



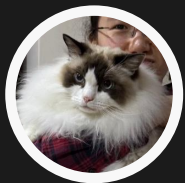
Introduction to Course and Unity

Introduction to Game Development in Unity
Spring 2026, 98-127, Lecture 1

Instructors: Jingxuan Chen, Dario Quintero, Shangyi Zhu, Jeffrey Wang



Course Instructors



**Jingxuan
Chen**

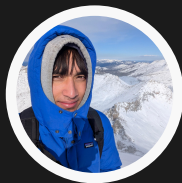
Senior in CS

Favourite Game Genres:

Metroidvania, Open World,
Narrative Driven

AndrewID: jingxua4

Preferred Contact: Discord



**Dario
Quintero**

Junior BXA (Art+CS)

Favorite Games: Sekiro,
Death Stranding, Spelunky

AndrewID: darioq

Preferred Contact: Discord



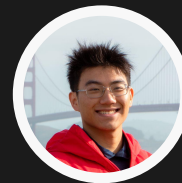
**Shangyi
Zhu**

Master's in Music Tech

Favorite Games: Okami,
Don't Feed the Monkeys

AndrewID: shangyiz

Preferred Contact: Discord



**Jeffrey
Wang**

Junior BXA (Art+CS)

Favorite Games:

Outerworlds, Catan,
Minecraft

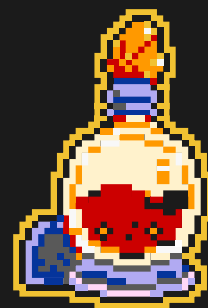
AndrewID: jw8

Preferred Contact: Discord

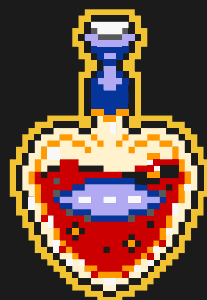
Grading and Logistics

- Assignments - 40%
 - 4 assignments, each worth 10%
 - Consists mostly of coding and implementation in Unity
- Final Project - 32%
 - Extend the work done from the assignments into a small playable game
 - More details will be provided in the project write up when it's released
- Attendance - 28%
 - Attending each class is worth 2% of final grade
 - Attendance will be taken in the middle of class
- Late-work Policy
 - **No late days**
 - 10% grade penalty on the submitted work for every day late





Let's get
started :D



What is a game engine?

A game engine provides the framework and many of the tools necessary to develop and game. For example, they:

- Can render your scenes
- Already have asset pipelines in place
- Have a game loop and common systems like physics already coded





“Developing a game without a game engine is like asking a painter to chop down a tree to make their own canvas”

—ME



How to install Unity

- Install Unity Hub: <https://unity.com/download>
- Download the recommended Editor version using Unity Hub

The screenshot displays the Unity Hub interface. On the left sidebar, the 'Installs' menu item is highlighted with a red box and labeled '1'. The main content area shows the 'Installs' section with a search bar and a 'Locate' button. The 'Install Editor' button is highlighted with a red box and labeled '2'. Below the search bar, there are tabs for 'All', 'Official releases', and 'Pre-releases'. The 'All' tab is selected. Underneath, the 'UNITY 6' section is visible, showing the Unity 6 (6000.0.29f1) version with a 'Security Alert' icon, 'Silicon' and 'LTS' labels, and a 'Manage' dropdown menu. The file path is shown as '/Applications/Unity/Hub/Editor/6000.0.29f1/Unity.app'. Below the file path, there are buttons for 'Windows' and 'macOS'.



Create new project

Select Editor version and template, though default usually works

← New project

Templates
Select an Editor version, then choose a template to start your project.

Editor version *

6000.0.29f1 Security Alert Silicon
LTS

All Core Sample >

Search

Core SRP
Universal 2D

Core SRP
Universal 3D

Unity organization
Eishe

Project name *
My project

Location *
/Users/chenjingxuan/Desktop/stuff/unit... Folder icon

Use Unity Version Control ?

+ Create project

Name your project

Select location on disc to save project

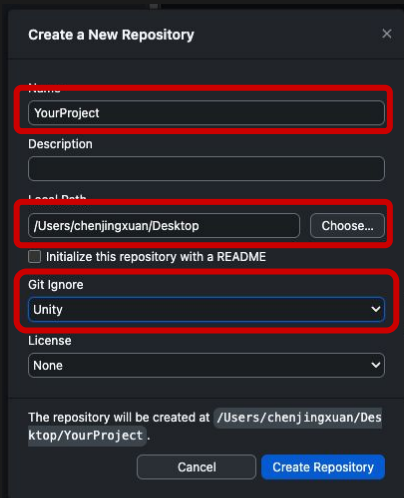


Version control

- Download GitHub Desktop: <https://desktop.github.com/download/>

Set same name as your
newly created project

Set same path as your project



Create a New Repository

Name
YourProject

Description

Local Path
/Users/chenjingxuan/Desktop Choose...

Initialize this repository with a README

Git Ignore
Unity

License
None

The repository will be created at /Users/chenjingxuan/Desktop/YourProject.

Cancel Create Repository

**Important: Select
Unity for git ignore !!**

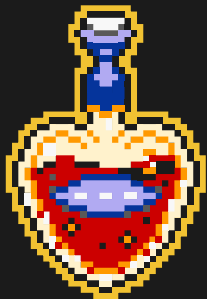
If done correctly, your Unity project's root directory will now be recognised as a GitHub repository





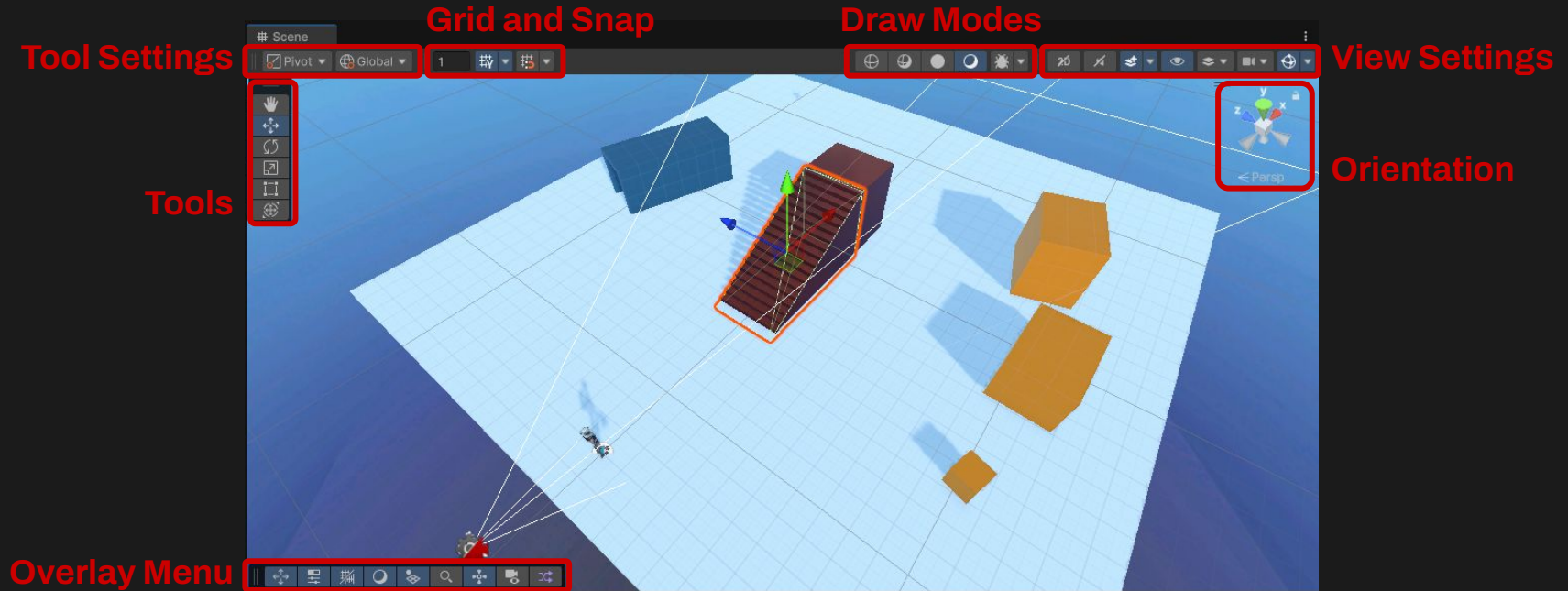
Introducing the Editor

The Editor interface consists of a toolbar and a modular collection of windows



Windows - Scene

Visualize and interact with the world you create



Useful Shortcuts



Basic Navigation

MMB-Drag / Ctrl-Alt-LMB-Drag:
Pan view

Alt-LMB-Drag: Orbit view

Alt-RMB-Drag / Scroll: Zoom view

F: Center view on selected object



Flythrough Mode

Hold-RMB to activate mode

Drag: Rotate view

QWEASD: Move

Hold-Shift: Move faster

Scroll: Change speed



Transform Tools

QWERTY: Switch between tools

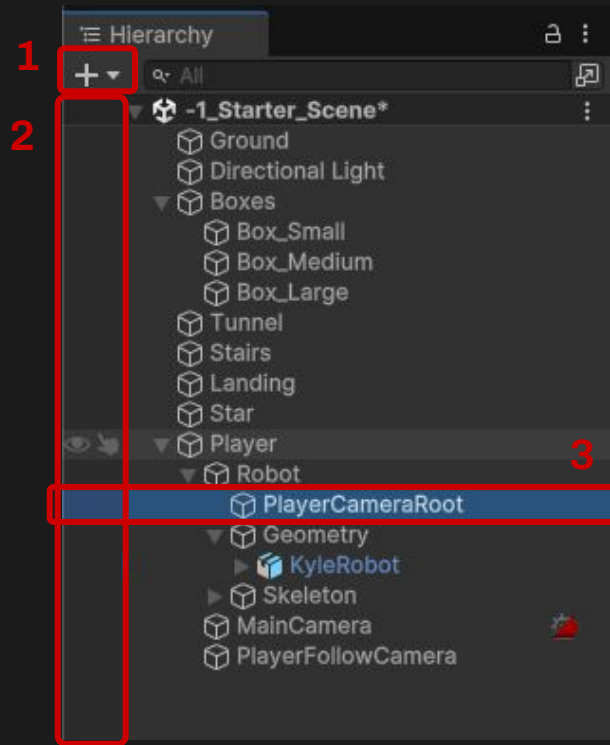
Hold-Ctrl: Toggle on snapping



Windows - Hierarchy

View and manage objects in the scene

1. Create new objects. Right-click in the window to open the same context menu
2. Toggle visibility and pickability of objects within the scene
3. Current selected object is highlighted in blue



Windows - Game

Preview how your final built application will look



Toolbar

Located at the top of the Editor interface



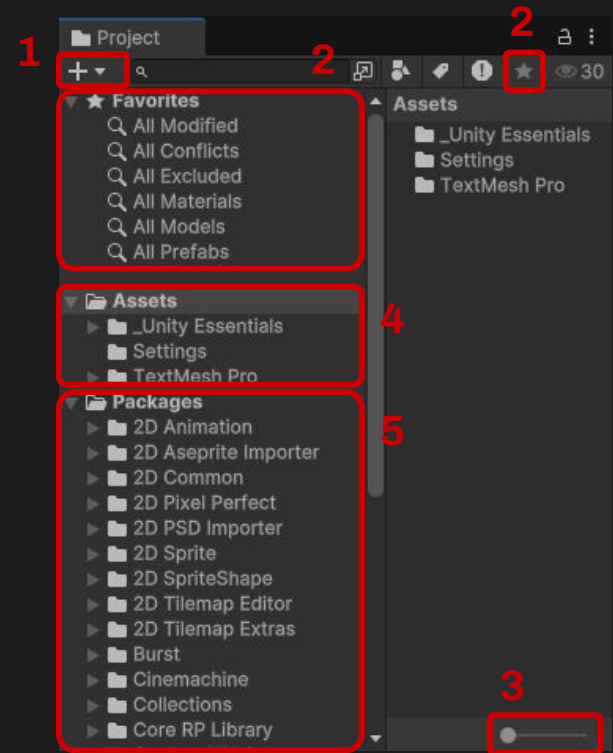
1. Toggle between Editor and Play mode; Pause Play mode; Step Play mode by one frame
2. Change, save and/or load Editor window layouts



Windows - Project

Navigate and find assets and files

1. Create new files/folder within the currently selected folder. Can also be accessed with right-click and navigating to Create
2. Save common searches for quick access
3. Adjust icon size of files



Windows - Project

4. The main folder within your project. All new files created should be placed under this folder. Physically located in the root project directory
5. Contains external packages downloaded by the Package Manager. Physically, it's metadata is stored under Packages while it's actual contents are located within Library/PackageCache

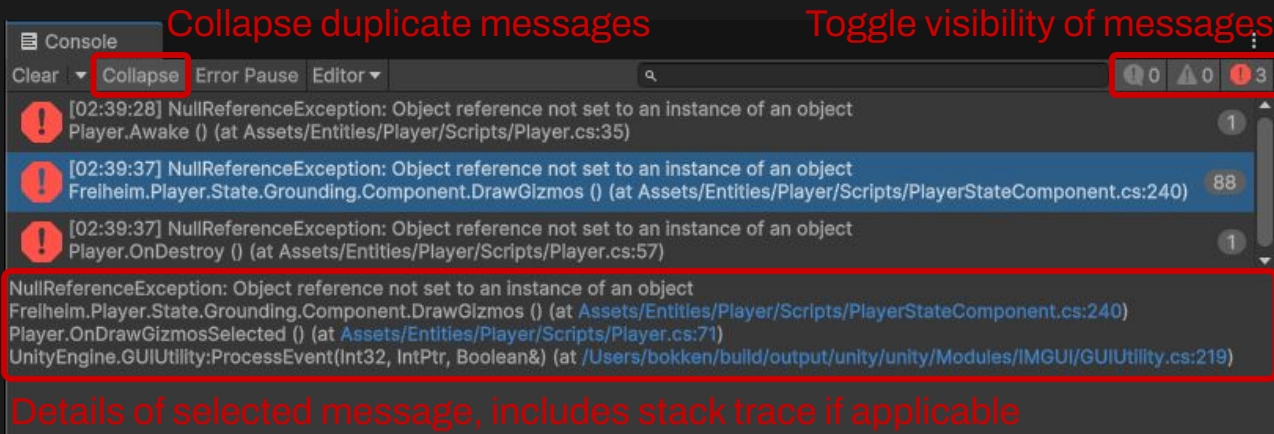
| Name | Date Modified | Size | Kind |
|-------------------|---------------------|------|--------|
| > Assets | Yesterday, 11:40 PM | -- | Folder |
| > Library | Today, 1:59 AM | -- | Folder |
| > Logs | Yesterday, 11:42 PM | -- | Folder |
| > Packages | Yesterday, 11:41 PM | -- | Folder |
| > ProjectSettings | Yesterday, 11:41 PM | -- | Folder |
| > Temp | Today, 2:11 AM | -- | Folder |
| > UserSettings | Yesterday, 11:54 PM | -- | Folder |

Contents of the root project directory



Windows - Console

Displays errors, warnings, and other messages generated by the Editor

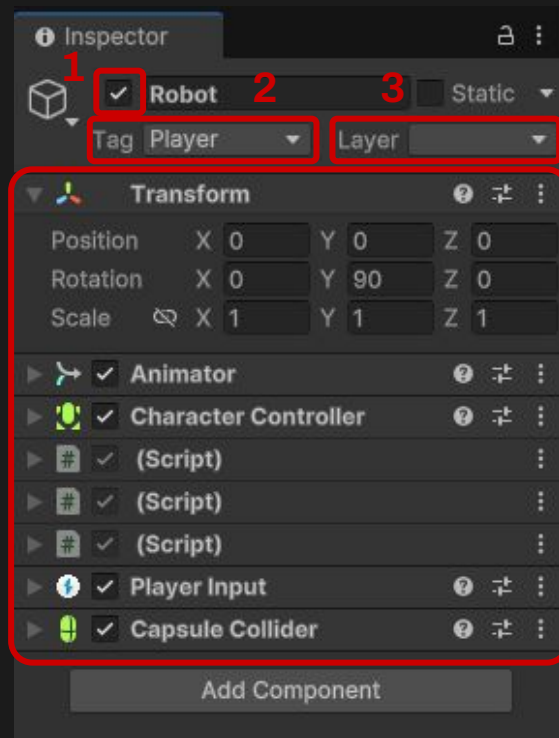


Windows - Inspector

Displays properties of the currently selected object or asset.

Let us consider the properties of a GameObject:

1. Toggles whether the object is active
2. An optional identifier that can be assigned to the object
3. Similar in function to tags, but additionally, collisions between layers can be dictated by a collision matrix
4. Components attached to this GameObject



What is a component?

An entity component system is an architectural pattern often used in video game development to manage game world objects



Entity

A barebones container with a unique identifier



Component

A struct that usually contains only data



System

Code that act on entities containing particular sets of components





How to code gameplay?

By creating custom components that derive from the
MonoBehaviour class in Unity



The MonoBehaviour class

```
1 using UnityEngine;
2
3 0 references
4 public class YourScript : MonoBehaviour
5 {
6     // Start is called once before the first execution
7     // of Update after the MonoBehaviour is created
8     0 references
9     void Start() { }
10
11     // Update is called once per frame
12     0 references
13     void Update() { }
14 }
```

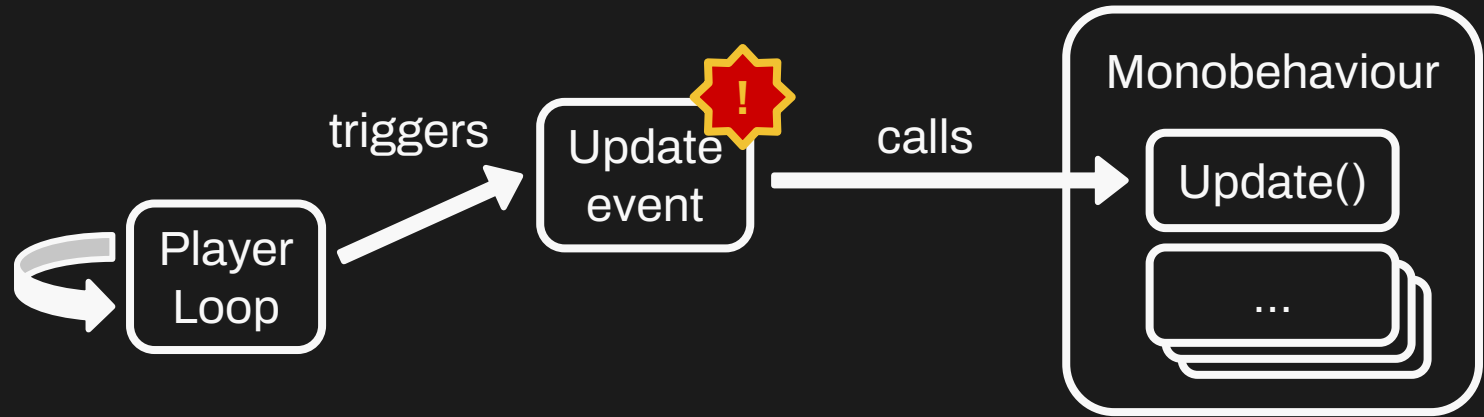
Start and Update are among what Unity calls event functions



What are event functions?

Event functions are predefined callbacks that are triggered by various Editor and Engine events.

By defining a method with the expected name, Unity is able to detect and automatically call these functions.



Execution Order

Event functions always run in a fixed order. For example:

- Awake always runs before Start
- LateUpdate always runs after Update

The [Unity Manual](#) provides a full flow chart of the event function execution order for MonoBehaviour scripts.



Resources

This lecture only introduced some of the most common features of Unity.

If you're interested in knowing more, we highly recommend referencing the [Unity Manual](#) and [Scripting API](#)

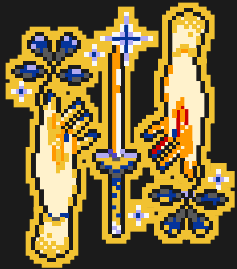
It is also ideal to set up proper IDE support before coding in Unity:

- [Unity Development with VS Code](#)
- [Unity - Manual: Visual Studio C# Integration](#)
- [Game Development for Unity | JetBrains Rider](#)



End of Lecture

We will take a 10 min break before
continuing with lab



CREDITS: This presentation template was created by [Slidesgo](#),
and includes icons by [Flaticon](#), and infographics & images by
[Freepik](#)

