

A?

Aalto University
School of Arts, Design
and Architecture

Coding Virtual Worlds Unity for VR

Prof. Sebastian J. Schlecht
Jan Vornhagen

Today's outline

- **Administration**
- **Discover VR**
- **Feedback on last assignment**
- **Unity for VR**
- **Prepare for the next assignment**

A?

Aalto University
School of Arts, Design
and Architecture

Administration



Gear Status

- **Oculus Pick-up?**
- **Other logistics?**

Pair / Group Assignments

Questionnaires

- **Do you want to work alone or in groups/pairs?**
 - **Weekly assignment?**
 - **Final project?**
-
- **Share .apk in a shared Dropbox folder?**

The background is a dark space filled with a glowing yellow grid of light lines, creating a perspective that leads towards a central, brightly lit doorway. The doorway has a white frame and a red light strip above it. The floor is also covered in the yellow grid lines, receding into the distance.

A?

Aalto University
School of Arts, Design
and Architecture

Discover VR

discover design

- Podcast on VR design
- Spotify, Overcast, etc.



Discover VR

Social VR - Altspace VR or Mozilla Hubs

- **Overview Impressions - What kind of spaces did you visit?**
 - **Getting around - Usability?**
 - **Locomotion - Was moving comfortable?**
 - **Interaction - Could you do stuff?**
 - **Social - Did you meet someone? How did you communicate?**
-
- **... break out and make a few notes together?**

A?

Aalto University
School of Arts, Design
and Architecture

Last Assignment

A futuristic, dark corridor with a glowing yellow grid pattern on the walls and floor. The corridor leads to a doorway with a red light above it. The text "Last Assignment" is overlaid in the center.

Questionnaire


1

How was the assignment?

Average rank ↓

1	2	3	4	5
---	---	---	---	---

How difficult was the assignment? (1 = Very Easy; 5 = Impossible)  3.4

Was the tutorial useful? (1 = not at all; 5 = highly)  3.8

Was the assignment interesting/informative? (1 = not at all; 5 = highly)  3.8

Responses	1	2	3	4	5	Total
How difficult was the assignment? (1 = Very Easy; 5 = Impossible)	0	4 (31%)	2 (15%)	5 (38%)	2 (15%)	13
Was the tutorial useful? (1 = not at all; 5 = highly)	0	3 (23%)	2 (15%)	3 (23%)	5 (38%)	13
Was the assignment interesting/informative? (1 = not at all; 5 = highly)	1 (8%)	2 (15%)	1 (8%)	4 (31%)	5 (38%)	13

Questionnaire

What was the hardest part of the assignment?

- *It took a lot of time ... (~15-20 hrs)*
- *difficult to estimate how long the tasks would take, and what were the parts that needed most attention*
- *It wasn't very technically challenging, but it took much longer than I expected*
- *It should be more clearly emphasized that the only stuff that needs to actually be implemented are the labs*
- *clearer guidelines on which version of Unity*
- *Everything was pretty easy for me*

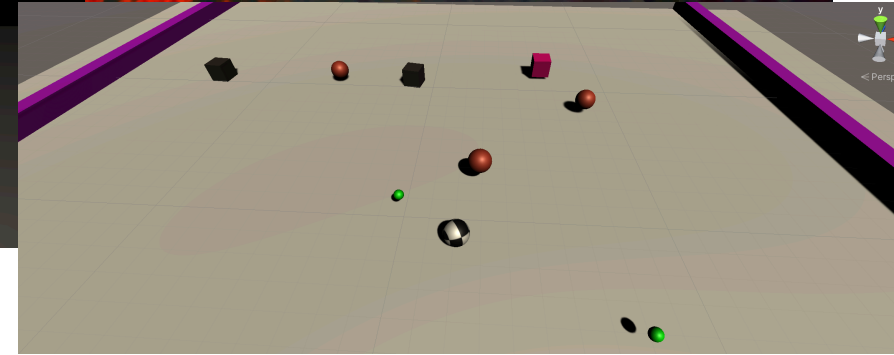
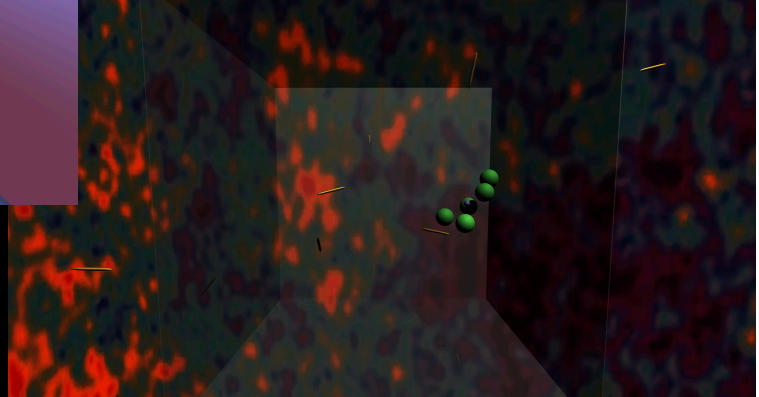
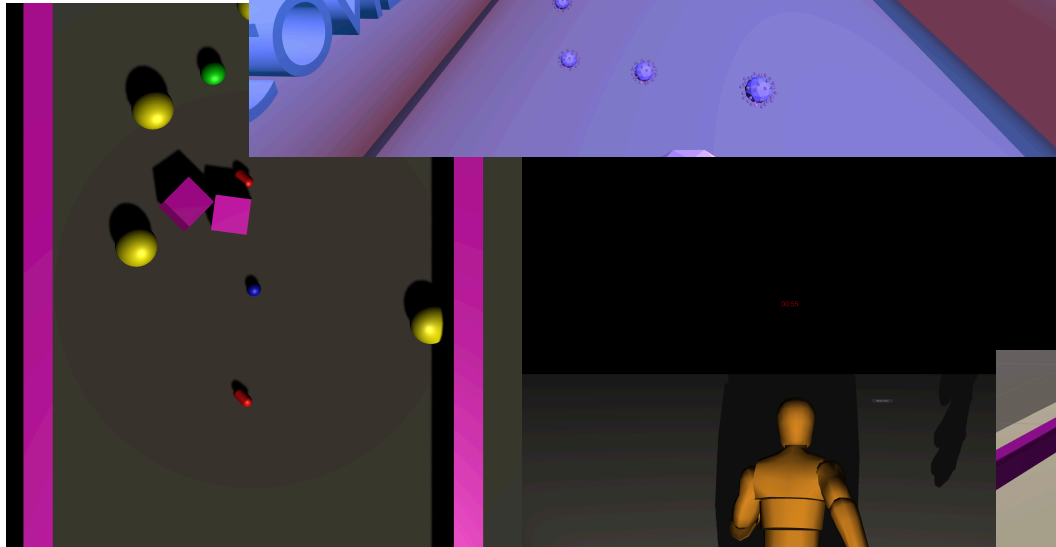
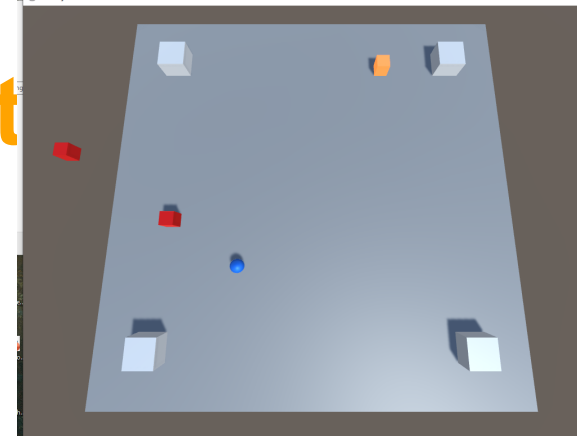
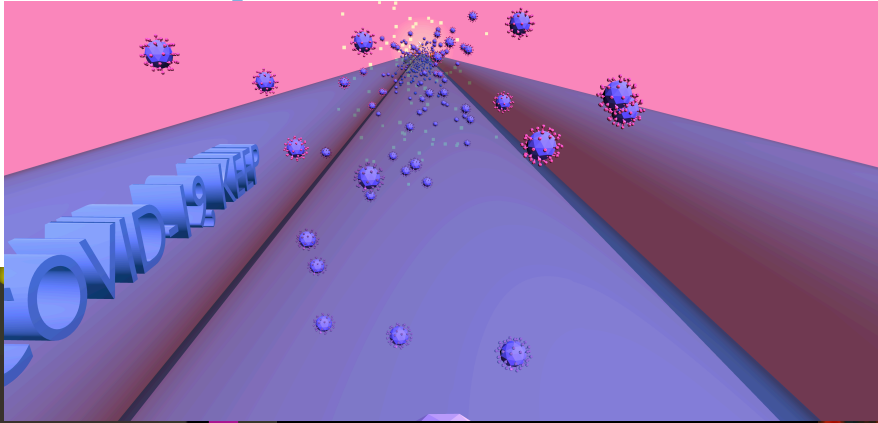
Questionnaire

Do you have further comments?

- ***The scope of what was required for the submission wasn't very clear, partially because the lessons themselves are broken into "lessons", "challenges", and "labs".***
- ***it's hard to estimate the amount of work it takes and that can create frustration even if the task isn't that hard. But the fact that I was forced to watch these tutorials and do the assignment was GREAT. Now I feel much, much better about the next steps as I know I can follow the teaching and learn this stuff.***
- ***I also learned a ton about using Unity and I am really happy about that.***
- ***We could have instead just done a simple game in class to get started, would have learned it much better, and more efficiently. I find this style of DIY learning a bit inconvenient.***
- ***It seems premature to begin a designing a game we'll track throughout the course, as we haven't been exposed to much in the way of conceptual approaches, and the techniques we know are limited.***

Feedback on last assignment

A few examples



A?

Aalto University
School of Arts, Design
and Architecture

Unity for VR

The background of the slide is a 3D-rendered virtual reality environment. It features a dark space filled with a glowing yellow grid of lines that recede into the distance, creating a sense of depth and perspective. In the center, there is a white, arched doorway or entrance. Above the doorway, there are several small, glowing lights in blue, green, and red. The interior of the doorway is brightly lit, showing a white, futuristic-looking interior with various panels and structures. The overall aesthetic is clean, modern, and high-tech.

Unity Overview

What's a game engine?

Tool for games and experiences



- + Importing
- + Input
- + 2D
- + Graphics
- + Physics
- + Scripting
- + Multiplayer and Networking
- + Audio
- + Video overview
- + Animation
- + User interfaces (UI)
- + Navigation and Pathfinding
- + Unity Services
- + XR
- + Open-source repositories
- + Asset Store Publishing
- + Platform development

Unity for VR

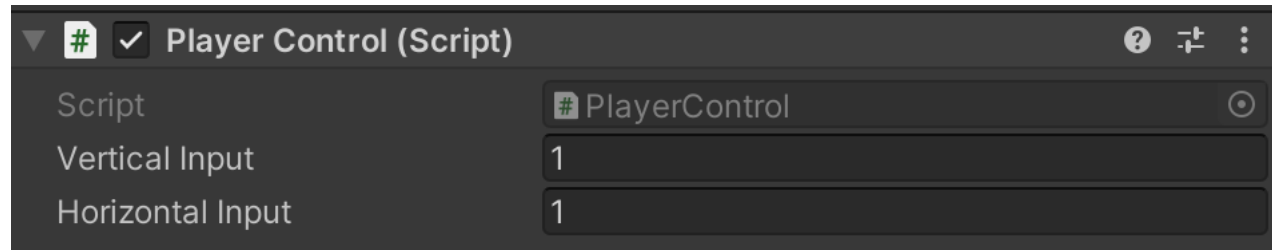
C# - object-oriented programming (OOP) language

```
1  ⚡ using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Test : MonoBehaviour
6  {
7      // Start is called before the first frame update
8      void Start()
9      {
10
11     }
12
13     // Update is called once per frame
14     void Update()
15     {
16
17     }
18 }
```


Unity for VR

C# - Variables - Types & Access Modifiers

```
private float speed = 5.0f;  
  
Rigidbody playerRb;  
  
public float verticalInput = 0.0f;  
  
[SerializeField]  
float horizontalInput = 0.0f;
```



Unity for VR

C# - Functions - Arguments and Return Values

```
private void OnCollisionEnter(Collision collision)
{
    if(collision.gameObject.tag == "Enemy")
    {
    }
}
```

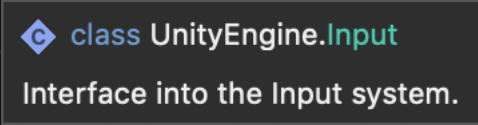
Unity for VR

C# - API - Header and Tooltip

```
if (Input.GetKeyDown(KeyCode.DownArrow))
{
    verticalInput = -1.0f;
}
else if (Input.GetKeyDown(KeyCode.UpArrow))
{
    verticalInput = 1.0f;
}
else {
    verticalInput = 0.0f;
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

```
else if (Input.GetKeyDown(KeyCode.UpArrow))
{
    verticalInput = 1.0f;
}
```

 class UnityEngine.Input
Interface into the Input system.

Unity for VR

Visual Studio - IntelliSense

- Support for Unity API

```
// Start is called before the first frame update
void Start()
{
    Destroy(gameObject, 3f);
}
```

▲ 2 of 2 ▼ void Object.Destroy(Object obj, float t)
Removes a gameobject, component or asset.
t: The optional amount of time to delay before destroying the object.

```
6 public class TestScript : MonoBehaviour
7
8     // Use this for initialization
9     void Start ()
10    {
11
12    }
13    ontri
14
15
```

OnSerializeNetworkView
OnServerInitialized
OnTransformChildrenChanged
OnTransformParentChanged
OnTriggerEnter
OnTriggerEnter is called when the Collider other enters the trigger volume.
OnTriggerExit
OnTriggerExit2D
OnTriggerStay

Unity for VR

Already familiar?

- **Vector3 and Rotate**
- **Transform**
- **GameObject**
- **Rigidbody**
- **GetComponent<Type>**
- **Input**
- **Instantiate**
- **InvokeRepeating**

Unity for VR

Interaction matrix

Collider interaction matrix

	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Rigidbody Trigger Collider
Static Collider		collision			trigger	trigger
Rigidbody Collider	collision	collision	collision	trigger	trigger	trigger
Kinematic Rigidbody Collider		collision		trigger	trigger	trigger
Static Trigger Collider		trigger	trigger		trigger	trigger
Rigidbody Trigger Collider	trigger	trigger	trigger	trigger	trigger	trigger
Kinematic Rigidbody Trigger Collider	trigger	trigger	trigger	trigger	trigger	trigger

Derived from <http://docs.unity3d.com/Manual/CollidersOverview.html>

Unity for VR

MonoBehavior - Documentation

- <https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>

Find

- Start
- Update
- OnCollisionEnter
- Invoke
- GetComponent
- Destroy

MonoBehaviour

class in UnityEngine / Inherits from:Behaviour/ Implemented in:UnityEngine.CoreModule

SWITCH TO MANUAL

Description

MonoBehaviour is the base class from which every Unity script derives.

When you use C#, you must explicitly derive from MonoBehaviour.

This class doesn't support the [null-conditional operator](#) `?.` and the [null-coalescing operator](#) `??`.

For code samples, see the individual MonoBehaviour methods.

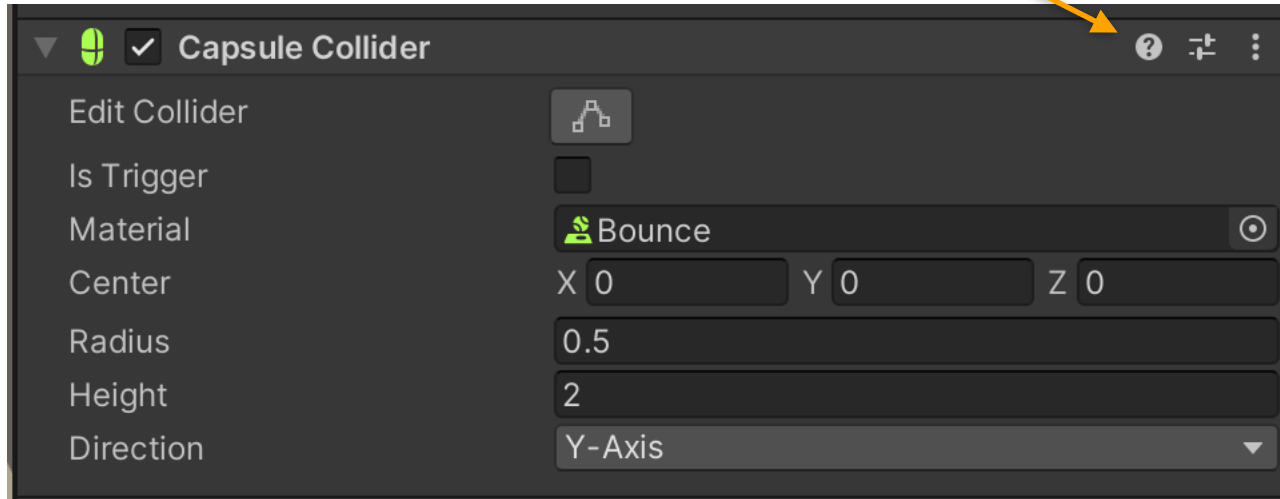
Note: There is a checkbox for enabling or disabling MonoBehaviour in the Unity Editor. It disables functions when unticked. If none of the functions are enabled, the checkbox does not display. The functions are:

[Start\(\)](#)
[Update\(\)](#)
[FixedUpdate\(\)](#)
[LateUpdate\(\)](#)
[OnGUI\(\)](#)
[OnDisable\(\)](#)
[OnEnable\(\)](#)

See Also: The [Deactivating GameObjects](#) page in the manual.

Unity for VR

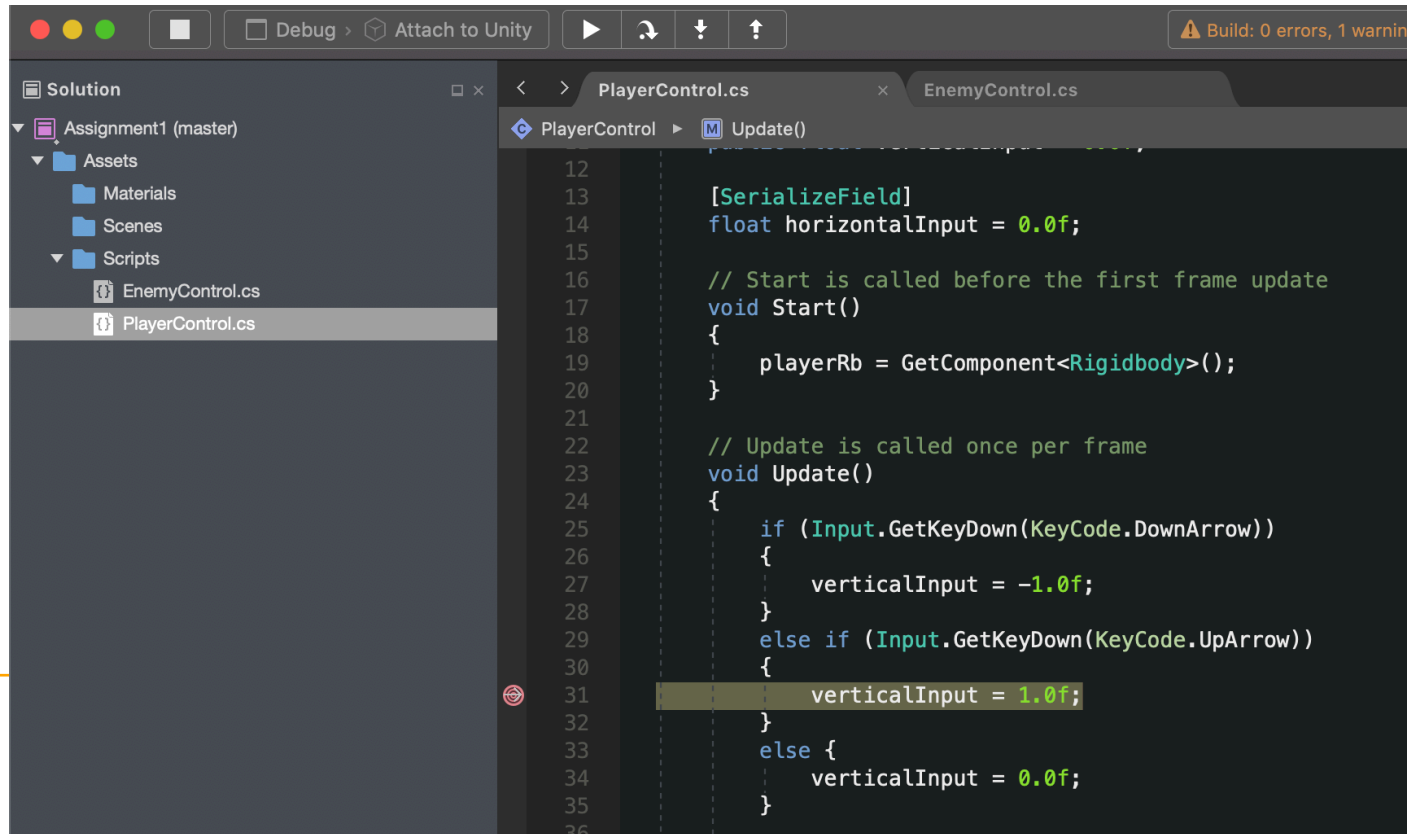
Find Documentation



Unity for VR

Visual Studio - Debugging

- **Attach to Play**



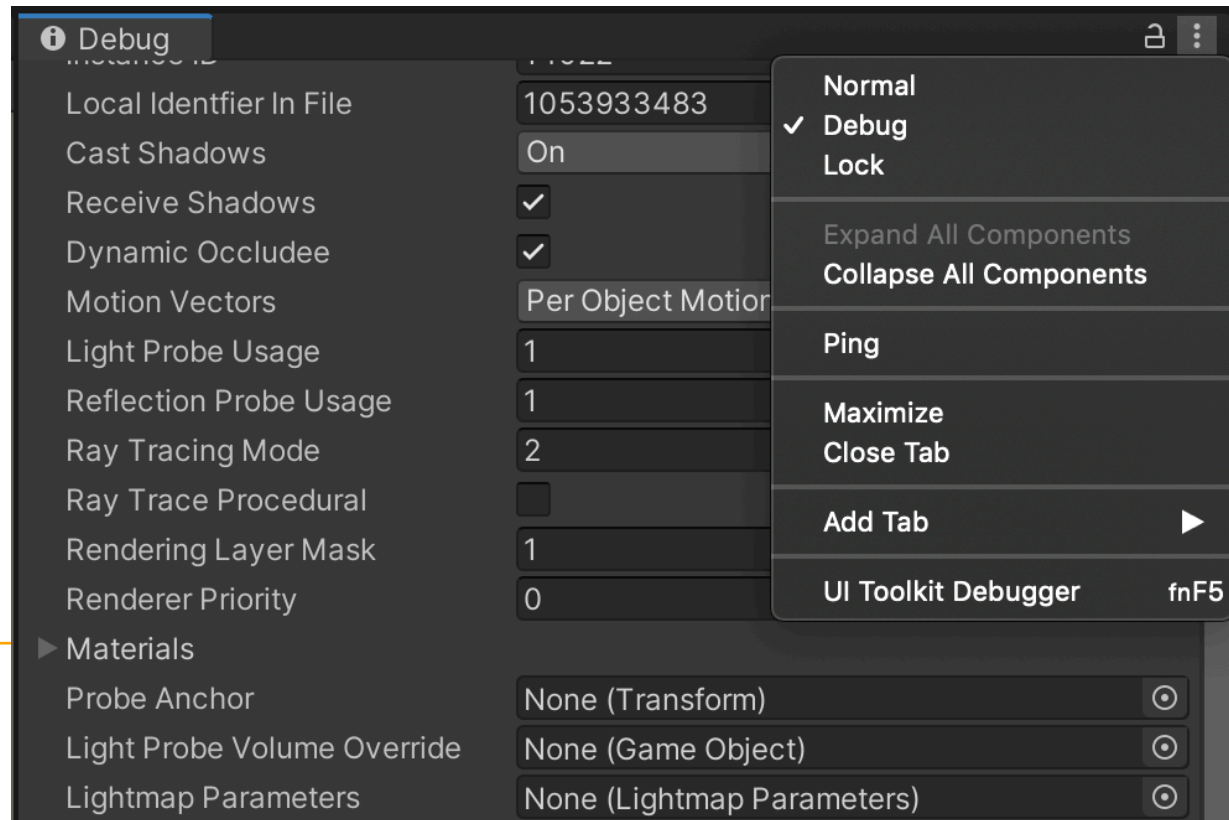
The screenshot shows the Visual Studio IDE with a Unity project. The Solution Explorer on the left shows a project named 'Assignment1 (master)' with folders for 'Assets' (Materials, Scenes) and 'Scripts' (EnemyControl.cs, PlayerControl.cs). The main editor displays the 'PlayerControl.cs' script, which is currently being debugged. The code in the 'Update()' method is as follows:

```
12  
13 [SerializeField]  
14 float horizontalInput = 0.0f;  
15  
16 // Start is called before the first frame update  
17 void Start()  
18 {  
19     playerRb = GetComponent<Rigidbody>();  
20 }  
21  
22 // Update is called once per frame  
23 void Update()  
24 {  
25     if (Input.GetKeyDown(KeyCode.DownArrow))  
26     {  
27         verticalInput = -1.0f;  
28     }  
29     else if (Input.GetKeyDown(KeyCode.UpArrow))  
30     {  
31         verticalInput = 1.0f;  
32     }  
33     else {  
34         verticalInput = 0.0f;  
35     }  
36 }
```

Unity for VR

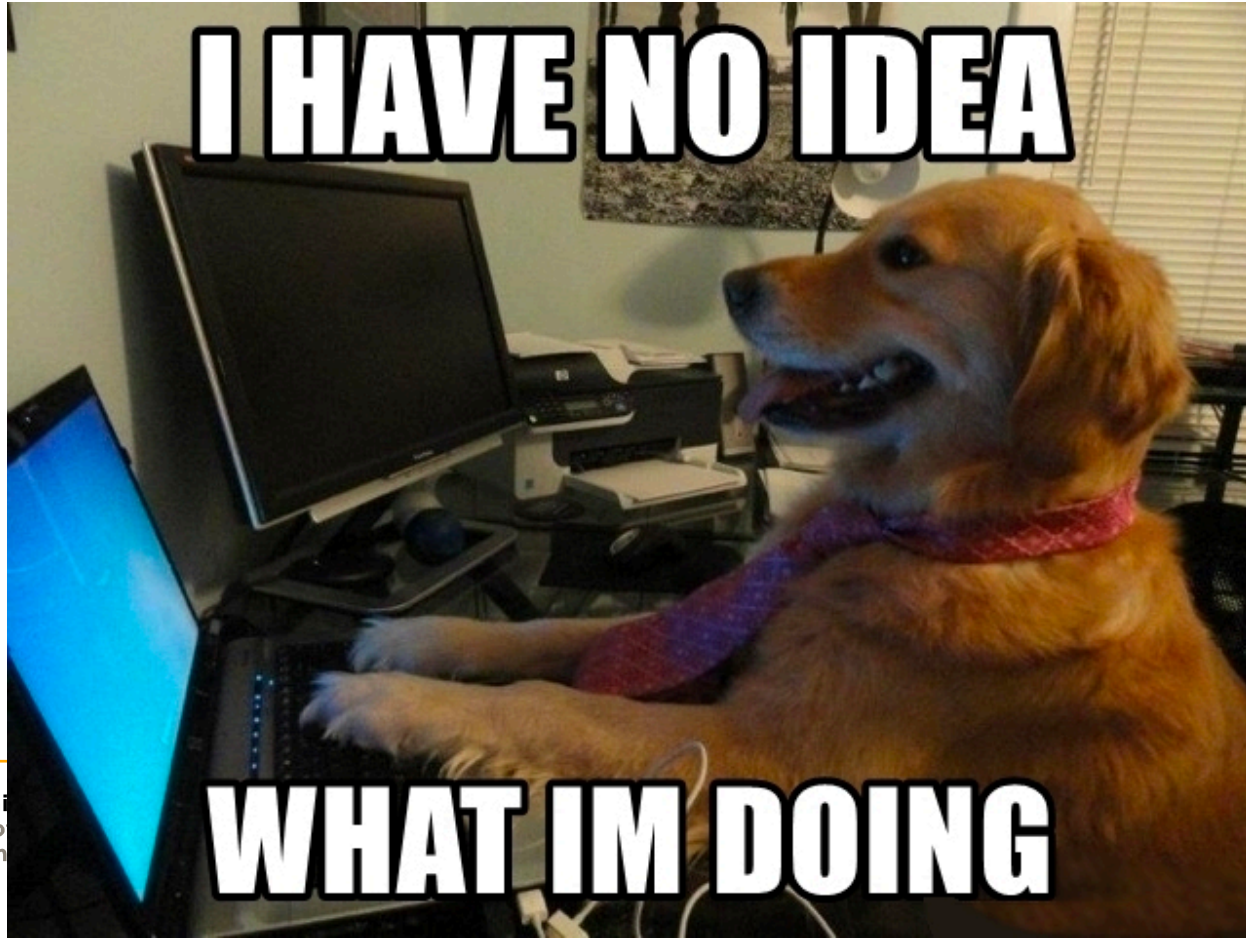
Unity Editor - Debugging

- Show private variables



Being stuck?

... you're likely not the only one



Unity for VR

What to do when stuck?

- **<SearchEngine> is your friend**
- **Toy examples**

Thomson's Rule for First-Time Telescope Makers:

"It is faster to make a four-inch mirror than a six-inch mirror than to make a six-inch mirror."

Unity for VR

What to do when stuck?

Ask, but

- Show that you care about the answer
- Proof that you've done the work
- Give minimum working example
- Appreciate the effort

<https://stackoverflow.com/help/how-to-ask>

Unity for VR

Question Etiquette

Workload of Asking Person > Workload of Helping Person

Unity for VR

Overview

Build-in VR toolkits

- XR Plug-in Framework
- Audio Spatializers

Many add-ons

- VRTK
- RUIS

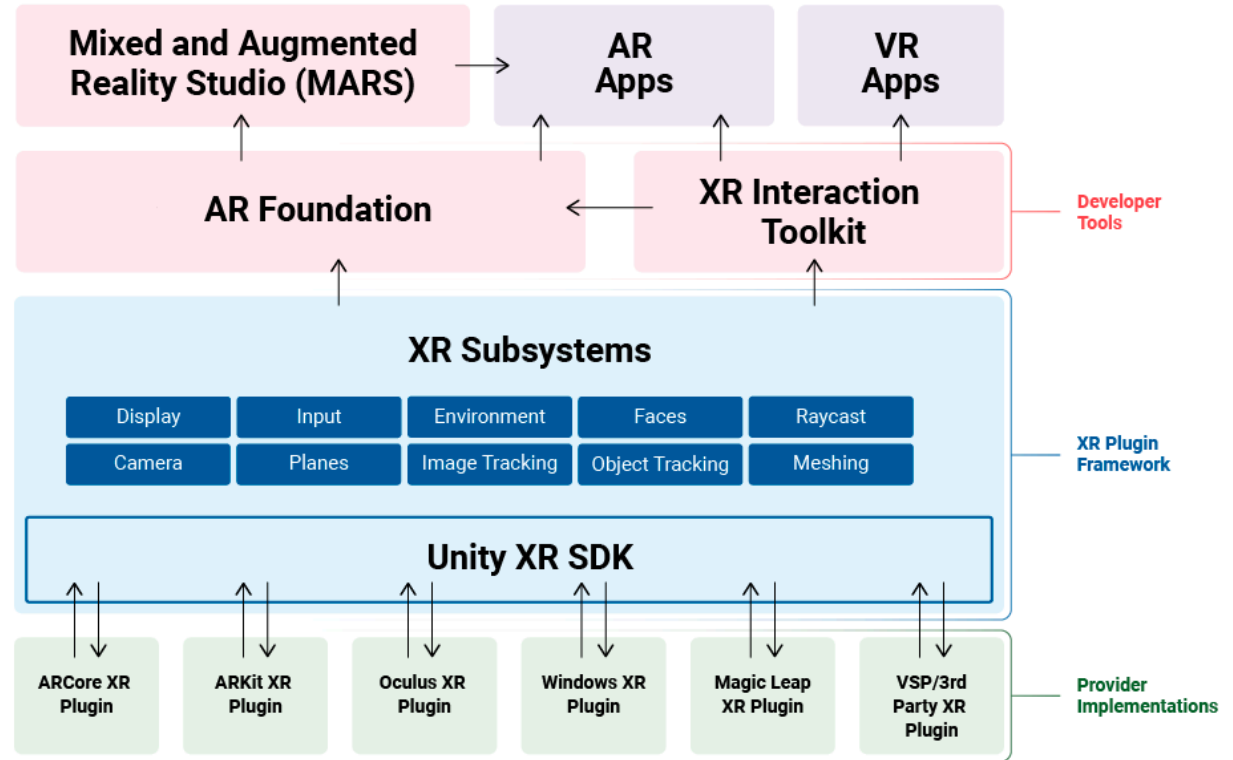
- XR

- Getting started with AR development in Unity
- Getting started with VR development in Unity
- XR Plug-in Framework
- Configuring your Unity Project for XR
- XR API reference
- + Single Pass Stereo rendering (Double-Wide rendering)
- VR Audio Spatializers
- VR frame timing
- + Unity XR SDK

Unity for VR

Since 2019.3

Unity XR Tech Stack





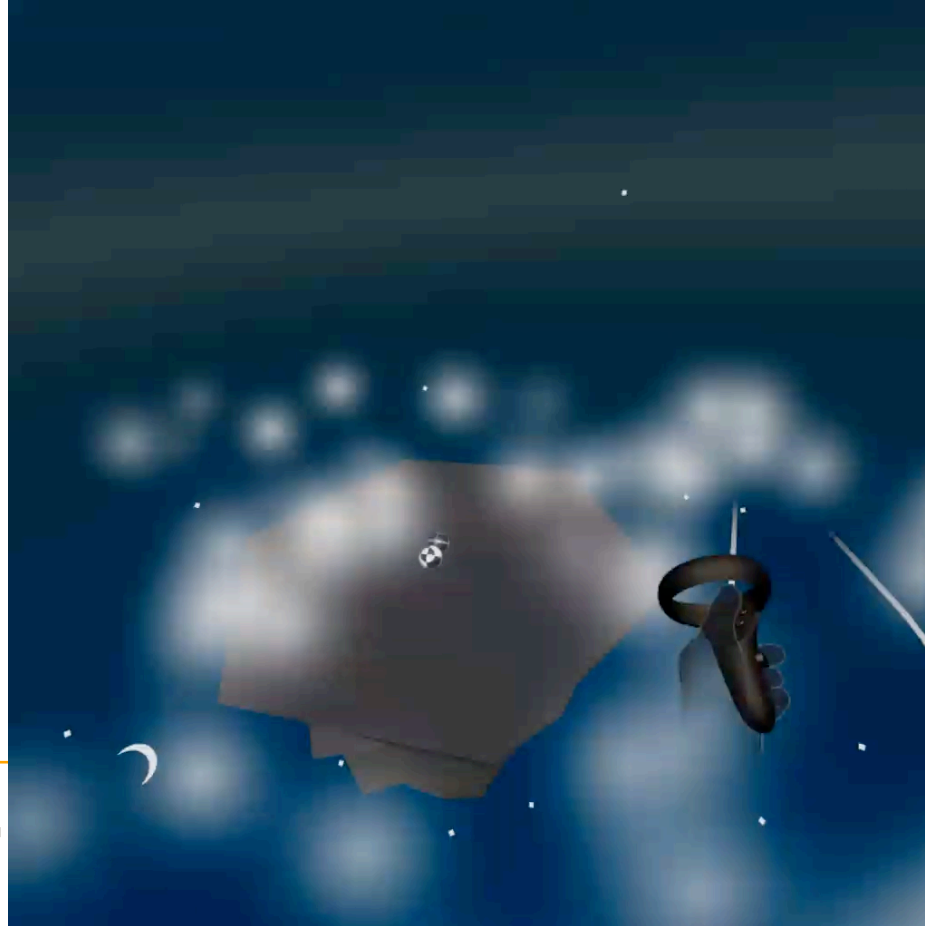
A?

Aalto University
School of Arts, Design
and Architecture

Next Assignment

Next Assignment

Goal - Table Top/3rd Person View & Controller Input



Unity for VR

Assignment Carousel

- Git give access to next person
- Git Forks



Unity for VR

Deliverables

- **.apk**
- **SideQuest**