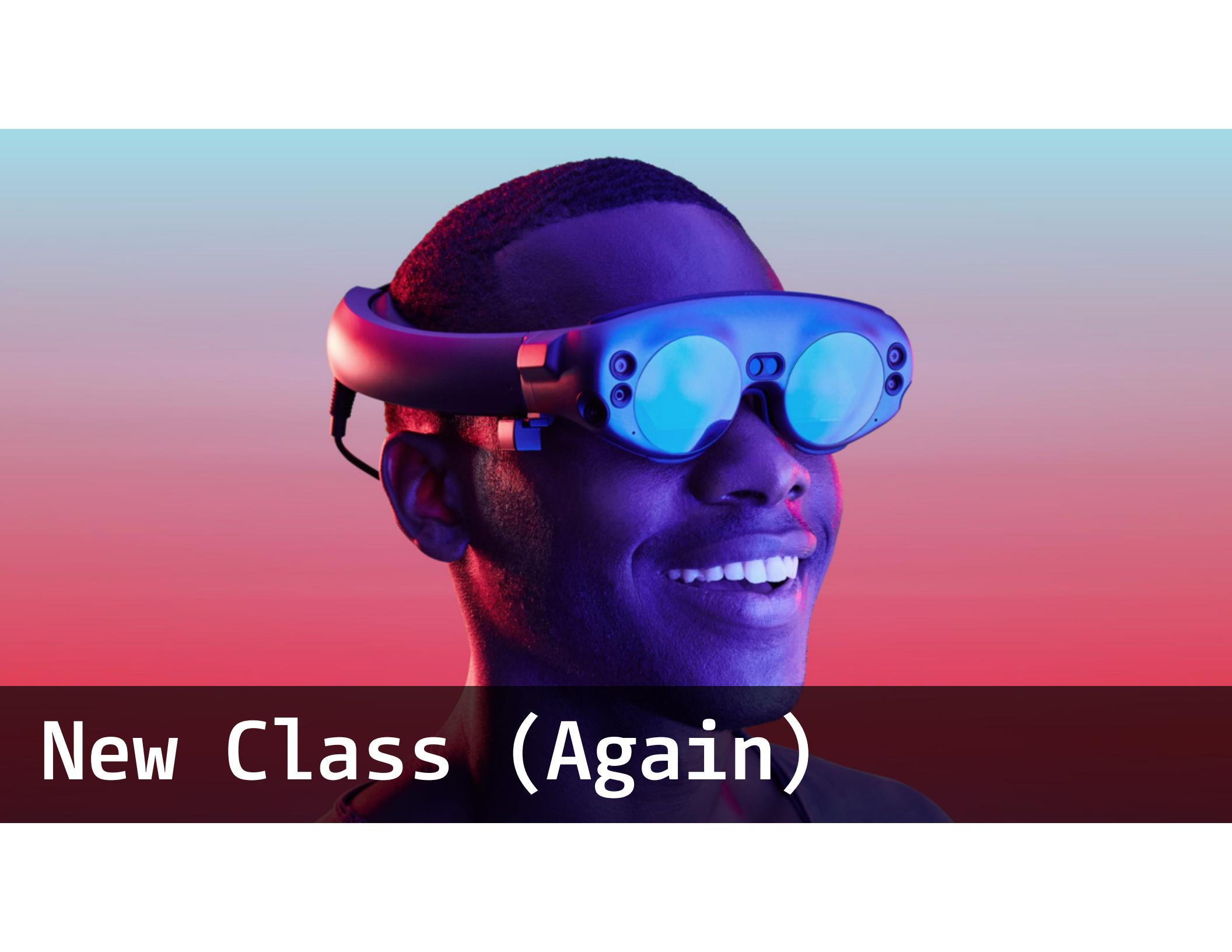


TECH 3707 - Advanced Mixed Reality

A close-up profile shot of a person wearing a VR headset. The person has dark skin and is smiling. The VR headset has large, reflective blue lenses and a black headband with red lighting. The background is a vibrant gradient from blue at the top to red at the bottom.

New Class (Again)

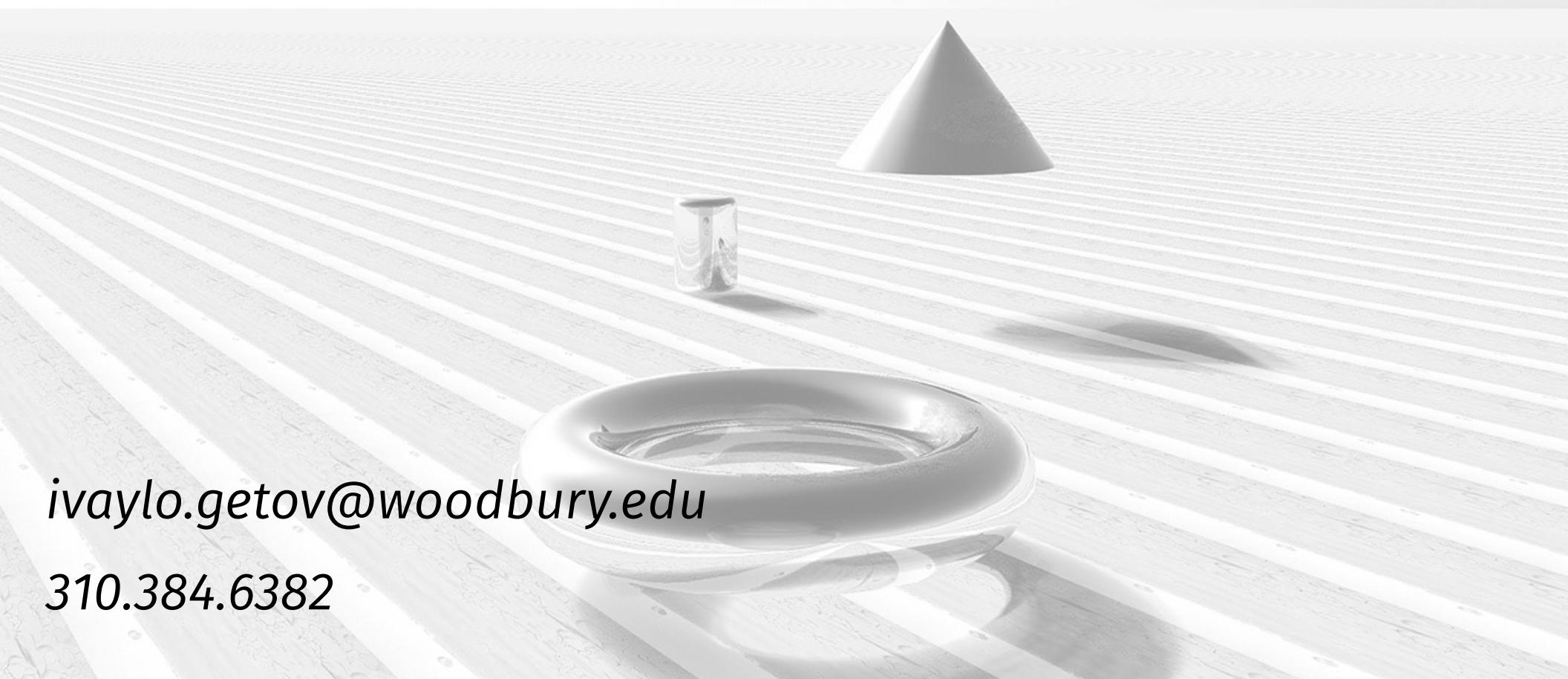
Most Up-to-date Syllabus:

<https://github.com/ivaylopg/AdvancedMixedRealityStudio>

Contact Me

ivaylo.getov@woodbury.edu

310.384.6382





ADVANCED?

pro·gram·mer

n. an organism that turns
caffeine and pizza
into software

desk keyboard



Broad Understanding About A Wide Range







The background features a dark navy blue gradient. In the foreground, there are two stylized, purple-colored wireframe mountain ranges. A large, solid red circle is positioned in the center, partially overlapping the mountains. The text is placed within this red circle.

Broad Understanding About A Wide Range

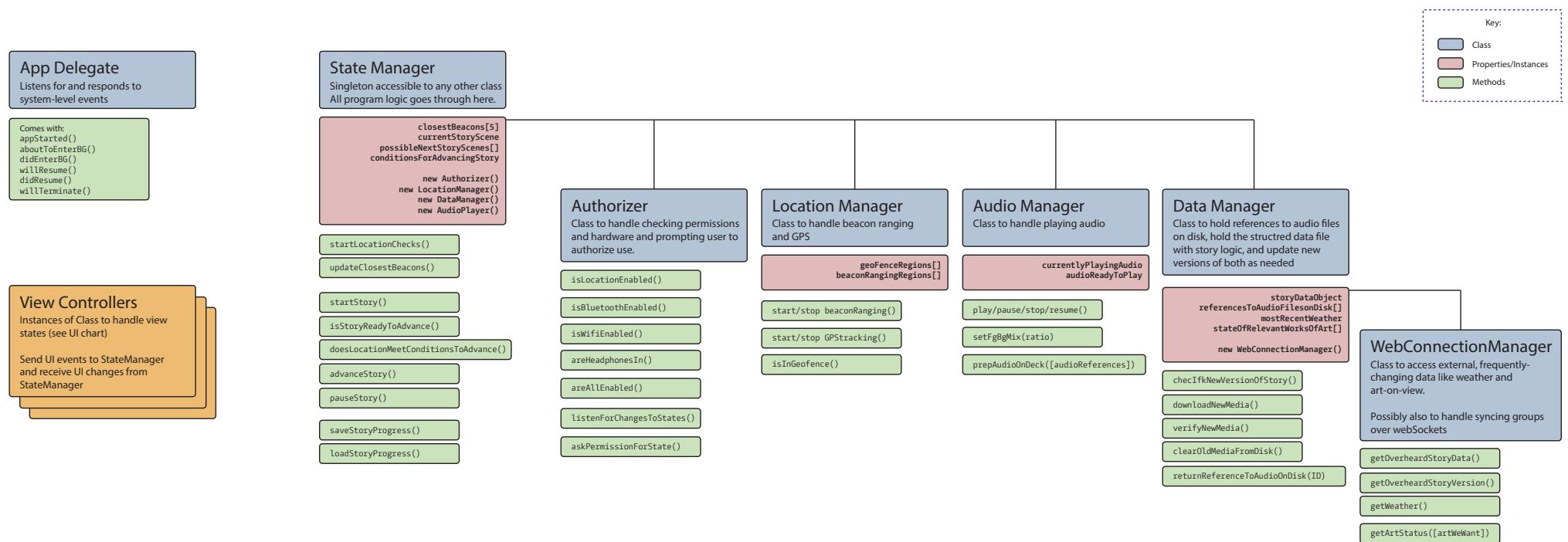
Deep Knowledge About
A Specific Area

Good Practices

Commented Code

Plan/Pseudocode

The project actually...uh...works.



State Manager

Singleton accessible to any other class
All program logic goes through here.

```
closestBeacons[5]
currentStoryScene
possibleNextStoryScenes[]
conditionsForAdvancingStory

new Authorizer()
new LocationManager()
new DataManager()
new AudioPlayer()
```

```
startLocationChecks()
```

```
updateClosestBeacons()
```

```
startStory()
```

```
isStoryReadyToAdvance()
```

```
doesLocationMeetConditionsToAdvance()
```

```
advanceStory()
```

```
pauseStory()
```

```
saveStoryProgress()
```

```
loadStoryProgress()
```

Authorizer

Class to handle checking permissions and hardware and prompting user to authorize use.

```
isLocationEnabled()
```

```
isBluetoothEnabled()
```

```
isWifiEnabled()
```

```
areHeadphonesIn()
```

```
areAllEnabled()
```

```
listenForChangesToStates()
```

```
askPermissionForState()
```

Location Manager

Class to handle beacon ranging and GPS

```
geoFenceRegions[]
beaconRangingRegions[]
```

```
start/stop beaconRanging()
```

```
start/stop GPStracking()
```

```
isInGeofence()
```

Audio Manager

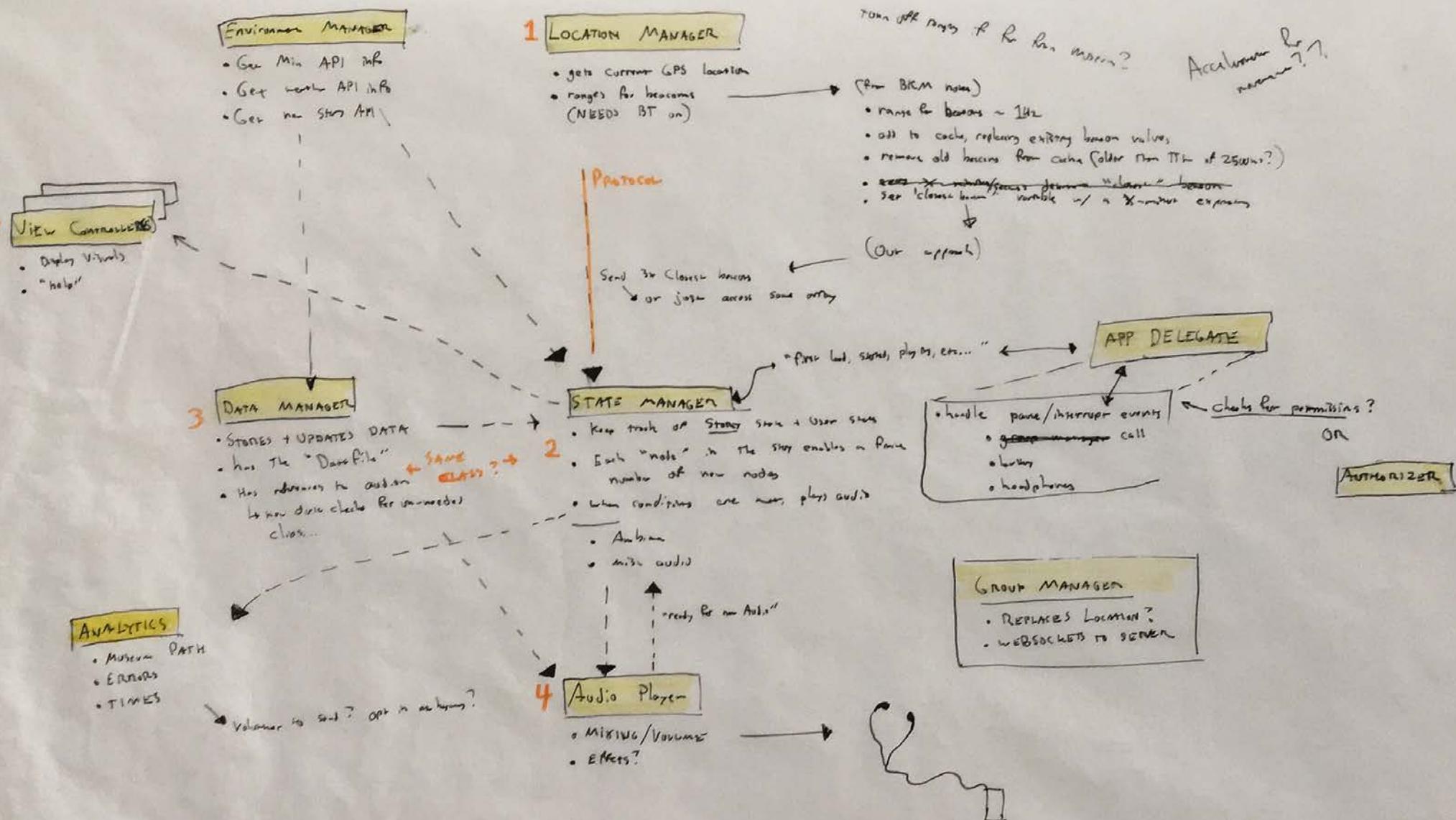
Class to handle playing audio

```
currentlyPl
audioR
```

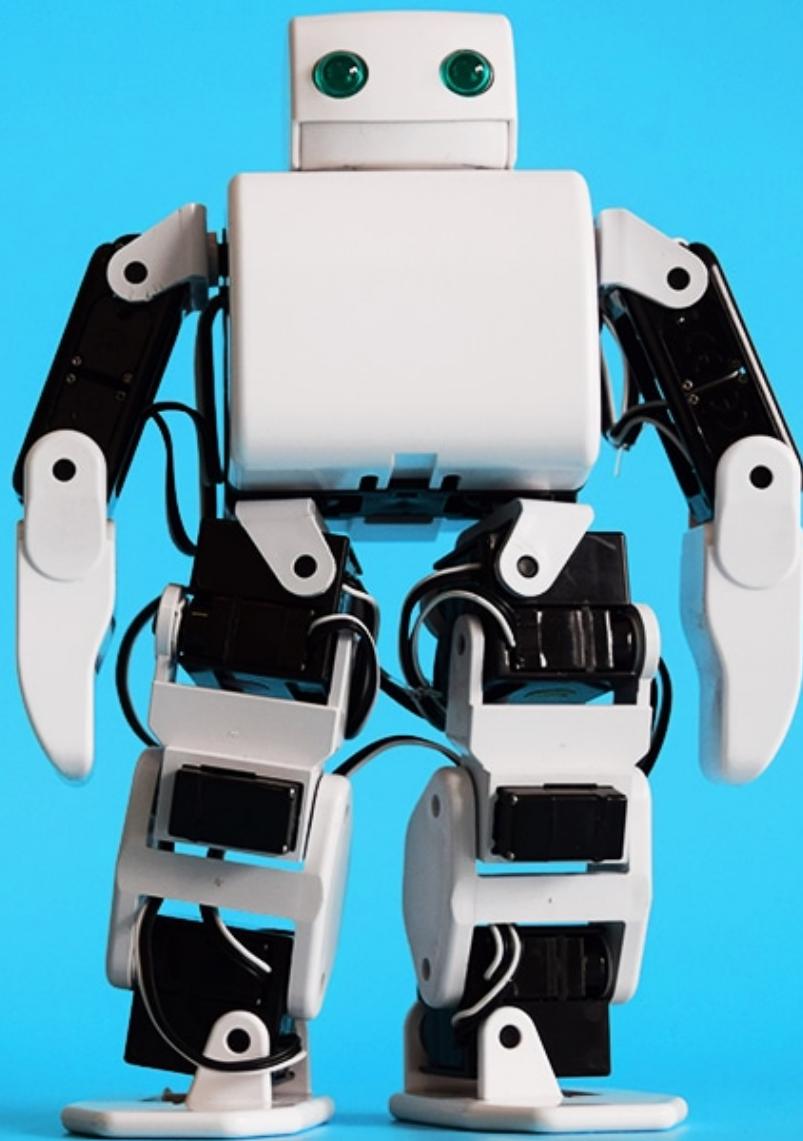
```
play/pause/stop/resume()
```

```
setFgBgMix(ratio)
```

```
prepAudioOnDeck([audioRefe
```







The background features a dark navy blue gradient. In the foreground, there are two large, stylized mountain peaks rendered with a low-poly, wireframe aesthetic in a muted purple color. Between the peaks, a large, solid red circle is positioned, resembling a rising sun or moon. A light gray grid floor is visible at the bottom.

What are we focusing on?



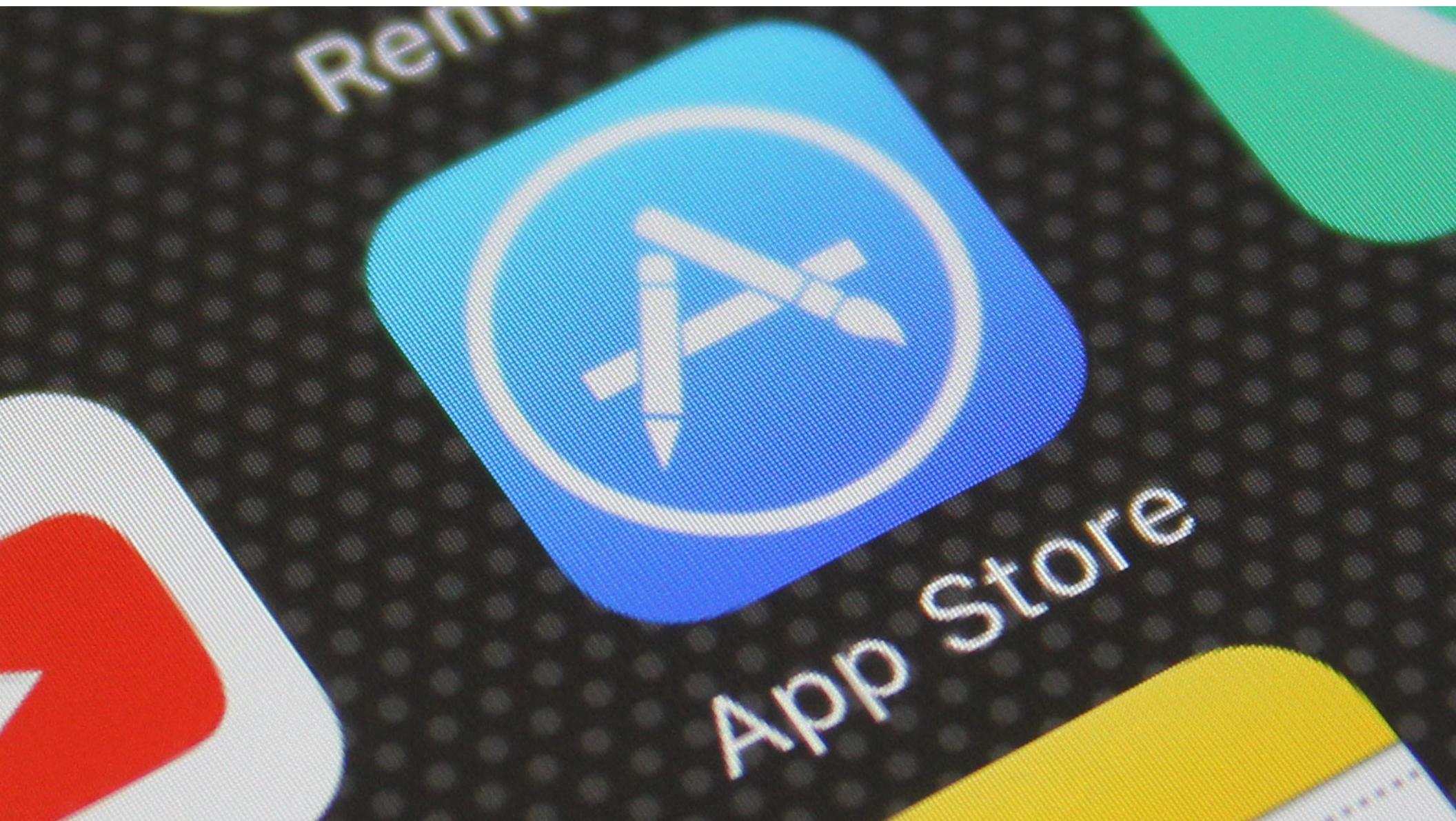


Powered by
Vuforia

© SCIONS OF HELIOS



Smarter Objects by Valentin Heun, Shunichi Kasahara, Pattie Maes - MIT Media Lab



Assignments

First Project - Interactive AR - 20%

Final Project - Shared AR Experience - 30%

Participation, attendance, and classwork - 50%

Format

Chapter 0 Recap Topics

Making sure everyone is back up to speed
and re-familiarized with Unity and C# development.

Format

Chapter 1 **Expanding on Fundamentals**

A series of class discussions, speakers, and exercises.
Deep dive into specific concepts and explore techniques.

Format

Chapter 2 **Interactive AR**

We will be developing a mobile AR application
based on the Vuforia framework

Format

Chapter 3 **Shared AR Experience**

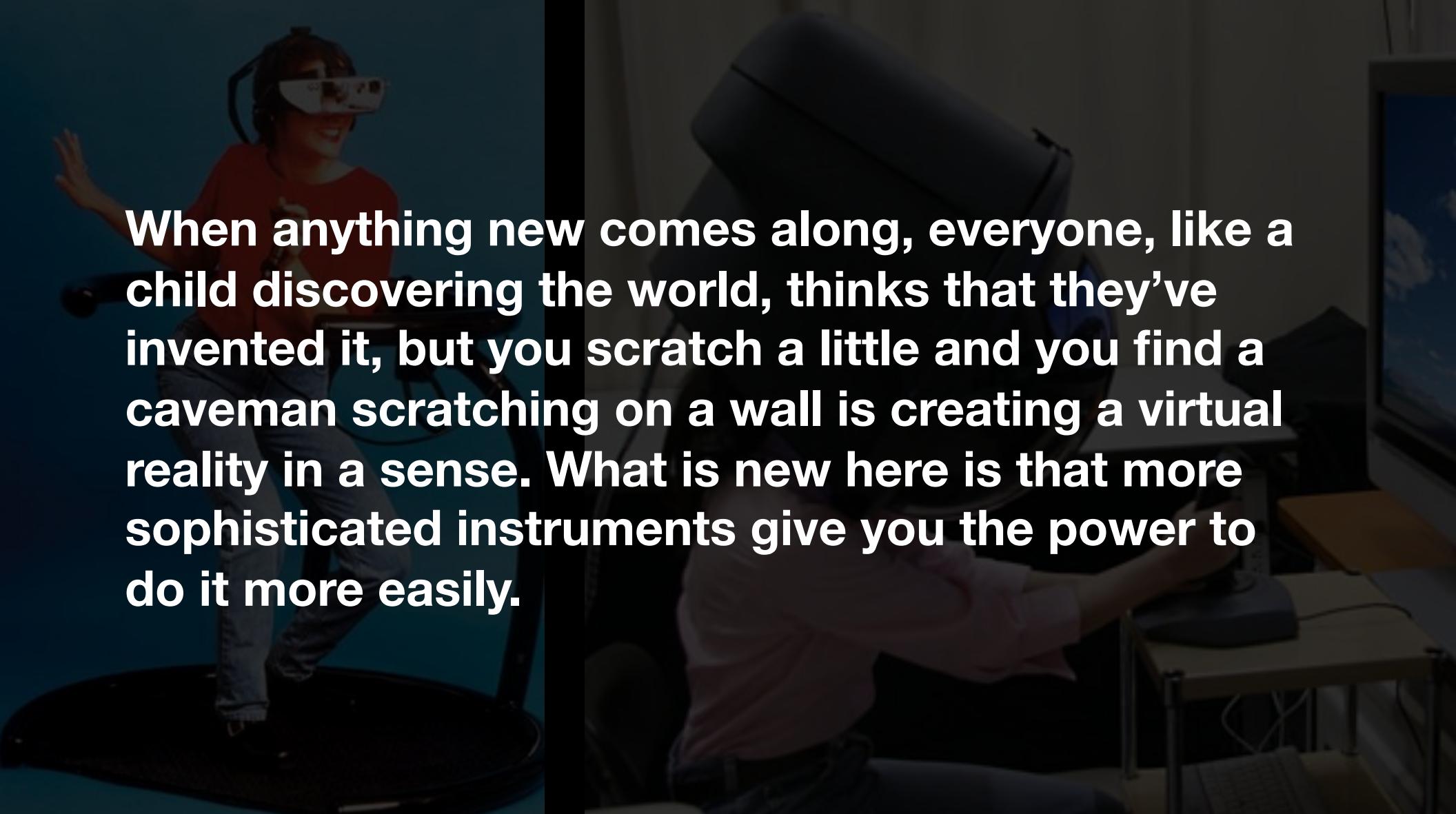
We will build on the previous project
to develop a multi-user mobile AR app





Let's Go!





When anything new comes along, everyone, like a child discovering the world, thinks that they've invented it, but you scratch a little and you find a caveman scratching on a wall is creating a virtual reality in a sense. What is new here is that more sophisticated instruments give you the power to do it more easily.

Augmented Reality

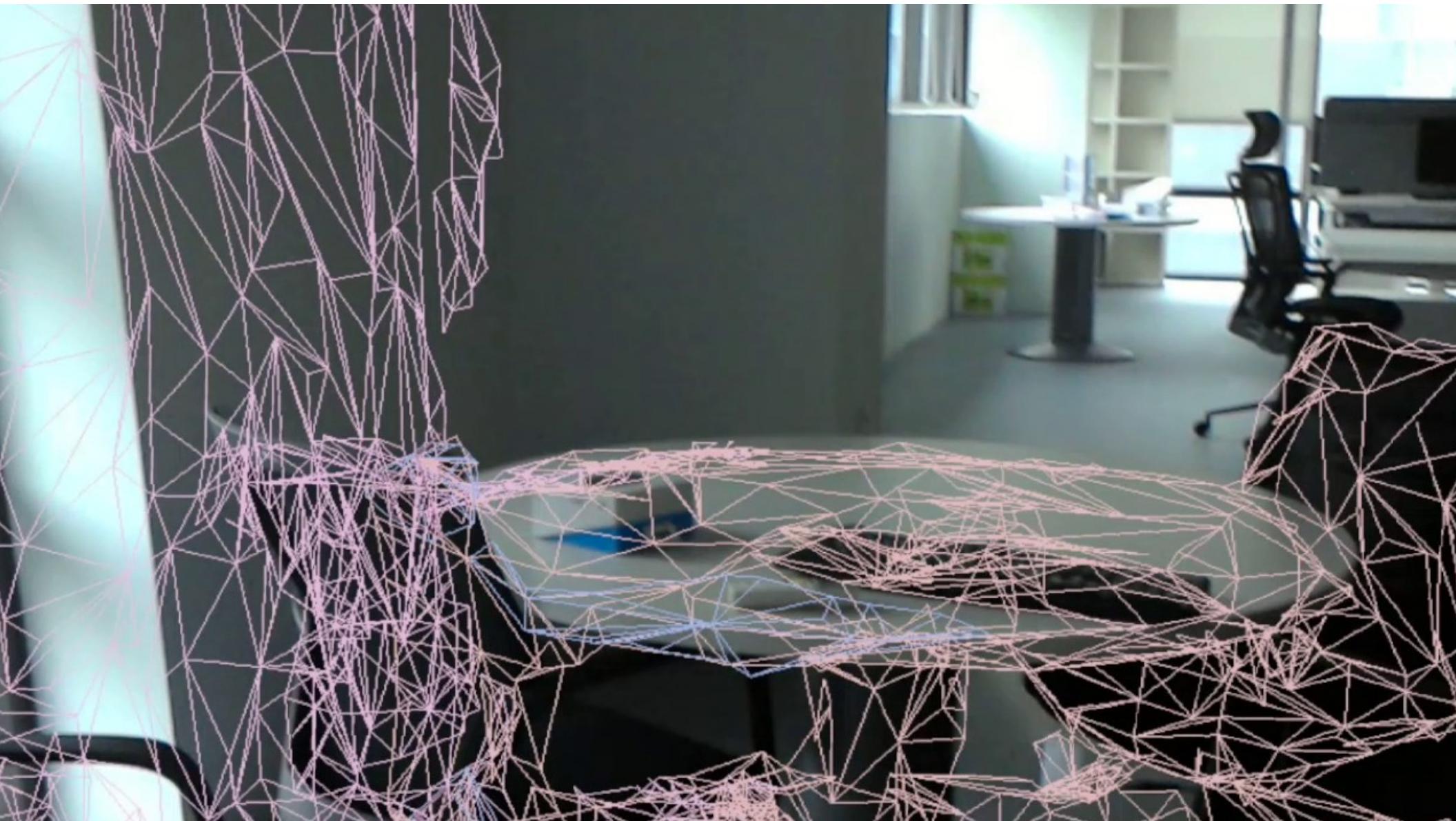


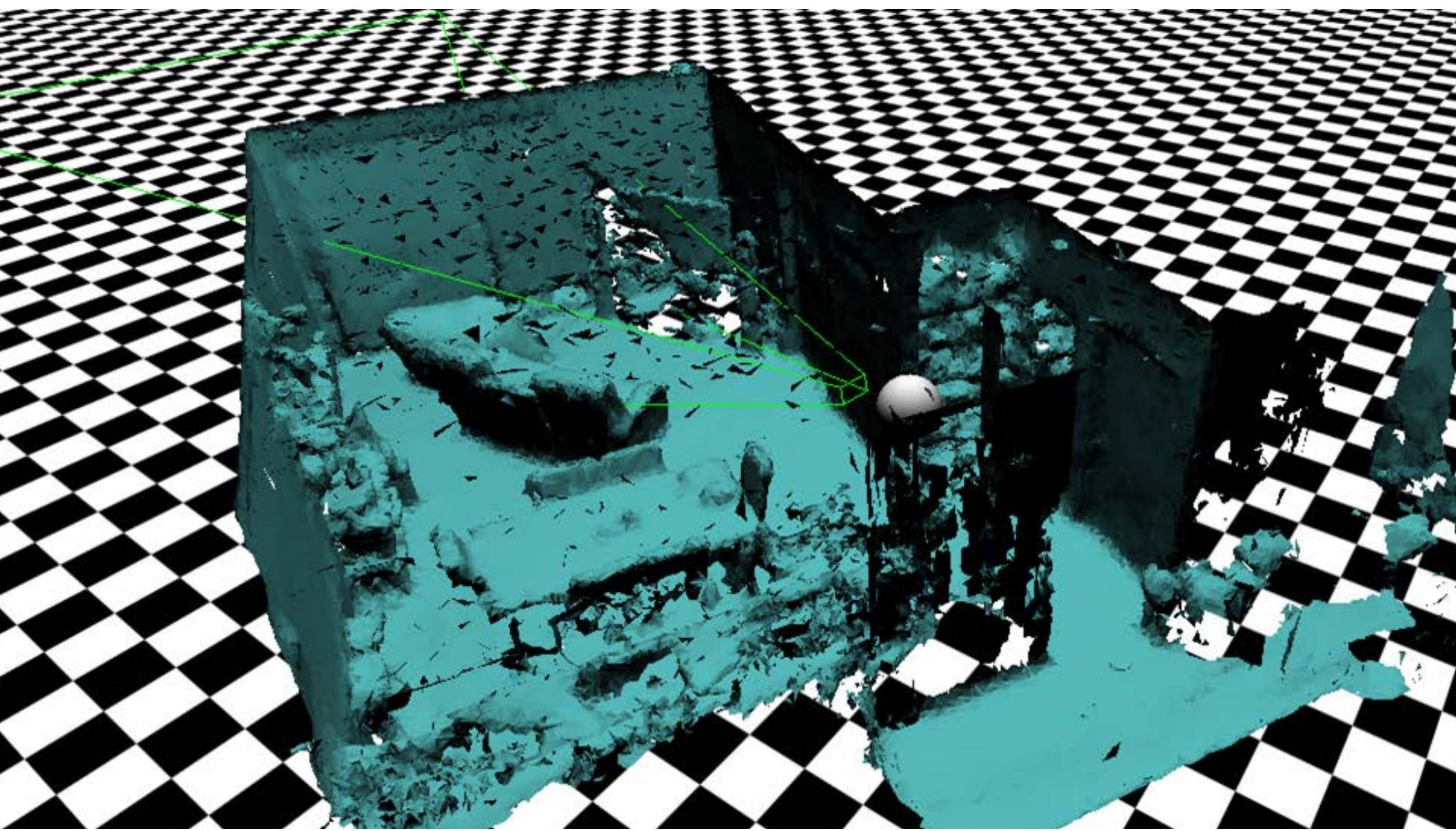
What is AR/VR/MR?





Inducing targeted behavior in an organism by using artificial sensory stimulation, while the organism has little or no awareness of the interference.









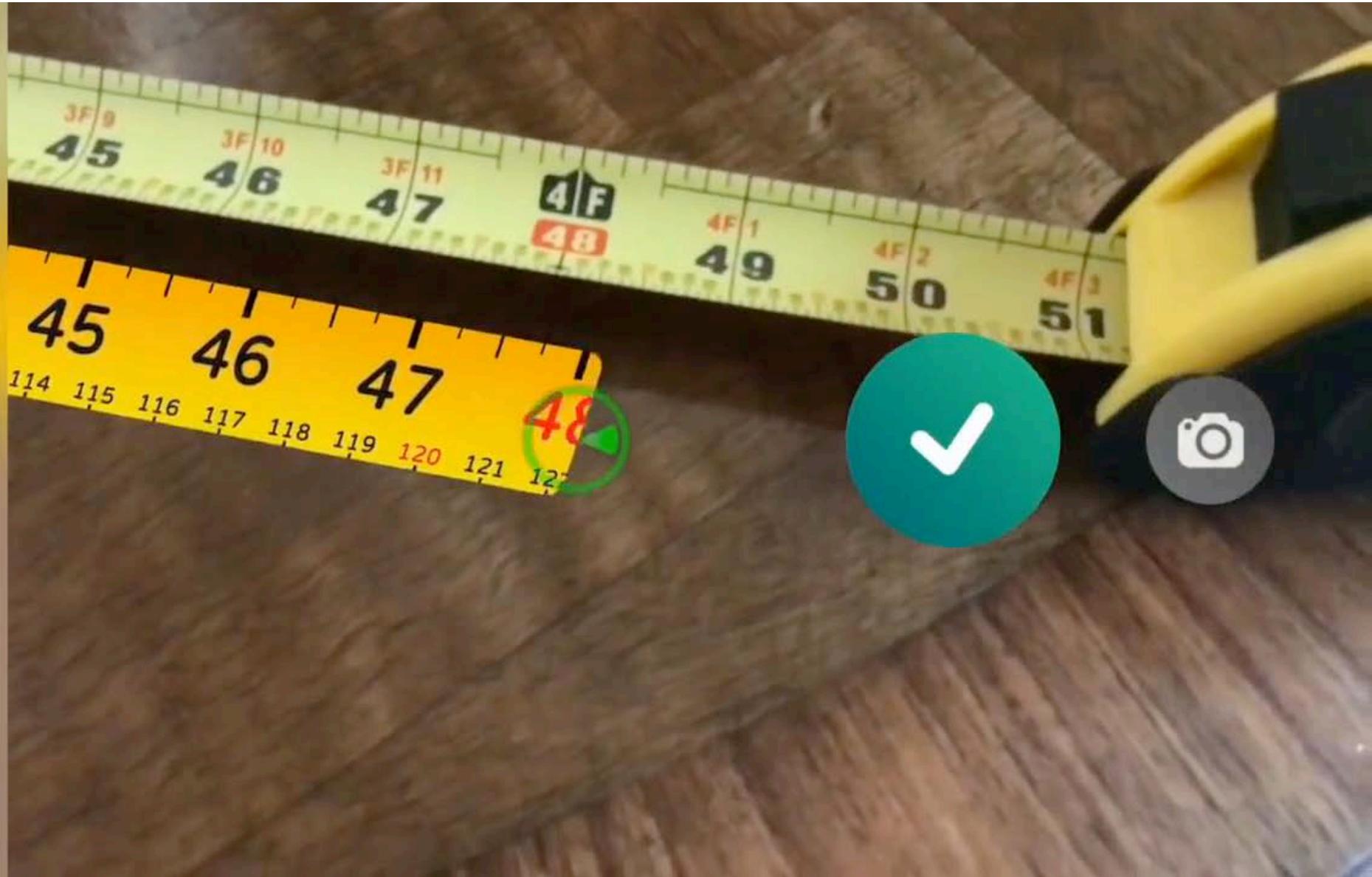




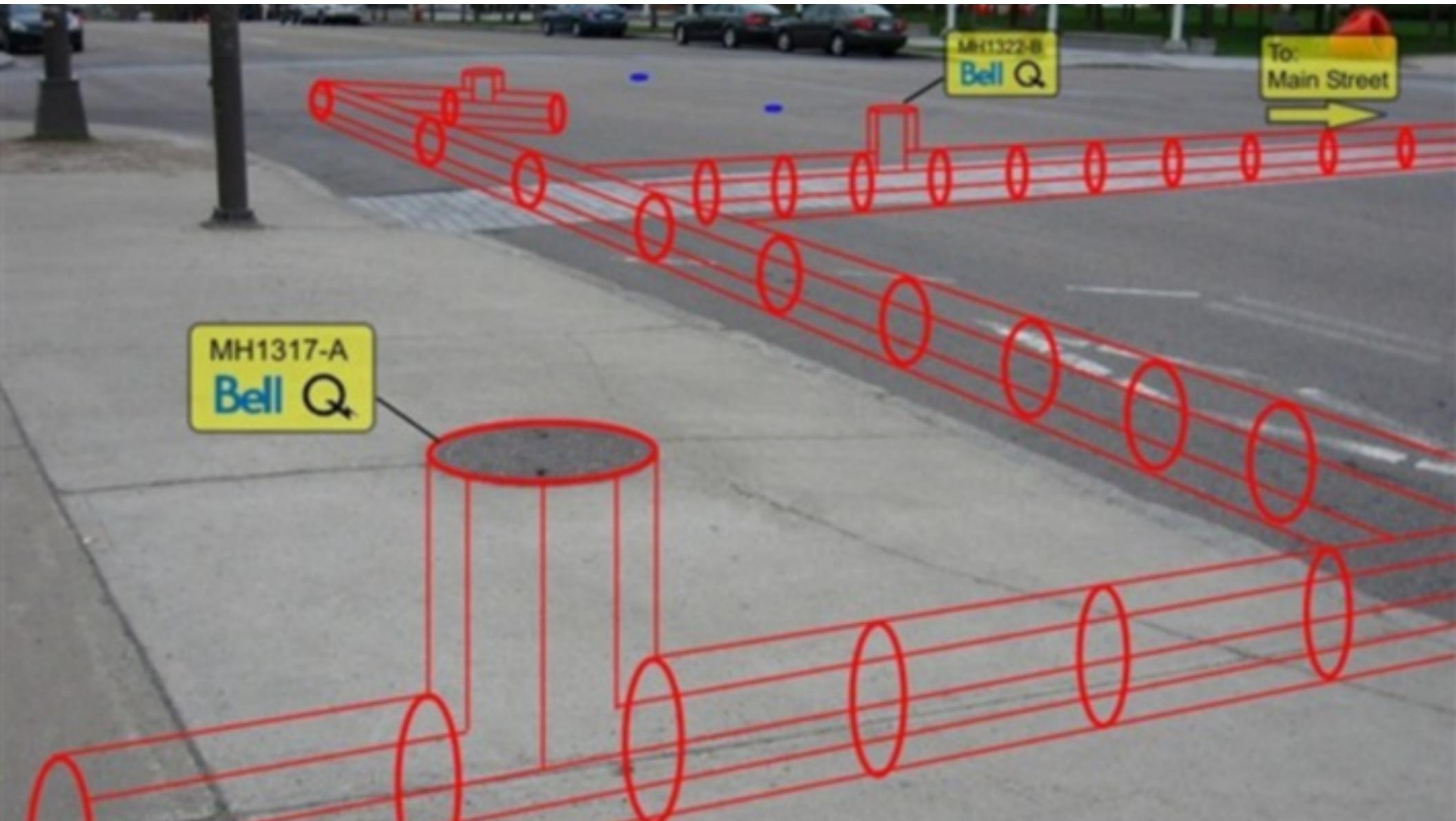
48.0 in

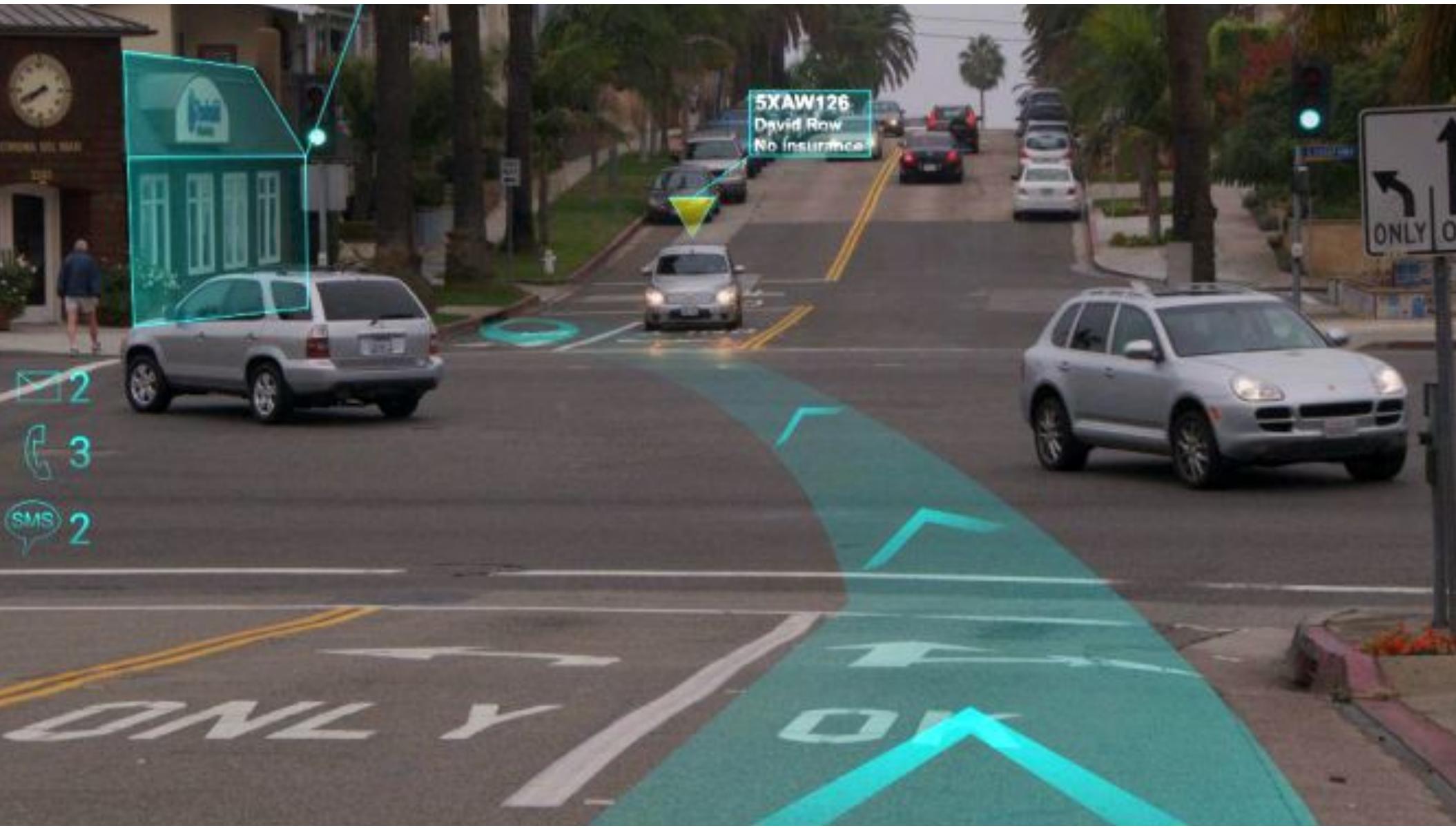
122 cm

METRIC



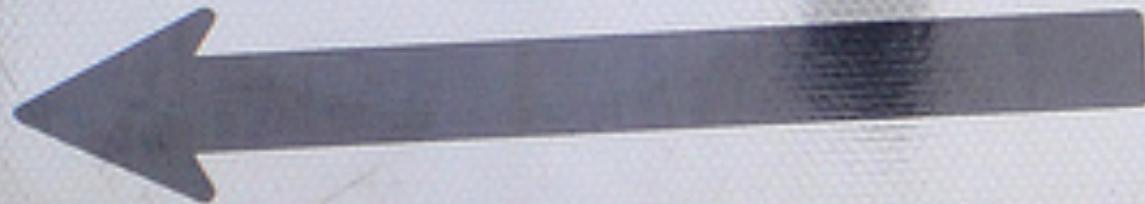




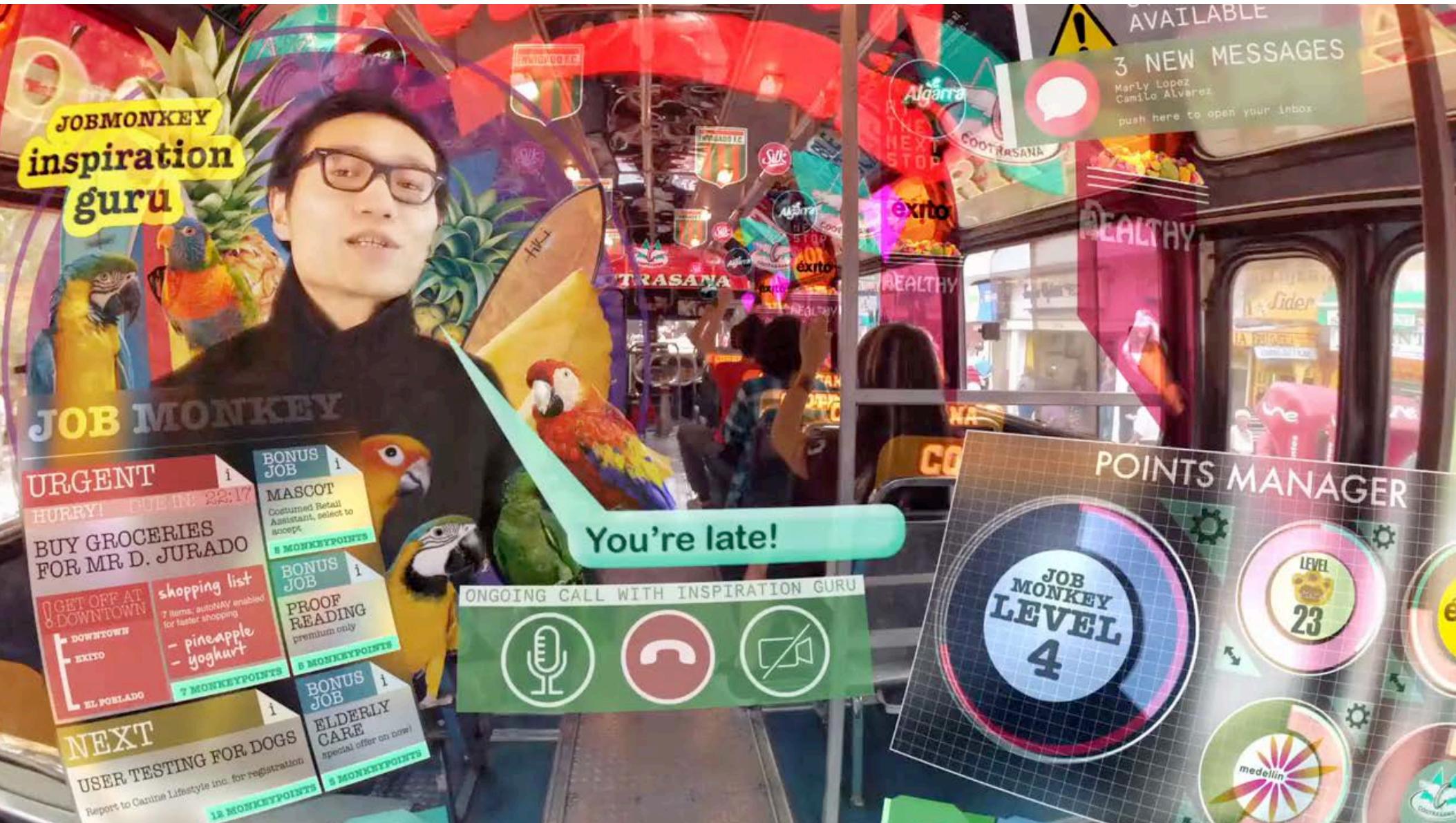




自転車を除く
一方通行





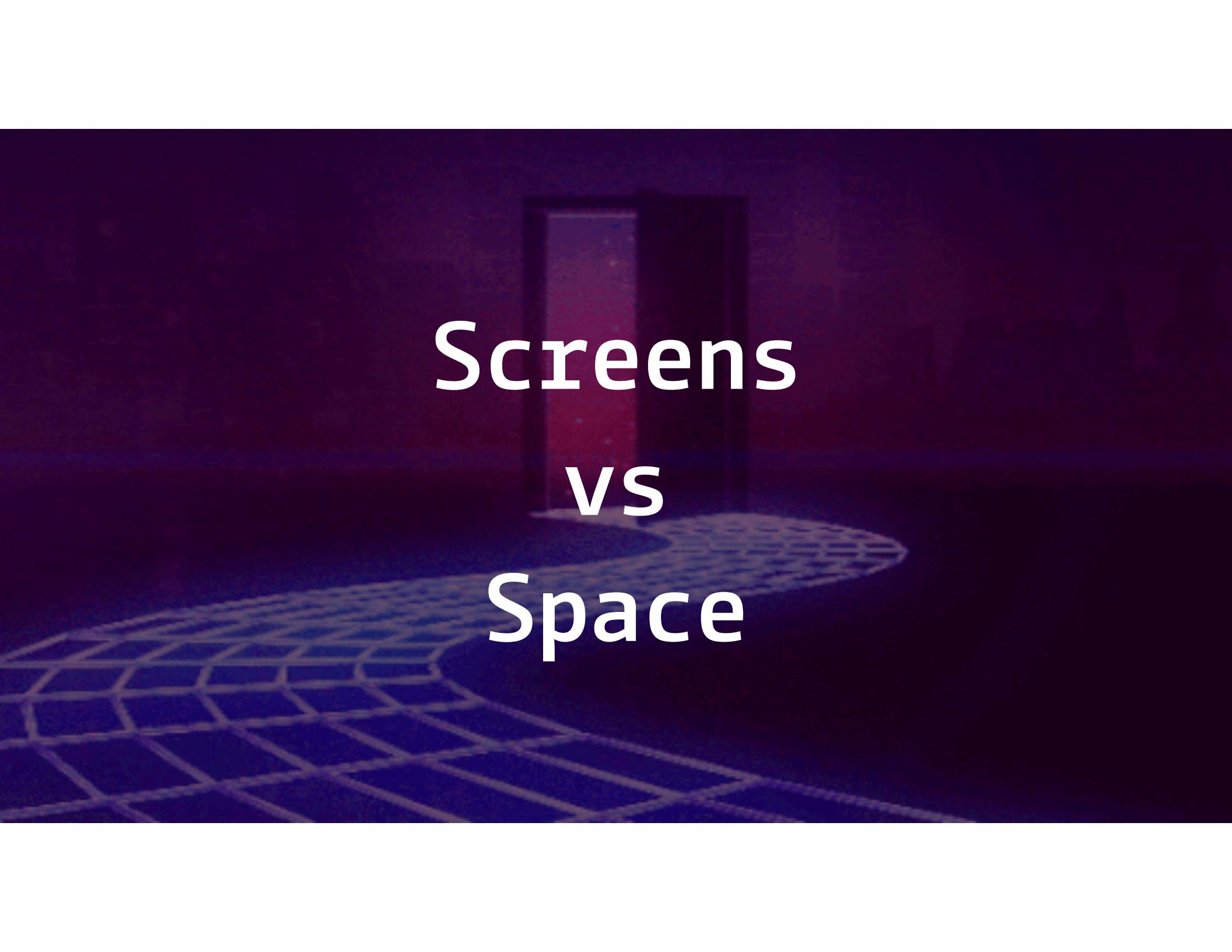






Expectations





Screens
vs
Space

Netflix

www.netflix.com/browse

NETFLIX

Browse DVD

Search Michelle

My List

NETFLIX CHEF'S TABLE

NETFLIX DAREDEVIL

NETFLIX BLOODLINE

NETFLIX UNBREAKABLE KIMMY SCHMIDT

NETFLIX MARCO POLO

NETFLIX

Marvel's Daredevil

★★★★★ 2015 TV-MA 1 Season

Blinded as a young boy, Matt Murdock fights injustice by day as a lawyer and by night as the Super Hero Daredevil in Hell's Kitchen, New York City.

Starring: Charlie Cox, Deborah Ann Woll, Vincent D'Onofrio
Genres: TV Shows, Comic Book & Superhero TV, Crime TV Shows
This show is: Exciting, Gritty, Dark

Law & Order: Criminal Intent star Vincent D'Onofrio plays Daredevil's nemesis Wilson Fisk, a.k.a. Kingpin.

MY LIST

OVERVIEW EPISODES MORE LIKE THIS DETAILS

A promotional image for the Netflix series 'Marvel's Daredevil'. It features Matt Murdock, played by Charlie Cox, in his signature black suit and mask, standing in front of a dark, cityscape background with blurred lights from buildings. A large white play button icon is overlaid on the right side of the image.





Private Settings

Off

On

Off

Press

Press

Press

UI and UX



Manipulation



A photograph of a museum gallery featuring several large-scale classical oil paintings in gold frames. A woman in a white cardigan and dark pants stands near a doorway, looking at a painting. A man in a red shirt and jeans walks away from the camera towards the right. In the foreground, a dark wooden railing with a decorative scroll pattern runs across the bottom. The floor is made of polished wood.

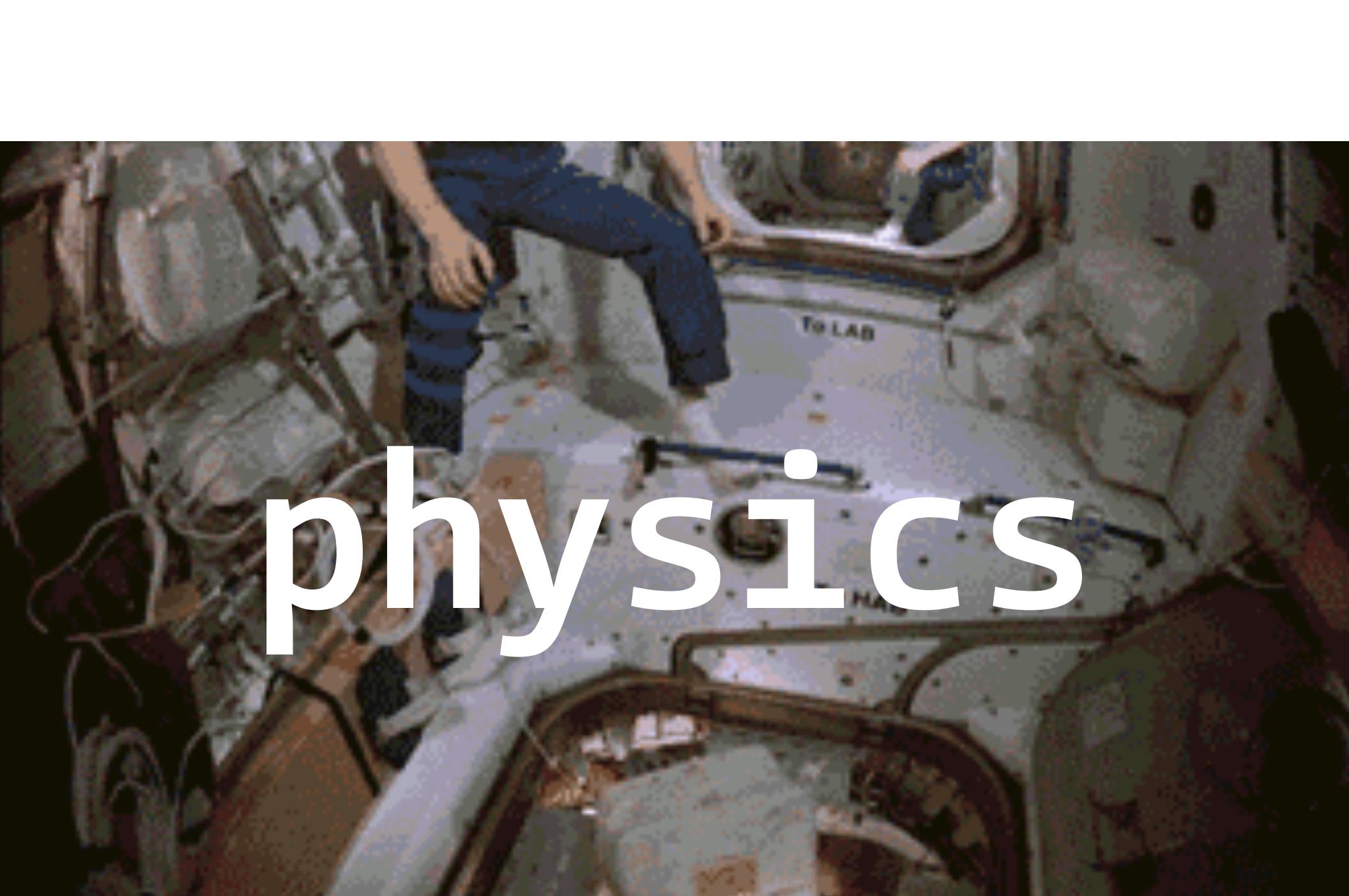
Exploration







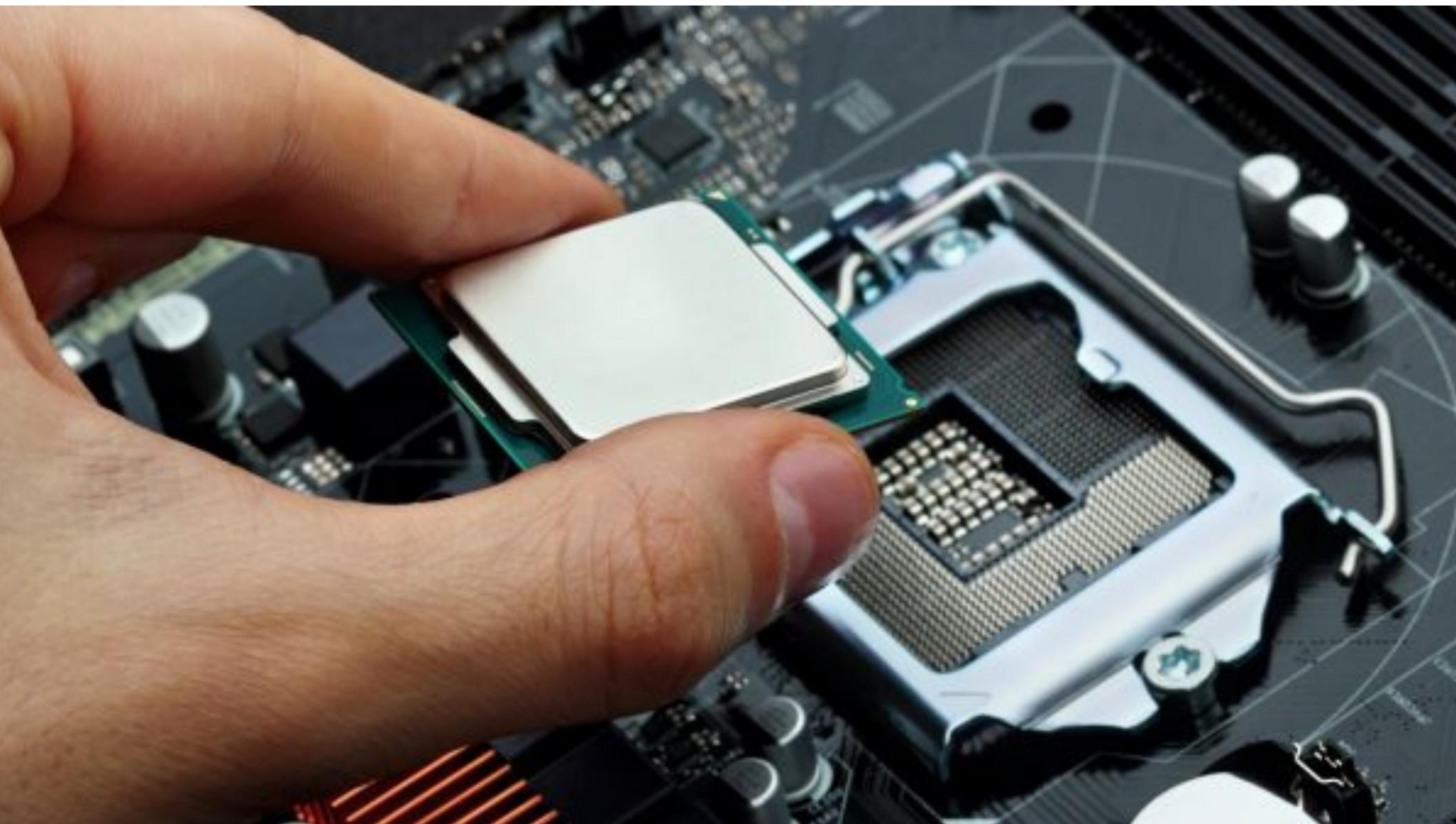
abstraction

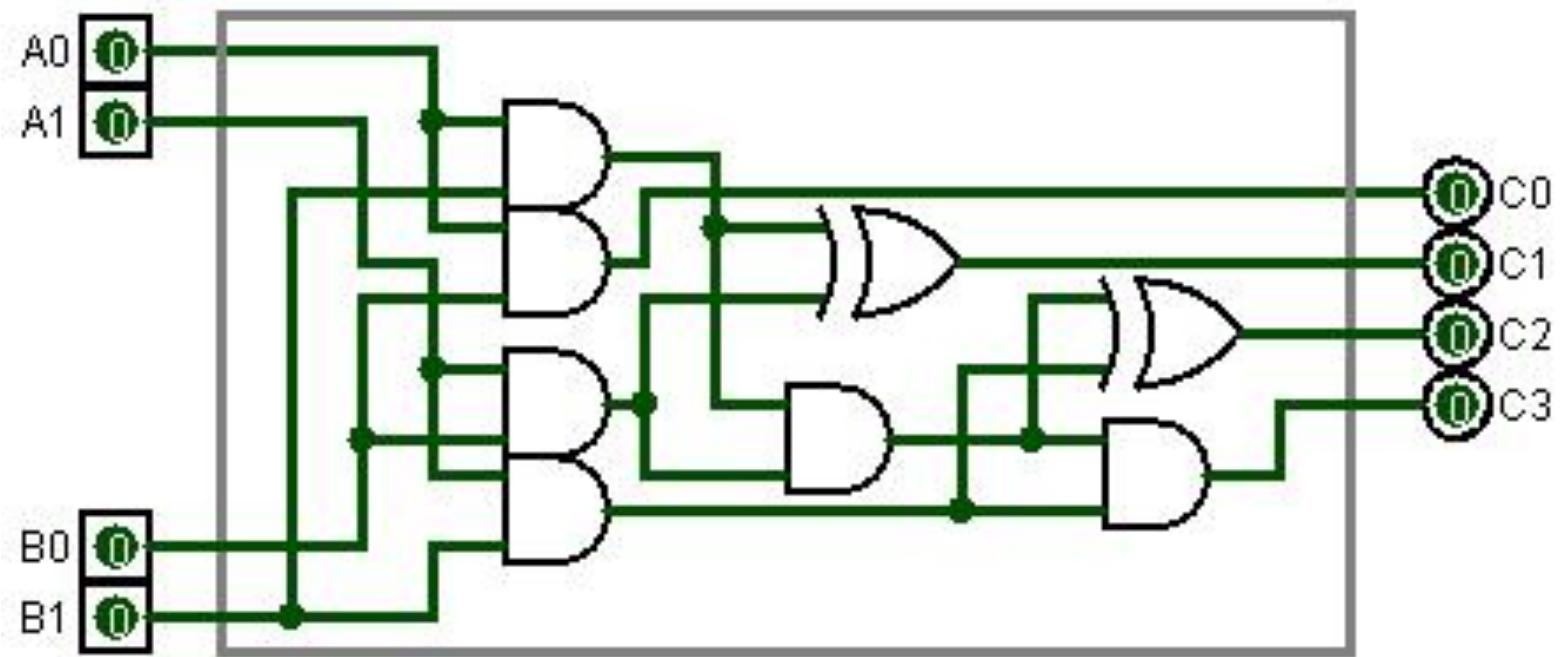


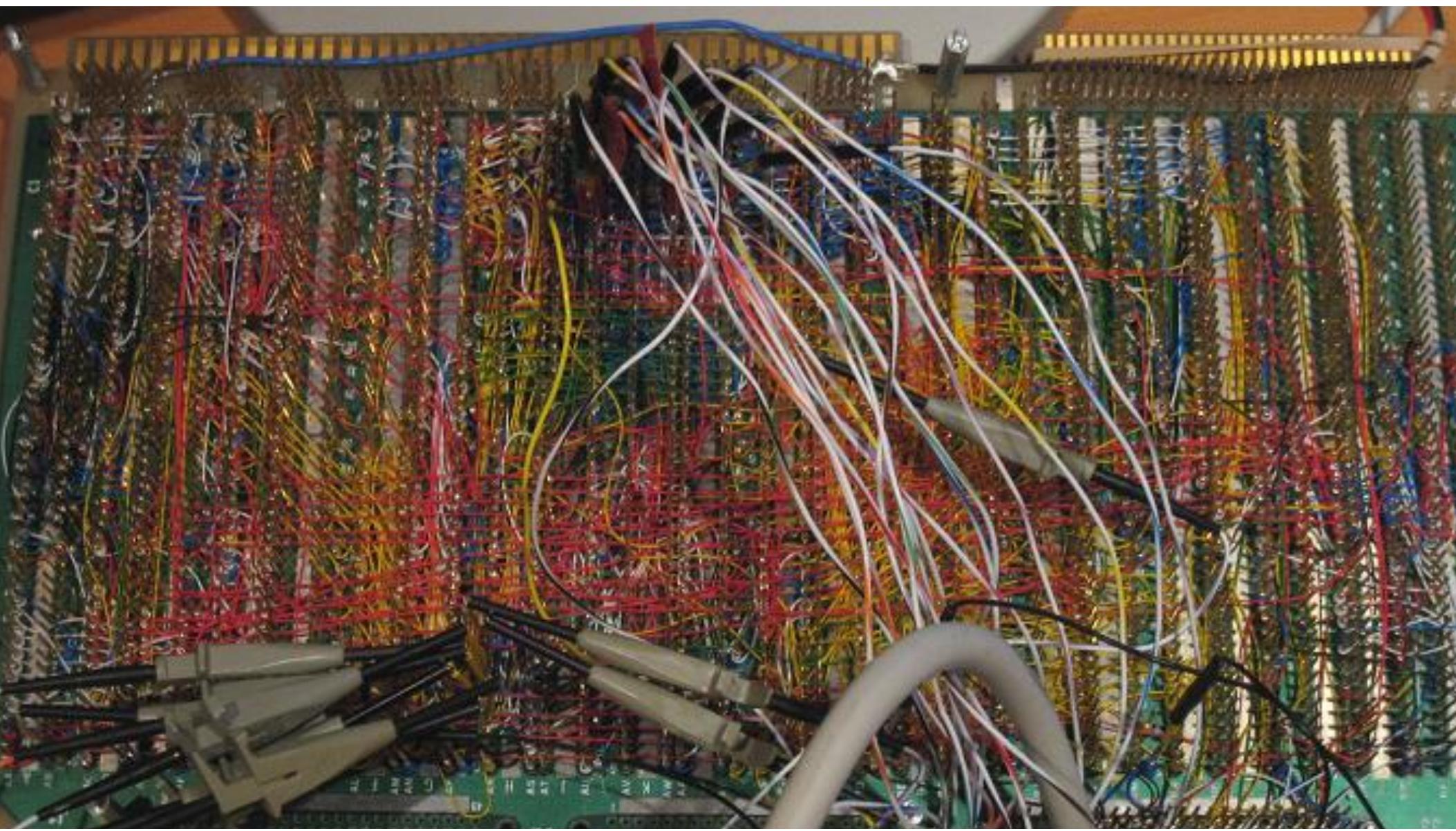
physics

designing for the real world

abstraction







```
// UdpPacket provides packetIO over UDP
public class UDPPacketIO {

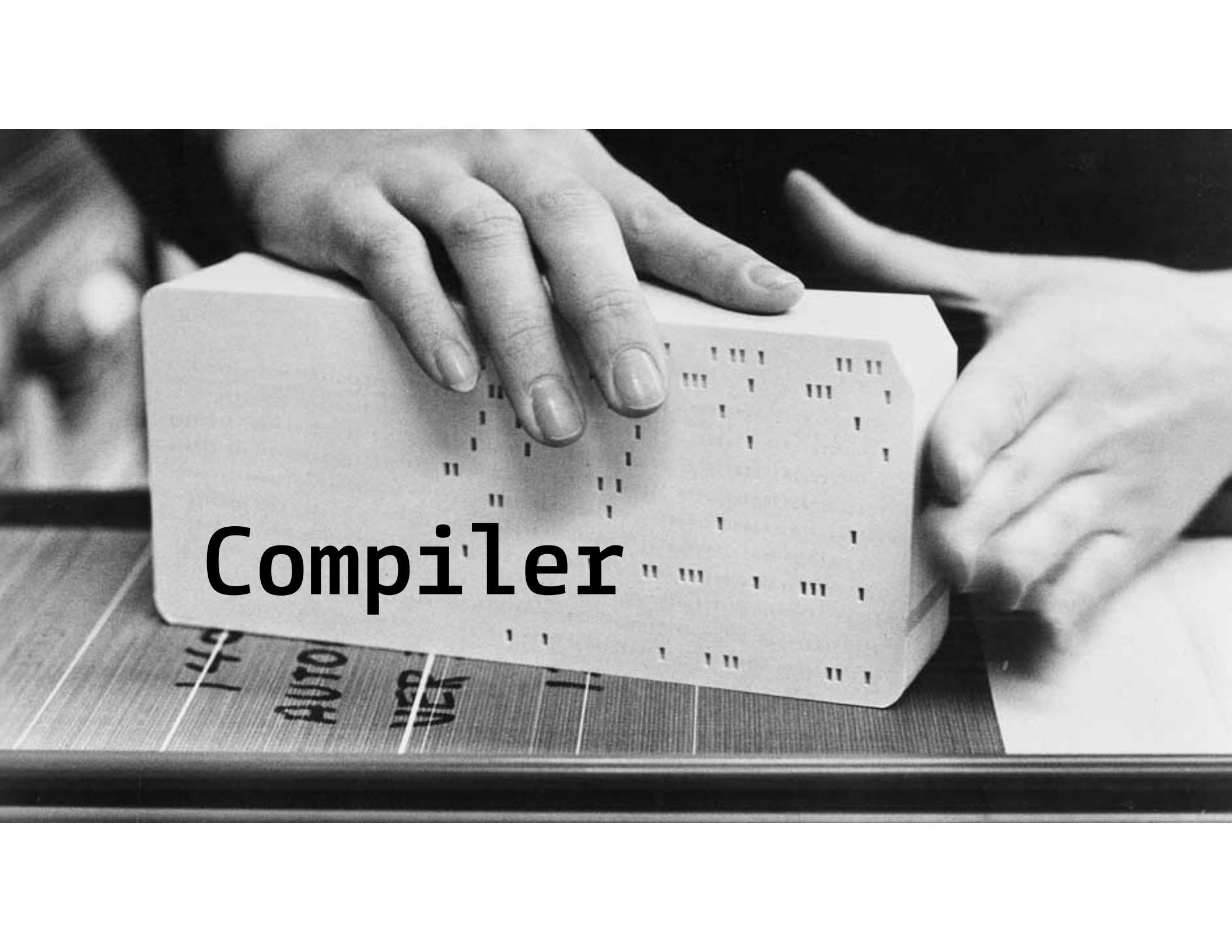
    private UdpClient Sender;
    private UdpClient Receiver;
    private bool socketsOpen;
    private string remoteHostName;
    private int remotePort;
    private int localPort;

    private string multicastAddress;
    private bool enableMulticast;

    public UDPPacketIO(string hostIP, int remotePort, int localPort, bool enableMulticast=false, string multicastAddress=null) {
        RemoteHostName = hostIP;
        RemotePort = remotePort;
        LocalPort = localPort;

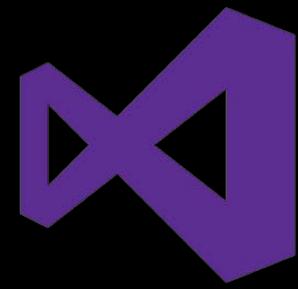
        EnableMulticast = enableMulticast;
        MulticastAddress = multicastAddress;
        socketsOpen = false;
    }

    ~UDPPacketIO() {
        // latest time for this socket to be closed
        if (IsOpen()) {
            BobRemoteControl.BobDebug.Log("closing udpclient listener on port " + localPort);
            Close();
        }
    }
}
```



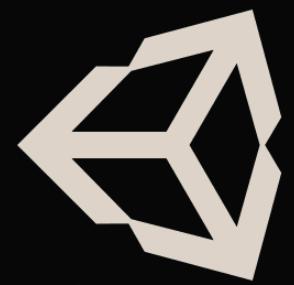
Compiler

b8 00 b8 8e c0 8d 36 20 03 e8 fd 01 bf a2 00 b9
02 00 eb 2b b4 06 b2 ff cd 21 3c 71 0f 84 e5 01
3c 50 b9 a0 00 74 18 3c 48 b9 a0 00 0f 84 d9 00
b9 02 00 3c 4d 74 08 3c 4b 0f 84 cc 00 eb d5 89
3e b5 09 01 cf 89 3e b3 09 e8 87 01 8b 3e b5 09
b0 20 26 88 05 26 88 45 fe 26 88 85 62 ff 26 88
85 60 ff 26 88 85 5e ff 26 88 85 9e 00 b0 07 26
88 45 01 8b 3e b3 09 89 fb 83 eb 02 d1 fb 8a 00
26 88 45 fe 89 fb 81 eb a2 00 d1 fb 8a 00 26 88
85 5e ff 89 fb 81 eb a0 00 d1 fb 8a 00 26 88 85
60 ff 89 fb 81 eb 9e 00 d1 fb 8a 00 26 88 85 62
ff 89 fb 81 eb a2 00 d1 fb 8a 00 26 88 85 5e ff
89 fb 83 c3 02 d1 fb 8a 00 26 88 45 02 89 fb 81
c3 9e 00 d1 fb 8a 00 26 88 85 9e 00 89 fb 81 c3
a0 00 d1 fb 8a 00 26 88 85 a0 00 89 fb 81 c3 a2
00 d1 fb 8a 00 26 88 85 a2 00 b0 03 26 88 05 a0
b7 09 26 88 45 01 e9 0b ff 89 3e b5 09 29 cf 89
3e b3 09 e8 bd 00 8b 3e b5 09 b0 20 26 88 05 26
88 45 02 26 88 85 9e 00 26 88 85 a0 00 26 88 85
a2 00 26 88 85 62 ff b0 07 26 88 45 01 8b 3e b3
09 89 fb 83 eb 02 d1 fb 8a 00 26 88 45 fe 89 fb
81 eb a2 00 d1 fb 8a 00 26 88 85 5e ff 89 fb 81
eb a0 00 d1 fb 8a 00 26 88 85 60 ff 89 fb 81 eb
9e 00 d1 fb 8a 00 26 88 85 62 ff 89 fb 81 eb a2
00 d1 fb 8a 00 26 88 85 5e ff 89 fb 83 c3 02 d1



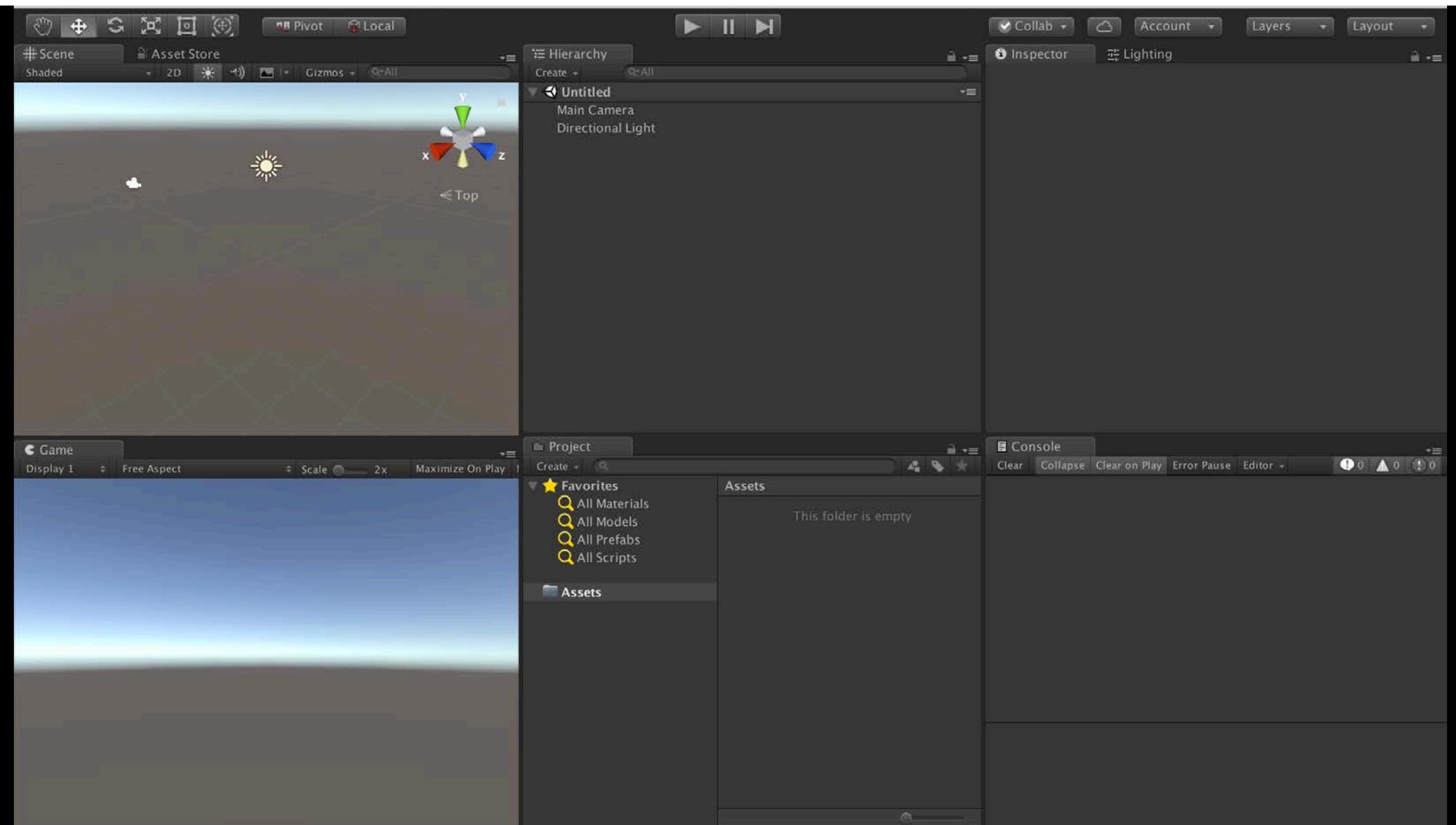
Visual
Studio

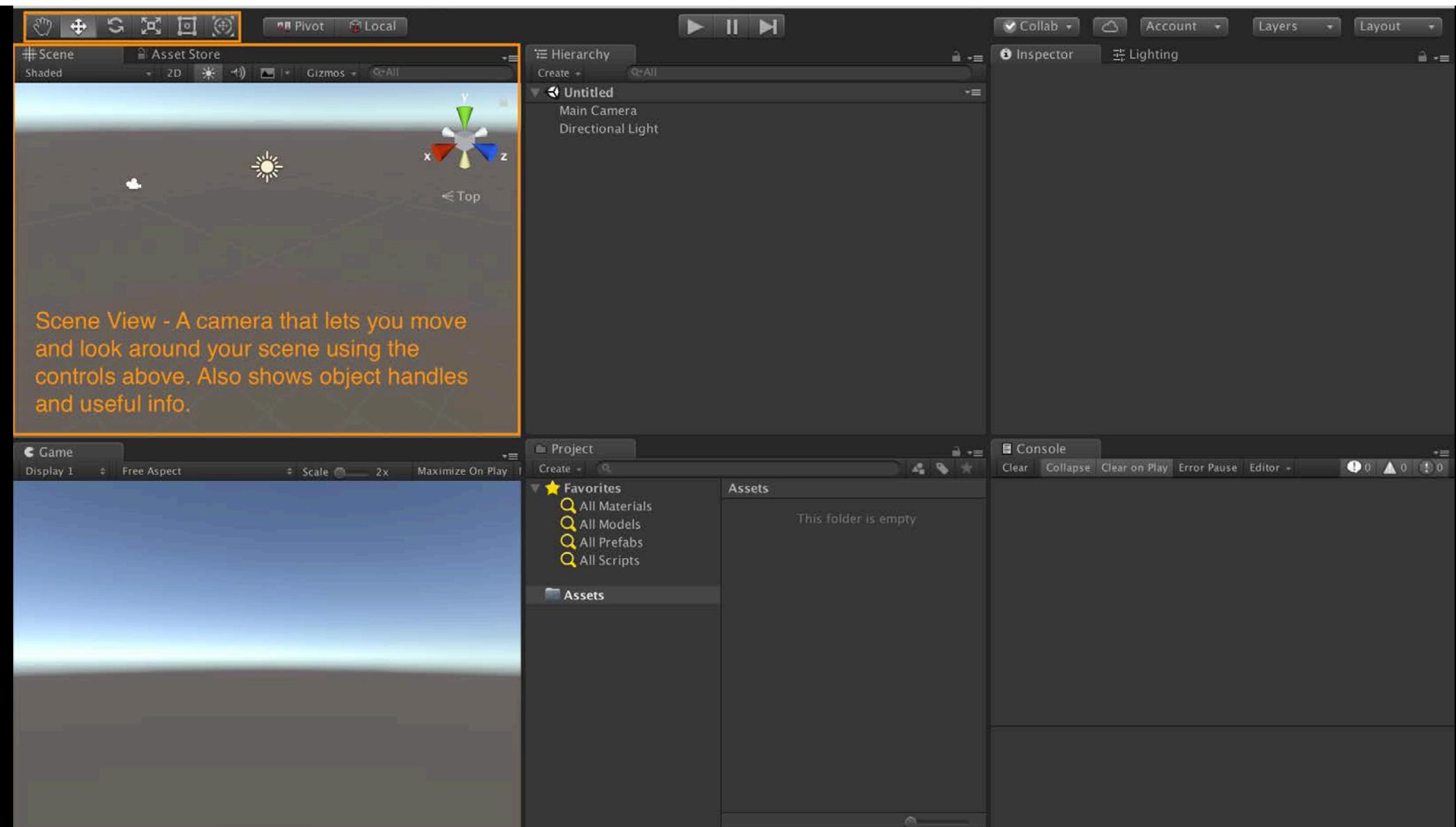


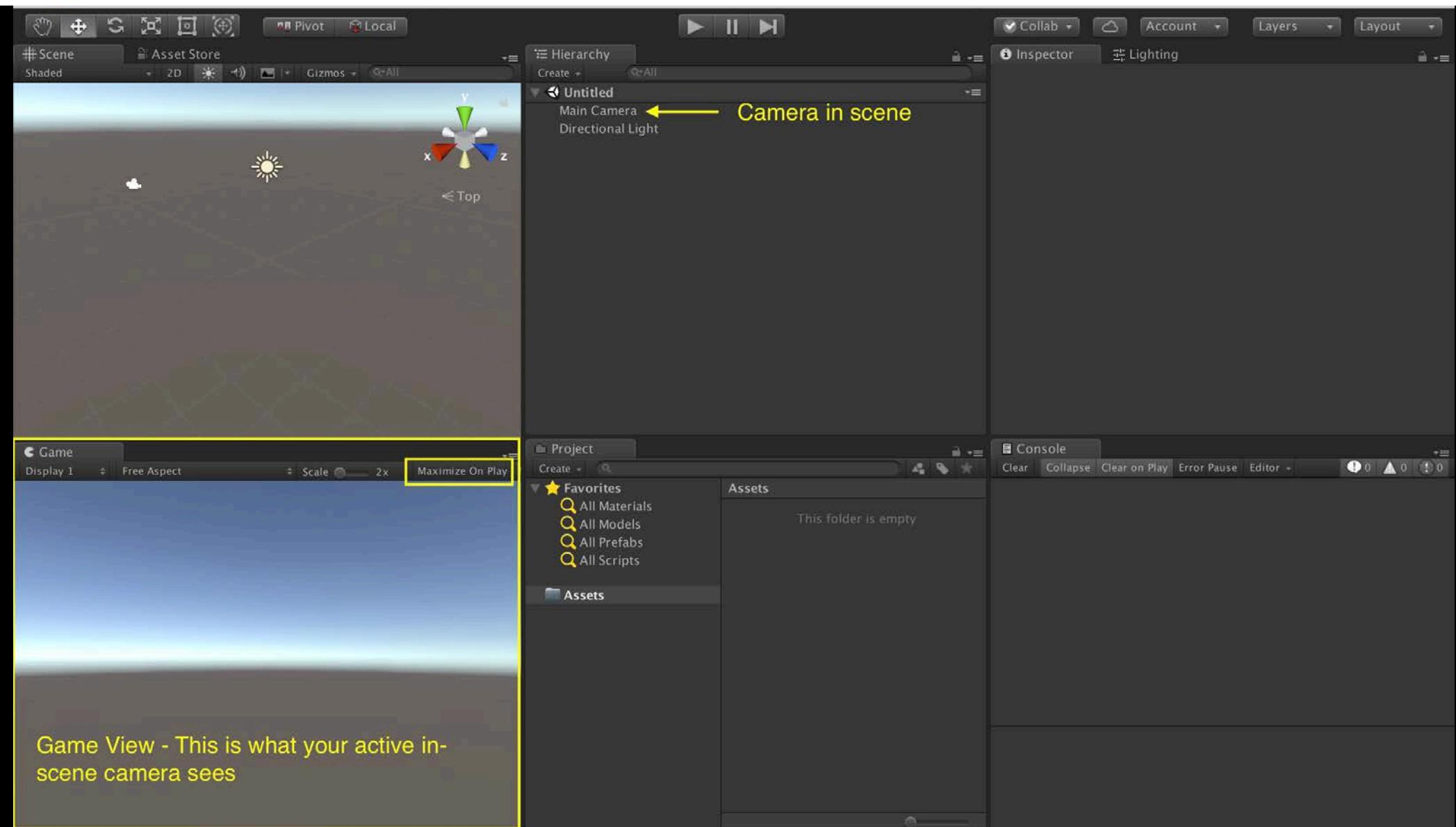


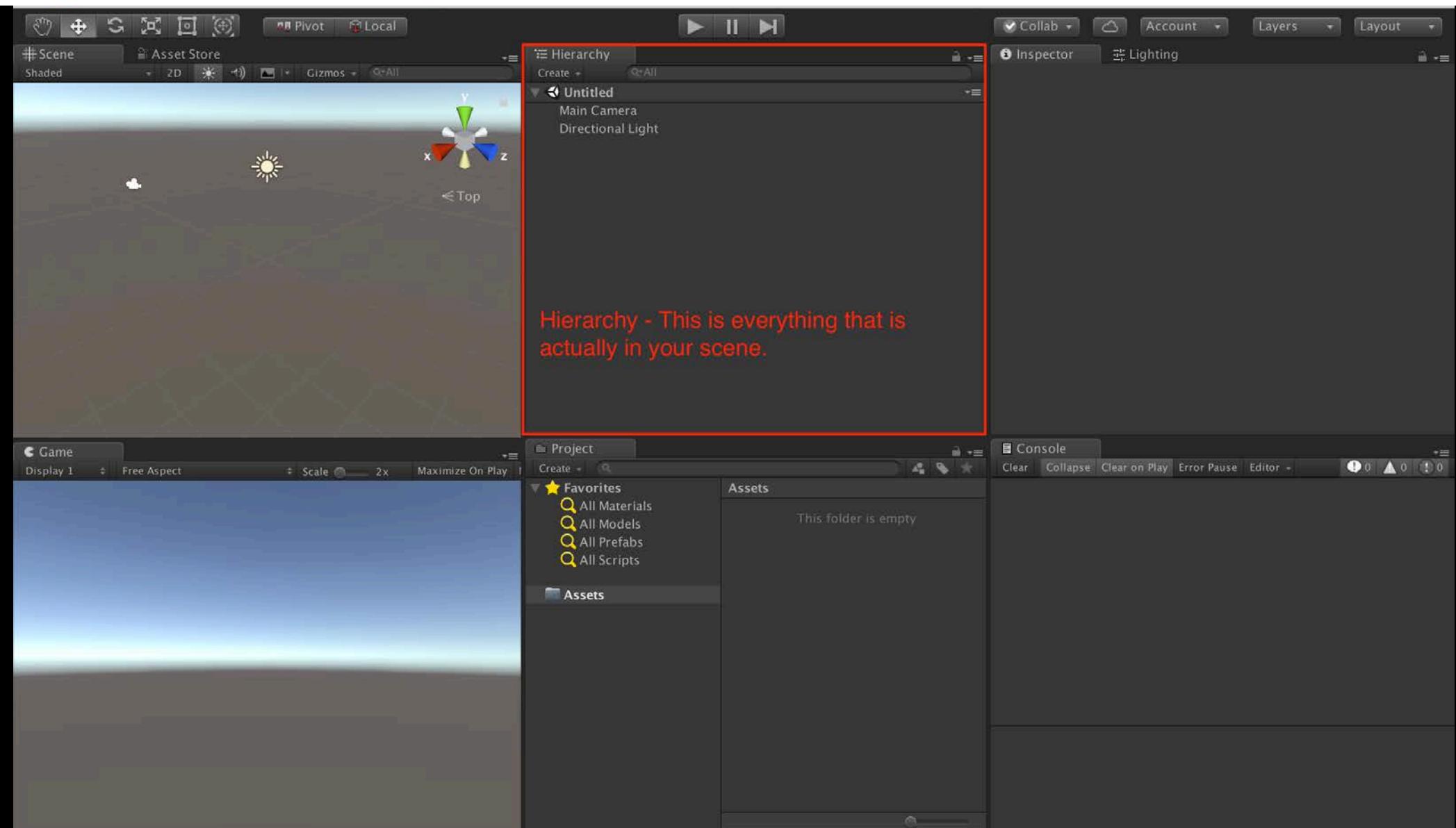
unity

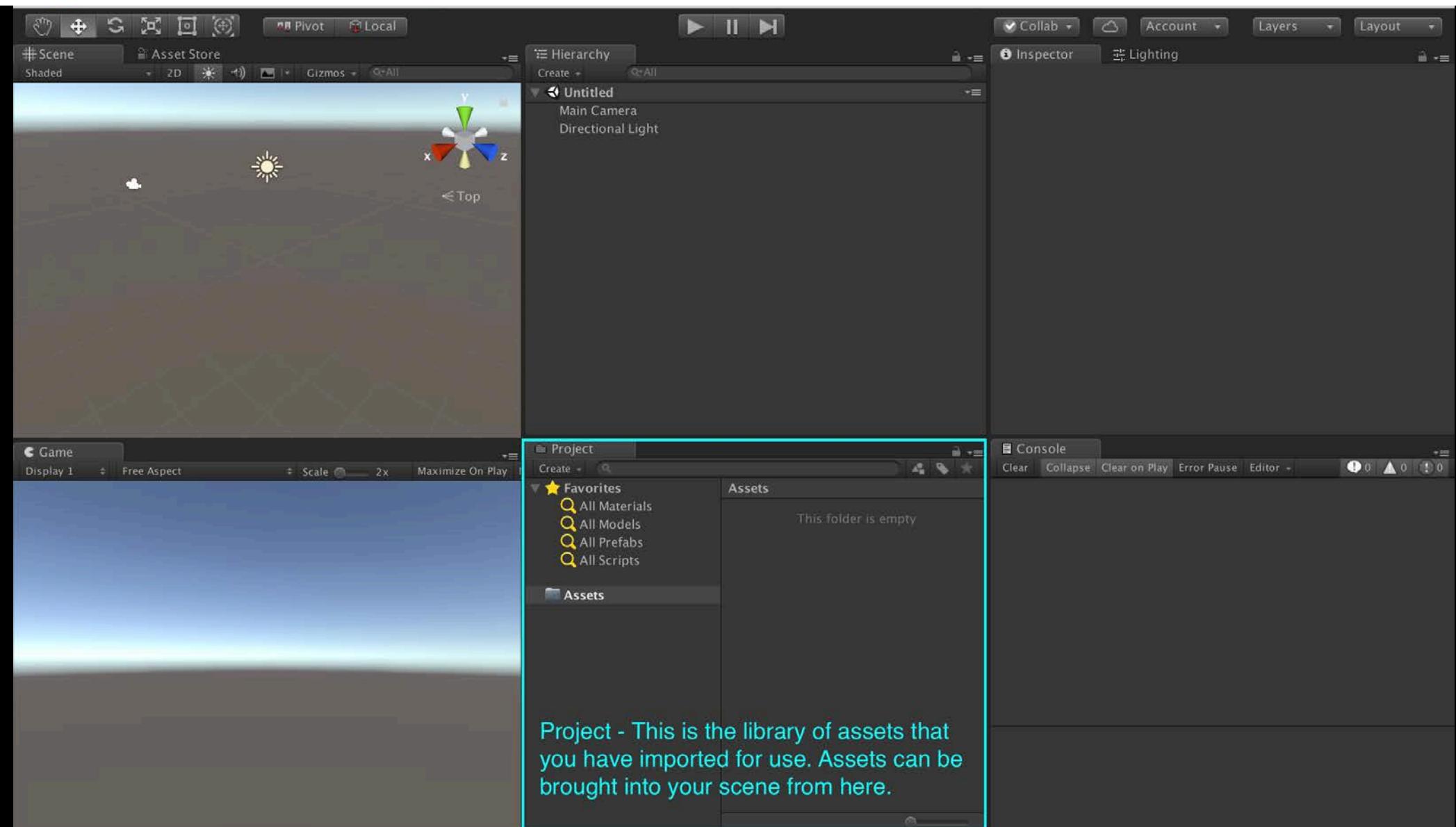


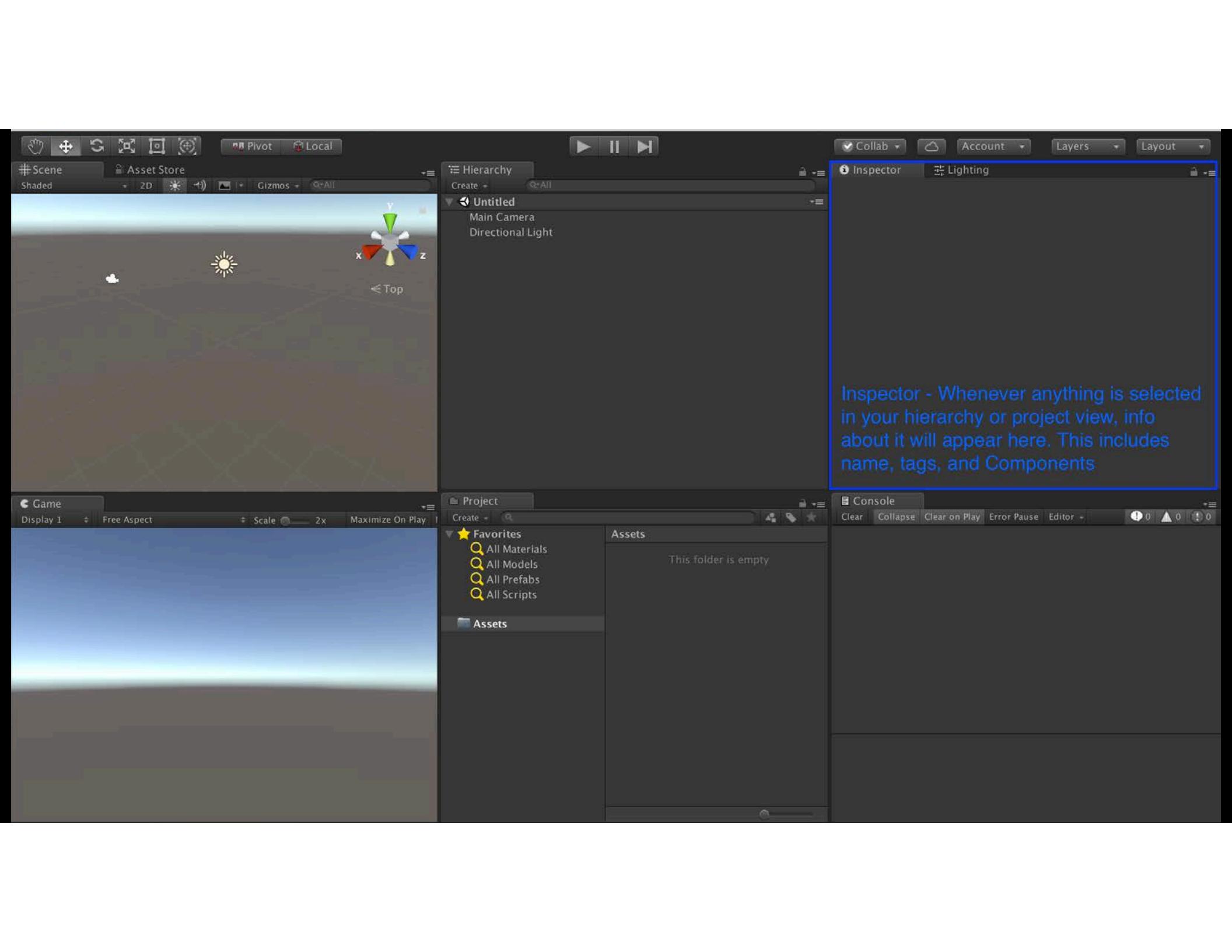




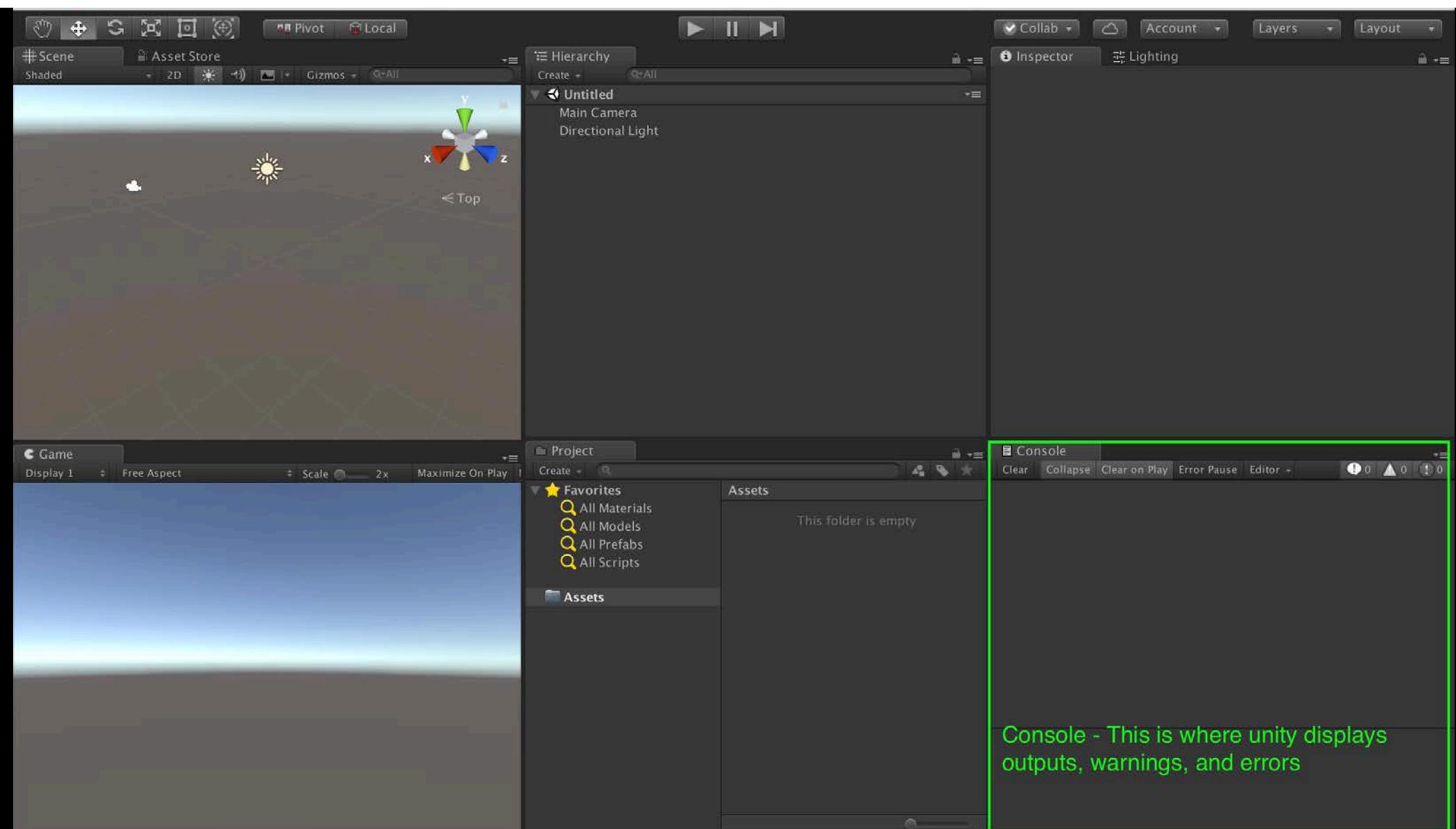








Inspector - Whenever anything is selected in your hierarchy or project view, info about it will appear here. This includes name, tags, and Components



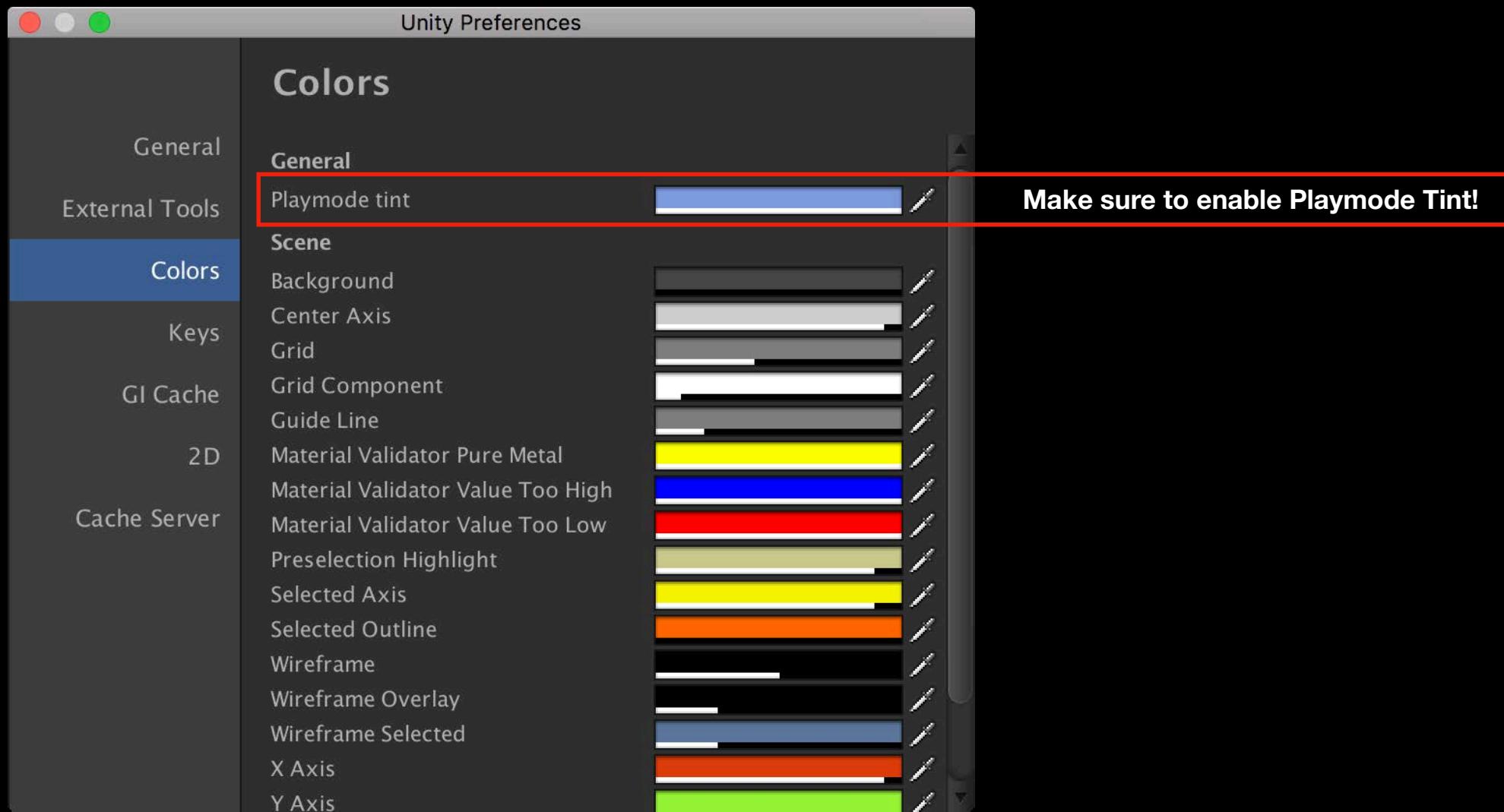
Console - This is where unity displays outputs, warnings, and errors

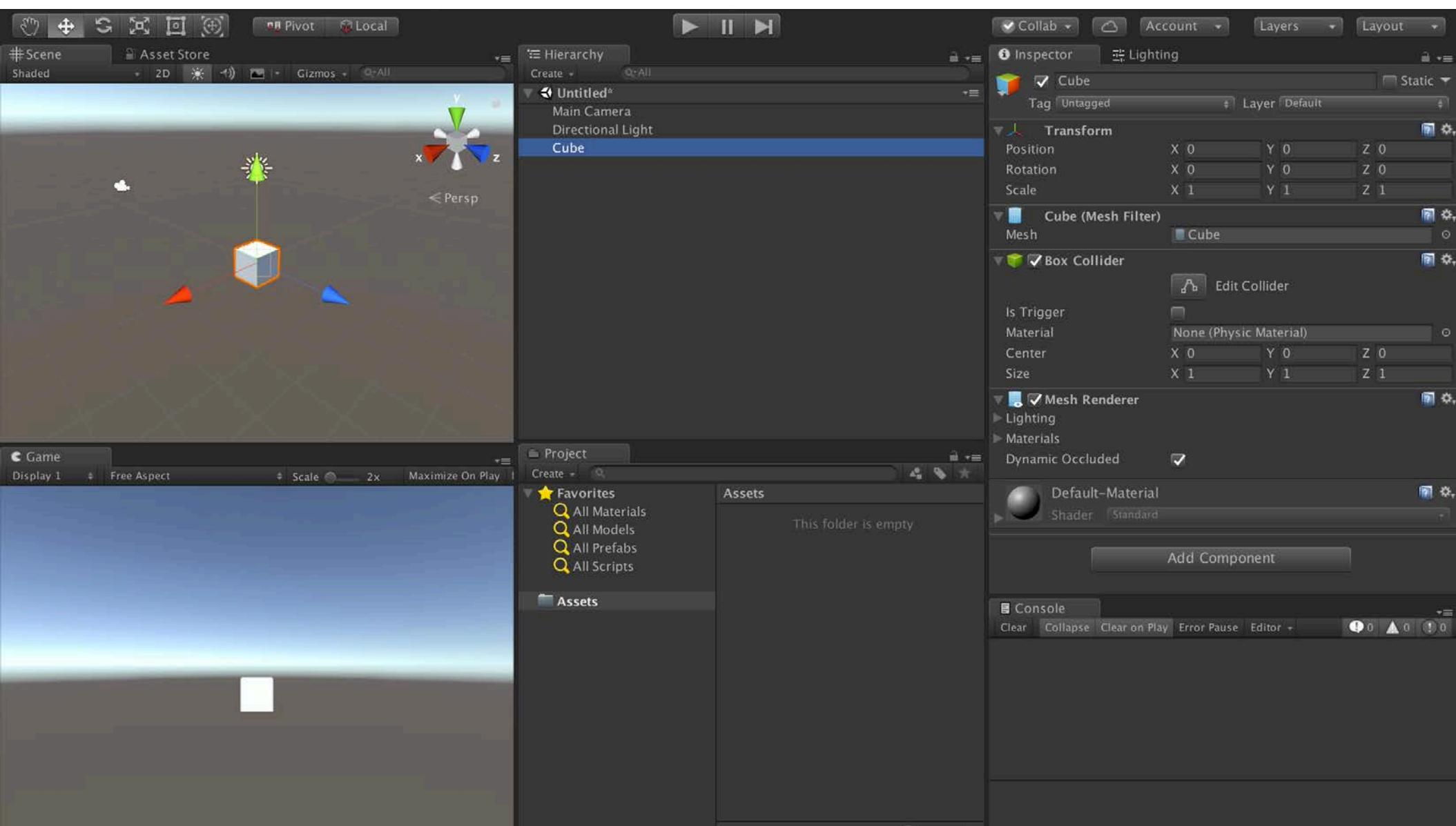
The image shows the Unity Editor interface with several windows open:

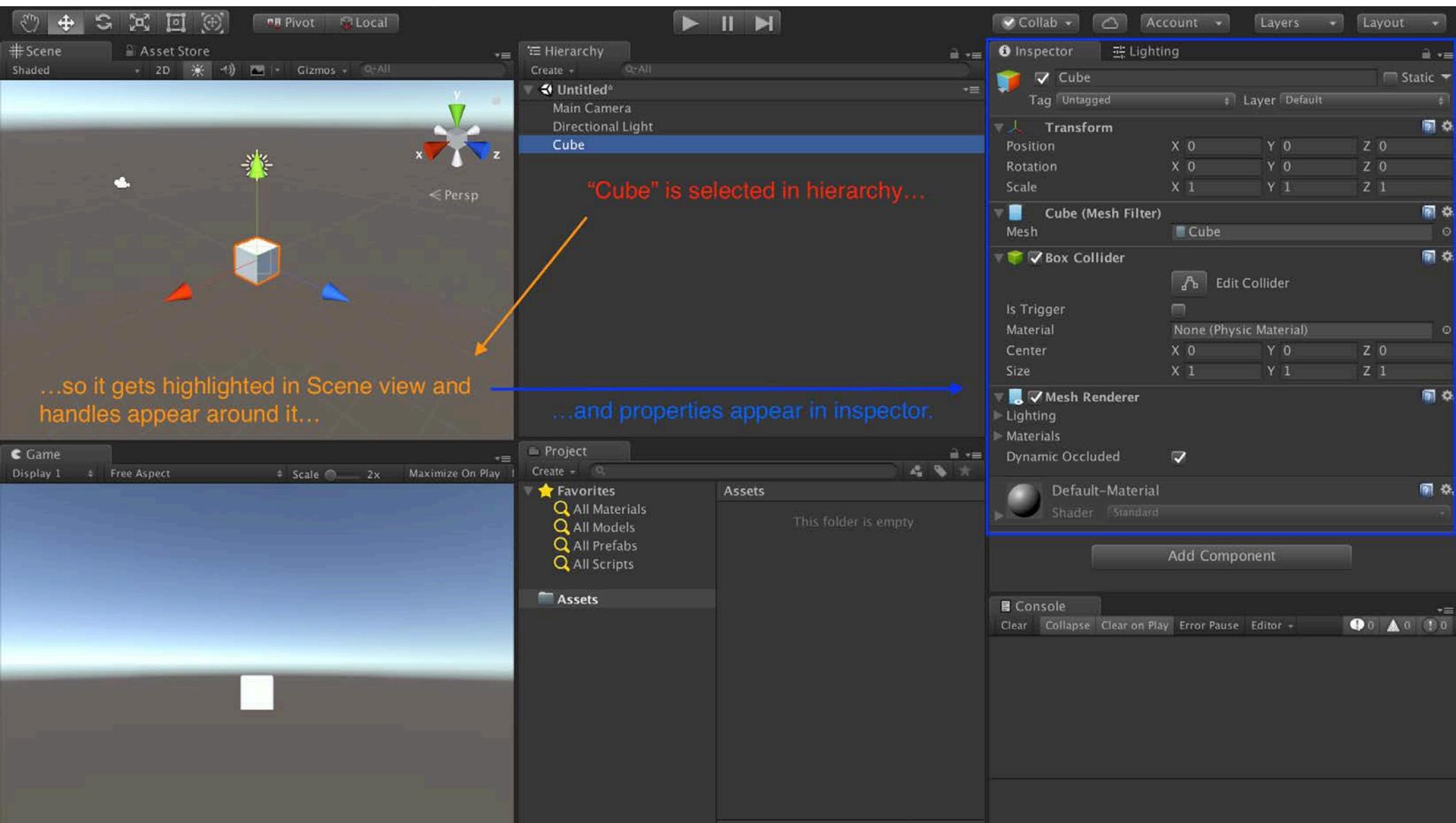
- Scene View** (Top Left): A camera view showing a landscape with a sun and a directional light. It includes controls for moving and looking around the scene.
- Game View** (Bottom Left): A camera view showing a blue sky and ground, representing what the active camera sees.
- Hierarchy** (Top Middle): A tree view showing the scene structure. The root node is "Untitled", which contains "Main Camera" and "Directional Light".
- Project** (Bottom Middle): A library of assets. The "Favorites" section lists "All Materials", "All Models", "All Prefabs", and "All Scripts". The "Assets" section shows a folder named "Assets" with the message "This folder is empty".
- Inspector** (Top Right): Shows information about selected objects. In this case, it's empty because nothing is selected.
- Console** (Bottom Right): Displays outputs, warnings, and errors. It is currently empty.

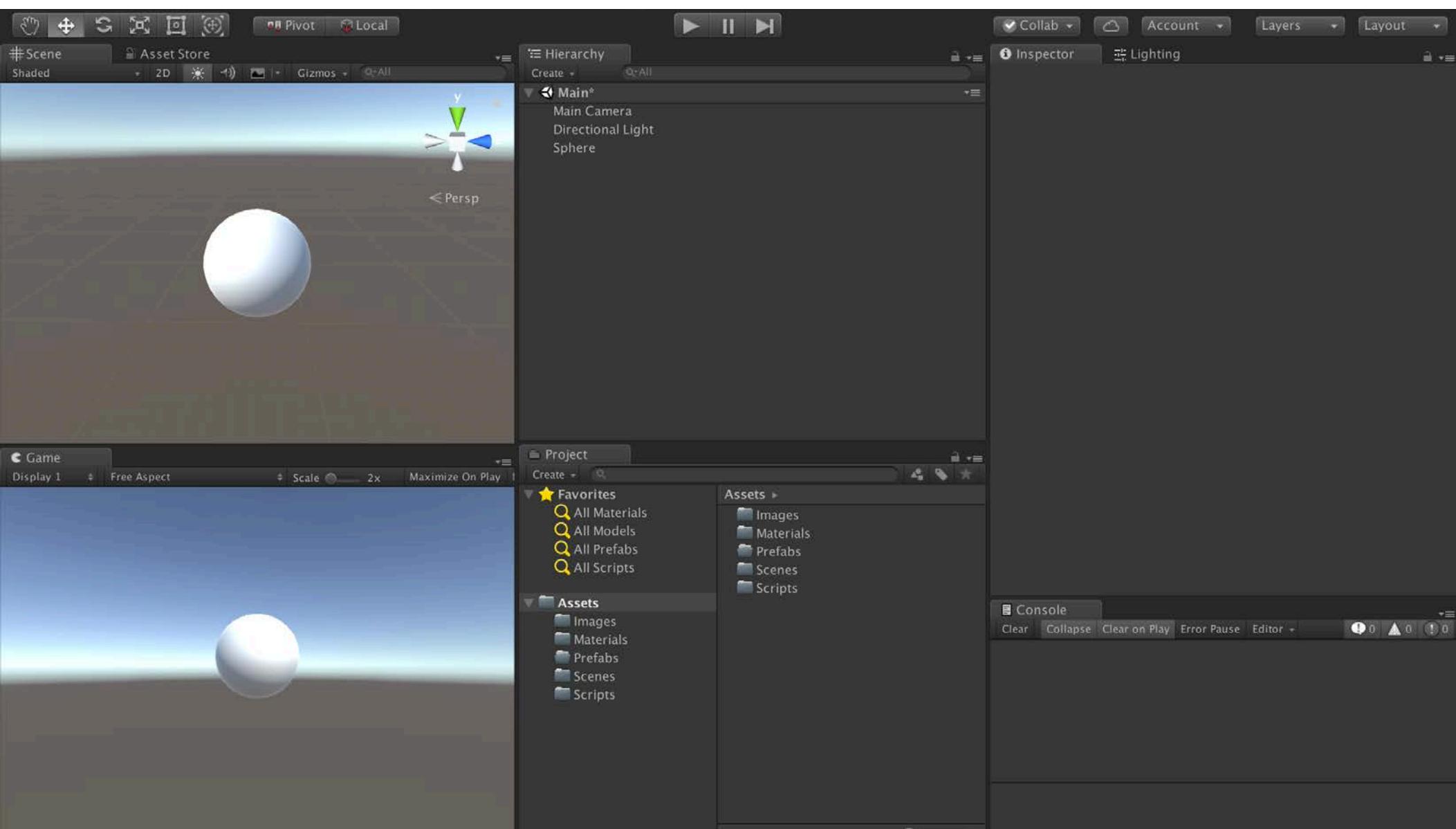
Annotations provide descriptions for each window:

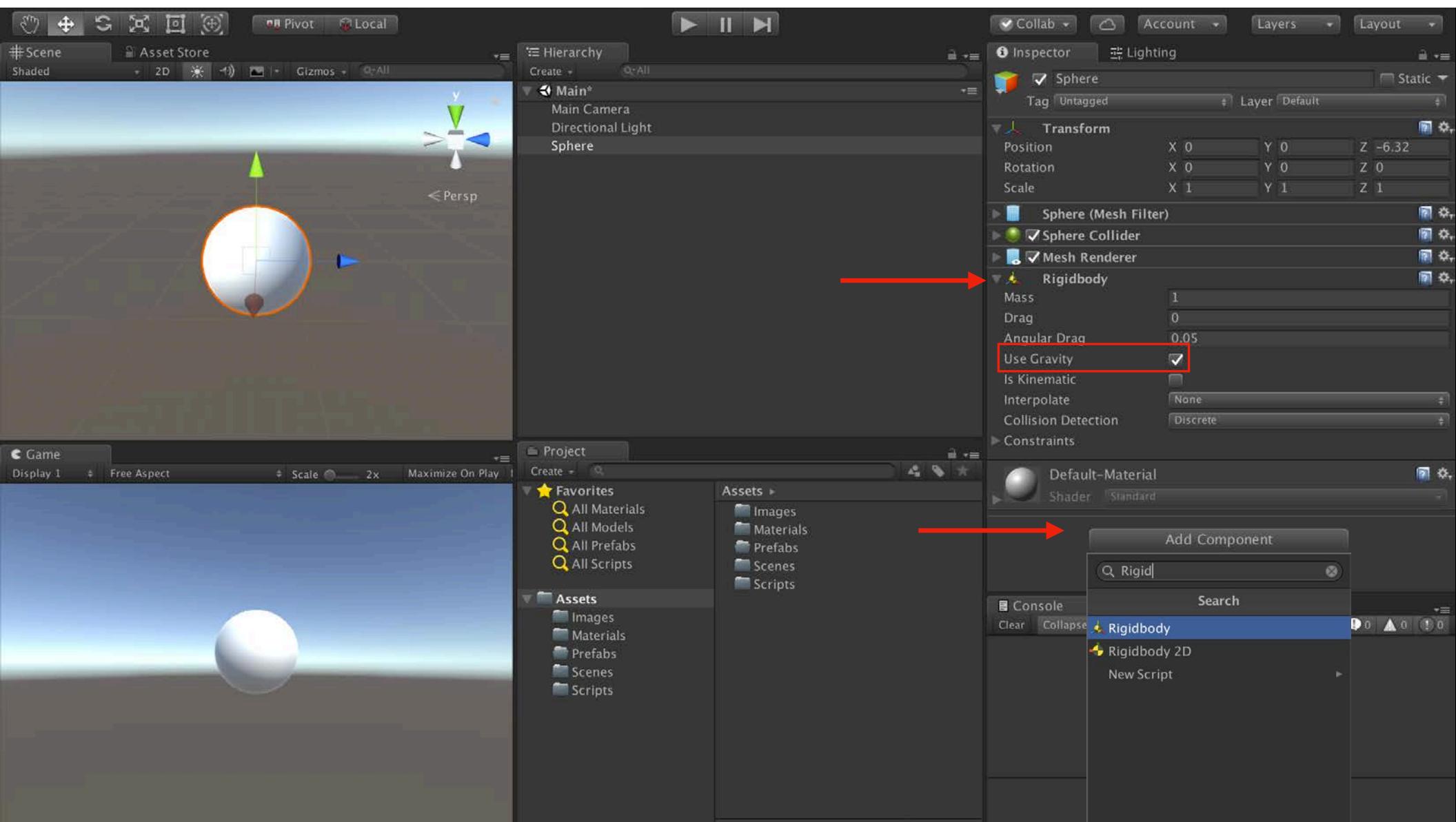
- Scene View**: A camera that lets you move and look around your scene using the controls above. Also shows object handles and useful info.
- Hierarchy**: This is everything that is actually in your scene.
- Inspector**: Whenever anything is selected in your hierarchy or project view, info about it will appear here. This includes name, tags, and Components.
- Project**: This is the library of assets that you have imported for use. Assets can be brought into your scene from here.
- Console**: This is where unity displays outputs, warnings, and errors.

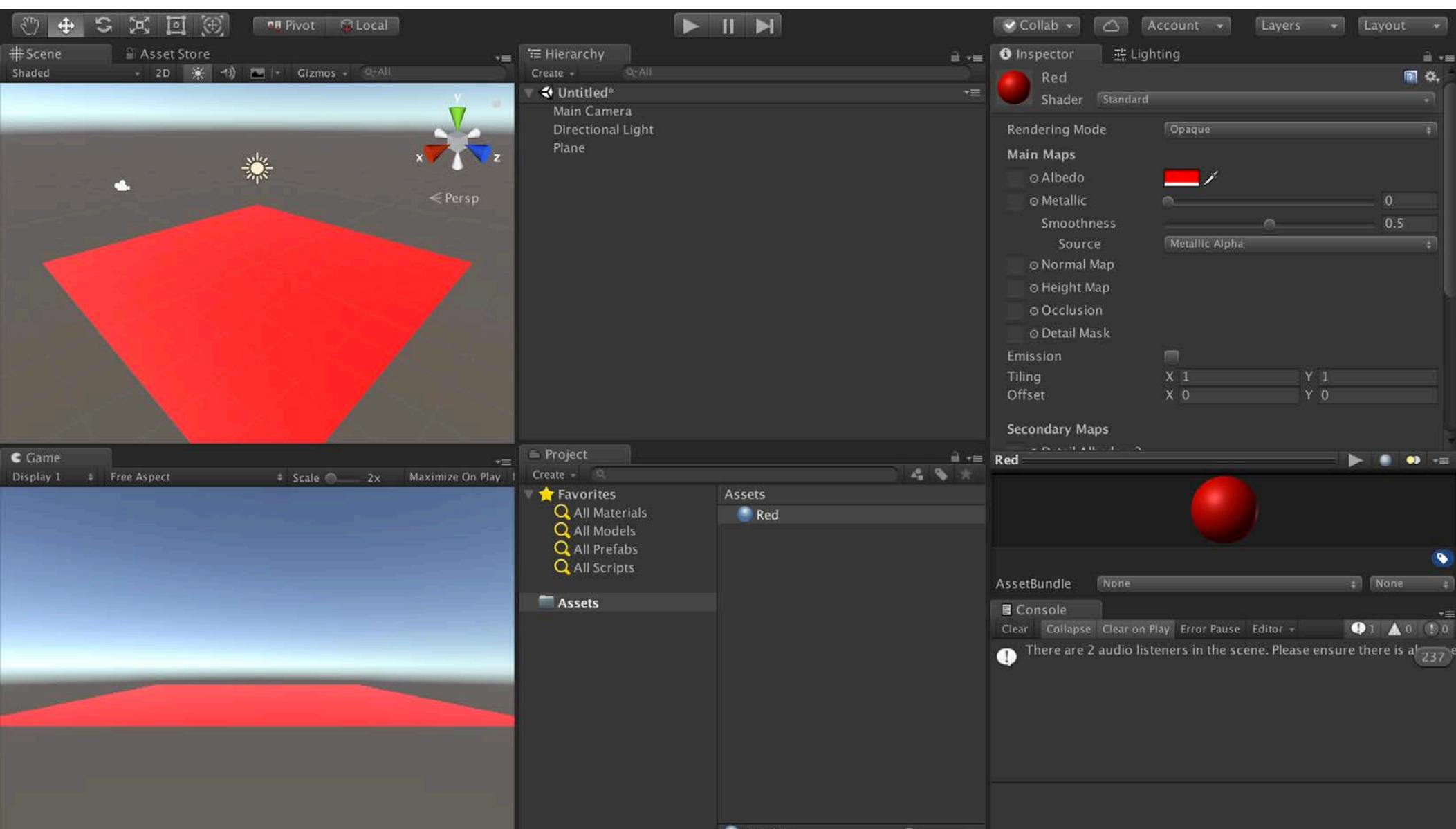


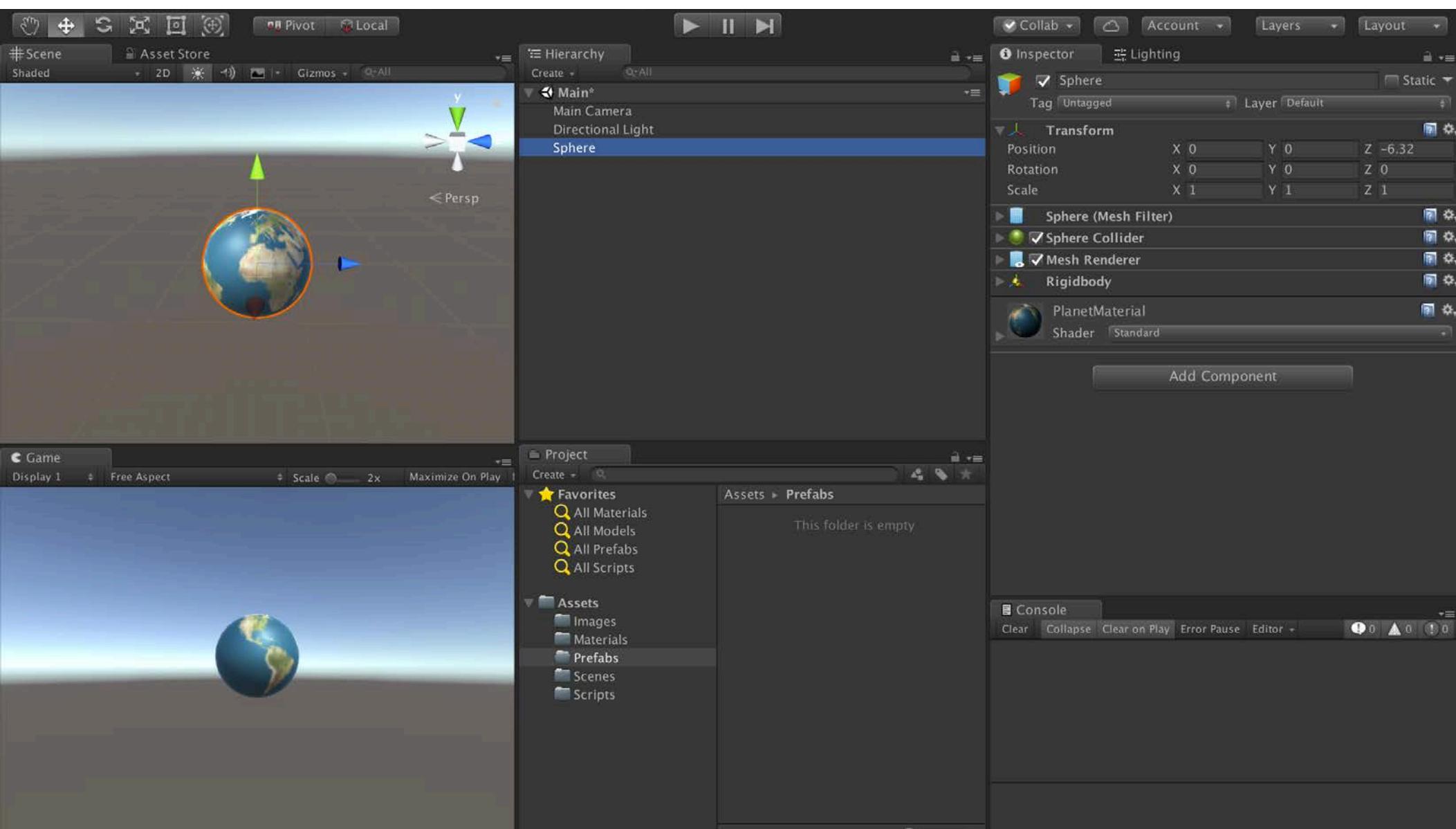


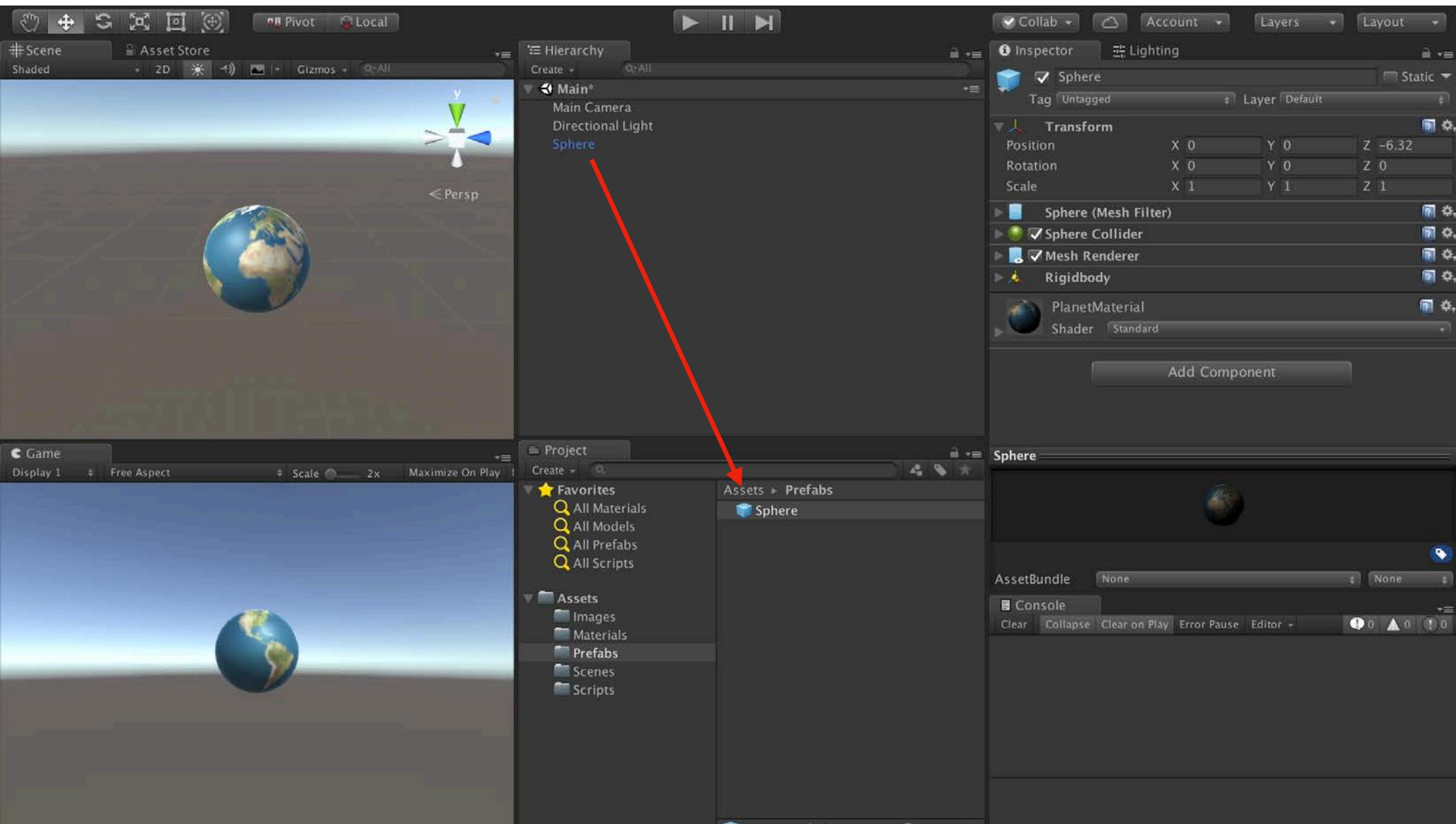


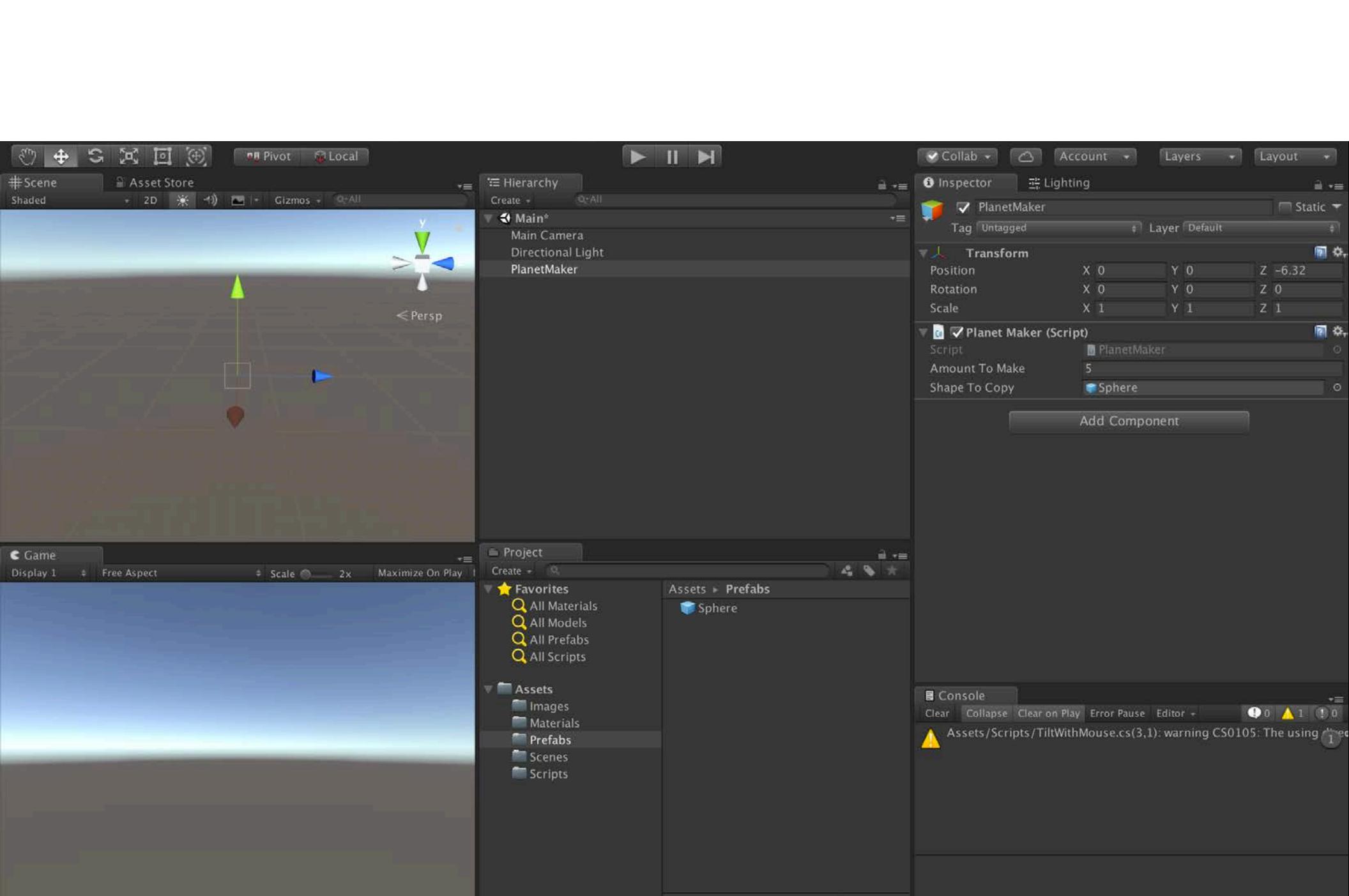












The screenshot shows the Unity Asset Store interface. The top navigation bar includes icons for selection, transform, pivot, local, collab, account, layers, and layout. Below the bar are standard UI elements like back, forward, home, search, and a notification icon with 5 messages. The main content area features a large banner with the text "Create your first game" and a "Start creating" button. A descriptive text block below the banner explains game development courses. To the right is a sidebar with filters and a list of categories: Home, 3D Models, Animation, Audio, Complete Projects, Editor Extensions, Particle Systems, Scripting, Services, Shaders, Textures & Materials, and Unity Essentials. A "NEW" section highlights top paid and free packages. The bottom section shows cards for the Global Game Jam 2018, prototyping a 3rd person game, top paid packages, and top free packages.

Asset Store

Pivot Local

Collab Account Layers Layout

Language: English luxloop

Search...

Create your first game

Start creating

Learn all the essentials of game development with Unity Game Dev Courses. We've created these courses with one of the world's leading technology learning companies to give you everything you need to build an amazing game in Unity. When complete, you will have built the [Swords and Shovels](#) game!

Global Game Jam 2018

Check it out

Quickly prototype a 3rd Person Game

Top Paid Packages

Top Free Packages

NEW

Filters

- Home
- 3D Models
- Animation
- Audio
- Complete Projects
- Editor Extensions
- Particle Systems
- Scripting
- Services
- Shaders
- Textures & Materials
- Unity Essentials

Top Paid

Dynamic Bone Scripting/Animation

Muscle Animation Editor Editor Extensions/Anim...

Animal pack deluxe

Asset Store

Pivot Local

Collab Account Layers Layout

Language: English luxloop

Standard Assets

MAXIMUM PRICE \$

FREE 5 10 20 50 100 200 ∞

FREE ONLY **PAID ONLY**

MINIMUM RATING ★★★★★

SUPPORTED UNITY VERSION < 2017.3.0 e.g. 5.2.0

PACKAGES ONLY **LISTS ONLY**

MAXIMUM SIZE MB

1MB 5MB 50MB 100MB 250MB 500MB 1GB 4GB

RELEASED days ago

1d 7d 14d 1m 3m 6m 1y 5y

UPDATED days ago

1d 7d 14d 1m 3m 6m 1y 5y

SORT BY RELEVANCE / POPULARITY / NAME / PRICE / RATING / UPDATED

1 2 3 4 5 6 7 8 9 10 Next Last 1 - 36 of 1630

 Standard Assets Unity Essentials/A... Unity Technologies ★★★★★ (13137) FREE	 Standard Assets f... Unity Essentials/A... Unity Technologies ★★★★★ (1246) FREE	 Assets_Kitchenroom 3D Models/Enviro... Argyle Co.,Ltd Not enough ratings \$18.00
 Assets_classroom 3D Models/Enviro... Argyle Co.,Ltd Not enough ratings \$24.00	 Standard Scooter 3D Models/Vehicel... Virtualware ★★★ (13) \$2.00	 Assets_School_Ha... 3D Models/Enviro... Argyle Co.,Ltd Not enough ratings \$10.00
 Affordable Assets... 3D Models/Props/... Creation Wasteland Not enough ratings \$5.99	 Grab Yer Assets Editor Extensions/... Xeir ★★★★★ (71) \$49.00	 Graveyard Assets ... 3D Models/Enviro... Kittens and Elves ... Not enough ratings \$5.00
WORLD CREATOR World Creator Sta...	Color Mask Stand...	10+1 Standard M...

Filters

- Home
- 3D Models
- Animation
- Audio
- Complete Projects
- Editor Extensions
- Particle Systems
- Scripting
- Services
- Shaders
- Textures & Materials
- Unity Essentials

NEW

Top Paid

- Dynamic Bone Scripting/Animation
- Muscle Animation Editor Editor Extensions/Anim...
- Animal pack deluxe

Asset Store

Pivot Local

Collab Account Layers Layout

Language: English luxloop

Standard Assets

Unity Essentials/Asset Packs

Unity Technologies

★★★★★ (13137)

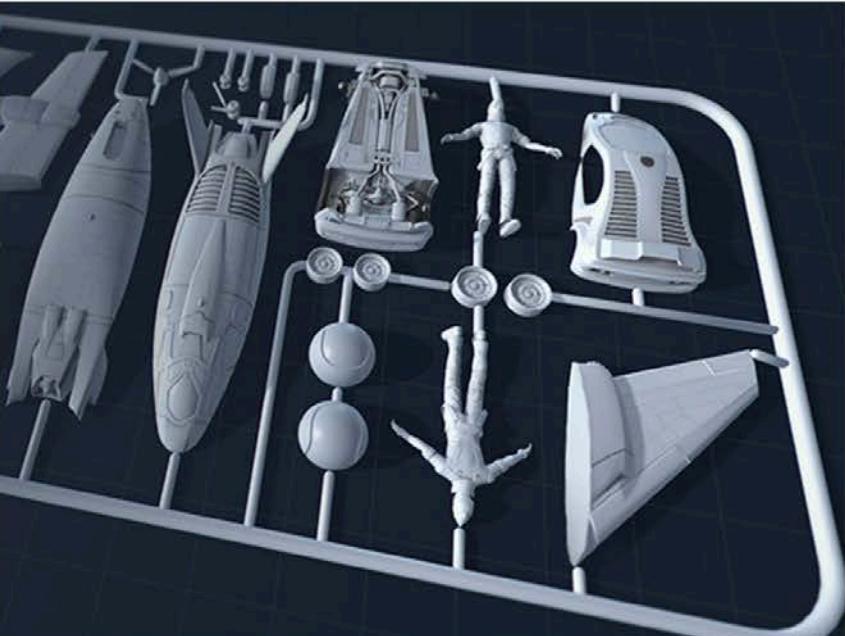
FREE

Update

This collection of assets, scripts, and example scenes can be used to kickstart your Unity learning or be used as the basis for your own projects.

The package includes:

First Person Character Controller
Third Person Character Controller
Car Controller



Version: 1.1.4 (Jan 03, 2018) Size: 191.8 MB

Originally released: 16 March 2015

Package has been submitted using Unity 5.0.0, 5.1.4, 5.2.4, 5.3.1, 5.4.2, 5.5.0, 5.6.0, 2017.1.0, and 2017.3.0 to improve compatibility within the range of these versions of Unity.

Support E-mail Support Website Visit Publisher's Website

Filters

Home

3D Models

Animation

Audio

Complete Projects

Editor Extensions

Particle Systems

Scripting

Services

Shaders

Textures & Materials

Unity Essentials

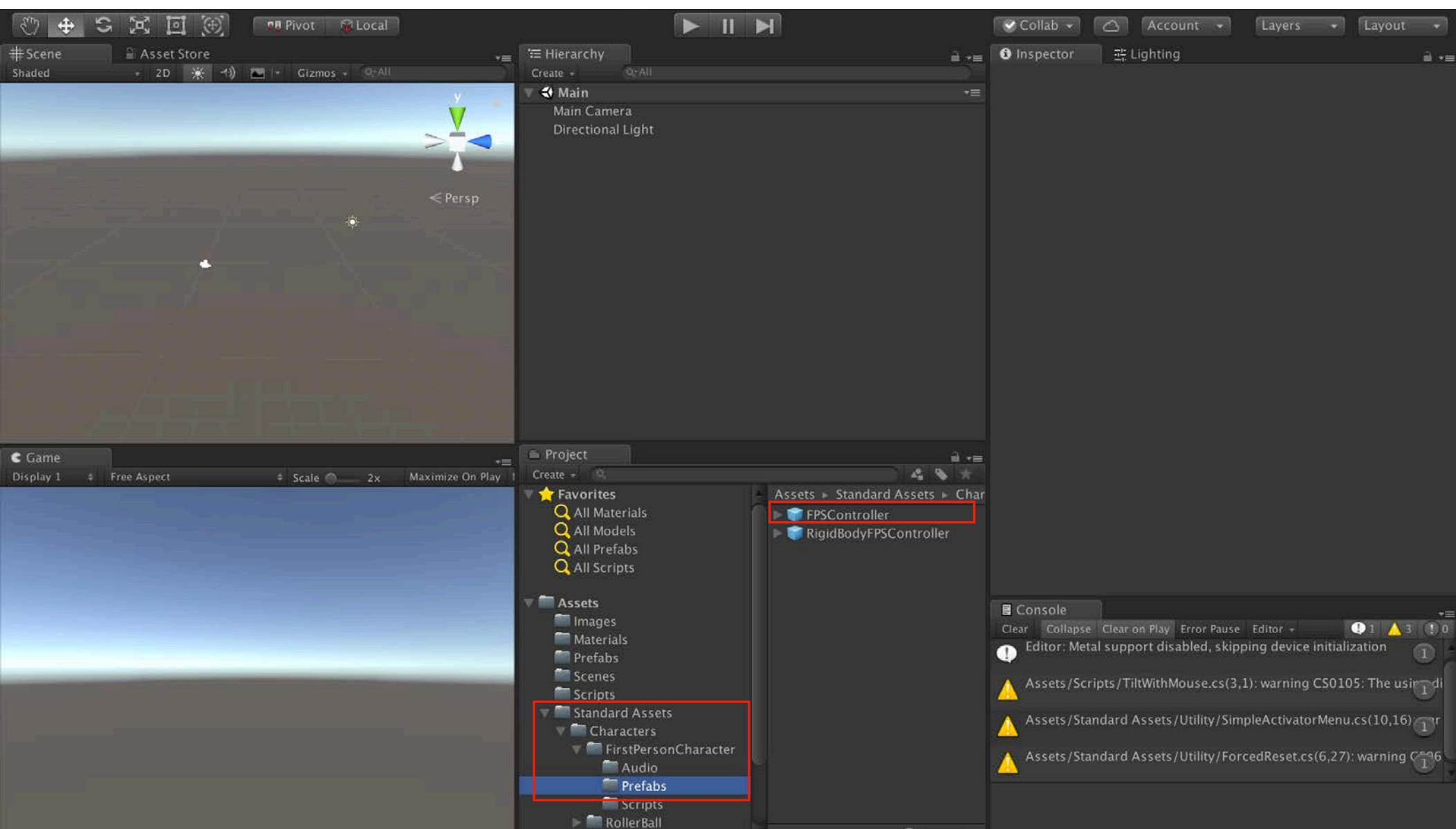
- Asset Packs
- Beta Content
- Certification
- Sample Projects
- Other

NEW

Top Paid

There are no packages under this category

Top Free





Vectors

In programming terms, you can think of Vectors as a way to store 2, 3, or 4 values in one easy-to-use package:

```
Vector2 someNumbers = new Vector2(1.0, 2.2);  
Vector3 someOtherNumbers = new Vector3(5.3, 2.6, 12.0);  
Vector4 evenMoreNumbers = new Vector4(7.4, 2.1, 12.0, 9.8);
```

Vectors

We can use vectors to:

- Store multiple numbers in one variable
- Describe the position of something in our world
 - For example: (2.1, 8.9, 7.4) represents the point in space 2.1 units along the X-axis, 8.9 units along the Y-axis, and 7.4 units along the Z-axis.

Vectors

We can use vectors to:

- Describe a direction

- For example: $(0.0, 1.0, 0.0)$ represents a point 1 unit directly above (along Y) the origin.
- If we drew an arrow from the origin to this point, it would point straight up.
- It doesn't matter how long the Vector is:
 - $(0.0, 1.0, 0.0)$ and $(0.0, 5.2, 0.0)$ are different points, but they both describe the same *direction* (straight up).

Vectors

Unity has some built-in direction shorthands:

```
Vector3 example = Vector3.up;
```

is the same as:

```
Vector3 example = new Vector3(0.0, 1.0, 0.0);
```

Vectors

Other shorthands:

`Vector3.up` (pointing along Y-axis)

`Vector3.forward` (pointing along Z-axis)

`Vector3.right` (pointing along X-axis)

`Vector3.one` (Equal to $(1.0, 1.0, 1.0)$)

RayCasting

RayCasting is when we shoot an invisible line into our scene to see if we hit something in that direction.

To understand RayCasting, you must understand **Vectors**.

RayCasting

`Physics.Raycast()` is a function built in to Unity.
There are many, many different forms it can take. Here is
the easiest:

```
Physics.Raycast(Vector3 originOfTheRay, Vector3 directionOfTheRay);
```

All this function actually does is return `true` or `false` to
answer “did this Ray hit anything?”

RayCasting

To store information about *what* was hit, and more importantly *where* the hit is in space, we have to do two things:

1. Declare a variable of the type `RaycastHit` to store the information about the hit point.
2. Use a slightly different version of `Physics.Raycast()` to pass the hit info out of it:

```
RaycastHit hitInfoVariable  
Physics.Raycast(Vector3 originOfTheRay, Vector3 directionOfTheRay, out hitInfoVariable)
```

RayCasting

So if wanted to Raycast from a GameObject (for example a Vive tracker or the user's headset POV):

We want to shoot a ray from:

gameObject.transform.position

in the direction of:

gameObject.transform.forward

(**gameObject.transform.forward** is the local Z-axis of the *object*, which may be different from the *world Z-axis*, which is Vector3.forward)

RayCasting

```
void Update() {  
    RaycastHit hit;  
    if ( Physics.Raycast(gameObject.transform.position, gameObject.transform.forward, out hit) ) {  
  
        Debug.DrawLine(gameObject.transform.position, hit.point, Color.red);  
        Debug.DrawRay(hit.point, hit.normal, Color.green);  
  
    }  
}
```

RayCasting

the **hit** variable that stores information about the result of the Raycast has a few useful properties:

`hit.point` (The coordinates of the collision as a `Vector3`)

`hit.normal` (A `Vector3` direction that describes the direction coming *straight out* of the face of the `hit` object)

RayCasting

These visual Debug functions help you see what's going on. They will draw lines in your *Editor*, but never in the actual *Game* view:

```
Debug.DrawLine(Vector3 lineStartCoordinate, Vector3 lineEndCoordinate, Color color);
```

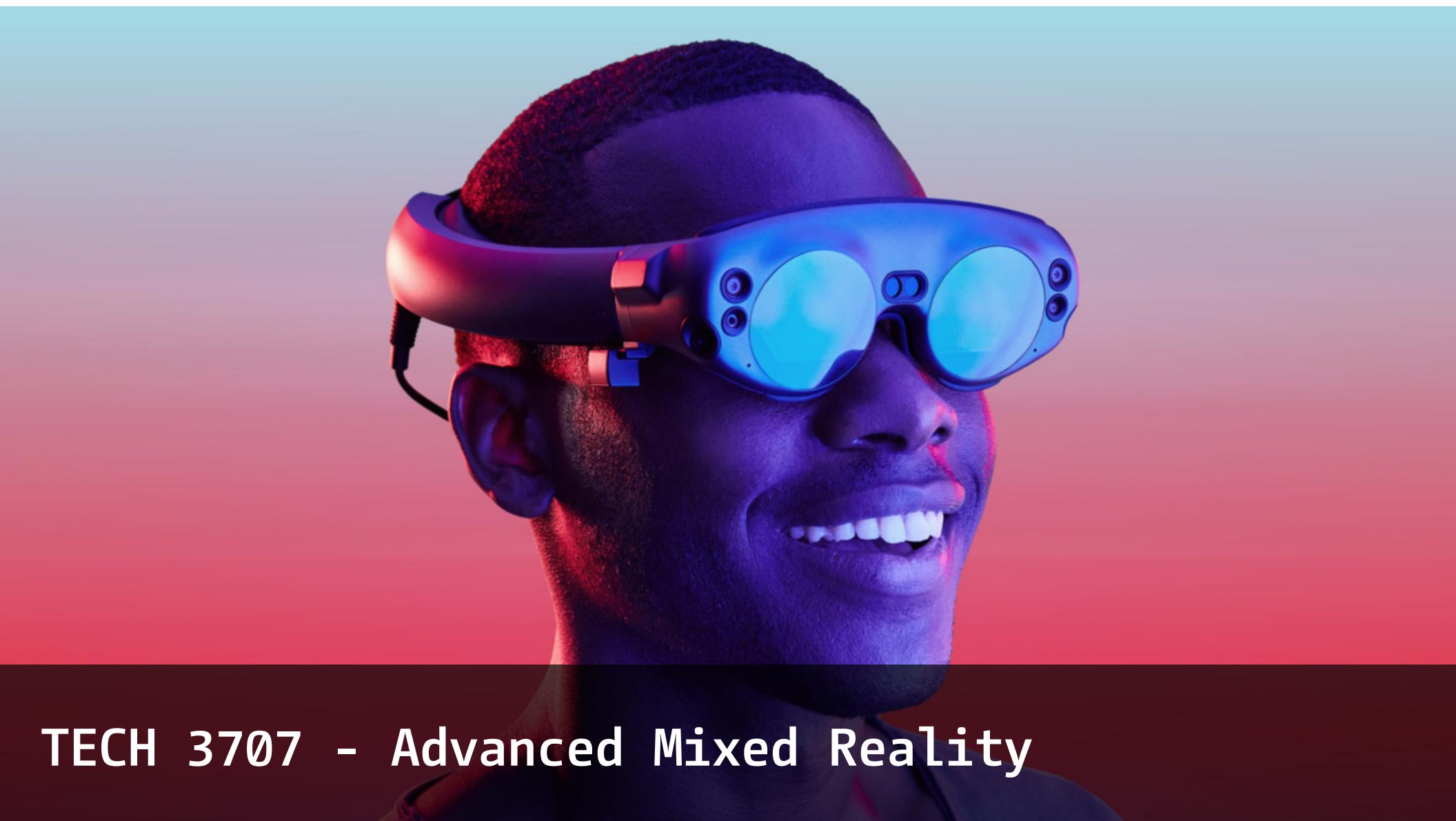
```
Debug.DrawRay(Vector3 lineStartCoordinate, Vector3 lineDirection, Color color);
```

**ARx Designers: The Future Kings and
Queens of Silicon Valley**

<https://goo.gl/dfWGeA>

Design For Humanity - Parts 4,5

<https://goo.gl/Jv5EFZ>



TECH 3707 - Advanced Mixed Reality