



From Eye to AI: Digital Phantoms for Medical Imaging

Lecture 2 – Unity3D Basics

Joe Xing, Ph.D
CTO and Co-founder of C. Light Tech
Visiting professor at Tsinghua University

Fall Season 2025



C. LIGHT
TECHNOLOGIES



Unity3D Basics

- **A little about Unity3D:**

A cross-platform game engine developed by Unity Technologies. It allows developers to create **2D, 3D, VR (Virtual Reality), and AR (Augmented Reality)** experiences.

Cross-Platform Support: **Windows, macOS, Linux, iOS**, Android, **WebGL**, PlayStation, Xbox, and more.

Editor & Tools: visual editor, asset store, and tools for physics, lighting, animation, and UI.

Programming Languages: Primarily uses **C#** for scripting (OOD)

Asset Store: A marketplace where developers can buy/sell 3D models, animations, scripts, and plugins.

Real-Time Rendering: High-quality graphics with support for real-time lighting, shaders, and particle systems. Pre-baked light, etc.

VR/AR Support: Widely used for **XR** applications with support for devices like **Oculus**, HoloLens, and ARKit/ARCore.

Git Repo

Unity3D Repo

git@gitlab.com:c5952/developer/digital_phantom.git

Windows development environment
WSL, using Linux within Windows

Mac development
environment

Recommend to use WSL

```
(base) | git clone git@gitlab.com:c5952/developer/digital_phantom.git
Cloning into 'digital_phantom'...
remote: Enumerating objects: 104, done.
remote: Counting objects: 100% (104/104), done.
remote: Compressing objects: 100% (77/77), done.
Receiving objects: 27% (29/104), 15.81 MiB | 7.90 MiB/s
Receiving objects: 27% (29/104), 26.38 MiB | 8.79 MiB/s
remote: Total 104 (delta 23), reused 101 (delta 23), pack-reused 0 (from 0)
Receiving objects: 100% (104/104), 147.73 MiB | 11.46 MiB/s, done.
Resolving deltas: 100% (23/23), done.
Checking out files: 100% (85/85), done.
```

Terminal

```
(base) joe.xing@Joes-MacBook-Pro:~/work$ git clone git@gitlab.com:c5952/developer/digital_phantom.git
Cloning into 'digital_phantom'...
remote: Enumerating objects: 104, done.
remote: Counting objects: 100% (104/104), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 104 (delta 23), reused 101 (delta 23), pack-reused 0 (from 0)
Receiving objects: 100% (104/104), 147.73 MiB | 21.77 MiB/s, done.
Resolving deltas: 100% (23/23), done.
```

<https://learn.microsoft.com/en-us/windows/wsl/install>

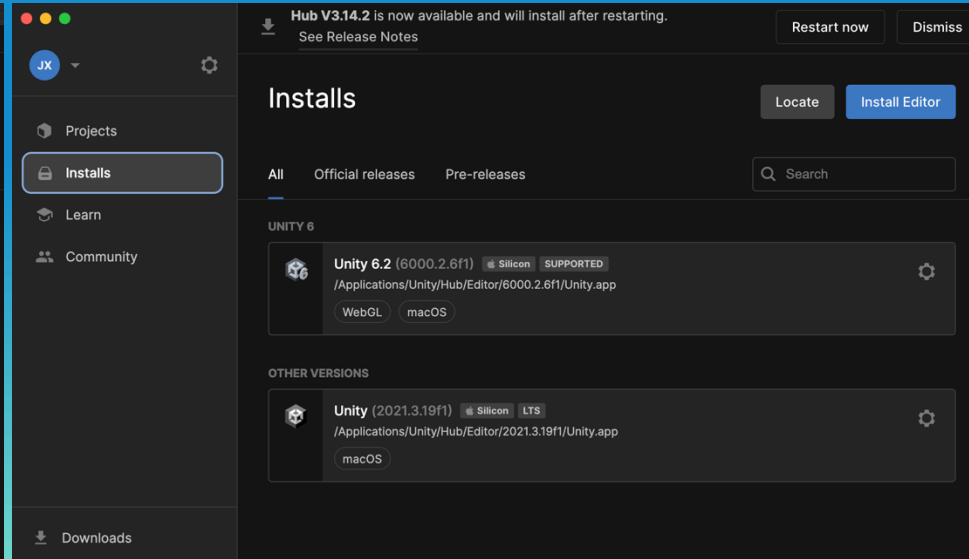
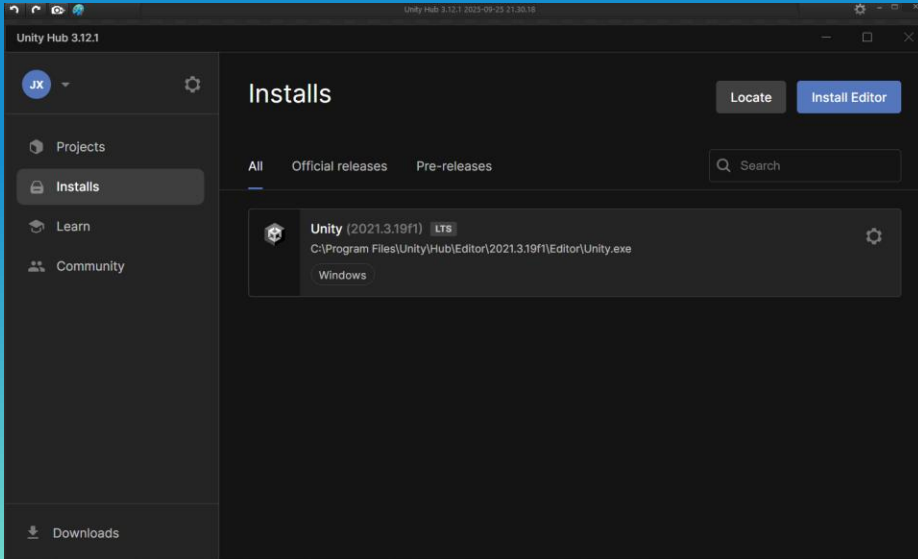
Gitlab account setup, RSA key
<https://docs.gitlab.com/user/ssh/>

Unity Hub

<https://docs.unity3d.com/hub/manual/InstallHub.html>

Microsoft Windows 11
x64-based PC
CPU i-9 cores, NVIDIA GeForce RTX 3080 GPU
64 GB RAM

Apple Silicon (M4 Max)
ARM64

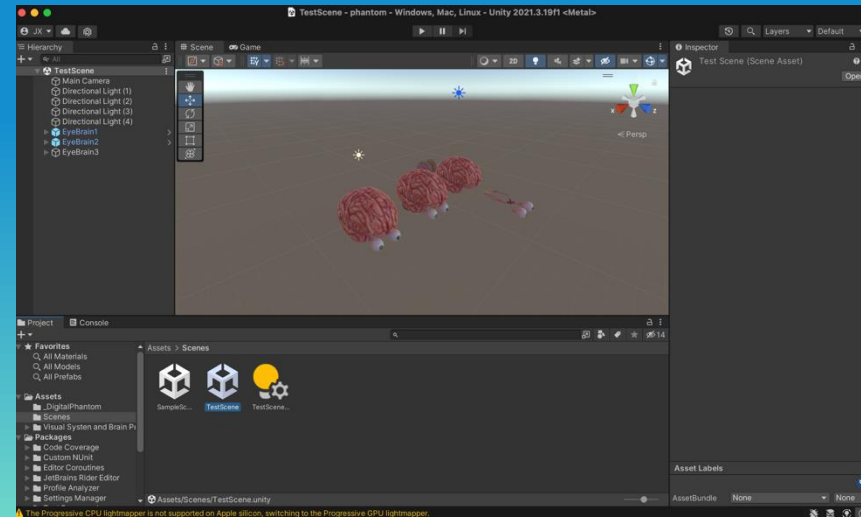
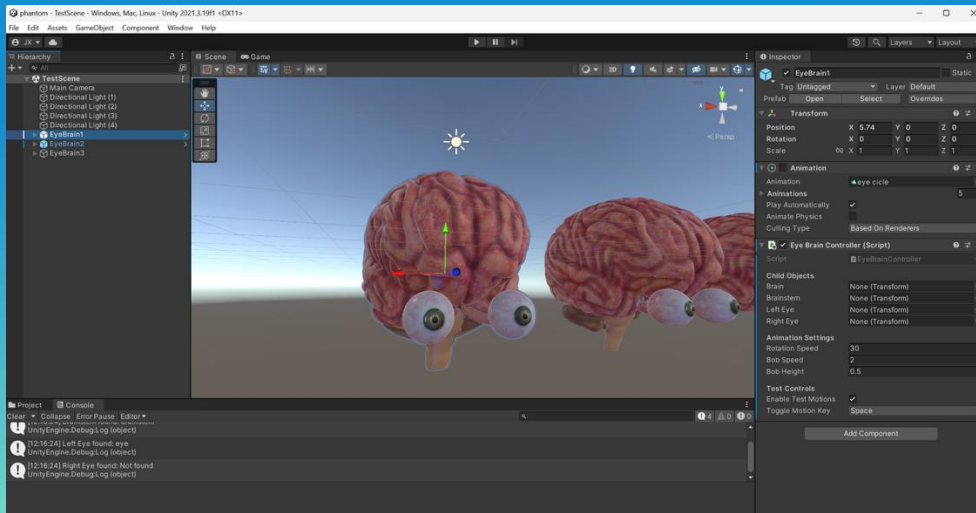


Unity3D Editor

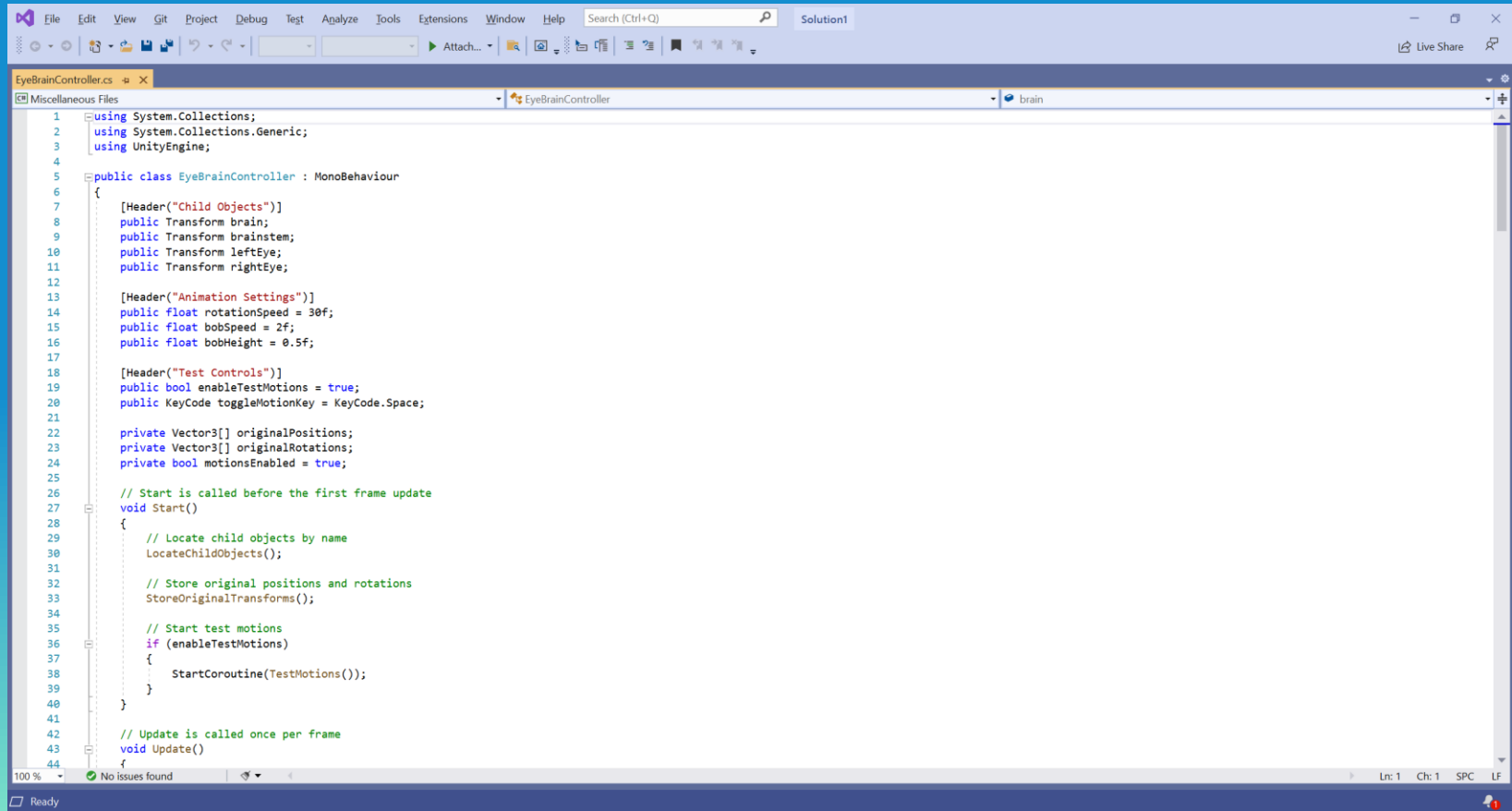
Version tested: 2021.3.19f1 (SOFA dependencies)

Microsoft Windows 11
x64-based PC
CPU i-9 cores, NVIDIA GeForce RTX
3080 GPU
64 GB RAM

Apple Silicon (M4 Max)
ARM64



Visual Studio



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class EyeBrainController : MonoBehaviour
6 {
7     [Header("Child Objects")]
8     public Transform brain;
9     public Transform brainstem;
10    public Transform leftEye;
11    public Transform rightEye;
12
13    [Header("Animation Settings")]
14    public float rotationsSpeed = 30f;
15    public float bobSpeed = 2f;
16    public float bobHeight = 0.5f;
17
18    [Header("Test Controls")]
19    public bool enableTestMotions = true;
20    public KeyCode toggleMotionKey = KeyCode.Space;
21
22    private Vector3[] originalPositions;
23    private Vector3[] originalRotations;
24    private bool motionsEnabled = true;
25
26    // Start is called before the first frame update
27    void Start()
28    {
29        // Locate child objects by name
30        LocateChildObjects();
31
32        // Store original positions and rotations
33        StoreOriginalTransforms();
34
35        // Start test motions
36        if (enableTestMotions)
37        {
38            StartCoroutine(TestMotions());
39        }
40    }
41
42    // Update is called once per frame
43    void Update()
44    {
```

100 % No issues found Ln: 1 Ch: 1 SPC LF

Simple Test of Monobehavior

PascalCase

```
EyeBrainController.cs
digital_phantom > unity > phantom > Assets > Scripts > EyeBrainController.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class EyeBrainController : MonoBehaviour
6 {
7     [Header("Child Objects")]
8     public Transform brain;
9     public Transform brainstem;
10    public Transform leftEye;
11    public Transform rightEye;
12
13    [Header("Animation Settings")]
14    public float rotationSpeed = 30f;
15    public float bobSpeed = 2f;
16    public float bobHeight = 0.5f;
17
18    [Header("Test Controls")]
19    public bool enableTestMotions = true;
20    public KeyCode toggleMotionKey = KeyCode.Space;
21
22    private Vector3[] originalPositions;
23    private Vector3[] originalRotations;
24    private bool motionsEnabled = true;
25
26    // Start is called before the first frame update
27    void Start()
28    {
29        // Locate child objects by name
30        LocateChildObjects();
31
32        // Store original positions and rotations
33        StoreOriginalTransforms();
34    }
35}
```

Eye_brain_controller.cs

Eye-brain-controller.cs

A class in Unity is a C# blueprint that describes how a GameObject behaves or stores data.

The Transform controls where a GameObject is, how it's rotated, and how big it is in the game world.

Event Sequence
Awake → prepare
Start → initialize
Update → per-frame logic
FixedUpdate → physics
LateUpdate → afterward

THANKS

DO YOU HAVE ANY QUESTIONS?

contact@joexing.me

