Table of Contents

1. Targeting system documentation 2.0

1.1 Introduction

Installation of the plug-in is quite straightforward and easy, if you know basics about Unity. Free version does not have all the features, so this is a stripped version of the documentation and may contain small references to the other features. The reason for this is that working with the preview packages is not always easy and there are developers new to Unity. This documentation should get you up to speed and if not, then join the discord channel or send email. Contact information at the end.

The system is made of features. Each feature can be toggled on/off at code level.

This happens by using scripting symbols. More info at chapter 1.2.2 page 7.
The demo project requires starter assets and inputs.
ENABLE_INPUT_SYSTEM
STARTER_ASSETS_PACKAGES_CHECKED

1.2 Installation Quick start.

Before installation remember to have backups of your project. After targeting system v2.0.0, entities related packages are no longer needed for game object only mode. If you don´t care about entities, then use the game object only guide if necessary, else use the entities guide.

Before downloading the plug-in make sure following unity or newer.

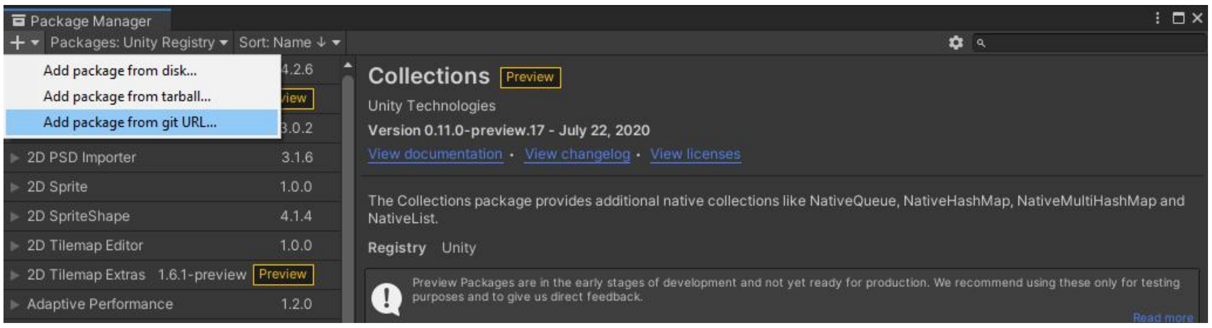
- ✓ *Unity 2021.3.4f1*

If you are already familiar with package manager and installing packages, then you can skip to "Basic usage" in chapter 2 and here just check the packages required. Or use the manifest included to quickly get working packages.

Packages can be installed from Unity´s package manager. Here is a guide on how to install packages in case you get stuck. Also the base plugin does not need the new input system or UI elements, if you don´t use them yourself, then its safe to delete Demo and examples folders from plugin hierarchy.

https://medium.com/@jeffreymlynch/where-are-the-missing-preview-packages-in-unity-2020-3ad0935e4193

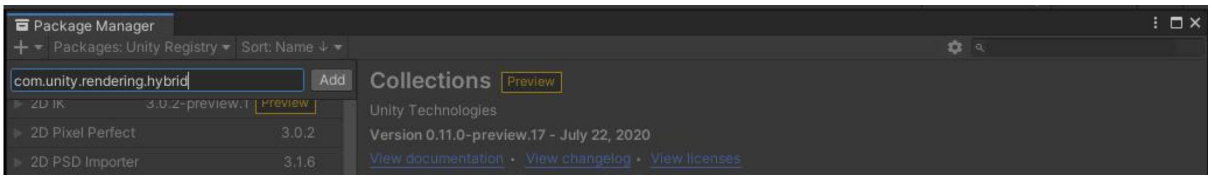
Open Package manager and click plus icon at the upper left corner. In Unity 2020 there should be an option for adding package from url.



The Collections package provides additional native collections like NativeQueue, NativeHashMap, NativeMultiHashMap and NativeList.

Preview Packages are in the early stages of development and not yet ready for production. We recommend using these only for testing purposes and to give us direct feedback.

after that, this one comes up.



com.unity.rendering.hybrid

Here you can add the package addresses. You will need that next to install the plug-in.

1.2.1 Installation game object/mono behaviour only version

Link to video. Installation in less than 3 minutes. Alternative way is to use the manifest included in the documentation folder. Replace the manifest in the project packages folder. You will also need Universal rendering pipeline for the examples to work. This is due to Unity dropping standard pipeline support for entities.

Just make sure at least these Unity´s packages are installed. Newer versions of these packages should work too. After they are installed you can download the plug-in from asset store. It also should work the other way around, but this way there should not be any errors.

- ✓ Burst. Version Version 1.6.4 - January 11, 2022. Git url: com.unity.burst
- ✓ Editor Coroutines 1.0.0. Git url: com.unity.editorcoroutines
- ✓ Jobs. Version Version 0.8.0-preview.23 – January 22, 2021 Git url: com.unity.jobs
- ✓ Mathematics Version 1.2.5 - November 08, 2021. Git url: com.unity.mathematics
- ✓ Collections. Version 0.15.0-preview.21 – January 22, 2021. Git url: com.unity.collections
- ✓ Input System. Version 1.3.0 - January 05, 2022 Git url: com.unity.inputsystem

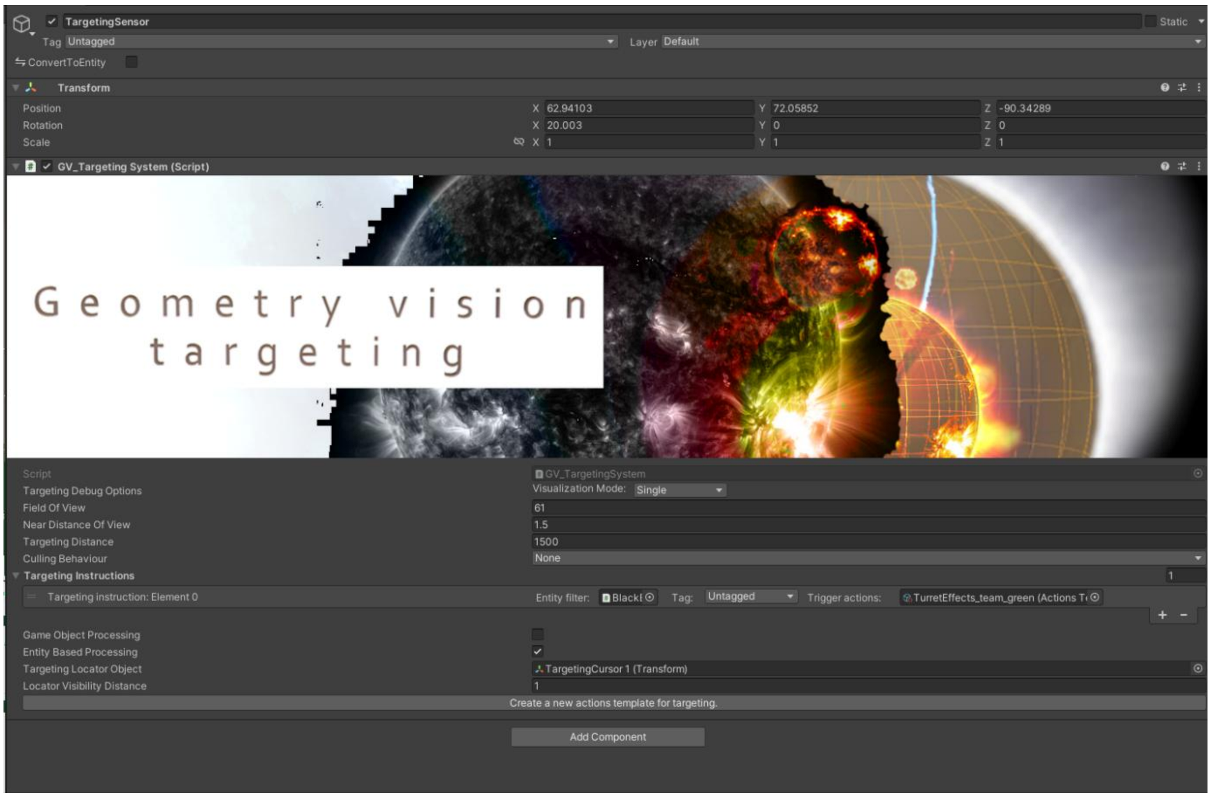
2. BASIC USAGE

2.1 GETTING STARTED IN THE EDITOR

You can find the full scripting/coding function,,, documentation from my website. The system also comes with tool tips that can help you understand the parameters.

2.1.1 Targeting system component

The easiest way to get started with the system is to add the system to an game object from the editor inspectors AddComponent menu.



TargetingSensor
Static
Tag Untagged
Layer Default
ConvertToEntity

Transform

	X	Y	Z
Position	62.94103	72.05852	-90.34289
Rotation	20.003	0	0
Scale	1	1	1

GV_Targeting System (Script)

Script	GV_TargetingSystem
Targeting Debug Options	Visualization Mode: Single
Field Of View	61
Near Distance Of View	1.5
Targeting Distance	1500
Culling Behaviour	None
Targeting Instructions	1
Targeting instruction: Element 0	Entity filter: BlackI Tag: Untagged Trigger actions: TurretEffects_team_green (Actions T
Game Object Processing	
Entity Based Processing	✓
Targeting Locator Object	TargetingCursor 1 (Transform)
Locator Visibility Distance	1

Create a new actions template for targeting.

Add Component

Figure 2: This how the component will look like(entities enabled). The view looks different, if the entities are disabled.

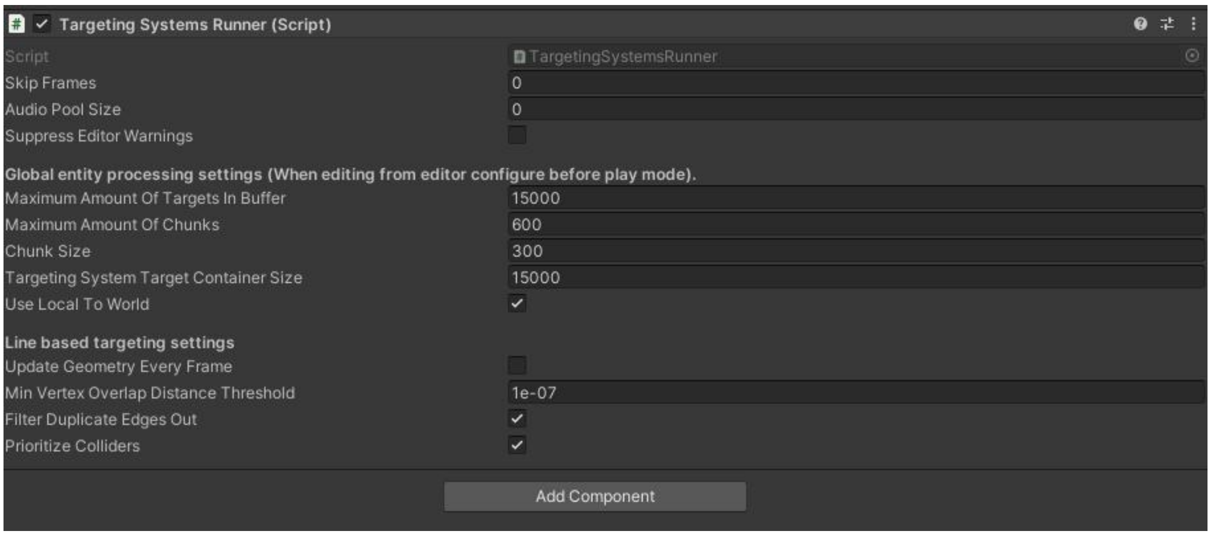
This will reveal the most important features. Entity options will only be available, if the entities are enabled from the player settings with the help of scripting symbols as covered in the entities installation section.

Parameter:	**Action:**
Targeting visualization options:	Will show targeting drawings in the editor.
Field of view:	How width the targeting visibility area is.
Near distance of view:	Closest distance to start getting target data. Useful if the sensor is inside an object, so you can add an this as an

	offset, so the object does not block the targeting.
Targeting distance:	Targeting range.
Culling behaviour:	If you don´t want targets behind colliders.
Targeting instructions:	Here you can define what gets targeted and also to add actions template, if you need to add effects. If you need to target multiple game objects that have different tags for example you can add 2 instructions where both have different tags.
Game object based processing:	If you don`t want the system to process game objects then toggle this off.
Entity object based processing:	Same, but for entities.
Targeting cursor:	You can add targeting cursor game object here. If it is a prefab, it will be instantiated. It can also be something from the scene.
Locator visibility distance:	The minimum distance when to start showing the cursor.

2.1.2 Targeting system runner component.

Runner is responsible for running global targeting jobs. Also gives user the ability to make global changes to targeting.



Targeting Systems Runner (Script)

Script	TargetingSystemsRunner
Skip Frames	0
Audio Pool Size	0
Suppress Editor Warnings	☐

Global entity processing settings (When editing from editor configure before play mode).

Maximum Amount Of Targets In Buffer	15000
Maximum Amount Of Chunks	600
Chunk Size	300
Targeting System Target Container Size	15000
Use Local To World	✓

Line based targeting settings

Update Geometry Every Frame	☐
Min Vertex Overlap Distance Threshold	1e-07
Filter Duplicate Edges Out	✓
Prioritize Colliders	✓

Figure 3: Targeting system runner component. Looks small and meaningless compared to the other component, but handles a lot of things behind the curtains.

Here is the parameter definitions. This area will be improved in the future. Hopefully getting rid of options that define maximums, without hurting the performance. Here below are some of the parameters. More info can be found from the tool tips of the parameters while in the Unity editor.

Parameter:	Action:
Skip frames:	If you want, you an use this to control update frequency.
Audio pool size:	How many game-objects/ audio-sources to build for the audio pool.
Suppress editor warnings:	In case you don´t like getting warnings from the system.
Chunk size:	Chunk size can be used as an optimization fine tuning tool. It controls the area size for chunk of targets. In large scenes it might be better to have bigger area and in smaller more dense scenes smaller.

Targeting System Container size	How many targets can one system hold. After exceeding this value there will be message like hash map full. This can be dodge by adjusting the level.
Use local to world	Entities only option. If you have your translations in Tranlation component. Then check this to false. The system will work its translations in the Translation component, otherwise LocalToWorld component will be used.
Update geometry every frame.	Normally geometry is cached, but in case shape is changed during runtime, then edges from mesh needs to be updated in order to get accurate results. If you do not have deforming meshes, it is highly recommended to keeps this to false for increased performance.

2.2 Coding

2.2.1 Quick start

To get closest target you need the targeting system that can be added to game object with the add component method.

To add targeting component to a game object:

```
gameObject.AddComponent<GV_TargetingSystem>();
```

To get closest target to view angle use:

//get targeting component

```
var targetingSystem = gameObject.GetComponen<GV_TargetingSystem>();
```

//use component to get target

```
var closestTarget = targetingSystem.GetClosestTarget(false);
```

To get closest target as game object:

// use component to get target
```
var closestTarget = targetingSystem.GetClosestTargetAsGameObject(false);
```

Tip: The forceUpdate parameter set to true will force the system to run. This is a lot slower than the false option. So only use it, if you need the updated results instantly. Similar to Destroy() vs. DestroyImmediately()

Target groups (targeting instructions):

Lets say you have an npc and you want it to start targeting air units instead of ground units or maybe you want to target both groups. This can be achieved by modifying or creating a new targeting instruction.

//Get the first targeting instruction and change type for entities
```
targetingSystem.targetingInstructions[0].SetCurrentEntityFilterType(typeof(RedEnemy));
```

//Get the first targeting instruction and change tag for gameObjects
```
targetingSystem.targetingInstructions[0].TargetTag = "RedEnemy";
```

2.2 Basic components and architecture

Targeting system is made of 2 main components. Targeting system component itself and targeting systems runner component. Runner is used to run the targeting systems and also to provide global parameters/options for the users.

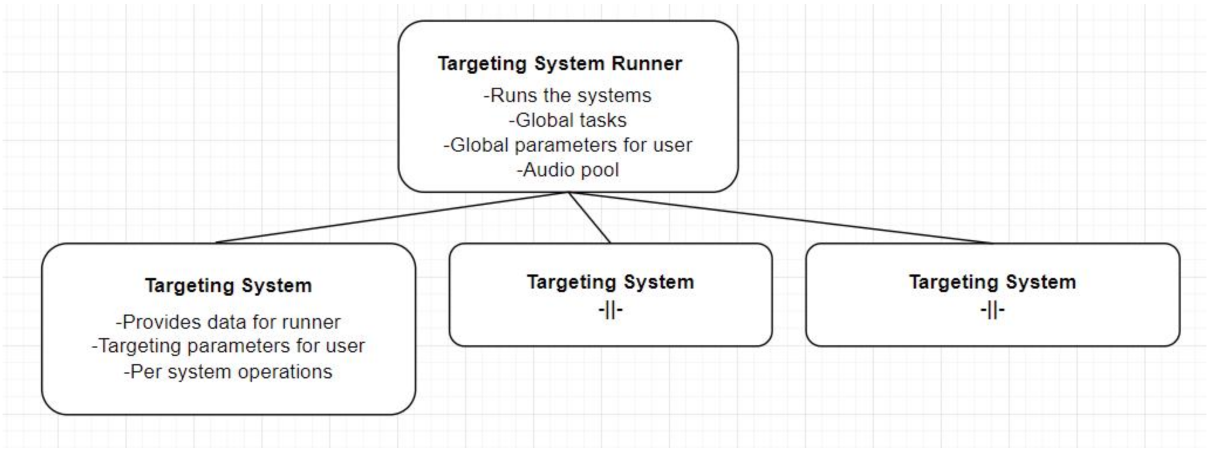


Figure 4: High level simplified architecture showing important components for the user.

3. Advanced usage

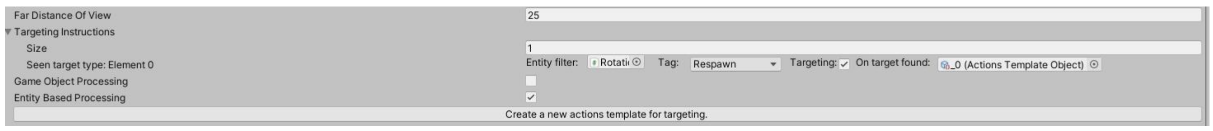
3.1 Culling.

Culling will make targeting system to check if there are obstacles in front of the target. By default Targeting system ignores layers 2 and 8. Ignore raycast layer is the layer that is ignored internally. So in case you want something to invisiboe to the system, then put it in the ignore ray cast layer or layer 8.

4. Actions template object (optional).

Action template object is used to spawn prefabs that can be used with the system to achieve various effects. The prefab conversion from template object happens when entering play mode. So no conversion is done on every time these effects are triggered. Also the template object can be shared with multiple targeting systems. You can create your own by clicking the button “Create new actions template for targeting” .

This will create a script able object that can be dragged to the slot inside targeting instruction

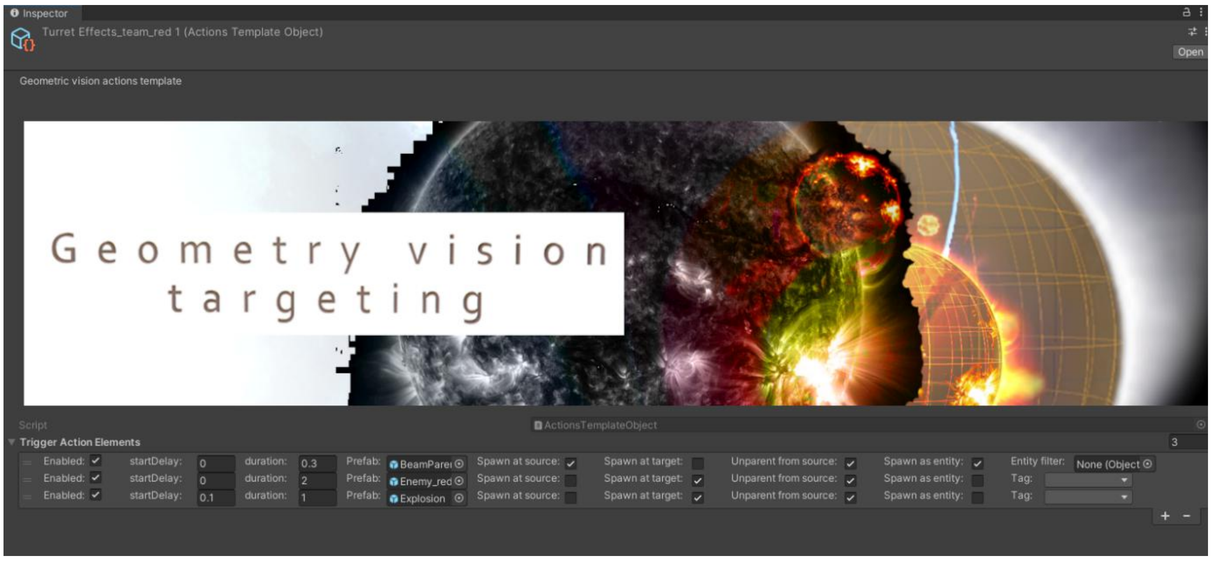


Far Distance Of View: 25
Targeting Instructions
Size: 1
Seen target type: Element 0 — Entity filter: Rotati — Tag: Respawn — Targeting: ✓ — On target found: _0 (Actions Template Object)
Game Object Processing: ☐
Entity Based Processing: ✓

Create a new actions template for targeting.

in the inspector UI.

When clicking the object, you´ll see various parameters to configure the template. It is recommended to look for more help on the example projects that use this feature. The actions template object can also recognize audio sources from prefabs parent object. If audio pool size is set to bigger than zero from the targeting systems runner object audio pool will be created and these audios will be played when prefabs are triggered.


Inspector
Turret Effects_team_red 1 (Actions Template Object)
Open
Geometric vision actions template
Geometry vision targeting

Script: ActionsTemplateObject
Trigger Action Elements: 3

Enabled: ✓	startDelay: 0	duration: 0.3	Prefab: BeamPare	Spawn at source: ✓	Spawn at target: ☐	Unparent from source: ✓	Spawn as entity: ✓	Entity filter: None (Object
Enabled: ✓	startDelay: 0	duration: 2	Prefab: Enemy_red	Spawn at source: ☐	Spawn at target: ✓	Unparent from source: ✓	Spawn as entity: ☐	Tag:
Enabled: ✓	startDelay: 0.1	duration: 1	Prefab: Explosion	Spawn at source: ☐	Spawn at target: ✓	Unparent from source: ✓	Spawn as entity: ☐	Tag:

Here is a quick overlook on the parameters.

- Delay is used to delay the spawn of the object.
- Duration is used to define how long the action object is alive on the world.
- Spawn at source will spawn at the targeting systems position
- Parenting works with game objects only currently. It will parent the spawned effect to the target game objects transform. This is useful if you want the effect to inherit targets position and rotation
- Spawn as entity will spawn the prefab as entity and enables filtering option.
- Filtering and tag can be used to give the spawned effect tag or entity filter. This is useful, if you want to hook your own systems or query these effects with Unity' s find game object with tag command. Example project "multiple turrets" missile launcher uses this feature.

By default, the targeting system does trigger the actions template. You will need the following code to do that

```
//Get targeting system component and trigger prefabs from the actions template object
 GetComponent<GV_TargetingSystem>().TriggerTargetingActions();
```

See example project for object picking on creative ways to utilise the system. Prefabs that are spawned can be anything and contain scripts as usual.

Example use case could be a weapon that has a starting effect like smoke from the gun barrel when shooting. When user switches weapon so is the template.

5. Trouble shooting

1.1 Type or namespace TargetingSystem not found.

In case you are using asmdef files. In order for Unity to find the plugin it needs an assembly definition reference(asmdef).

1. Go to your folder and create asmdef file.

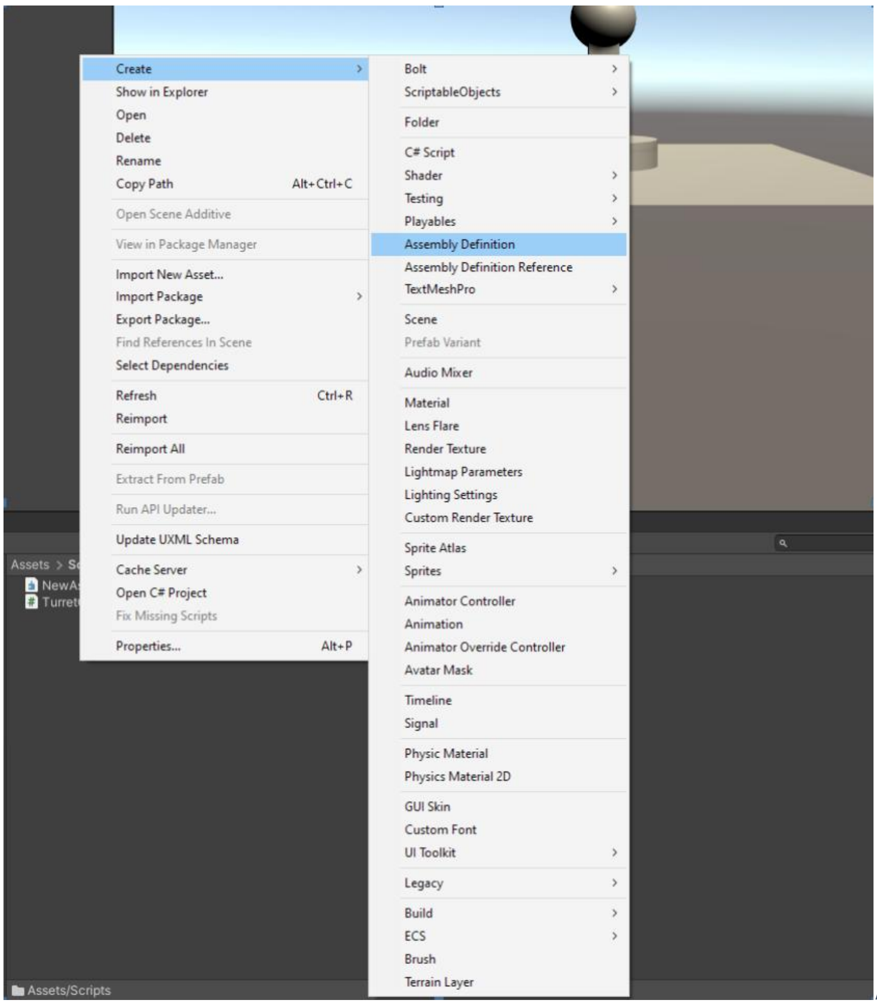
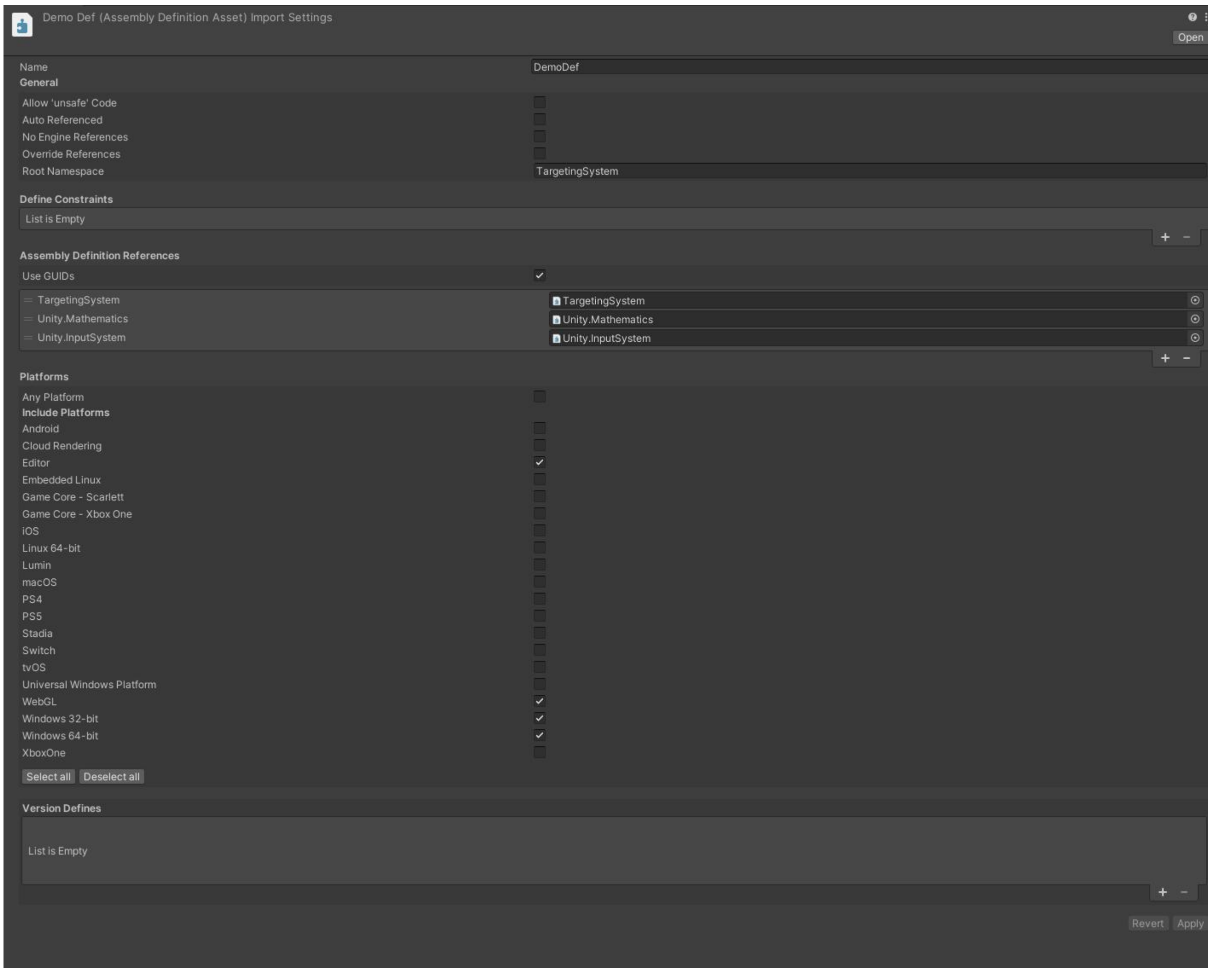


Figure 7: Image showing how to create asmdef file

2. Then reference the plugin TargetingSystem asmdef file.

This will connect your assembly to the plug-in. So your development environment can recognize it. Drag TargetingSystem.asmdef to just created asmdef files references slot like in the following image.


Demo Def (Assembly Definition Asset) Import Settings

Name: DemoDef

General

Allow 'unsafe' Code	☐
Auto Referenced	☐
No Engine References	☐
Override References	☐
Root Namespace	TargetingSystem

Define Constraints

List is Empty

Assembly Definition References

Use GUIDs ✓

TargetingSystem	TargetingSystem
Unity.Mathematics	Unity.Mathematics
Unity.InputSystem	Unity.InputSystem

Platforms

Any Platform ☐

Include Platforms

Android	☐
Cloud Rendering	☐
Editor	✓
Embedded Linux	☐
Game Core - Scarlett	☐
Game Core - Xbox One	☐
iOS	☐
Linux 64-bit	☐
Lumin	☐
macOS	☐
PS4	☐
PS5	☐
Stadia	☐
Switch	☐
tvOS	☐
Universal Windows Platform	☐
WebGL	✓
Windows 32-bit	✓
Windows 64-bit	✓
XboxOne	☐

Version Defines

List is Empty

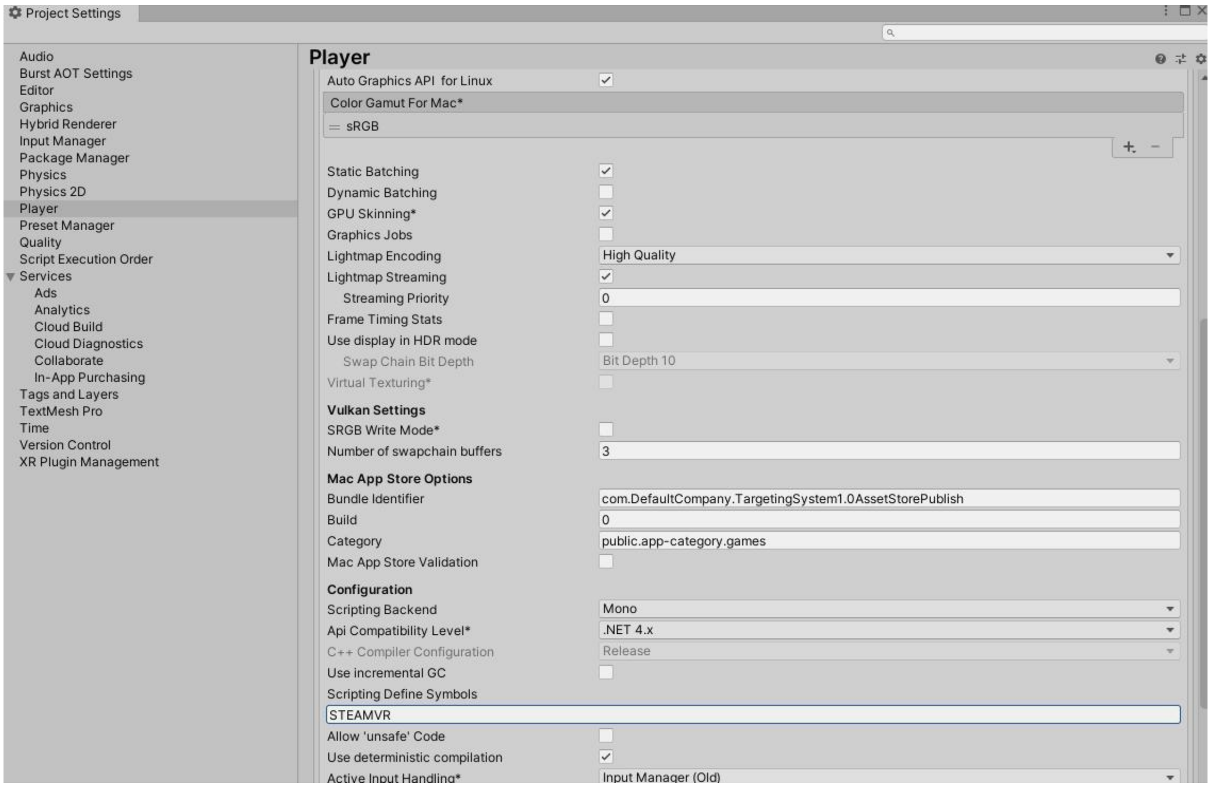
Then hit apply. Your development environment folder should now be connected to the plugin.

You should now be able to import the targeting system on to your scripts.

```
using Plugins.GeometricVision.TargetingSystem.Code.MainClass;
```

6. Virtual reality devices

To get the virtual reality working you need the steam vr plugin that should also work with Oculus and add the key word STEAMVR into Player settings. Look for Scripting define assemblies as in the following image.



Player

Auto Graphics API for Linux	✓
Color Gamut For Mac*	
sRGB	
Static Batching	✓
Dynamic Batching	
GPU Skinning*	✓
Graphics Jobs	
Lightmap Encoding	High Quality
Lightmap Streaming	✓
Streaming Priority	0
Frame Timing Stats	
Use display in HDR mode	
Swap Chain Bit Depth	Bit Depth 10
Virtual Texturing*	
Vulkan Settings	
SRGB Write Mode*	
Number of swapchain buffers	3
Mac App Store Options	
Bundle Identifier	com.DefaultCompany.TargetingSystem1.0AssetStorePublish
Build	0
Category	public.app-category.games
Mac App Store Validation	
Configuration	
Scripting Backend	Mono
Api Compatibility Level*	.NET 4.x
C++ Compiler Configuration	Release
Use incremental GC	
Scripting Define Symbols	
STEAMVR	
Allow 'unsafe' Code	
Use deterministic compilation	✓
Active Input Handling*	Input Manager (Old)

Figure 8: Image from older unity showing location where scripting define symbols can be found.

If you have not installed steam-vr it will complain about missing references to Valve. So, in case you have not installed, you' ll need the Steam-vr package from the assetstore or git. I think their git repository is more up to date.
https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647

If you are adding the targeting system to the camera then you should create an empty game object and add Targeting system component to that instead. After that add that game object as a child for the camera containing game object. Plugin has an inspector override to hide camera component of the game object it is added to. So, in case you want to see your camera do as recommended.

7. CONTACT

-” support@mikael-korpinen.com” or “mikael.korpinen@gmail.com”

-Discord channel is the best place to get fast help, because usually there are also other helpful people out there. Join link. https://discord.gg/QtHTQ4J