

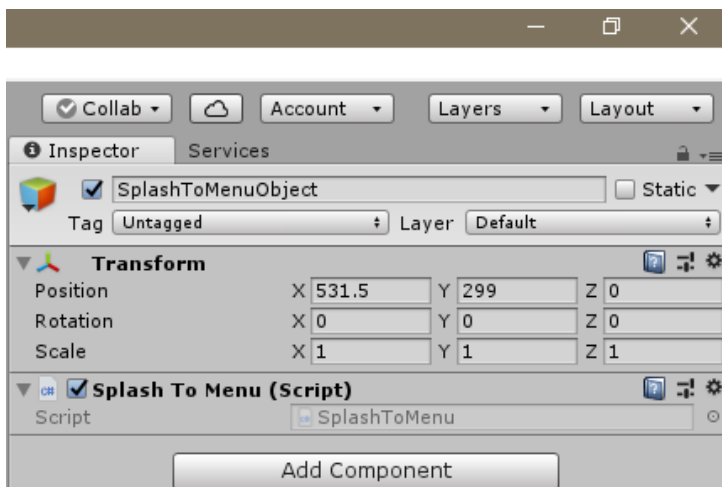
## Racing Project Kit by [SpinMotion Games](#)

The content it's separated by 9 scenes:

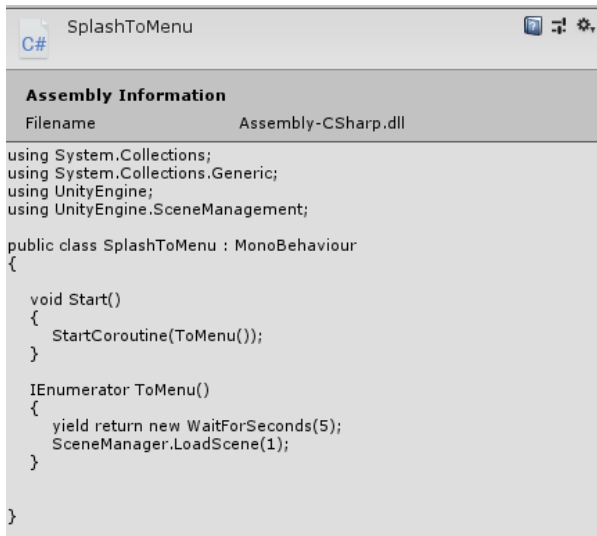
1. **Splash Screen**
2. **Main Menu**
3. **Play Menu**
4. **Track01**
5. **Track02**
6. **Credits**
7. **Reset Player Prefs**
8. **Track Creator**
9. **Settings Menu**

From page 25 you will find the latest features added in new versions (Mobile Support, Real-Time Positioning System, Track Editor and much more).

So let's start with the first scene: [Splash Screen](#)



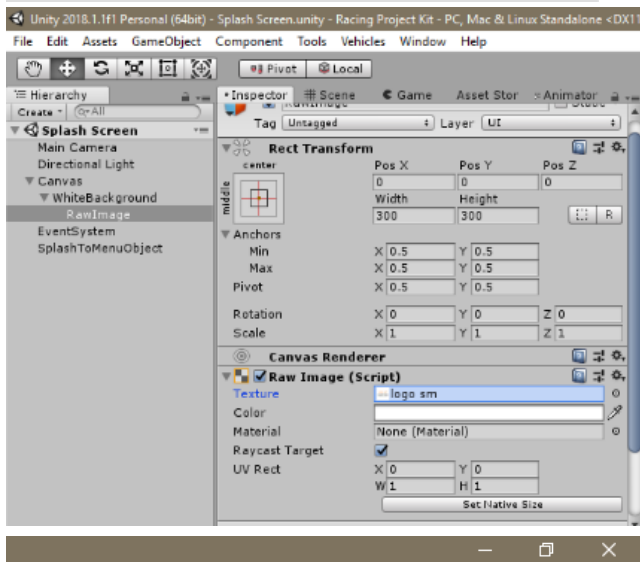
This scene uses the `SplashToMenu.cs` script and it's attached to an object in the inspector named "SplashToMenuObject" to be functional.



Inside the script if you want to change the seconds to make the transition from the splash to the menu you must change the line in the script that says:

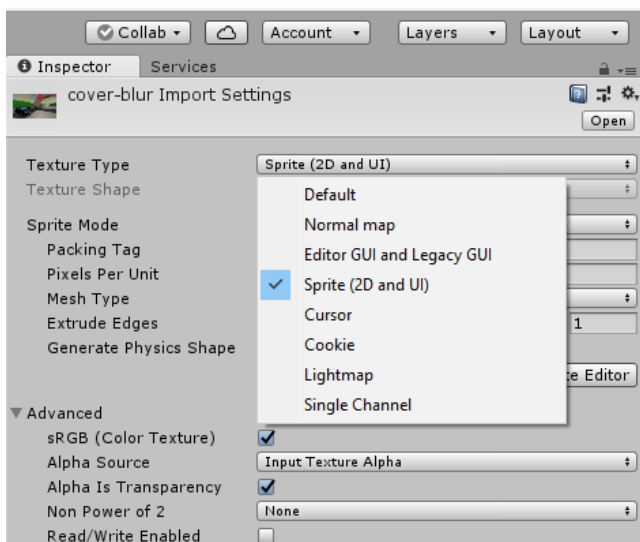
yield return new WaitForSeconds (5);

The (5) are the seconds, change the number as you like it.



And in the inspector to change the photo, import a new one by clicking right-click > “import new asset” and then select your photo.

Once you have done this replace the texture inside the raw image object with the photo you imported and done.

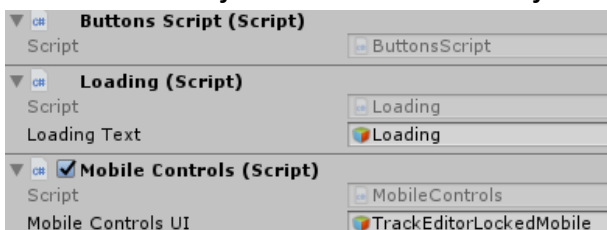


Remember: Once you imported your photo into the inspector, change it to Sprite (2D and UI) mode so you can use it in the UI Canvas.

Moving up to the second scene: [Main Menu](#)



All the content of this scene it's the main menu Background raw image, the panel content (Play Game, Credits, Quit Game and the Cash Display), and then the Button Object and the Cash Object.



The most important object in this scene it's the button object since it's the one attached to the buttons and it sends functions to them.

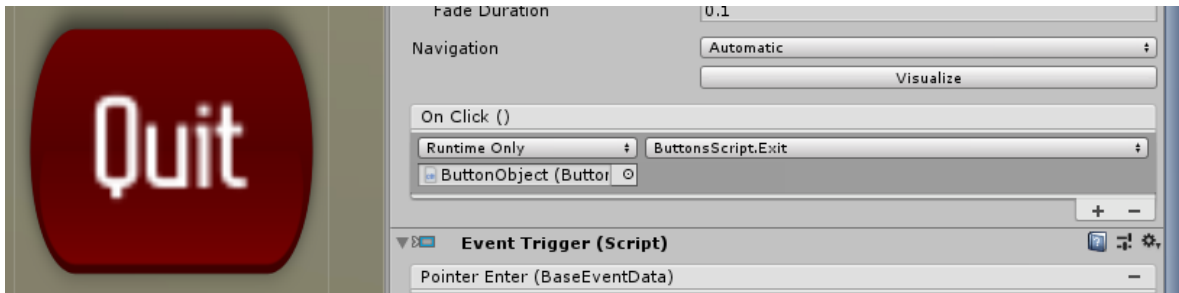
Inside this object, there are 3 scripts: Buttons, Mobile Controls, and Loading.

The Buttons Script has all the scenes of the project inside of public voids, which means you can assign them to a button so this one it's going to load the mentioned scene of the void.

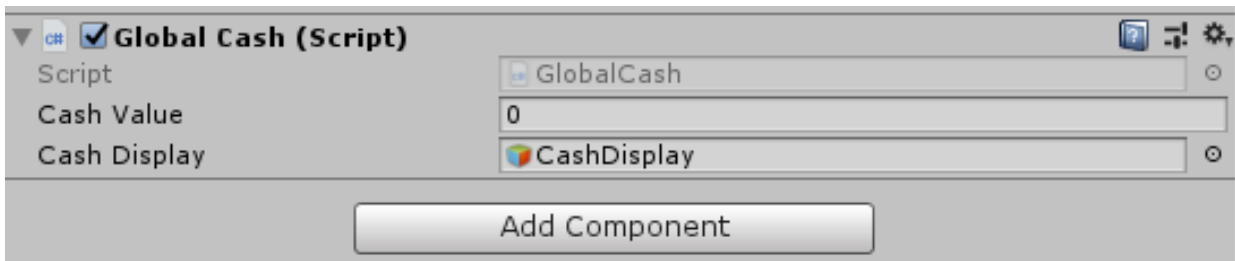


This is an example of a button using a “void” from the button script which it's attached to the button object.

In the On Click () you add the button object and then select the desired script, in this case, the button script and inside of that script what scene do you want to load when pressing the button, in this case, the play mode.



The same method it's used for other menu buttons in other scenes, in the example of the picture of the quit button you can see that the button uses the Exit void of the buttons script in the OnClick routine.



And then, the “global cash object” it's used to attach the global cash script which saves the money you win in the races and then shows it in the cash display.

The Loading scripts turns on a text when you click play or track editor to indicate that the project is now loading the next scene.

You will see further in this documentation how to set up the race rewards when finishing a race and how to save the money in the player preferences to show later in menus and save it when closing and re-opening the game again.

## Moving up to the third scene: Play Menu



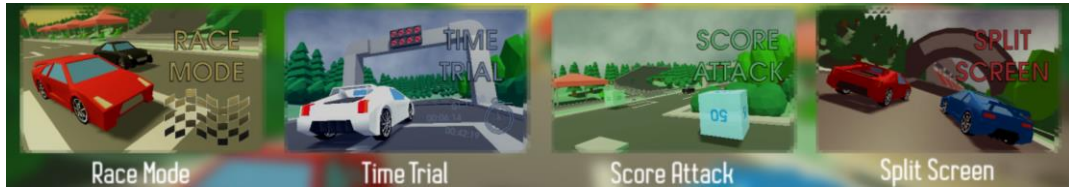
In the Canvas, you have the buttons to select the race mode, back to the main menu button in the top left corner, and the cash display when selecting the car. Then, when you select the race mode, the car selector appears and after that, the track select panel with the laps and bots selector appears too. Also, you have the background image, the “PLAY MENU” title, some texts that shows what mode, car, and color you’ve selected, and a loading game object. Also, in the bottom corner at the left you have a show info button which turns on some texts to explain each game mode.

In the Inspector you can find the button object (the same one used in the main menu) but modified, to make the play menu buttons work.

The game objects you can find are 1- Mode Buttons, 2- Car Buttons, 3- Cash Object, 4- Unlockable Object, and 5- Button Object (this one It’s explained in the main menu scene).

So, let me explain these game objects.

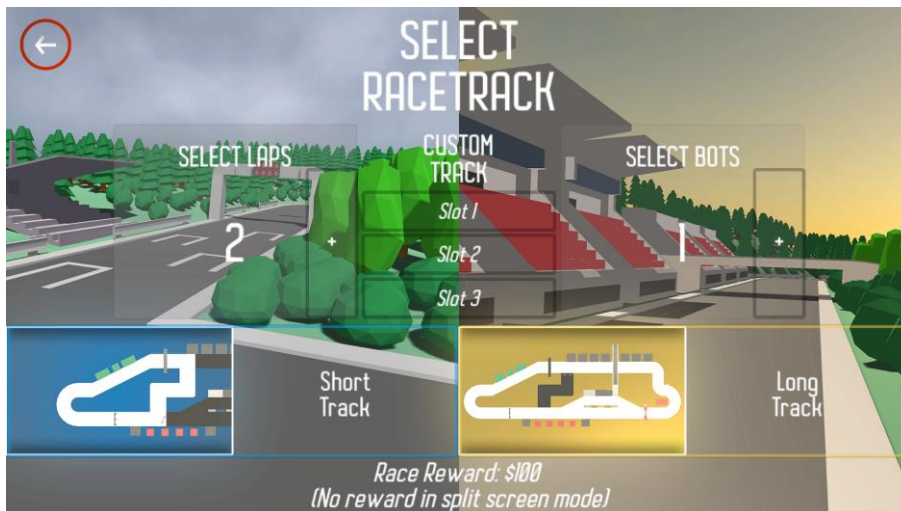
## 1- Mode Buttons:



This object assigns an integer value to a variable called “RaceMode” in the *mode select* script that it has attached. That integer value it's later used in the track01 and track02 scenes to see what race mode the player wants. Also, the mode buttons script activates the **continue button** once the mode is selected, and it shows some effects around the mode button that were pressed.

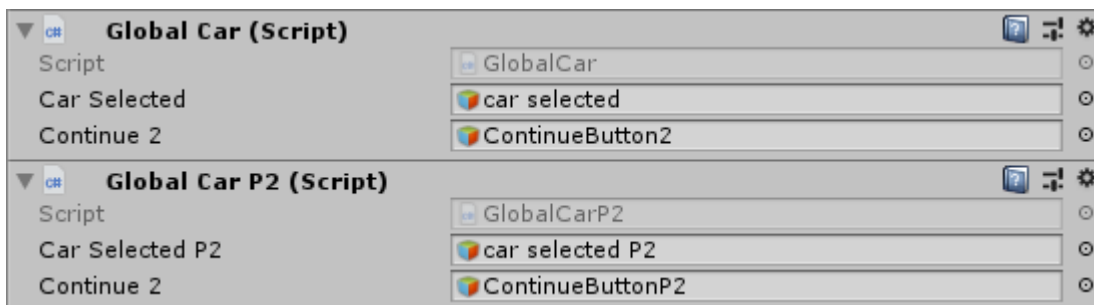
Once the player hits the **continue button** after selecting the mode, the **Car Selector** appears, and once the car is selected the second selector (**Car Selector P2**) will also appear but only if the split-screen mode was selected. The **continue button** is a green button that appears at the bottom of the screen when you select a race mode. When you hit it the car selector will appear and then when you select a car, it will appear one last time so the player can hit it again and move to the **Tracks Panel**. From this panel, you can select 2 tracks that the project has (track01 and track02) or a custom track created with the Track Creator, also, you can select the laps (only in race mode and split-screen), and the number of AI bots (only in race mode). We will see the custom tracks slots and how to customize the number of laps and bots later.





## 2- Car Buttons:

This object has 2 scripts attached: **Global Car** and **Global Car P2**, which it's the same but for player 2 if split-screen mode it's selected.

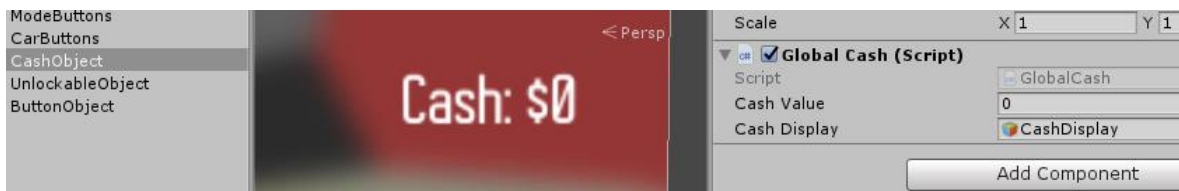


When the car logo is pressed, **Global Car** script sets an integer value in a variable called Car Type, that later track01 or track02 gets to see what car did the player picked, it's the same method that the mode selection uses.

In Button Object you will find **Car Selector** script that makes the left and right arrows work, the same goes for the player 2 in split-screen with **Car Selector P2**.

In the body of each car of the play menu you will find the **Color Picker** script. **Color Picker** & **Color Picker P2** scripts have a system to change the color of the car material with 3 sliders: **HUE** color, **Saturation**, and **Brightness**. Via inspector, you have to attach these scripts to the body of the cars and set the UI sliders. Then, in track scenes, you have to attach via inspector the car bodies to the **ColorSelect** script inside **ColorSelectObject**, the same goes for player 2.

### 3- Cash Object:



This object it's used to attach the global cash script which saves the money you win in the races and then shows it in the cash display (it's the same one of the main menu).

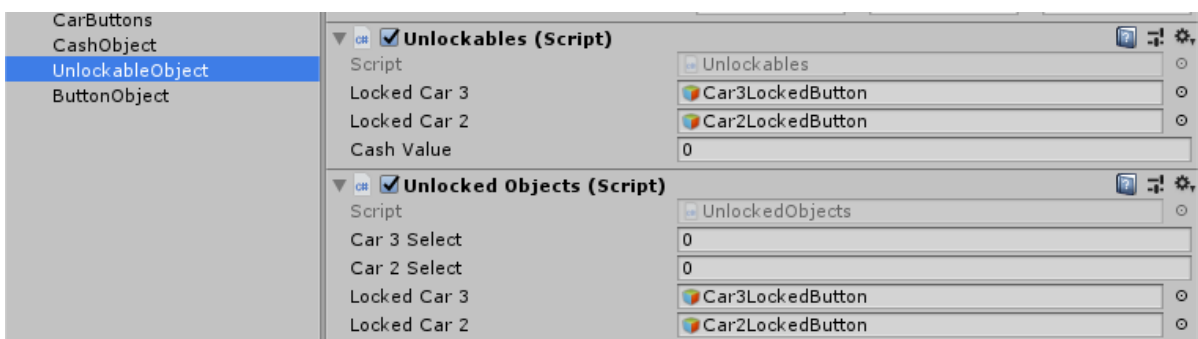
### 4- Unlockable Object:

This object it's used to unlock with the game cash the 2 cars that are locked in the project. It uses 2 scripts: **Unlockables** and **Unlocked Objects**. You can use this method to make locked things and unlock it for anything (cars, colors, maps and more depending on your project) it's simple and effective.

You will need 2 game objects: the locked one and the unlocked one, and set the locked one to the unlockable script to let unity know that object it's possible to unlock and set it also to unlocked objects to save if you have unlocked it yet.

In the scene Player Prefs, you can add money to buy the locked objects without doing races or you can reset all the preferences (unlocked objects go again as locked objects and money goes to \$0, basically it resets everything to default) this scene is for test purposes and it's not included in the game build.

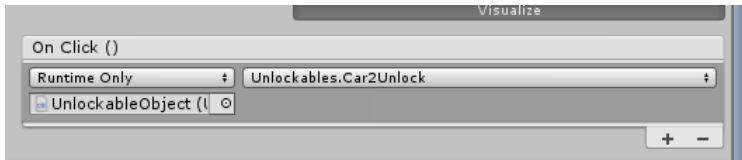
You will see more information about that scene further, now going back to the unlockable object, I'll explain the usage of the 2 scripts.





In **Unlockables** and **Unlocked Objects** you need to set the locked objects, these objects must be buttons, because it needs to set up an OnClick routine where the script gets the money and deactivates the locked object so you can use the unlocked one, so behind the locked object put the unlocked one.

Both of the scripts have the variables of the 2 locked objects in the project (locked car 2 & locked car 3) you can set new ones by following how the others were already made.

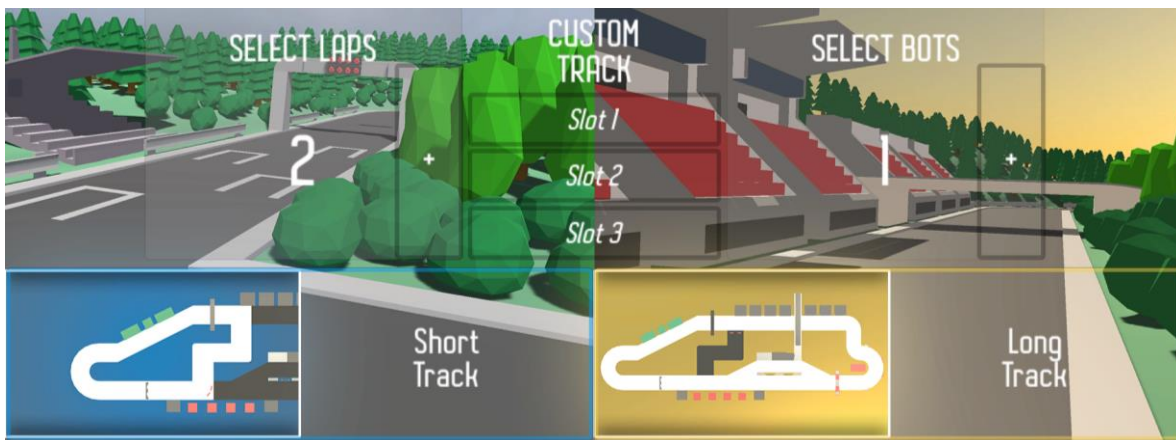


As I said before the locked object must be a button because in the OnClick routine you have to attach the unlockable object, select the unlockables and in the case of the Ferrari: Car2Unlock and Lamborghini: Car3Unlock, you can set your names as I said earlier to put your own unlockables objects.

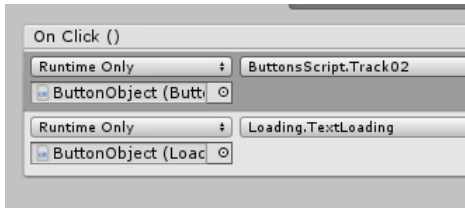


Once you click in the game the locked object the unlocked one pass to be visible and with the global car routine ready to select it to play.

Track select & Laps Selector:



Once the player selected a race mode, a car, and a color for the car, the Track Panel it's activated. In this panel you can select 2 tracks (short & long) and the quantity of laps required to do to finish the race, this only appears in race mode and split-screen mode, because time & score modes use only 2 laps: 1 for warmup and 1 more to complete the objective.



The short & long buttons work with the button object in the OnClick routine, it loads the track and activates the Loading Panel to show the game it's loading.

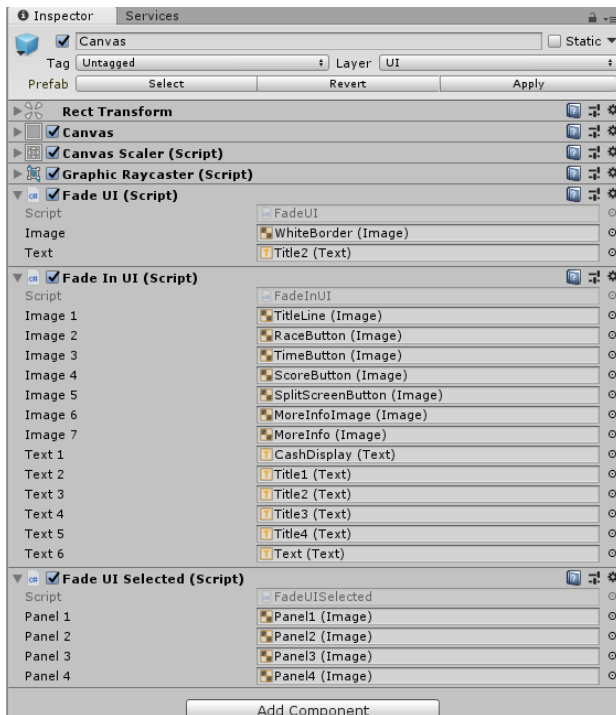
The lap selector - & + buttons work with voids in the Lap Selector script which it's attached to the Button Object. You can change the min & max of laps in that script too.

So this is it for the play menu scene, now before moving to the scenes of the track, as you may have noticed there are some game objects deactivated by default. For example, the cars panel, track panel, loading, and more, **leave this deactivated**. After all, the scripts automatically activate these game objects when it's needed, if you enter to button object you will see all that game objects assigned because they are programmed to be activated in the scripts.

From the v1.3 version of Racing Project Kit to the nowadays one, the cars are visible in 3D with real-time color-changing instead of pictures. Also, a lot of effects were added.

So, the first change in the hierarchy it's a new game object called **Audio Files** which has a **wind FX** that it's used when you enter the car panel mode after choosing a game mode.

And then, the rest of the changes are mainly in the *Canvas* and *Button Object* and then some new scripts in *Car Buttons*.

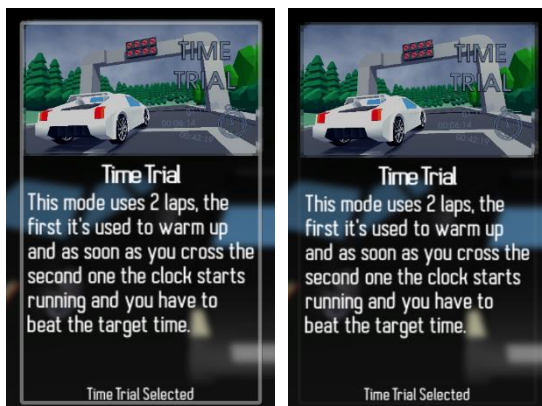


Let's start with the **Canvas**, we can see 3 new scripts in here: **1-Fade UI**, **2-Fade in UI** & **3-Fade UI Selected**.

**1-Fade UI** is used to keep the *white border* and *title 2* (Select Game Mode text) fading in and out changing the alpha of the image color.

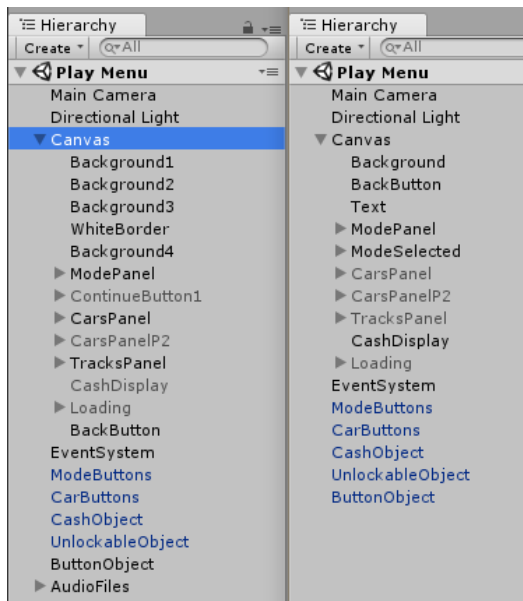
**2-Fade in UI** makes a fade-in effect on the game objects when the scene starts. You can set the seconds of how much each object will wait to fade in by changing the order inside the script.

All of these scripts are commented so it's easy to understand and modify them.



**3-Fade UI Selected** appears when you select a game mode and a white border makes a fade in and out around the game mode button.

This it's used to indicate the player which game mode selected in a good looking way.



1.3 hierarchy at the left, 1.2 at the right.

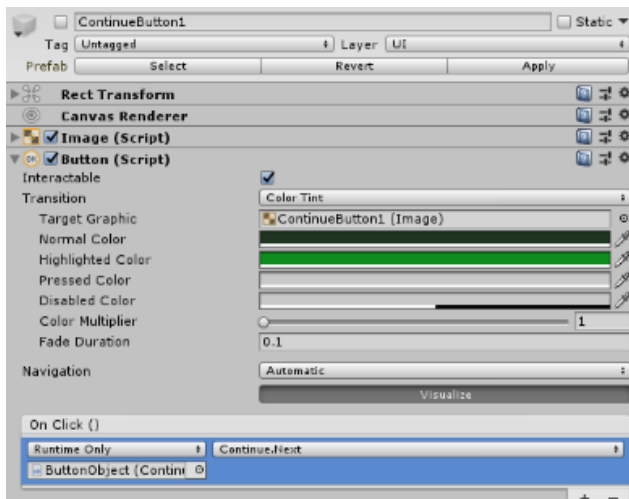
Keeping up with changes in the Canvas, another thing we can see it's that we have **4 backgrounds** instead of 1.

**Background 1** it's the basic one that appears when selecting a game mode, then **Background 2** it's the background for the **cars panel** and it appears by sliding from the right to the center. It has a script attached to it called *Fade in Background 2*, remember to keep this script deactivated because it gets activated when you press continue in the mode button, we will see that later.

Then, **Background 3** it's a detail added to the Background 2 and it fills from right to left with a script called *Fill Amount* that it's also activated by the continue button of the mode panel menu, so leave this script deactivated. The final one, **Background 4** appears in the tracks panel and it also uses the *Fill Amount* script like the third background.

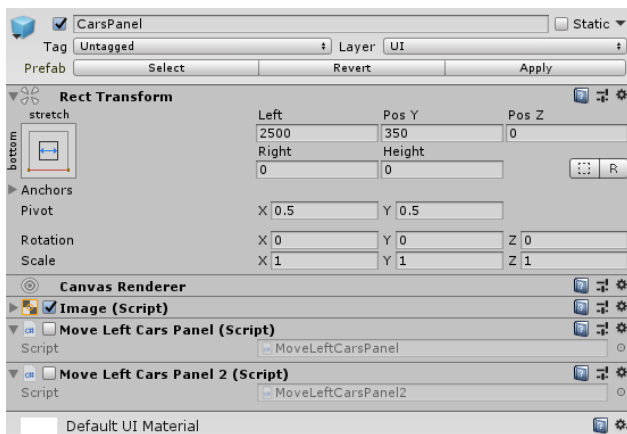
**White Border** it's a line around the mode panel which it's attached to the *Fade UI* script we saw recently in the canvas to make the fade effect.

Moving up to **Mode Panel** this has a script attached, called *Move Left Titles* which is deactivated by default because it gets active when you select a game mode and press continue, so it moves the entire mode panel to the left. Inside mode panel we have the 4 game mode buttons (**RaceButton TimeButton, ScoreButton & SplitScreenButton**), a **MoreInfo** button which shows the information of each game mode making appear some texts (**Descriptions**), the **Titles** has the texts which say: "Select Game Mode", "Play Menu" and a white line that has below. All of these objects are attached to the scripts in the Canvas that we saw (*Fade UI & Fade in UI*) to play the fade effects when you enter the scene, and then we have the **Mode Selected** that gets activated when a game mode it's selected with one of the 4 game mode buttons, and it also activates the **Continue Button 1**, which it has to be deactivated by default obviously to be only active after selecting a game mode.



So, the **Continue Button 1** has the Next void of the *Continue* script attached to the **Button Object** game object of the hierarchy.

This void's the one that activates the scripts to move the backgrounds and the entire mode panel to the left to be off-screen and move the **Cars Panel** to be visible.



**Cars Panel** needs to be moved to the left to be visible only when the move scripts attached to it are activated. By default, its Left position in the *rect transform* is 2500 the scripts are the ones that move this number to 0 when they are activated by the continue button of the ModePanel.

Then, inside the **Car Panel** we still have the car buttons (called: **Car 1 Button**, **Car 2 Button** & **Car 3 Button**) but now with the 3D model of the car inside and the 4 body colors, and a script called *Menu Car Rotate* which makes the car spin 360 degrees in the menu.

**Car Select** works the same as it was in the previous version.

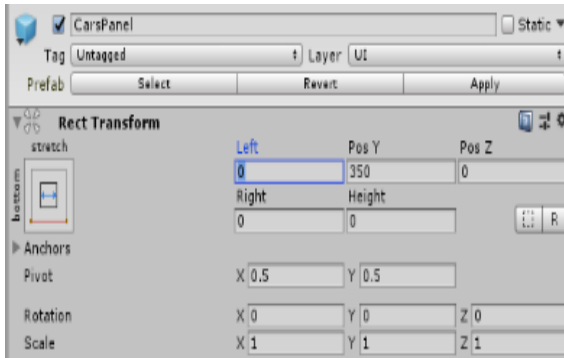
**Spot Light** was added to make brighter the car in the menu.

**Left** and **Right** are buttons with the form of arrows and works like the Lap Selector script that reads which car the player has selected.

Once we selected color and a car, the **Continue Button 2** appears and it has the Next2 void of *Continue2* script, it works the same as the first continue script, but it moves to the left the **Cars Panel** and brings to the center the **Tracks Panel** (activating *Move Left Tracks Panel* script, located in **Tracks Panel**) and activates the **Background 4**, also if the mode selected it's the third one

(split-screen) it activates the **Cars Panel P2** which it's the same but it sets the parameters for player 2 and after choosing car and color the **Continue Button P2** appears doing the same routine that **Continue Button 2** does.

Locked cars and unlockables work the same as the previous version but now the **Car2LockedButton** and **Car3LockedButton** are located inside the **Car 2** and **Car3** game objects.



If you need to edit something of the **Car Panel** or **Tracks Panel**, just simply write 0 in the Left section of the *rect transform* instead of 2500 and deactivate **Mode Panel**, then remember to put again Cars & Tracks in 2500 and Mode Panel activated by default.

**Tracks Panel** works the same as it was in the previous version, the main changes are aesthetic. **Laps Panel** it's off by default now and obviously, it gets activated if race or split-screen mode it's selected. Also, the background in v1.3 uses a separate game object called **Background 4**.

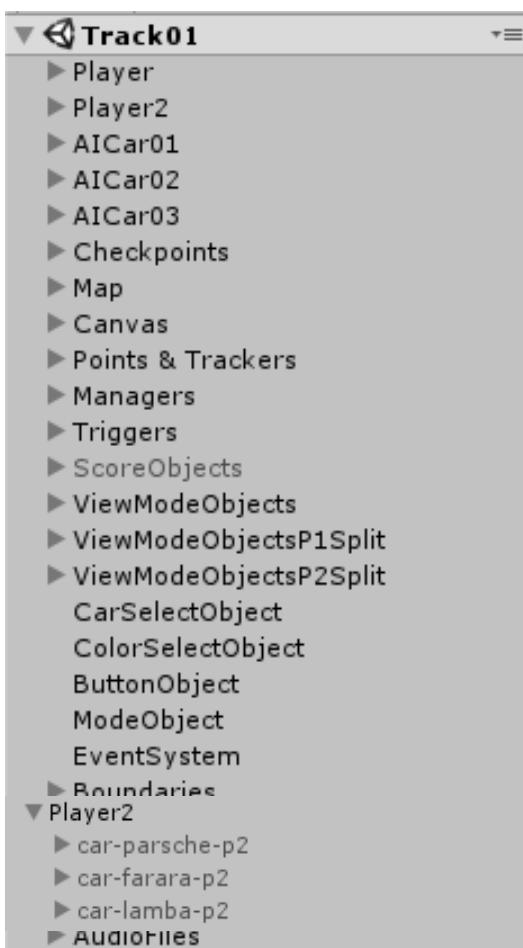
The last change we can see it's that the **Cash Display** in the Canvas it's now deactivated by default, and it only gets activated in the **Cars Panel** since it's the only moment that money system is used.



Moving up to the third and fourth scenes: [Track01](#) & [Track02](#)



Both of the tracks work with the same objects and in the same way, so it can be explained once for both scenes.



First off, the **Player** has inside **the 3 cars**. These cars get activated depending on which one the user selected from the menu. Then, the **Audio Files**: FX: Get Ready and Go / Music: Level Music and Finish Music.

Inside each car, in **Player** Game Object you will find all the components to make it work (physics scripts, wheel colliders, and more) and visible (car parts, lights, etc.).

**Player 2** game object it's the same that **Player** but for the second player in split-screen, the car selected and color, it also uses audio files.



Moving up to the next game object **AICar01**, this object uses the same components of player cars but the difference it's that it doesn't have the option to change color neither has audio files.

You can see more information on how to set up a car with all the features in the [SpinMotion Games YouTube channel](#).

**Remember to set these Tags when making changes in some game objects (you only have to check this when changing the car models, otherwise all of these tags are set by default):** In all of the player 1 cars set the **PlayerCar** tag and **PlayerCarP2** for player 2 cars. Then set **AICar01**, **AICar02**, and **AICar03** for the AI cars in the scene, you have to put the tag in the parent gameobject of the AICar and in the colliders. In the CamParent game objects put the **CameraParent** tag and **CameraParentSplit** for the split-screen camera parents. Then, put **CarPos** tag in the CarPosition and put the tag **Player** in the colliders of the car), even for Player 2 cars (by default in the project there are 2 colliders: body and bottom). If you are using headlights, tag the light game objects with the tag **Spots**. To make the color import work set the body of the car with **CarBodyP1** tag and **CarBodyP2** for player 2 cars.

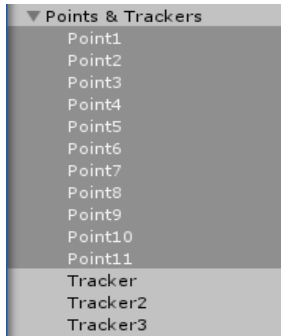
Moving up to the **Map** object, this one has the terrain, the assets that made the track and the environment, a WindZone to simulate wind, and a **Sky Rotate** which has a sky rotate script. You can change the rotate speed of the skybox.



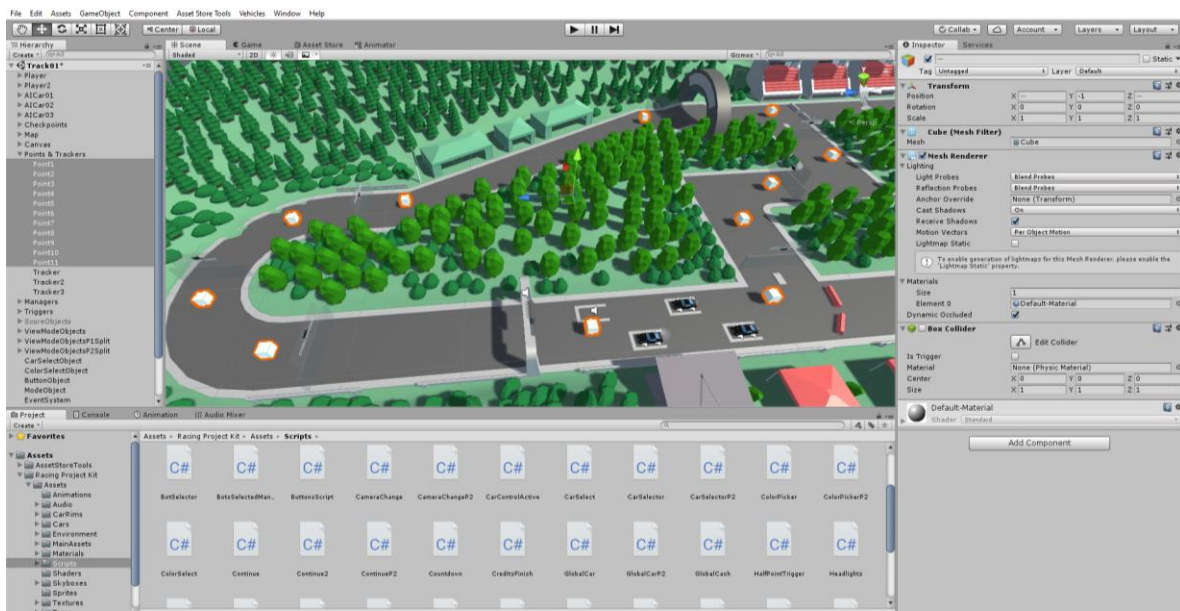
Now we go with the **Canvas**, there's a ton of work made in here. **Left Panel** shows everything located at that side: PosDisplay, Time Target, and some score mode panels & texts. Then **Left Panel Player 2** shows the PosDisplay in split-screen for the second player. The **right Panel** has the min/sec/mili time counter.

**Lap Panel** shows the **lap done** and the **lap requirement**. The **Countdown UI** it's 3, 2, 1, Go animation that it works with a script located in Countdown Manager, that you will see later. **Pause Button** it's the button that you click to pause the game (you can pause with ESC too) and **Pause Menu** it's the mini-menu that appears once you pause that lets you restart or go to the main menu. **Extra Time / Some Score UI** it's used in time and score mode to show some info. **Finish Panels** are the ones used when you finish the race (if you finish 1<sup>st</sup>, 2<sup>nd</sup>, if your time is out in time or score mode or if you don't reach the score requested in score mode and if you finish 1<sup>st</sup> or 2<sup>nd</sup> in split-screen, etc.).

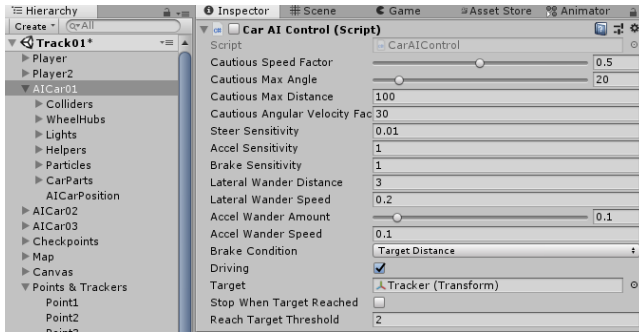
All of this Canvas objects are already assigned to be shown correctly, you can check the game objects in the inspector, how they have these UI game objects attached, now moving up to the next game objects:



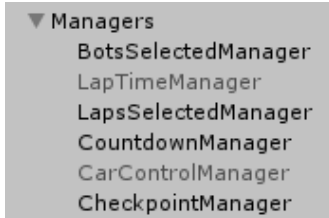
**Points & Tracker** are used only by the AI Cars to follow the course of the track. The **Points** are game objects used to build the route that the AI Cars will follow, and the **Tracker** it's the active point of where's the AI Car going. If you select the **Points** and turn on the Mesh Renderer you will see the route waypoints, you can move the cubes or add new ones to create the waypoint route, pictures below:



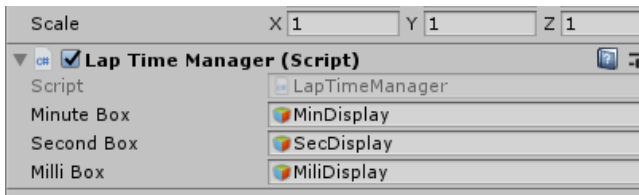
You can put the mesh renderer in the **Tracker** and test the game and see if the AI Car it's following the desired route. You can add all the points that you need, the script (**AI CarTrack**) will detect automatically the first and the last one and move the tracker sequentially. **Important:** the points name must be **Point** and followed, the number (for example: 19). So the complete name must be something like *Point27* or *Point4*. In this way the script will detect them automatically, don't use 01, 02, use directly the number: 1, 2.



There are 3 trackers, each one with an **AI Car Track** script inside, used for each one of the AI Cars. Also, remember to set the Tracker as **target** in the AI Waypoint script (do this if you're creating a new scene, by default is already set).

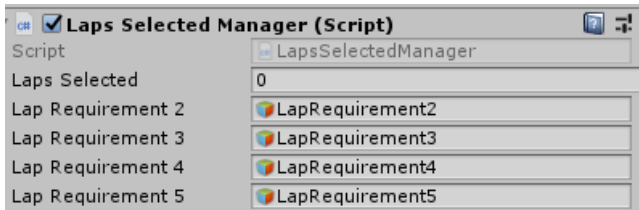


**Managers** are game objects to complete some functions to make the race work. These are 5 in total: 1- Lap Time 2- Laps Selected 3- Countdown 4- Car Control 5- Car Control P2 6- Bots Selected Manager.



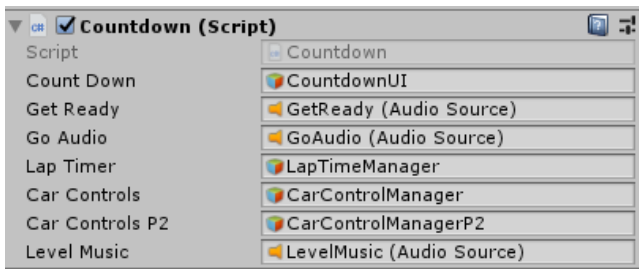
#### N°1 - Lap Time:

This manager has a script that makes the race time work, and it has the UI components to show it.



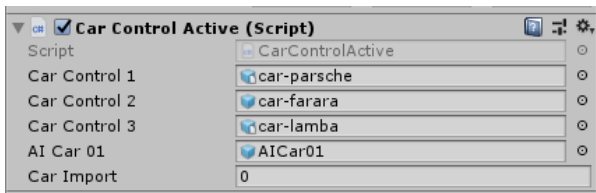
#### N°2 - Laps Selected:

It has a Script that takes how many laps you selected and shows the correct lap requirement in the UI.



#### N°3 - Countdown:

Shows the 3, 2, 1 animation, plays the FX sounds, doesn't let you move the cars before the Go and after the Go plays the level music.

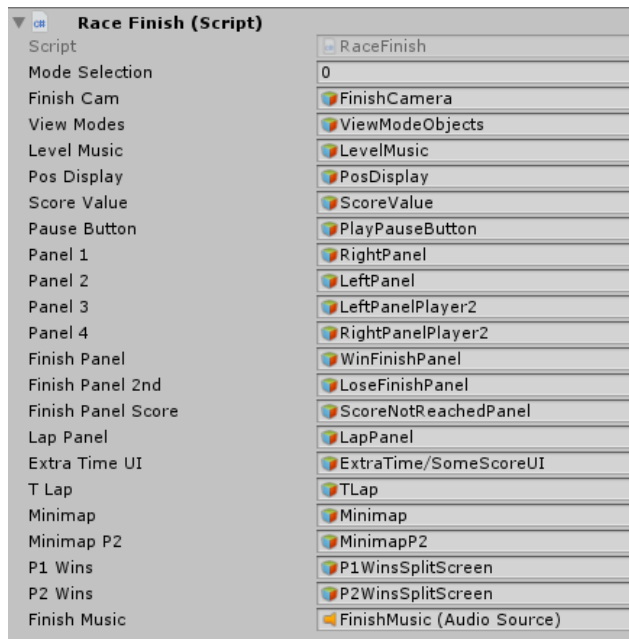


#### N°4 - Car Control:

Once the countdown ends, the car control gets activated and the car can move. N°5 - **Car Control P2** it's for P2.

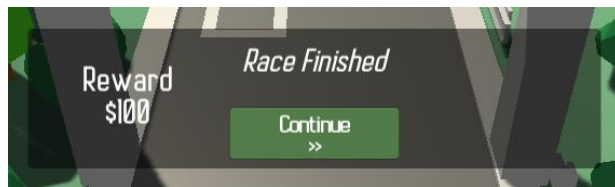
N°6 – **Bots Selected**: It has a script that detects how many AI bots you selected in the play menu and it activates or deactivates the minimap arrows and AI cars depending on how many bots the user has selected.

Moving up to the **Race Finish trigger**:



**Race Finish Trigger** gets activated once you finish the required laps and pass the first checkpoint so it finishes the race. The script deactivates the controls, activates a **finish camera**, **view modes** gets deactivated so you can't change the camera, **level music** stops, **finish music** plays, **PosDisplay**, **Score Value**, **Pause Button** and all the **UI panels** get deactivated and the **Finish Panel** activates instead:

**Race Finish Types (activated by RaceFinish script):**



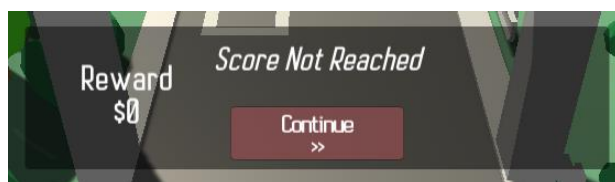
**WinFinishPanel** (all names are from Canvas) this one appears when you complete the objective of the race, it gives you a \$100 reward.



**LoseFinishPanel** appears when you finish in 2<sup>nd</sup> or 3<sup>rd</sup> position in race mode. Reward \$0.



**TimeOutFinishPanel** appears if you don't reach the finish line before the time target requested, appears in time & score mode. Reward \$0.

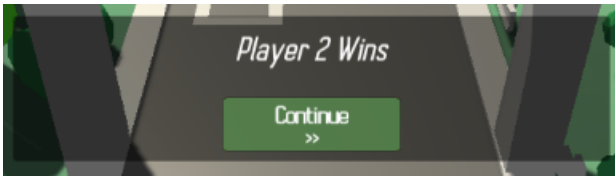


**ScoreNotReachedPanel** appears if you finish the score mode race in time but without the score requested collected. Reward \$0.





**P1WinsSplitScreen** appears if the player 1 finishes the race first in split-screen mode. No reward.



**P2WinsSplitScreen** appears if the player 2 finishes the race first in split-screen mode. No reward.

### Race Reward (given by RaceFinish script):

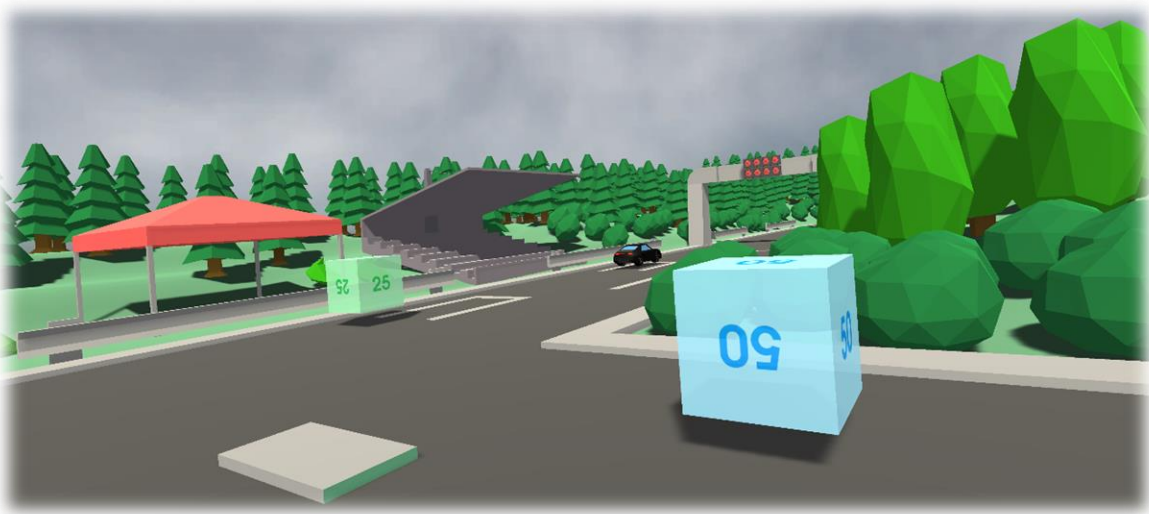
```
//race mode
if (ModeSelection == 0)
{
    //if you win (you finish 1st position)
    if (PosDisplay.GetComponent<Text>().text == "1st Place")
    {
        //you win $100
        GlobalCash.TotalCash += 100;
        PlayerPrefs.SetInt("SavedCash", GlobalCash.TotalCash);
        FinishMusic.Play();
    }
    //if you lose (you finish 2nd position)
    else
    {
        GlobalCash.TotalCash += 0;
    }
}
```

You can change the race reward values and statements in the RaceFinish script, it's all commented and organized, so it isn't a problem, also you can see how are the finish panels activated, etc.

▼ ScoreObjects

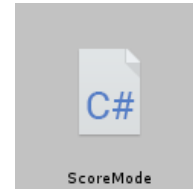
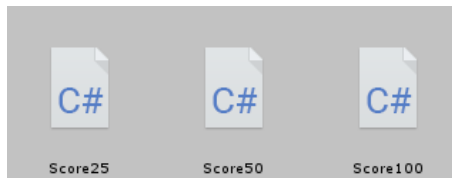
- Score100 1
- Score100 2
- Score100 3
- Score50 1
- Score50 2
- Score50 3
- Score25 1
- Score25 2

Moving up, **Score Objects**, these are only used in the "Score Attack" Race Mode. In this mode the first lap it's for warmup and then once you pass to the second lap, these objects appear. These are cubes with values that you hit and gives the player score points.

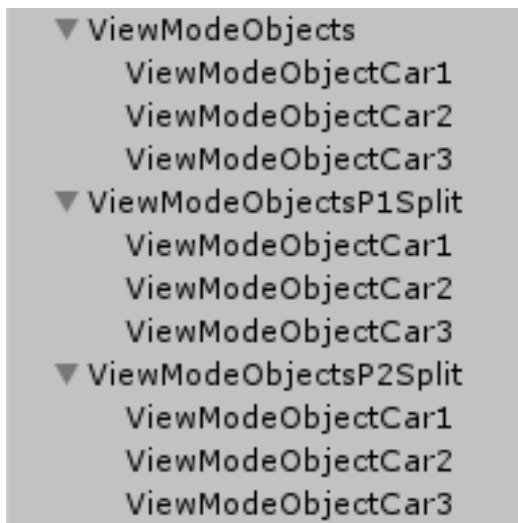




There green score objects (25 points), blue (50 points), and orange (100 points), there are 3 of each score objects in both tracks (short & long).



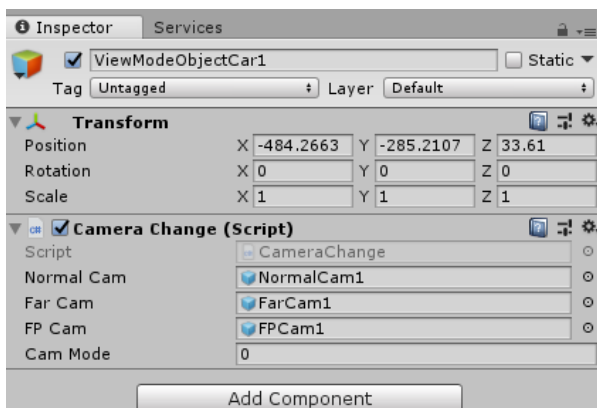
These 4 C# Scripts are the ones that make the whole score mode work. **Score25, 50 & 100** scripts are the ones that tell how much score needs to be added to the UI and **Score Mode** script it's the one that receives that information, adds the score collected to the UI and it also deactivates what it's not part of this race mode. Like **Time Mode & Split Screen Mode** scripts, but we will see that in the Mode Object further.



**View Mode Objects;** these are the ones that make the car camera cycle between 3 cameras with the “v” key or “c” key for player 2.

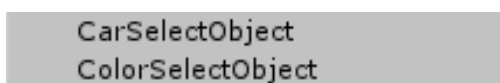
The Split ones are cameras with the viewport changed so you can play split-screen.

You can select the cameras inside the “view mode object car” and move it if you want the camera to film in another angle.



Every View Mode Object Car has a Camera Change script with the cameras attached, these cameras are located in the cam parent of each car in the scene.

You can also change the camera change key in edit > project settings > input > ViewMode / ViewMode P2

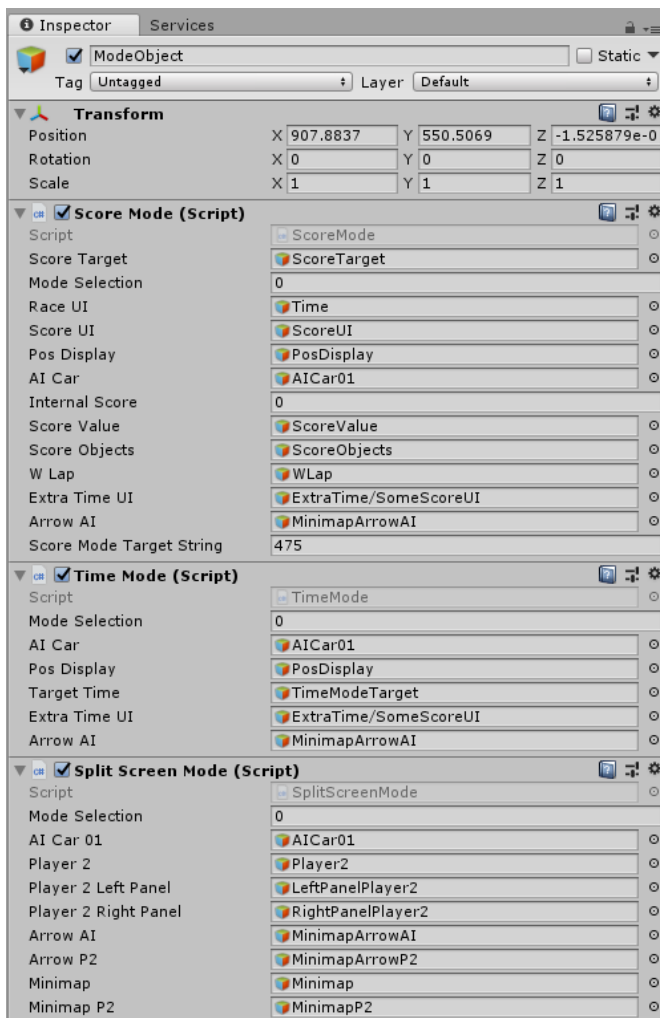


**Car Select & Color Select** objects. These have the car select & color select scripts

attached to it. The functions of these scripts are enabling the car that the player selected in the play menu and changing the color of the body of the car with the one selected in the play menu too. It works for player 2 cars too.

### ButtonObject

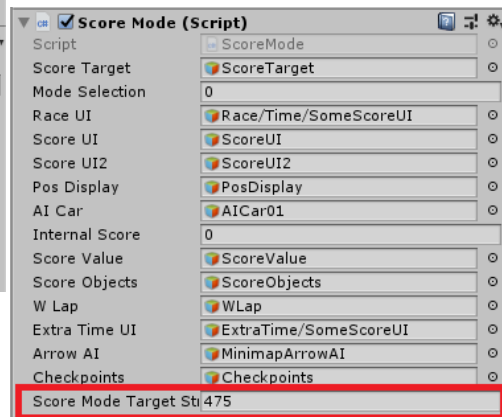
**Button Object** has the button script attached to it and it's used for example in the pause menu to go back to the main menu or restart because it can load scenes, like the one used in Play Menu and Main Menu.



**Mode Object** as you can see has 3 scripts attached to it: Score, Time, and Split-Screen Mode. These scripts have attached the game objects that each game mode needs to activate or turn off to make the game work.

By default the objects that are activated are the ones for Race Mode, that's why there's no Race Mode script.

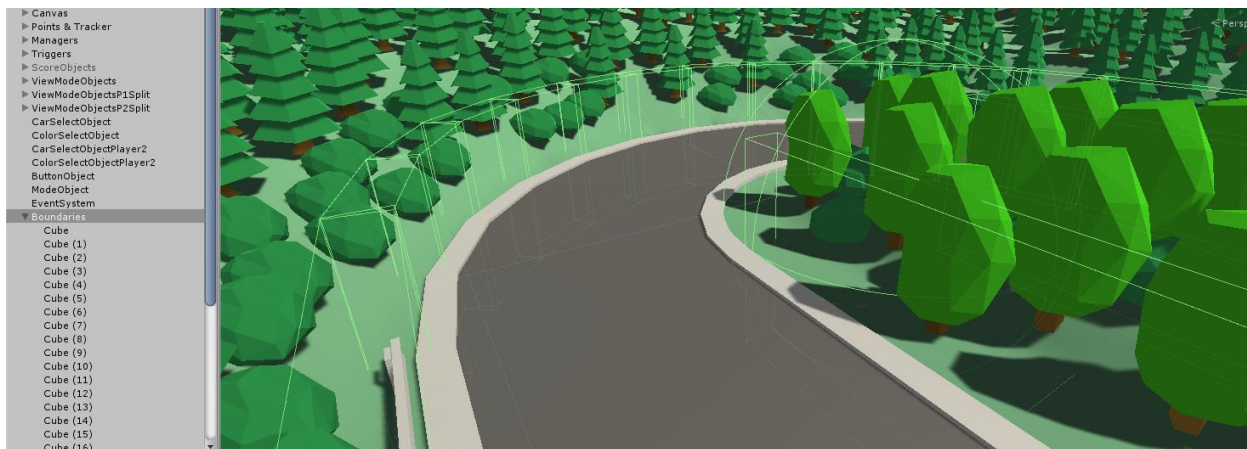
[V1.5.2 Update: Now you can set up the score target from Score Mode script in Unity inspector:](#)



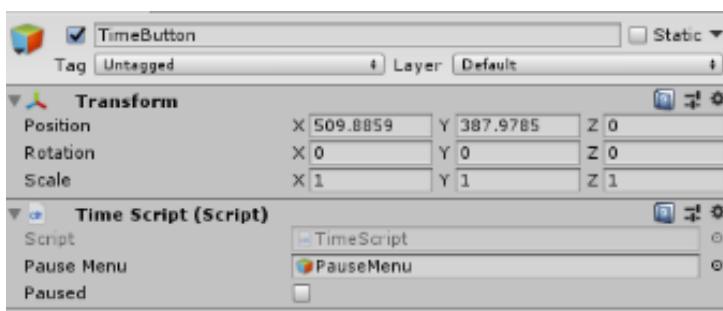
The scripts work if the Mode Selection has its value, 0 = Race Mode, 1 = Score 2 = Time & 3= Split Screen, these values are created in the play menu when you press the race mode button:



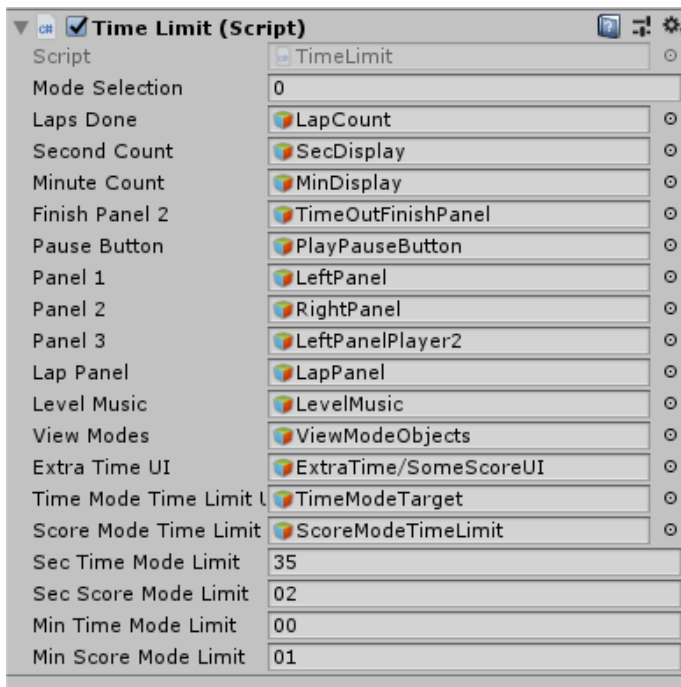
The **Boundaries** are cubes and spheres with Box Colliders that work as invisible walls to not get out of the track, they have the mesh renderer off to be invisible and are made of basics shapes (cube, sphere) to use fewer resources.



Once you finish the race the **Finish Camera** gets activated (that's why it's deactivated by default and you should leave it like that). This camera can be moved obviously and it was made to show how the car crosses the race finish.



The **time button** it's used to play and pause the game and show the pause menu. Also, it changes the audio listener volume when it paused and when it's not.



Time Limit in Track02 Example.

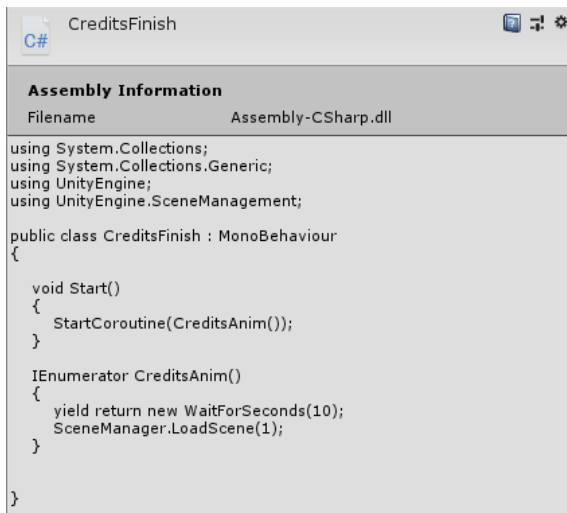
Set the time limit in *score mode* with the variables **Min Score Mode Limit** (for the minutes) and **Sec Score Mode Limit** (for seconds). The same goes for *time mode*. **Remember always to put a two-digit number. For example, if you want the limit to be 1 minute and 2 seconds (1:02) put 01 in minutes and 02 in seconds.**

Moving up to the next scene: [Credits](#)



This scene uses a camera to render the scene, a **Canvas** to show the text and the **back button**, a **Finish Credits** object with a script, and the **Button Object**.

In the **Canvas**, you have the black raw image as a background, the back button at the top left corner, and the text.



In the **Finish Credits** game object, you have the Credits Finish script.

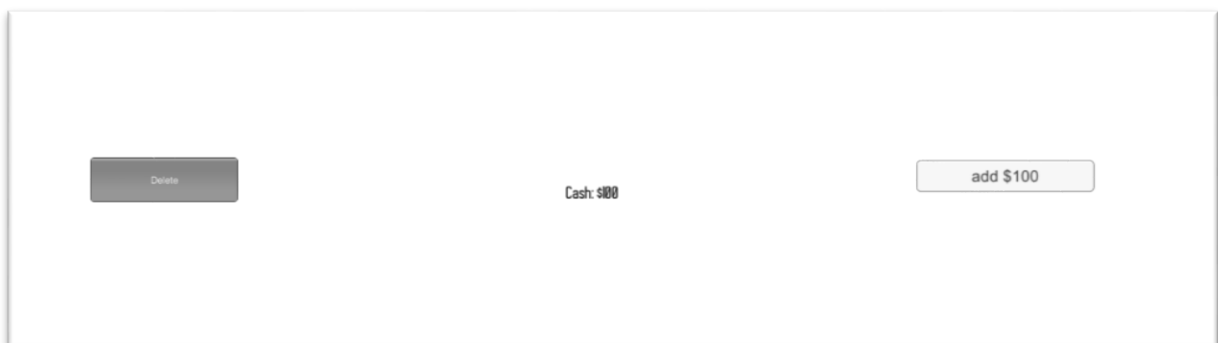
This script shows the Credits Animation.

In this script you can change the length of the credits, it's easy to change, follow the comments in the script.

Once the length of the credits is done, it goes back to the main menu.

The **Button Object** it's used to press the **back button** and go back to the menu.

Now, the last scene [Player Prefs](#):



This scene has a camera to render the scene, a **Canvas** to show the 2 buttons and the text and 2 game objects: **Game Object & Cash Object**.

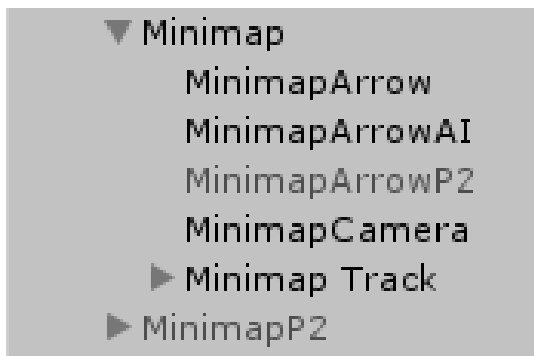
The **Game Object** has 2 scripts: Add 100 & Reset PP. These scripts are used for the 2 buttons in the scene, the Delete button resets the player preferences (locks again the unlockables objects, and puts the cash to \$0), you have to exit the scene with the Play button of unity and enter again to see the change done. Then the Add 100 script it's for the other button that adds \$100 and it shows the changes instantly, you don't have to exit and enter again the scene.

The **Cash Object** shows the cash you have in the text of the scene, that text it's attached as cash display to the script.

## V1.2 Update: Minimap

Inside Racing Project Kit > Assets > Textures, you will find a render texture called Minimap (and MinimapP2 for split-screen mode).

This render texture it's used as a Raw Image in the Canvas child object called “Minimap” and so with the player 2 minimap.



**Inside the Minimap child object of Canvas, we have 5 Game Objects.**

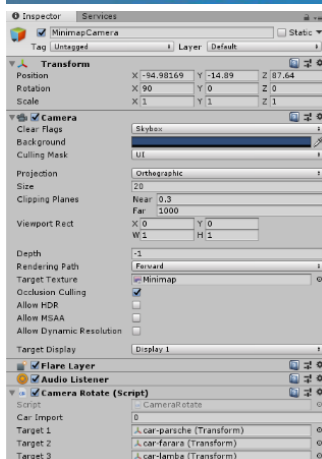
**Minimap Arrow** it's the orange arrow of the player 1 in the minimap, **Minimap Arrow AI**, the blue one for AI rival, **Minimap Arrow P2**, a green arrow for player 2 in split-screen.

Inside each Minimap Arrow (P1, AI & P2) we have 2 scripts: **Arrow Rotate**, the script that makes the Arrow follow the car, and rotate when the car turns and **Camera Rotate**, that makes the camera follow the Car & Arrow in Minimap.

Then, **Minimap Camera** it's the camera above the player 1 and 2 arrows, facing down to the minimap to show it in the Minimap render texture attached to the canvas.



And **Minimap Track** it's a representation of the real track above it. Each piece of track has a 0.0006 Z scale to be 2D and it has a white texture to show the road.



**Note: all objects must have “UI” in layer to be showed.**



Also, **Minimap Camera** has a Culling Mask to only show UI tagged game objects and not the real track. So the camera will only send to the minimap render texture the player arrows & minimap racetrack. This camera also has the **Camera Rotate**. a script where every target



it's the car that the player decides to choose. All of these objects are duplicated in MinimapP2 to work with the player 2 in split-screen. For example, the Camera Rotate Targets in Minimap Camera P2 are the cars that player 2 select.

In **SplitScreenMode.cs** script you will find the code line of the player 1 minimap to go up when you select split-screen mode to show the player 2 minimap.

```
37         //Deactivates the AI blue arrow and activates the P2 (Player 2) green one.  
38         ArrowAI.SetActive(false);  
39         ArrowP2.SetActive(true);  
40         //Moves the Player 1 minimap to the first half of the screen.  
41         Minimap.GetComponent<RectTransform>().anchoredPosition = (new Vector2(120,650));  
42         //Actiivates the minimap for P2  
43         MinimapP2.SetActive(true);  
44     }  
45 }
```



As you can see the Player 2 minimap it's now located where the Player 1 is in other game modes, and Player 1 minimap its located up on the first half of the screen. The position can be set with the Vector2 codes (x position, y position).

Also, you can use anchors to fix other build resolutions position for the minimap.

## V1.4 Update: Real Positioning System, and a lot of new stuff



Welcome to the v1.4 update of Racing Project Kit, there are a lot of new features, and all of them are listed and explained down here.

### **1- Real Positioning System:**

- **Checkpoints.**
- **Debug Menus (F12).**
- **Checkpoint Fades and Rotates.**

2- AI Bot doesn't affect race finish trigger.

3- Lap Requirements simplified.

4- Car select lightning & background changes.

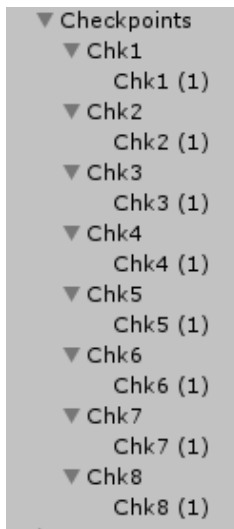
5- Revamped Minimap (Background & UI objects height).

6- Reworked Lightning for the track scenes.

7- Sound FX in all menus.

As you can see the new content is divided into 7 parts so it becomes much easier to explain. So, we're starting up with the *Real Positioning System* in the next page, this is the best and most important feature in Racing Project Kit.

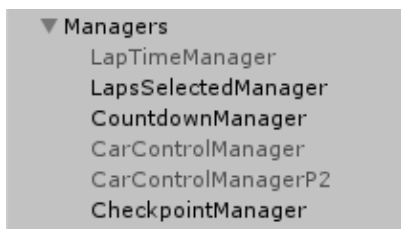
## 1- Real-Time Positioning System



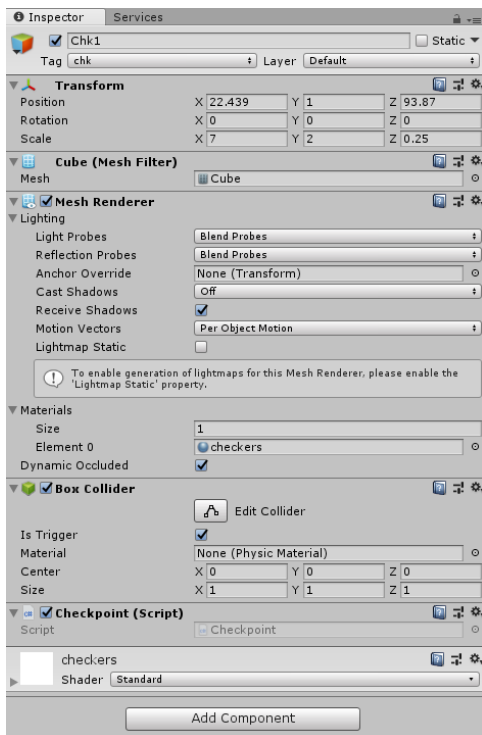
**Checkpoints:** New parent game object which includes Chk & Chk (1).

**Chk:** 8 checkpoints act as a distance meter. Both, the players and the bot, when they cross each one in sequential order obtain a position measure used to make an overall comparison (lap count + check count + distance to the last check) and in this way, we obtain the absolute race position in run time.

**Chk (1):** These are duplicates from the original Chk and are used for some decoration above the checks. From the inspector you can change how much you want the box to rotate.



**Managers:** This game object existed in previous versions but now it has a new child game object inside called Checkpoint Manager. This game object updates the position of the players in real-time and shows it in the Canvas (PosDisplay).



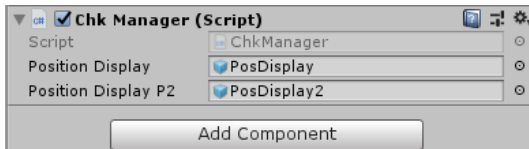
**Chk:** Each one of the Chk objects has a **Box Collider** that requires to have the *Is Trigger* option checked and a tag with the name **chk**.

Each checkpoint has the **Checkpoint** script that determines every player's position in the race.

The Checkpoint scripts will take its transform location and compare it with the player one, to calculate the distance to the last passed checkpoint in real-time.

Only one checkpoint is activated per player at a time. For instance, if you are playing with player 1 and player 2 in split-screen mode when player 1 is between Chk1 and Chk2,

Chk1 will be measuring player 1. If, at the same time player 2 is between Chk2 and Chk3, is Chk2 who will measure player 2. Also, there's a general script called **Checkpoint Manager** that tracks the overall player's race positions, it means that if player 2 is one lap ahead of player 1, it will mark player 2 in the first place, same with the distance.



Checkpoint Manager has a script inside called **Chk Manager** which manages a score count used to make an overall comparison between player 1 and player 2 (each lap passed = 10.000 score + each checkpoint passed = 100 score + distance to the last check = meters from car to checkpoint, real distance. Then, if the player 1 score it's higher than player 2 or vice versa, it updates the position in the Canvas via **PosDisplay** and **PosDisplay 2** game objects. The whole script it's commented so you can understand each calculating process.

Once you complete all the selected laps for the race and pass the first checkpoint, the **RaceFinishTrigger** will turn on. The **LapCount** will be updated when you complete a lap and show it in the game UI.

**Also, to get the positioning system properly working remember to set the correct tags in CarPos game objects.** *CarPos* tag for Player, *CarPosAI* for AI Bot, *CarPosAI2* for AI Bot 2, *CarPosAI3* for AI Bot 3 and *CarPosP2* for Player 2.

## 2- AI Bot behavior with Race Finish Trigger

**Race Finish Trigger** script is activated only by the player 1 or player 2 in split-screen mode. This means that the laps will be counted only if one of the real players crosses the triggers, not the AI Bot.

## 3- Lap Requirement simplified

If you remember back in v1.3 and earlier versions, there was a script called: **LapsSelectedManager**, this script had 4 game objects attached to work with how much laps you selected (*Lap Requirement 2*, *Lap Requirement 3*, etc.) this has been simplified to 1 game object called *Laps Requirement* which leads from how much laps you selected in the Lap Selector.

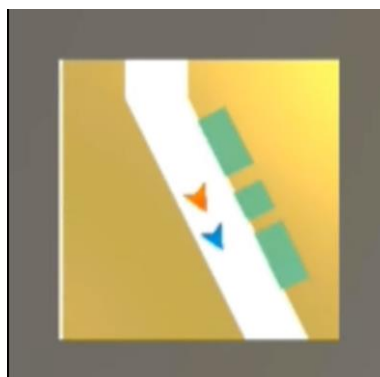
#### 4- Car Select changes



Car selection in v1.4

As you can see the background has another image and the car looks brighter. This is because inside the *Cars Panel* in the *Inspector* there is a second Spot Light instead of one like the previous version had, also, these lights got reallocated to get a better light angle for the cars. You can find these lights in the *Inspector* as **Spot Light** and **Spot Light (1)**.

#### 5- Revamped Minimap



Now it has a different background, new UI sprites placings, and 2 layers, which means that the arrow can go under things that are higher, for example, if the arrow passes under a bridge.

New Minimap in v1.4

vs. Old Minimap in v1.3



## 6- Reworked Lighting

RPK now uses *Progressive* Lightmapping, you can change it back to *Enlighten* if you experience performance problems or you might want to port it to Android in **Window > Lighting > Settings**. It is recommended to use Enlighten for mobile versions of the project (exporting to Android or iOS).



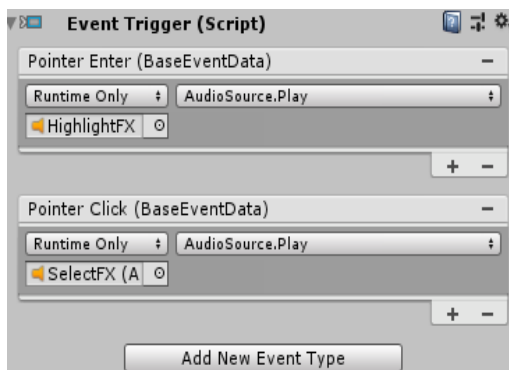
Progressive (new) – left picture and Enlighten (old) – right picture

## 7- Sound FX in all menus



All scenes in the project have an Audio-Files game object in the inspector, which includes 6 sound FX.

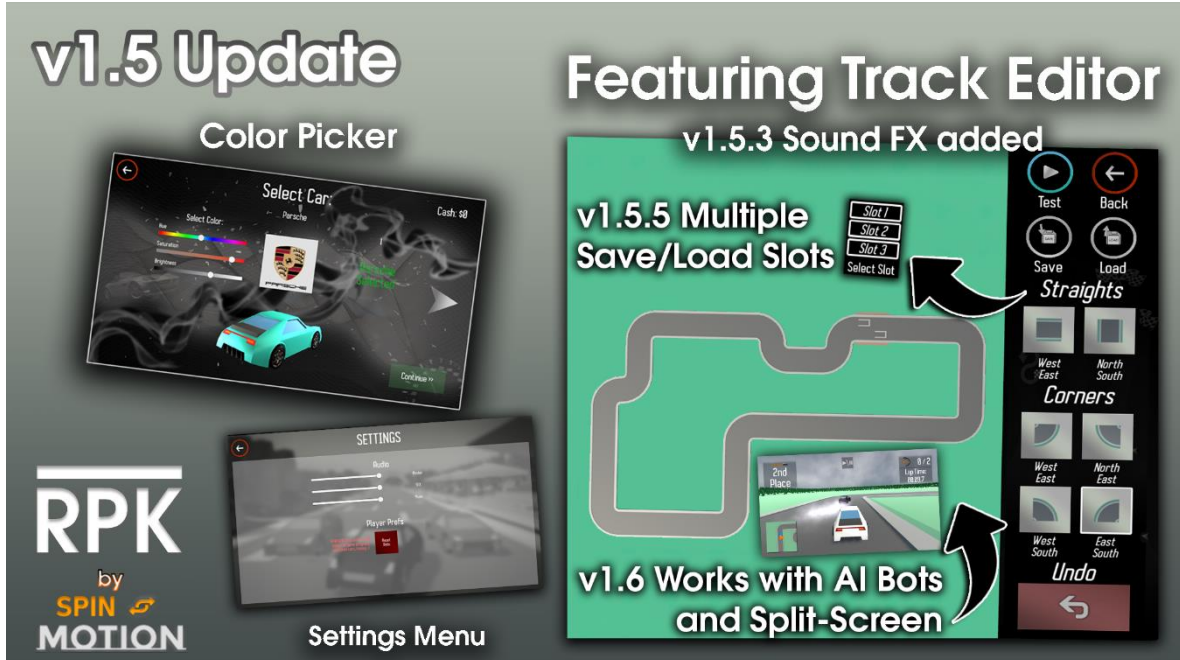
**ContinueFX**, used when you pass from mode to car select, **HighlightFX**, it sounds every time the mouse passes above a button, **SelectFX**, which plays every time you choose something (like a race mode, a car, etc.), **Select2FX**, it's the same as the previous one but it's used for minor things like change car pressing the arrows in the car panel or press the show race mode information button. And the last 2 ones are used only in the car select panel, **PaintFX**, sounds when you select the car's color and **EngineFX**, plays when you select the car to race.



These Audio Files work because each button of all scenes that plays a sound has an Event Trigger, in *Pointer Enter* it's when the button gets highlighted so it plays the **HighlightFX** and *Pointer Click* depends on each button. The one you're seeing in the image it's a race mode one so it plays the **SelectFX**, another case would be the car paint button where it plays the **PaintFX** in *Pointer Click*.



## V1.5 Update: Track Editor, and a bunch of new additions.



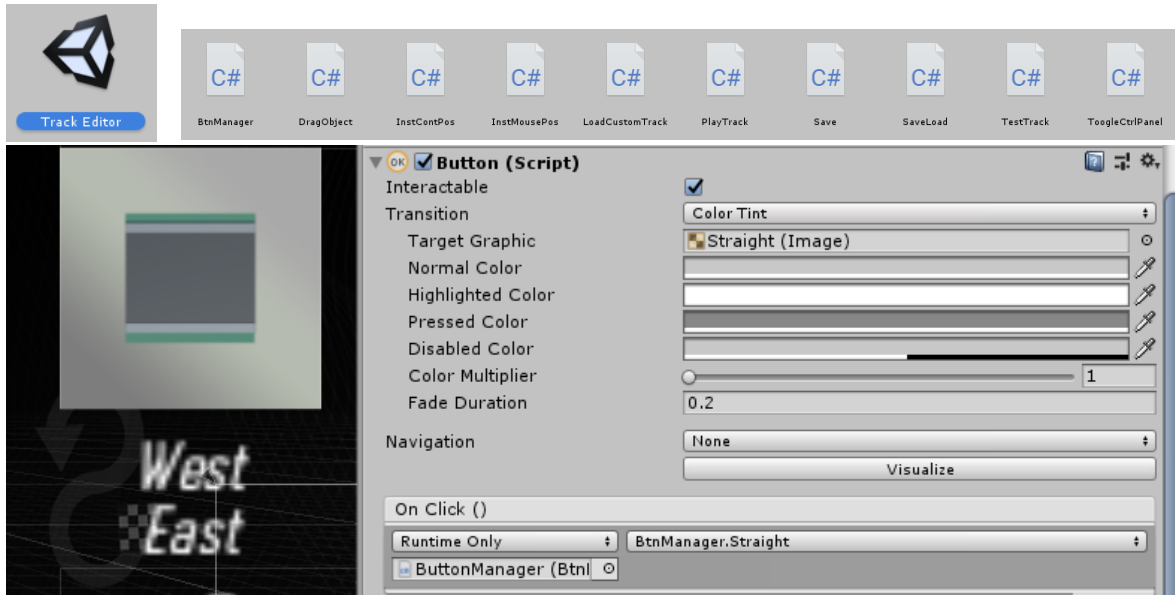
Say hello to the v1.5 update of Racing Project Kit.

- 1- Track Editor.
- 2- Color Picker.
- 3- Fast Forward.
- 4- Settings Menu.
- 5- Headlights System.

As you can see the new content is divided into 5 parts so it becomes much easier to explain. So, we're starting up with the *Track Editor* in the next page.

## 1- Track Editor

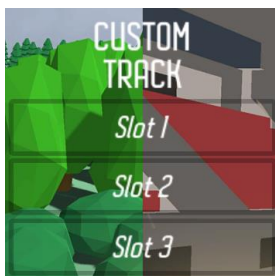
The *Track Editor* features new c# scripts located in the **TrackEditor Scripts** folder and a new scene named **Track Editor** located in the **Scenes** folder.



**BtnManager** script has each of the functions that the Canvas in the Track Editor scene uses. For example, each track part button, save, or load button. For example, the Straight track part uses the Straight void of **BtnManager**.

**DragObject** script manages the mouse click behavior (mouse click, mouse stops clicking and dragging track parts with the mouse). Also, it detects if you click undo or delete button and destroys the requested object.

**InstMousePos** script has the function to instantiate the selected decoration object with the right-click button of the mouse and it applies an algorithm to align it to the terrain grid. Also, for each object that you spawn or delete, it gets serialized and saved to load it later with the load button. The same goes for **InstContPos** but this one creates track parts and next to the last one placed.



**LoadCustomTrack** It's used in the custom track slots to play the track editor tracks selecting it from the play menu.

**PlayTrack** script its used when you play the track created from the play menu, selecting game mode, car, and color. It starts the camera animation, changes the UI, and spawns the selected cars and AI bots to drive it in the created track.

**Save** script it's used when you hit the save button, it serializes in a list all the instantiated objects so you can load it later with the load button. You have 3 slots to save and load your tracks as well. The **SaveLoad** script it's the one called from the UI button to save or load the created tracks in the 3 slots.

**TestTrack** is the script that activates the test car when you hit the test button. It uses the same camera animation of **PlayTrack** and the UI changes hiding all buttons from the track creation and showing only a back button to return to the non-testing mode so you can keep creating the track.



**ToggleCtrlPanel** toggles on / off with H the controls panel. You can change the activation key in the code.

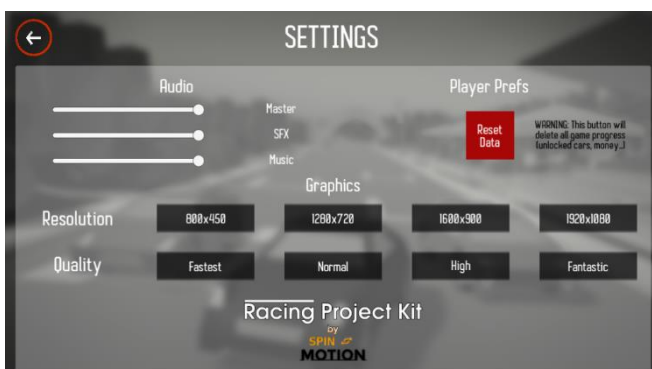
## 2- Color Picker

The car color select system has been replaced with a color picker, so first of all the c# scripts of the old system were removed (*car menu color.cs, car menu color p2.cs, global color.cs, global color p2.cs*) and were replaced by the new ones (*color picker.cs, color picker p2.cs*) and some scripts were modified (*color select.cs, color select p2.cs*).

## 3- Fast Forward

Little debug tool, pressing **Tab** key of the keyboard will set the game to 3x speed, you can change the speed in the **Fast** script. In-game you can go back to the normal speed (1x) by pressing **Esc** to pause and then resume the game.

## 4- Settings Menu



The settings menu functions work because of the **Settings** script. These functions are: changing the volume of the in-race music, the SFX, and the master volume. This works because an audio mix was created and it's used in the script,

you can find it in Assets>Audio>**RPKMix**. Also, the script has a void to reset all the Player Prefs, it is used in the red button that says Reset Data.



Also, in the 1.7 update you can change the resolution and graphics of the game in the settings menu. These settings are locked in the PC version because you can use the window that unity shows when you run the .exe

### 5- Headlights System

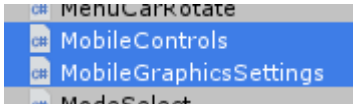
You can press the *H* from the keyboard to turn on/off the car headlights (only for player 1). Change the key bound in **Headlights** script.

Moving up to the mobile support added in v1.7:

### V1.7 Update: Mobile Support



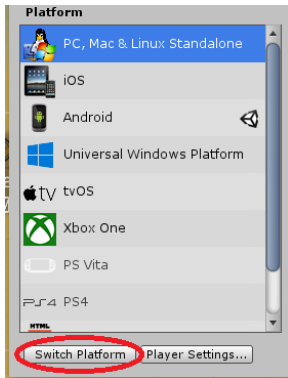
When you open the project in a phone, there are two codes that will start to optimize the project: **Mobile Controls** and **Mobile Graphics Settings**.



These scripts detect when the project it's running in a phone and **Mobile Graphics Settings** will change the screen resolution to 800x450 to improve performance and it will save the resolution and graphics quality selected from the settings menu for the next time that you open the game. And **Mobile Controls** will turn on the UI arrows so you can control the car.

Also, you will find **Mobile Controls** in Play Menu because two buttons are activated locking the split-screen mode and the track editor tracks because those are PC-exclusive features. In main menu you will find **Mobile Controls** too to lock the Track Editor menu. In the future, these features will appear unlocked for mobile.

Remember to download the [Java Development Kit](#) and follow the Unity guides to do a correct export of your application to another operative system.



**Important note:** When you are working for mobile, choose the operating system from build settings and hit **switch platform**, this will load the UI arrows.

If you're working for desktop PC, choose PC, Mac & Linux and hit **switch platform**.

The mobile UI will not load properly if you don't switch to a mobile platform (Android/iOS).

---

That's it for Racing Project Kit, if you have any questions, send an email to [spinmotiongames@gmail.com](mailto:spinmotiongames@gmail.com) or DM on Twitter [@SpinMotionGames](#).

You can contact me, the developer, at [@igna\\_338](#) on Twitter

*Please rate the asset in the Unity asset store if you liked it :)*

Asset video link: <https://www.youtube.com/watch?v=x100g8Xeig0>