

# Backhoe Controller

## User Manual v1.0



**WSM GAME STUDIO**

Doing all the hard work for you!

# SUMMARY

<b>1. Intro</b>	<b>2</b>
<b>2. How to Use This Asset</b>	<b>2</b>
2.1. What can it do? (Quick Start Guide)	2
2.2. How it Works	4
<b>3. Backhoe Modules</b>	<b>5</b>
3.1. Backhoe Input	5
3.2. Backhoe Controller	7
<b>4. Vehicle Modules</b>	<b>9</b>
4.1. Vehicle Input	9
4.2. Vehicle Controller	10
4.3. Vehicle Camera	14
<b>5. Additional Info</b>	<b>16</b>
5.1. Customization	16
5.2. Backhoe Carrying Weight Capacity	16
5.3. Rear Steering Vehicle Guidelines	17
<b>6. License</b>	<b>17</b>
<b>7. Contact Info &amp; Support</b>	<b>17</b>

# 1. Intro

Thank you for purchasing “Backhoe Controller”!

This package contains all that you need to include a simple and functional backhoe to your game.

More backhoe models may be included in the future and/or sold separately as extensions ([Here](#)).

It's really simple to use and customize.

## 2. How to Use This Asset

In this section you will learn how this asset works and how to use it based on the sample prefabs.

### 2.1. What can it do? (Quick Start Guide)

This asset contains ready to use backhoe prefabs which you can drive and use to lift and move cargo around your game. It also contains a Stationary Dumpster and a ready to use Rock prefab to be used as cargo.



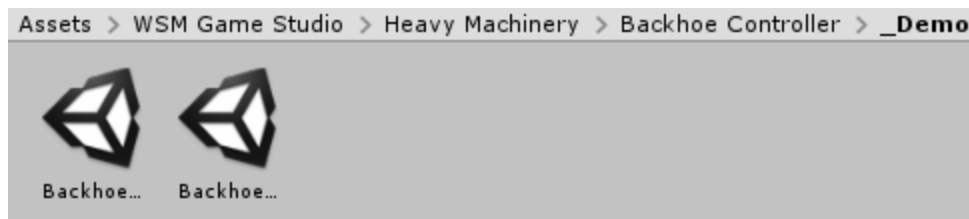
**Figure: Sample Prefabs**

There are 3 backhoe colors by default (yellow, orange and blue), all of which have an old and new texture variation. (albedo, metallic, AO and normal maps included).



**Figure: Brand New x Old Textures Variations**

To see these prefabs in action, please take a look at the demo scene included on the package.



**Figure: Demo Scene**

**Hint:** By pressing the Tab key on the demo scenes you can show or hide the current backhoe input keys. This can be useful for testing the backhoe for the first time.

This asset supports both **Self Leveling** and **Non-Self Leveling** backhoes. (See the [Backhoe Controller](#) section for more information on how to switch the leveling mode).



**Figure: Self Leveling backhoe**

Self Leveling loaders automatically adjust the front bucket angle while raising or lowering the loader frame to keep the bucket leveled and avoid cargo dropping.



**Figure: Non-Self Leveling backhoe**

Non-self leveling doesn't adjust the front bucket angle while raising or lowering the loader frame, which means the bucket will rotate as much as the loader frame.

**Note:** Backhoes prefabs are set to Self leveling by default.

## 2.2. How it Works

This asset was designed with modularity and easy customization in mind. Therefore, the Component Game Design Pattern was implemented to keep player input, backhoe behaviour and vehicle behaviour as decoupled as possible. Each backhoe is composed of four main modules:

- Backhoe Player Input
- Backhoe Controller

- Vehicle Player Input
- Vehicle Controller

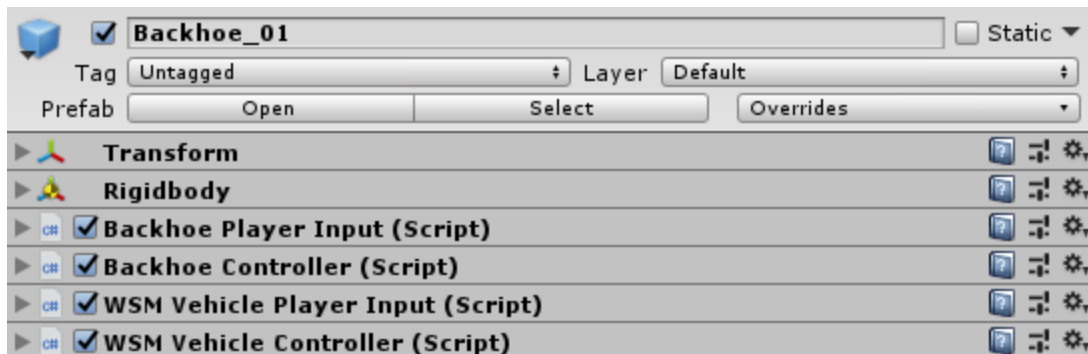


Figure: Backhoe Components

The backhoe and vehicle input modules are responsible to gather and feed player input into the corresponding controller modules. For more information about these modules and how to customize input, please take a look at the [Backhoe Input](#) and [Vehicle Input](#) sections.

The [Backhoe Controller](#) component is responsible for handling all backhoe specific behaviour. Which means, handling the loader frame, boom, arm, stabilizer legs, front and rear buckets movements and SFX's. For more information, please take a look at the [Backhoe Controller](#) section.

The [Vehicle Controller](#) component is responsible for handling all vehicle specific behaviour (Ex: acceleration, steering, vehicle physics, etc.). For more information, please take a look at the [Vehicle Controller](#) section.

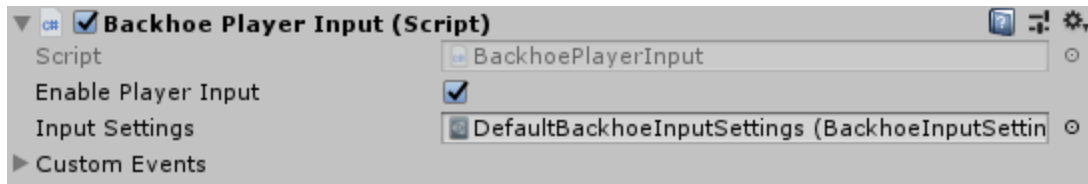
This modular approach was adopted to allow customization by the replacement of any module for third party or custom modules. For example, a client that already owns a custom or third party vehicle controller, may wish to replace the default vehicle related modules and still be able to control the backhoe mechanical parts by using the Backhoe Controller component.

## 3. Backhoe Modules

This section covers the backhoe related modules and settings.

### 3.1. Backhoe Input

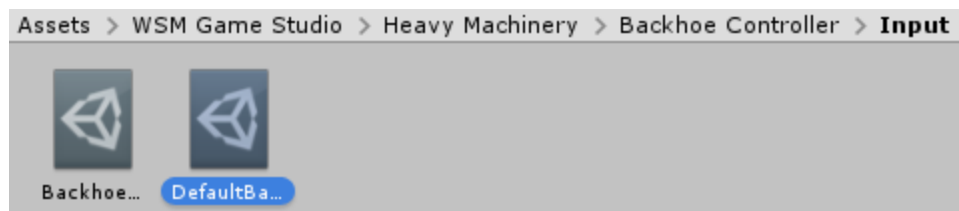
This component is responsible for handling player input for all backhoe specific behaviours. It holds a reference to the Default Backhoe Input Settings file, which contains the keyboard key configurations for controlling the mechanical parts of your backhoe.



**Figure: Backhoe Player Input**

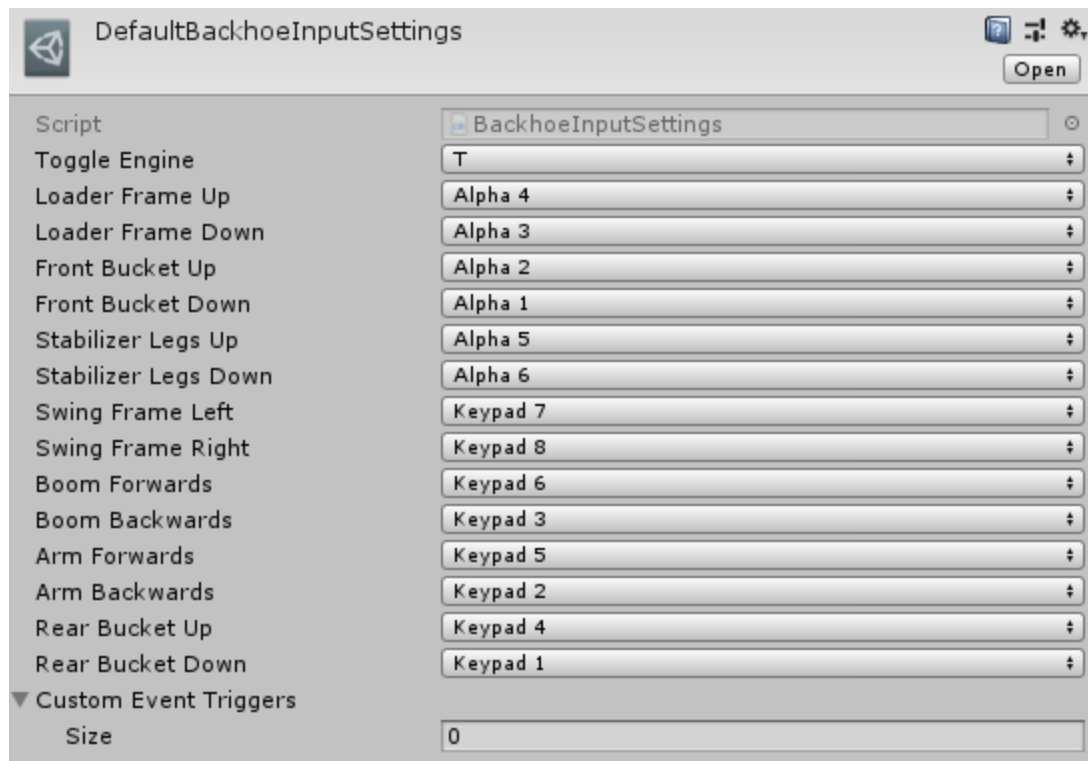
**Note:** Enable Player Input can be used to toggle input on/off.

The Default Backhoe Input Settings file is located under the “WSM Game Studio ► Heavy Machinery ► Backhoe Controller ► Input” folder.



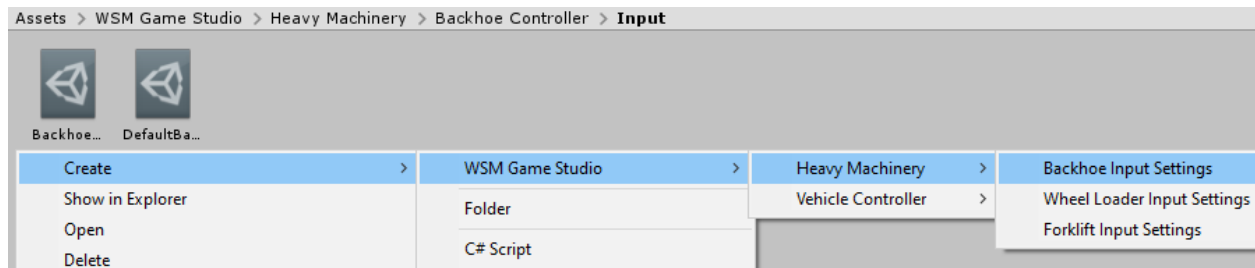
**Figure: Backhoe Input Settings Location**

By selecting this file, you can see and customize the input keys used to control your backhoe.



**Figure: Backhoe Input Settings**

You can also create a new custom input settings by right-clicking on your project and using the Unity Creation menu: “Create ► WSM Game Studio ► Heavy Machinery ► Backhoe Input Settings”.

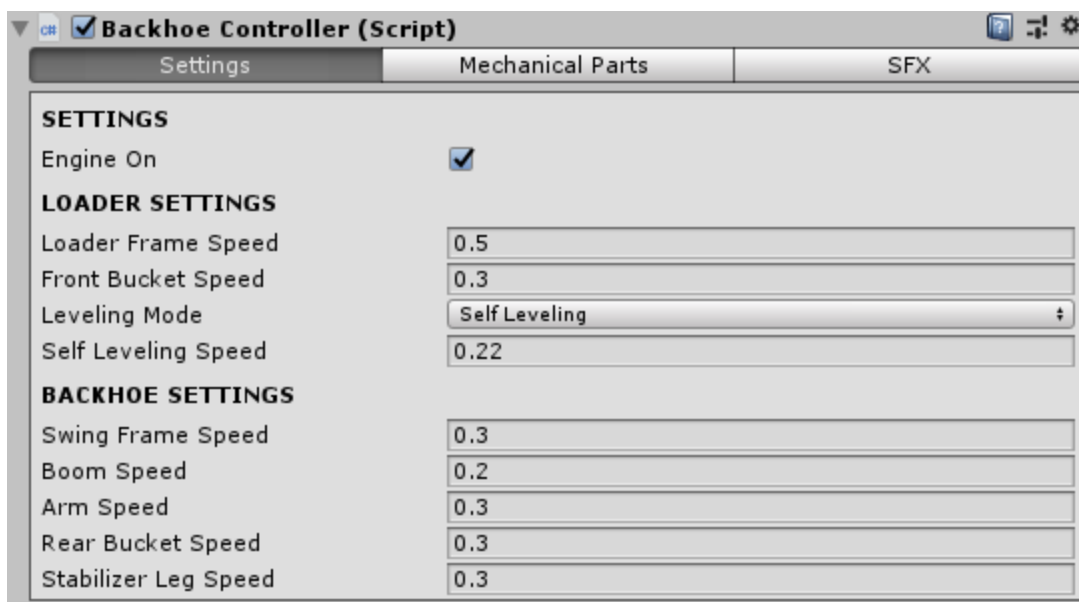


**Figure: Creating a new Backhoe Input Settings**

You can create as many input settings as you wish, this can be very useful for testing different input settings for your game.

## 3.2. Backhoe Controller

This component is responsible for controlling the loader frame, boom, arm, stabilizer legs, front and rear buckets mechanical movements.



**Figure: Backhoe Controller**

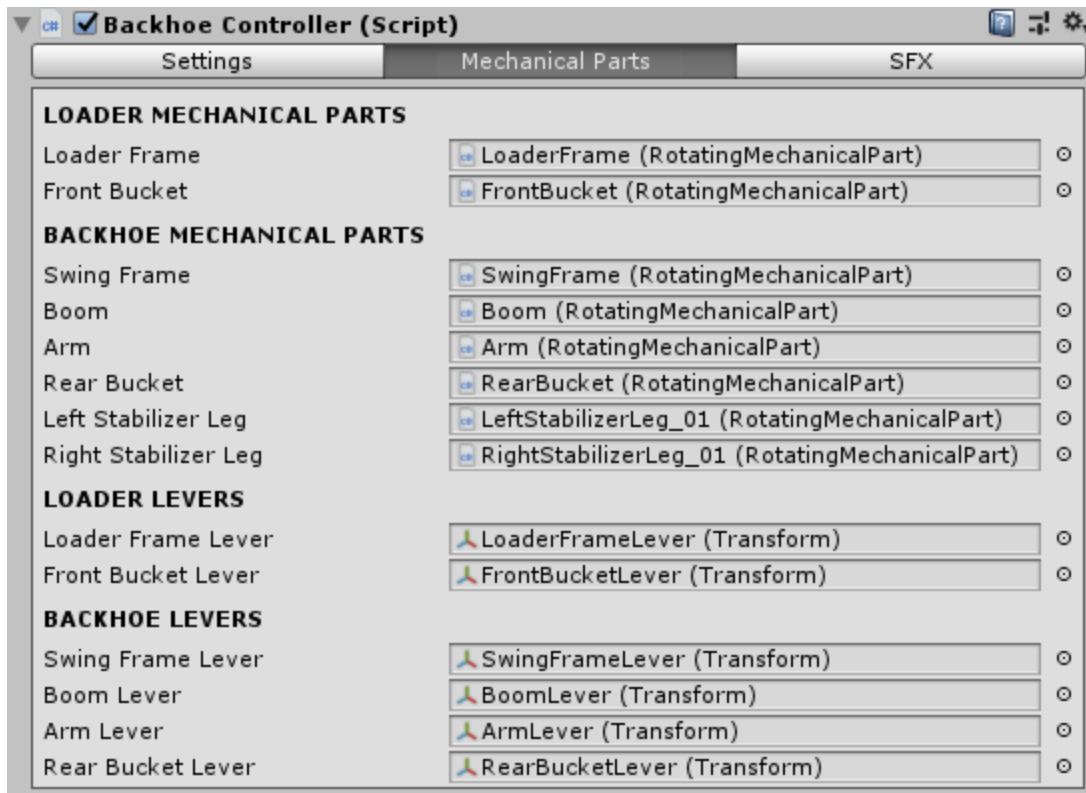
At the Settings tab, you can set up the all mechanical parts movement speed and enable or disable the mechanical parts engine.



You can also switch between **Self Leveling** and **Non-self leveling** front bucket configuration, by changing the Leveling Mode property.

**Note:** The backhoe mechanical parts engine and vehicle engine are separated for the sake of modularity.

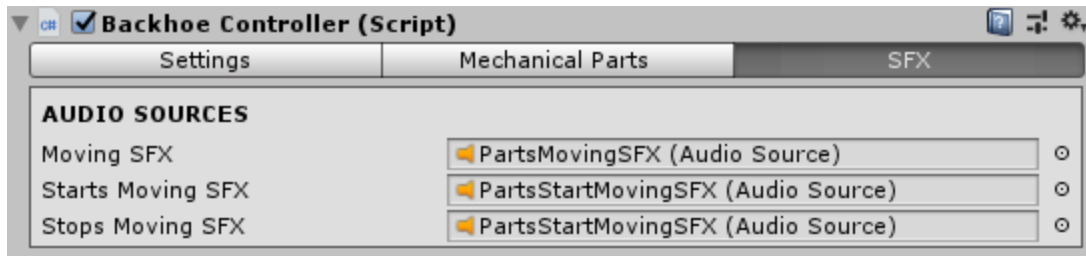
Also, If you change either the loader frame or front bucket speeds, you may need to also adjust the Self Leveling Speed property, to fine tune the bucket self leveling.



**Figure: Backhoe Mechanical Parts**

The Mechanical Parts tab holds the references to all backhoe mechanical parts.

**Note:** Besides the main backhoe mechanical parts, it also supports levers, to simulate the operator manually moving the levers inside the cabin.



**Figure: Backhoe SFX**

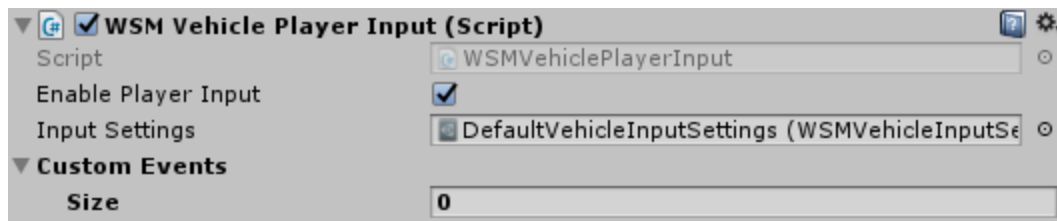
The SFX tab holds references to all the mechanical movements related [Audio Sources](#).

## 4. Vehicle Modules

This section covers the vehicle related modules and settings.

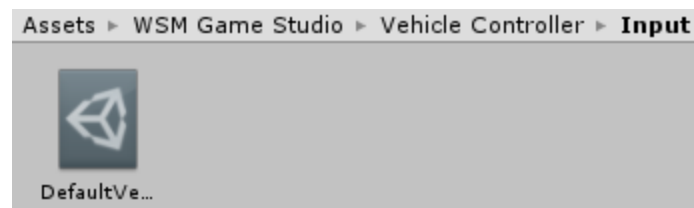
### 4.1. Vehicle Input

This component is responsible for handling player input for all vehicle specific behaviours. It holds a reference to the Default Vehicle Input Settings file, which contains the keyboard key configurations for driving your backhoe.



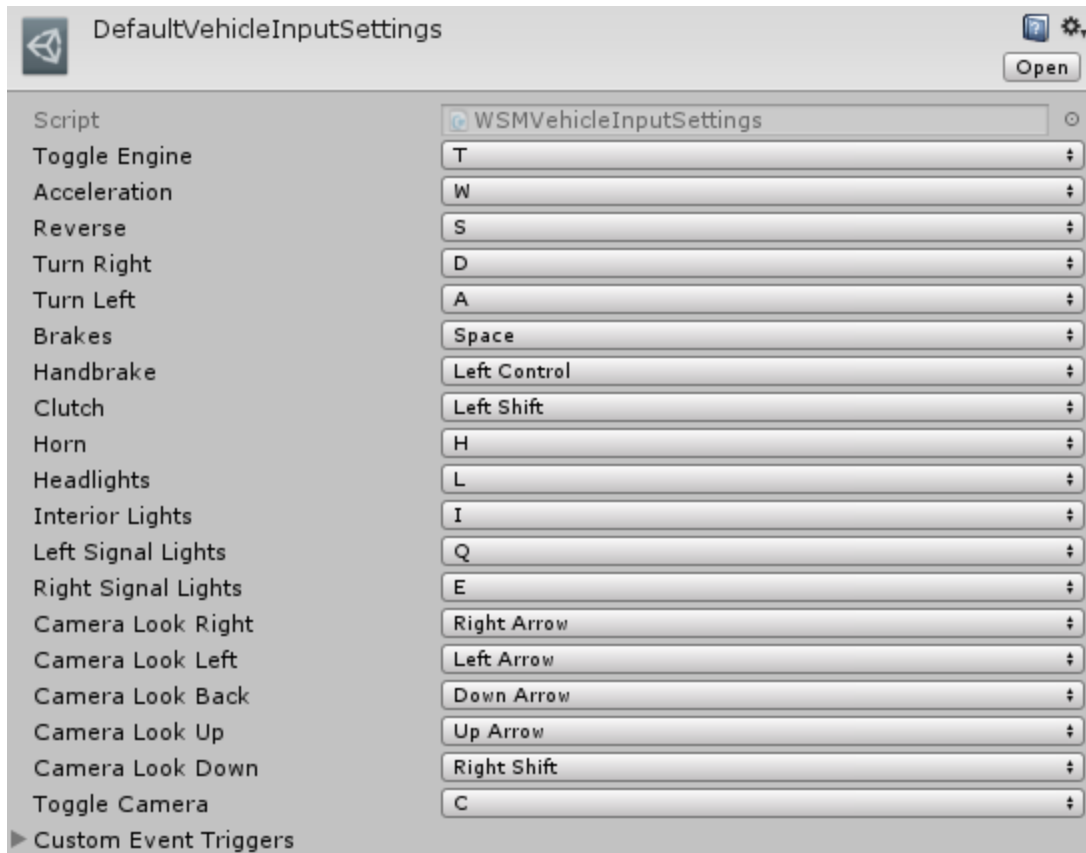
**Figure: Vehicle Player Input**

The Default Vehicle Input Settings file is located under the "WSM Game Studio ► Vehicle Controller ► Input" folder.



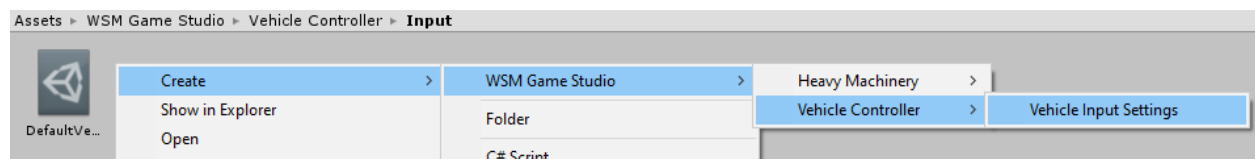
**Figure: Vehicle Player Input Location**

By selecting this file, you can see and customize the input keys used to control your vehicle.



**Figure: Vehicle Input Settings**

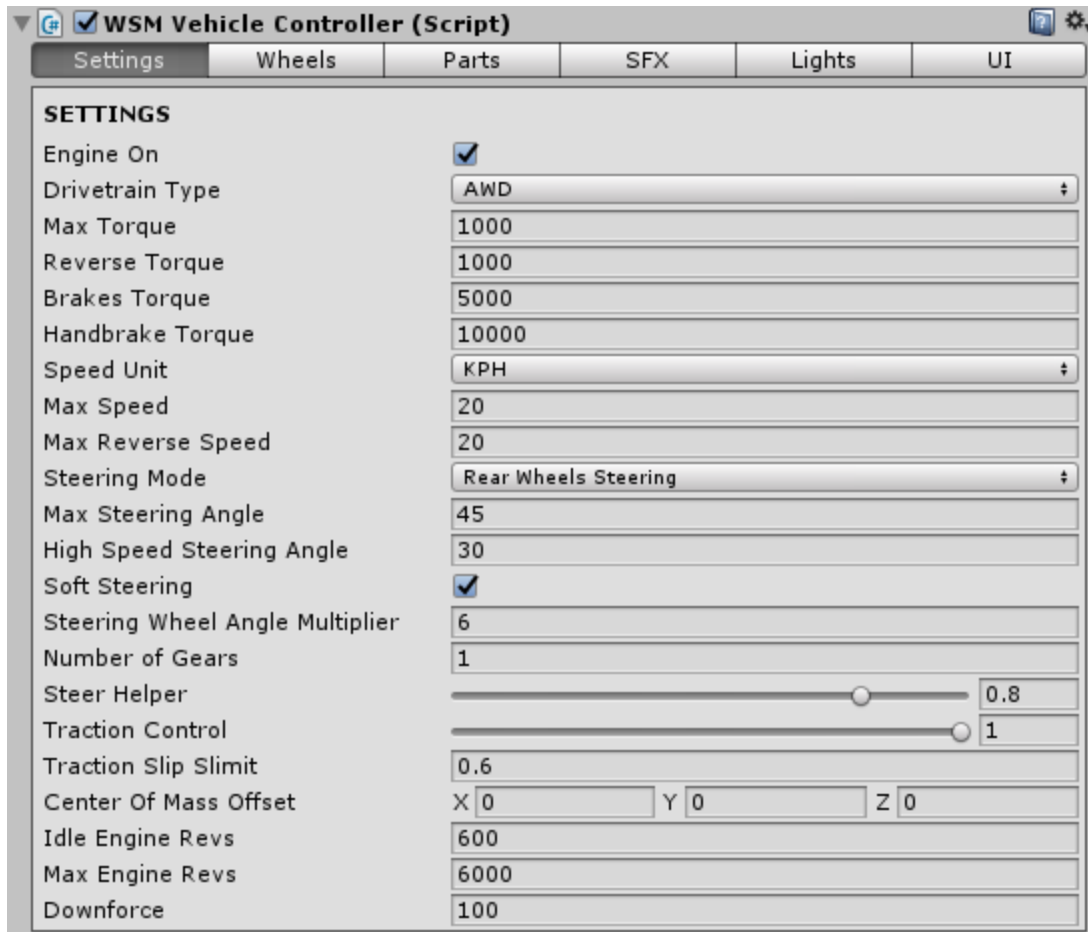
You can also create a new custom input settings by right-clicking on your project and using the Unity Creation menu: “Create ► WSM Game Studio ► Vehicle Controller ► Vehicle Input Settings”.



**Figure: Creating a new Vehicle Input Settings**

## 4.2. Vehicle Controller

This component is responsible for controlling vehicle related behaviours and vehicles physics.



**Figure: Vehicle Settings**

Most vehicle settings located here are very straightforward and use the same nomenclature as the corresponding real world counterpart. The most specific settings will be covered in this section.

**Note:** If you have any questions about any specific setting that was not covered in this section, feel free to email me about it (See [Contact Info & Support](#) section)

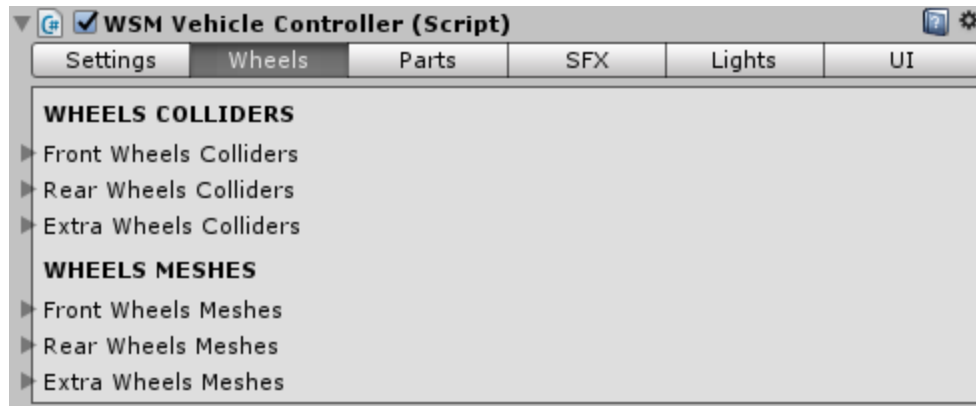


Figure: Vehicle Wheels

The Wheels tab holds references to all [Wheel Colliders](#) and Wheels Meshes. Front and rear wheel references are separated from each other. It was designed this way, so it would be possible to set up which wheels receive traction and which wheels are used for steering.

**Note:** Wheels traction is defined by the **Drivetrain Type** property and steering is defined by the **Steering Mode** property (Settings tab)

If your vehicle has more than four wheels, you can also use the Extra Wheels collections to assign the additional wheels references.

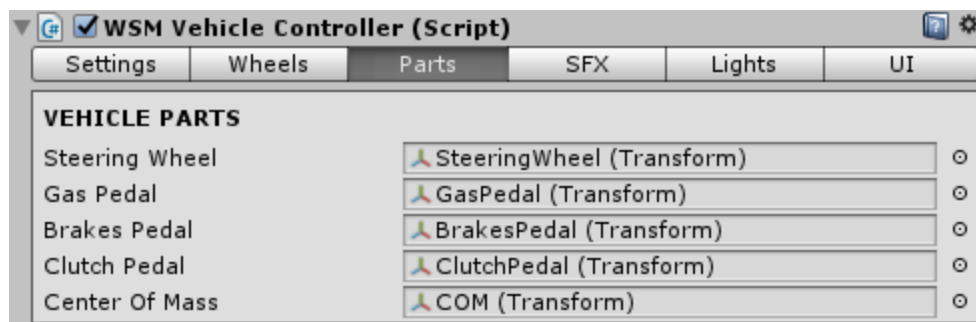


Figure: Vehicle Parts

The Parts tab holds references to all vehicles moving parts. If your vehicle has an open cabin or if your game supports cabin interior view, this is where you can set a reference to your steering wheel and pedals.

The rotation ratio between the wheels and the steering wheel is defined by the **Steering Wheel Angle Multiplier** property (Settings tab).

The Center of Mass reference is used to visualize the vehicle center of mass position. Which can be adjusted by the **Center of Mass Offset** property (Settings tab).

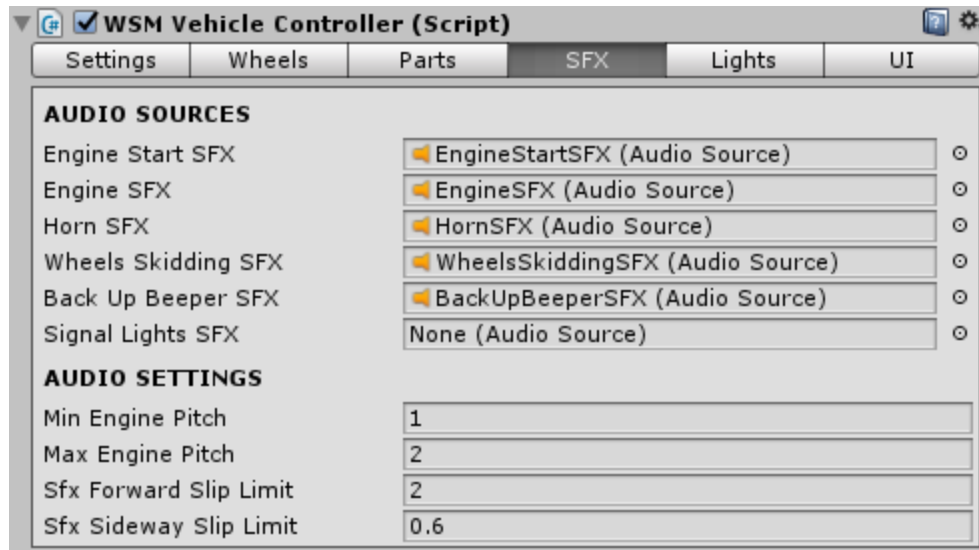


Figure: Vehicle SFX

The SFX tab holds references to the vehicle's [Audio Sources](#). Audio settings are also located here.

The min and max engine pitch properties are used to simulate engine revving and gear switching sounds. The slip limit properties are used to adjust when the wheel skidding SFX should be played.

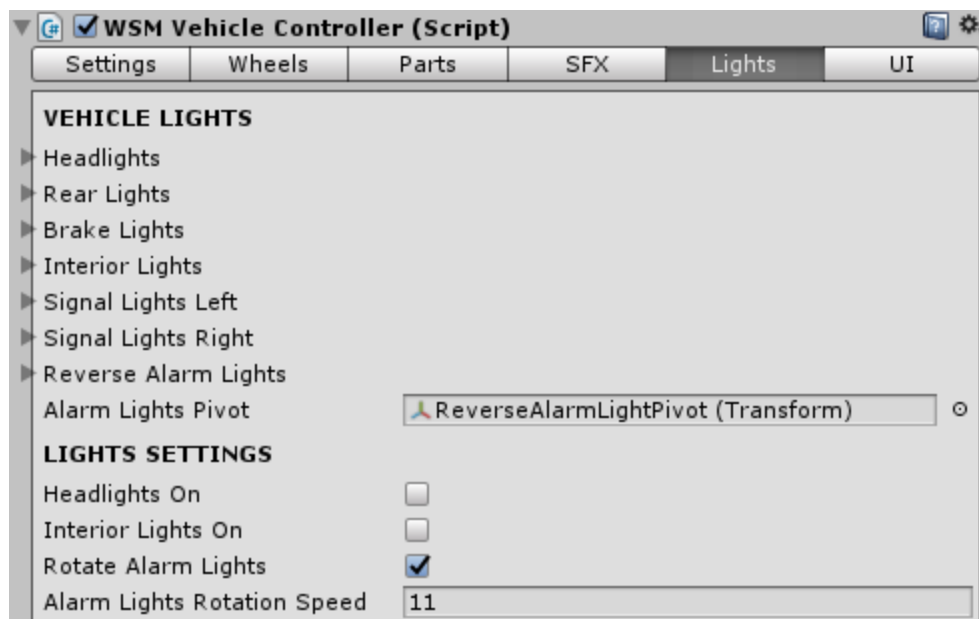


Figure: Vehicle Lights

The Lights tab holds references to the vehicle's lights. Lights settings are also located here.

**Note:** Alarm lights rotation refers to the **Reverse Alarm Lights**. Some trucks and other heavy vehicles usually have rotating lights to warn pedestrians, besides the backup beeper sound.

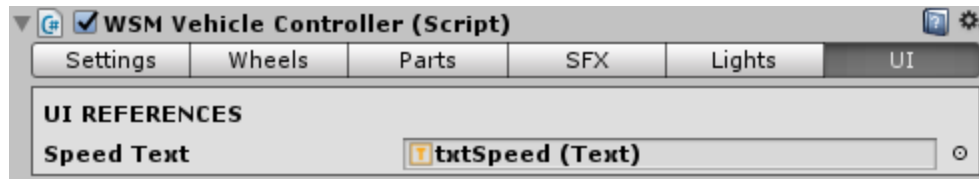


Figure: Vehicle UI

The UI tab holds a reference to the vehicle related UI. For now, it contains only an option for outputting the current vehicle speed.

**Note:** You can choose KPH and MPH by changing the **Speed Unit** property (Settings tab).

### 4.3. Vehicle Camera

A simple vehicle camera is also included in this package. The following camera types are currently supported:

- TPS (Third Person)
- FPS (First Person)
- Top Down

Also, runtime camera toggle can be enabled or disabled here. (See [Vehicle Input](#) section for more details on camera input keys)

**Note:** The Vehicle Camera component only shows the settings of the currently selected camera type.

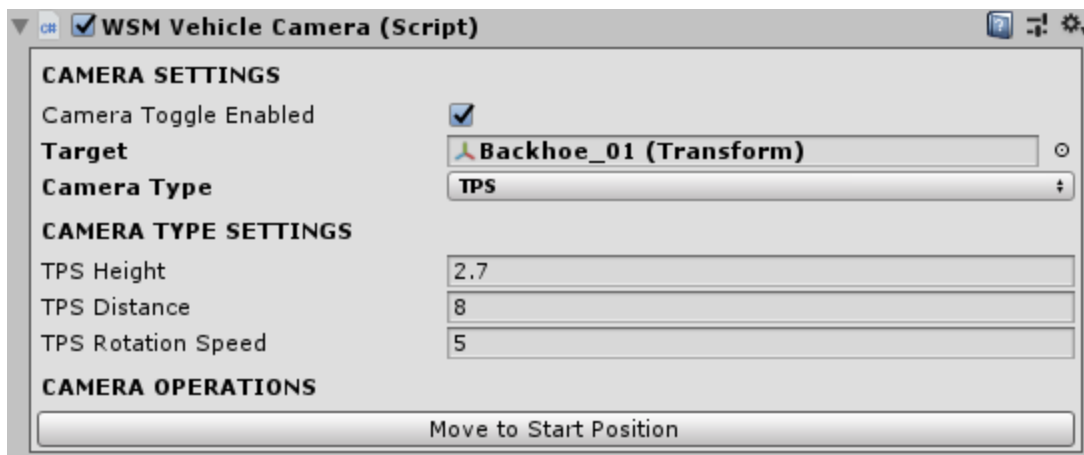


Figure: TPS Camera Settings

You can adjust the height, distance and speed of the TPS camera. While the TPS camera is selected, you can also look left, right or backwards.

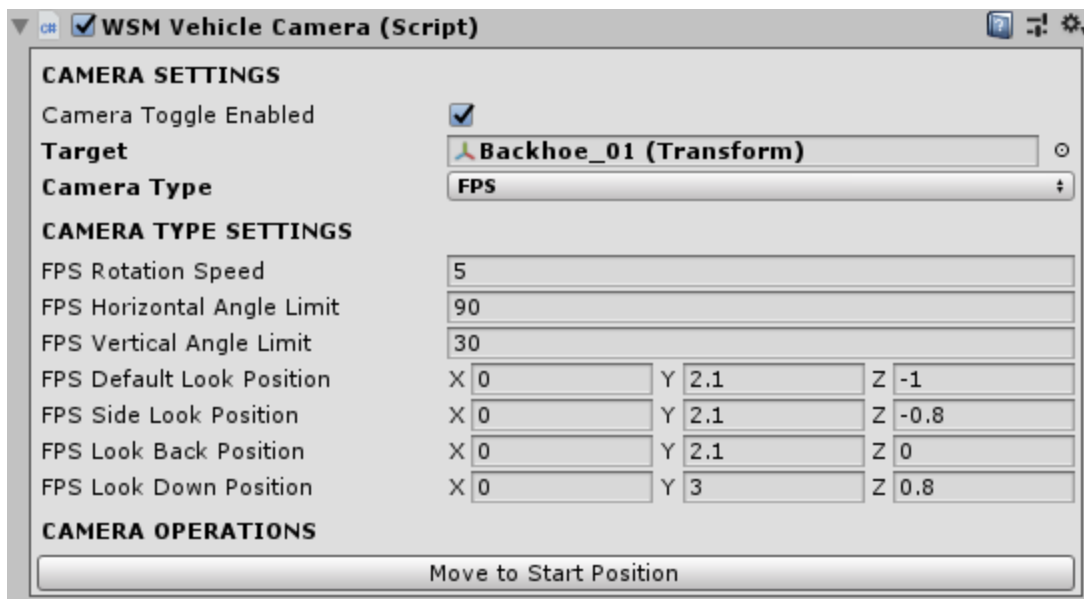


Figure: FPS Camera Settings

While the FPS camera is selected, you can look left, right, backwards, up and down. You can also change the camera position for each look direction. This allows you to simulate some interesting behaviours, like putting the head out of the window when looking backwards.

**Note:** The position shift technique was used to focus on the backhoe bucket while looking down on the FPS camera.

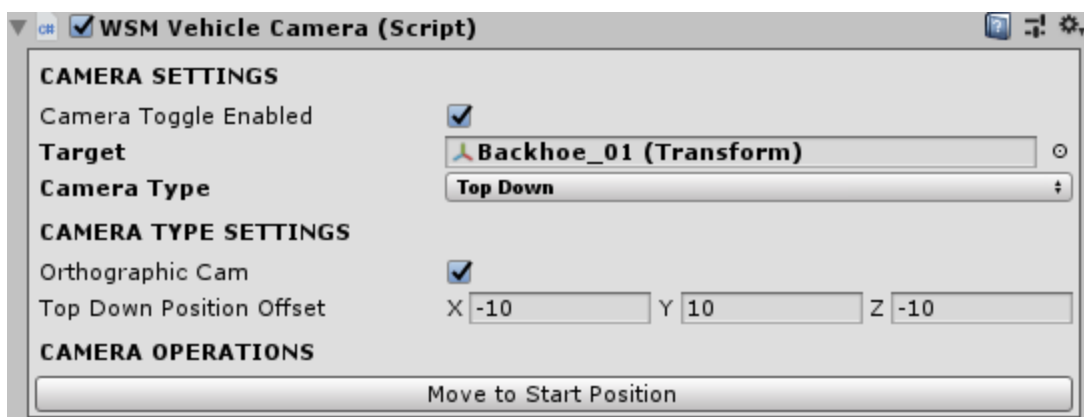


Figure: Top Down Camera Settings

While the Top Down camera is selected, you can adjust the camera position offset and choose between orthographic or perspective camera projection.



**Tip:** Use the *Move to Start Position* operation to preview the camera position after adjusting the settings.

## 5. Additional Info

### 5.1. Customization

This package includes a simple backhoe model for sample purposes. More backhoe models may be included in the future and/or sold separately as extensions ([Here](#)).

If you have your own models, it is possible to create custom backhoes by using the sample prefabs as a starting point and replacing the models using the Unity Editor default prefab editing workflow.

Just make sure your custom backhoe main body, bucket, loader frame, doors, cylinders and wheels models are separated from each other as shown in the sample image below.



**Figure: Default Backhoe Models**

### 5.2. Backhoe Carrying Weight Capacity

As in real life, the backhoes included on this asset have limited carrying weight capacity. Trying to raise cargo above this limit will shift the backhoe center of mass, making it harder to safely transport the cargo or, in case of really heavy cargo, to keep the backhoe wheels on the ground.

Rule of thumb is, a backhoe cannot carry cargo heavier than itself. For example, if you intend to lift a backhoe, you need a bigger and heavier tractor to do the job.

For this asset, the recommended rigidbody mass value for cargo must not exceed 40% of the backhoe mass.

Since sample prefabs are set to 10000 rigidbody mass, the maximum recommended mass sum for cargo is 4000.

**Note:** Due to Unity physics engine limitations, the backhoe rigidbody cannot exceed 10000 mass. Rigidbodies with mass greater than 10000 may cause unpredictable physics behavior on Unity games.

## 5.3. Rear Steering Vehicle Guidelines

The Vehicle Controller included in this package does support rear wheel steering vehicles.

Rear wheel steering vehicles, like forklifts for example, tend to easily [oversteer](#) on higher speeds which can result in undesired vehicle fishtailing.

To avoid [fishtailing](#), increase the Steer Helper property value until your vehicle reaches the desired trade-off between realism and control.

Since vehicles are [nonlinear systems](#), the optimal Steer Helper value will change accordingly to your vehicle aerodynamics, center of mass and settings. It's recommended to test each vehicle and adjust the settings until the desired behaviour is achieved.

## 6. License

By purchasing this asset you are allowed to use it for unlimited games and/or 3D projects (like animations, simulation softwares, etc). Both personal and commercial use.

You are **NOT** allowed to resell or distribute the assets components individually or as part of another asset package (including, models, scripts, etc).

## 7. Contact Info & Support

If you have any questions, need support or have some business inquiries, feel free to get in touch.

The best way to reach us is by email on [wsmgamestudio@gmail.com](mailto:wsmgamestudio@gmail.com).

[Asset Store](#)

[Discord](#)

[Sketchfab](#)

[Instagram](#)

[Facebook](#)

[Twitter](#)

[Youtube Channel](#)