

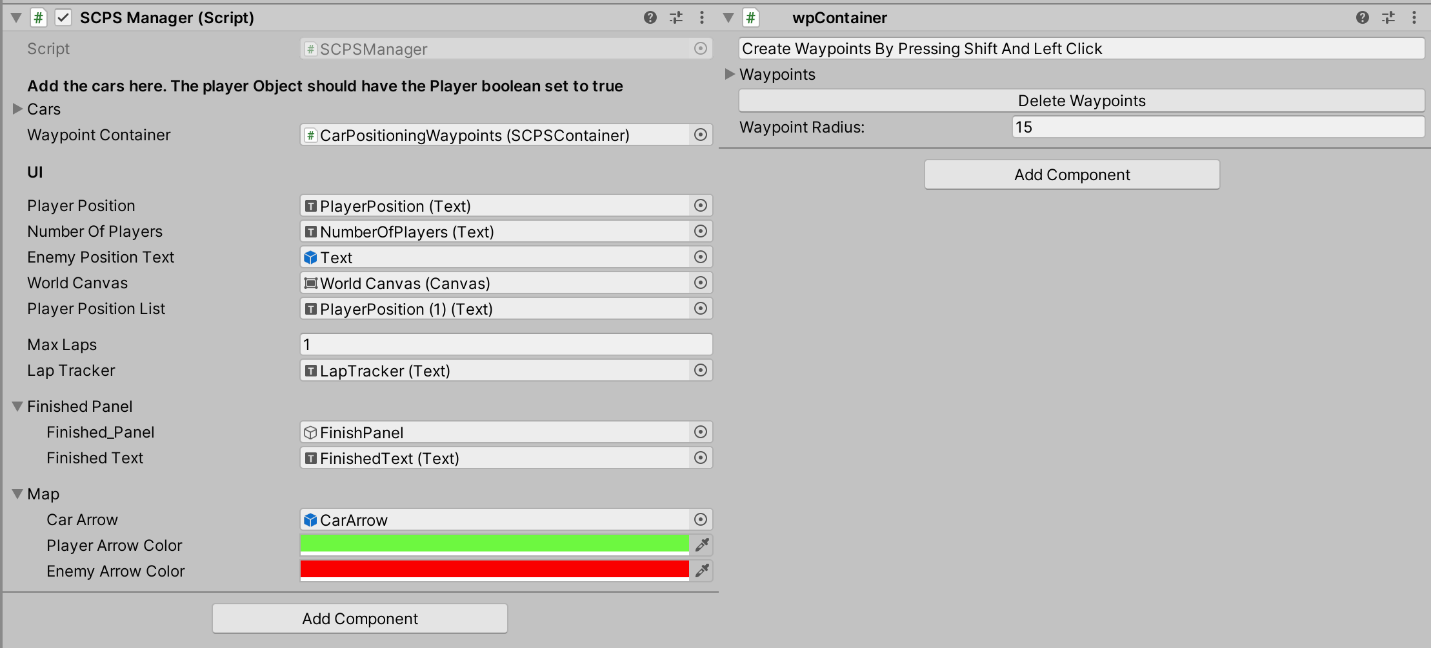
Before starting I would like to thank you for purchasing my asset and I sincerely hope that you find it useful to your project. If you need any help feel free to contact me at: gnanousis@gmail.com

# Table of contents:

* How to use
  + Main Scripts
    - SCPS Manager
    - wpContainer
    - Quick SetUp
  + SCPS In Depth
    - Cars
    - UI
      * UI Texts
      * UI World Canvas and Enemy Position Text
      * Finished Panel
    - Main Menu
    - Finish Line Collider
    - GPS Arrow and Show Next Checkpoint Bool
  + SCPSContainer; how to create a track
* Credits

# How to use

## Main scripts



## SCPS Manager

SCPS Manager is the main script you will be using for the car positioning. It is split in three parts. The cars, a place where you put all the moving objects you want to track, the UI which displays the information and finally the map which handles how the car arrows show in the minimap.

SCPSContainer

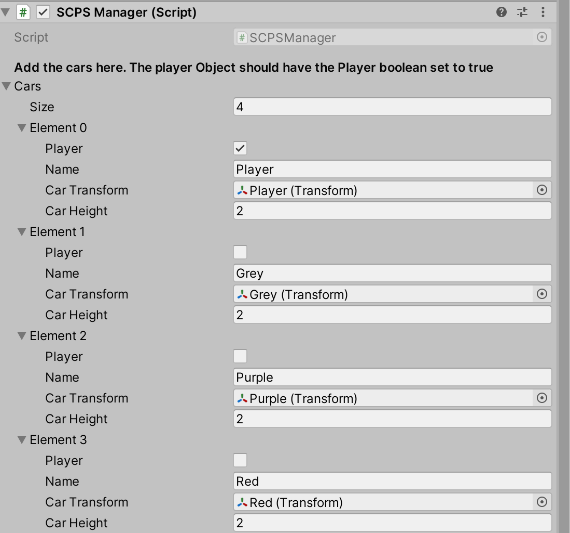
The SCPSContainer contains all the waypoints for the SCPS to track and calculate the positions of each car. It is very easy to use since you can just draw the waypoints by just pressing shift and clicking on the map in order to draw out the track with waypoints.

## To quickly set up

To quickly set up SCPS You can drag and drop the SCPS Manager Prefab from the SCPS/Prefabs folder

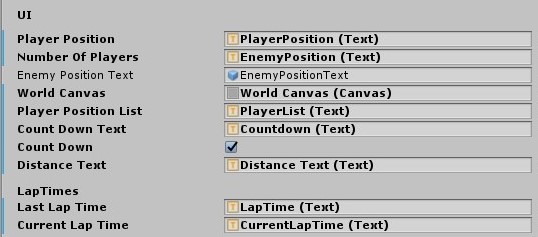
# SCPS in depth

# Cars

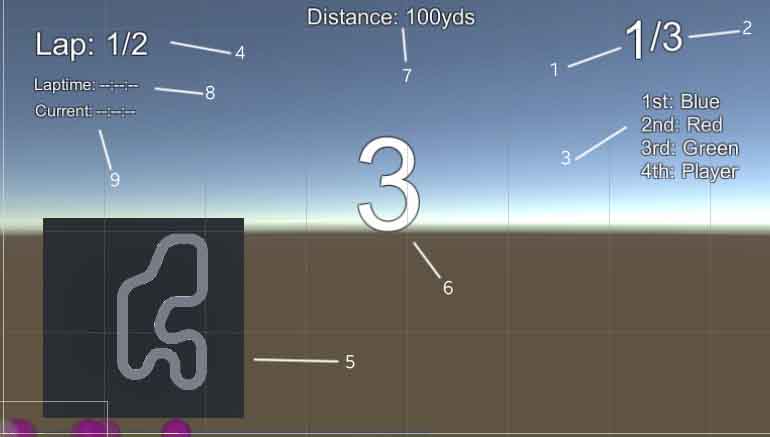
The cars component is a big list that stores the vital information of each car during the race. It also differentiates the player from the ais.

* On the name field you can input the name you want to show up during the race for that given car.
* For the player Boolean you need to enable it for the Player GameObject and only that.
* The Car transform is as it’s name suggest the actual transform of the car you want to handle.
* Finally, the Car Height component is how high you want the position text to appear

# UI



The UI displays the information from the script to the user using the Unity UI with canvas.



1. Player Position: The player position text is the ui text that shows what is the current position of the player. For example the player is 2nd of the 5 players
2. Number Of Players: The number of players text is the text that shows how many cars are in the race. From this example there are a total of 5 cars including the player.
3. Player Position List: This is the list that shows all the car’s position. It shows them from the 1st car’s name to the last
4. Lap Tracker: The lap tracker text shows the player on which lap he is out of how many laps there are on this race. For this example the player is on Lap 2 of 3 laps.
5. MiniMap: The minimap is a raw image that displays whatever the map camera is seeing. We will get more in depth in the end of this documentation.
6. CountDown\*: To enable this simply enable the bool Count Down and put a big text in the middle which will count down the start of the race
7. Distance\*: To enable this simply add a text that will show the distance of the player car to the next checkpoint
8. Last Lap Time Text\*: This will show the last recorded lap time of the player car. Useful to keep track of what was your time and if you can improve on your performance or not
9. Current Time\*: Simply shows the current lap time since the start of the lap

\*Not mandatory in order for the asset to work. If you don’t want them simply do not use them.

## World Canvas and Enemy Position Text



* The world canvas is simply a canvas that has the render mode set to “World Space” and is used to render the text on top of the enemies to inform the players who the car in front of them is, it’s position and it’s name
* The Enemy Position Text is the actual text prefab that spawns on top of the enemy and can be edited by going to SCPS/Perfabs/EnemyPositionText prefab or by creating a new text component in a world space canvas.

These two need to be inserted in the script in order for the enemy text to be seen while playing.

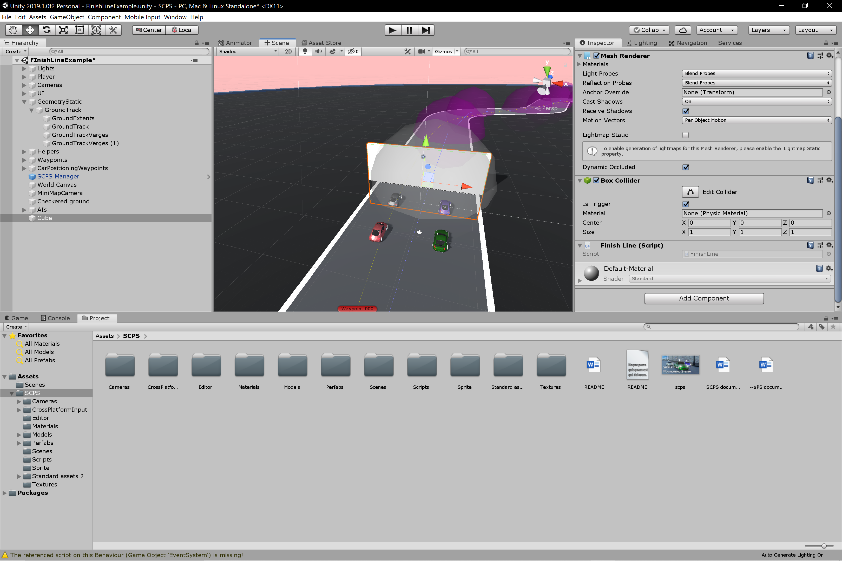
## Finished Panel

### 

The finished panel is the panel that shows up when the player has finished the race and displays the position of the player when he finished.

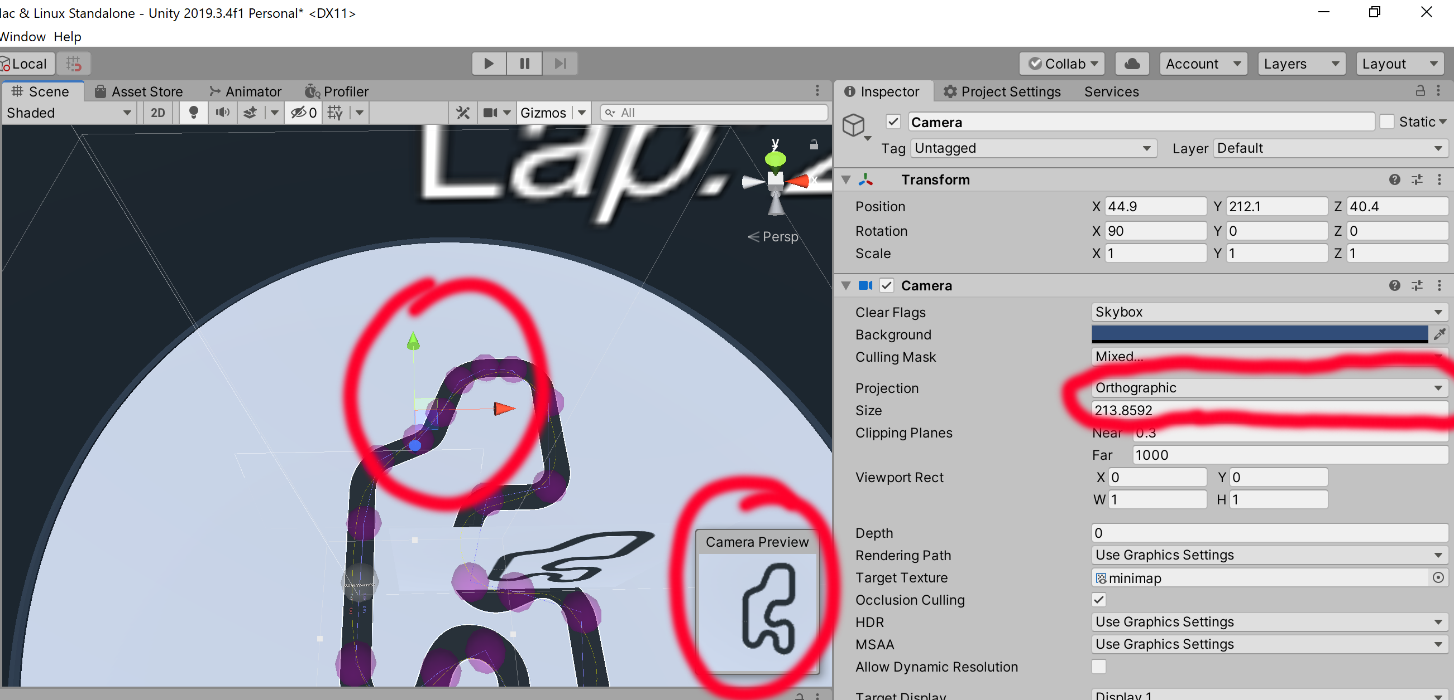
The Finished Panel is the panel inside your UI canvas that shows enables when the user finishes and the Finished Text is the text that displays the finishing position of the player. For example on the screenshot the player finished 3rd.

# Finish Line



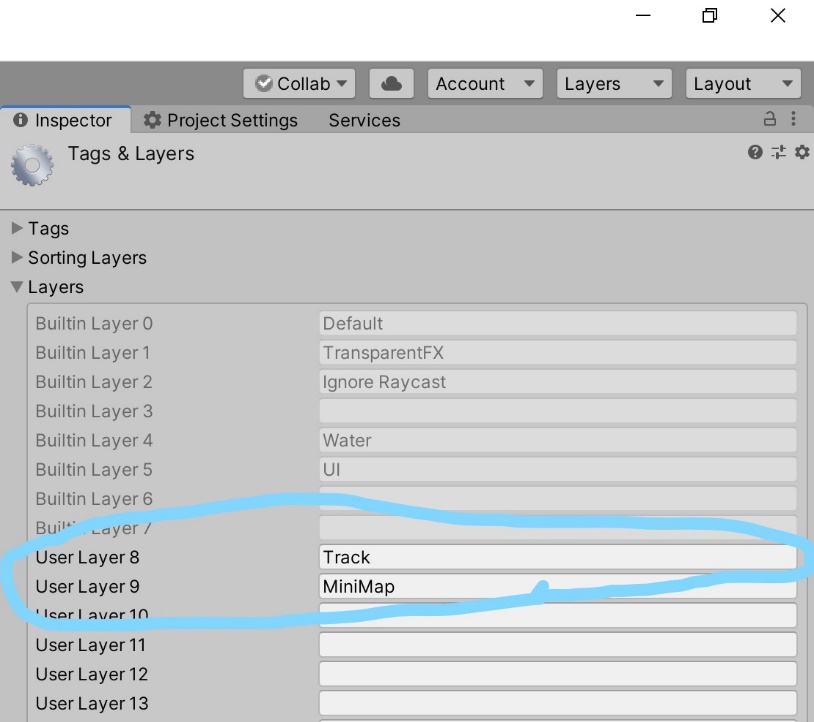
*In order to create a finish line with collision all you have to do is create a new collider to use as a finish line and insert the Finish Line Script. After that just enable the “Uses finish lap collider” on the SCPS Manager script.*

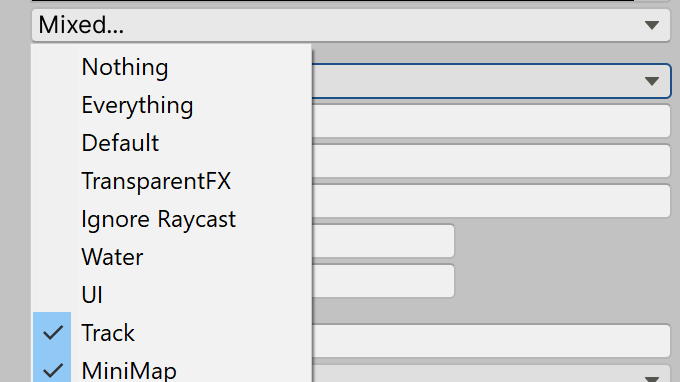
# MiniMap



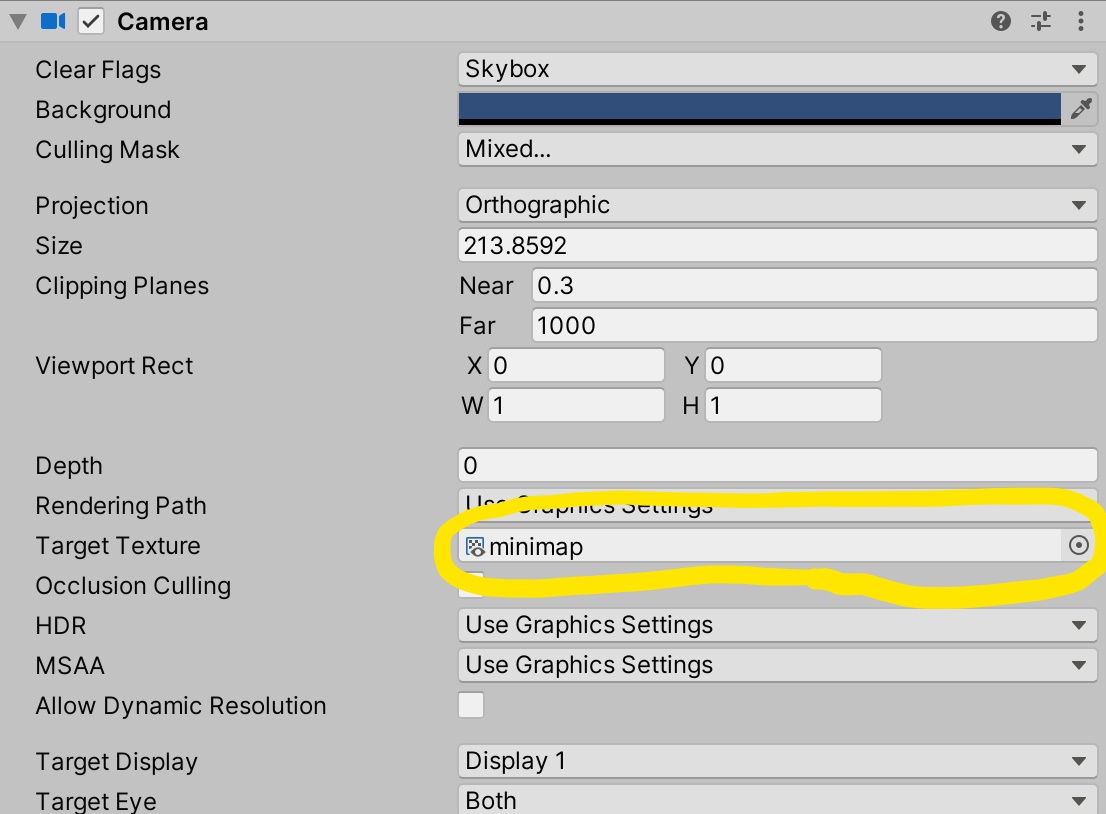
To use a MiniMap you will start by needing to create an orthographic camera and setting it up on top of your map.

Next you will have to set up the layers.

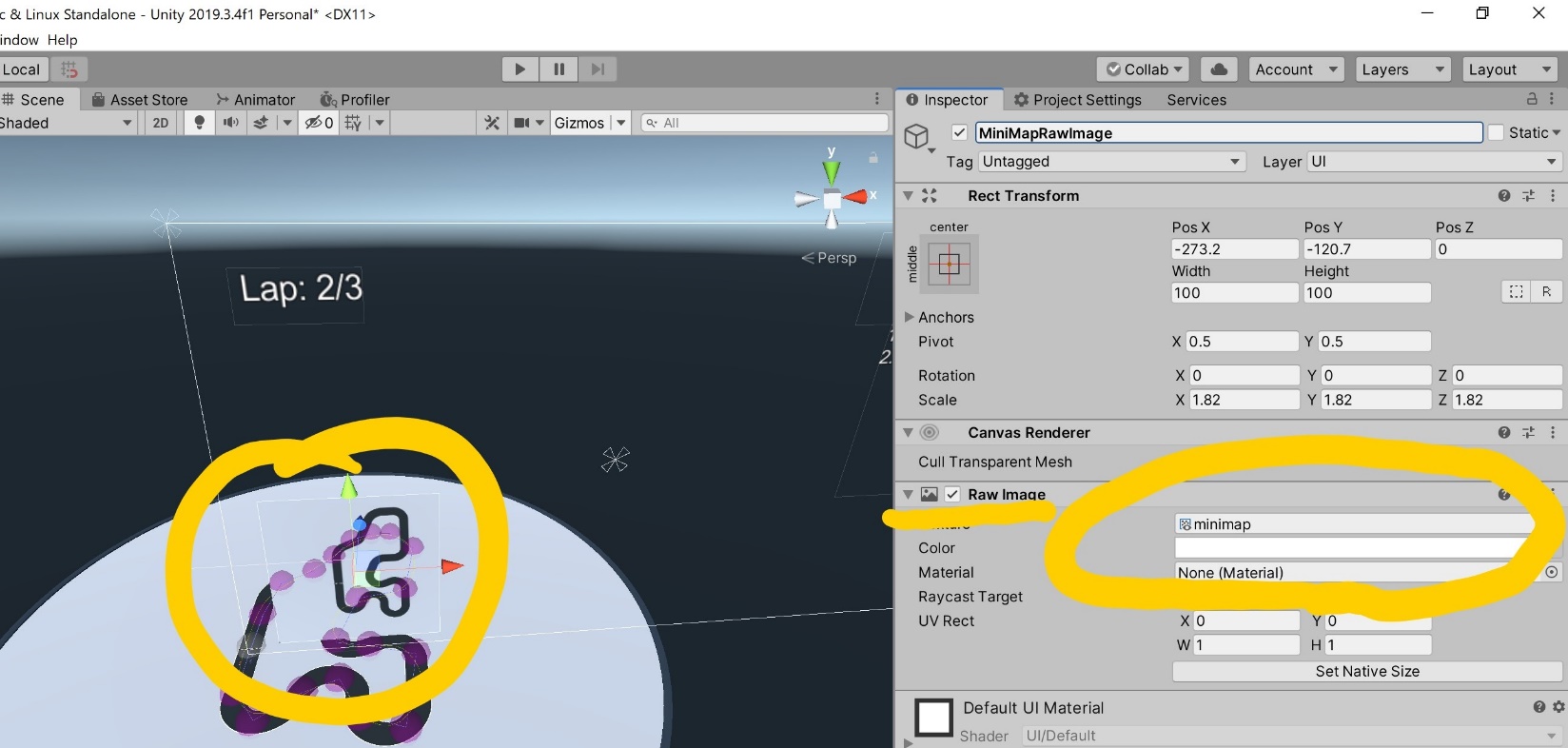
You will need to add these two layers on your project if they are not there already . The Track layer will be for the camera to render the track but also the minimap camera to render it. The MiniMap layer will be only for the Minimap camera to render it. The Car arrows will be on this layers and you don’t want them to be visible.

Next you will need to set those layers as the only layers rendered by your minimap camera. And make sure to disable the MiniMap Layer from your Main Game Camera

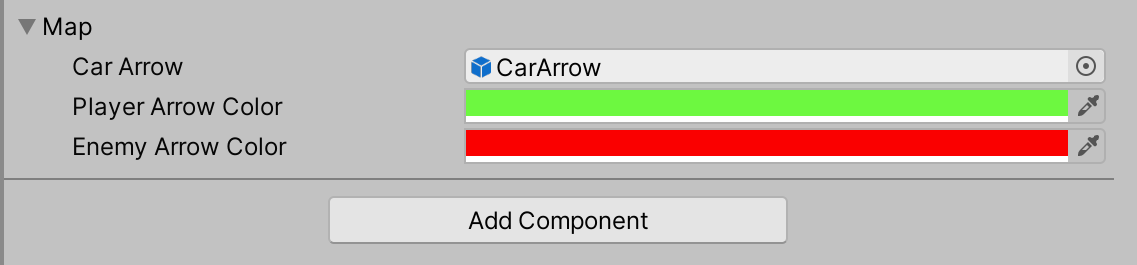
For the next step you need to create a Custom Rendering Texture like shown below and insert it on the MiniMapCamera



Finally create a Raw Image and insert the render texture as it’s texture.



As for the SCPS Manager you need to setup car arrow colors for the Player and the Ais.

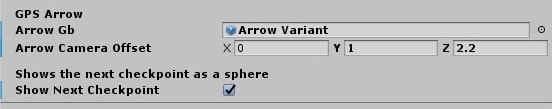


The Car Arrow perfab is the car arrow gameobject that will spawn above the Cars based on if it’s an AI or the Player and can be edited from SCPS/Prefabs folder.

The end result should look something like this

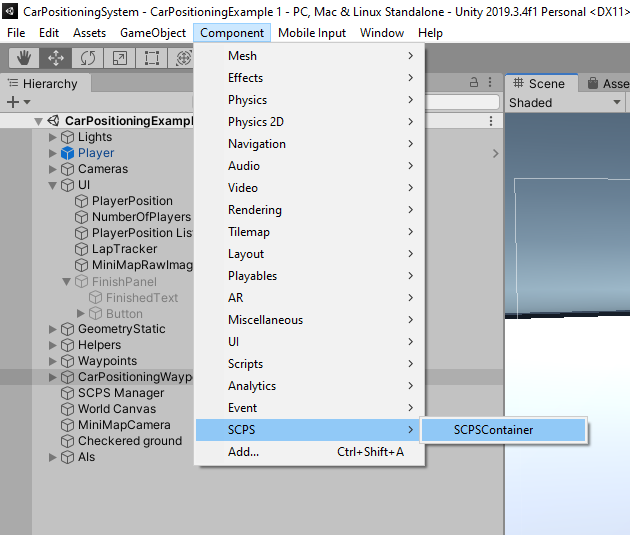
# GPS Arrow and Show Next CheckpointBool

* The GPS arrow is an arrow that points to the direction of the next checkpoint the user has to go to. To enable it simply use the arrow prefab in SCPS>Prefabs and put it inot the Arrow Gb space. To aline it to the camera correctly it will need a bit of trial and error show you will need to play a bit with the Arrow Camera Offset.
* The Show Next Checkpoint bool is a bool that allows the player to see where is the next checkpoint. It simply enables the renderer of the checkpoint so as to make it visible. By default the checkpoints should be a sphere

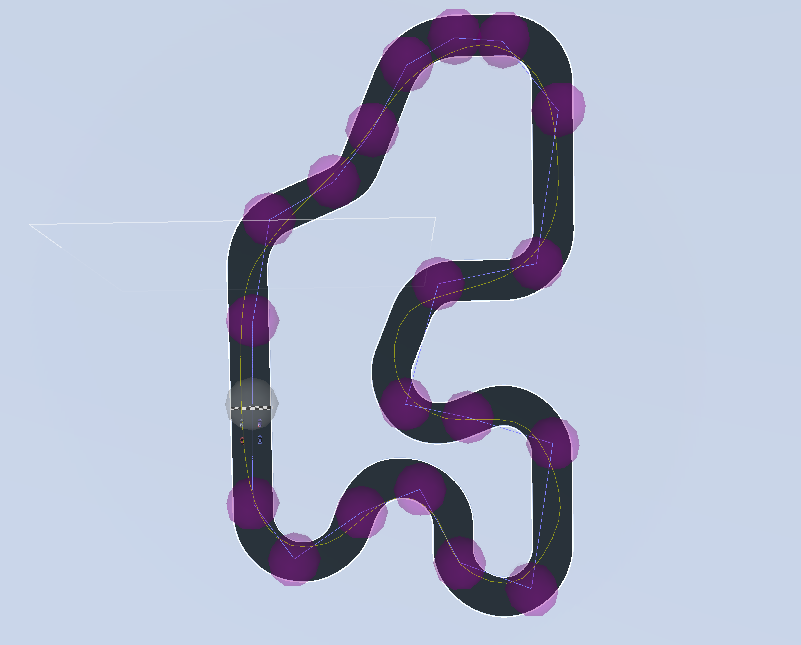


# SCPS Container; how to create a track

First of all create a empty gameObject which will hold the waypoints. Next you will need to add the SCPSContainer script from Component menu > SCPS > SCPSContainer



After that all you need to do is Shift and press the Left Click of your mouse to set up waypoints. Keep in mind that the last waypoint you set will be the waypoint that marks the end of the lap and will be displayed with white instead of purple. A good example of waypoints is this:



The purple waypoints indicate the middle waypoints and the white one indicates the final waypoint and the start of a lap. Keep in mind that the user will have to enter through all the waypoints in order for the lap to count so make it as hard as possible for the player to miss the waypoints

# Credits

Made by Panagiotis Nanousis