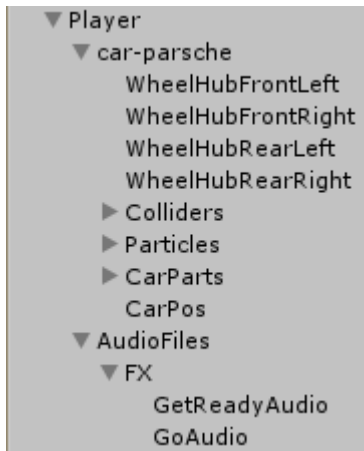


This documentation will get you started with RSK. We will be looking at each game object from the hierarchy. Skip to page 9 for adding more AI cars tutorial.

Custom cars and physics system tutorial video: <https://youtu.be/FkE3sgYxCfI>

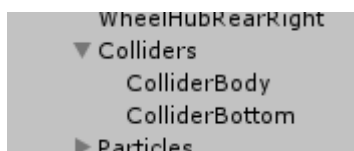


Starting with the first game object, the **Player**. This game object has 2 child objects: **car-parsche** and **AudioFiles**. These FX audio files are used in the countdown at the start of the race.

Inside **car-parsche** you will find all the components to make the player car work (physics scripts, wheel colliders, car parts, and more).

The parent object (**car-parsche**) has the Tag *PlayerCar* and the following components: A rigidbody to indicate the weight of the car in the Mass option. A Car Controller script to configure the car wheels models and colliders, drive type, torque, speed, and other settings. Another script called Car User Control that it's turned off by default because the countdown script enables the script after the 3,2,1 go coroutine, so the player can move the car after those 3 seconds. The last script it's a simple Car Audio.

Inside the **WheelHub** game objects, you will find each wheel collider of the car with a skid Audio Source used when the car loses traction and a Wheel Effects script to make a skid trail on the road. Moving on to the **Colliders**:

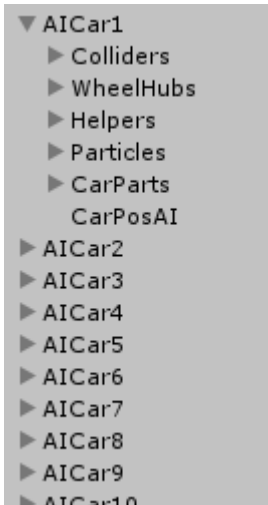


ColliderBody and Bottom have the Tag *Player*. These objects have a box collider to make the player's car have a collision with any other object of the game.

**Particles** game object has a particle system inside. These particles are attached to the Wheel Effects script inside the **WheelHubs** of the car.

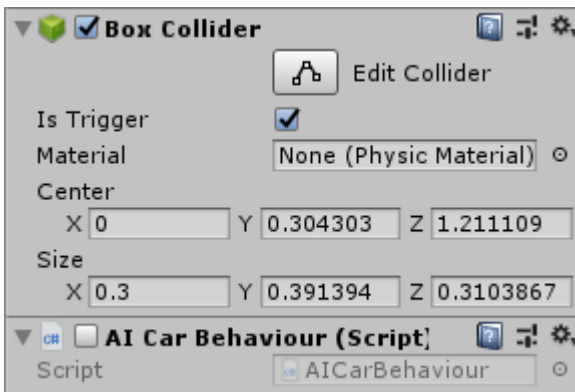
The game object **CarParts** has all the visual parts of the vehicle. Inside Body game object you can change the color of the car for example. And BrakeLights object has a BrakeLight script that changes the color of the object when braking.

The last game object, **CarPos**, it's used for the positioning system (page 8).



Moving up to the next game objects: the **AI Cars**.

The parent object (**AI Car**) has the following components: A rigidbody to indicate the weight of the car in the Mass option. A Car Controller script to configure the car wheels models and colliders, drive type, torque, speed, and other settings. Another script called Car AI Control that it's turned off by default because the countdown script enables the script after the 3,2,1 go coroutine, so the AI cars will move after those 3 seconds. Another script is Car Audio that uses 4 audio clips to recreate an engine.



The next script it's AI Car Behaviour and its function is to detect the AI car speed to see if the car it's stuck so it will start going reverse for 1 second to get back on track. Also, it uses a Box collider with IsTrigger option checked located in the front of the car to detect if an AI or player car triggers, activating the car brakes to avoid crashing it at all speed.

Leave AI Car Behaviour deactivated by default, it is activated automatically in the *CarControlActive* script when the race starts.

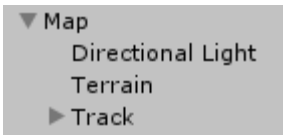
The **Colliders** (ColliderBody and Bottom) have the Tag *AICarCollider*. These objects have a box collider to make the AI cars have a collision with other objects of the game.

Inside the **WheelHubs** game object, you will find each wheel collider of the car with a skid Audio Source used when the car loses traction and a Wheel Effects script to make a skid trail on the road.

**Particles** game object has a particle system inside. These particles are attached to the Wheel Effects script inside the **WheelHubs** of the car.

The game object **CarParts** has all the visual parts of the vehicle.

The last game object, **CarPosAI**, it's used for the positioning system (page 8).



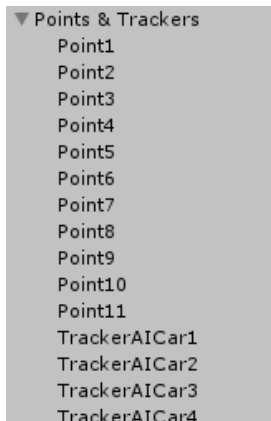
Moving up to the **Map** object, this one has the terrain, the assets that made the track and the environment, and a directional light that it's used as the main source of light.



Moving up to the **Canvas**, you have 2 main objects inside the Canvas: **Laps&BotsPanel** and **RaceUI**.

Inside **Laps&BotsPanel** you will find the objects used at the start of the scene before playing the race. There are 3 main objects: **Laps** which has 2 buttons used to increase the number of laps to race and the other to decrease (LapUp and LapDown). There are also some text labels like the title that says: Select Laps and the current number. Inside the script *LapSelector* you will find the public voids used in the LapDown and LapUp buttons to act on changing the number of laps. You will also find the conversions and variables that are used in scripts like *LapsSelectedManager* that check the number of laps selected by the player and assign them as the **lap requirement** for the race. Feel free to check the scripts I will mention during this documentation, there's a lot of commentaries in it so it's easier to understand. The same method is used for the **Bots** object, and the corresponding script is *BotSelector*. The last game object is **Play** which uses the void of *Continue* script.

**RaceUI** game objects are the **Left Panel** that shows everything located at that side like the **PosDisplay**. The **right Panel** has the min/sec/mili time counter, then the **Lap Panel** that shows the number of laps done and the **lap requirement**. The **Countdown UI** it's a 3, 2, 1, Go animation that works with a script located in Countdown Manager. **PlayPauseButton** it's the button that you click to pause the game (you can pause with ESC key too) and **Pause Menu** it's the mini-menu that appears once you paused so it lets you quit or resume the race. **Finish Panels** are the ones used when you finish the race (if you finish in first or second position).



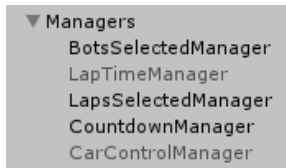
**Points & Trackers** are used only by the AI Cars to follow the course of the track. The **Points** are game objects used to build the route that the AI Cars will follow, and the **Tracker** it's the active point of where's the AI Car going. If you select the **Points** and turn on the Mesh Renderer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



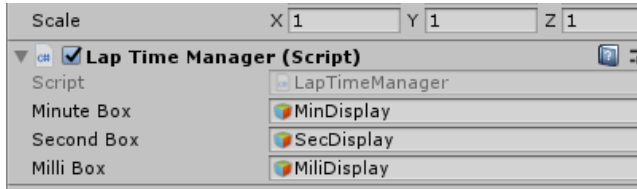
You can put the mesh renderer in the **Tracker** and test the game and see if the AI Car it's following the desired route. You can add all the points that you need, the script (*AICarTrack*) will detect automatically the first and the last one and move the tracker sequentially. **Important:** the point's 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.

The second part of the AI system is the **trackers**. Each one has a box collider with the is trigger option checked and the *AICarTrack* script attached.

Each AI car has its tracker. The tracker names must follow the same naming scheme of the points. The complete name must be something like *TrackerAICar27* or *TrackerAICar4*.

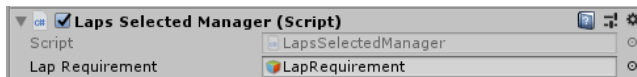


**Managers** are game objects to complete some functions to make the race work. These are 5 in total: 1- **Lap Time** 2- **Laps Selected** 3- **Countdown** 4- **Car Control** 5- **Bots Selected Manager**.



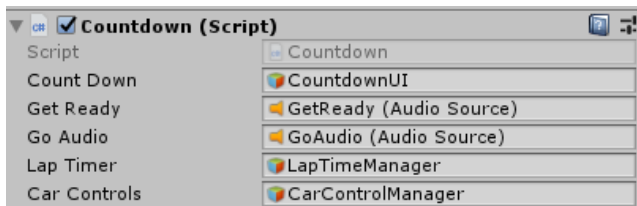
#### N°1 - Lap Time:

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 UI text, plays the FX sounds, and doesn't let you move the cars before the Go.



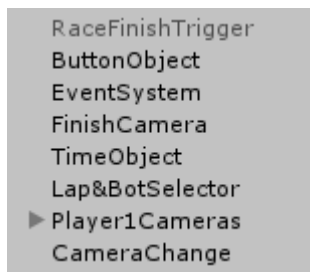
#### N°4 - Car Control:

Once the countdown ends, the car controls get activated and the cars can move (player car and AI cars).



#### N°5 – Bots Selected:

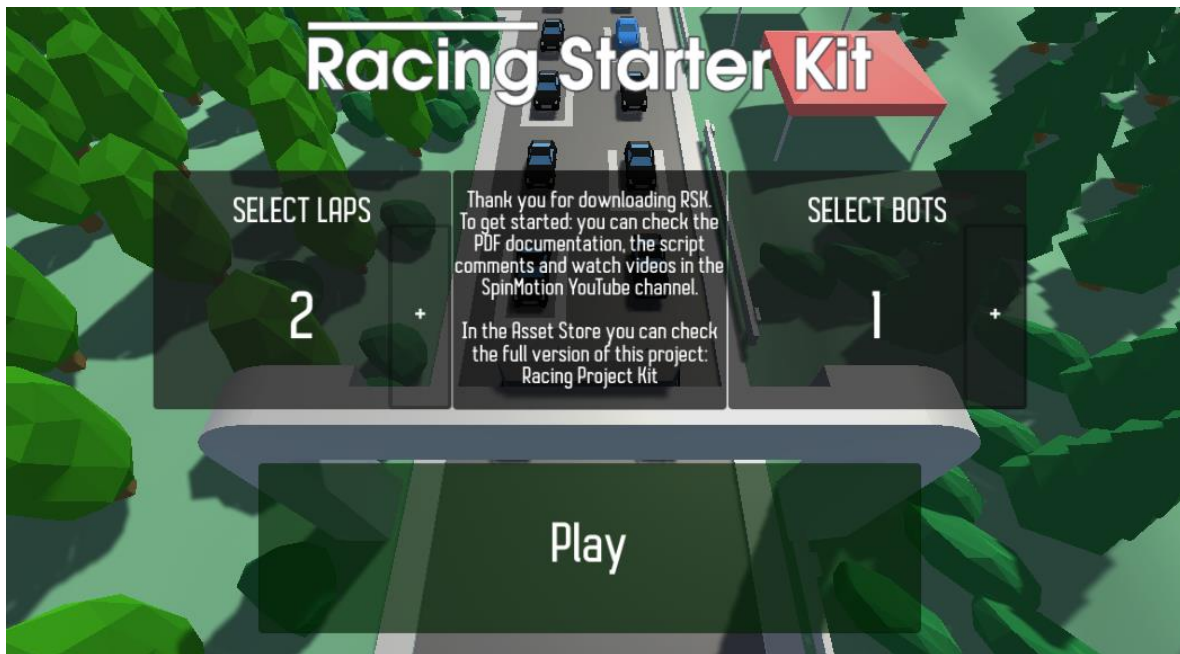
It has a script that detects how many AI bots you selected and it activates or deactivates the AI cars in the scene.



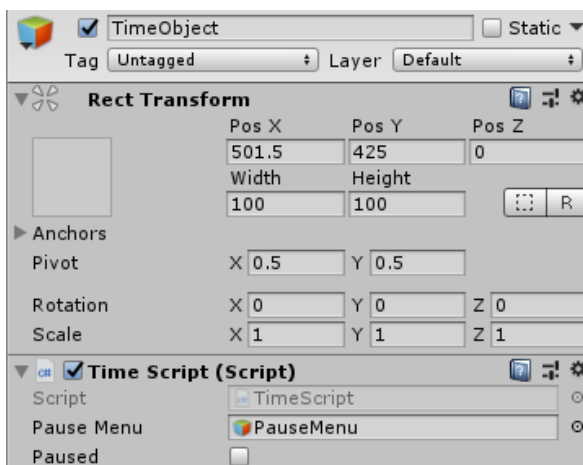
Moving up to the **Race Finish Trigger**, this one gets activated once you finish the required laps and pass the first checkpoint so it finishes the race. The script deactivates the controls and it activates the finish camera and a Finish Panel.

Race Finish Panels (activated by *RaceFinish* script): WinFinishPanel This one appears when you finish 1<sup>st</sup> in the race. LoseFinishPanel Appears when you finish 2<sup>nd</sup>, 3<sup>rd</sup>, or 4<sup>th</sup> position. These panels can be found inside the Canvas > RaceUI > Finish Panels.

**Button Object** has the *Continue* script attached to it and it's used to start the race when pressing the Play button in the starting panel.

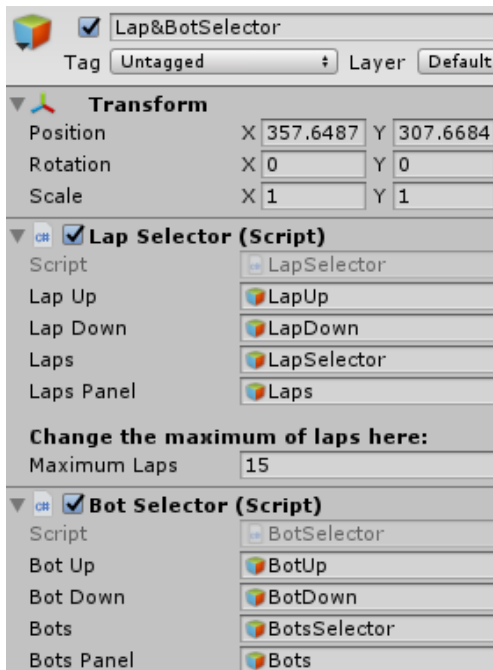


The **Finish Camera** it's used in the starting panel or once you finish a race.

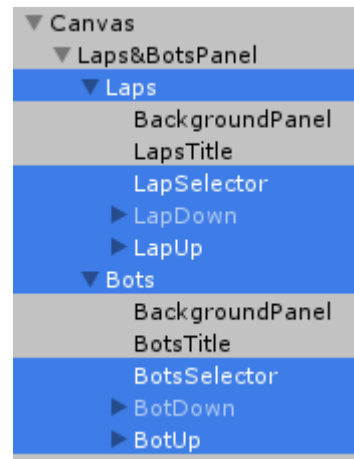


The **TimeObject** it's used to play and pause the game and show the pause menu. Also, it changes the audio listener volume when it's paused and when it's not. Also, a little debug tool, pressing *Tab* key of the keyboard will set the game to 3x speed. You can change the speed in the *TimeScript*. In-game you can go back to the normal speed (1x) by pressing *Esc* to pause and then resume the game (by pressing *esc* again or the resume button).





**Lap&BotSelector** is the gameobject that contains the *LapSelector* and *BotSelector* scripts. These scripts have the UI objects of the laps&bots panel attached.



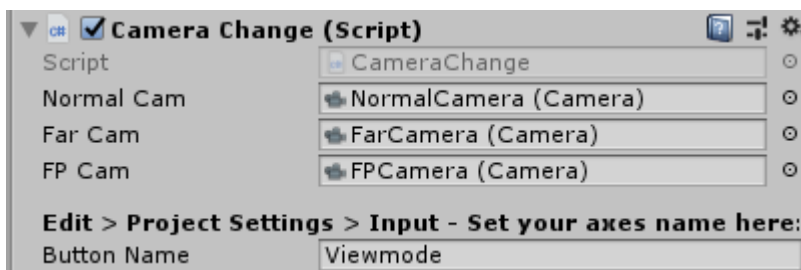
Change the maximum of laps that you can select from the Unity inspector.

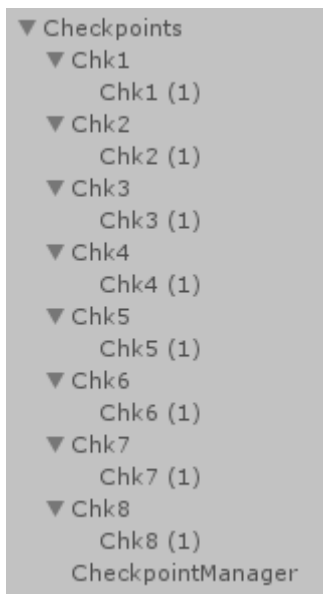
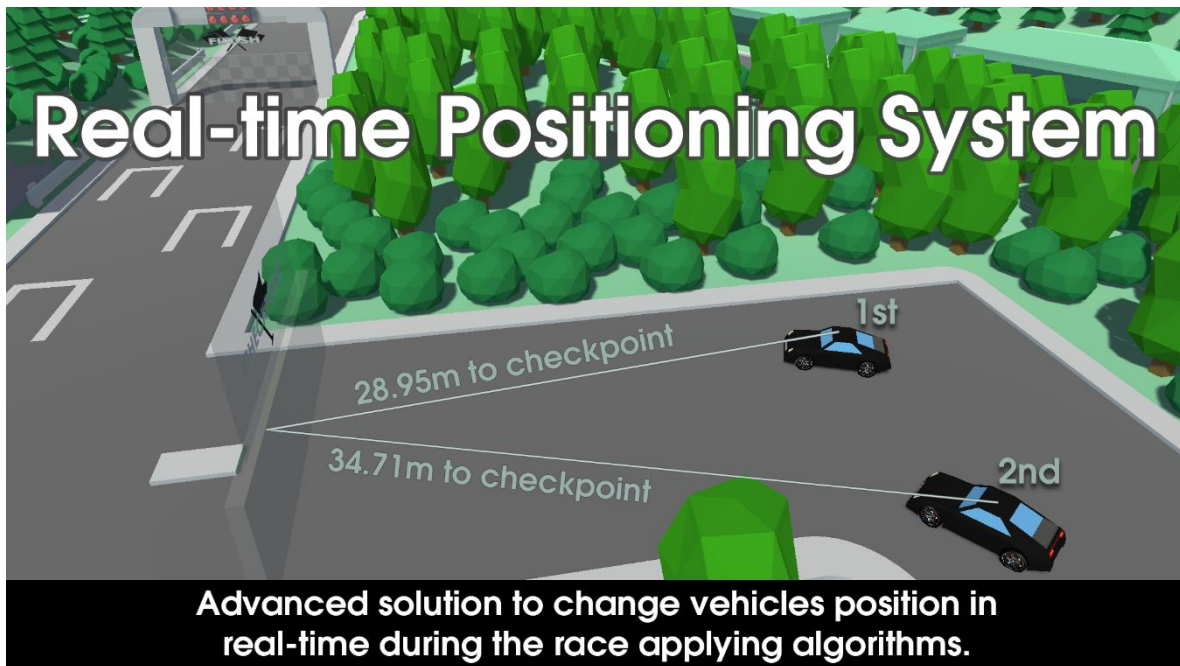
The bot selector will automatically check how many AI bots are in the scene and it will update the maximum of bots that the player can select



**Player1Cameras:** Inside this game object you can find 3 cameras (Normal, Far, and FirstPerson). Each camera has a **CarCamera** script where you can change some position and rotation/following speed values.

**CameraChange:** The car camera can cycle between 3 cameras with the “v” key. Change the activation key in Edit > Project Settings > Input and changing the value key of Viewmode axes. Also, you can assign your own cameras or change the axes' input activation.



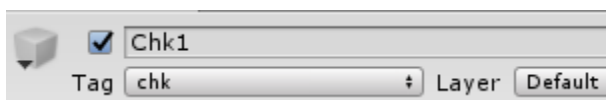


### Checkpoints:

Chk: Checkpoints act as a distance meter. Both, the player and the bots, 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. **You can duplicate a checkpoint object and put the following number to have more checkpoint in your race, the script will detect it automatically! (i.e: Chk9, Chk10)**

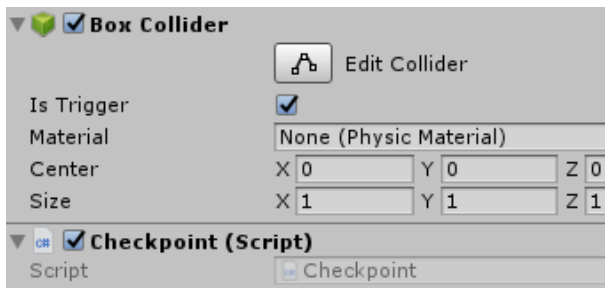
Chk (1): These are child objects of the Chk game objects and are used as decoration above the checks.

The Checkpoint Manager game object updates the position of the cars (player and AI) in real-time and shows it in the Canvas (PosDisplay).



Chk: Each one of the checkpoints objects needs the *chk* Tag.

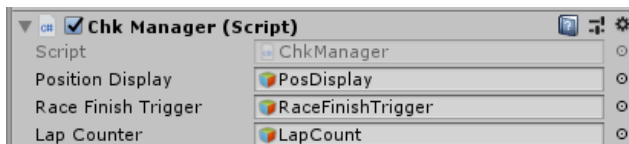




Also, each checkpoint object has a box collider with the **IsTrigger** option checked and the **Checkpoint** script that determines every player's distance from the last passed checkpoint in the race. Also, this

script will take the checkpoint transform location and compare it with the cars in the race, calculating the distance to the last passed checkpoint in real-time.

There's a general script called **Checkpoint Manager** that tracks the overall player and AI bots race positions.

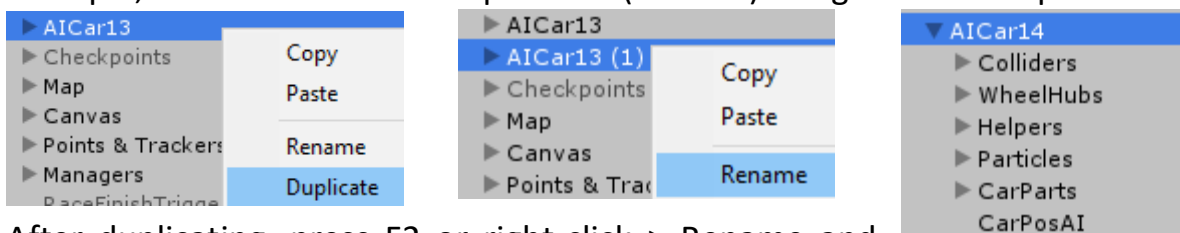


Checkpoint Manager has a script inside called **Chk Manager** which manages a score count used

to make an overall comparison between the player and AI bots (each lap passed = 10.000 score + each checkpoint passed = 100 score + distance from car to last passed checkpoint). Then, if the player score it's higher than the AI bots, it updates the position in the Canvas via the **PosDisplay** game object. 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.

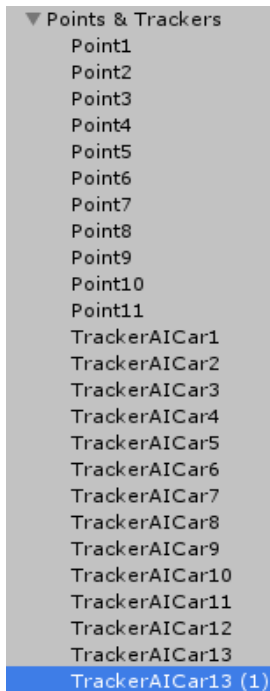
If you want to add more bots, the process it's really simple:

Step 1: Duplicate an existing AICar and change the name to the correct number. By default the project has 13 AI Cars, if you want to add a 14<sup>th</sup> for example, select AICar13 and duplicate it (CTRL+D) or right-click > Duplicate.



After duplicating, press F2 or right-click > Rename and change the number in the name. In this example, it is the 14<sup>th</sup> AI car so the name has to be: AICar14.

In the case of one-digit number AI cars put AICar5, not AICar05.



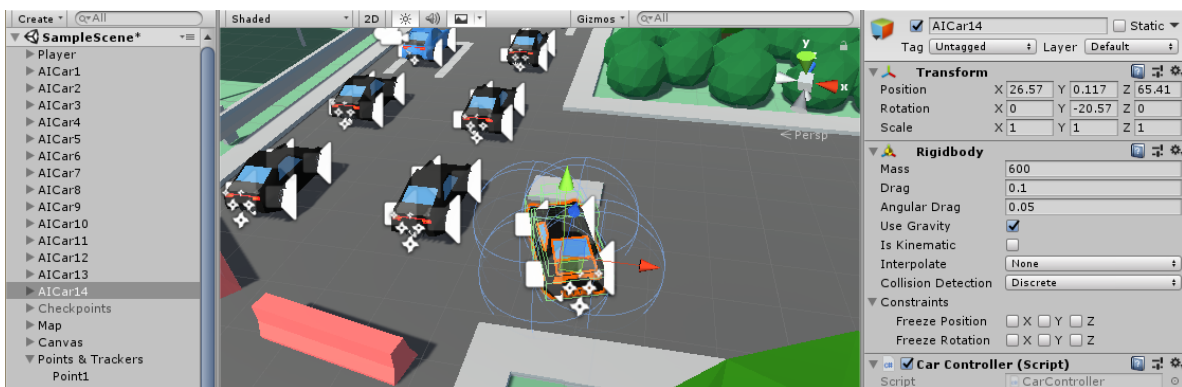
Step 2:

Go to Points & Trackers game object, duplicate an existing Tracker (CTRL+D or Right-click > Duplicate).

Then rename the tracker (F2 or Right-click > Rename) to the number of your AI car.

In this example, we are creating the 14<sup>th</sup> AI car so the tracker name has to be: TrackerAICar14.

Now you can select your new AI car and move it wherever you want it to spawn!



The car will automatically follow the Points and the bot selector will automatically set its maximum of bots to 14.



That's it for Racing Starter 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, feedback is appreciated.*

Other SpinMotion assets: <https://assetstore.unity.com/publishers/41442>

*(You can find more game modes, split-screen, car customization, track creator, cash & unlockables system, mobile support, settings menu, music integration, open-world, minimap system, and more.)*