when Remap page is displayed new buttons are displayed after Right Input.

The process is the same for slider or checkbox (bool and float).



Scripting: Useful methods to access the inputs

Each input has delegates and variables associate to it.

As an example open InfoExample.cs script.

(Project tab → Assets → Scripts → TS → Inputs → InfoExample)

This script gives you the different possibilities to access an input.

Info: To test the script: In your scene (Main menu scene or a gameplay scene) attached the script to an empty objet.

Press Play.

You have info in the console tab when you press left input.

Other Example

You can have look to VehicleInputs.cs script.

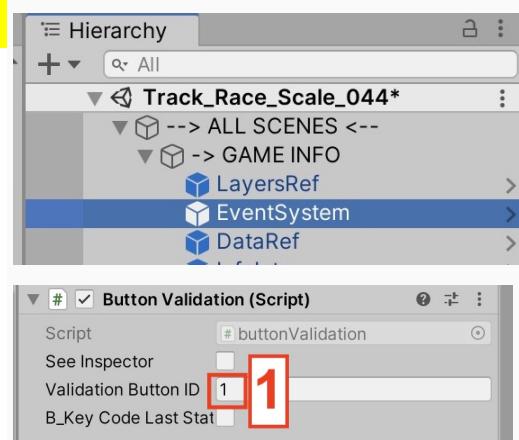
(Project tab → Assets → Scripts → TS → Vehicles → Vehicleinputs)

This script manages vehicle input.

How to: Change validation button ID

-In any scene select **EventSystem** in the Hierarchy.
(Hierarchy: ALL SCENES → GAME INFO → EventSystem)

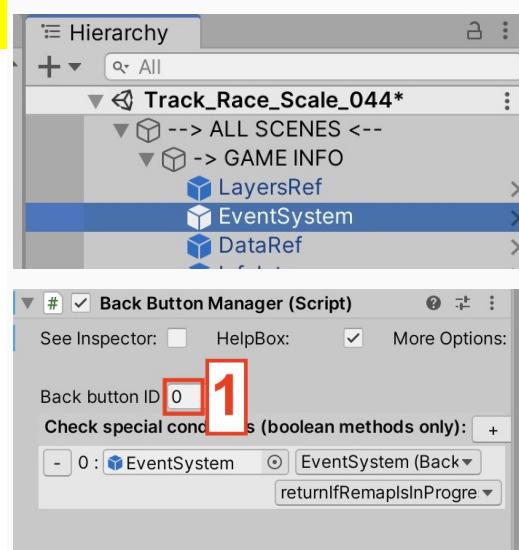
-In the Inspector go to **ButtonValidation** component
then choose your input ID (spot 1).



How to: Change back button ID

-In any scene select **EventSystem** in the Hierarchy.
(Hierarchy: ALL SCENES → GAME INFO → EventSystem)

-In the Inspector go to **Back Button Manager**
component
then choose your back button input ID (spot 1).



How to: Change pause button ID

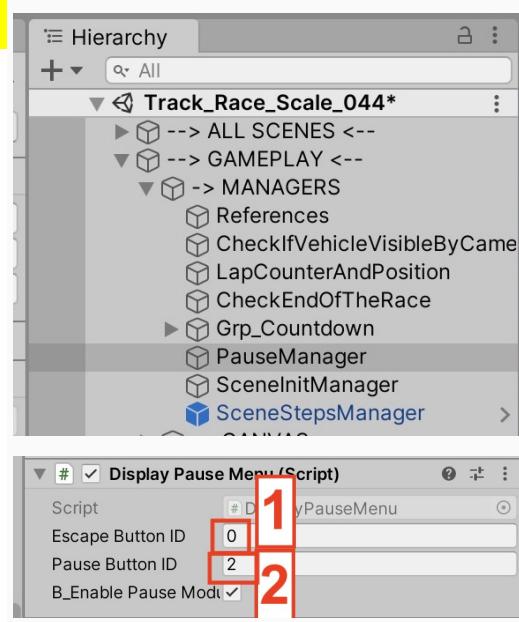
-In a gameplay scene select **PauseManager** in the Hierarchy.

(Hierarchy: GAMEPLAY → MANAGERS → PauseManager)

-In the Inspector go to **Display Pause Menu**

component then choose:

- The Escape button input ID (spot 1).
- The Pause button input ID (spot 2).



Pause Manager

This section describes **PauseManager**. This object manages pause in gameplay scene.
(Hierarchy: GAMEPLAY → MANAGERS → PauseManager)

Table of content:

Overview	Link
How to: Add a custom event when Pause is called	Link
How to: Add a custom event when UnPause is called	Link
How to: Change inputs used for Pause	Link
How to: Choose the pages that trigger the pause	Link
Scripting	Link

Overview

Pause manager (spot 1) is used to manage game pause. (Hierarchy: GAMEPLAY → MANAGERS → PauseManager)

(By default Pause Manager is enabled in track scenes. Pause Manager doesn't exist in the Main Menu Scene)

When the player presses Pause button the script **PauseManager.cs** is called. (By default Escape or P).

When Pause is called:

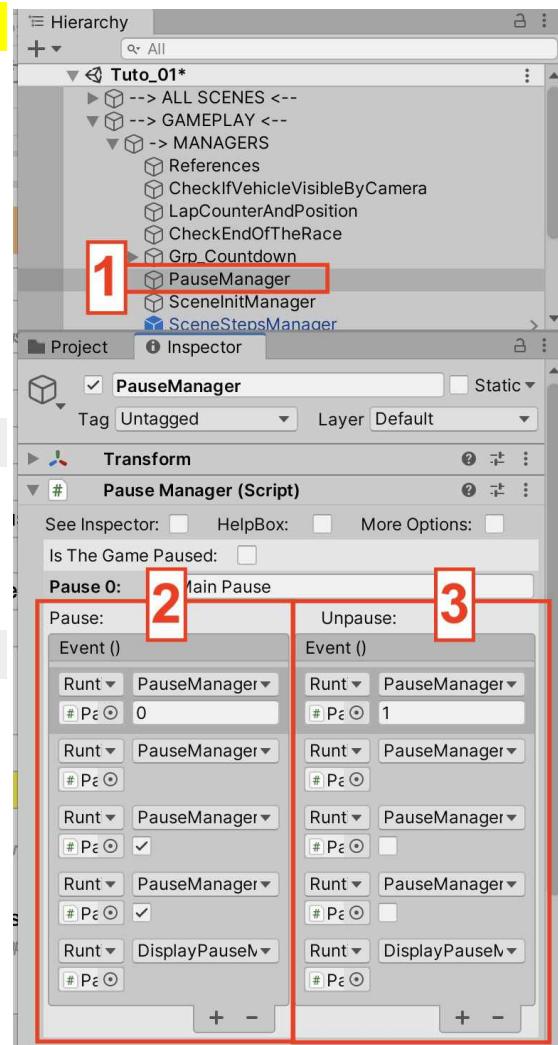
- Events into **Pause()** section are called (spot 2).
- Delegate named **OnPause** is invoked.

When UnPause is called:

- Events into **UnPause()** section are called (spot 3).
- Delegate named **UnPause** is invoked.

By default, during the pause:
animations, audio and particles are paused.

The object **PauseManager** is destroyed when a new scene is loaded.



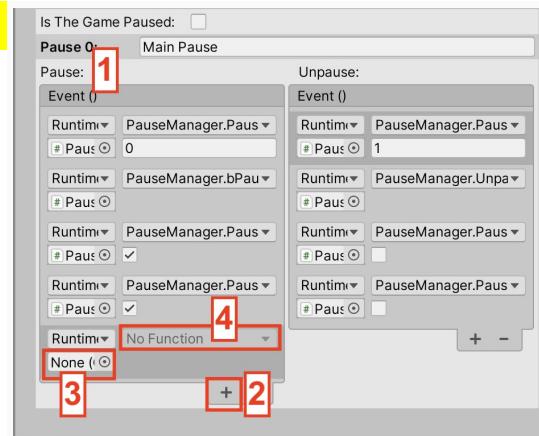
How to: Add a custom event when Pause is called

In section **Pause** (spot 1):

-Press **+** button (spot 2)

-Drag and drop an object from the hierarchy that contains the script that need to be use (spot 3).

-Select the method (spot 4)



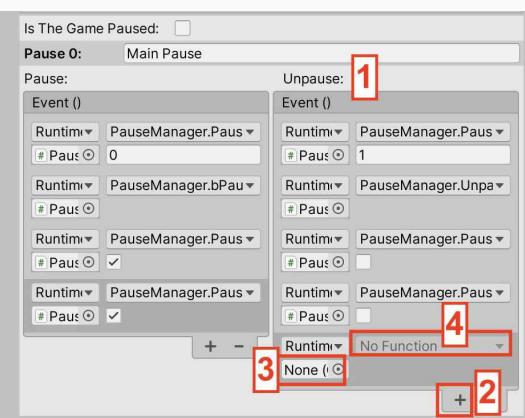
How to: Add a custom event when UnPause is called

In section **UnPause** (spot 1):

-Press **+** button (spot 2)

-Drag and drop an object from the hierarchy that contains the script that need to be use (spot 3).

-Select the method (spot 4)

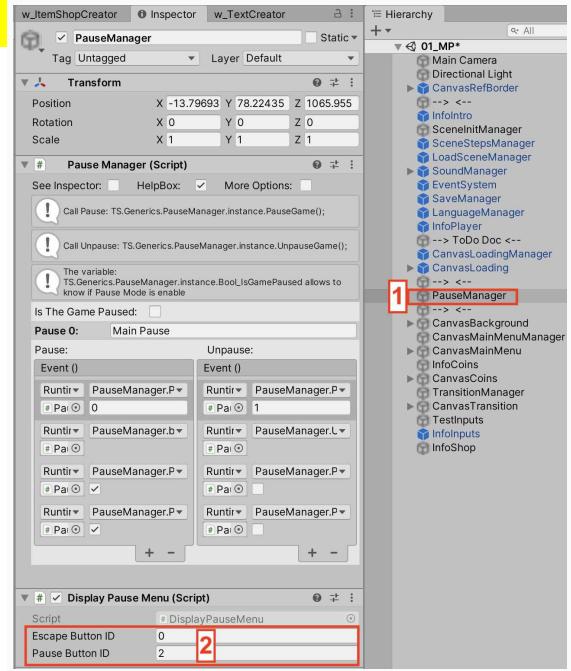


How to: Change inputs use for Pause

-In the Inspector select **PauseManager** (spot 1).
(Hierarchy: GAMEPLAY → MANAGERS → PauseManager)

-Select the input ID in field spot 2.

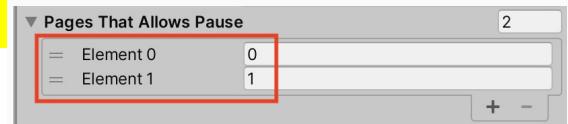
(Input ID can be found in object **InfoInputs**)



How to: Choose the pages that trigger the pause

-In the Inspector select **PauseManager**.

(Hierarchy: GAMEPLAY → MANAGERS → PauseManager)



In the Inspector, by default, **PagesThatAllowsPause** allows to open the pause menu if the player has page 0 or 1 opened.

It is possible to add or remove pages in the list.

Scripting

Pause the game:

```
TS.Generics.PauseManager.instance.PauseGame();
```

Unpause the game:

```
TS.Generics.PauseManager.instance.UnpauseGame();
```

Return if Pause is enable:

```
TS.Generics.PauseManager.instance.Bool_IsGamePaused
```

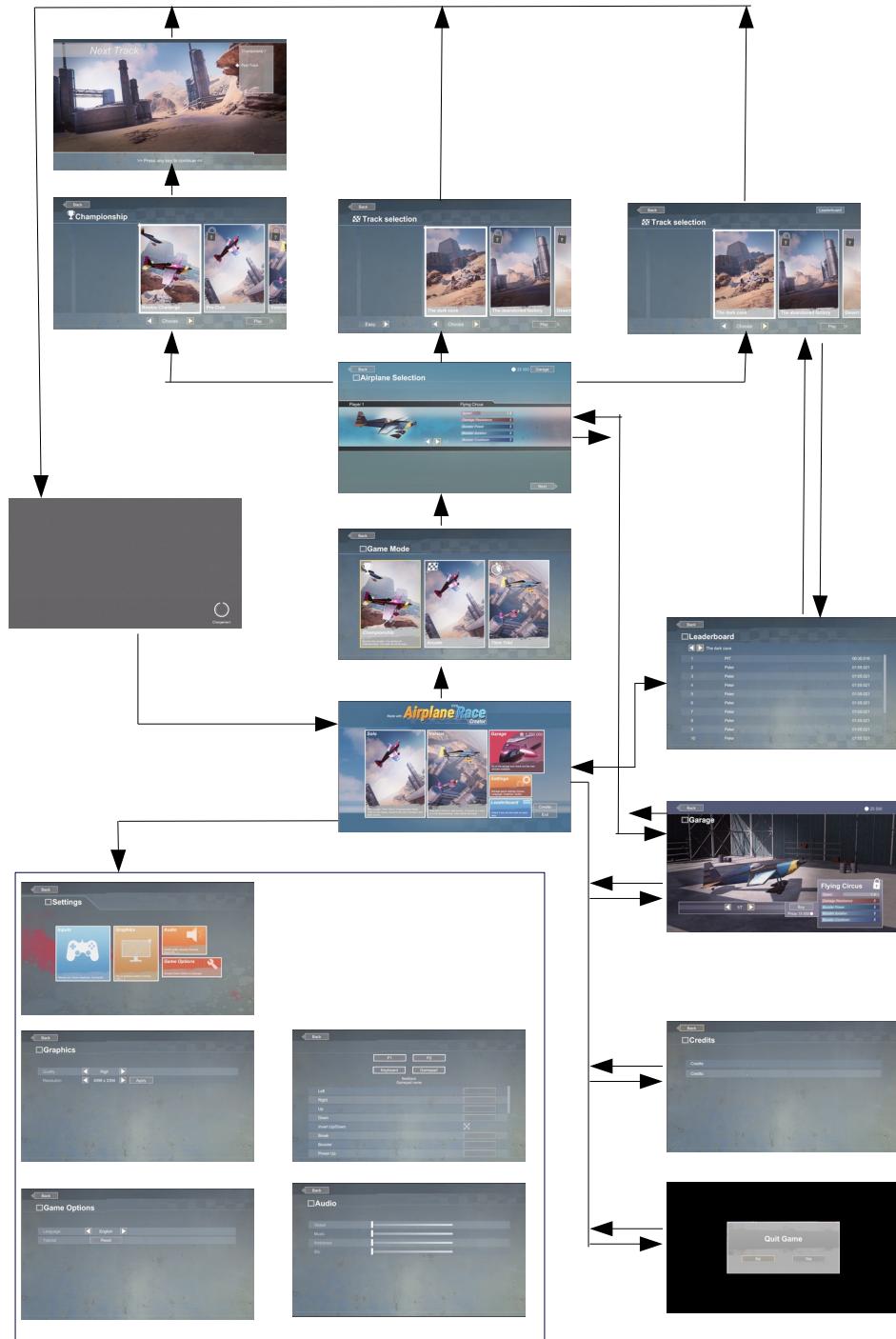
UI

This section describes the UI system included in the asset. This section explains how to customize the UI menus.

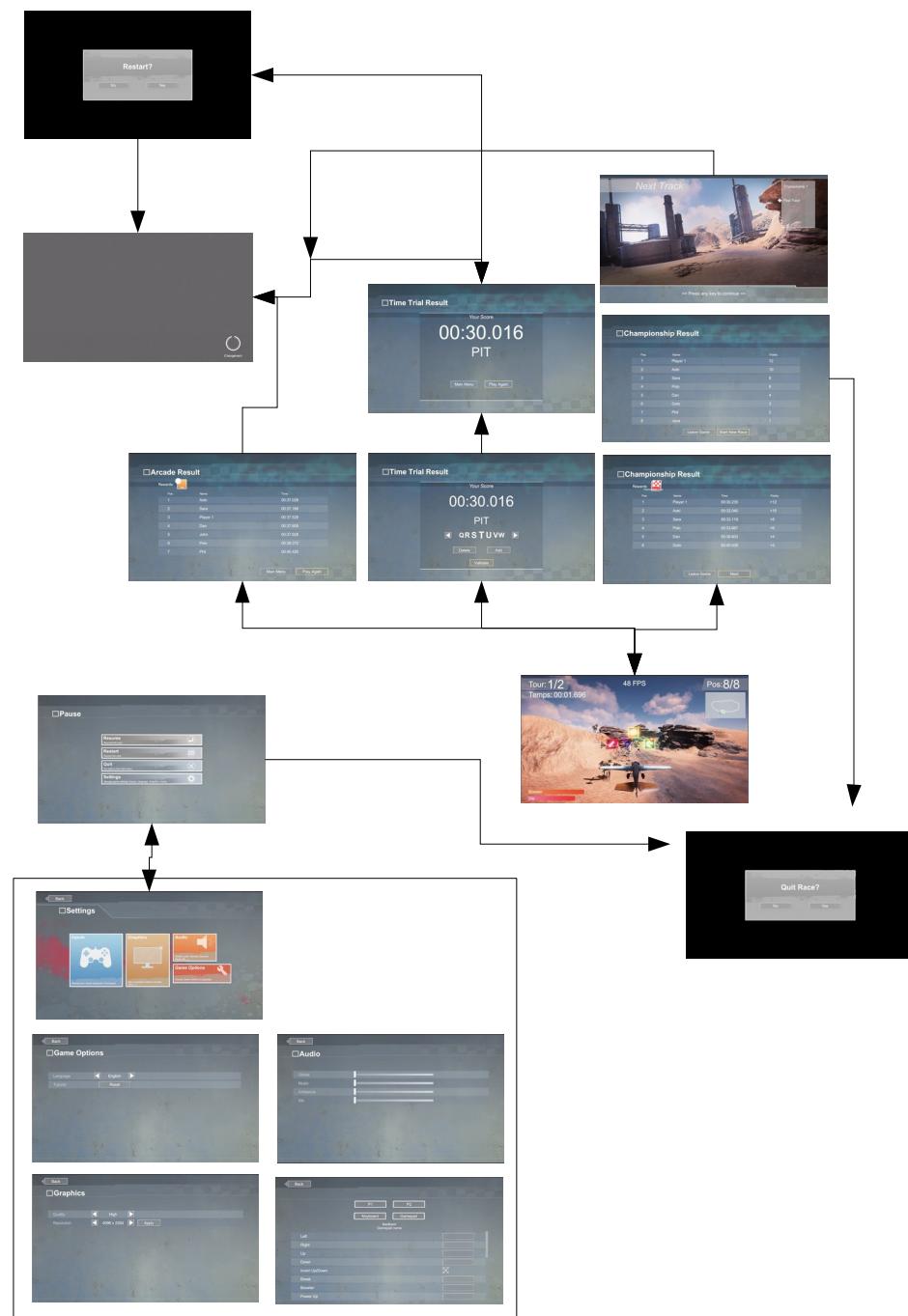
Table of contents:

Main Menu UI (Diagram)	Link
Race scene UI (Diagram)	Link
Global UI Parameters	Link
How to: Change the main menu scene ID	Link
Overview: The UI Page System	Link
How to: Create a new UI page	Link
How to: Create a button to go from one page to another	Link
How to: Add custom event during the initialization of a page	Link
Overview: Open/ Close page options	Link
Open a page: Options overview	Link
Close a page: Options overview	Link
How to: Add an event when a page is opened	Link
How to: Create method that can be used as an event when the page is opened	Link
How to: Add action when a page is closed	Link
How to: Create method used as an event when the page is closed	Link

Main Menu UI (Diagram)



Gameplay Menu (Diagram)



Global UI Settings

Info

Some user interface settings are common to the entire project.

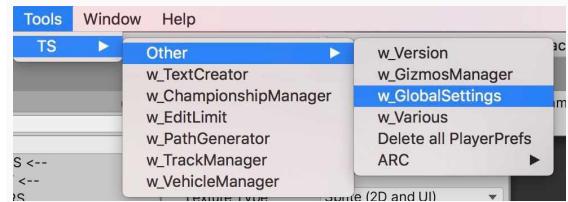
The **w_GlobalSettings** window allows to manage these settings.

To access the Global UI settings go to:

Tools → TS → Other → **w_GlobalSettings**

A new window appears.

This window displays the UI global settings available in the asset.



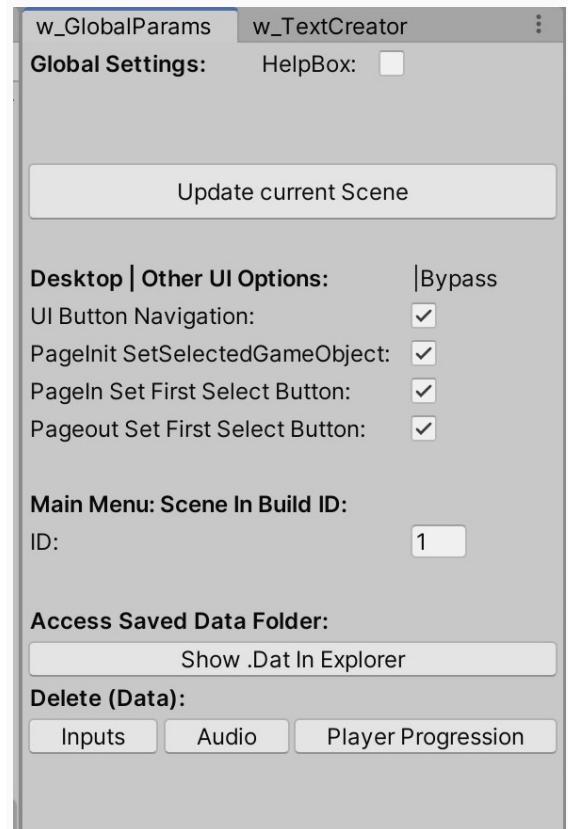
-The button **Update current scene** applies the modification to the current opened scene (spot 2).

-The other settings allow to choose the behavior of the buttons.

Main Menu: Scene In Build ID: Allows to choose the ID of the Main Menu scene. By default ID = 1.

Show .Dat In Explorer: Open the folder that contains the Player data.

Delete (3 buttons): Delete player data (Inputs, Audio, Player Progression).

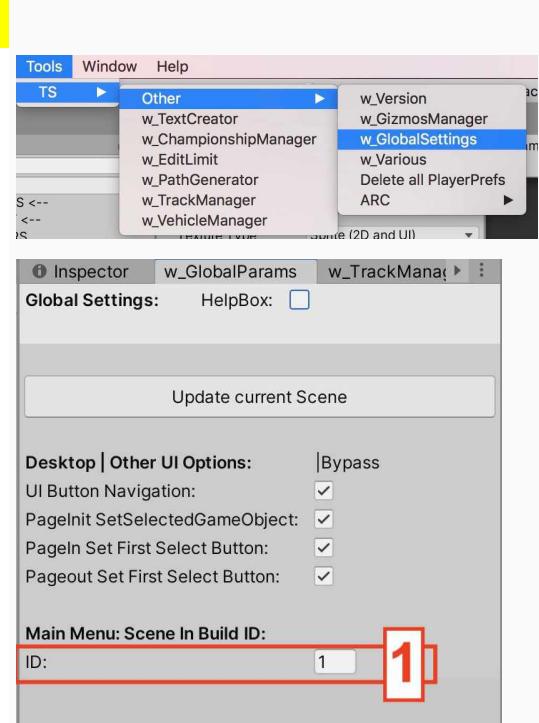


How to: Change the main menu scene ID

By default the Scene In Build 1 is used as the Main Menu. If you want to change the scene used for Main Menu follow these steps.

To access the Global UI settings go to:

Tools → TS → Other → w_GlobalSettings



A new window appears.

In **Main Menu: Scene InBuild ID** change the ID
(spot 1).

Overview: The UI Page System

The asset includes a system that allows to:

- Add pages in a UI menu.
- Manage the behavior of each page.

The page creation is managed by **CanvasManager_** in the Hierarchy (spot 1).

IMPORTANT:

In each scene UI are manage by the object named **CanvasManager_**.

In the Main Menu scene its name ends by **MM** (**CanvasManager_MM**).

In a Track scene its name ends by **TM** (**CanvasManager_TM**).

They are working the same way but they are using 2 different Prefabs.

To keep it simple in the documentation the name **CanvasManager_** is used for both.

Pages are contained in the Hierarchy into the object **Canvas_Pages** (spot 1).

Each page has its behavior. It is possible to:

- Manage how the page is initialized (spot 1).
*(By default the page is initialized using the script **PageInit.cs** when a new scene is loaded).*

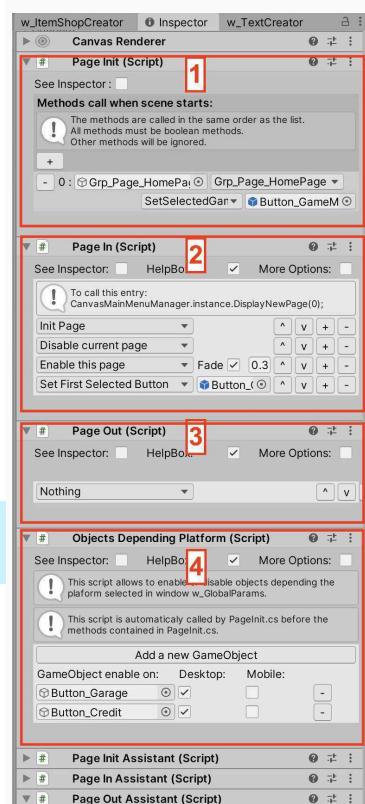
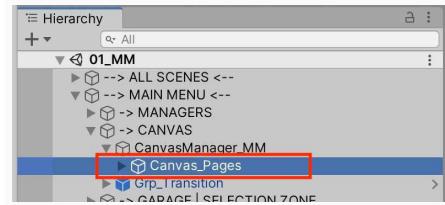
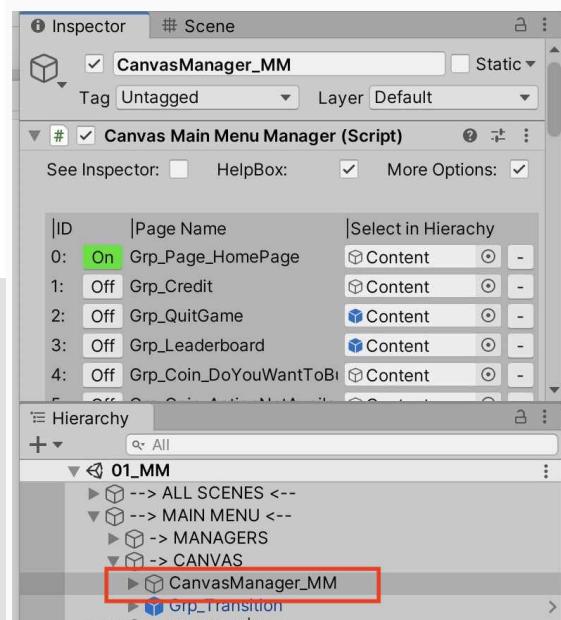
-Manage transition settings from one page to another page (spot 2) **PageIn.cs**.

-Manage transition settings to go to the previous selected pages (spot 3) **PageOut.cs**.

-Manage objects that needs to be enabled or disabled depending the selected platform (spot 4).

More info in section

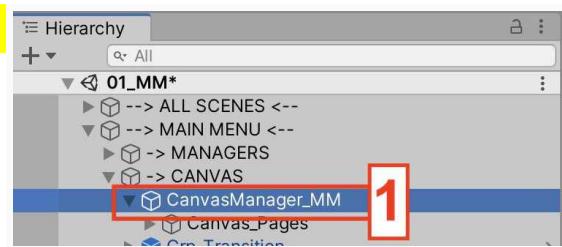
Overview: Open/ Close page options → [Link](#)



How to: Create a new UI Page

This section explains how to create a new menu page using the object **CanvasManager_** (Hierarchy).

-In the Hierarchy select **CanvasManager_** (spot 1)
(Hierarchy: MAIN MENU or GAMEPLAY → CANVAS → **CanvasManager_**)



-Go the end of the Inspector.

Write the name of the new page in the field (spot 2).
For the example write **Grp_MyNewPage**.

(You must choose a unique name for each menu page)

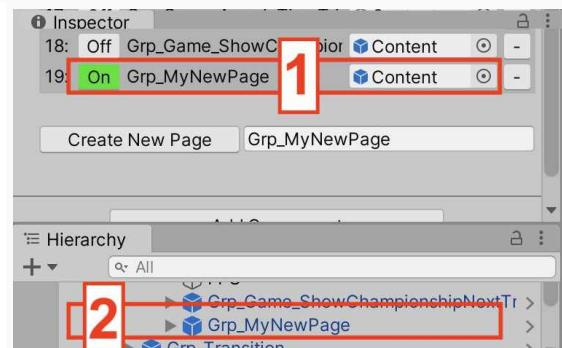


-In the Inspector press button **Create New Page** (spot 3).

Info:

The page is created inside object
Canvas_Pages (spot 1).

The new page is automatically selected as the current page
displayed in the Scene view (spot 2).



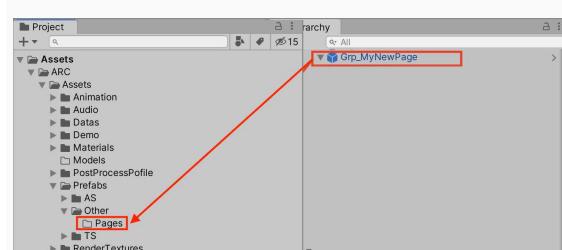
Info:

By default the page contains:

- A button to go to an other page (spot 1).
- A button to go to a previous page (spot 2).



-Drag and drop **Grp_MyNewPage** from the Hierarchy
to folder **Pages** in Project Tab.
(Assets → ARC → Assets → Prefabs → Other → Pages)



(A new window appears on screen)

-Press OriginalPrefab button (spot 1).

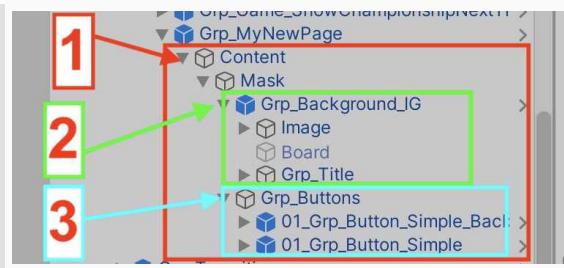
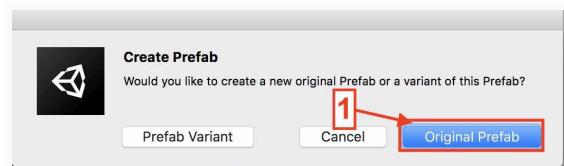
(The menu is now saved as a prefab)

IMPORTANT

-Put all your page contents inside the object **Content** (spot 1).

-Put your background contents inside **Grp_Background** (spot 2).

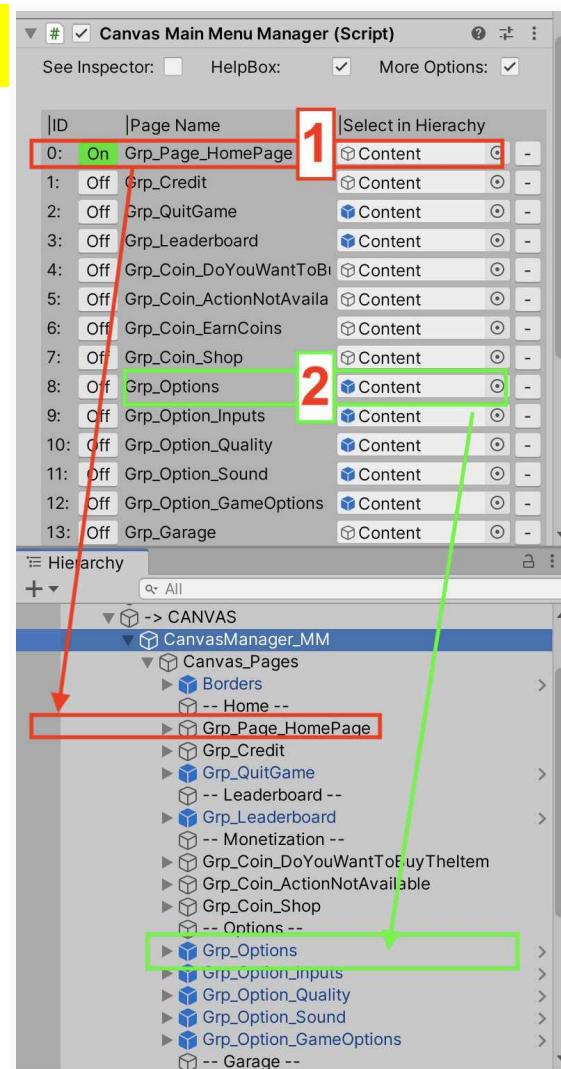
-Put your buttons contents inside **Grp_Butons** (spot 3).



How to: Create a button to go from one page to another

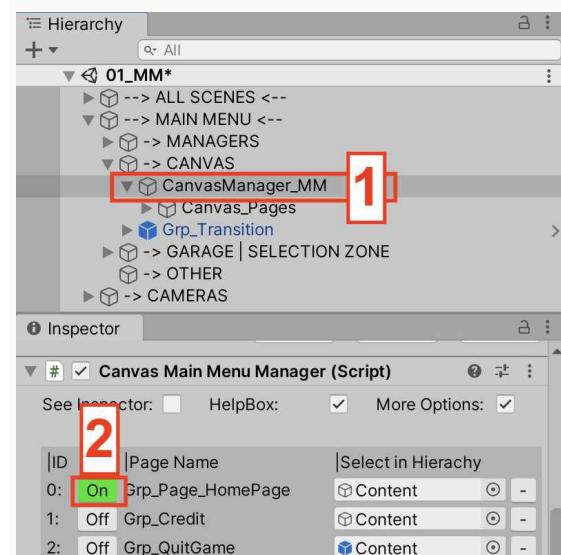
As an example we are going to create a button inside **Grp_Page_HomePage** (spot 1).

This button will open the page **Grp_Options** (spot 2).



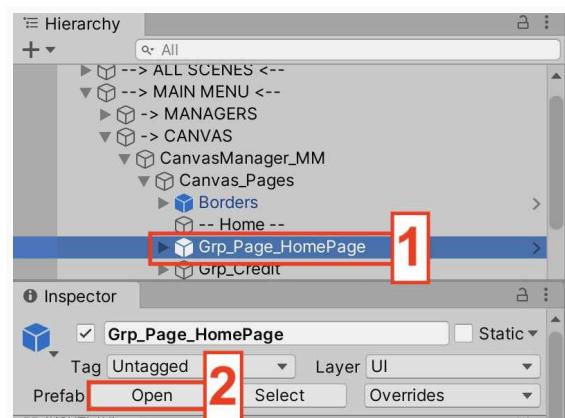
In the Hierarchy select **CanvasManager_** (spot 1).
(Hierarchy: MAIN MENU or GAMEPLAY → **CanvasManager_**)

In the Inspector press button next to **ID 0** to display the **Grp_Page_HomePage** in scene view (spot 2).



-In the Hierarchy select **Grp_Page_HomePage** (spot 1).

-In the Inspector press **Open** to open the prefab (spot 2).

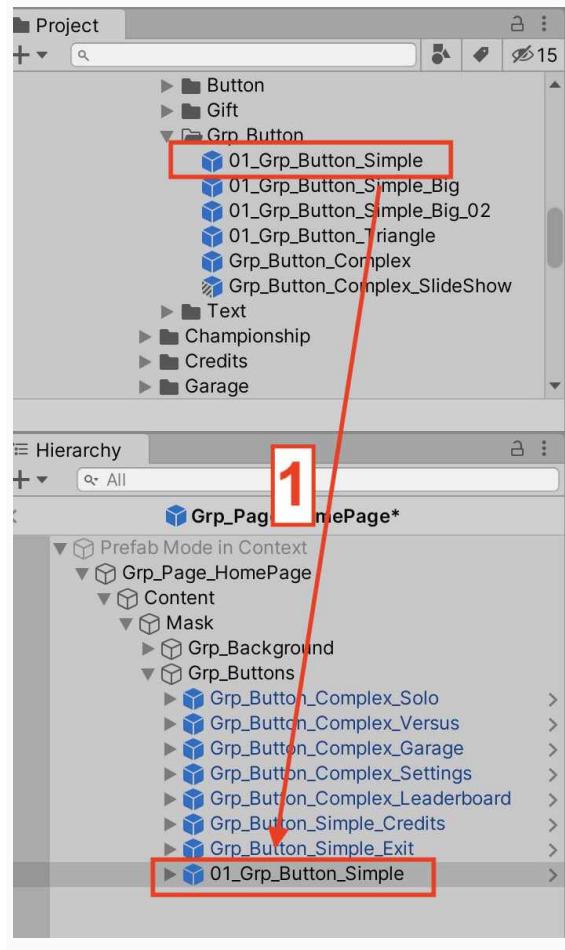


-From Project Tab drag and drop the prefab **Button_NextPage**.

(Project tab: Assets → Prefabs → TS → UI → Basics → **Grp_Button** → **01_Grp_Button_Simple**)

inside **Grp_Buttons** in the Hierarchy (spot 1).

(Hierarchy: **Grp_Page_HomePage** → **Content** → **Mask** → **Grp_Buttons** →)



The button is added to the prefab.

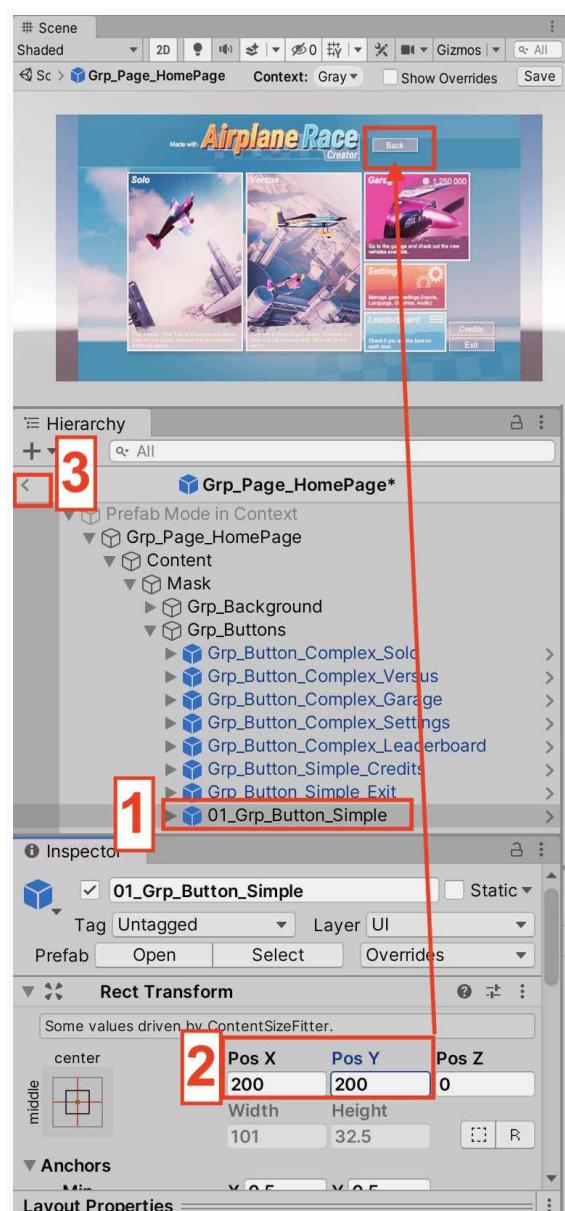
For the example:

-Select 01_Grp_Button_Simple

-Change Pos X in Rect Transform to 200.

-Change Pos Y in Rect Transform to 200.
(spot 2)

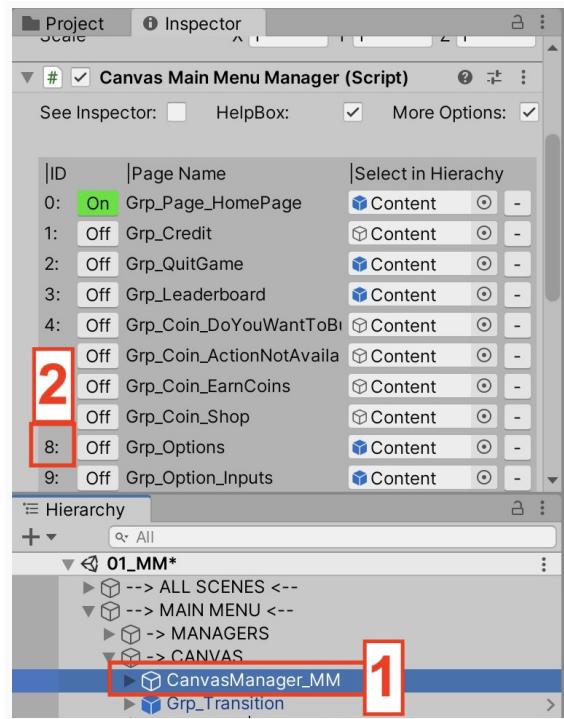
-Close the prefab by pressing < button (spot 3).



For the next step, we need to know the page ID corresponding to the page opened by our button.

In the Hierarchy select **CanvasManager_** (spot 1).
(Hierarchy: MAIN MENU or GAMEPLAY → **CanvasManager_**)

In this example we want to open the page **Grp_Options**.
The **ID** is **8** (spot 2).



Now we are going to set up the button.

-In the Hierarchy select **Button_Simple** in **Grp_Page_HomePage** (spot 1).

(Hierarchy: MAIN MENU → **CanvasManager_MM** → **Canvas_Pages** → **Grp_Page_HomePage** → **Content** → **Mask** → **Grp_Buttons** → **01_Grp_Button_Simple** → **Button_Simple**)

-In the Inspector go to script **Button Custom** (spot 2).

-In **On Click ()** section press **+** button (spot 3).

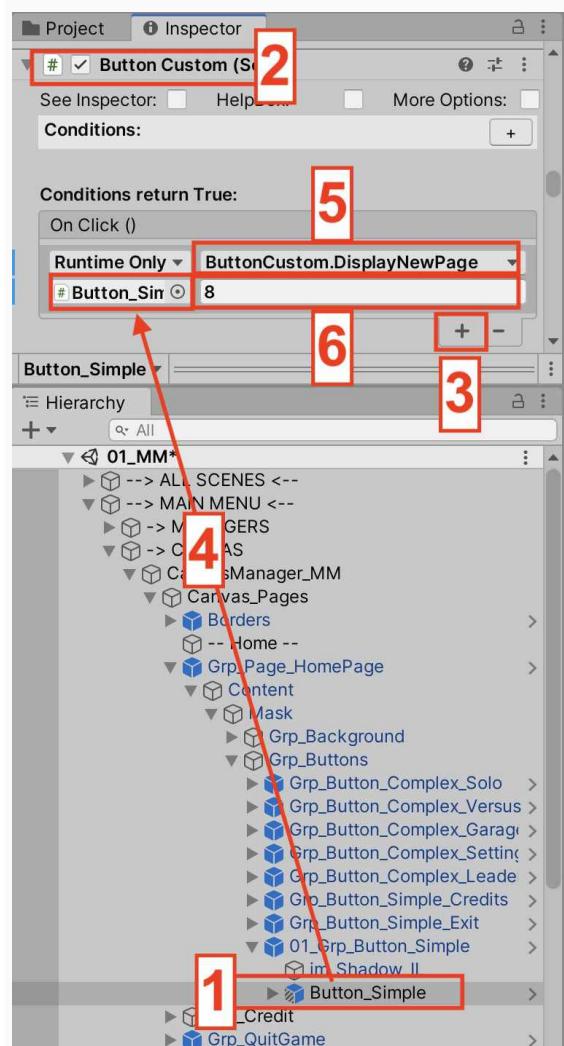
-Drag and drop **Button_Simple** object in the empty field (spot 4).

-In the dropdown menu select the **ButtonCustom** script then **DisplayNewPage** method (spot 5).

-Write the **ID** corresponding to the page that need to be opened when the button is pressed.

In our example the **ID** is **8** to open the page **Grp_Options**.

Now when the player presses the button **Button_Example** the page **Grp_Options** is opened.



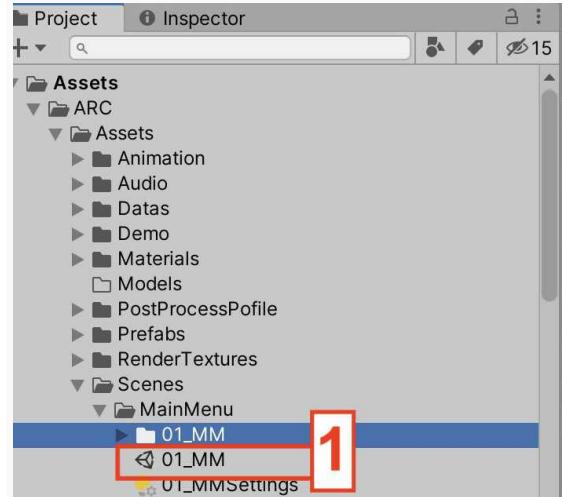
How to: Add custom page event during the page initialization

As an example we are going to add an event to the Main Menu Page.

In this example we are going to display the text **Main Page is initialized** in the console tab when the page is initialized.

-Open the Main Menu Scene (**01_MM**)
(spot 1).

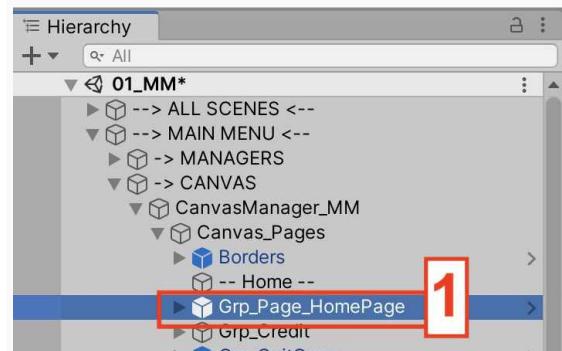
(Project tab: Assets → Scenes → MainMenu → 01_MM)



-In the Hierarchy

select **Grp_Page_HomePage** (spot 1).

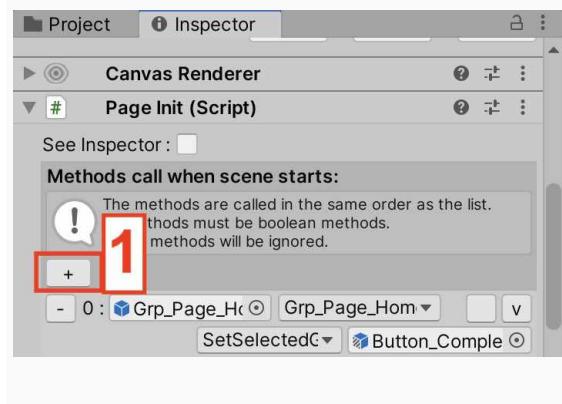
(Hierarchy: MAIN MENU → CANVAS → CanvasManager_MM → Canvas_Pages → Grp_Page_HomePage)



In this example we want to display the text **Main Page is initialized** in the console tab when the page is initialized.

The script **PageInitAssistant.cs** contained a method named **InitPageEventExample** that displays this text in the console tab. This script is attached to **Grp_Page_HomePage** object.

-In the Inspector press + button in script **PageInit** (spot 1).

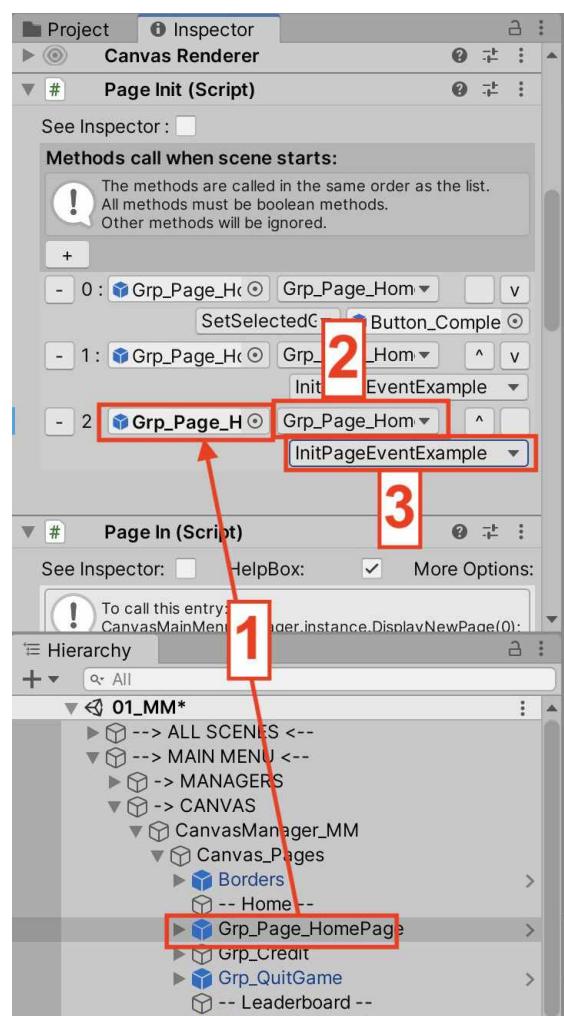


-Drag and drop the object **Grp_Page_HomePage** into the empty field (spot 1).

-Select the script that contains the method that need to be use. In our case choose script **PageInitAssistant.cs** (spot 2).

-Select the method that need to be use. In our case the boolean method named **InitPageEventExample** (spot 3).

Now when the page is initialized the text **Main Page Is Initialized** is displayed in the console tab.

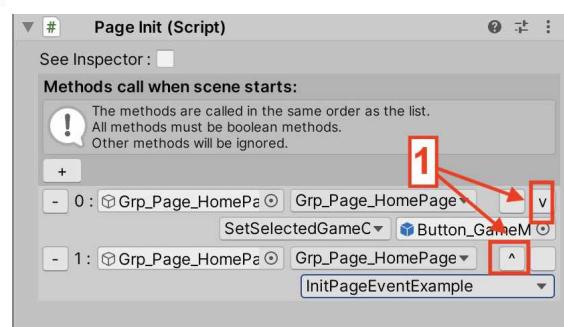


IMPORTANT:

The method used **MUST** be a boolean method.
The boolean method can use one argument as int, GameObject, float, AudioClip)

Info:

It is possible to change the order of the page events by pressing **^** and **v** button (spot 1).



Overview: Open/ Close page options

When a page is opened or closed it is possible to:

- Choose the transition to go from one page to another one.
- Choose the new page to opened.
- Call a custom methods...

The following sections explain the different options available when a page is opened or closed.

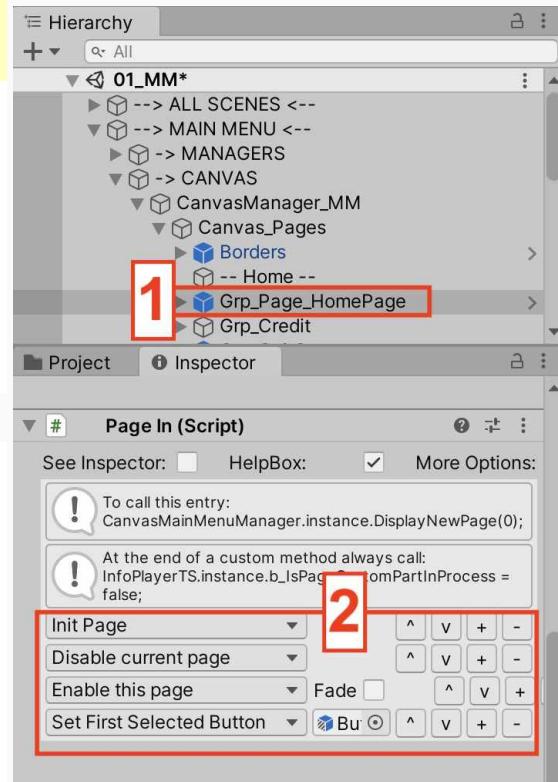
How to access and modify events called when a page is opened

-Select the page you want to modify in the hierarchy.
(For example the page [Grp_Page_HomePage](#) (spot 1)).

-Then in [Page In](#) component you can access the events called when the page is opened (spot 2).

Info:

The order of the list is important. An event in the list is called only when the previous one is finished.



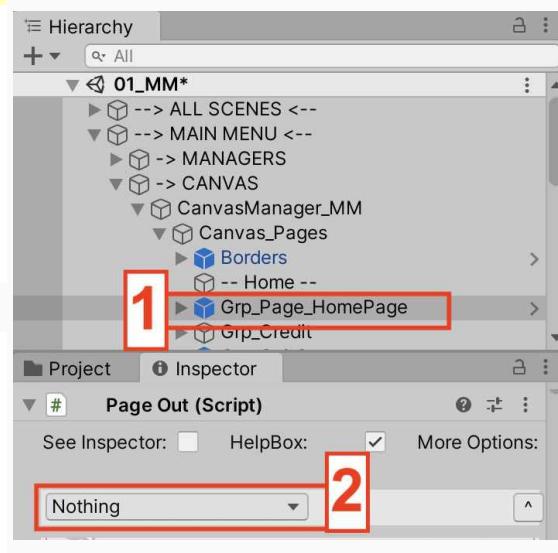
How to access and modify events called when a page is closed

-Select the page you want to modify in the hierarchy.
(For example the page [Grp_Page_HomePage](#) (spot 1)).

-Then in [Page Out](#) component you can access the events called when the page is closed (spot 2).

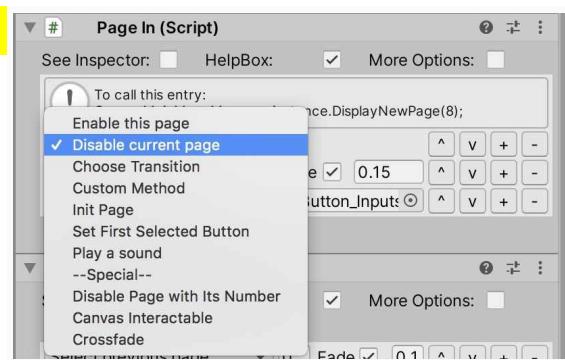
Info:

The order of the list is important. An event in the list is called only when the previous one is finished.



Open a page: Options overview

This section describes all the events that can be used when a new page is opened.



Enable this page

Enable and display the new page on scene view.

If **Fade** is checked:

The transition between the current page and the new page is done with a fade.

It is possible to choose the fade duration.



Disable current page

The current page displayed on screen is disabled in the scene view.

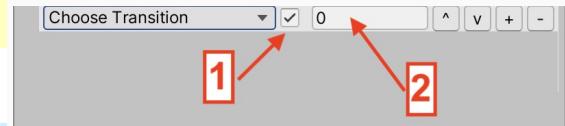
IMPORTANT: Disable current page **MUST** always be called before the event Enable this page.



Choose transition

Use a transition when a new page is displayed.

More info in section [Transition System](#)
→ Page 162



It is possible to bypass this parameter by pressing the checkbox spot 1.

The int field spot 2 corresponds to the transition ID use when the new page is displayed.

Custom Method

It allows to call a custom method when the new page opened.

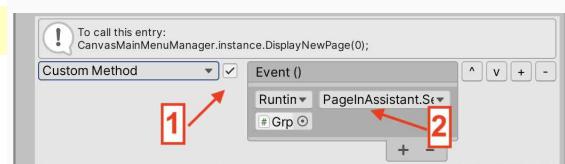
It is possible to bypass this parameter by pressing the checkbox spot 1.

Drag and drop your method into the field spot 2.

IMPORTANT:

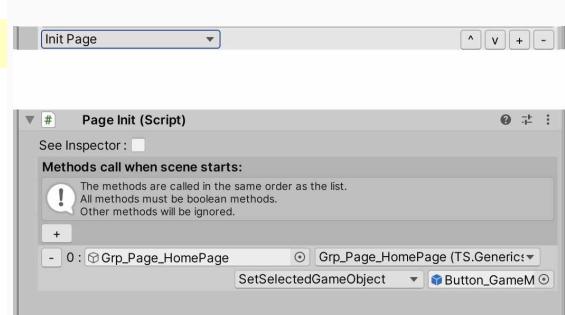
To know how to create method that can be used as a page event when the page is opened.

More info → [Link](#)



Init Page

It allows to call the list of methods contained in the script **PageInit.cs**.



Set First Selected Button

Use to choose the first button selected in the new page.

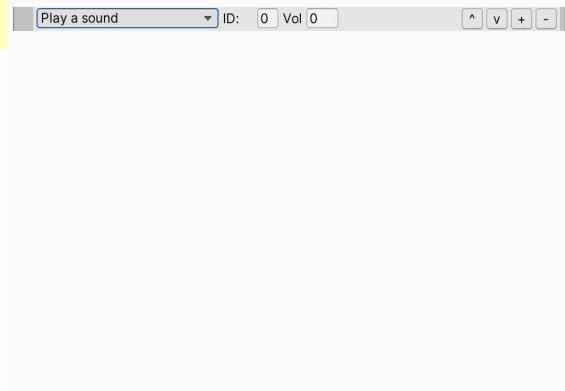


Play a sound

Play a sound when the new page is opened.

Choose its **ID** and volume.

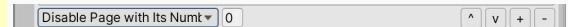
The sound must be a Sfx contained in SfxList script
(Hierarchy: ALL SCENES → MANAGERS → SoundManager → AudioClipManager → Sfx)



Disable Page with its Number

Disable a page using its **ID** number.

(The ID number can be find in object *CanvasManager_*
(Hierarchy: MAIN MENU or GAMEPLAY → CANVAS →
CanvasManager_))



Canvas Interactable

Each page has a **canvas group** component attached to it.

It is possible to choose if the canvas group of a specific page is interactable or not.

ID allows to choose the page

State checkbox allows to choose if the canvas group for the specific page is enable or disable.



Crossfade

Allows to use crossfade between the page that need to be closed and the page that need to be opened.

From Page: The ID of he page to open.

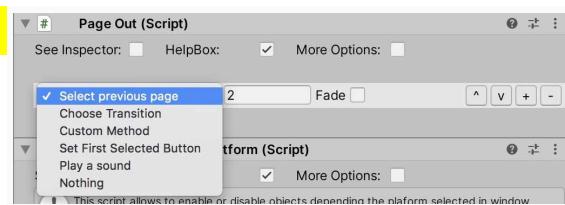
To: The ID of he page to close.

Fade: The speed of the fade.



Close a page: Options overview

This section describes all the events that can be used when a new page is closed.



Select previous page

Display a new page when the current page is closed.

Select the new page using its ID (spot 1)

(The ID number can be find in object CanvasManager_ (Hierarchy: MAIN MENU or GAMEPLAY → CANVAS → CanvasManager_))

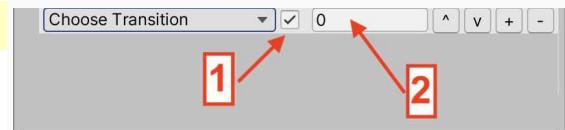
Fade checkbox allows to use a fade for the transition (spot 2).

With the float field spot 3 choose the fade speed.



Select previous page

Use a transition when a new page is closed.



More info in section [Transition System](#)
→ Page 162

It is possible to bypass this parameter by pressing the checkbox spot 1.

The int field spot 2 corresponds to the transition ID use when the new page is displayed.

Custom Method

It allows to call a custom method when the page is closed.

It is possible to bypass this parameter by pressing the checkbox spot 1.

Drag and drop your method into the field spot 2.

IMPORTANT:

To know how to create method that can be used as a page event when the page is closed.

More info → [Link](#)



Set First Selected Button

Use to choose the first button selected in the new page.

Set First Selected Button ▾ Content ○ ^ v + -

Play a sound

Play a sound when the new page is opened.

Play a sound ▾ ID: 0 Vol 0 ▾ v + -

Choose its **ID** and volume.

*The sound must be a Sfx contained in SfxList script
(Hierarchy: ALL SCENES → MANAGERS → SoundManager →
AudioClipManager → Sfx)*

Nothing

Used in case the player cannot return to another menu by pressing the back key.

Nothing ▾ ▾ v + -

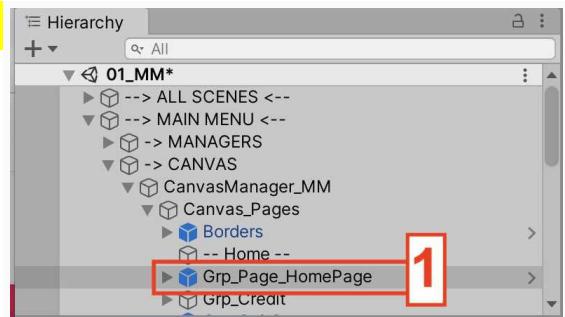
How to: Add an event when a page is opened

As an example we are going to add an event to the Main Menu Page. The event will play a sound fx.

-In the Hierarchy

select **Grp_Page_HomePage** (spot 1)

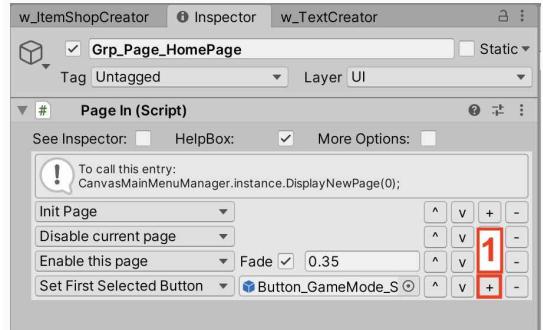
(Hierarchy: MAIN MENU → CANVAS → CanvasManager_ → Canvas_Pages→ Grp_Page_HomePage)



In this example we want to play a Sfx when the Main Menu Page is displayed.

-In the Inspector go to script **PageIn**.

-Press the last + button (spot 1).



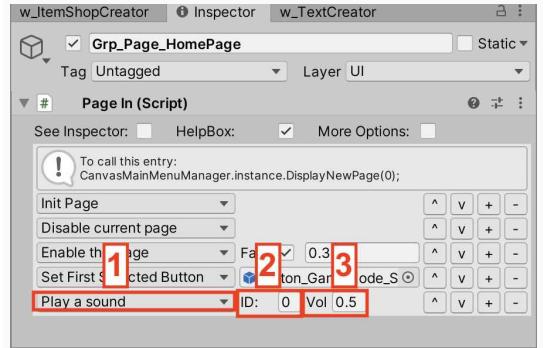
-In the last dropdown list select **Play a sound** (spot 1).

-Select the Sfx ID (spot 2).

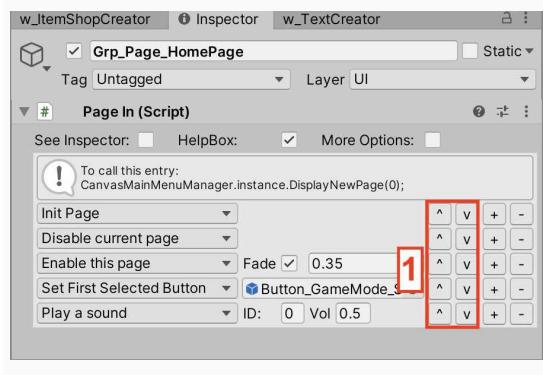
In the example set the **ID** to 0.

-Choose the volume of the Sfx (spot 3).

In the example set the volume to **0.5**.



It is possible to change the order of the page events by pressing ^ and v button (spot 1).



How to: Create method that can be used as an event when the page is opened

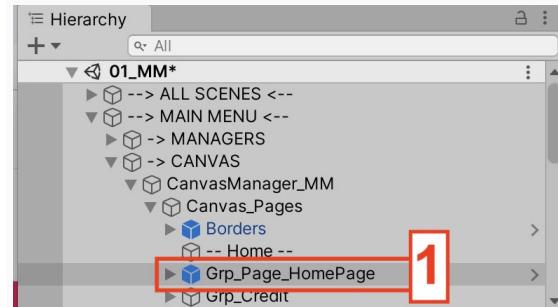
As an example we are going to create a custom method that can be used as an event when the Main Menu Page is opened.

The event will play a new music.

-In the Hierarchy

select **Grp_Page_HomePage** (spot 1).

(Hierarchy: MAIN MENU → CANVAS → CanvasManager_ → Canvas_Pages→ Grp_Page_HomePage)

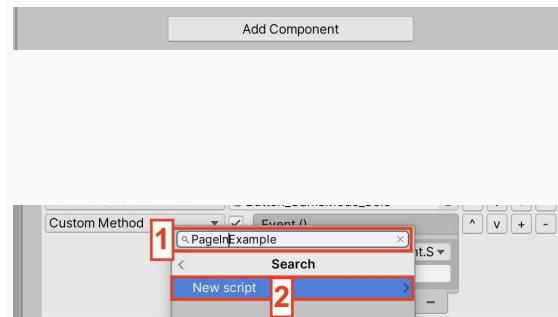


-In the Inspector create a new script by pressing **Add Component**.

A new box appears.

-Write **PageInExample** in the search field (spot 1).

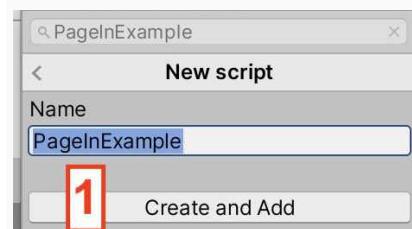
-Click on **New script** (spot 2).



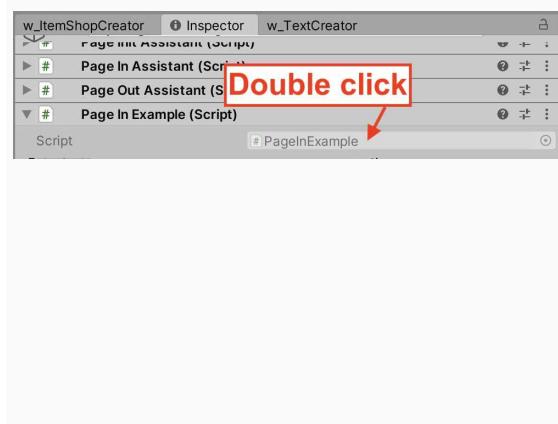
A new box appears.

-Press button **Create and Add** (spot 1).

The new script **PageInExample** is now attached to the main menu page.



-In the Inspector double click on **PageInExample** script to open it on your script editor.



In the script editor add line 4 (spot 1):

using TS.Generics;

This line allows to access some classes included in the assets

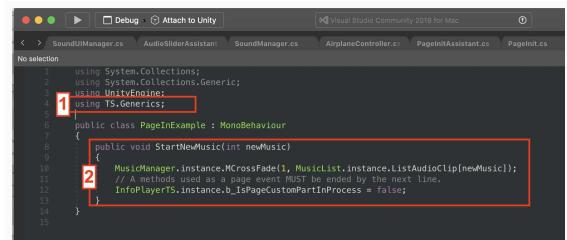
-Remove Start() and Update() methods.

-Add the following lines to the code (spot 2)

```
public void StartNewMusic(int newMusic)
{
    MusicManager.instance.MCrossFade(1,
        MusicList.instance.ListAudioClip[newMusic]);
    // A methods used as a page event MUST be ended by the next line.
    InfoPlayerTS.instance.b_IsPageCustomPartInProcess = false;
}
```

The first line plays a new music using the Music system included in the asset.

The second line tells the PageIn script that the event is finished.



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using Unity;
4 using TS.Generics;
5
6 public class PageInExample : MonoBehaviour
7 {
8     public void StartNewMusic(int newMusic)
9     {
10         MusicManager.instance.MCrossFade(1, MusicList.instance.ListAudioClip[newMusic]);
11         // A methods used as a page event MUST be ended by the next line.
12         InfoPlayerTS.instance.b_IsPageCustomPartInProcess = false;
13     }
14 }
```

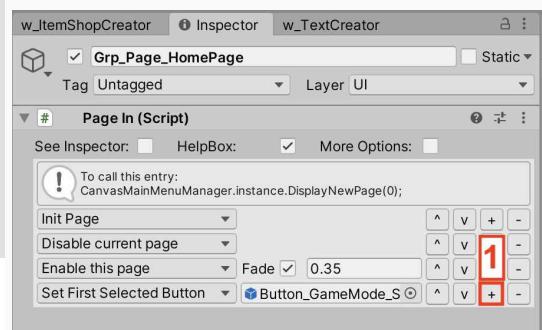
VERY IMPORTANT:

Your method must ended by this line of code:

InfoPlayerTS.instance.b_IsPageCustomPartInProcess = false;

This line is needed to know that the event is finished.

If this line is missing, the scene will stuck.



Now let's add the method to the page event list.

-In the Inspector go to script **PageIn**.

-Press the last + button (spot 1).

-In the last dropdown list select **Custom Method** (spot 1).

-Check the box spot 2 to enable the custom method (spot 2).

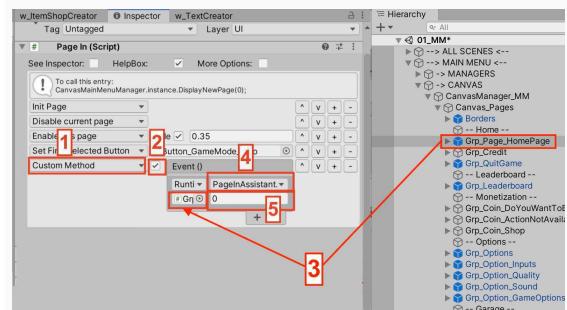
-Drag and drop **Grp_Page_HomePage** to the empty field (spot 3).

-Select the script **StartNewMusic** (spot 4).

PageInExample → StartNewMusic

-Select the music ID. In our case ID = 0.

Now when the Main Menu page is opened the music with ID = 0 is played.



Info:

It is possible to change the order of the page events by pressing ^ and v button (spot 1).

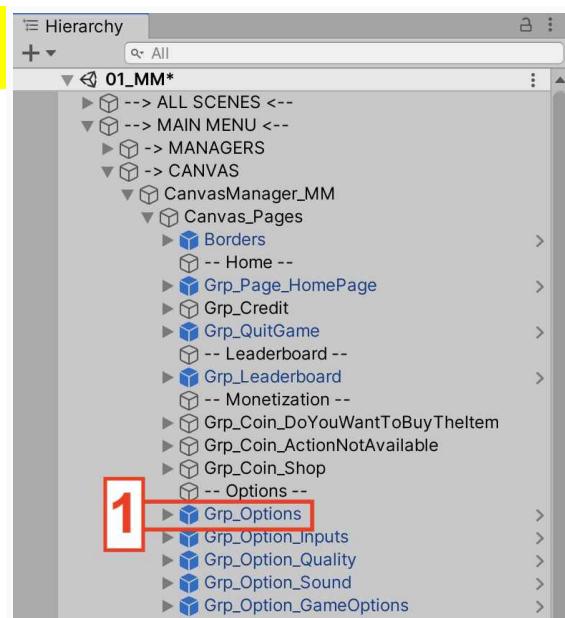
How to: Add an event when a page is closed

As an example we are going to add an event to the Options Menu Page.

-In the Hierarchy

select **Grp_Options** (spot 1).

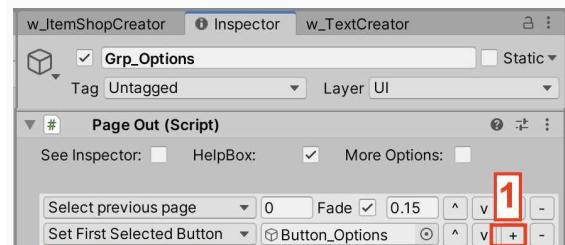
(Hierarchy: MAIN MENU → CANVAS→ CanvasManager_ → Canvas_Pages→ Grp_Options)



For the example we want to play a Sfx when the Options Menu Page is closed.

-In the Inspector go to script **PageOut**.

-Press the last + button (spot 1).



-In the last dropdown list select **Play a sound** (spot 1).

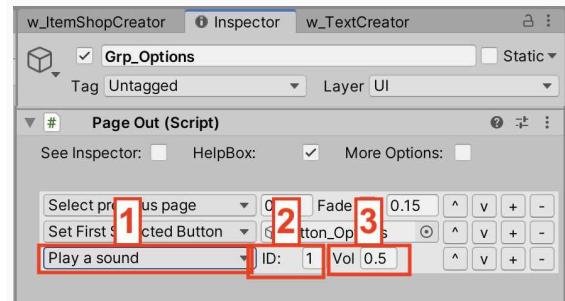
-Select the Sfx **ID**.

In the example set the **ID** to **1** (spot 2).

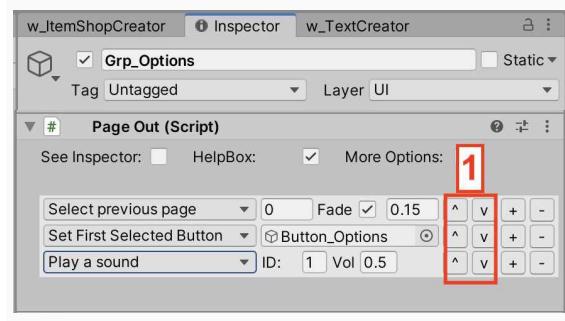
-Choose the volume of the Sfx.

In the example set the **volume** to **0.5** (spot 3).

Now when the Options Menu page is closed the Sfx with ID = 1 is played.



It is possible to change the order of the page events by pressing ^ and v button (spot 1).



How to: Create method that can be used as an event when the page is closed

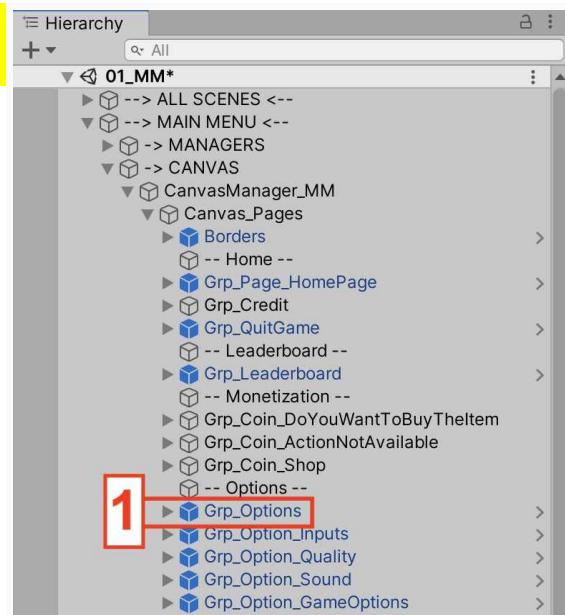
As an example we are going to create a custom method that can be used as an event when the Options Menu Page is closed.

The event will play a new music.

-In the Hierarchy

select **Grp_Options** (spot 1) .

(Hierarchy: MAIN MENU → CANVAS → CanvasManager_ → Canvas_Pages→ Grp_Options)

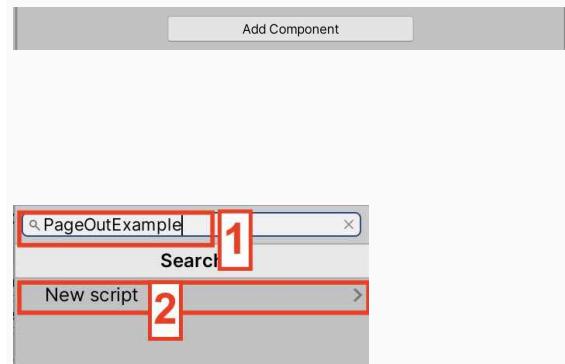


-In the Inspector create a new script by pressing **Add Component**.

A new box appears.

-Write **PageOutExample** in the search field (spot 1).

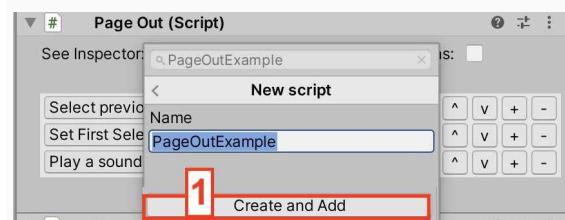
-Click on **New script** (spot 2).



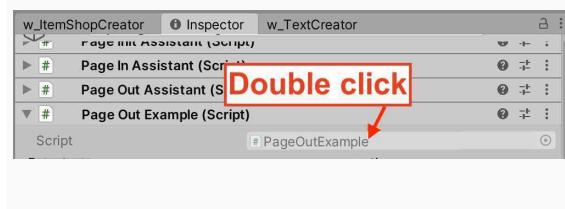
A new box appears.

-Press button **Create and Add** (spot 1).

The new script **PageOutExample** is now attached to the main menu page.



-In the Inspector double click on **PageOutExample** script to open it on your script editor.



In the script editor add line 4 (spot 1):

`using TS.Generics;`

This line allows to access some classes included in the assets.

-Remove `Start()` and `Update()` methods.

-Add the following lines to the code (spot 2)

```
public void StartNewMusic(int newMusic)
{MusicManager.instance.MCrossFade(1,
MusicList.instance.ListAudioClip[newMusic]);
// A methods used as a page event MUST be ended by the next
line.
InfoPlayerTS.instance.b_IsPageCustomPartInProcess = false;}
```

The first line plays a new music..

The second line tells the `PageOut` script that the event is finished.

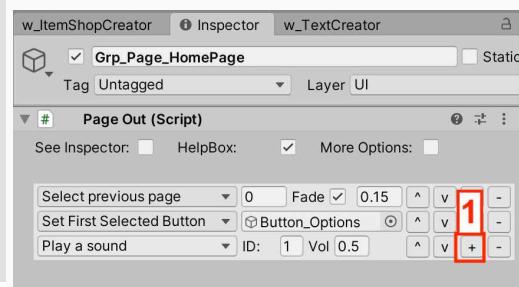
```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityVoxine;
4 1 using TS.Generics;
5 public class PageOutExample : MonoBehaviour
6 {
7     public void StartNewMusic(int newMusic)
8     {
9         MusicManager.instance.MCrossFade(1, MusicList.instance.ListAudioClip[newMusic]);
10        // A methods used as a page event MUST be ended by the next line.
11        InfoPlayerTS.instance.b_IsPageCustomPartInProcess = false;
12    }
13 }
```

VERY IMPORTANT: Your method must ended by this line of code:

`InfoPlayerTS.instance.b_IsPageCustomPartInProcess = false;`

This line is needed to know that the page event is finished.

If this line is missing, the scene will stuck.



Now let's add the method to the page event list.

-In the Inspector go to script `PageOut`.

-Press the last + button (spot 1)

-In the last dropdown list select **Custom Method** (spot 1).

-Check the box spot 2 to enable the custom method (spot 2).

-Drag and drop `Grp_Options` to the empty field (spot 3).

-Select the script `StartNewMusic` (spot 4).

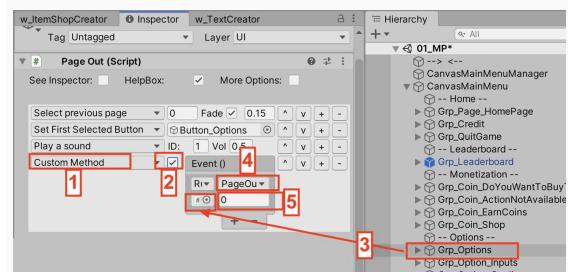
`PageOutExample` → `StartNewMusic`

-Select the music ID. In our case `ID = 0`.

Now when the Options Menu page is closed the music with `ID = 0` is played.

Info:

It is possible to change the order of the page events by pressing ^ and v button (spot 1).



Leaderboard

This section describes how to customize the process to save and load score in the leaderboard.
Leaderboard are used to save Time Trial player Name + Time.

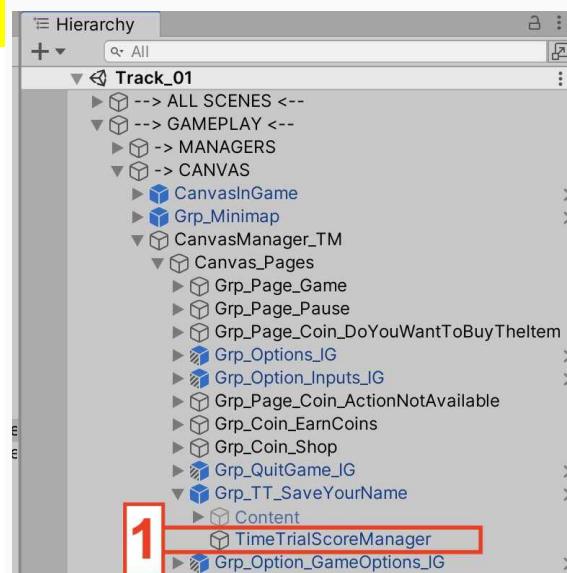
Table of contents:

Overview	Link
How to: Choose the number of entries saved in the leaderboard	Link
Scripting: Access Player name and time after Time Trial race	Link
How to: Customize leaderboard save process	Link
How to: Load custom data in the leaderboard	Link

Overview

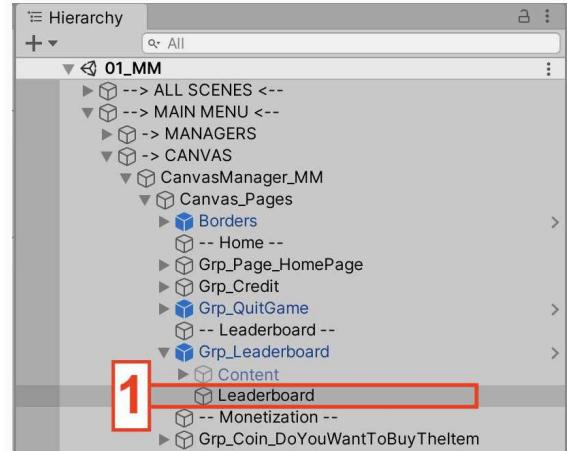
Leaderboard is separated into 2 parts:

-In gameplay scenes, **TimeTrialScoreManager** object saves the player Name + time after the race (spot 1).
(Hierarchy: GAMEPLAY → CANVAS → CanvasManager_TT → CanvasPages → Grp_TT_SaveYourName → TimeTrialScoreManager)



-In the Main Menu scene, **Leaderboard** object is used to display the leaderboard for each track (Time Trial Mode) (spot 1).

(Hierarchy: MAIN MENU → CANVAS → CanvasManager_MM → CanvasPages → Grp_Leaderboard → Leaderboard)



How to: Choose the number of entries saved in the leaderboard

It is possible to choose how many Name + time are saved by track.

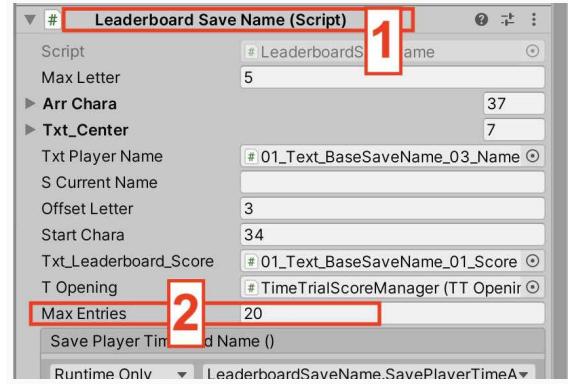
- In gameplay scenes, **TimeTrialScoreManager** object saves the player Name + time after the race (spot 1).

(Hierarchy: GAMEPLAY → CANVAS → CanvasMainMenuManager → CanvasMainMenu → Grp_TT_SaveYourName → TimeTrialScoreManager)



- In the Inspector go to **LeaderboardSaveName** script (spot 1).

- Change **Max Entries** (spot 2).



Scripting: Access Player name and time after Time Trial race

In Time Trial, after a race, you can access Player name and time.

Access Player Name:

```
LeaderboardSaveName.instance.sCurrentName
```

Access Player Time:

Raw time:

```
LapCounterAndPosition.instance.posList[0].globalTime
```

Time as an int:

```
LeaderboardSaveName.instance.timeScore
```

```
timeScore = (int)(LapCounterAndPosition.instance.posList[0].globalTime * 1000.0f);
```

Time formatted (minute, seconds, milliseconds)

```
LeaderboardSaveName.instance.FormatTimer(LeaderboardSaveName.instance.timeScore)
```

How to: Customize leaderboard save process

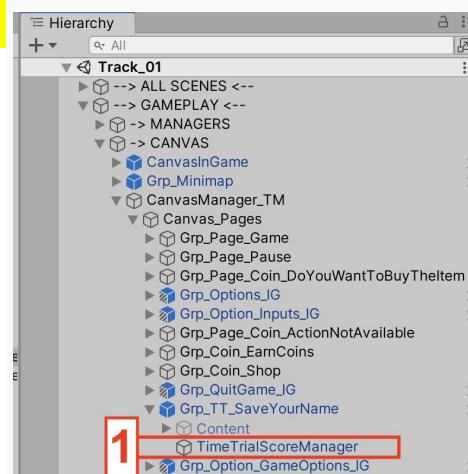
If you don't want to use the local leaderboard system to save player name and time, you can use a custom method to save the data.

It is useful if you want to save the data in an online leaderboard.

-In gameplay scenes, **TimeTrialScoreManager** object saves the player Name + time after the race (spot 1).

(Hierarchy: GAMEPLAY → CANVAS →

CanvasMainMenuManager → CanvasMainMenu → Grp_TT_SaveYourName → TimeTrialScoreManager



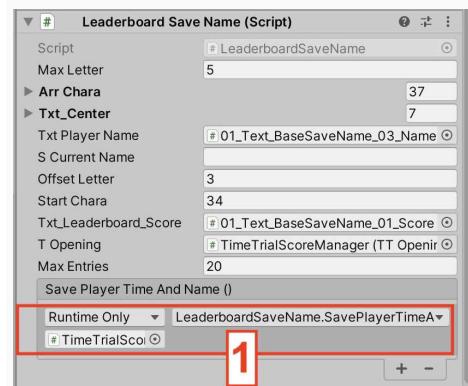
-In the Inspector go to **LeaderboardSaveName** script (spot 1).

-Remove the method (spot 1)

-Replace this method with your custom method.

You will find an example of method in **LeaderboardSaveCustom** script.

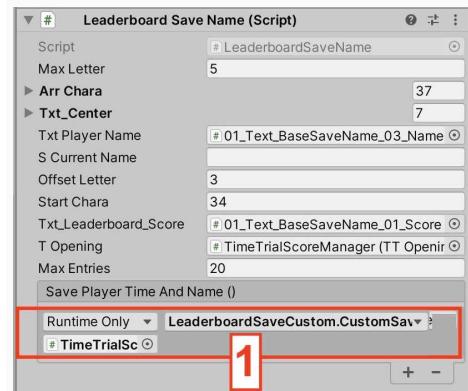
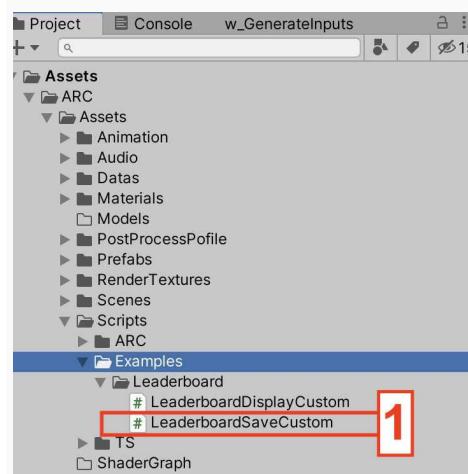
(Project Tab: Assets → Scripts → Examples → LeaderboardSaveCustom)



-Attached **LeaderboardSaveCustom** script to **TimeTrialScoreManager** object.

-In section **Save PlayerTime And Name ()** select the **LeaderboardSaveCustom** script and the method **CustomSave** (spot 1)

You can use this method as a base to your custom save method.



How to: Load custom data in the leaderboard

By default, the leaderboard load the Time Tiral data saved in local.

It is possible to load custom data.

This section gives you an example to understand how to load custom data.

You will find an example of method in **LeaderboardDisplayCustom** script.

(Project Tab: Assets → Scripts → Examples → Leaderboard → LeaderboardDisplayCustom)

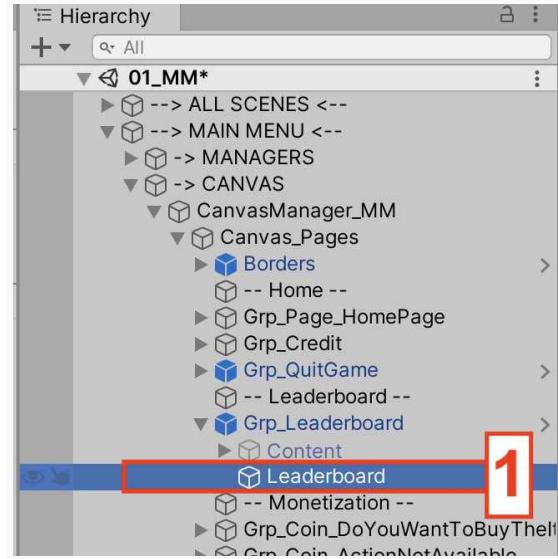
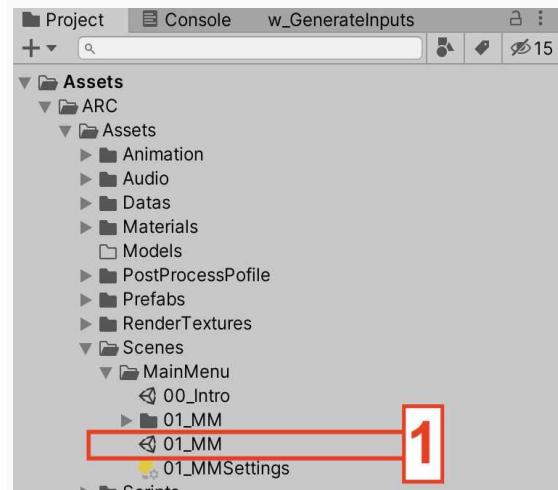
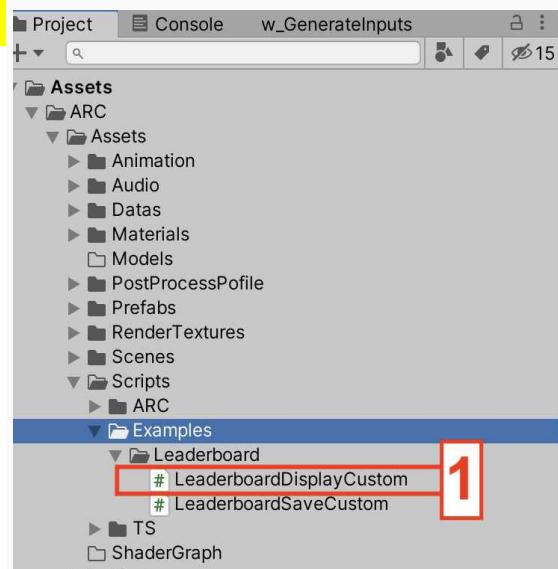
Leaderboard are displayed only in the Main Menu.

-Open the **01_MM** scene (spot 1)

(Project tab: Assets → Scenes → Min Menu → 01_MM)

-In the Hierarchy select **Leaderboard** object (spot 1).

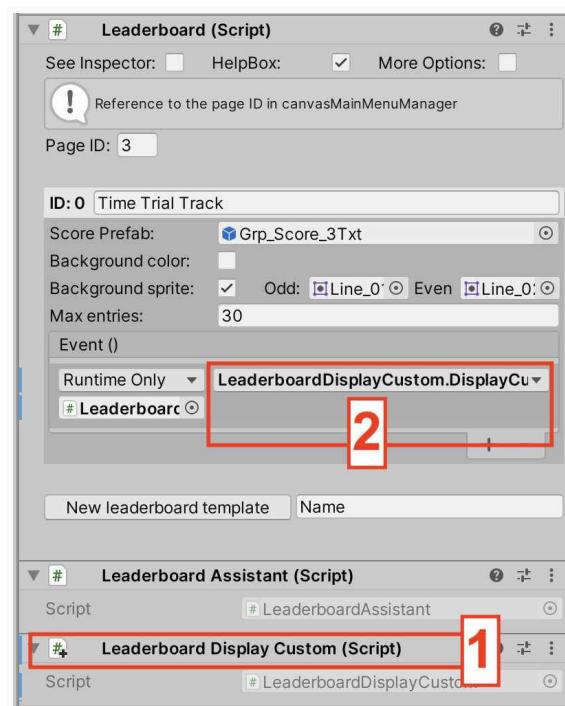
(Hierarchy: MAIN MENU → CANVAS → CanvasManager_MM → Canvas_Pages → Leaderboard)



-In the Inspector add **LeaderboardDisplayCustom** script (spot 1).

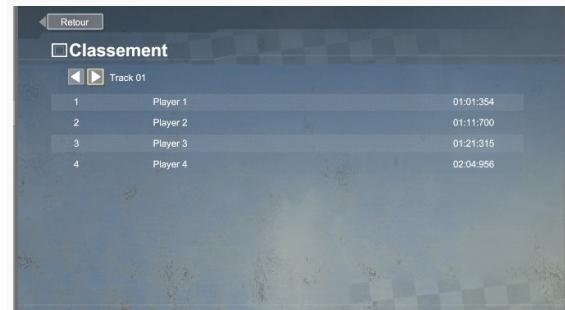
-Select in the dropdown menu (spot 2) the method **DisplayCustomLeaderboard**.

(*LeaderboardDisplayCustom → DisplayCustomLeaderboard*)



Now when the leaderboard is opened, it displays:

1	Player 1	01:01:354
2	Player 2	01:11:700
3	Player 3	01:21:315
4	Player 4	02:04:956



-Open **LeaderboardDisplayCustom** script in your script editor.

(Project Tab: Assets → Scripts → Examples → Leaderboard → *LeaderboardDisplayCustom*)

The first part of the method gives access to the current selected Time Trial race.

```

29     public void DisplayCustomLeaderboard()
30     {
31         //=> Choose the track to display
32         int selectedTrack = Leaderboard.instance.currentSelectedLead;
33
34         //=> Find the parameters of the track
35         TrackData.trackParams trackParams = DataRef.instance.tracks.listTrackParams[selectedTrack];
36         if (!DataRef.instance.tracks[selectedTrack].isMultiData)
37             trackParams = DataRef.instance.tracks[selectedTrack].listTrackParams[0];
38
39         //=> Find the name of the leaderboard
40         string dataFolder = trackParams.selectedListNameMultilanguage;
41         int textID = trackParams.nameIDMultilanguage;
42         string leaderboardName = LanguageManager.instance.String_ReturnText(dataFolder, textID);
43
44         //=> Update selected track if bDisplayTrackUsingTrackListOrder = false
45         if (!DataRef.instance.timeTrialModeData.bDisplayTrackUsingTrackListOrder)
46             selectedTrack = DataRef.instance.timeTrialModeData.customTrackList[selectedTrack];
47
48     }

```

Replace this list with a list of Name and Time you want to display.

Create a list with:

Name 1, Time 1, Name 2, Time 2,...,...

Last part calls the method that display the leaderboard.

```

39     //=> Create the list of best score to displayed.
40     //=> As an example you can load an online leaderboard and save to value in a List.
41     List<string> scoreList = new List<string>()
42     {
43         "Player 1", "01:01:354",
44         "Player 2", "01:11:700",
45         "Player 3", "01:21:315",
46         "Player 4", "02:04:956",
47     };

```

```

48     //=> Display the leaderboard
49     Leaderboard.instance.NewLeaderboard(0, scoreList.ToArray(), leaderboardName, false);

```

Rendering

This section describes how to setup rendering

Table of contents:

Lightmaps settings	Link
Reflection Probs	Link
Global Post Effects	Link
Local Post Effects	Link
Night	Link
Glow	Link

Lightmaps settings

This section describes the rendering presets used for the demo game included in the asset

If you want to use the demo track lighting settings:

- Open lighting window
- From project tab drag and drop **New Lighting Settings** to the **lighting settings** slot (spot1)

(Project tab: Asset → Light Settings → New Lighting Settings)

Lighting mode use is: **Baked Indirect**

In this mode, you can change the time of day without needing to recalculate the lightmaps:

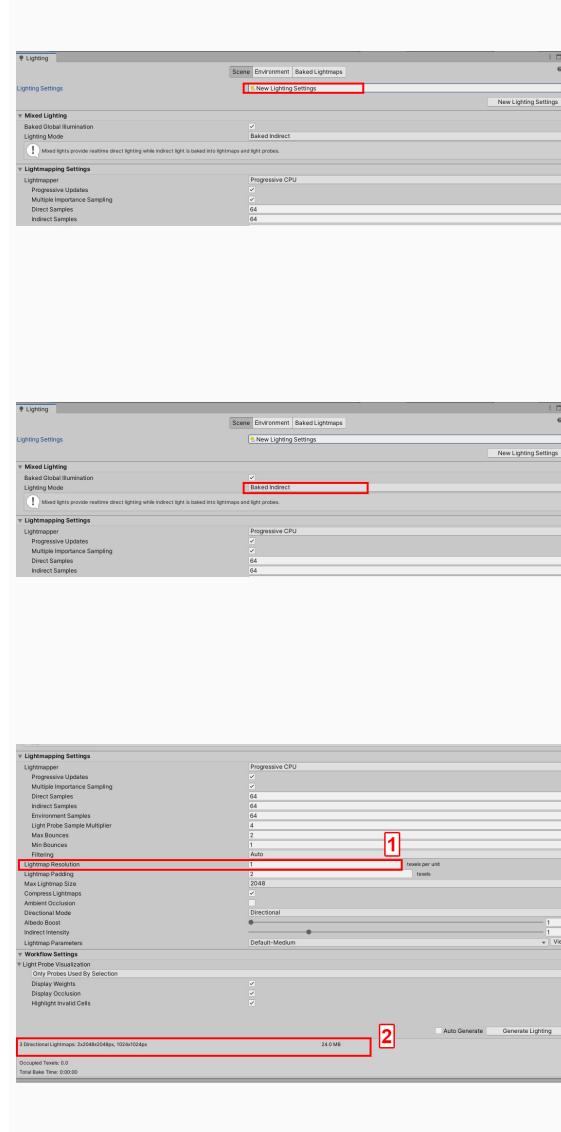
- First calculate the lightmaps
- Rotate the directional light to change the time of the day

Demo scene is very large then a very low **Lightmap Resolution** is used (spot1)

Increase the **Lightmap resolution** value increase the quality of the lightmaps but:

- Lightmaps computation is longer
- Lightmaps size are bigger

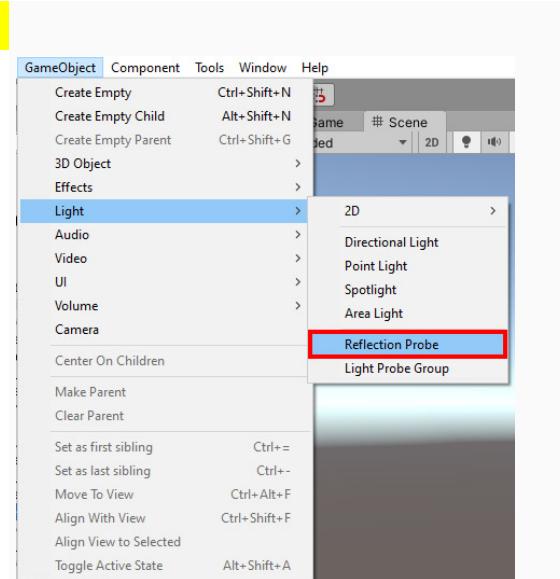
The size of the lightmaps are visible at the bottom of the lighting tab (spot 2)



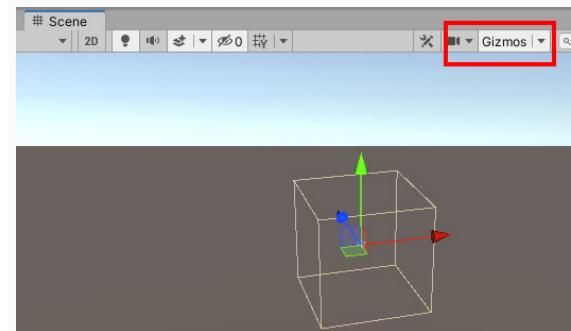
Reflection Probs

This section describes how to add reflection prob for the entire scene

-Go to GameObject > Light > Reflection Probe



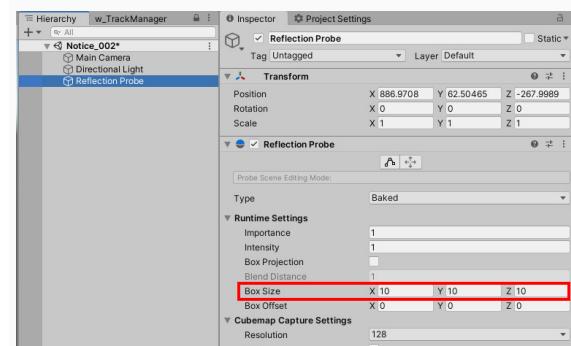
-Click on **Gizmo bar** to toggle all gizmos visibility in the scene view



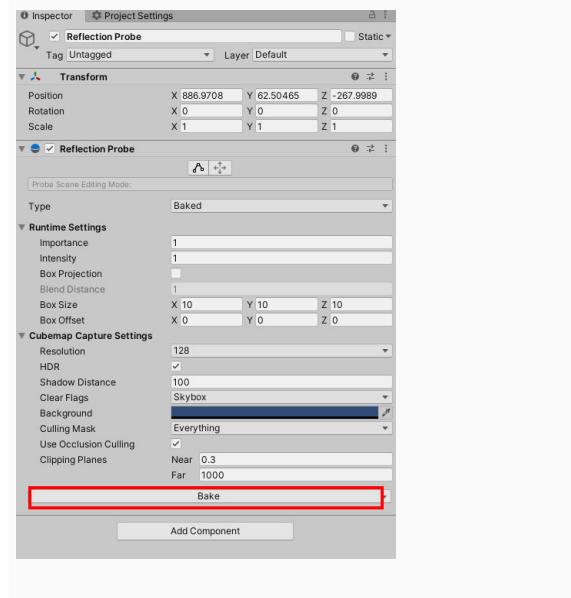
Important: The reflection Prob must cover the whole scene

To modify the size of the Reflection Prob:

- In hierarchy tab select the Reflection Prob
- Modify **box size** values to cover the whole scene



-Press **Bake** Button to calculate the Reflection Prob



Global Post Effects

This section describes how to add global postFx for player 1 and player 2

For improved image quality Post Effects are used.

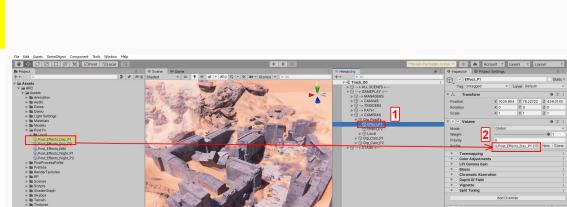
-In hierarchy tab select Effects P1 (spot1)

(Hierarchy : GAMEPLAY → CAMERAS → Grp_PostFX→ Effects_P1)

-From project tab drag and drop

Post_Effects_Day_P1 postFx volume profile in Profile slot (spot2)

(Project tab: Asset → Post Fx → Post_Effects_Day_P1)



There is a split screen versus mode so you have to put postFx profile for player 2

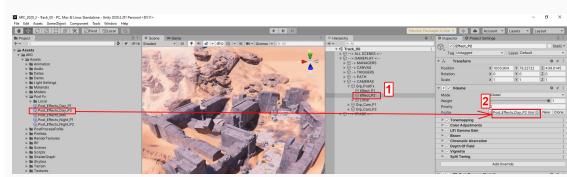
-In hierarchy tab select Effects P2 (spot1)

(Hierarchy : GAMEPLAY → CAMERAS → Grp_PostFX→ Effects_P2)

-From project tab drag and drop

Post_Effects_Day_P2 postFx volume profile in Profile slot (spot2)

(Project tab: Asset → Post Fx → Post_Effects_Day_P2)



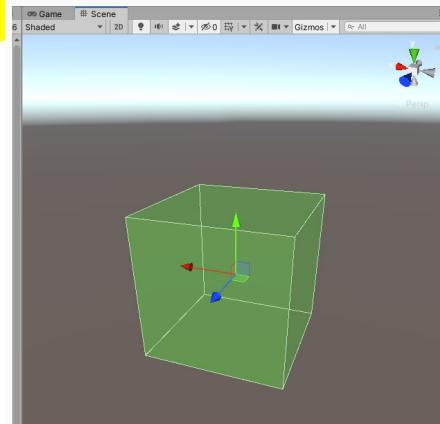
Local Post Effects

This section describes how to add local postFx

It is possible to use a post fx only in a defined area.

For this we use a local volume.

When the player collides with the volume collider, the local postFx are triggered



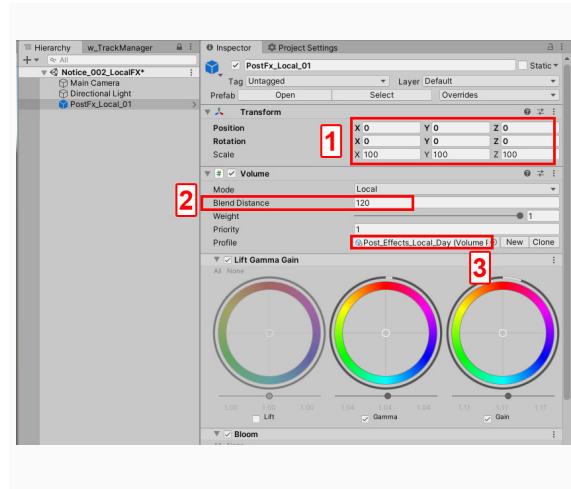
From Project tab drag and drop PostFx_Local_01 in hierarchy tab

(Project tab: Asset → Prefabs → AS → Env → Prefabs → Local_PostFX → PostFx_Local_01)

-Modify transform parameters to adapt local postFx to your needs (spot1)

-Set **Blend Distance** value (spot2)

-Add your own postFx volume profile if needed (spot3)



Night

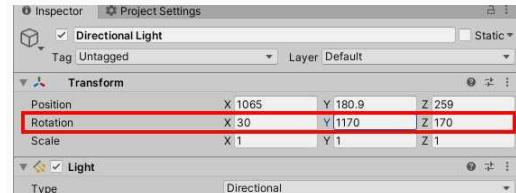
This section describes how to simulate a night scene using an "American night".

To achieve American night we need to:

- Position the directional light to have sunset lighting
- Add a postFX volume profile to simulate night
- Add a night Skybox
- Modify Fog color
- Add stars in the sky

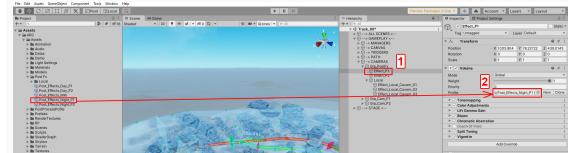
In hierarchy tab select **Directional light**

- Set Rotate x to 30
- Set Rotate y to 1170
- Set Rotate z to 170



-In hierarchy tab select **Effects P1** (spot1)

(Hierarchy : GAMEPLAY → CAMERAS → Grp_PostFX→ Effects_P1)



-From project tab drag and drop

Post_Effects_Night_P1 volume profile in **Profile Slot** (spot2)

(Project tab: Asset → Post Fx → Post_Effects_Night_P1)

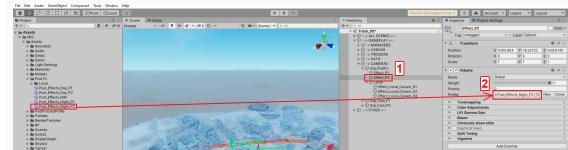
-In hierarchy tab select **Effects P2** (spot1)

(Hierarchy : GAMEPLAY → CAMERAS → Grp_PostFX→ Effects_P2)

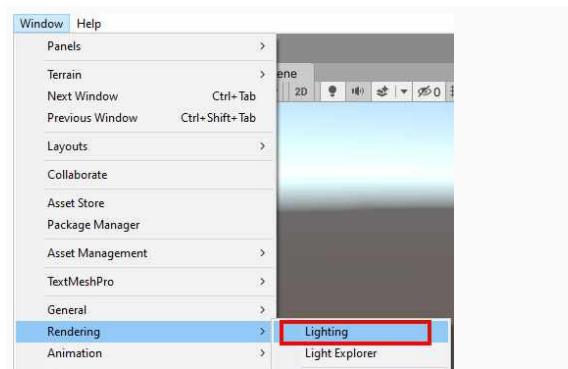
-From project tab drag and drop

Post_Effects_Night_P2 volume profile in **Profile Slot** (spot2)

(Project tab: Asset → Post Fx → Post_Effects_Night_P2)

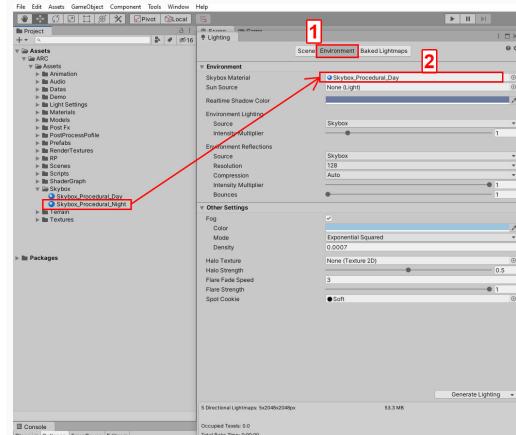


Go to Window > Rendering > Lighting



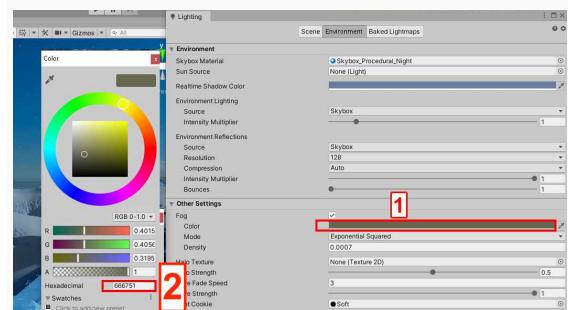
-Select Environment Tab (spot1)

-From project tab drag and drop Skybox_Procedural_Night in Skybox Material slot (spot2)
(Project tab: Asset → Skybox → Skybox_Procedural_Night)



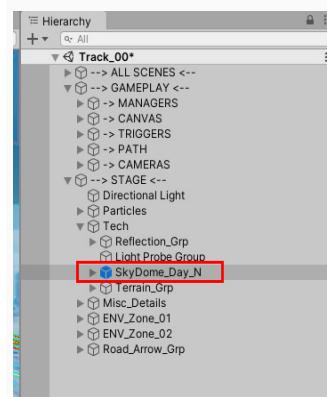
-Click on color slot (spot1)

-Set Hexadecimal value to 666751 (spot2)



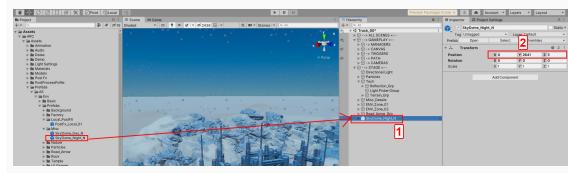
-In hierarchy tab delete SkyDome_Day_N (spot1)

(Hierarchy : STAGE→ Tech → SkyDome_Day_N)



From Project tab drag and drop SkyDome_Night_N in hierarchy tab (spot 1)

(Project tab: Asset → Prefabs → AS → Env → Prefabs → Misc → SkyDome_Night_N)



Set position to : X:0 Y:2641 Z:5 (spot 2)

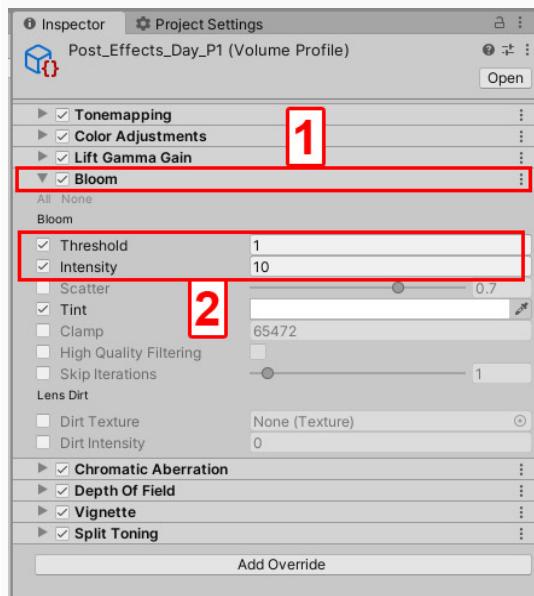
Glow

This section describes how to adjust glow intensity

-In Project tab select a Volume Profile.
(Project tab: Asset → Post Fx)

-In Inspector tab: open Bloom effect tab

-Modify Threshold and/or intensity value



Some materials use glow effects.

It is possible to modify the glow intensity of this materials.

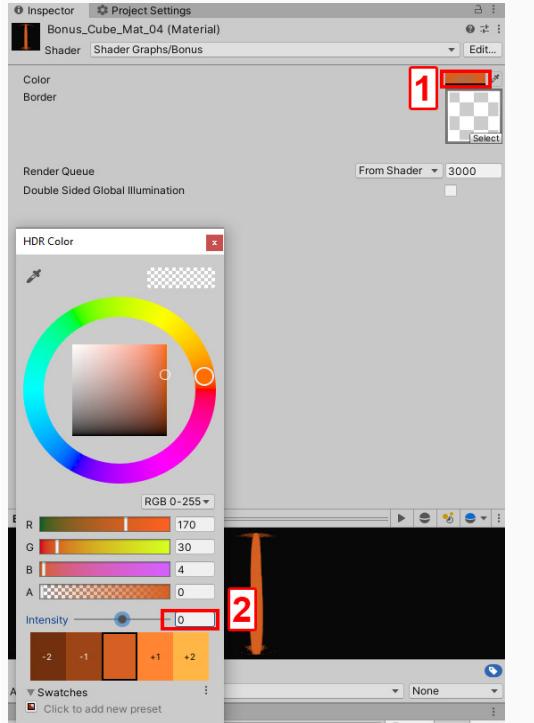
-For example, in Project tab select

Bonus_Cube_Mat_04

(Project tab: Asset → Materials → Weapons → Bonus_Cube_Mat_04)

-Click on color (spot 1)

-Adjust Intensity (spot 2)



Optimization

This section introduce optimization techniques.

Table of contents:

Occlusion Culling

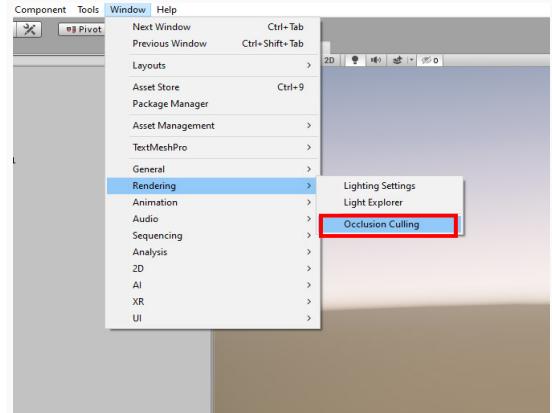
[Link](#)

Occlusion Culling

What is the occlusion culling used for?

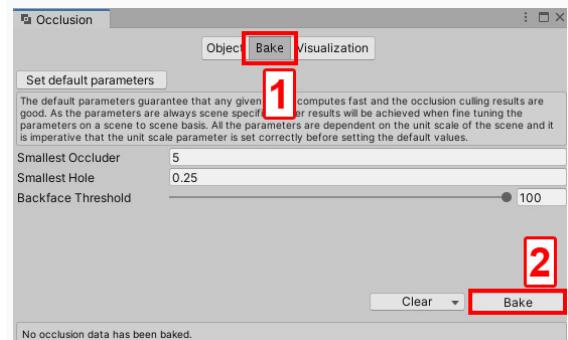
Objects that are not visible are not displayed (for example objects behind a wall).

-Go to Window > Rendering > Occlusion Culling



-Press button **Bake** at the top of the occlusion window (spot 1)

-Set **Smallest Occluder** and/or **Smallest Hole** values if needed



For information values used for the demo are:

Smallest Occluder: 30

Smallest Hole: 1

Backface Threshold : 100

-Press button **Bake** (spot 2)

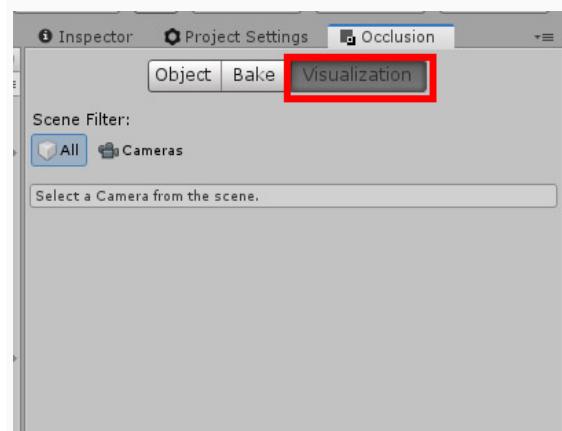
Wait until the process is complete

-Press button **visualization** to see the result

-In Hierarchy tab select **Cam_P1**

(Hierarchy : GAMEPLAY → CAMERAS → Grp_Cam_P1 → Pivot
→ BoosterShake → Cam_P1)

-Save your scene



Environment Prefabs

This section describes how to used environment prefabs included in the asset.

Table of contents:

Overview

[Link](#)

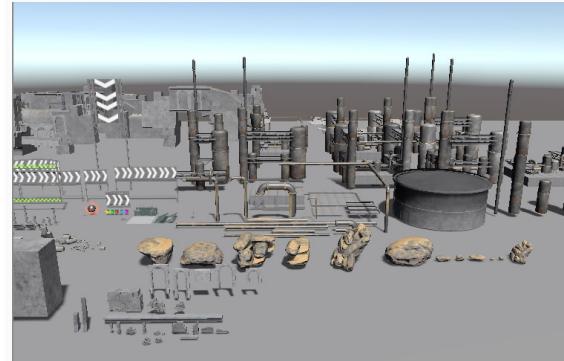
Tips

[Link](#)

Overview

If you want to see all environments prefabs included in the asset open **Showroom** scene.

(Project tab: Asset → Scenes → Showroom → Showroom)



IMPORTANT:

Do not use the prefabs contained in **Basic** folder.

(Project tab: Asset → Prefabs → AS → Env → Basic)

These prefabs are not useful as is.

They serve as a basis for building the useful prefabs.

IMPORTANT:

Without breaking a prefabs it is possible to

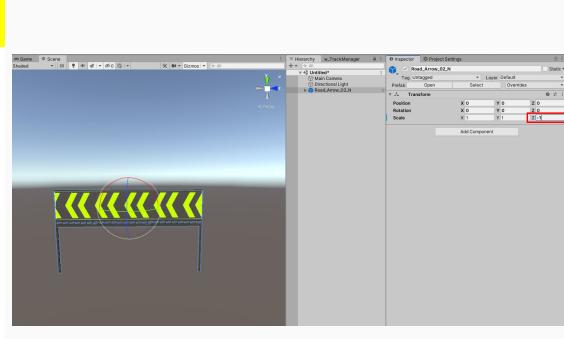
- hide / unhide object included in the prefab
- move, rotate, scale an object included in the prefab

Tips

To reverse the direction of the arrows:

-Select arrow prefabs

-In hierarchy tab set z to -1



Unity Terrain

This section describes how to create and setup Terrain.

Table of contents:

Overview	Link
Terrain Setup	Link
Add and connect new terrain	Link
Terrain Tips	Link

Overview

IMPORTANT: You need to create new Terrain for each new track scene.

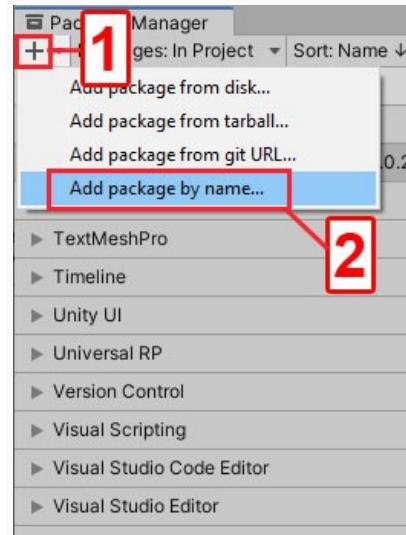
Make changes to a Terrain modifies it at project level.
Don't duplicate a Terrain, always create new one.

To make the best use of all the possibilities of the Terrain, it is recommended to install the [Terrain Tools](#) package

-Go to Window > Package Manager

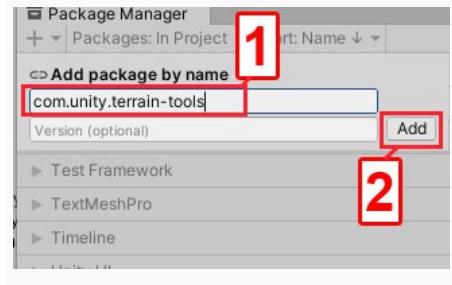
-Click on **arrow** on the right of plus icon (spot 1)

-Choose **add Package by name** (spot 2)



-Write the text: [com.unity.terrain-tools](#) (spot 1)

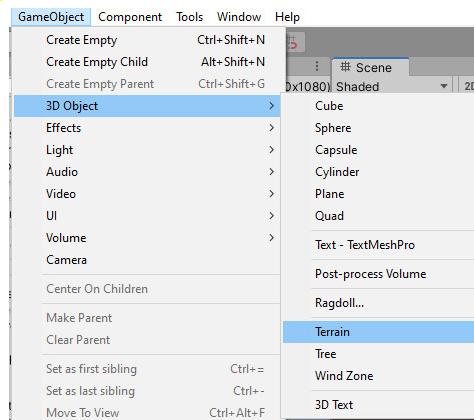
-Press **Add** button (spot 2)



Terrain Setup

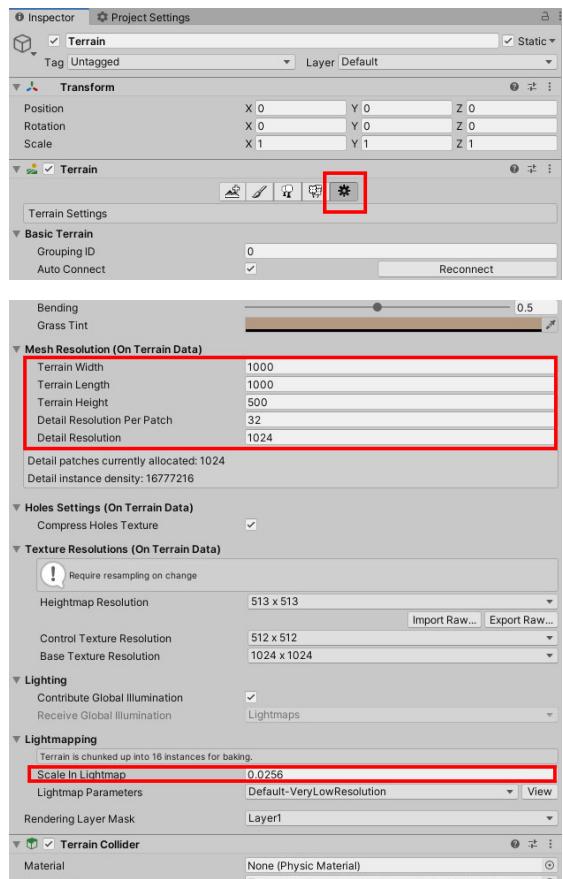
-To create a new Terrain :

Game Object → 3D Object → Terrain



-Select **Terrain** in hierarchy tab.

In Inspector tab click on **Terrain Settings** button



The demo included in the asset uses the following parameters:

Terrain Width : 1000

Terrain Length : 1000

Terrain Height : 500

Detail Resolution Per Patch: 32

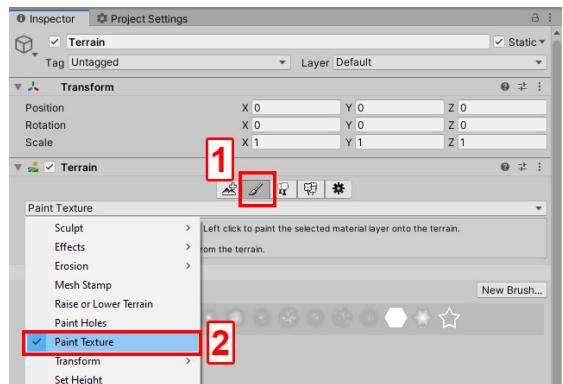
Detail Resolution : 1024

Scale in lightmap : 0.05

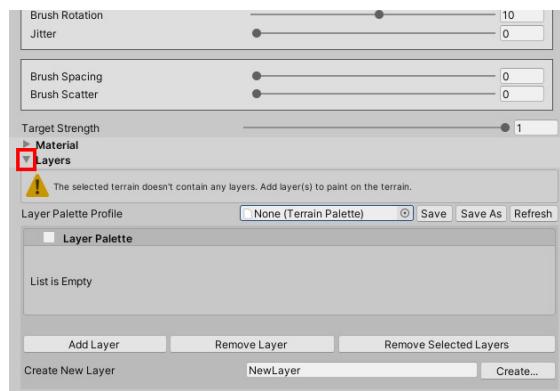
-Select **Terrain** in hierarchy tab.

In Inspector tab click on **Paint Terrain** button (spot1)

Then select **Paint Texture** (spot2)

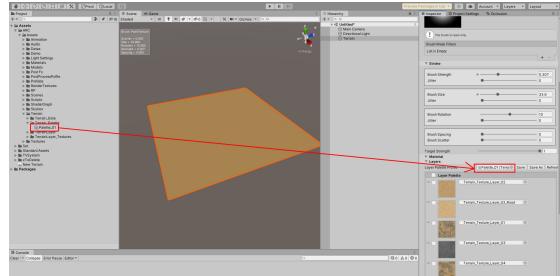


-Click on triangle to open layers Tab

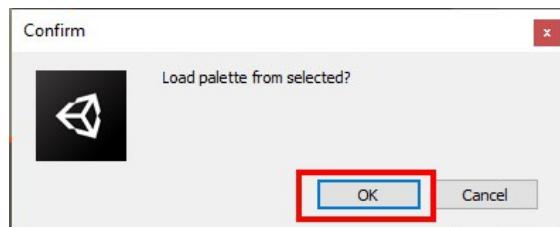


-From project tab drag and drop **Palette_01** to **Layer palette profile** in **Inspector Tab**

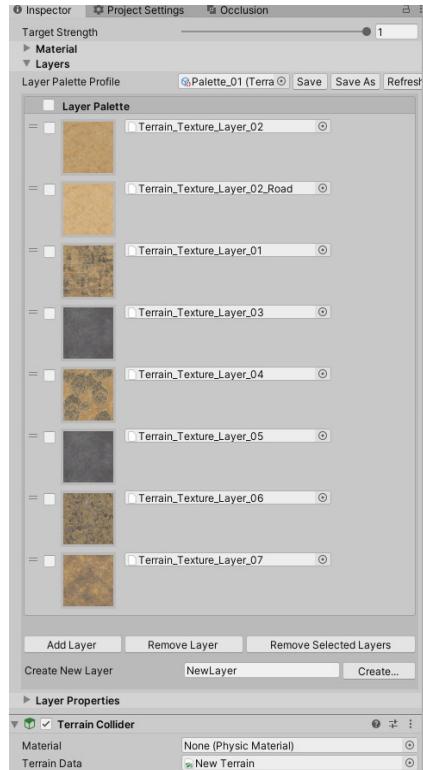
(Project tab: Asset → Terrain → Terrain_Palette → Palette_01)



-Confirm by clicking **ok** button



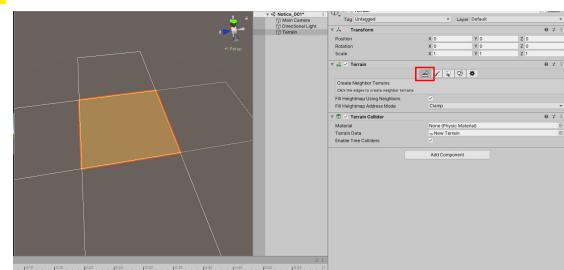
Terrain Layers textures are now visible in tab Layer



Add and connect new terrain

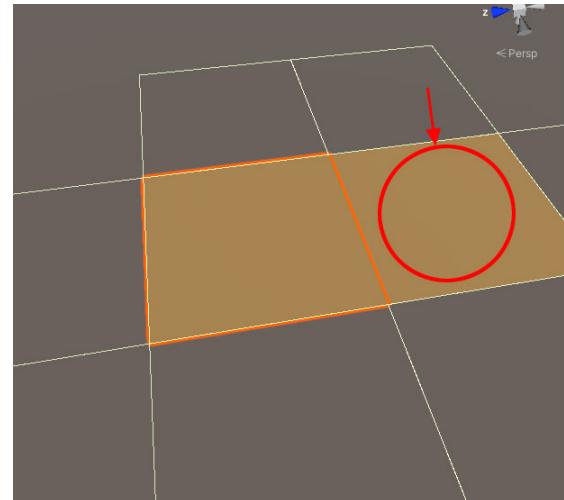
-In hierarchy tab select a Terrain

-In Inspector tab click on **Create Neighbor Terrain** button (spot1)

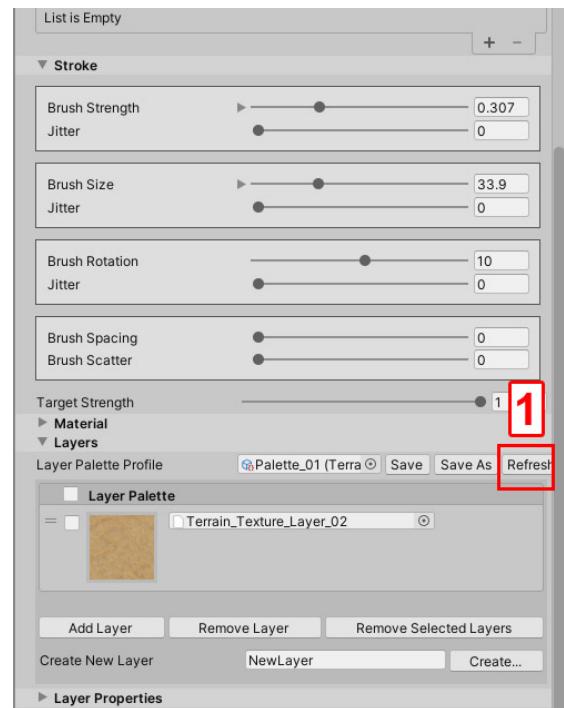


-In Scene view click on a square.

A new Terrain is add and connect to the other Terrains

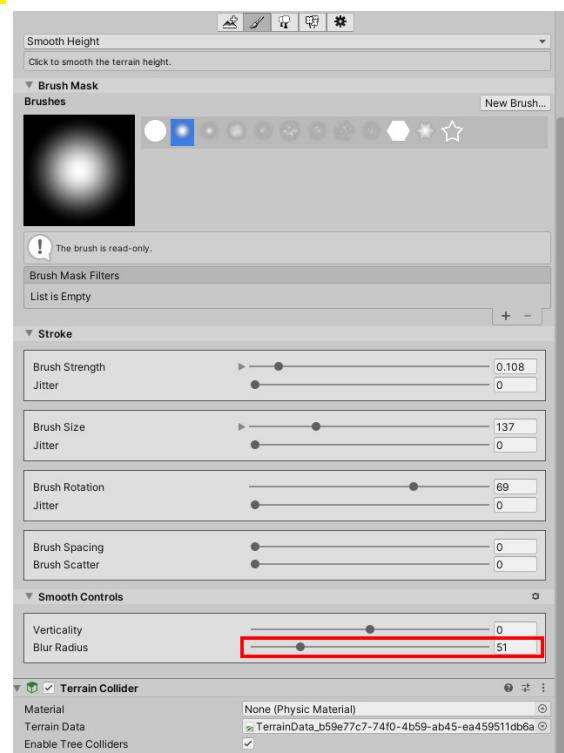


-In Inspector tab click on **Refresh** button to refresh all terrain textures included in **Palette_01** (spot1)



Terrain Tips

When you use **Smooth Height** :
To increase the smooth value increase **Blur Radius** parameters



Screenshots

This section describes how to take screenshots for the user interface.

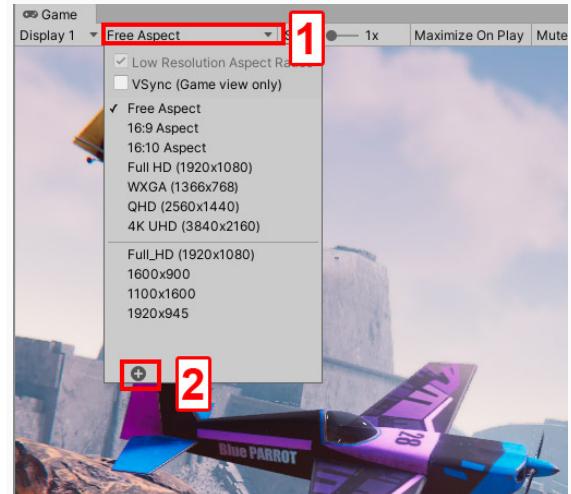
Table of contents:

How to create thumbnail and full screen images for track and championship

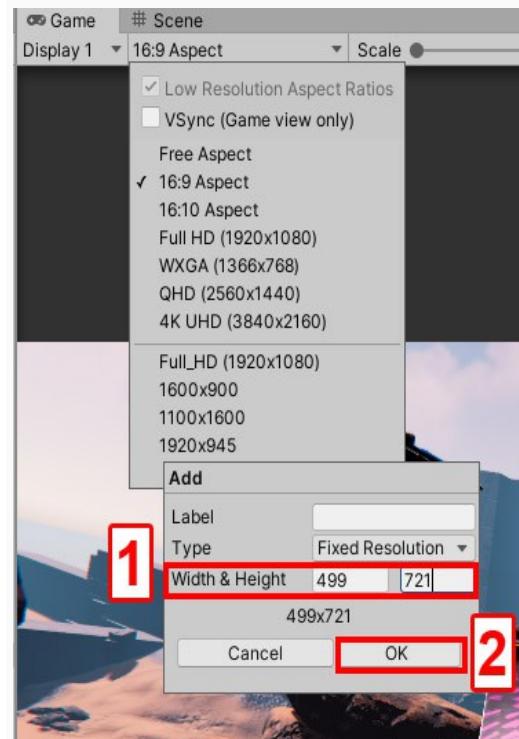
[Link](#)

How to create thumbnail and full screen images for track and championship

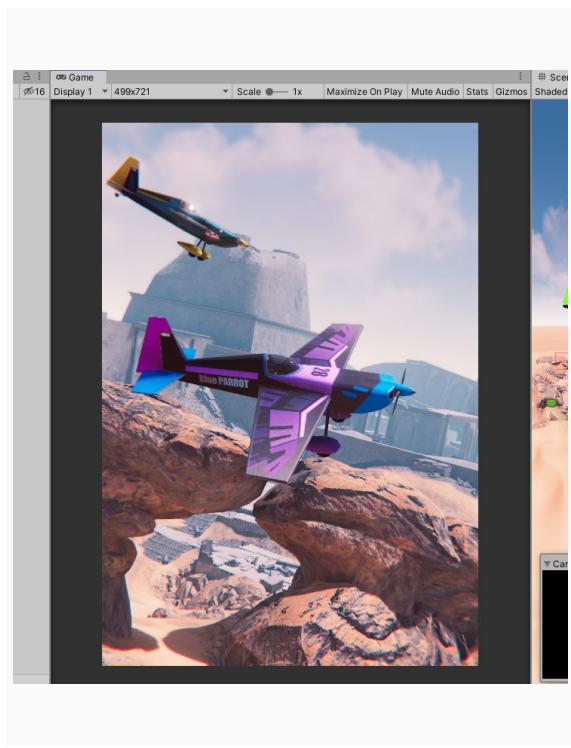
- Open Game tab
- Click on triangle in Resolution tab (spot1)
- Click on plus icon (spot2)



- Set width to 499 (spot1)
- Set height to 721 (spot1)
- Press Ok Button



-Take a screenshot



Shadow Distance

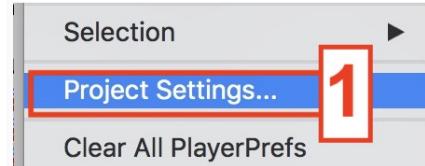
This section describes how to visualize and choose the shadow distance depending the scene (Main Menu, Track,...)

Table of contents:

How to: Visualize Shadow Distance	Link
How to: Edit the Shadow Distance for the Main Menu	Link
How to: Edit the Shadow Distance for tracks (all tracks at the same time)	Link
How to: Edit the Shadow Distance in a specific track	Link

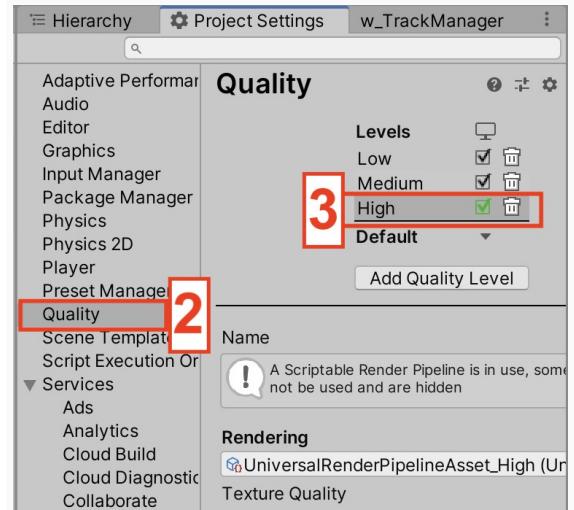
How to: Visualize Shadow Distance

Go to [Edit → Project Settings](#) (spot 1)



Select [Quality](#) (spot 2)

-Select the quality you wanted to test (spot 3).
(As an example press [High](#) quality button)

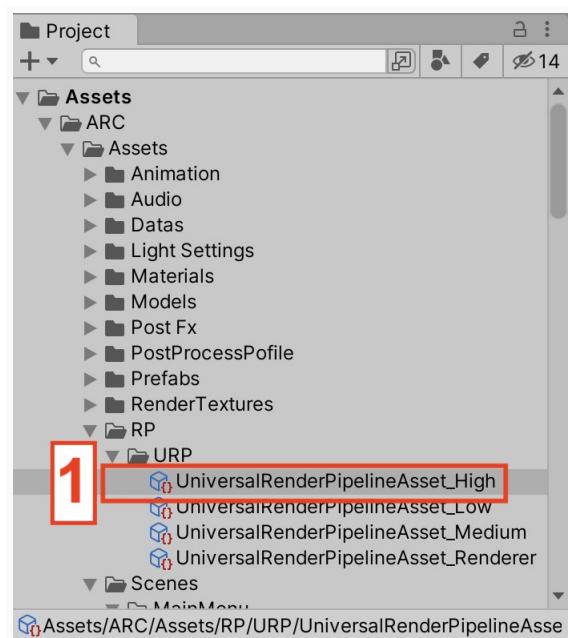


-In Project select the URP pipeline corresponding to the quality we have already selected.

In our example select

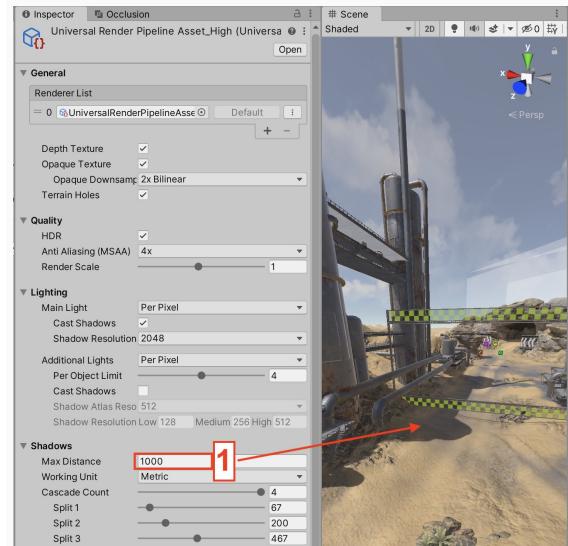
UniversalRenderPipelineAsset_High (spot 1)

(Project tab: Assets → RP → UniversalRenderPipelineAsset_High)



-In the Inspector change the **Max Distance Shadow** (spot1)

The modification is visible in the scene view.



IMPORTANT:

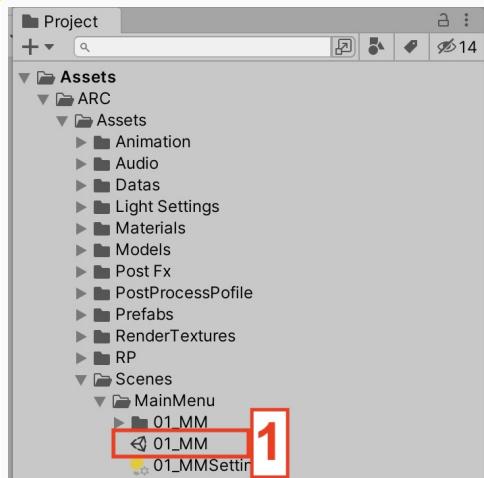
Max Distance shadow is override when a scene is loaded.

The next 3 sections explains how to change the shadow distance when a scene is loaded.

How to: Edit the Shadow Distance for the Main Menu

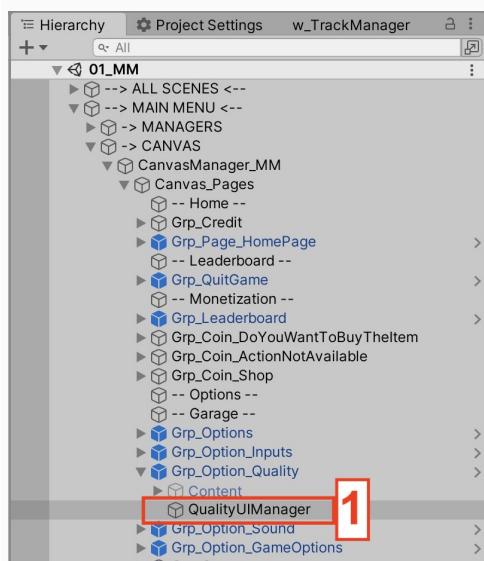
-Open the main menu scene **01_MM** (spot1)

(Project tab: Assets → Scenes → 01_MM)



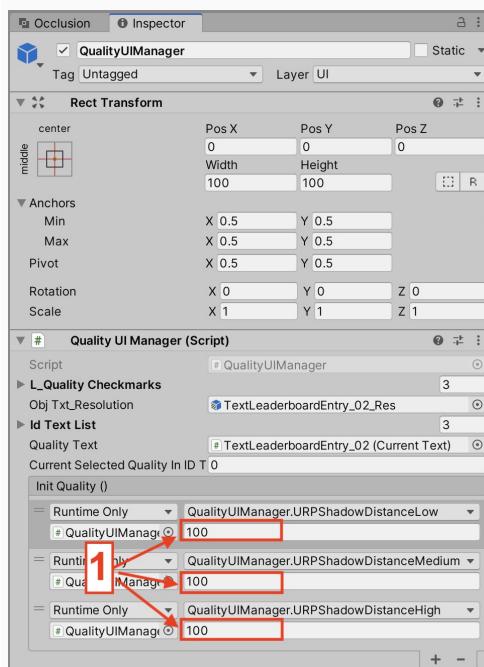
-In the Hierarchy select **QualityUIManager** (spot1)

(Hierarchy: MAIN MENU → CANVAS → CanvasManager_MM → Canvas_Pages → Grp_Option_Quality → QualityUIManager)



-In the Inspector change the distance shadow for each quality (spot 1)

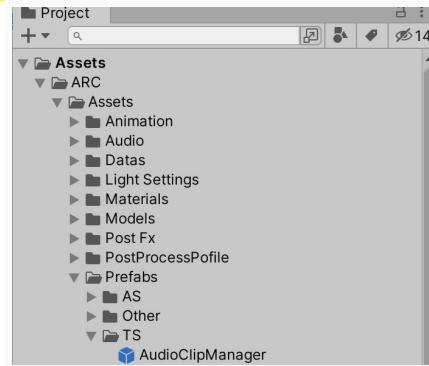
-Then save the scene **Ctrl+S**



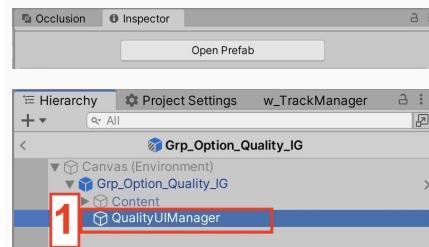
How to: Edit the Shadow Distance for tracks (all tracks at the same time)

-In Project tab select **Grp_Option_Quality_IG** (spot1)

(Project tab: Assets → Prefabs → TS → Settings → Grp_Option_Quality_IG)



-In the Inspector press **Open Prefab**.

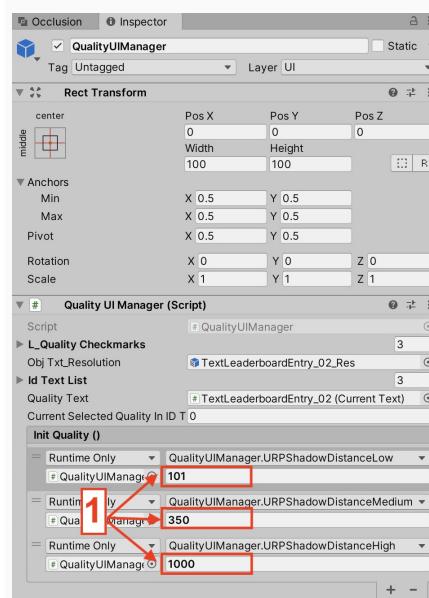


-In the Hierarchy select **QualityUIManager** (spot 1)

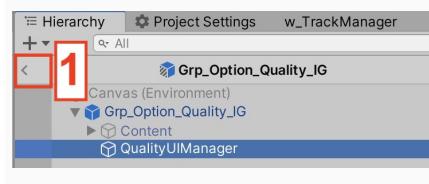
(Hierarchy: Grp_Option_Quality_IG → QualityUI Manager)



In the Inspector change the distance shadow for each quality (spot 1)



-In the Hierarchy press < button to leave the prefab;

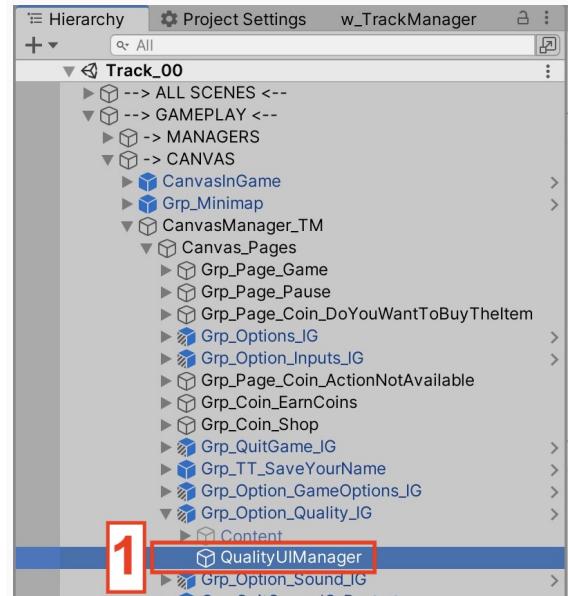


How to: Edit the Shadow Distance in a specific track

-Open your track scene (spot1)

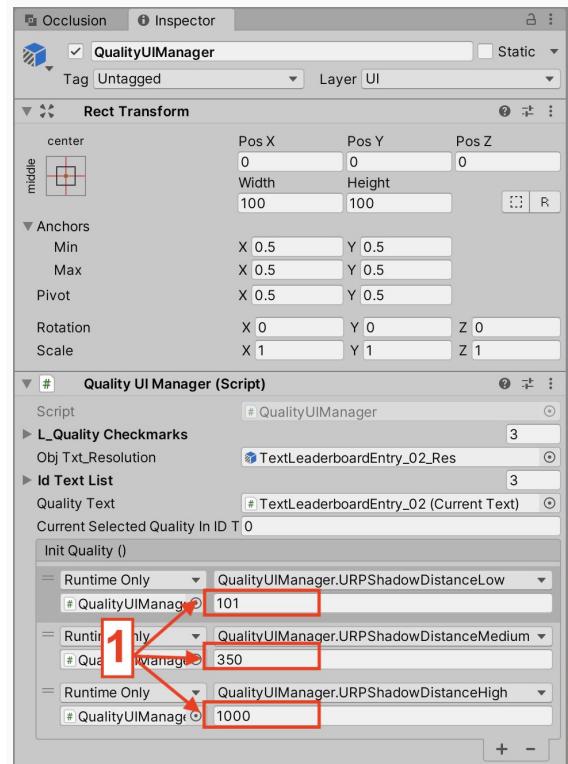
-In the Hierarchy select **QualityUIManager** (spot1)

(Hierarchy: GAMEPLAY → CANVAS → CanvasManager_MM → Canvas_Pages → Grp_Option_Quality_IG → QualityUIManager)



-In the Inspector change the distance shadow for each quality (spot 1)

-Then save the scene **Ctrl+S**



FAQ

Table of contents:

Common issues with Alternative path	Link
Airplanes turn too suddenly	Link
How to: Optimize AI Booster	Link
How to: Disable the Area Limit to easily edit the track	Link

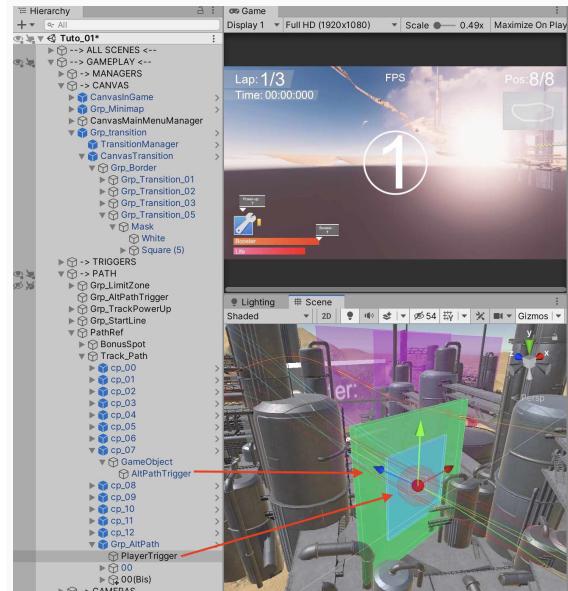
Common issues with Alternative path

This section gives some examples to prevent some common issues with Alternative path.

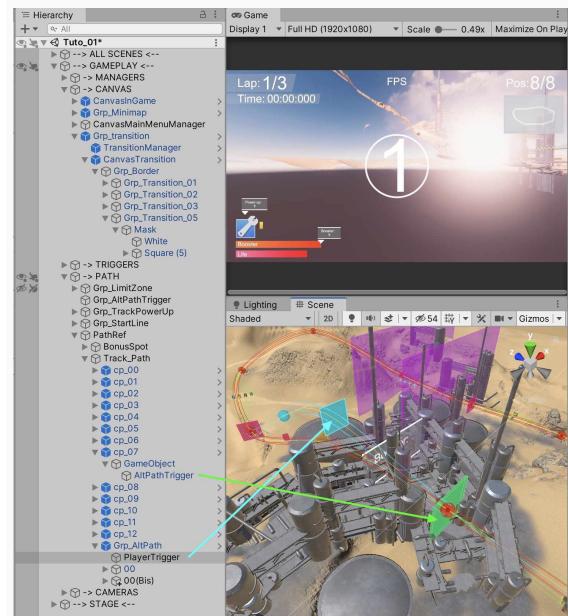
1-Issue: Some trigger are ignored. (2 triggers in the same position)

A trigger **must not** touch another trigger.
It will cause issues. Some triggers can be ignored.

1st case: Alt Path and Player Trigger touch each other.
(Wrong) It will cause issue.



2nd case: Alt Path and Player Trigger don't touch each other.
(Ok).



2-Issue with Alt Path and Power-up group.

First, the vehicle must choose a path (spot 1)

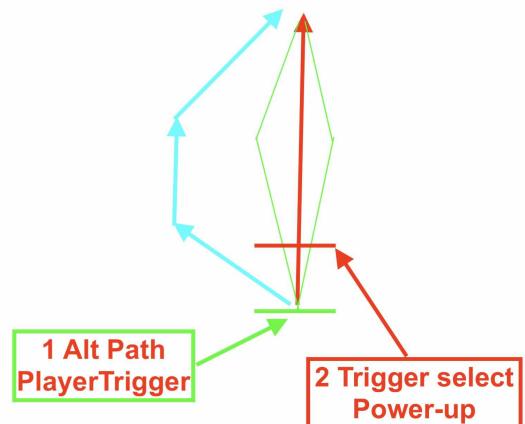
Then if the vehicle follows the red path, the vehicle chooses a power-up (spot 2) .

(Ok)

It creates an issue if:

- The player select a power-up on the red path
- Then choose the blue path.

(Wrong)



More info in section Track → Alternative Path
→ Page 125

3-A power-up is connected to the last Alternative Path point and it creates an issue.

Do not create a Power-ups group connected to the last point of the Alternative Path.

Remember that you can create as many points as you want on a Alternative Path.

Issue: Airplanes turn too suddenly

Smooth the path followed by the airplanes.

Check:

- Power-up Path
- Alt Path
- Angle between 2 checkpoints.

How to: Optimize AI Booster

-You can Add AI Booster every 2 or 3 checkpoints.

-Avoid adding AI Booster at the end of the track.
It is not pleasant for the player to be passed at the
end of the race.

How to: Disable the Area Limit to easily edit the track

-In the Hierarchy on the left of **Grp_LimitZone**
(Hierarchy: GAMEPLAY → PATH → Grp_LimitZone)

-Click on the **Eye** icon (spot 1) to hide the **Grp_LimitZone** in the scene view.

-Click on the **Hand** icon (spot 2) to disable the ability to select **Grp_LimitZone** in the scene view.

