10 - Platform Specific Game Development

CS 3160 - Game Programming
Max Gilson

Platforms Are Very Specific

- Games are played on many different platforms today
- Every platform has specific limitations that reduce the scope and capabilities of games developed for them
  - Example: an iPhone does not have powerful enough hardware for the same graphical fidelity as a Playstation 5
  - Example: the controls of a VR game are fundamentally different than a keyboard and mouse

Understanding Your Audience

- Players of different platforms have different expectations
  - There’s many reasons why Candy Crush-like games are well received on mobile and not on PC
- Think of the context in which your audience plays games
  - How long? 5 min bursts or 5 hours?
  - Where? On the bus or on the couch?
  - Why? Just passing time or grinding for max level?
  - What? Battle royale or match 3?

Platforms (Sorted by Estimated Player Count)

1. Mobile - iPhone, Android, Amazon Fire Tablet
2. PC - Windows, Linux, Mac
3. Console - Playstation 5, Xbox Series X/S
4. Handhelds - Nintendo Switch 1 & 2, Steam Deck, ROG Ally
5. Web - HTML5, WebAssembly, Javascript
6. VR/AR - Meta Quest 3, PSVR2, Valve Index
7. Microconsoles - Playdate, PICO-8, Thumby
8. Retro - NES, N64, Commodore 64
9. Other - Smart TVs, Roku, Apple Watch

Tools and SDKs

- Developing for any platform requires support from the manufacturer of the platform
  - Or at the very least, reverse engineering efforts from a passionate community
- Before embarking on a development journey, gather existing resources for your target platform(s)
  - We are spoiled with choice when working with Unity

Tools and SDKs Documentation

- Hopefully, there is some documentation, tutorials, or sample code for whatever tools you decide to use
  - Official documentation
  - Wikis
  - Github
  - YouTube
  - Textbooks
- Using tools that do not have documentation is not recommended

Tools and SDKs Comparison by Platform Example

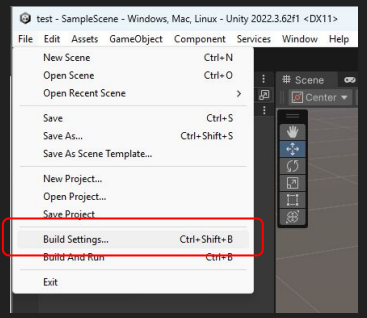
- Mobile Game Development Tools:
  - Unity
  - Unreal
  - Godot
  - GDevelop
  - Defold
  - Cocos2d
  - libGDX
  - Flax Engine
  - Ren'Py
- Smart TV Game Development Tools:
  - Web App???
  - Roku-gameEngine???
  - AndroidTV???
  - tvOS???

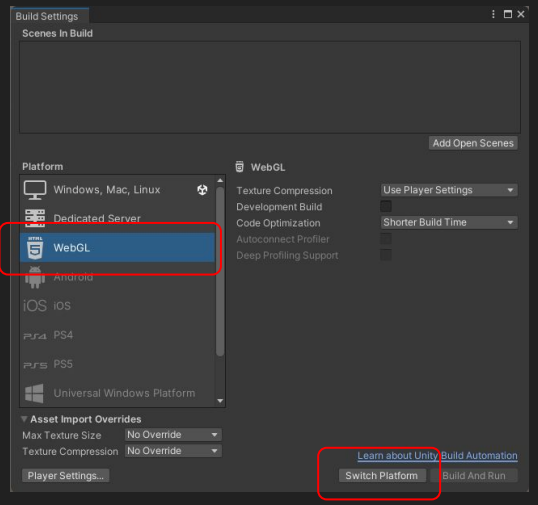
Tools and SDKs Comparison on Unity

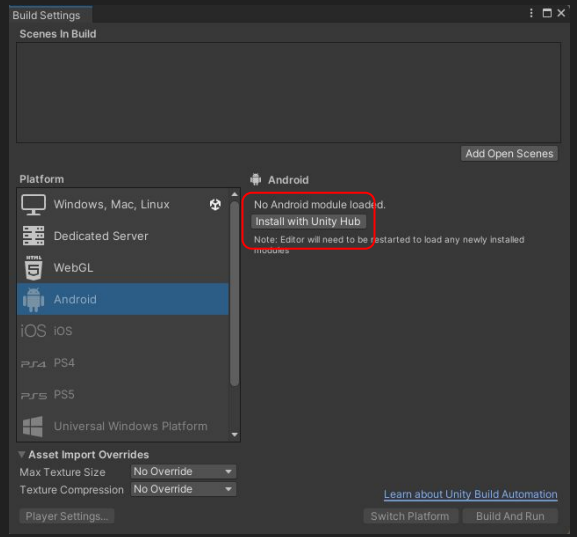
- Unity Supported Platforms:
  - Windows, Mac, Linux
  - iOS, Android, Fire OS
  - Oculus, Playstation VR, ARCore
  - PS5, PS4, Xbox One, Xbox Series X/S, Nintendo Switch
  - Web
  - Embedded systems
  - tvOS
- Some platforms have emulators for testing
  - Example: in Unity, you can emulate an Android phone to test your app if you do not have an Android phone

Selecting Build Target in Unity

- To build for different platforms you must go into the “Build Settings” window
- Next, select the platform you want to build for and click “Switch Platform”
- Building for some platforms requires installing extra packages

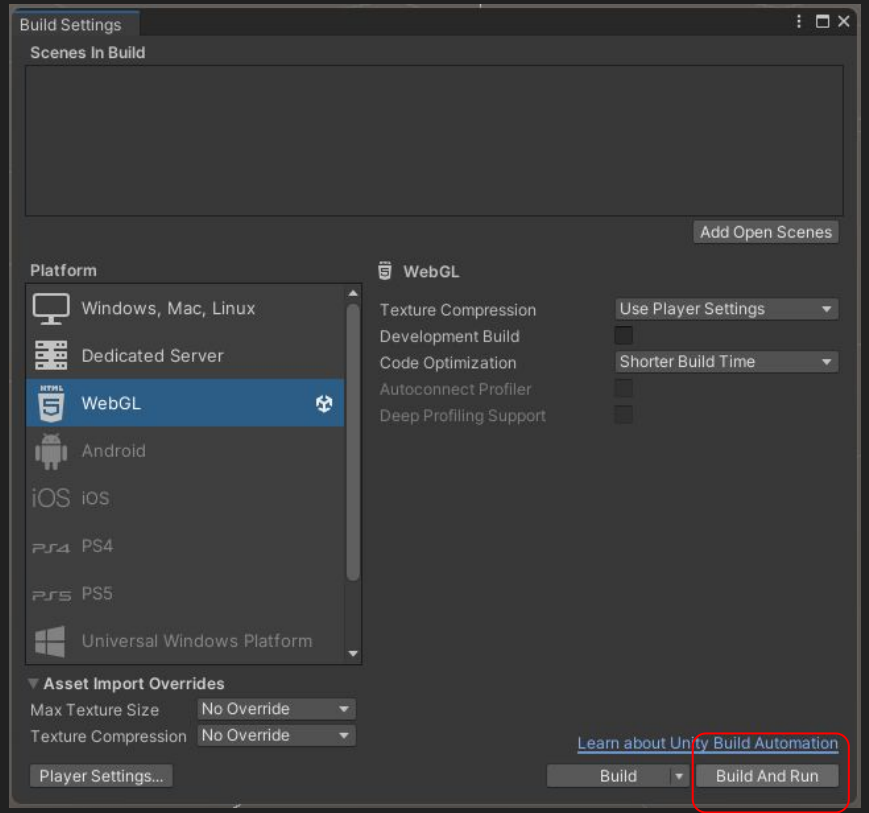






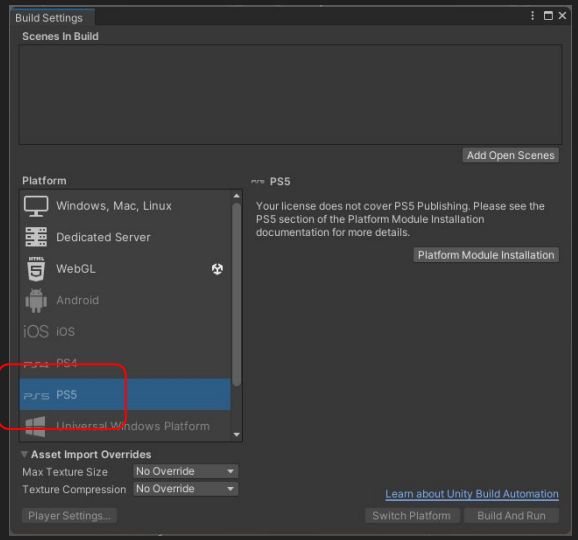
Selecting Build Target in Unity (cont.)

- After a build target is selected, you can use “Build and Run” to test your game on your platform



Building a Unity Project for Consoles

- Building a Unity project for consoles is not really possible solo
- Requires a legitimate business
  - Only credible businesses are eligible for producing console games
- Requires a Unity Pro subscription
  - $200/mo per user
- Requires a license to make games for the specific platform
  - Provided after approval by the console manufacturer
- Requires a console dev kit
  - $2,000 to $2,500
  - Only provided after license approval
- Requires official SDK
  - Only provided after license approval



Building a Unity Project for Consoles (cont.)

- Some consoles require platform specific features implemented
  - Achievements, user accounts, file storage access, etc
- Uploading builds and patching builds on The Playstation Store requires following dozens of rules and regulations

Building a Unity Project for Mobile

- Mobile store fronts are still restrictive, but not as restrictive as consoles
- Meet SDK/API level (as of 2024)
  - Android requires API level 34 minimum
  - iOS requires SDK version 18 minimum
- Implement other Android/iOS specific build requirements
  - iOS requires building on a Mac
  - Apps are required to be digitally signed with a key so they can be verified by Google/Apple
  - This prevents a rogue employee from uploading a “fake” or malicious app
- Create company/app page for the app
- Setup closed beta testing via email list
- Publish app
  - Your published app must be reviewed and can be rejected for any reason
  - Try again! You are at the mercy of Apple and Google!

Platform Storefronts Features and Integrations

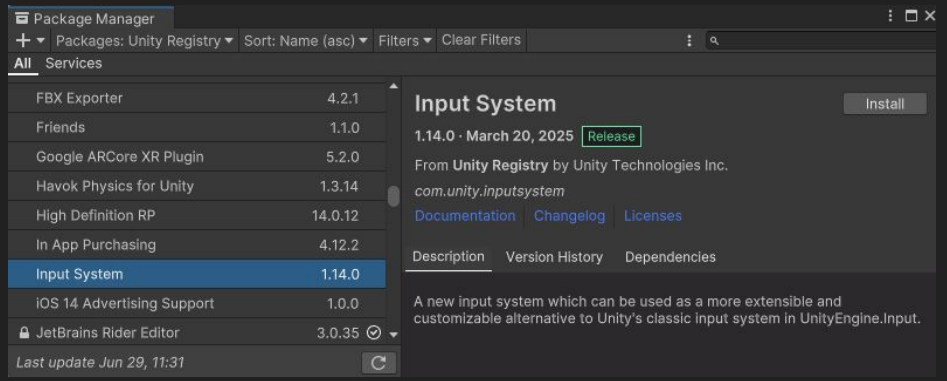
- There are many stores that allow you to distribute your game
- These stores may have special features that you can take advantage of:
  - Mobile - App Store (iOS) and Google Play Store (Android)
    - Leaderboards
    - Achievements
    - In App Purchasing (IAP)
    - Google Ads Integration
    - Push notifications
    - Camera, GPS, Motion, etc. access
  - PC - Steam, Epic Games Store, and GOG
    - Achievements
    - Steam Deck Verified
    - Friends list
    - Cloud save
  - Web - Itch.io
    - Downloadable vs in-browser

Platform Specific Inputs

- Many platforms have unique controls:
  - Consoles have controllers
  - Mobile has touch
  - PC has keyboard and mouse
  - VR has motion controllers, hand gestures, eye tracking
- Many platforms can share controls:
  - Example: some people play games on PC or mobile using a controller

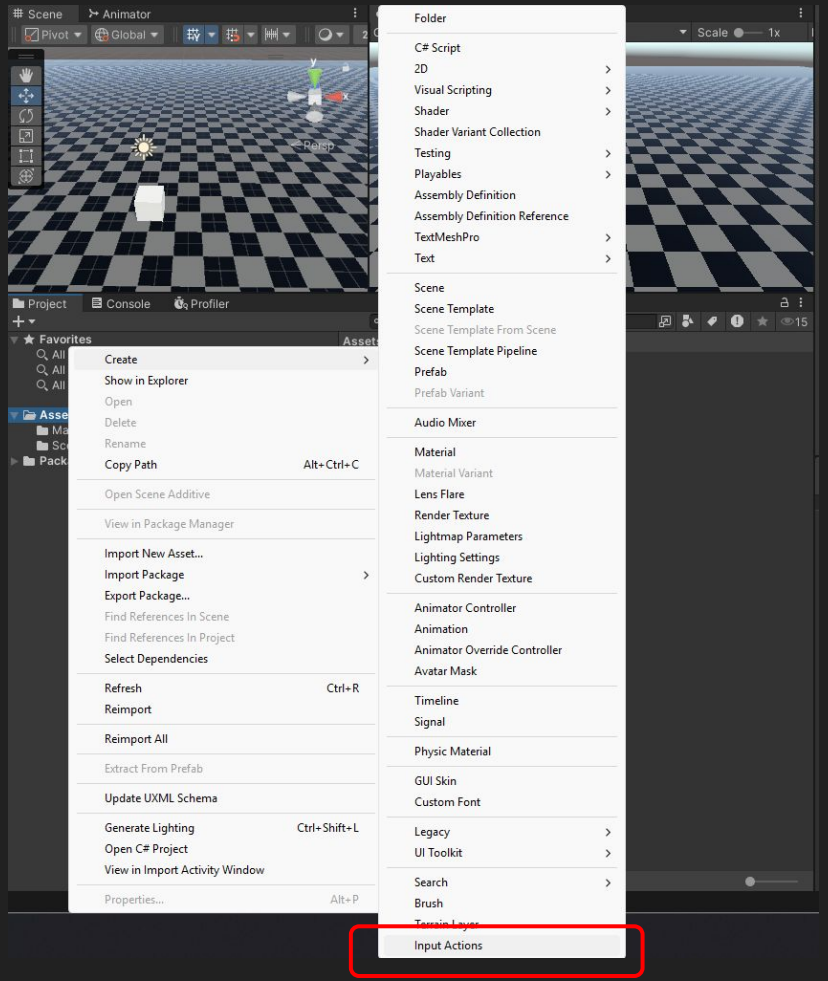
Platform Specific Inputs in Unity

- Unity has a “new” input system designed for handling input for any platform
- To add the new input system it must be added as a package:
  - Unity Registry -> Input System
- Next, restart the editor



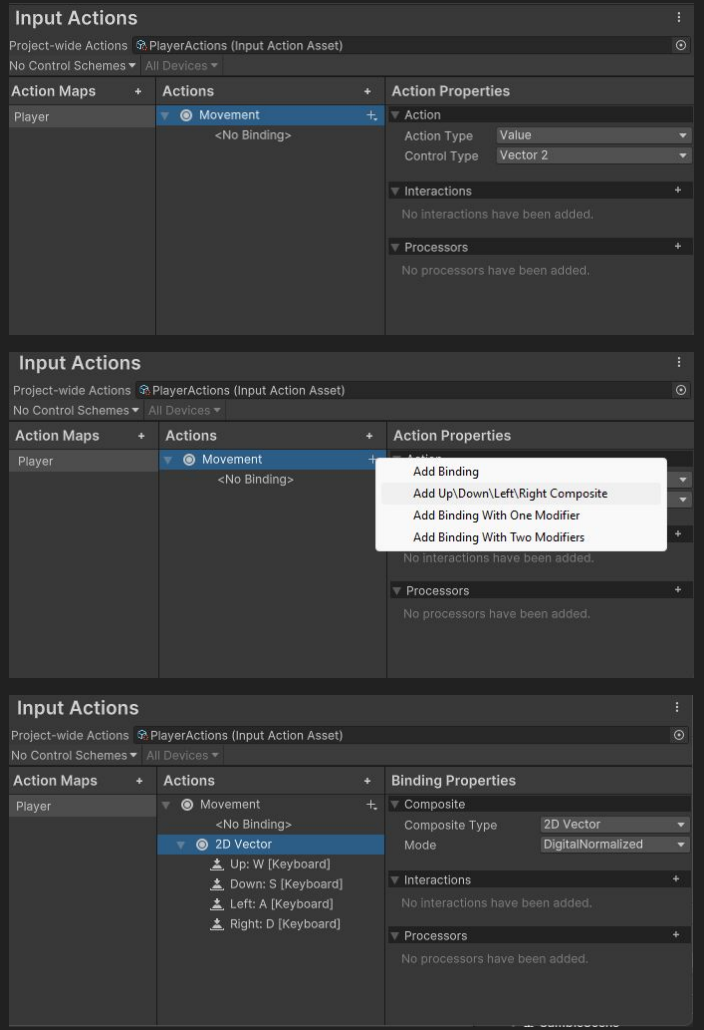
Platform Specific Inputs in Unity (cont.)

- Create new “Input Actions”
- Select “Generate C# Class”
- Go to “Project Settings” and modify the “Action Maps”



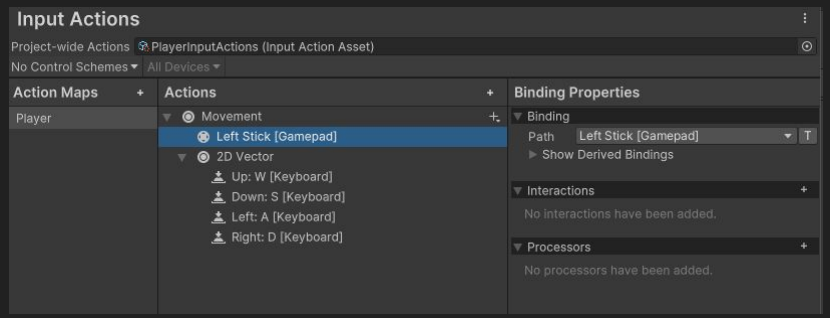
Platform Specific Inputs in Unity (cont.)

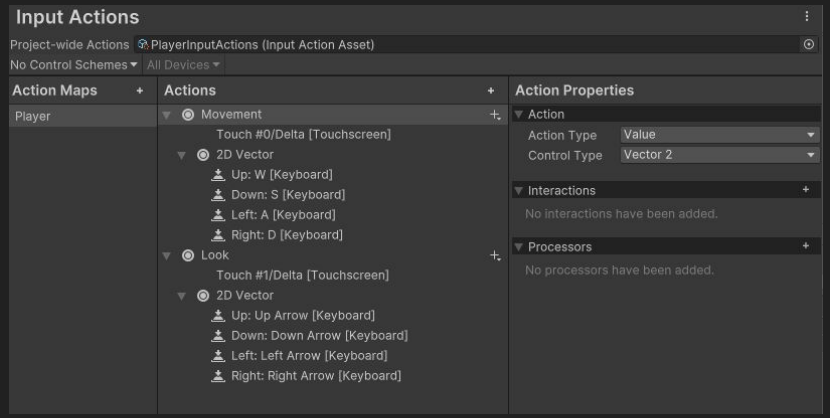
- Add a “Player” Action Map and set all of the Actions for your desired platforms
- For keyboard WASD controls:
  - Set “Action Type” to “Value”
  - Set “Control Type” to “Vector 2”
  - “Add Up/Down/Left/Right Composite”
  - Bind WASD

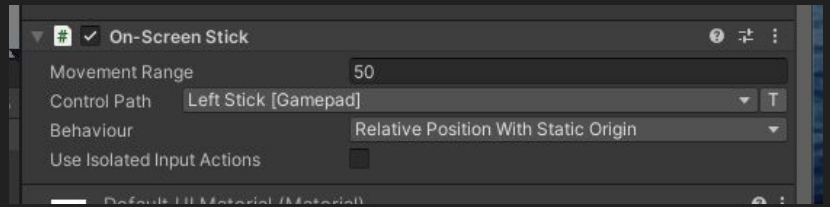


Platform Specific Inputs in Unity (cont.)

- For controller:
  - Add “Left Stick” binding
- Another Action can be similarly added for rotating
- For touch controls:
  - Create a UI element that uses “On-Screen Stick” that is mapped to “Left Stick”







Platform Specific Inputs in Unity (cont.)

CubeController.cs (old input system)

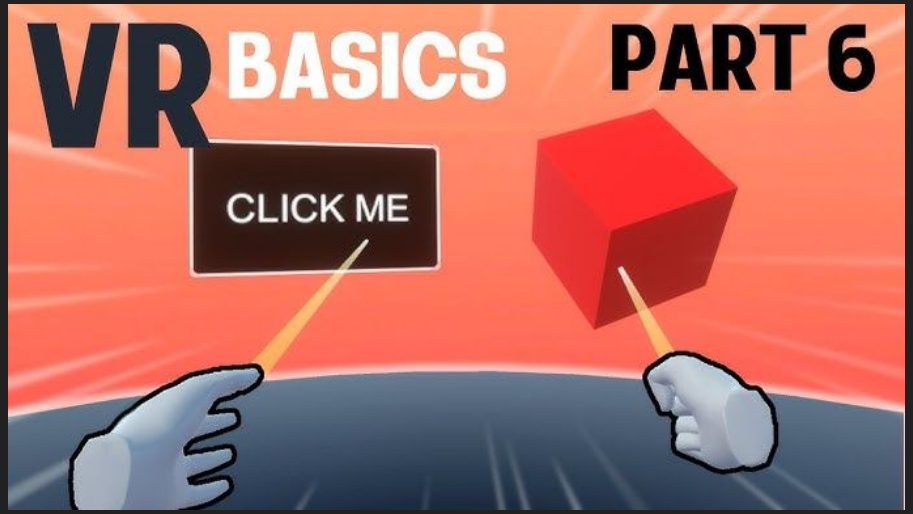
```
public class CubeController : MonoBehaviour
{
    public float velocity = 5f;
    void Update()
    {
        UseOldInputSystem();
    }

    private void UseOldInputSystem()
    {
        Cursor.lockState = CursorLockMode.Locked;

        var movement = new Vector2(Input.GetAxis("Horizontal"), Input.GetAxis("Vertical"));
        var look = Input.GetAxis("Mouse X");

        MoveAndRotate(movement, look);
    }

    private void MoveAndRotate(Vector2 movementDelta, float rotationDelta)
    {
        var position = transform.position;
        position.x += velocity * movementDelta.x;
        position.z += velocity * movementDelta.y;
        transform.position = Vector3.Lerp(transform.position, position, Time.deltaTime);

        var rotation = transform.rotation.eulerAngles;
        rotation.y += rotationDelta;
        transform.rotation = Quaternion.Euler(rotation);
    }
}
```

CubeController.cs (new input system)

```
public class CubeController : MonoBehaviour
{
    public PlayerInputActions playerInputActions;
    public float velocity = 5f;
    void Update()
    {
        UseNewInputSystem();
    }

    private void UseNewInputSystem()
    {
        if (playerInputActions == null)
        {
            playerInputActions = new();
            playerInputActions.Enable();
        }

        var movement = playerInputActions.Player.Movement.ReadValue<Vector2>();
        var look = playerInputActions.Player.Look.ReadValue<Vector2>().x;

        MoveAndRotate(movement, look);
    }

    private void MoveAndRotate(Vector2 movementDelta, float rotationDelta)
    {
        var position = transform.position;
        position.x += velocity * movementDelta.x;
        position.z += velocity * movementDelta.y;
        transform.position = Vector3.Lerp(transform.position, position, Time.deltaTime);

        var rotation = transform.rotation.eulerAngles;
        rotation.y += rotationDelta;
        transform.rotation = Quaternion.Euler(rotation);
    }
}
```

Platform Specific Inputs in Unity (cont.)

- VR games require more advanced interactions
- VR is out of the scope of this course but if you want to learn how create VR games:
  - https://www.youtube.com/@ValemVR/videos



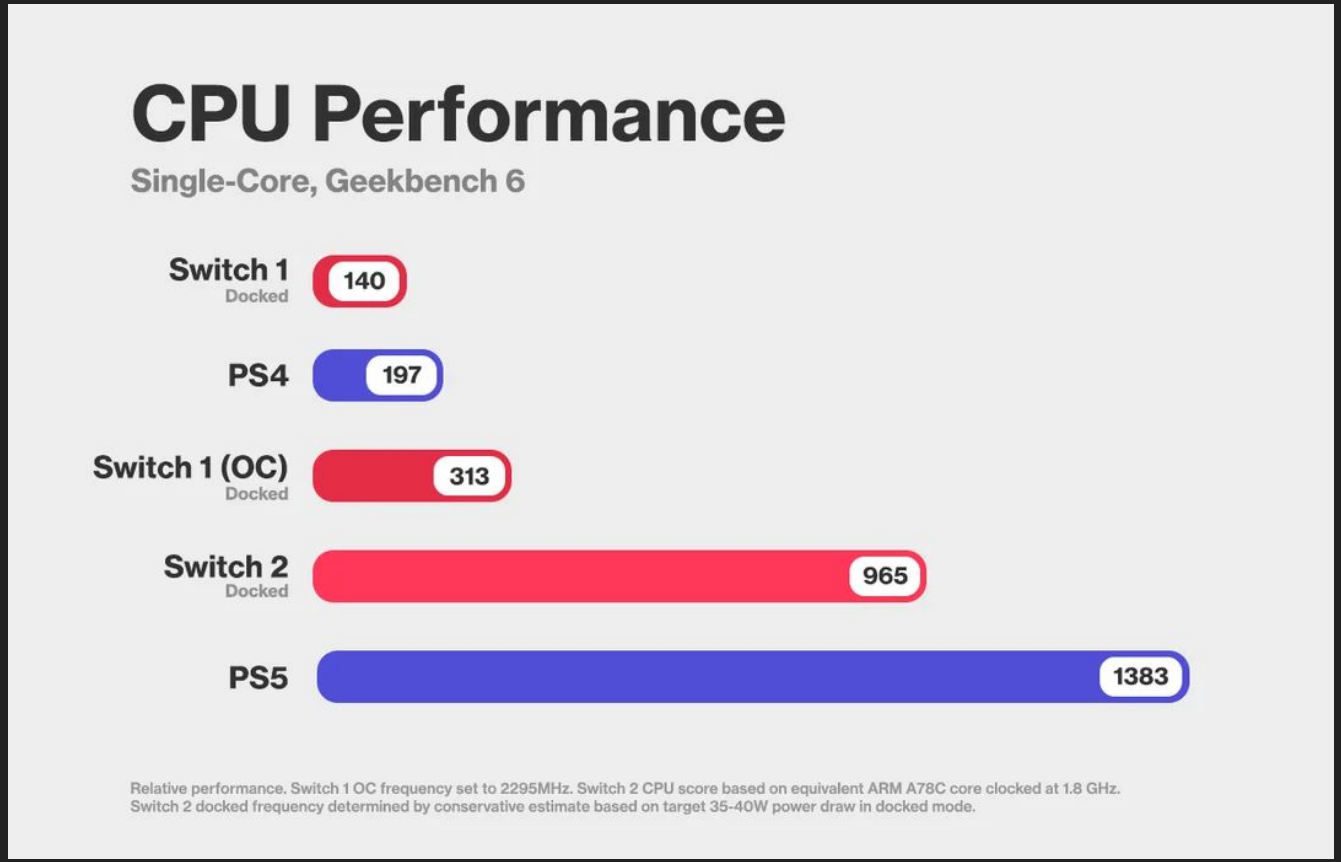
Hardware Capabilities

- It is important to test often on your target hardware with minimum specifications
  - https://youtu.be/C7fExZx1QyU?si=bfACIjt_PUWkDpeC&t=297
- Some platforms have variable system specifications
  - Everyone's PC is different
  - Everyone's smartphone is different
- Some platforms have consistent system specifications
  - Everyone's Nintendo Switch is the same
- What runs on a gaming PC might not necessarily run on a handheld device or console
- Power consumption may be a significant factor if your game is on mobile or handheld
  - PC gamers will also be wary if your game is unnecessarily power hungry
- Only one way to know: test often on actual hardware

Hardware Capabilities (cont.)

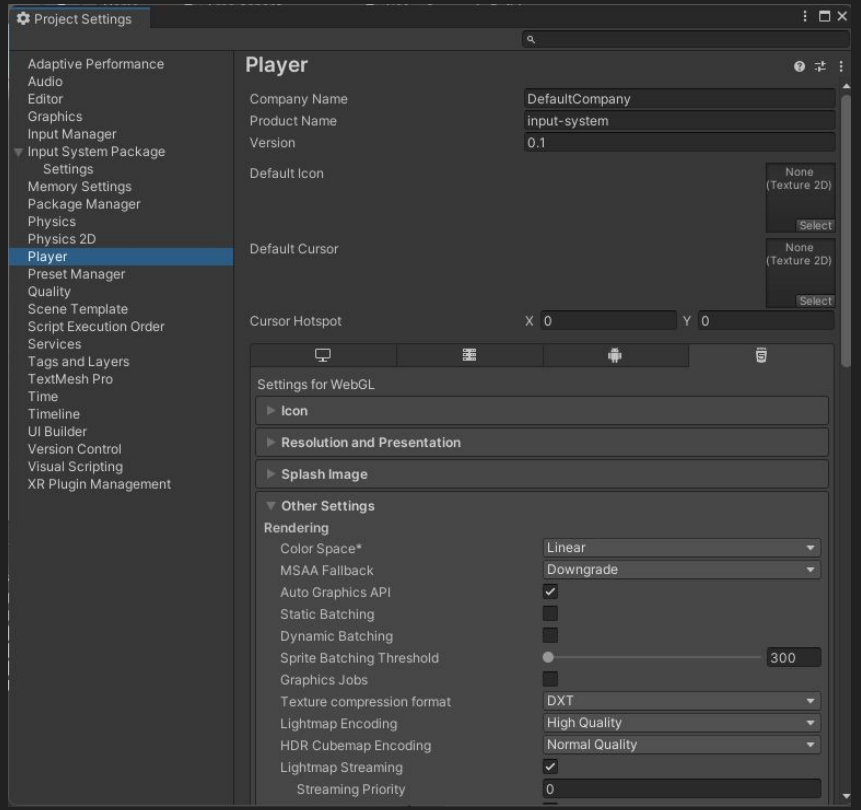
Property	PS5 Physical	PS5 Digital	XSX	XSS
CPU Type	Custom Zen 2	Custom Zen 2	Custom Zen 2	Custom Zen 2
CPU Cores	8	8	8	8
CPU Frequency	3.5 Ghz Variable	3.5 Ghz Variable	3.8 Ghz	3.6 Ghz
CPU Frequency w/ SMT	3.5 Ghz Variable	3.5 Ghz Variable	3.66 Ghz	3.4 Ghz
GPU	Custom RDNA 2	Custom RDNA 2	Custom RDNA 2	Custom RDNA 2
TFLOPS	10.3	10.3	12	4
GPU CU	36	36	52	20
GPU CU Frequency	2.23 Ghz	2.23 Ghz	1.825 Ghz	1.565 Ghz
SoC Die Size	360.45 mm2	360.45 mm2	360.45 mm2	197.05 mm2
Process Node	7nm Enhanced	7nm Enhanced	7nm Enhanced	7nm Enhanced
Memory Size	16 GB	16 GB	16 GB	10 GB
Bus Width	256 Bit	256 Bit	320 Bit	128 Bit
Memory Type	GDDR6	GDDR6	GDDR6	GDDR6
Memory Bandwidth 1	16 GB @ 448 GB/s	16 GB @ 448 GB/s	10 GB @ 560 GB/s	8 GB @ 224 GB/s
Memory Bandwidth 2	NA	NA	6 GB @ 336 GB/s	2 GB @ 56 GB/s
Internal Storage	825 GB Custom NVMe	825 GB Custom NVMe	1 TB Custom NVMe	512 GB Custom NVMe
I/O Throughput (Raw)	5.5 GB/s	5.5 GB/s	2.4 GB/s	2.4 GB/s
I/O Throughput (Compressed)	8-9 GB/s	8-9 GB/s	4.8 GB/s	4.8 GB/s
Expandable Storage	NVMe Expansion Slot	NVMe Expansion Slot	1 TB Properietary Seagate / Alternate 3.1	1 TB Properietary Seagate / Alternate 3.1
Audio	Tempest 3D	Tempest 3D	Dolby 5.1, DTS, Atoms, 7.1 LPCM	Dolby 5.1, DTS, Atoms, 7.1 LPCM
HDMI	1 x HDMI 2.1	1 x HDMI 2.1	1 x HDMI 2.1	1 x HDMI 2.1
USB	USB 3.1 Gen 2 (USB-C)	USB 3.1 Gen 2 (USB-C)	USB 3.1 Gen 1	USB 3.1 Gen 1
USB Throughput	10 Gbps	10 Gbps	5 Gbps	5 Gbps
Wireless	802.11ax (Wifi 6)	802.11ax (Wifi 6)	802.11ac Dual Band (Wifi 5)	802.11ac Dual Band (Wifi 5)
Ethernet	Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet
Dimensions	26cm (D) x 10.4cm (B) x 39.12cm (H)	26cm (D) x 9.2cm (B) x 39.12cm (H)	15.1cm (D) x 15.1cm (B) x 30.1cm (H)	6.5cm (D) x 15.1cm (B) x 27.5cm (H)
Weight	9.92 lbs	8.6 lbs	9.8 lbs	4.25 lbs
Price	**$499**	**$399**	**$499**	**$299**
Sources:				
Playstation.com				
Xbox.Com				
Eurogamer.net				

Hardware Capabilities (cont.)



Cross Platform Development

- You may want to release your game on multiple platforms
- You may want to NOT rewrite everything for a new platform
  - Unity makes configuring different platforms easy
    - Project Settings -> Player
- Other game engines are not so lucky
  - If you’re making your own engine, good luck
  - Interfacing directly with the hardware for multiple platforms is hard
  - SDL, Sokol, Raylib, etc.



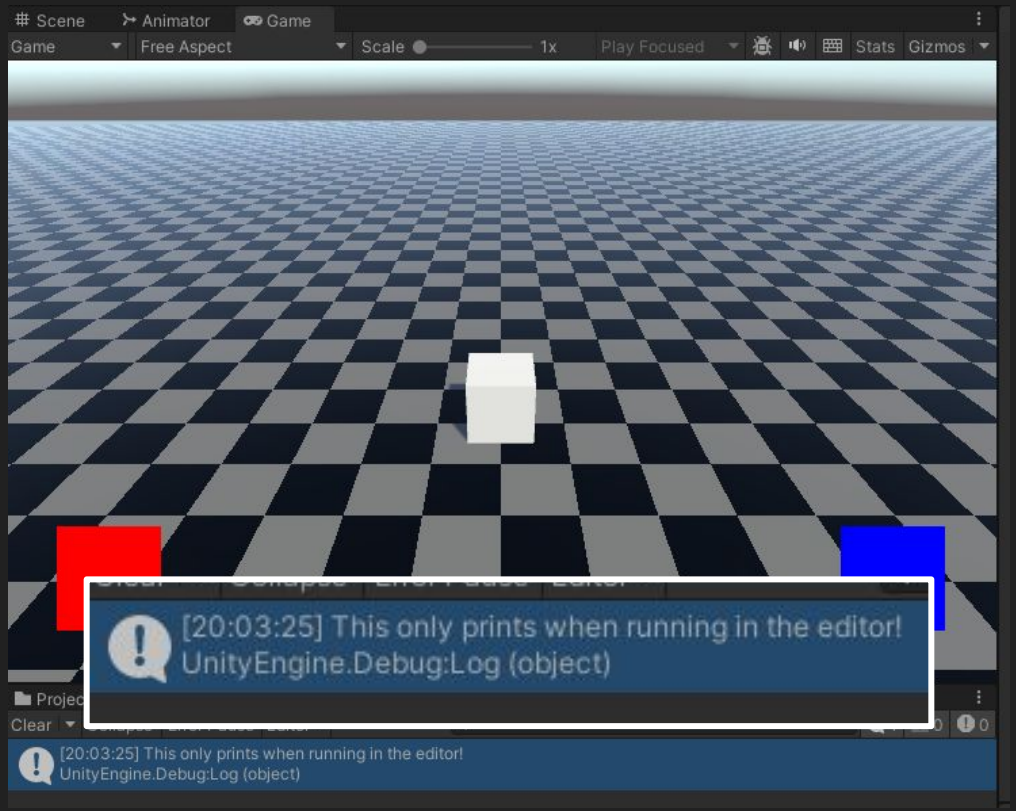
Cross Platform Development

- Conditional compilation is sometimes required for different platforms
  - Example: the libraries for interfacing with iOS are not available on PC and thus won’t compile
- Use conditional compilation

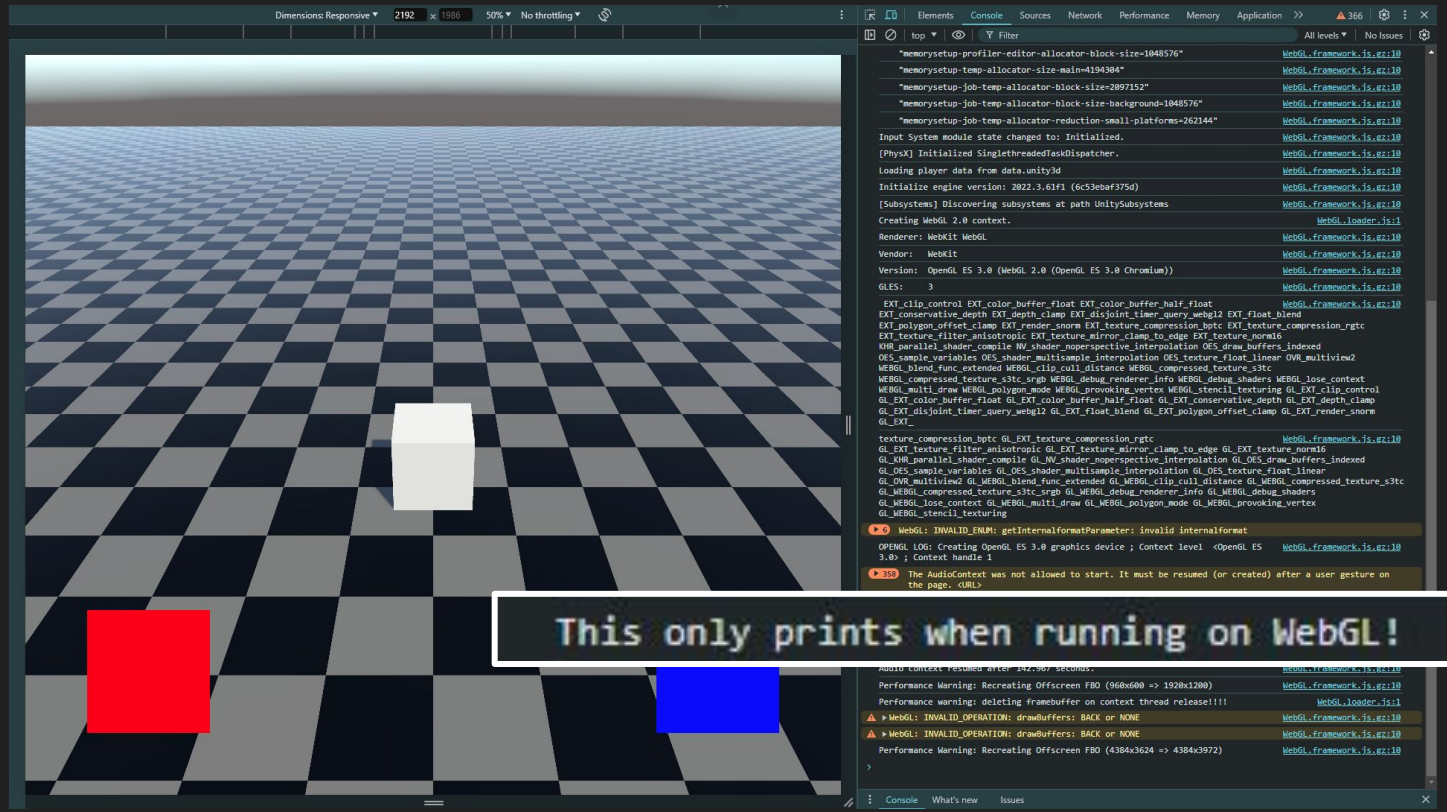
PlatformDetector.cs

```
public class PlatformDetector : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
#if UNITY_EDITOR
        Debug.Log("This only prints when running in the editor!");
#elif UNITY_STANDALONE_WIN
        Debug.Log("This only prints when running on Windows!");
#elif UNITY_STANDALONE_LINUX
        Debug.Log("This only prints when running on Linux!");
#elif UNITY_STANDALONE_OSX
        Debug.Log("This only prints when running on MacOS!");
#elif UNITY_IOS
        Debug.Log("This only prints when running on iOS!");
#elif UNITY_ANDROID
        Debug.Log("This only prints when running on Android!");
#elif UNITY_WEBGL
        Debug.Log("This only prints when running on WebGL!");
#endif
    }
}
```

Cross Platform Development (cont.)



Cross Platform Development (cont.)



Cross Platform Development (cont.)

- Conditional compilation is also useful for debug code that can only be accessed in the editor
- Since “EditorUtility” can only be accessed within the editor it requires conditional compilation
  - Otherwise, it would not compile for any platform!

BuildHelper.cs

```
#if UNITY_EDITOR
using UnityEditor;
#endif
using UnityEngine;

public class BuildHelper : MonoBehaviour
{
    void Start()
    {
#if UNITY_EDITOR
        EditorUtility.DisplayDialog("Hello", "This is an editor-only dialog.", "OK");
#endif
    }
}
```

Porting Existing Games

- Porting games is converting games to run on other platforms
  - Allows for re-releasing older games
  - Allows for reaching new audiences
  - Requires a deep understanding of the target hardware and existing game code
- Often times, porting games is NOT simple and can be very expensive
  - Even AAA companies fail at this miserably

Porting Existing Games

- Porting games is converting games to run on other platforms
  - Allows for re-releasing older games
  - Allows for reaching new audiences
  - Requires a deep understanding of the target hardware and existing game code
- Often times, porting games is NOT simple and can be very expensive
  - Even AAA companies fail at this miserably
  - https://www.youtube.com/watch?v=W0-NEwh1m2c