



# Proteus VR Template

**First release: March 2016 (under the name "SteamVR Template")**

**Latest Version 5.1 (under the name "Proteus VR Template")**

**Latest release February 15, 2019**

## What can I do with that, that I cannot do with the "VR Template" in Unreal?

Well, rapidly: It's single or multiplayer (LAN or Oculus Home); It uses Oculus Avatars (or not); It uses voice over IP (VOIP); full of nice features fully using Oculus functions; compatible Go/Quest/Rift; minimum use of C++ (but still some, since some methods not possible only in blueprints); hey, we were the first to propose a template back in 2016!

**FOR NOOBS:** This is super simple. Even if you think Visual Studio is the name of a VFX/Hollywood startup somewhere around LA with people working in open spaces and flip-flops, it requires few manipulations to set up; I could add few instructions how-to on demand.

This template consists of virtual reality compatible pawn and settings, ready to drop in your single or multiplayer app. So what you can do is choose the features you want, and build yourself/modify a pawn based on the functionalities you need.

Discussion on Unreal forum at <https://forums.unrealengine.com/development-discussion/vr-ar-development/106631-single-multiplayer-touch-vive-proteus-blueprint-only-template>

Download it at <https://github.com/ProteusVRpublic/ProteusTemplate>

## Compatible with the following Unreal Engine versions:

Unreal Engine 4.20\*, Oculus-branch, latest commit e8c8be1 (Nov 14, 2018), found at <https://github.com/Oculus-VR/UnrealEngine/tree/4.20>

Unreal Engine 4.21\*, Oculus-branch, latest commit 91f23fa (Feb 12, 2019), found at <https://github.com/Oculus-VR/UnrealEngine>

\* minor changes to the engine possibly / mandatory required



## What's new in version 5.1

- Custom hand poses, both for off-line and on-line (Oculus) Avatars
- Both off-line and on-line hands now share the Oculus Avatars skeleton mesh

## Features of version 5.0

- Entirely redesigned
- More stable
- Use latest Oculus implementations (splash screen, play area limits, smart use of camera fade, etc).
- Unreal 4.20 & 4.21
- Unreal 4.20: Uses Oculus Platform SDK 1.24, OVR Plugin 1.31, Audio SDK 1.18
- Unreal 4.21: Uses Oculus Platform SDK 1.34, OVR Plugin 1.32, Audio SDK 1.34
- Uses Oculus Avatar SDK 1.31
- Single & Multiplayer Oculus Avatars
- Compatible Oculus Rift & Oculus Go, Oculus Quest-ready

**To install as a template**, just unzip into the appropriate templates directory like C:\Program Files\Unreal Engine[Version]\Templates for launcher version or[ForkLocation]\UE4\Templates for source version. Launch a new project, and you'll find it in the blueprint section.

- **To open as a project file**, open the project with the launcher or directly from the .uproject file.

### TO USE IT:

- 1) Download the Unreal Engine .zip file from github (or sync through github app)
- 2) Unzip
- 3) Setup.bat
- 4) GenerateProjectFiles.bat
- 5) Open UE4.sln with Visual Studio
- 6) Make the modifications needed to the engine, as specified in pdf document
- 7) Build UE4
- 8) Download the latest Proteus Template .zip from github (or sync through github app)
- 9) Right-mouse click on Proteus.uproject, Switch Unreal Engine Version, for correct engine version
- 10) Open Proteus.sln with Visual Studio
- 11) Make changes to the ProteusOvrAvatar.cpp if using Unreal 4.21, as specified in pdf document
- 12) Build Proteus
- 13) Now ready to open Proteus.uproject



## How can I install it in my project?

- 1) Compile correct Unreal Engine version
- 2) Bring proper modifications to the engine (see page 16), if:
  - a. You want to enable Retrieve Oculus ID and Verify Entitlement with the Oculus Go / Quest
  - b. You want to nativize assets on the Oculus Go / Quest
  - c. You use 4.21 version
- 3) Compile app and let's go!

### In the template

- Open "Proteus.uproject"
- Inside the editor, right-click on the folder "Proteus\_Multi" and select Migrate
- Migrate everything inside the folder to your project, inside the folder your project/Content/

### In your project

- If needed, adjust Project settings (see "[What are the best Project Settings for VR?](#)")
- In ProjectSettings/Maps & Mode (see page 10):
  - Ensure that Game Instance Class/GameInfoInstance is selected
  - Select entryMap as Editor Starting Map and Game Default Map
  - Select MainMenuGM as GameMode
- Don't forget to put a navmesh bound volume to allow teleportation!
- Pawn will spawn at playerstart (placed on the floor) and will teleport on navmesh
  - Put a PlayerStart tag in entryMap level: PS1
  - In the all other maps: PS1, PS2, PS3, PS4
- In the control panel found in MainMenuPC, select your options



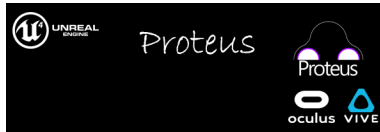
## What is the default input mapping for the Oculus Rift / Quest Touch Controllers?

- **Both controllers**
  - Thumbstick directions / Controller orientation: Play area rotation before teleportation
  - Index Trigger: Grab / Release
- **Right Controller**
  - A Button: Teleport
  - B Button: Widget Interaction
  - Thumbstick Button: Spawn Menu on/off
- **Left Controller**
  - X Button: Teleport
  - Y Button: Guardian corners on/off
  - Thumbstick Button: Spectator Mode on/off

## What is the default input mapping for the Oculus Go?

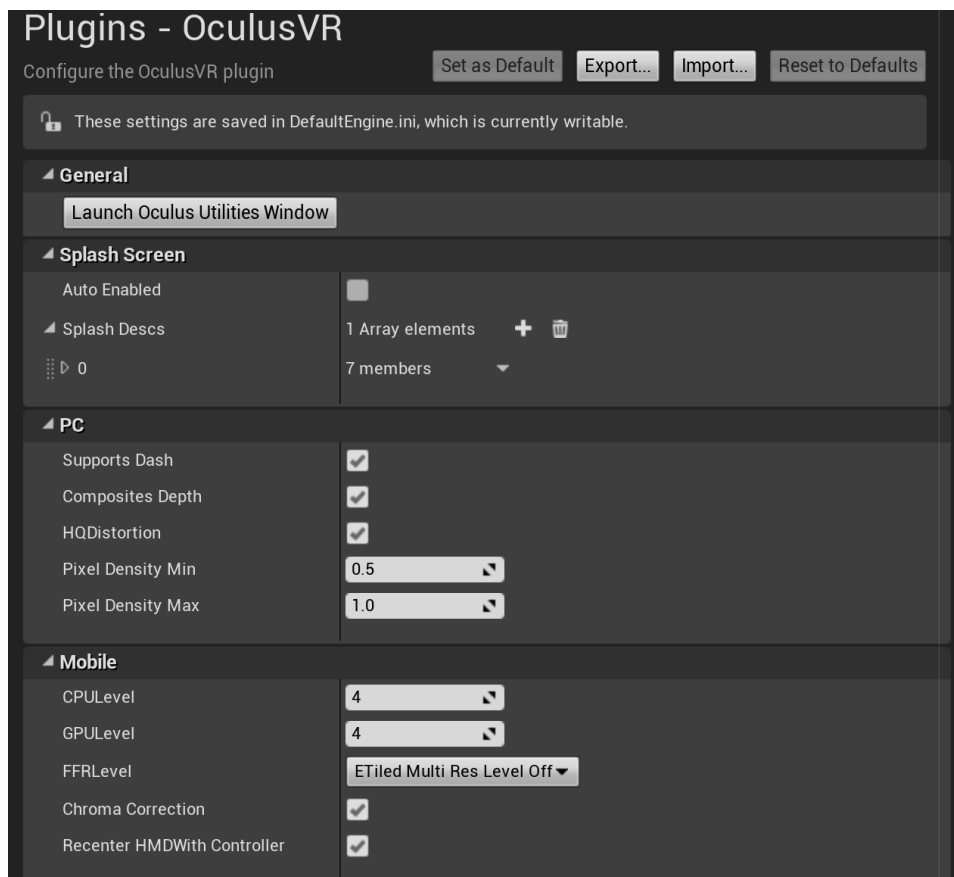
They are not mapped, but they correspond to:

- Oculus Go Home Button: Same as (Right) Touch A Button
- Oculus Go Trackpad (x-y axis): Same as Right Touch Thumbstick x-y axis
- Oculus Go Trackpad Button: Same as Right Touch Thumbstick Button
- Oculus Go Trigger: Same as Right Touch Trigger
- Oculus Go Back Button: Reserved

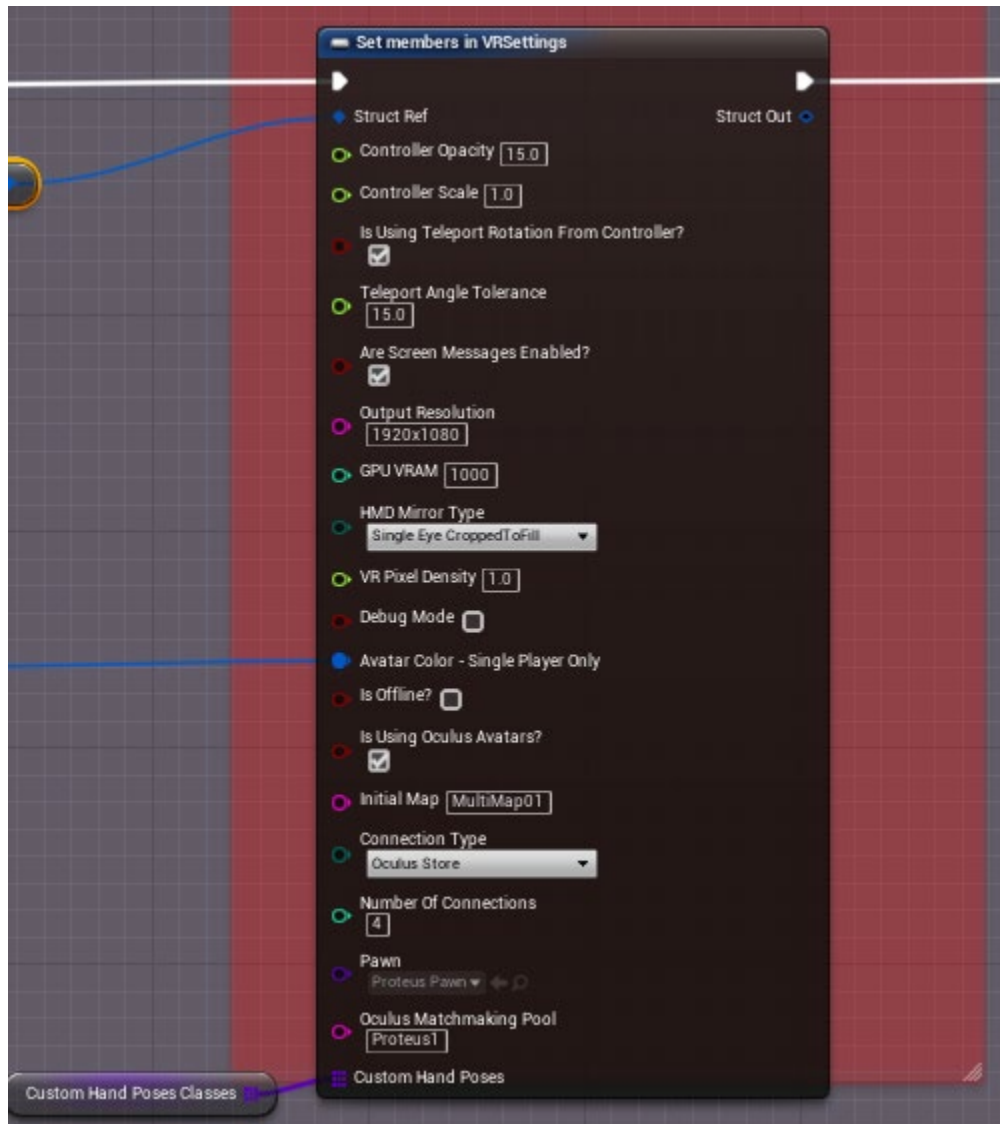


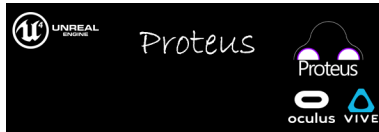
## What are the best Project Settings for VR?

- **You will find the best settings for Rift/Go already set in ProjectSettings**
- Go / Quest
  - works both in OpenGL ES2 and ES 3.1, use Android SDK 21
  - Oculus Go is a 32-bits device, so use armeabi-v7a
  - Use Mobile Multi-View, Mobile Multi-View Direct, Mask-base Foveated-Rendering
  - Avoid Monoscopic Far Field (crash – BUG)
  - Mobile HDR off
  - Occlusion Culling off (bug)
- Don't forget the settings in Project Settings:



## The Control Panel

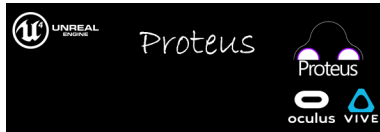




Setting	
Controller Opacity	0 up to whatever, 1 is not entirely opaque, just 15 should be fine
Controller Scale	For the hands, the scale is 1:1 with Epic VR template. Reduce scale to approx. 0.75 to come close to real-life for the hands, 0.93 to Oculus Avatar Hands
Teleport Rotation from Controller	If unselected, rotation is from thumbstick / trackpad
Teleport Angle Tolerance	Safety feature to teleport only on surfaces under a certain inclination. Between 0 and 90 degrees. Put 90 degrees to disable it.
Enable Screen Messages	Yes / No (in development and editor mode)
Output Resolution	The resolution on your monitor.
GPU VRAM	GPU Video Memory, can be useful to tweak to stream large textures, by default 4000 (MB). You should put it around 1GB under your GPU VRAM
HMD Mirror Mode	See below
VR Pixel Density	Use a higher number for better quality, and a lower one for better performance
Debug Mode	Enable debug mode (no VR – see below)
Avatar Color	Avatar color when not using the Oculus Avatars
Is Offline?	You will be logged in but will not accept any connections
Is Using Oculus Avatars?	Use in-house hands+head or Oculus Avatars system
Initial Map	The map you'll load into following entryMap
ConnectionType	LAN / Oculus Home. You need to be connected to Oculus Home to use the Oculus Avatars
Oculus Matchmaking Pool	Oculus Matchmaking Pool, as set in Developer Dashboard
Custom Hand Poses	See Custom Hand Poses section

## HMD Mirror Modes

Mode	Oculus Rift	SteamVR	PS VR	Notes
Disabled	✓	✓		For the best performance on HMDs, this mode disables Spectator Screen output.
SingleEyeLetterboxed	✓	✓	✓	This mode is primarily intended for debugging purposes, showing only one letterboxed eye on the screen.
Undistorted	✓	✓	✓	This is a debug mode, showing the entire rendered area for both eyes.
Distorted	✓			This mode is only supported by Oculus. Specifically, this is an Oculus specific debug mode, showing chromatic aberrations, etc.
SingleEye	✓	✓	✓	Much like the <b>Undistorted</b> mode, this a debug mode for only one eye. Because this mode stretches the scene, it may be useful for identifying small artifacts in the scene.
SingleEyeCroppedToFit	✓	✓	✓	This mode crops the eye to fill the entire screen.



## What is the "Debug Mode"

In debug mode, you can control the pawn with the keyboard, without HMD. You can also test multiplayer with a single computer (see below). Controls are

Left – Right Arrows	Turn left / Turn Right (simulate HMD rotation)
Up – Down Arrows	Go forward / Go back (simulate moving in room-scale)
W,A,S,D:	Move along 2D X,Y axis (simulate HMD location)
Left Shift	Simulate Right Index Trigger / Trigger
Left Control	Simulate Right Hand Trigger / Grip
Left ALT	Simulate B Button / TrackPad Right
Spacebar	Simulate A Button / TrackPad Up
Z	Simulate R Thumbstick Pressed / R Trackpad Pressed

To simulate multiplayer on a single computer, use debug mode with LAN, launch it in Stand-Alone Game mode with:

The screenshot shows the 'Play in Standalone Game' and 'Multiplayer Options' sections of the Unreal Engine interface. The 'Multiplayer Options' section is expanded, showing settings for a LAN game. The 'Number of Players' is set to 2, and the 'Server Port' is 17777. The 'Editor Multiplayer Mode' is set to 'Play As Listen Server'. The 'Multiplayer Window Size' is set to 640x480 pixels. The 'Play on Device' section is also visible, with 'Build Game Before Launch' set to 'If project has code, or running a locally built editor' and 'Auto Compile Blueprints on Launch' checked.

Section	Option	Value
Play in Standalone Game	Command Line Options	
	Additional Launch Parameters	
	Disable Sound (-nosound)	<input type="checkbox"/>
Multiplayer Options	Number of Players	2
	Server Port	17777
	Server Game Options	
	Run Dedicated Server	<input type="checkbox"/>
	Use Single Process	<input type="checkbox"/>
	Editor Multiplayer Mode	Play As Listen Server
	Command Line Arguments	
	Multiplayer Window Size (in pixels)	Window Width: 640, Window Height: 480
	Create Audio Device for Every Player	<input type="checkbox"/>
	Play on Device	Build Game Before Launch
Launch Configuration		Same as Editor
Auto Compile Blueprints on Launch		<input checked="" type="checkbox"/>





# MULTIPLAYER MODE

## IMPORTANT

- Select ProjectSettings/NavigationSystem/Allow Client Side Navigation

- Don't forget to put

`[/Script/Engine.GameSession]`

`bRequiresPushToTalk=false`

in DefaultGame.ini

- LAN MODE:

- Use the DefaultEngine.ini found in /Config/LAN

- Specific configuration for LAN is:

`[OnlineSubsystem]`

`DefaultPlatformService=NULL`

`bHasVoiceEnabled=true`

`PollingIntervalInMs=20`

`VoiceNotificationDelta=0.2`

`AllowPeerConnections=True`

`AllowPeerVoice=True`

`[/Script/Engine.GameEngine]`

`!NetDriverDefinitions=ClearArray`

`+NetDriverDefinitions=(DefName="GameNetDriver",DriverClassName="/Script/OnlineSubsystemUtils.IpNetDriver",DriverClassNameFallback="/Script/OnlineSubsystemUtils.IpNetDriver")`

- Oculus Home can be logged in or out

- OCULUS HOME MODE

- Use the DefaultEngine.ini found in /Config/Oculus

- Specific configuration for OCULUS HOME is:

`[OnlineSubsystem]`

`DefaultPlatformService=Oculus`

`bHasVoiceEnabled=true`



PollingIntervalInMs=20

[OnlineSubsystemOculus]

bEnabled=true

RiftAppId=YOUR RIFT APP ID

GearVRAppId= YOUR GEAR VR / GO APP ID

[/Script/OnlineSubsystemOculus.OculusNetDriver]

NetConnectionClassName=OnlineSubsystemOculus.OculusNetConnection

[/Script/Engine.GameEngine]

!NetDriverDefinitions=ClearArray

+NetDriverDefinitions=(DefName="GameNetDriver",DriverClassName="OnlineSubsystemOculus.OculusNetDriver",DriverClassNameFallback="OnlineSubsystemUtils.IpNetDriver")

- Login to Oculus Home
- The Oculus Rift will use the RiftAppId found in Developer Dashboard FOR RIFT
- The Oculus Go will use GearVRAppID found in the Developer Dashboard FOR MOBILE
- Set the Matchmaking Mode in the Developer Dashboard to:



## View/Edit Matchmaking Pool

TestAvatar -

Pool Key <sup>?</sup>

Proteus1

Mode <sup>?</sup>

Browse

Users per Match <sup>?</sup>

Min Users

2

Min Preferred Users

2

Max Preferred Users

4

Max Users

4

Skill Component <sup>?</sup>

None

Should Consider Ping Time? <sup>?</sup>

☐ Yes

☒ No

[Show Advanced Tuning...](#)

Cancel

Save & Deploy

## JOINING A GAME

Spawn the Menu, refresh, select game.

# Custom Hand Poses

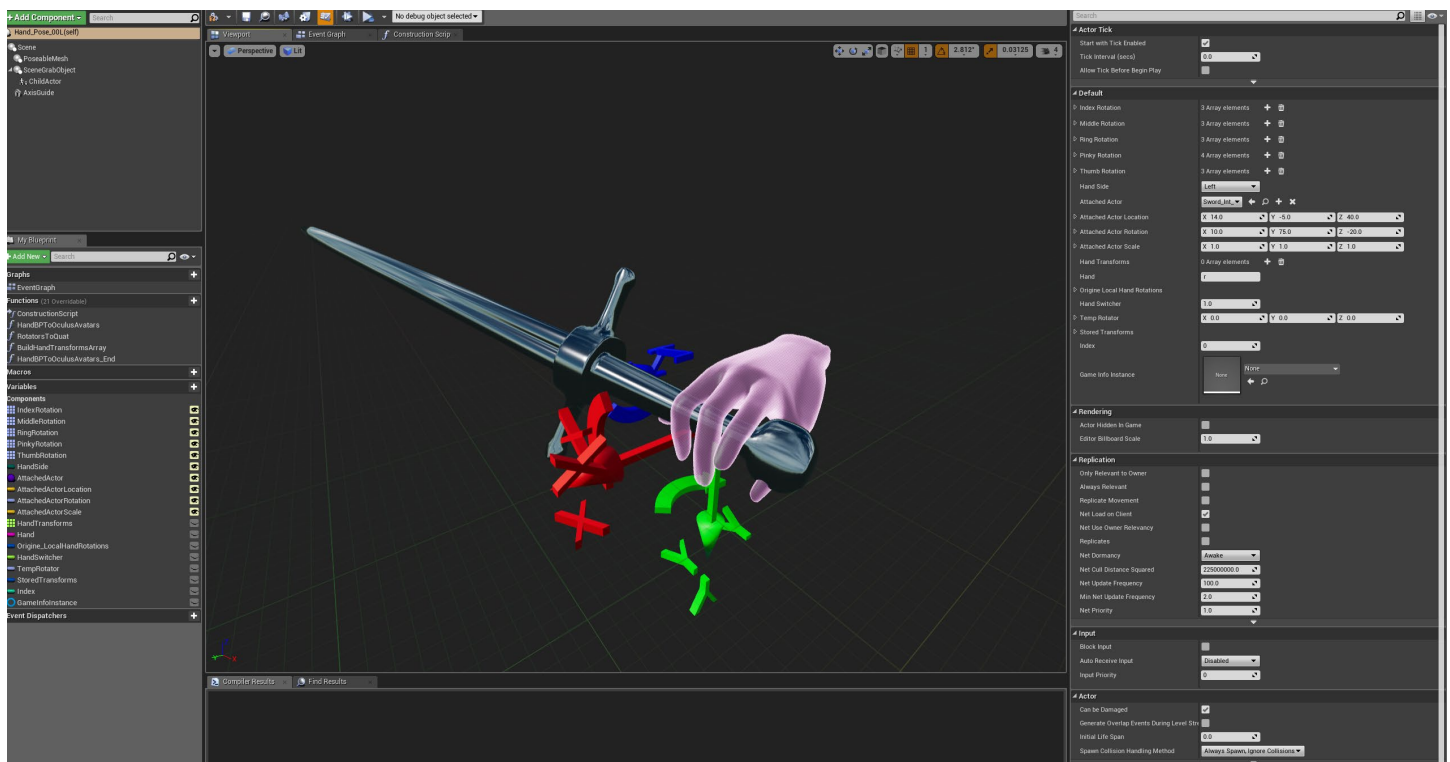
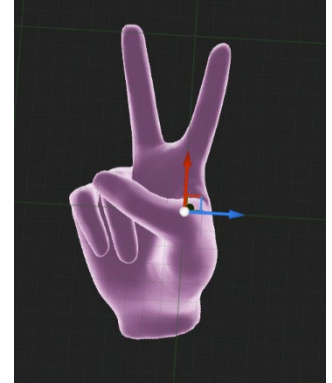
**This works with both off-line (custom) and on-line (Oculus) Avatars.**

Here's how to do it:

In Proteus Multi/Hand Poses, duplicate BlankPose actor blueprint, and make 2 copies for each pose (left and right hand)

I suggest to rename them left hand with even numbers, right hand with odd numbers, starting at 0. This will be also their index.

Inside HandPose:



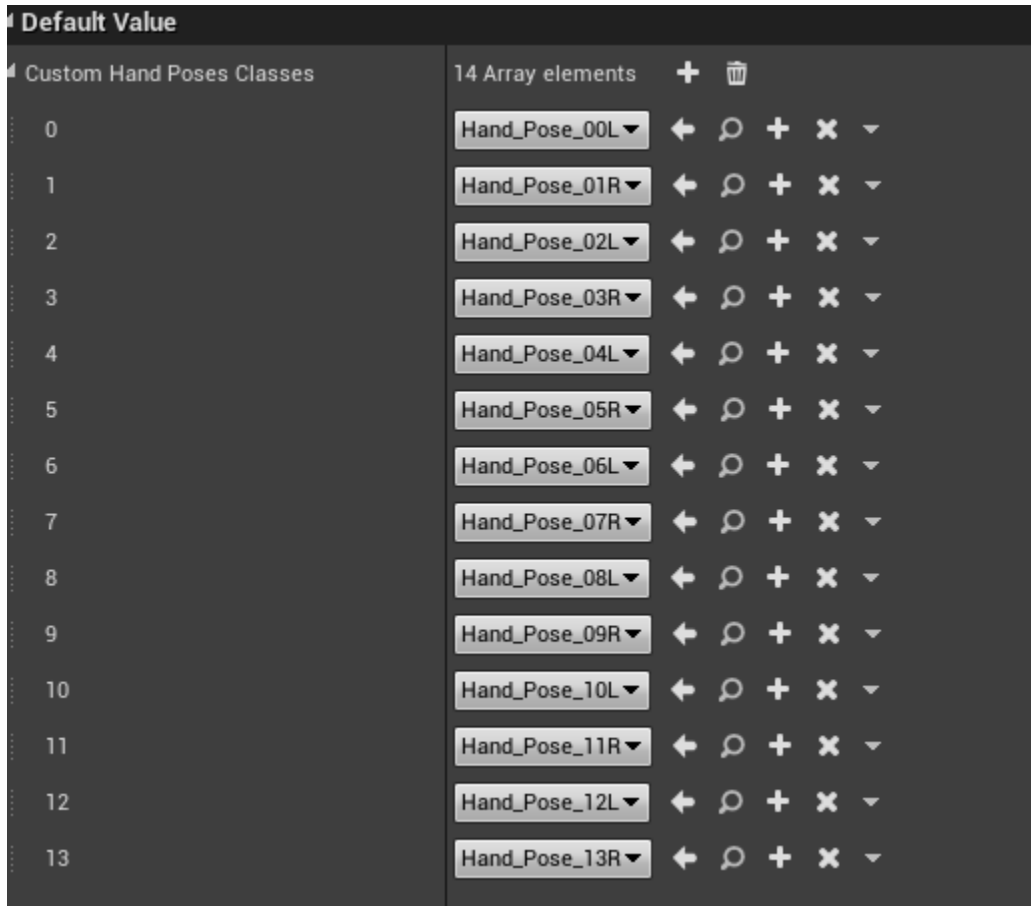
Change hand (l/r), fingers rotations, object reference (if any) to grab, object placement, object rotation. Do that for each hand.

You should only change these variables:

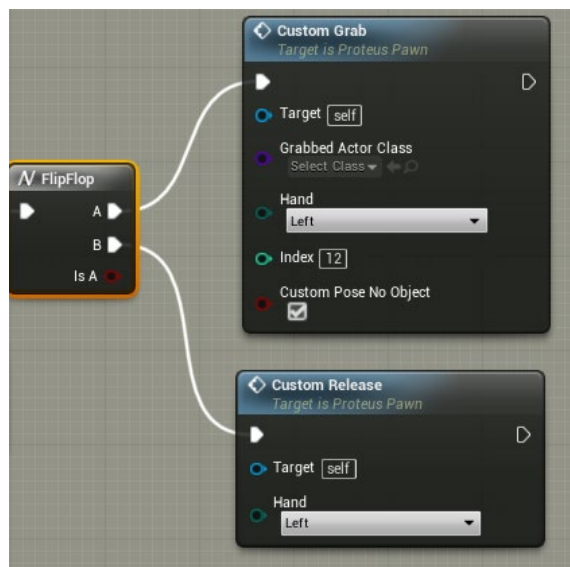
Index Rotation	3 Array elements	+	🗑️
▷ 0	X 10.0	Y 10.0	Z 0.0
▷ 1	X 40.0	Y 0.0	Z 0.0
▷ 2	X 0.0	Y 0.0	Z 0.0
Middle Rotation	3 Array elements	+	🗑️
▷ 0	X 10.0	Y 0.0	Z 0.0
▷ 1	X 20.0	Y 0.0	Z 0.0
▷ 2	X 10.0	Y 0.0	Z 0.0
Ring Rotation	3 Array elements	+	🗑️
▷ 0	X -10.0	Y 5.0	Z 0.0
▷ 1	X 0.0	Y 0.0	Z 0.0
▷ 2	X 10.0	Y 0.0	Z 0.0
Pinky Rotation	4 Array elements	+	🗑️
▷ 0	X 30.0	Y 0.0	Z 0.0
▷ 1	X -40.0	Y 40.0	Z 0.0
▷ 2	X 20.0	Y 0.0	Z 0.0
▷ 3	X 0.0	Y 0.0	Z 0.0
Thumb Rotation	3 Array elements	+	🗑️
▷ 0	X 0.0	Y 30.0	Z -20.0
▷ 1	X 0.0	Y 0.0	Z 0.0
▷ 2	X 0.0	Y 0.0	Z 0.0
Hand Side	Left		
Attached Actor	Sword_Int_		
Attached Actor Location	X 14.0	Y -5.0	Z 40.0
Attached Actor Rotation	X 10.0	Y 75.0	Z -20.0
Attached Actor Scale	X 1.0	Y 1.0	Z 1.0

Set Attached Actor Scale at the same scale as your desired attachment actor to place the pose correctly

When satisfied, add the poses inside the array in MainMenuPC/HandPoses. Place them in order of their index number (even-left, followed by odd-right, etc).



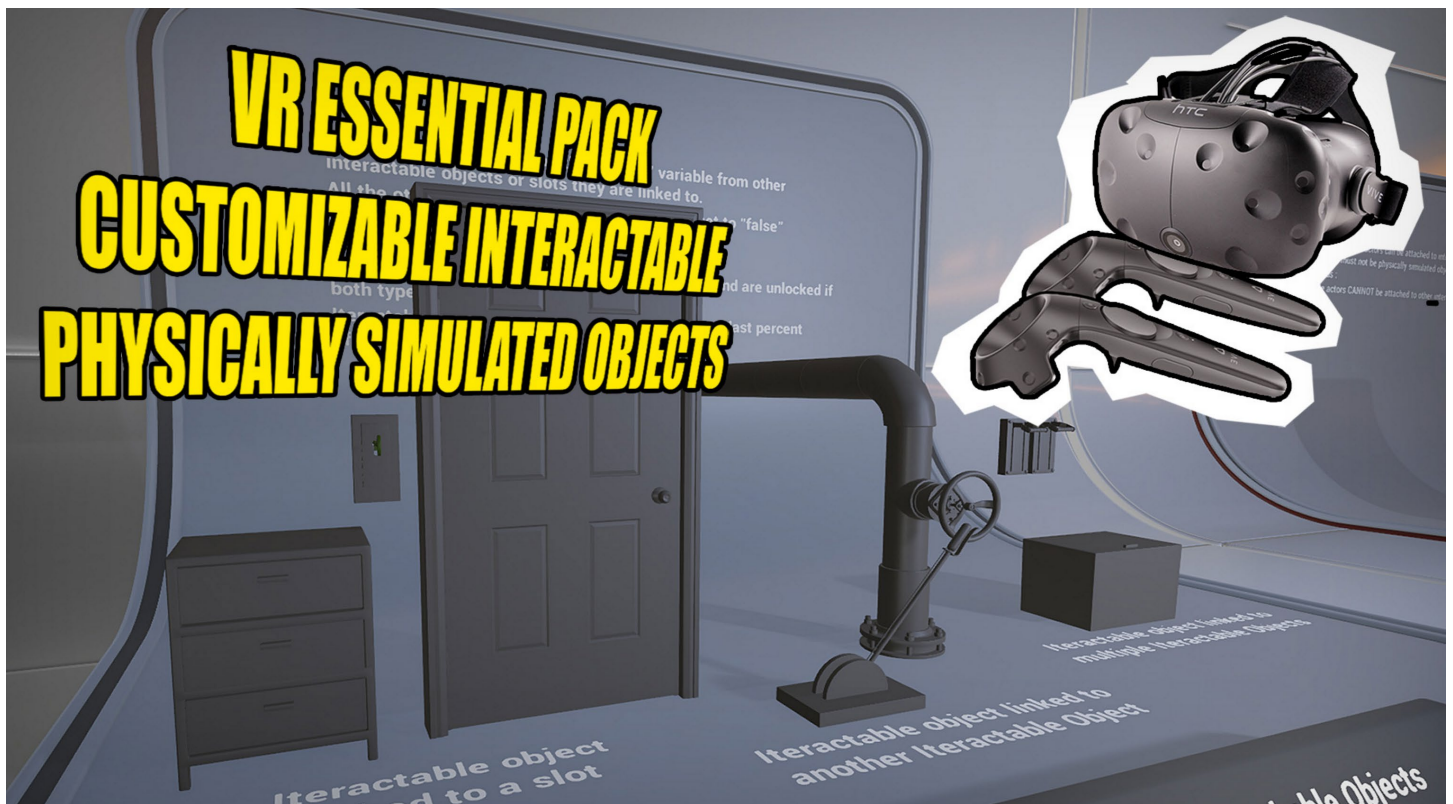
Finally, if you created handposes without an object reference (again one for each hand), you can assign them by their index number to an input in ProteusAvatar/Inputs. As an example, we put a peace sign pose at index 12 for left, 13 for right. Don't forget to check "Custom Pose No Object". That's it.



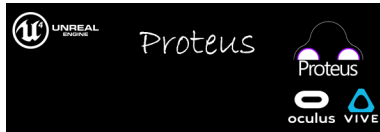
## Interacting with objects

You can do that many ways. To try many different type of interactions, @jamis's VR Essential Kit Map has been included with the template. The reason that it is included is that minor changes were made to @jamis' blueprints to be compatible with the AvatarMaster pawn. You'll find more infos at

<https://forums.unrealengine.com/showthread.php?131379-WIP-VR-Essential-Kit&highlight=vr+essential+kit>



- Note that the VR Essential Kit is a project made by @jamis. All questions concerning this project should be directed to him.



# Teleporting

Pawns are able to teleport on surfaces when these 2 conditions are met:

- Has a NavMeshBoundsVolume;
- Is below the teleportation angle limit (see VR settings) – put 90 to this setting to disable it

**IMPORTANT: Enable Project Settings / Engine / Navigation System / Allow Client Side Navigation for multiplayer teleportation; if not selected, clients won't be able to detect the Navigation Mesh.**





# What are the modifications to bring to the engine? Why on earth?

## Modification #1 (Unreal 4.20 AND 4.21)

**To be able to retrieve Oculus ID and verify Entitlement with the Oculus Go and Quest (Rift is OK)**

change

`\Engine\Plugins\Online\OnlineSubsystemOculus\Source\Private\OculusIdentityCallbackProxy.cpp`

Line 26

from:

`DelegateHandle = Online::GetIdentityInterface()->AddOnLoginCompleteDelegate_Handle(`

to:

`DelegateHandle = OculusIdentityInterface->AddOnLoginCompleteDelegate_Handle(`

Reference: <https://github.com/Oculus-VR/UnrealEngine/pull/26>

## Modification #2 (Unreal 4.20)

**To be able to Nativize assets on the Oculus Go and Quest**

change

`\Engine\Source\Runtime\Engine\Classes\GameFramework\PlayerController.h`

line 1053 private to public, because 'PlayDynamicForceFeedback' is a private member of 'APlayerController', for reasons unknown

Reference: <https://answers.unrealengine.com/questions/831405/playdynamicforcefeedback-node-in-420-fails-to-cook.html>

## Modification #3 (Unreal 4.21)

**Mandatory Modification to use the Proteus Avatar Plugin**

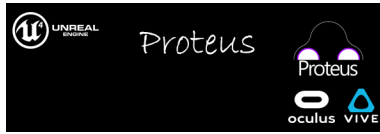
change

`\Engine\Source\Runtime\Engine\Classes\Components\PoseableMeshComponent.h`

Line 78 add (line after `friend class UOvrAvatar;`)

`friend class UProteusOvrAvatar;`

reference: <https://github.com/EpicGames/UnrealEngine/commit/2a9dbb5bdc8ee855a1a0fdb752966f4e3821895a>



## Modifications to bring to the plugin (Unreal 4.21 only)

### Mandatory Modification to use the Proteus Avatar Plugin

Uncomment the 3 following lines in

`\ProteusAvatars\Public\ProteusOvrAvatar.cpp`

Line 569

```
////////// ONLY FOR UNREAL 4.21 depthMesh->MarkRefreshTransformDirty();
```

Line 633

```
////////// ONLY FOR UNREAL 4.21 depthMesh->MarkRefreshTransformDirty();
```

Line 809

```
////////// ONLY FOR UNREAL 4.21 mesh.MarkRefreshTransformDirty();
```

reference: <https://github.com/EpicGames/UnrealEngine/commit/2a9dbb5bdc8ee855a1a0fdb752966f4e3821895a>



**As the world is not perfect, there's still bugs!!! They are:**

**Oculus Rift: There is no voice visualization with the Avatars!**

- No Voice visualization for Oculus Avatars; should be updated soon by Oculus (FUTURE FEATURE)

**Oculus Rift, Go, Quest : You don't use Stereo Layers?**

- We are not using Stereo Layers, because they cannot be destroyed !!! (BUG)

**Oculus Go / Quest**

- Occlusion culling doesn't work!!!!
  - Yes (BUG)
- Be gentle on the tick events, as you can easily crash the device
- Oculus Go cannot use Render Targets (i.e. the mirror)
  - That's life.
- Oculus Avatars
  - FOR NOW, ONLY THE BLANK AVATAR IS LOADED
    - Yes (FUTURE FEATURE)
  - INPUTS NOT REGISTERED WITH THE GO REMOTE
    - Yes (FUTURE FEATURE)
  - USE OPENGL ES3.1 TO COMPILE MOBILE AVATAR MATERIALS

**What's coming next in the next version:**

Oculus Avatars (Rift + Go): Set Custom hand poses / use hand sockets

Oculus Avatars (Rift): See Avatar head in mirror

Oculus Avatars (Go/Quest): Load correct mesh

**And after that...**

Oculus Avatars: Compatibility with SteamVR, Vive Controllers and Vive Knuckles

General template: Blueprints for Mixed-Reality & Oculus 3D Sound



# Q&A

## **Why did you not use the launcher version of UE4?**

- 1) Because the launcher version is always behind the Oculus branch for using latest Oculus SDK implementations;
- 2) Because it's always better to use UE4 source-based, to be able to make some changes to the engine;
- 3) Because the Oculus-branch version has some nice features, i.e. the Project Settings Menus

## **Where are the Windows MR / SteamVR functions?**

They will be back, but we targeted Oculus Go / Rift / Quest for this release.

## **You don't use the integrated Oculus Avatar plugin?**

You are such a detective. No. Because the official one is only single player, and we brought all sort of improvement to it, including multiplayer. So we packaged the improved/modified one under "ProteusAvatar plugin". We take the engagement to keep it updated following Unreal versions and Oculus SDKs.

## **I would like to log my users through Oculus Home, but I don't want to use Browse Matchmaking!**

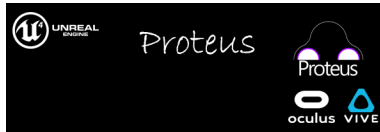
Browse matchmaking is the only way to match players while using blueprints, as of Unreal 4.21. We'll bring more ways soon through code (alternatively, there are C++ methods posted in the forums doing that).

## **My project doesn't work / crashes / doesn't compile! Your template is crap!**

There can be a zillion reasons why your project doesn't compile / crashes. Are you using the latest version of Visual Studio? Correct UE4 version? Windows 10, updated? Is it cold outside? etc. We just want to be sure it's not our template that is crashing your game. Passed that, we're of limited help since, as I said, there could be a zillion reasons why your game does not work.

## **Other settings**

- Follow latest guidelines for Rift / Go / Quest published on both Unreal and Oculus websites
- Yes, products are updated and something that was not possible before, works now (i.e. OpenGL ES 3.1 on the Go)
- Try to avoid any other materials than opaque and masked
- Avoid fancy collision boxes
- Eliminate / reduce to minimum dynamic lights and shadows
- Avoid meshes with high poly count
- Reduce the number of animated objects

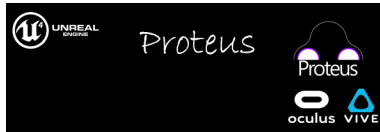


## Supplemental resources

- UE4 Forum/VR development: <https://forums.unrealengine.com/forumdisplay.php?27-VR-Development>
- UE4 Virtual Reality development: <https://docs.unrealengine.com/latest/INT/Platforms/VR/>
- UE4 Networking and Multiplayer: <https://docs.unrealengine.com/latest/INT/Gameplay/Networking/>
- Tom Looman getting started in VR: <http://www.tomlooman.com/getting-started-with-vr/>
- Cedrik Neukirchen UE4 Multiplayer Network Compendium: <http://cedric-neukirchen.net/2017/02/14/multiplayer-network-compendium/>
- Mitch McCaffrey Unreal Engine VR Cookbook: <http://ue4vrcookbook.com/>
- Oculus UE4 GitHub: <https://github.com/oculus-vr/unrealengine>
- Oculus UE4 Developer Guide: <https://developer3.oculus.com/documentation/game-engines/latest/concepts/book-unreal/>
- Vive Tracker for developers: <https://www.vive.com/ca/vive-tracker-for-developer/>
- SteamVR Developer Hardware: <https://steamcommunity.com/app/358720/discussions/>

## Questions?

Don't hesitate to contact me [mathieu.beaulieu@proteus-vr.com](mailto:mathieu.beaulieu@proteus-vr.com) .



## What are the license terms?

Assets created by Epic can be freely reused in any Unreal Engine 4 project.

Assets created by Proteus are under the MIT license terms.

### MIT licence terms (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Proteus