# Unity Bootcamp

UCF CAP6121 Spring 2023 Mykola Maslych, Ryan Ghamandi (some slides borrowed from Dr. Kevin Pfeil)

#### Lectures Coverage

- Mon, Jan 23
  - Unity Setup with XR Interaction Toolkit (XRIT)
  - Scripting
  - Colliders and Triggers
- Wed, Jan 25
  - Advanced examples with Oculus VR (OVR)

# Outline

- 1. (Optional) Pair Meta Quest phone app with Quest 2
- 2. Install Oculus on PC
- 3. Install Unity
- 4. Download Unity Assets
- 5. Understanding Unity
- 6. Live Demo

### (Optional) Pair Meta Quest App with Quest Device

This step is required if you have not done the initial setup of your Quest 2

- 1. The app is called "Meta Quest"
- 2. Follow instructions to pair the headset with the phone



# Install Oculus Software on your PC

Download from: <u>https://www.meta.com/quest/setup/</u> Install **OculusSetup.exe** and login using a Meta account Connect Quest 2 using a Link Cable



Ouest 2

SET-UP

Download the Oculus mobile app on your smartphone, then wirelessly connect your headset and phone to complete the setup process. You can shop top VR titles from the mobile app or while you're in virtual reality.



#### Install Oculus Software on your PC

0	← -		Searc	:h	<u> </u>	đ	×
Home							
Store		Account Privacy Payment General Beta					
Library		Unknown Sources					
Events		Allow apps that have not been reviewed by Oculus to run. Learn more.					
Devices		OpenXR Runtime Set Oculus	s as active				
Settings							
		Library Locations This is where we'll put apps you install from the Oculus Store. <u>Learn more</u> .	Edit				
		Automatic App Updates Update apps automatically.	-				
		Bandwidth Limit	No Limit 🝷				
		App Downloads While Using VR Allow apps to be installed and updated while you're using a VR app.	•				
Follow List		Randwidth Limit While Lising VR	No Limit *				
Notifications •							
Help Center		Notification Preferences					

#### How Oculus Link Menu Looks like



#### **Download Unity Hub**

#### https://unity.com/download

÷ → C ☆ 🔒 unity.com/download						୍ତି ବ୍	ė
	How to get started	System requirements	New users	Compare plans	Resources	Start 30-day Pro trial	

# Create with Unity in three steps

#### 1. Download the Unity Hub

Follow the instructions onscreen for guidance through the installation process and setup.

Download for Windows Download for Mac Instructions for Linux

#### 2. Choose your Unity version

Install the latest version of Unity, an older release, or a beta featuring the latest indevelopment features.

Visit the download archive

#### 3. Start your project

Begin creating from scratch, or pick a template to get your first project up and running quickly. Access tutorial videos designed to support creators, from beginners to experts.

#### **Download Unity Hub**

- Activate Personal License
- Download Latest Long Term Support version of unity

			Unity Hub 3.4.1			
		× .	MM ~			
	Get Personal Edition License $^ imes$		Projects	Install Unity Editor		
	To Install Unity you will need an active license. By agreeing to the terms below, Unity Hub will generate a free Personal Edition license for you.	>		Official releases Pre-releases Archive		
	By downloading Unity, I confirm that I am eligible to use Unity Personal		😙 Learn	LONG TERM SUPPORT (LTS)		
	or Student Plan per the Terms of service, as I or my company meet the following criteria:	>	Community	CO21.3.16f1 LTS Recommended version	Install	
	→ Do not make more than \$100k in annual gross revenues, regardless of whether Unity Personal is being used for commercial purposes, or for an internal project or prototyping.			2020.3.44f1 LTS	Install	
	$\rightarrow$ Have not raised funds in excess of \$100K.					
	ightarrow Not currently using Unity Plus or Pro Plans.			A 2000 0 04		
	$\rightarrow$ Have read and acknowledged Unity's Privacy Policy				instan	
		62				
	Agree and get personal edition license			Beta program webpage		
lp FAQ			💆 Downloads			

### **XRIT Setup**

- Create a new empty 3D Project
- Then, in: Window -> Package Manager
  - XR Plugin Management (<u>https://docs.unity3d.com/Packages/com.unity.xr.openxr@1.6/manual/ind</u> <u>ex.html</u>)
  - XR Interaction Toolkit

https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.3/)

- Click "Yes" when prompted about new Input System
- Also install "Starter Assets" from XR Interaction Toolkit
- Add all Action Set components to the Action Set

#### Assets for OpenXR project

- Edit -> Project Settings -> XR Plug-in Management -> Check OpenXR
- Edit -> Project Settings -> Preset Manager -> add "right", "left" in ActionBasedController
- Edit -> Project Settings -> XR Plug-in Management -> add "Oculus Touch Controller Profile" to interaction Profiles

🗘 Project Settings			į	
Adaptive Performance Audio	Preset Manager		0	* •
Editor	٩			
Graphics Input Manager Input System Package	# ActionBasedController Filter	(UnityEngine.XR.Interaction.Toolkit.ActionBase Preset	dCont	roller)
Package Manager	= 🗸 🔍 sright	× EXRI Default Right Controller		
Physics Physics 2D	= 🖌 Qleft	× EXRI Default Left Controller		
Preset Manager				
Quality	ActionBasedSnapTurnProvic (L	JnityEngine.XR.Interaction.Toolkit.ActionBasedSnapT	urnPrc	vider)
Scene Template		Preset		
Script Execution Order	= 🗸 🔍	🖻 XRI Default Snap Turn		$\odot$
Ads				
Cloud Build Cloud Diagnostics				



# Unity Intro

#### Tutorial - SampleScene - Windows, Mac, Linux - Unity 2021.3.16f1 Personal <DX11>

File Edit Assets GameObject Component Window Help

 Q Layers - Layout Sign in 🌰 🎎 👌 🗄 🌐 Scene 🛛 🛲 Game : O Inspector 🔾 🔻 2D 🌻 🍕 🛫 🕫 💷 🖛 🕁 🖛 🛛 🔁 SampleScene 🖈 🛟 🖸 🖬 🏵 2 . 🖿 Project 🛛 🗏 Console Clear 🔻 Collapse Error Pause Editor 🕶

0 X

#### Unity Intro

Tutorial - SampleScene - Windows, Mac, Linux - Unity 2021.3.16f1 Personal <DX11>

File Edit Assets GameObject Component Window Help

Sign in 🛛 📥 🙀 ► II ► ③ Q Layers ▼ Layout : 
 Inspector 👌 : 🖶 Scene 🛛 🛲 Game 🔾 🔻 2D 🥊 🔩 🛫 💋 💷 🖛 🔂 🖛 **☆** ▼ **☆** ▼ **☆** ▼ **☆** ▼ Tree of Scene editor/game view scene place to edit/test objects Place to edit properties of a specific Project 🖻 Console object Project files/application and editor output

n X

Game Object Hierarchy

- As you place new things into the world, they are listed in the hierarchy
  - Camera(s)
  - Scenery Objects
  - Player Avatar
  - Etc.
- Click on an object to see/edit details in Inspector
- **Double** click on object to zoom to it in the **Scene**

- Scene Viewer
  - Displays objects in the game world
  - Multiple ways to view your world
  - Game Objects can be manipulated directly
- Game Viewer
  - Displays objects according to the game camera
  - Consider it as a way to "preview" the world without running the whole game

- Inspector
  - For selected Game Object, displays components
  - Components take many forms that affect how the game object looks/behaves
    - Scripts, Meshes, Colliders, Audio Sources...
  - Most components can be modified in Inspector
    - Object "Transform" (position, rotation, scale)
    - Mesh properties
    - Script variables
    - Etc.

- Console
  - Displays errors, warnings, debugging lines, etc.

- Project Assets
  - Quick way to get to prefabs, scripts, and other resources that are part of the project

# Live Demo

- Simple Environment with
  - Ground Plane
  - $\circ$  A couple of boxes
  - XR Origin and navigation

# Set up simple XR Origin

- XR Origin
  - Delete the regular "Main Camera" in the hierarchy
  - Right-click in hierarchy -> XR -> XR Origin (Action-based)
- Locomotion
  - Right click on the XR Origin object in hierarchy -> XR -> Locomotion System (Action-based)
  - Create a plane
  - Click on the plane and "Add Component" with the name "Teleportation Area"
  - You can now teleport by pointing to the plane and hitting the "grab button" (side trigger button)

# Scripting

- In addition to physics-based behavior, for custom behavior we use custom scripts
- Create a new Script
- Add Script to game object
- Update Loop <a href="https://docs.unity3d.com/Manual/ExecutionOrder.html">https://docs.unity3d.com/Manual/ExecutionOrder.html</a>

# Scripting

**RotateAround** https://docs.unity3d. com/ScriptReferenc e/Transform.Rotate Around.html

vr_selection - Cy	/cleCube.cs				≒ Hierarchy				
					+-		<b>۹</b> ۰ All		
CycleCube.cs ⊣ ×				<u>- ≎</u>	🔻 🍕 New	Test*			
Jassembly-CSharp       1       2       1       2       1       1       1       2       1 <td><pre></pre></td> <td><ul> <li></li></ul></td> <td>tion, Ch: 6 SPC</td> <td></td> <td>A set we we</td> <td>eetional Ligh me Interaction M Origin Camera Offse Main Came Main Came Cube Cube Cube e agged form (Mesh Filter Renderer ollider Cube (Scrip It-Material ( Standard</td> <td>t Aanager t tra controller Controller X -3.78554 X 0 X -3.78554 X 0 Cube Cube t) CycleCu Add Compor</td> <td>Layer Default 3 Y 0.73 Y 0 9 Y 1 9 Default 0 9 Default 1 9 Defaul</td> <td></td>	<pre></pre>	<ul> <li></li></ul>	tion, Ch: 6 SPC		A set we	eetional Ligh me Interaction M Origin Camera Offse Main Came Main Came Cube Cube Cube e agged form (Mesh Filter Renderer ollider Cube (Scrip It-Material ( Standard	t Aanager t tra controller Controller X -3.78554 X 0 X -3.78554 X 0 Cube Cube t) CycleCu Add Compor	Layer Default 3 Y 0.73 Y 0 9 Y 1 9 Default 0 9 Default 1 9 Defaul	

0 1 :

0 # :

0 ± : Ø ‡ : Ø ‡ ;

0 i

Z -3.974191

- Colliders are generally for physics
- Triggers are for activating custom behavior
- Triggers also rely on colliders



#### source: LambdaGeneration

- Collision Matrix in Unity
- Unity Example:
- <u>https://docs.unity3d.com/ScriptReference/Collider.OnTriggerEnter.html</u>
- Triggers "trigger" specified behavior
- For triggers to work:
  - 1. Both objects must have colliders
  - 2. One must have "IsTrigger" and must have a RigidBody
- Let's convert the cubes into destroyable enemies!

- Check for triggers of different objects with
  - Object names (<u>other.name == "EnemyCube"</u>)
  - Object tags (<u>other.tag == "Enemy"</u>)
  - Presence of a certain component (<u>TryGetComponent</u>) (arguably the fastest one)

#### Spawning enemies - Object Instantiation

- Documentation:
  - https://docs.unity3d.com/ScriptReference/Object.Instantiate.html
- In the class demo we made a simple script that spawns an enemy randomly every single frame. In your assignments you will need to be smarter about it :)
- Make enemy spawning depend on conditions. For example, every N seconds, or when a single enemy was killed etc. You may need to do this asynchronously, and you could use Coroutines and <u>IEnumerator</u>.

#### More useful resources

- RayCasting helps to get an object that you aim at, and the exact hit location. RayCasting documentation <u>example</u>.
- Force can be used to make objects "jump", and for shooting projectiles. Add force documentation <u>example</u>.
- XRIT Setup full <u>guide</u>.
- **Object Pooling** helps reduce compute load. Documentation <u>example</u>.
- Object Instantiation documentation <u>example</u>.
- Which objects will collide with each other? Depends on the Layers. Collision matrix <u>documentation</u>.

#### Questions