

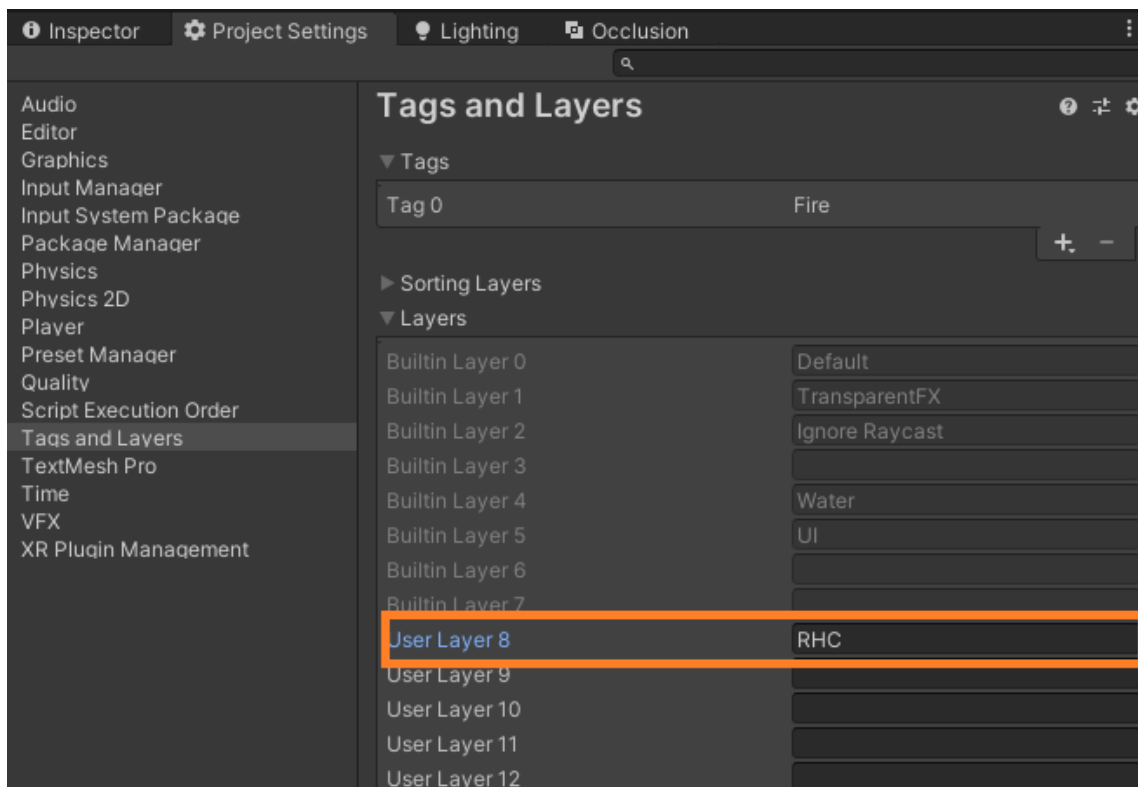
# Realistic Hovercraft Controller

First of all, thank you for purchasing and using Realistic Hovercraft Controller! This documentation will guide you to understand how the system works.

## First to Do!

Always backup your project before updating any asset or Unity Editor. Keep your own assets outside of the Realistic Hovercraft Controller folder. Delete the entire folder, and import updated version.

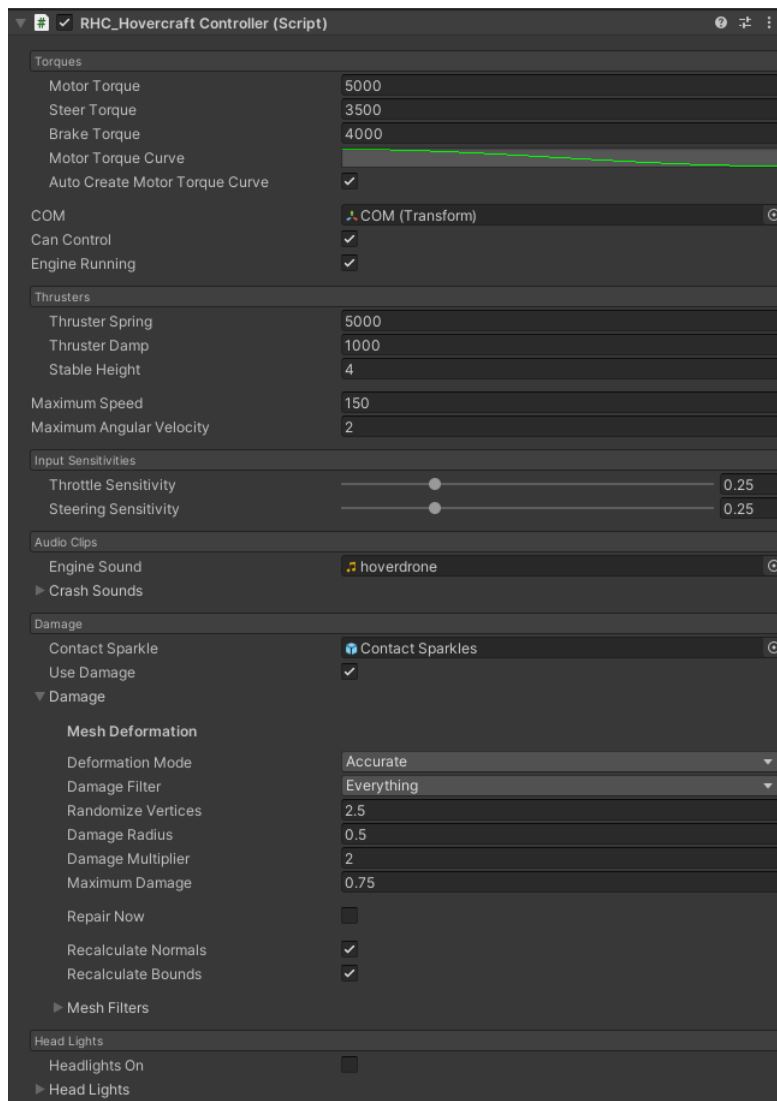
RHC is using [LayerMask](#) for avoid unwanted raycast hits, lens flare intensity and scale calculation, and ignore unnecessary projector layers.



## Overview

Each vehicle has it's own [RHC\\_HovercraftController.cs](#) script. Each vehicle is responsible for own [RHC\\_HovercraftController.cs](#). Lights, cameras, exhausts are addons and not required as an essential. Inputs are processed by [RHC\\_InputManager.cs](#) script. It will receive corresponding inputs from selected device.

## RHC\_HovercraftController.cs

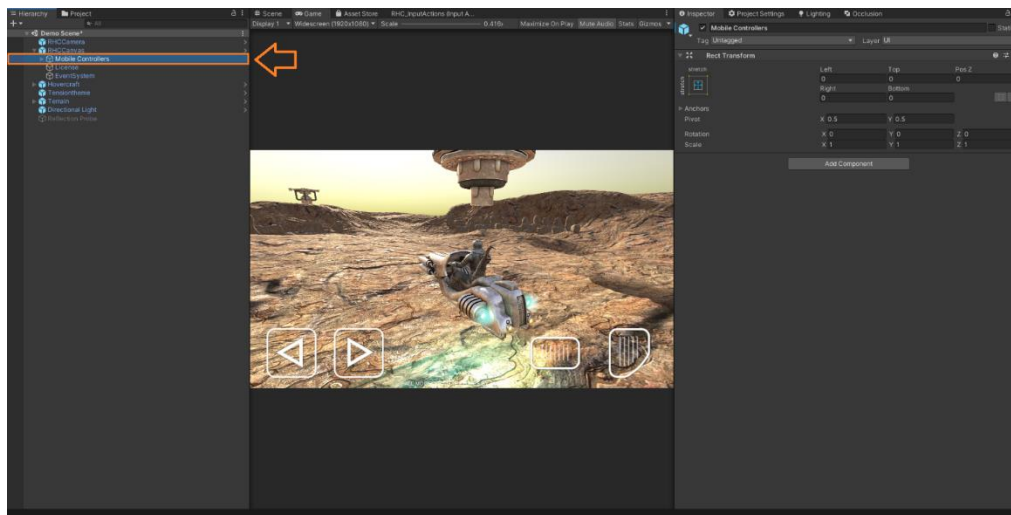


**6 Main Categories** for easily and understandable creating / configuring vehicles.

**Motor Torques, Thrusters, Input Sensitivities, Sounds, Lights and Damage.**

# Mobile Controller

Project is using new input system. Therefore, no need to use independent system for mobile controllers anymore. You can enable / disable mobile buttons on your **RHCCanvas** directly. Each button has “**OnScreenButton**” component attached to it. It will simulate keyboard inputs.



## How To Create New Vehicles

Be sure your vehicle has proper X, Y, Z orientation with correct pivot position. Script and behavior depends on vehicle **X, Y, Z** directions and pivots. So, your vehicle model and wheel models transform directions **should MUST** be correct. Unity has updated EULA about it at 2018, and any models with improper axes and directions will be rejected from Asset Store.

Be sure you are in **PIVOT** and **LOCAL** mode while checking axes of your model.



X should be **Right**,

Y should be **Up**,

Z should be **Forward**.

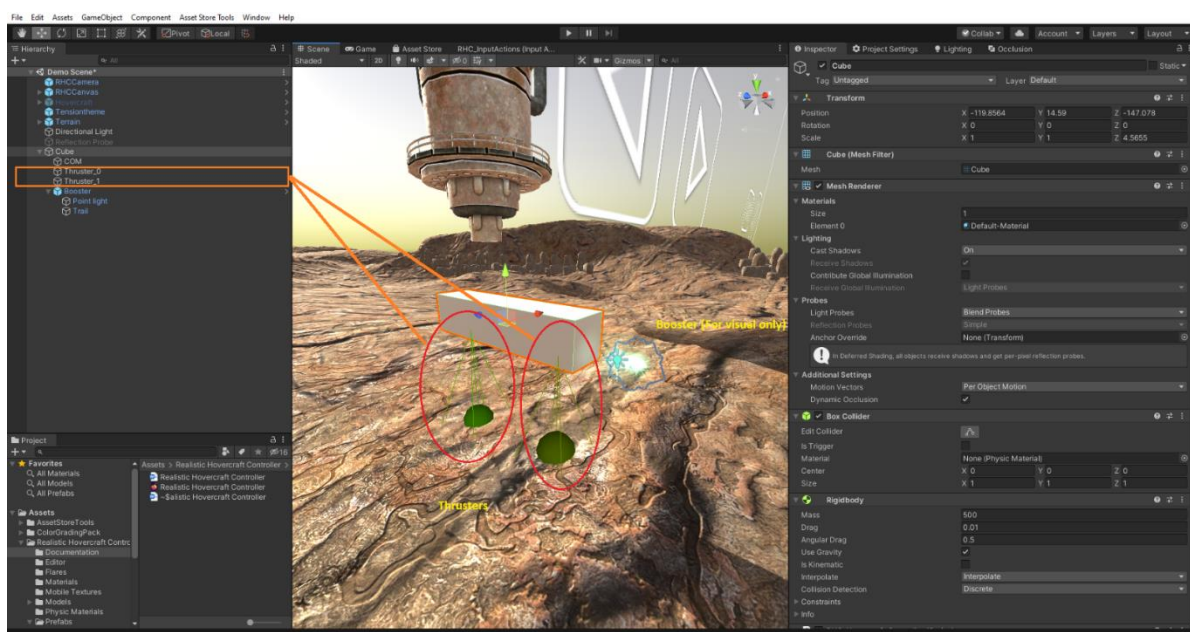
Drag and drop your vehicle model to your scene and let's get started;

You have to add **Main Controller (RHC\_HovercraftController)** to root of the vehicle. Just select your vehicle model on your scene, and **RHC\_HovercraftController** component.

As soon as when you added **RHC\_HovercraftController** script to your car, Rigidbody component will be added automatically. Set your mass to around **500** for this type of the vehicles. Interpolate Mode = Interpolate, Angular drag is around **0.1 – 0.5** for medium angular velocity. These rigidbody settings are applied when you create your vehicle automatically.

Your vehicle **MUST** have any kind of **Colliders** (Such as **Box Collider**, or **Mesh Collider** etc...) for the body. Otherwise, physics won't work. COM position is important also. Double check it's position.

Your vehicle will have two thrusters at the first initialization. Place them correctly. You can increase amount of the thrusters by adding more thrusters. Simply, duplicate and replace them. You can have any amount of the thrusters. If you want lighting effect, create a light and select it.



# New Input System for Realistic Hovercraft Controller

**RHC\_InputManager** is responsible for receiving player inputs via Unity's New Input System. Inputs in this class has been used for controlling the vehicles and the cameras.

Inputs in **RHC\_InputManager** and input types (axes/buttons/vectors) have been explained in the table below;

Input Name	Input Type	Button / Axis	Info
Throttle	Axis -1f, 1f	W, Right Trigger	
Brake	Axis 0f, 1f	S, Left Trigger	
Steering	Axis -1f, 1f	A/D, Left Stick Left, Left Stick Right, Mouse X	
Elevation	Axis -1f, 1f	Left Shift, Left CTRL	
Low Beam Lights	Button	L, D-Pad Up	
Orbit	2D Vector	Mouse Delta X/Mouse Delta Y, Right Stick	

Currently added controller types are

- **Keyboard & Mouse**
- **Gamepads**
- **Mobile**

## RHC\_InputActions as Input Actions

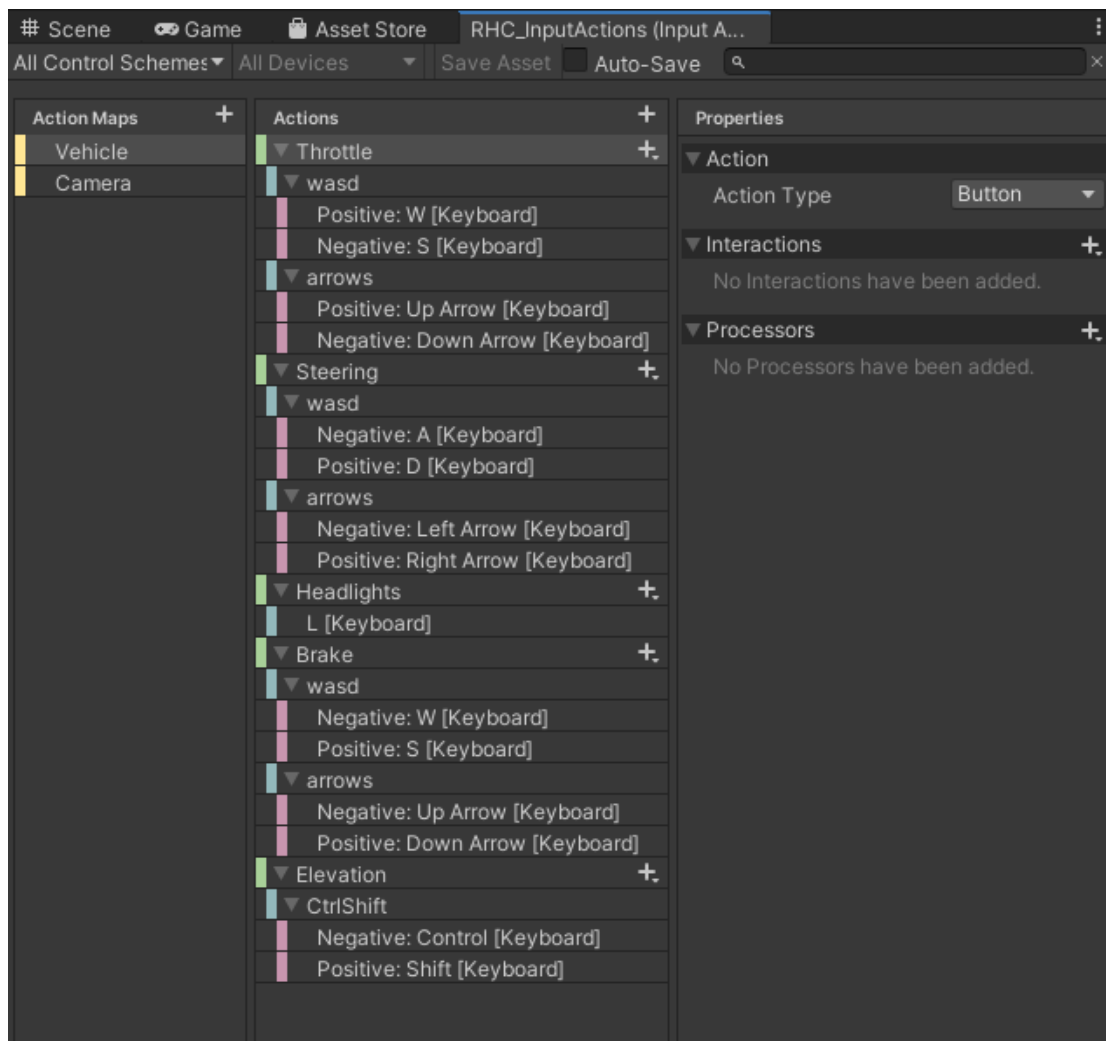
New Input System is using the Input Actions, which can be customized without any code. Each input can be customized with the scheme. You can access default Input Actions of the RHC from [Resources → RHC\\_InputActions](#).

[RHC\\_InputActions](#) have three action maps for **vehicles**, **cameras**, and **optional**. Each action has proper inputs for keyboard & mouse, gamepads.

## How to Add New Inputs, Change Inputs, Remove Inputs

Adding, changing on removing inputs directly from [RHC\\_InputActions](#), which can be found in the [Resources](#) folder of the RHC. Double click the [RHC\\_InputActions](#) to open up the input actions window. There are two controller schemes (keyboard/mouse, and gamepads). You may want to select “all controller schemes” to see all inputs. Do not change the name of the any action map, or action. Otherwise, it will generate new C# script with different variables. Reference scripts will not compile and editor will throw many errors.

Each action has child groups for wide range usement. For example, throttle has three child groups for wasd keys, arrow keys, and gamepad keys. Keys can be changed, or can be added here with the new group. To create a new group, click the plus sign near the action name. Select your positive and negative buttons, and you are done! To remove a group, right click it and click delete. In order to save changes, click “[Save Asset](#)” button at top of the window. Also you may want to enable “[Auto Save](#)” too.



## How RHC\_InputManager Works?

**RHC\_InputManager** is receiving player inputs with **Unity's New Input System**. In old system, inputs were using **Input.GetKey**, **Input.GetAxis**, **Input.GetButton** methods. They were many lines for each controller types, and hardcoded as well. PS4 controller has different inputs, Xbox controller has different inputs, keyboard has different inputs. Instead of using many hardcoded lines, only one line will do the whole job with new Input System.

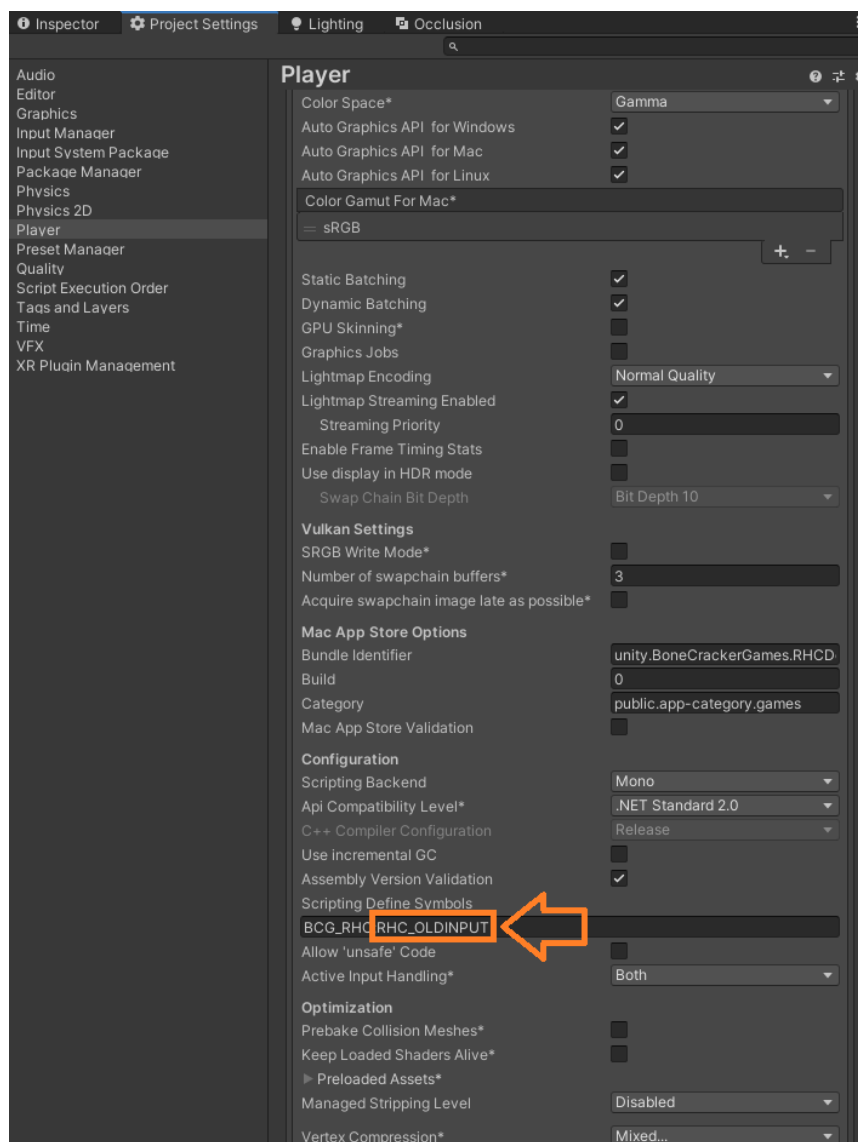
**RHC\_InputManager** is listening all events on **RHC\_InputActions**. For example, if player pushes start/stop engine, "**StartStopEngine\_performed()**" event will be fired. And whatever listens this event, gets notified. **RHC\_HovercraftController** is listening this event too. When player pushes that button, "**RCC\_InputManager\_OnStartStopEngine()**" in **RHC\_HovercraftController** will be fired and corresponding function will be played. In this case, engine will stop, or start.

Same things goes for axis too. There are positive and negative buttons. When player pushes the positive button, maximum range of the axis will be reached. When player pushes the negative button, minimum range of the axis will be reached. When player doesn't push any button, it will be at center. For example, when player pushes right steering button, axis will be 1f, and -1 for the left steering. 0 will be center.

**RHC\_HovercraftController** and **RHC\_Camera** scripts are listening events and receiving axis inputs from the **RHC\_InputManager**.

## How To Switch Old Input System?

All you have to do is, add “**RHC\_OLDINPUT**” to your scripting symbol for using old legacy input system.





## Credits

Driver Sofie, her animations, and her car model made by 3DMAesen. You can access 3DMAesen asset store from this link;

<http://u3d.as/2vg>

All sounds in package are completely Royalt Free. You can use them on any personal or commercial projects. You can't redistribute / resell them.

## License

You can use this package for unlimited games. Both personal and commercial use. **But you can't resell or redistribute any asset in the package on any store (not even any single asset in package)**. I got many reports from my customers about some fake developers are reselling my package on other stores. This is strictly forbidden. You can't resell or redistribute ANY asset from Unity's Asset Store, unless if developer gave you special license for making this. If anyone violates this, he will be banned, and his revenue from package sellings will be interrupted. You can read Unity EULA from this link;

[http://unity3d.com/legal/as\\_terms](http://unity3d.com/legal/as_terms)

So, how you holding up there? You can ask me anything about my assets! If you want change **minor things** in the package, don't waste your time by editing scripts. Just tell me, I'll do my best with no cost. I don't take any projects right now, and I'm not available for hire. Please mail me if you used any of my assets in your game. I'd like to see it in action!

***Ekrem Buğra Özdoğanlar***

***Bonecrackergames@gmail.com***