



VRC OWO System World Integration.

In this documentation we will teach you the basic install, setup and use of the OWO World Integration (O.W.I.)

This includes:

- Unity project setup (Basic vrc unity knowledge).
- Installing the O.W.I. files.
- Setting up the integration files in your vrc world.
- Learning how to make sensations.
- How to apply dynamic sensations to a collision object.
- How to apply static sensations to a collision object.

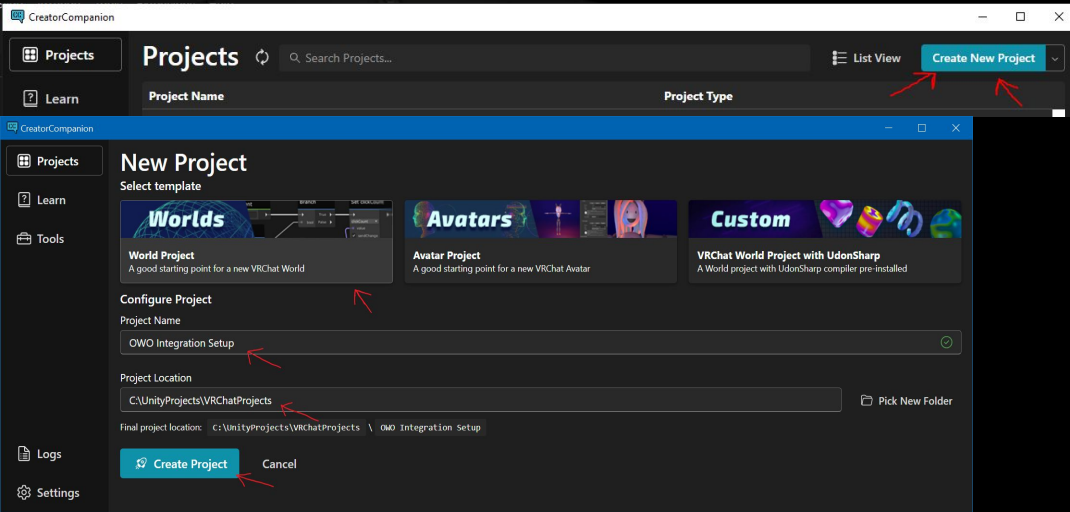
(There is more detailed information in the Specific usage in the Advanced documentation linked on our Github page or in the Assets/O.W.I/Documents folder.)

In this documentation we use O.W.I as an abbreviation of the OWO World Integration name to make it easier to read and less repetitive

VRC OWO System World integration.

Project Setup:

(if you already have a project with VRC SDK 3.4.0 or greater you can skip this page)



Requirements:

Half decent Pc

Current VRC supported version of Unity

VRChat Creator Companion app With SDK 3.4.0 or greater

(if you struggle to find what we are talking about take a look at the images on the left)

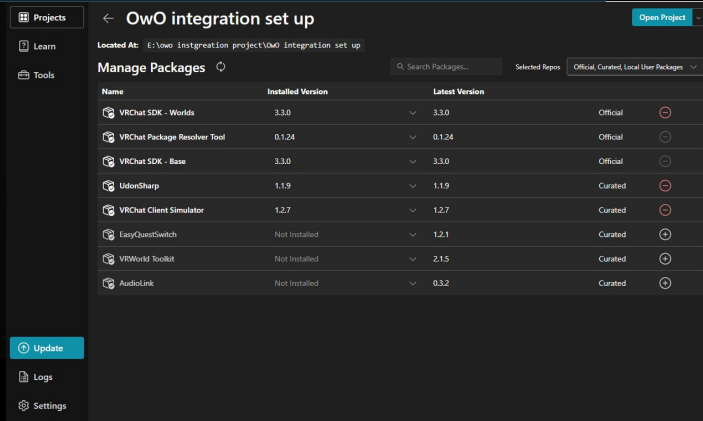
Using the VRChat Creator Companion app, we want to make a new project, we can do this by clicking the Create New Project.

Once you are on the next page we are going to click "World Project", name the project and pick a location for the project (we suggest you make a folder for the project to go into and not put it directly on your drive.)

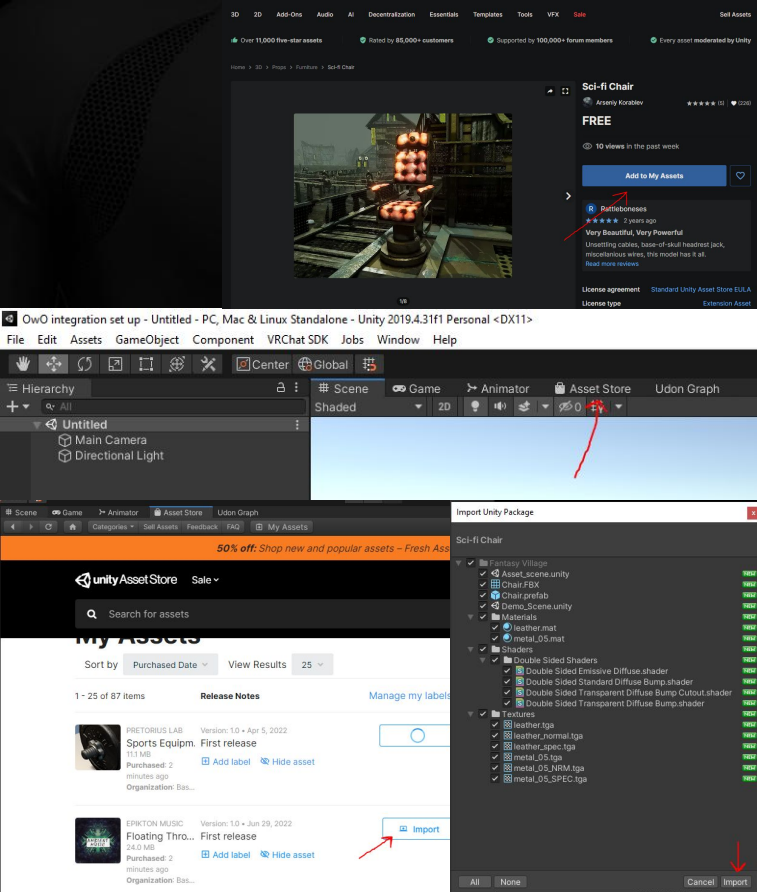
It should bring you to a screen that looks like the following.
All you want to do from here is press "open project" in the top right.

Congratulations we have finished setting up the base project.

(Don't get confused by this image, the name at the top is the name of the project not the integration process)



VRC OWO System World integration.



OWO World Integration(O.W.I)/ Integration Setup:

(some files in the images may be different from what you have due to updates to the GitHub and the Project Itself)

There are some free models that you'll need to import before importing the O.W.I files, they are listed in the read me section of the GitHub; make sure you have downloaded all of them and import them into your project before moving on, or some prefabs will not work.

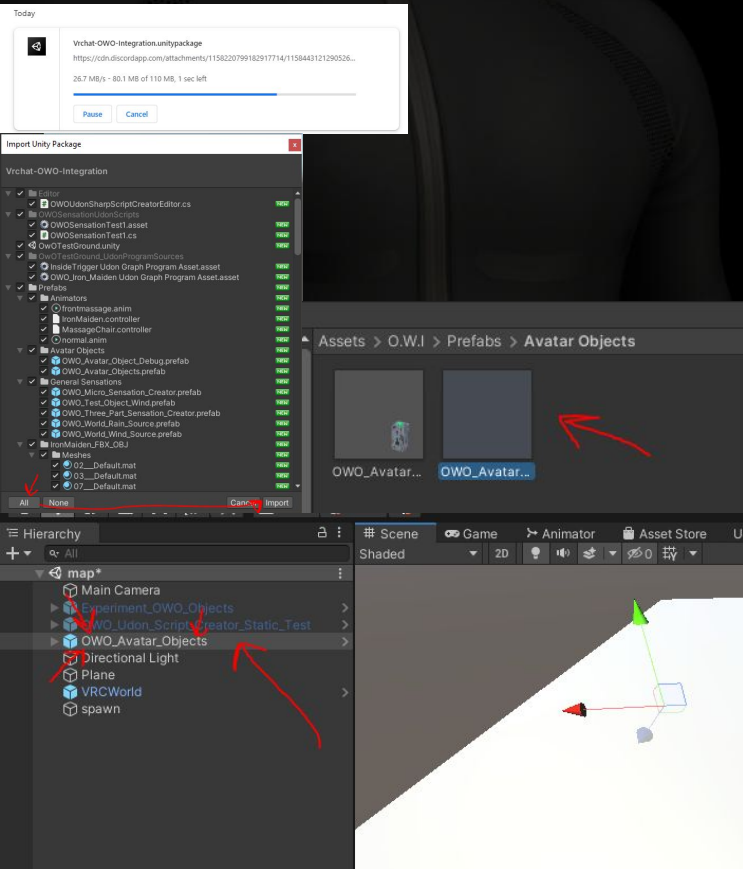
To import the files from the links in the GitHub ReadMe.md file open each of the links and check you are logged into your unity account.

Click "add to assets", once this is done you can go to your unity project, at the top of the Viewport click "Asset store".

Locate the assets you just added and click "download", once downloaded press "import".

An import screen will open on your project, just click "import" and wait for the importing process to finish. (the importing process is the same for all assets from the Asset store).

VRC OWO System World integration.



OWO World Integration(O.W.I)/ Integration Setup:

To start setting up the O.W.I files, you need to download the .unitypackage from our GitHub.

Once downloaded make sure you have your project open, then open the .unitypackage file. This will bring up an import screen.

Once the import screen is open, **click “all”, then click “import”** and wait for the import to finish. (in some cases the loading bar may disappear for a second but then it will come back, we advise you to wait at least 1-2 minutes for all the files to import properly.)

Once the import is finished the first thing you need to do is go to the Avatar Objects folder under the following dir “Assets/O.W.I/Prefabs/Avatar Objects”.

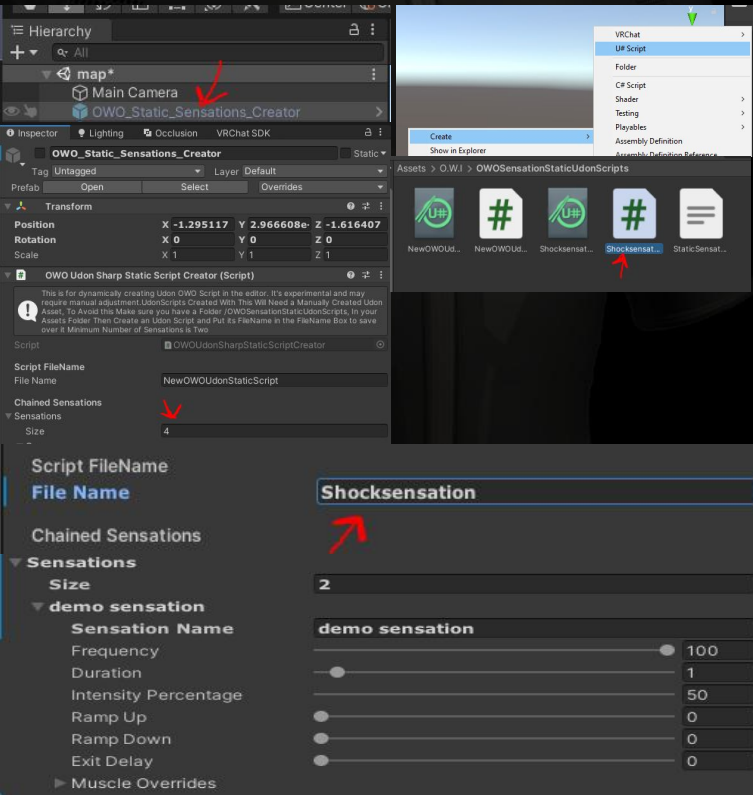
Once here you'll want to drag “OWO_Avatar_Objects.prefab” into the scene. This object spawns all the colliders onto the players so that collision detection works.

Now we can move on to setting up colliders and sensations for your map and we're almost done.

VRC OWO System

World integration.

(keep in mind the way the Static Sensation Creator works is that the first sensation is played then the next sensation then the next and so on for as many sensations that you have put in.)



OWO Sensation Collider setup using Static Sensation Creator (S.S.C):

To start this process drag the "O.W.I._S.S.C.prefab" into your map from this dir: "Assets/O.W.I/Prefabs".

We will be using this to make our static sensations when a player's muscles collide with our game object.

Open the script section of the S.S.C in the inspector, in here we will make our sensations.

We need to make a empty udon script in the follow path:

"Assets/O.W.I/O.W.I.SensationStaticUdonScripts".

You can do this by right clicking inside the folder heading over to "Create" then "U# Script".

We need to put the file name for the script into the S.S.C file name box (If you want to make a new sensation you'll need to make a new script each time or it will overwrite the previous one you made.)

In the S.S.C we need to set an amount of sensations to make the overall feel of the interaction.

(This has a minimum size requirement of at least 2 sensations. If you want just a single sensation, then you'll need to use the micro-sensation prefab in the prefabs folder.)

Once the amount of sensations is set, we can then move on to making each sensation.

(If you don't know how to make sensations, please see our cheat sheet that is in the GitHub.)

Sensation name: This is what you want to call your sensation. It has no effect.

Frequency: This corresponds to the frequency of pulses going to the selected muscle.

Duration: This corresponds to the amount of time the sensation will last.

Intensity: This corresponds to the strength of the event.

Ramp up: This corresponds to the time the sensation takes to hit the set intensity.

Ramp down: This lowers the intensity as it reaches the end of the sensation. (The larger the number the sooner the intensity will start to ramp down)

Exit Delay: This is the wait time before the next sensation starts.

Muscle overrides: This sets the individual muscle intensity limits. (This limits the intensity world side. It doesn't override the limits set in the OWO app.

VRC OWO System

World integration.

(keep in mind the way the Static Sensation Creator work is that the first sensation is played then the next sensation then the next and so on for as many sensations that you have put in.)

Generate Script

Muscle options: This works by telling our script how the sensation should behave according to what muscles were hit. There are four options:

Play first muscle triggered:

This causes the sensation to only be played on the first muscle that was hit.

Play last muscle triggered:

This causes the sensation to only be played on the last muscle that was hit

Play muscle triggered in order:

This causes the sensation to play in the order that muscles where hit.

Play all muscles triggered:

This causes the sensation to be played on all muscles at the same time that were hit.

Muscle Options 1

Play All Muscles Triggered

Sensation Priority

1

Collider Capture Timer

0.1

OWO Sensation Collider setup using Static Sensation Creator (S.S.C):

Now you have set all of your variables, press "generate script". (This may lag / take some time depending on your system.)

Now that we have our sensations script, we need to make an object with a collider and a rigid body.

This can be done by looking in the hierarchy , selecting your object. Using the inspector press "add component" now search for "rigid body". (if you don't want your object to move around you need to turn off "use gravity" and turn on " is kinematic". If you want your object to be moveable, you'll want to add another component " VRC Object sync". If you also want to pick up the object add " VRC Pick up" as well.)

We now need to add another component called a collider, if you want you can use a "mesh collider", but it needs to have "convex" enabled or the script will not work. We suggest you use a "Box collider" instead. (if you want to walk through your object you can turn on " Is Trigger". This is not supported by us and can cause issues with the script, use at your own risk.)

We are now at the last step. Drag the script (not the u# asset) we made onto the object in the inspector to add it as a component, or you can search for the script by its name when clicking "add component".

Once this is added, we have a few more things to set.

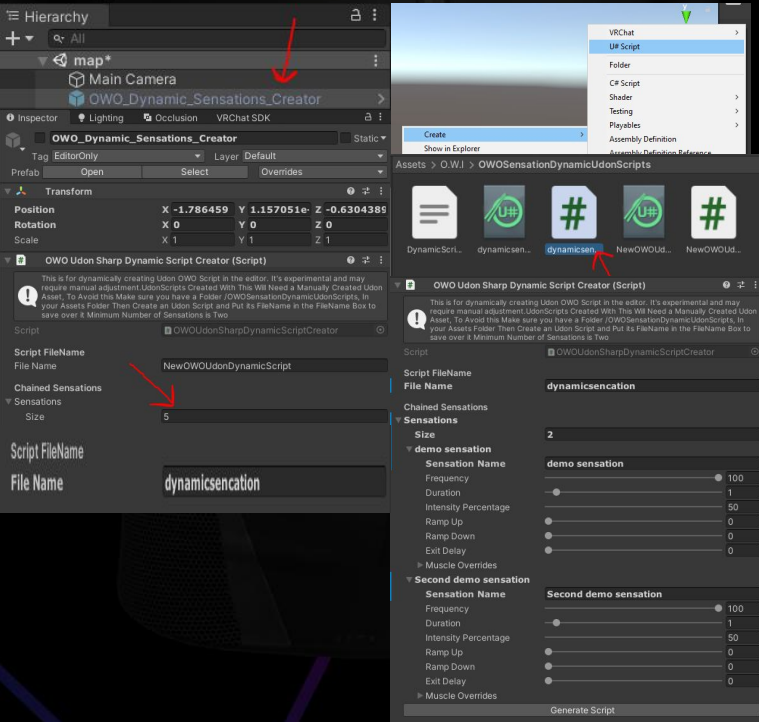
Sensation priority: This is used when you are already experiencing one sensation. If the new sensation has the same or a higher priority then it will override the current sensation and play its own.

Collider capture timer: This is the amount of time we allow for the script to capture each collider that was hit starting when the first collider is detected.

Muscle options can be seen to the left of this page.

VRC OWO System World integration.

(Keep in mind the way the dynamic sensations work is that the first sensation plays on the first muscle that was hit, the second sensation to the second muscle and so on. HOWEVER the last triggered sensation plays across all muscles that were hit, if you don't want this, Click the "Ignore Grouped Muscle Sensation" box this can also be changed in the Inspector after creation.



OWO Sensation Collider setup using Dynamic

Sensation Creator (D.S.C):

To start this process drag the "O.W.I_D.S.C.prefab" into your map, from this dir: "Assets/O.W.I/Prefabs".

We will be using this to make our dynamic sensations when a player's muscles collide with our game object.

Open the script section of the D.S.C in the inspector, in here we will make our sensations.

We need to make a empty udon script in the follow path:

"Assets/O.W.I/O.W.I.SensationDynamicUdonScripts"

You can do this by right clicking inside the folder heading over to "Create" then "U# Script".

We need to put the file name for the script into the D.S.C file name box (If you want to make a new sensation you'll need to make a new script each time or it will overwrite the previous one you made.)

In the D.S.C we need to set an amount of sensations to make the overall feel of the interaction. (This has a minimum size requirement of at least 2 sensations. If you want just a single sensation, then you'll need to use the micro-sensation prefab in the prefabs folder.)

Once the amount of sensations is set, we can then move on to making each sensation.

(If you don't know how to make sensations, please see our cheat sheet that is in the GitHub)

Sensation name: This is what you want to call your sensation. It has no effect.

Frequency: This corresponds to the frequency of pulses going to the selected muscle

Duration: This corresponds to the amount of time the sensation will last

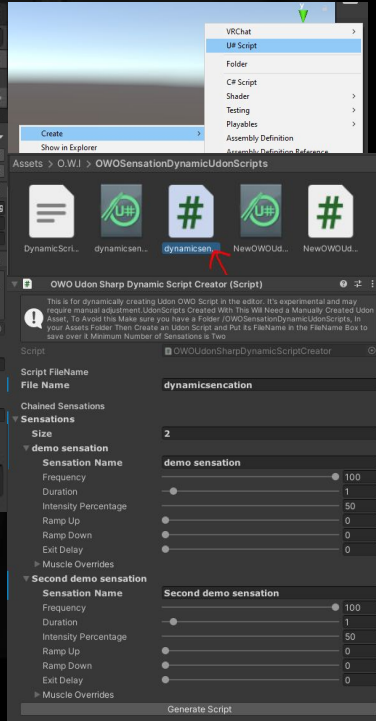
Intensity: This corresponds to the strength of the event

Ramp up: This corresponds to the time the sensation takes to hit the set intensity.

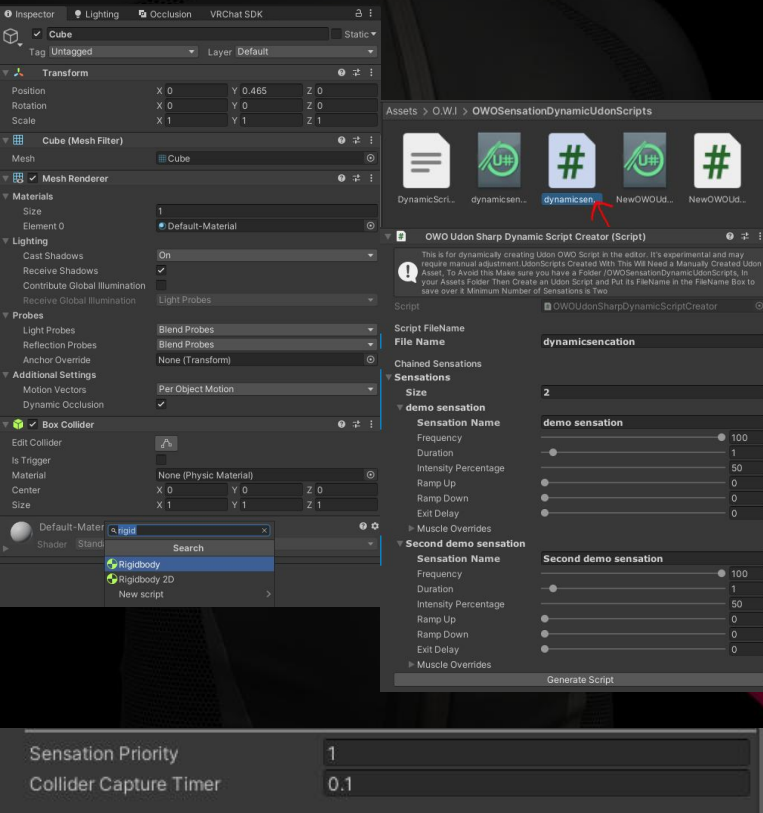
Ramp down: This lowers the intensity as it reaches the end of the sensation (the larger the number the sooner the intensity will start to ramp down)

Exit Delay: This is the wait time before the next sensation starts.

muscle overrides: This sets the individual muscle intensity limits. (this limits the intensity world side. It doesn't override the limits set in the OWO app)



VRC OWO System World integration.



OWO Sensation Collider setup using Dynamic Sensation Creator (D.S.C):

Now you have set all of your variables, press “generate script”. (This may lag / take some time depending on your system.)

Now that we have our sensations script, we need to make an object with a collider and a rigid body.

This can be done by looking in the hierarchy, selecting your object. Using the inspector press “add component” now search for “rigid body”. (If you don't want your object to move around you need to turn off “use gravity” and turn on “is kinematic”. If you want your object to be moveable, you'll want to add another component “VRC Object sync”. If you also want to pick up the object add “VRC Pick up” as well.)

We now need to add another component called a collider, if you want you can use a “mesh collider”, but it needs to have “convex” enabled or the script will not work. We suggest you use a “Box collider” instead. (If you want to walk through your object you can turn on “Is Trigger”. This is not supported by us and can cause issues with the script, use at your own risk.)

We are now at the last step. Drag the script (not the u# asset) we made onto the object in the inspector to add it as a component, or you can search for the script by its name when clicking “add component”.

Once this is added, we have a few more things to set.

Sensation priority: This is used when you are already experiencing one sensation. If the new sensation has the same or a higher priority then it will override the current sensation and play its own.

Collider capture timer: This is the amount of time we allow for the script to capture each collider that was hit starting when the first collider is detected.



VRC OWO System World Integration. Credits

Creators:

RevoForge
SonoVR

Document writer / VRC Contributor:

BassBoostedDuck

Testers:

areno127
minthnir
mutethecyberwolf
_nicedicer
spokeek
ahrivr
bigglesworth43
himynameskyle
kasri

kiffin
realmasterlink
nanoade
mrdings
baymaxxxxx
spike_felion
mcsolo
.pepie
rumrobot

Feel free to contact us on the
OWO discord server if you
have any issues or bugs.