UMKC

COESC AI+AR/VR

Spring 2025

View these Slides:
https://docs.google.com/presentation/d/1utVYhSMCL8EviY2QNkuyc3JoUDwlz9j4vyebENd_kko/edit?usp=sharing

Luke Miller

Course Summary

- Week 1: Foundations of Unity and ML Integration -   Jan 25, 2025
  - ○ Session 1: Setting Up and Unity Basics
  - ○ Session 2: ML Basics and Unity Integration
- Week 2: Building the Final AR Project - Feb 1, 2025
  - ○ Session 1: AR Development with Meta Oculuses
  - ○ Session 2: Finalizing and Presenting the AR Project

UMKC

Staff

Instructor

Yugyung Lee, PhD

Professor of Computer

University of Missouri - Kansas City

Email: leeyu@umkc.edu

Phone: 816-235-5932

Assistant Instructor

Luke Miller

Computer Science PhD Student

University of Missouri - Kansas City

Email: ljmbm5@umkc.edu

Logistics and Materials

- **Prerequisites:**
  - Basic programming knowledge (Python and/or C# preferred).
- **Tools and Setup:**
  - Python (with TensorFlow /PyTorch) for ML sessions.
  - Unity (with Oculus SDK) for AR/VR sessions.
  - Meta Oculuses provided for hands-on activities.
- **Team Size:**
  - Small groups (2–3 participants) to encourage collaboration.
- **Resources Provided:**
  - Presentation Materials
  - Pre-configured Software
  - Sample Datasets.
  - Tutorials

Objective:

- **By the end of the course, participants will:**
  - Understand the basics of ML and its applications in AR/VR.
  - Be familiar with using Meta Oculuses for AR development.
  - Create a simple, functional AR/VR application enhanced with an ML feature.

UMKC

COESC AI+AR/VR

- Week 1: Foundations of Unity and ML Integration -   Jan 25, 2025
    - Session 1: Setting Up and Unity Basics
    - Session 2: ML Basics and Unity Integration
    - Session 3: Tutorial/Hands-on Assignment

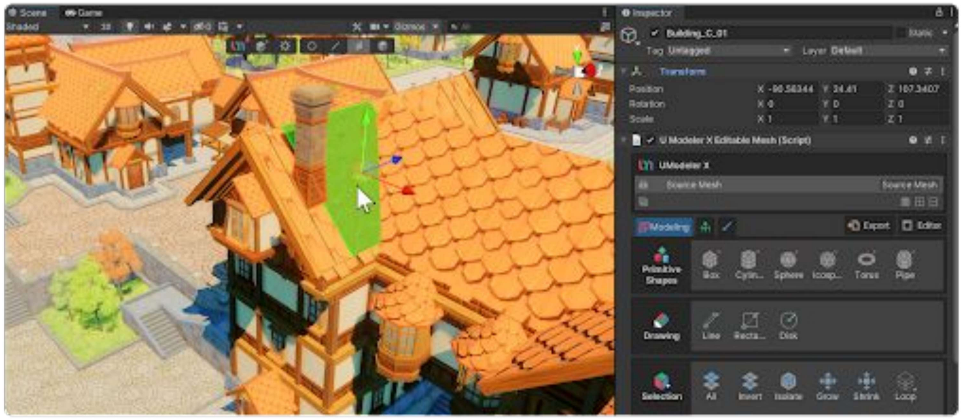
UMKC

Session 1: Setting Up and Unity Basics

- Getting Started with Unity
  - What is Unity
  - Overview of the Unity interface
- Building the First Scene
  - Adding objects (3D models, UI elements) and adjusting transforms.
  - Adding physics components: Rigidbodies and colliders.
- Basic Interactions with Unity
  - Adding scripts (in C#) to objects for interactivity.

Introduction to Unity

- What is Unity?
- Why is it used in AR/VR development?
- Installing and Configuring Unity



Building_C_01

Tag Untagged | Layer Default

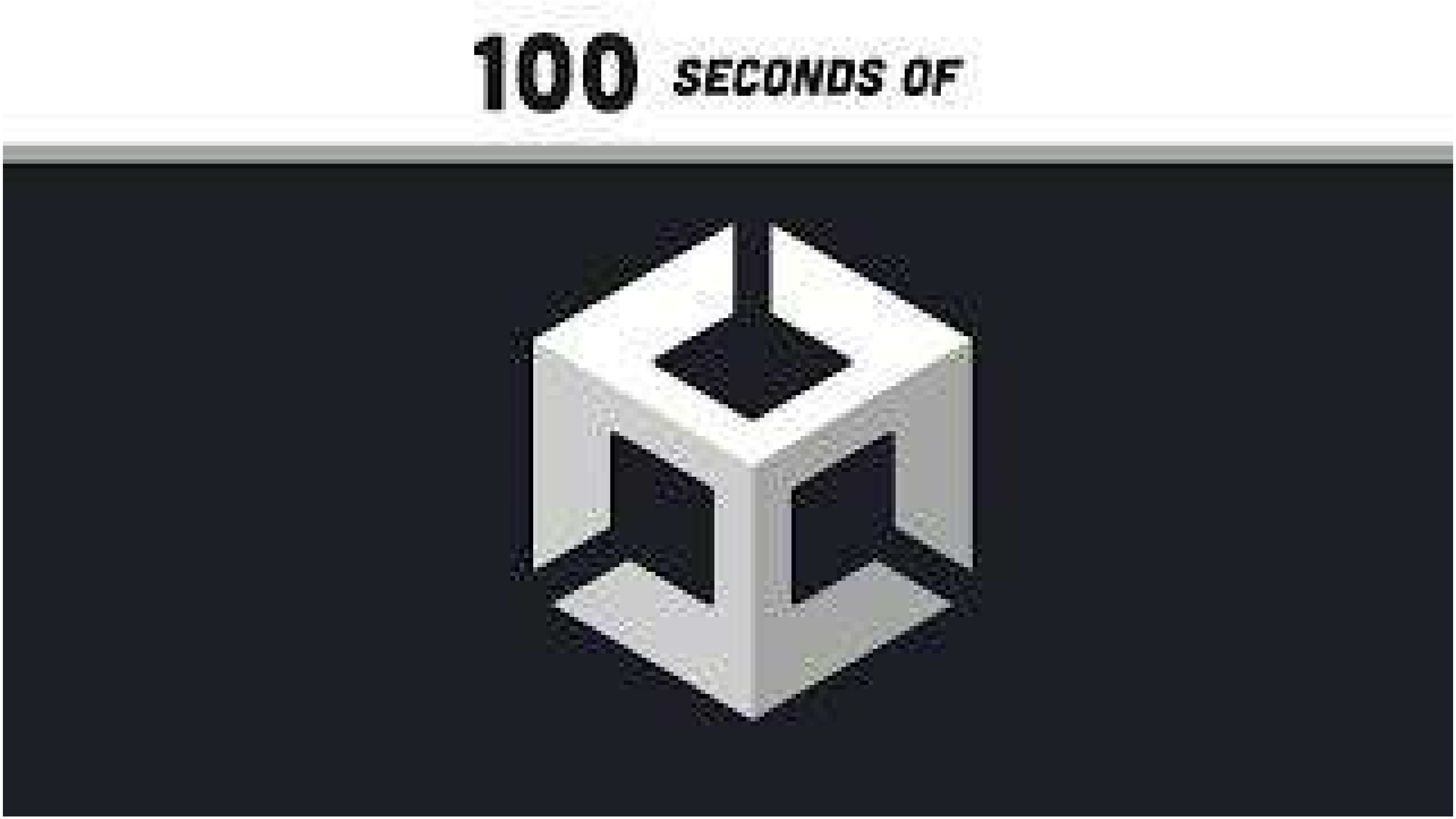
Transform

Position X -90.58344 Y 34.41 Z 107.3407

Rotation X 0 Y 0 Z 0

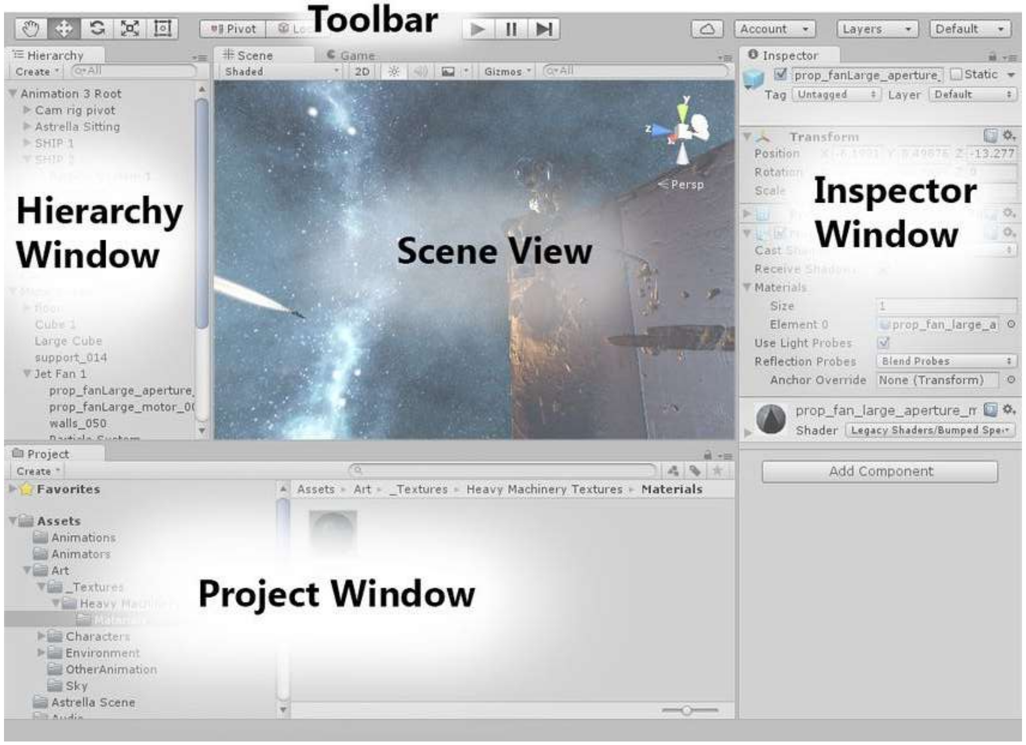
Scale X 1 Y 1 Z 1

U Modeler X Editable Mesh (Script)

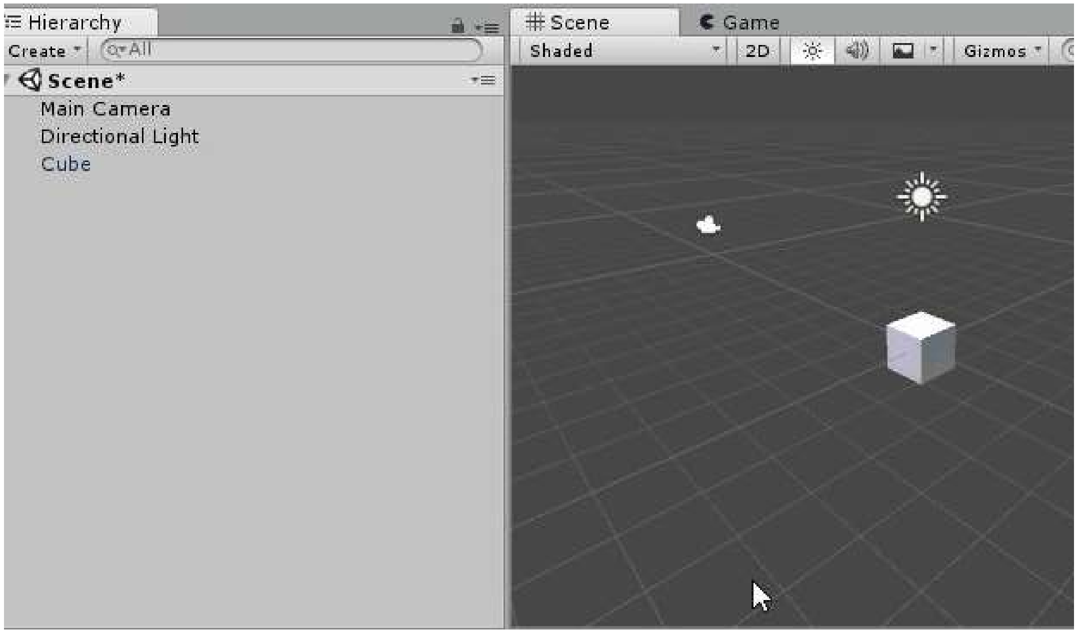

100 SECONDS OF

Unity Interface Overview

- Key components:
  - Hierarchy: List of all objects in the scene.
  - Scene View: Visual representation of the scene.
  - Inspector: Object properties and adjustments.
  - Toolbar: Provides options to the user
  - Project Window: Asset management.



Unity Interface: Hierarchy



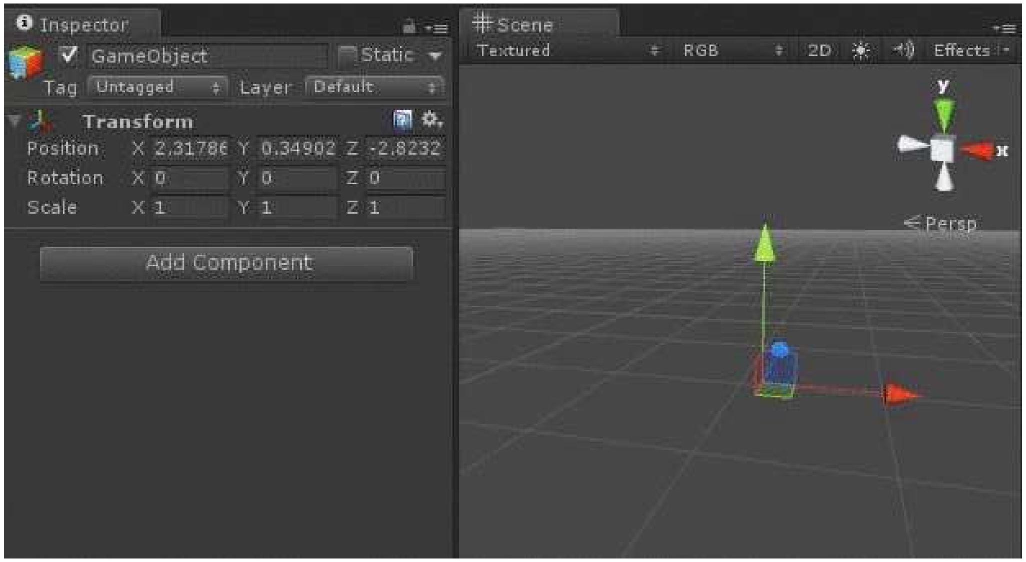
- The Hierarchy window displays every GameObject in a Scene
- You can use the Hierarchy window to sort and group the GameObjects you use in a Scene.
- When you add or remove GameObjects in the Scene view, you also add or remove them from the Hierarchy window.

Learn more: https://docs.unity3d.com/Manual/Hierarchy.html

UMKC

Unity Interface: Scene View

- The Scene view is where you visualize and interact with the world you create in the Editor.
- In the Scene view, you can select, manipulate, and modify GameObjects: scenery, characters, cameras, lights, and more.



Inspector

☑ GameObject □ Static

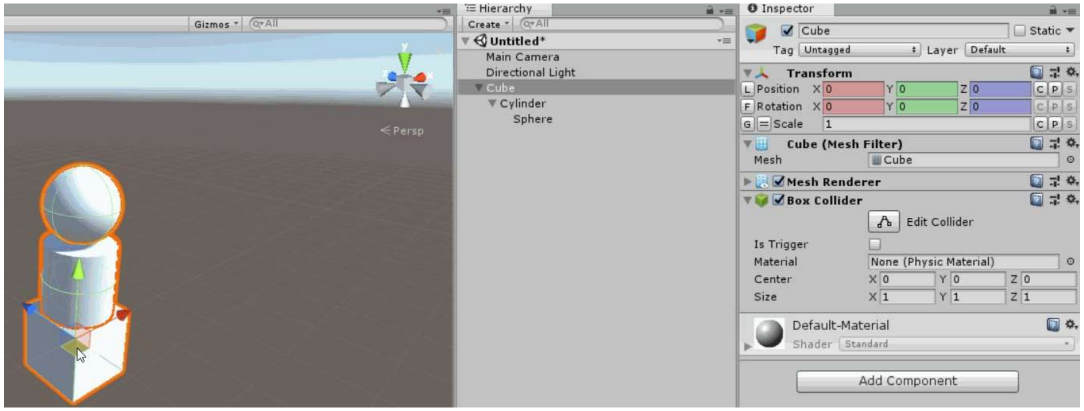
Tag Untagged Layer Default

Transform

Position	X 2.31786	Y 0.34902	Z -2.8232
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

https://docs.unity3d.com/Manual/UsingTheSceneView.html

Unity Interface: Inspector



Untitled*
Main Camera
Directional Light
Cube
Cylinder
Sphere

Cube
Static
Tag Untagged
Layer Default

Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1		

Cube (Mesh Filter)
Mesh Cube

Mesh Renderer

Box Collider
Is Trigger □
Material None (Physic Material)

	X	Y	Z
Center	0	0	0
Size	1	1	1

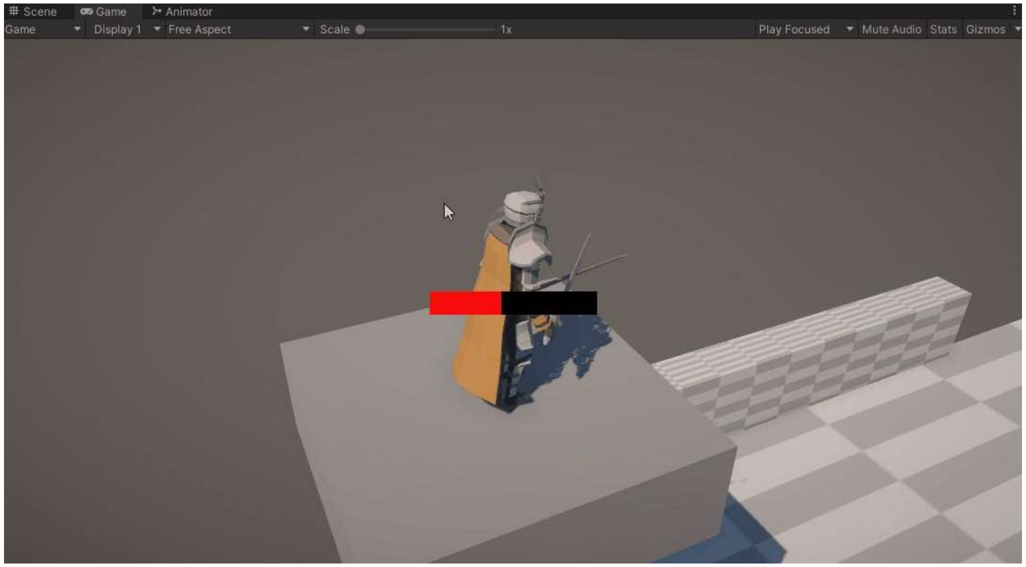
Default-Material
Shader Standard

- Use the Inspector window to view and edit properties and settings for almost everything in the Unity Editor:
  - GameObjects
  - Unity components
  - Assets
  - Materials
  - In-Editor settings and preferences

https://docs.unity3d.com/Manual/UsingTheInspector.html

Unity Interface: Game View

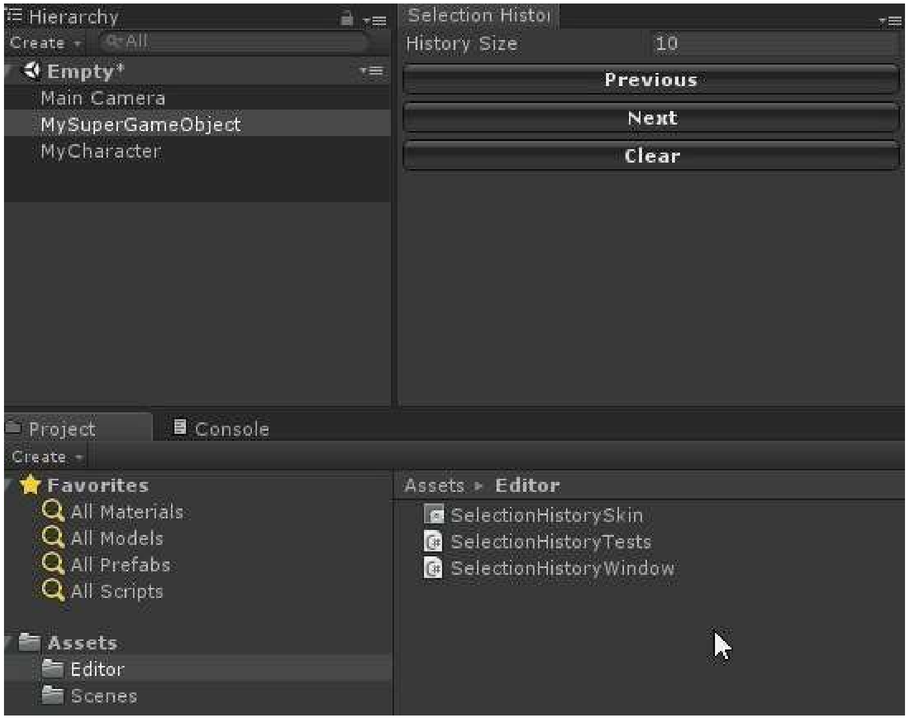
- The Game view is rendered from the Cameras in your application.
- The Game view displays how your final, built application looks.
- You need to use one or more Cameras to control what the player sees when they're using your application.



https://docs.unity3d.com/Manual/GameView.html

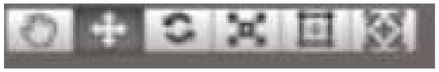
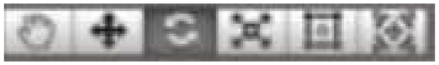
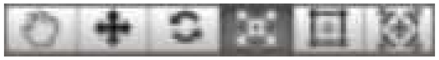
Unity Interface: Project Tab

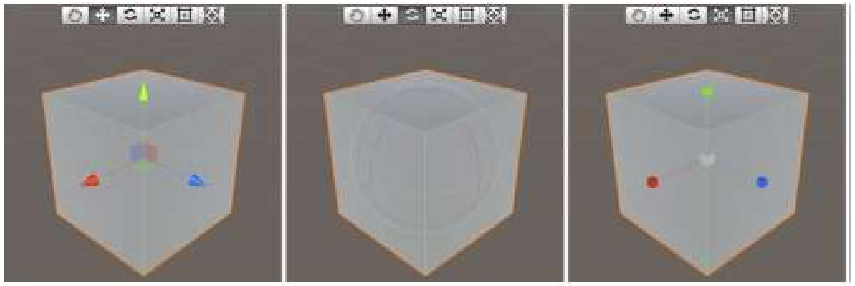
- The Project window displays all of the files related to your Project
- The main way you can navigate and find Assets and other Project files in your application.
- When you start a new Project by default this window is open.



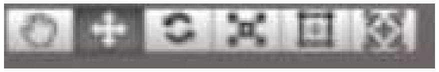
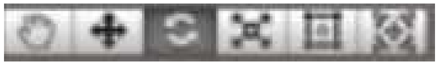
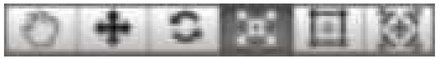
History Size 10

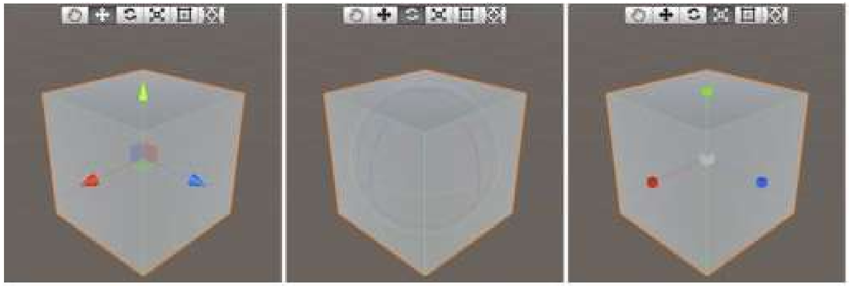
Building the First Scene

- Add 3D objects (cube, sphere, plane).
- Move, Rotate, and Scale tools.
    - Hand Tool. Pans around.
    - Move tool 
    - Rotate Tool 
    - Scale Tool 
-

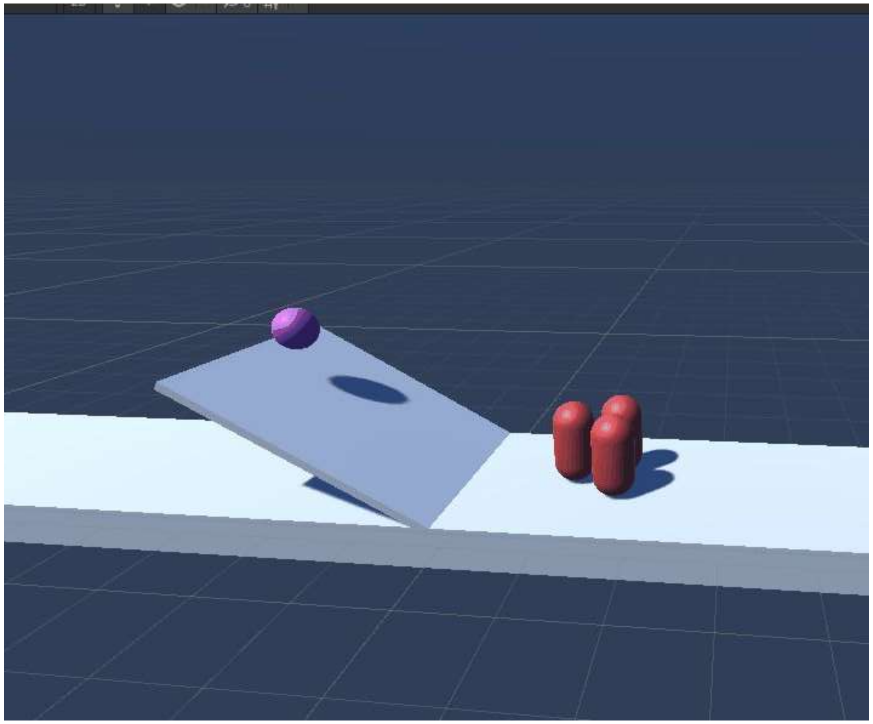


Building the First Scene: Objects

- Add 3D objects (cube, sphere, plane).
- Move, Rotate, and Scale tools.
  - Hand Tool. Pans around.
  - Move tool 
  - Rotate Tool 
  - Scale Tool 

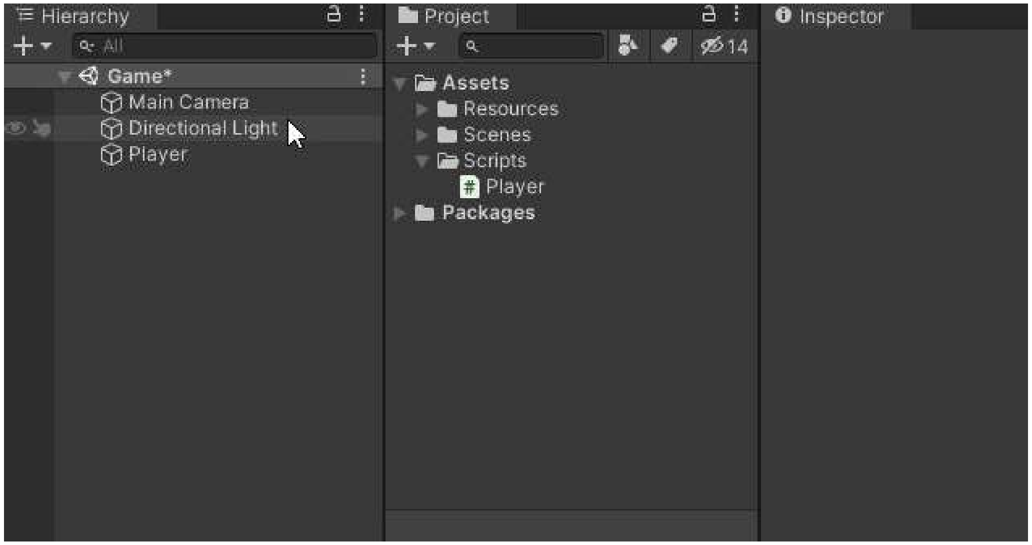


Building the First Scene: Physics



- Rigidbodies
  - Adding a Rigidbody component to an object will put its motion under the control of Unity's physics engine.
  - Even without adding any code, a Rigidbody object will be pulled downward by gravity and will react to collisions.
- Colliders
  - Collider components define the shape of an object for the purposes of physical collisions.
  - A collider is invisible
  - A rough approximation of the visual shape is often more efficient and indistinguishable in gameplay.

Basic Interactions with Unity



- What are Scripts
  - Scripts allow you to customize and extend the capabilities of your applicaton with C# code.
  - Unlike most other assets, scripts are usually created within Unity directly.
- Adding Scripts to Entities
  - From the main menu: go to **Assets > Create > Scripting** and select the type of script you want to create. *OR*
  - From the Create menu (plus sign) in the Project window toolbar: go to Scripting and select the type of script you want to create.

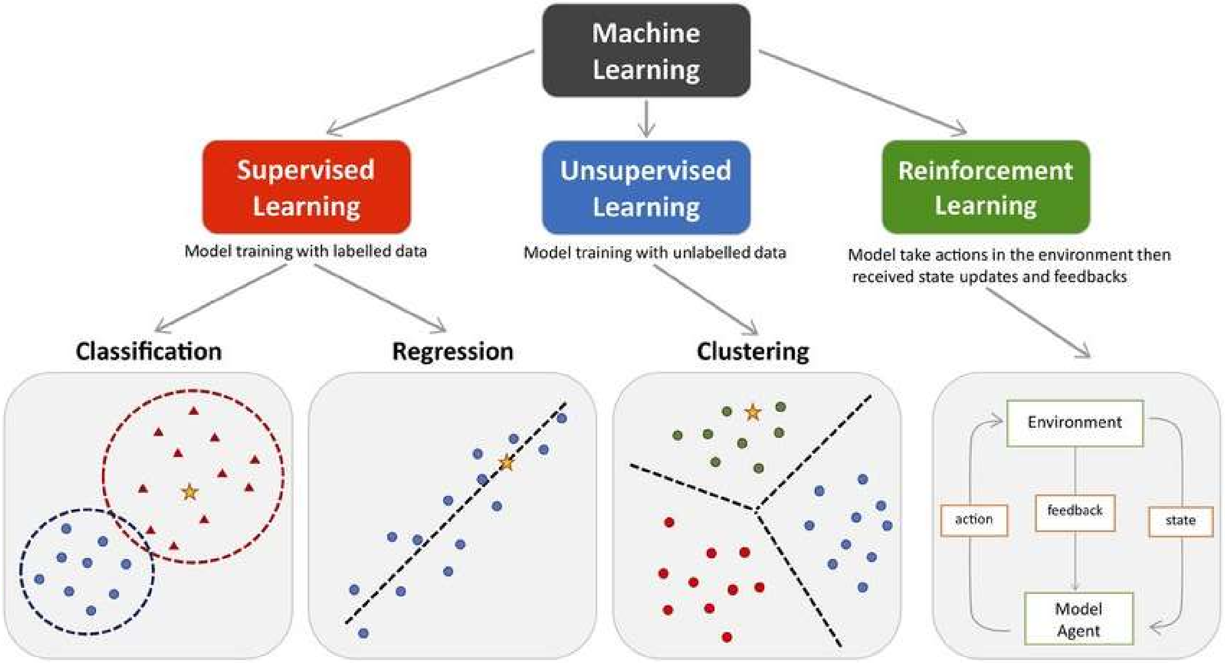
UMKC

Session 2: ML Basics/Unity Integration

- Introduction to ML
    - What is Machine Learning
    - ML applications in AR/VR
- Pre-Trained ML Models
    - Why use pre-trained models?
    - Examples of pre-trained models
    - How models process images
- Integrating ML into Unity
    - Tools for integration
    - Real-time object detection in Unity

Introduction to Machine Learning

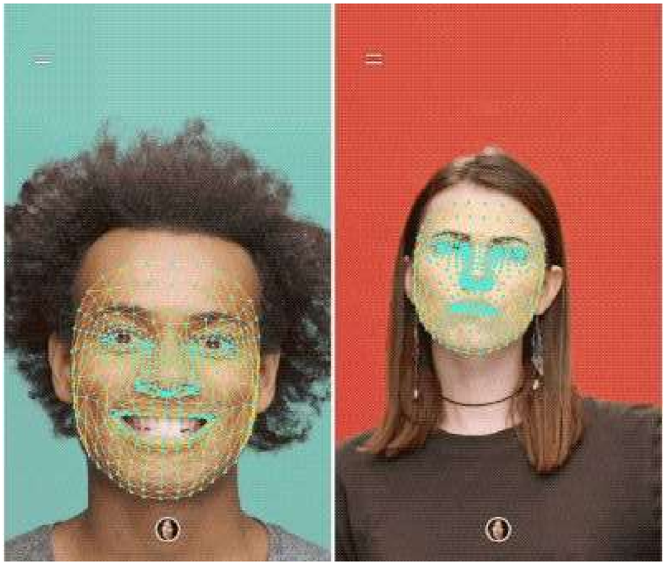
Introduction to Machine Learning



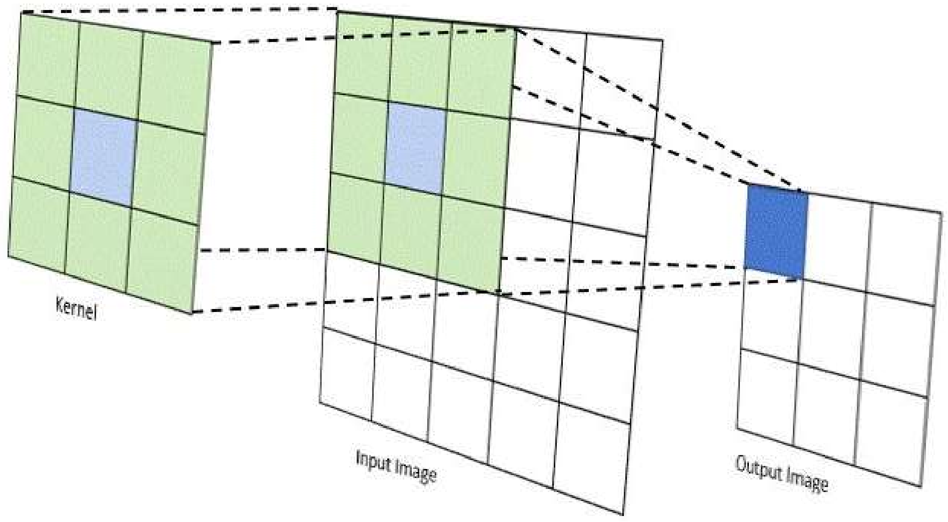
- Key concepts:
    - Definition
    - training data
    - models
    - predictions.
- Types of ML:
    - supervised
    - unsupervised
    - reinforcement learning.

ML Applications in AR/VR

- Object Detection
  - Identifying and locating objects within an image or video.
  - Commonly used for interacting with real-world objects in real time.
- Gesture Recognition
  - Enables systems to identify and interpret hand gestures or body movements
  - Creates intuitive controls: pinch-to-zoom or swipe gestures
- Scene Understanding
  - Analyzes an environment to identify its structure and the objects within it.
  - Maps walls, floors, and furniture in a room
  - Recognizes outdoor features like roads and trees.
  - Anchors virtual elements to real-world surfaces or adjust the experience based on the environment.



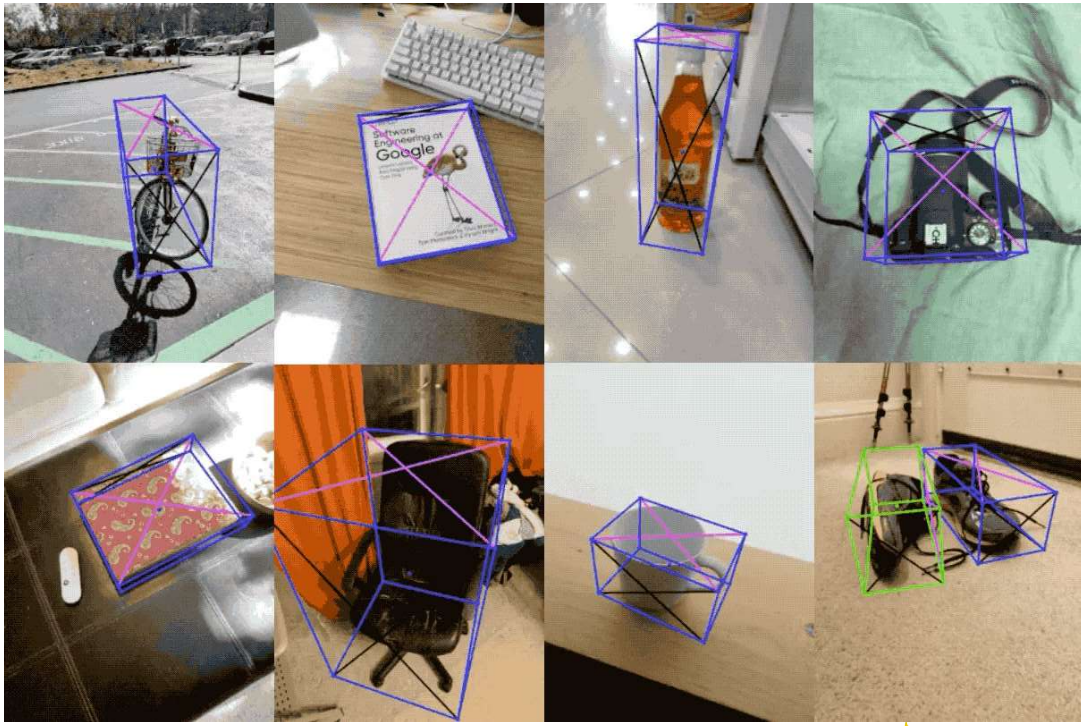
Pre-Trained ML Models



- Why use pre-trained models?
  - Ease of use
  - Saves training time
- Examples of pre-trained models:
  - MobileNet (lightweight image classification).
  - YOLO (real-time object detection).
- How models process images:
  - Convert input (images) to numerical data.
  - Perform predictions based on trained weights.

Integrating ML into Unity

- Tools for integration:
  - Unity Barracuda:
    - Lightweight and Unity-native.
  - TensorFlow for Unity:
    - Flexible and widely used.
- Real-time object detection in Unity:
  - Example workflow:
    - Feed live camera input to an ML model →
    - Process predictions →
    - Highlight detected objects.

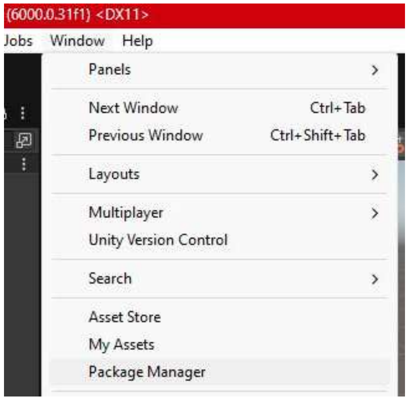


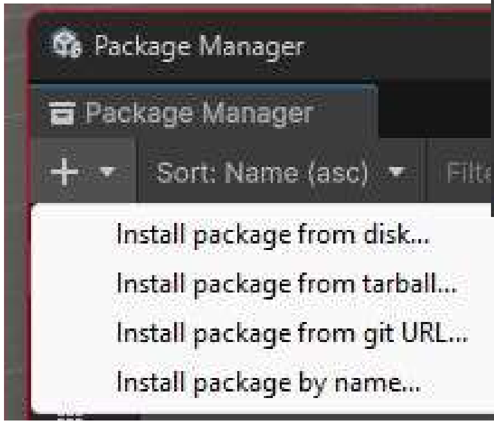
Session 3: Tutorial/Hands-on Assignment

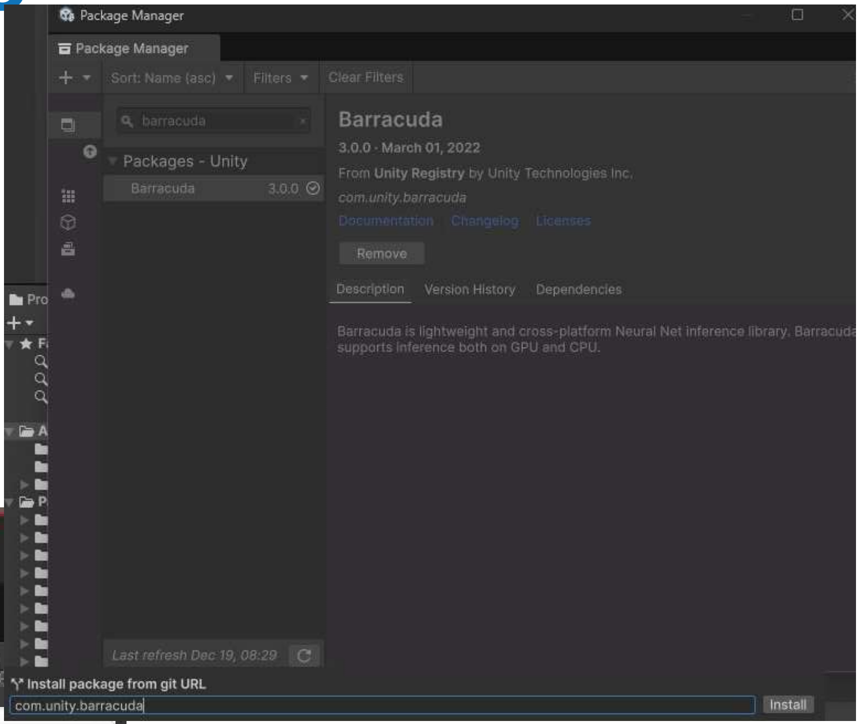
- Setup Unity for ML Integration
- Create the Scene
- Integrate the ML Model
- Display Object Detection Results

Setup Unity for ML Integration-Barracuda

- Install Unity Barracuda
  - Open your Unity project.
  - Go to Window → Package Manager.
  - Click the Plus Sign
  - Select "Add package from git url
  - Type "com.unity.barracuda"
  - Click Install




Package Manager
Package Manager
+ ▾
Sort: Name (asc) ▾
Filte
Install package from disk...
Install package from tarball...
Install package from git URL...
Install package by name...



Barracuda

3.0.0 · March 01, 2022

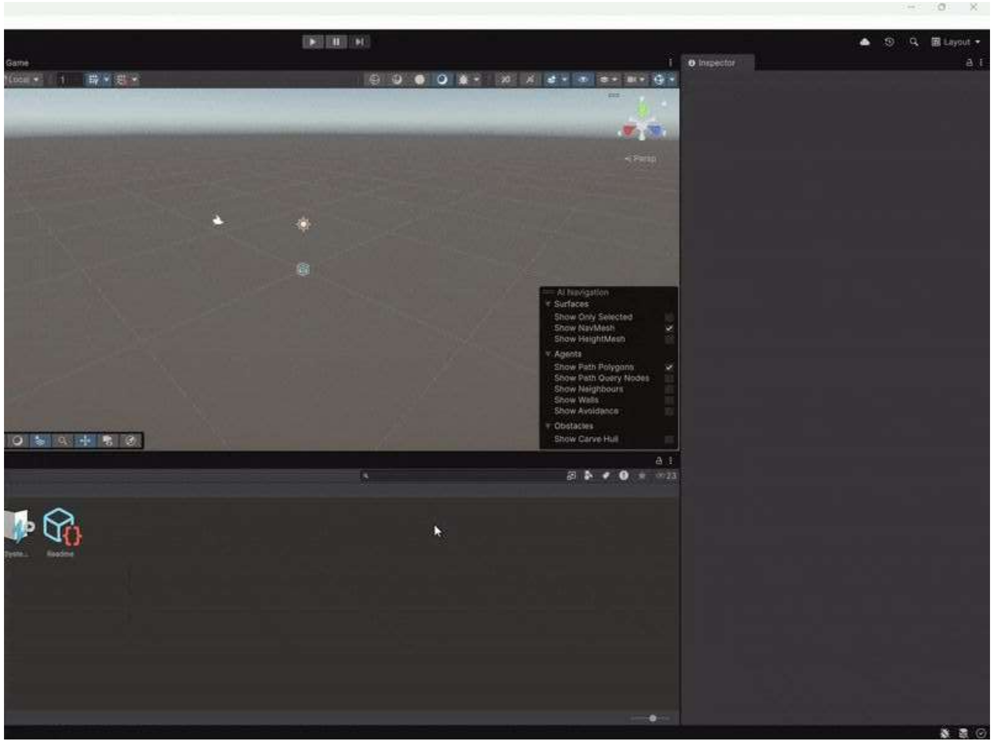
From Unity Registry by Unity Technologies Inc.

com.unity.barracuda

Barracuda is lightweight and cross-platform Neural Net inference library. Barracuda supports inference both on GPU and CPU.

com.unity.barracuda

Setup Unity for ML Integration-Models



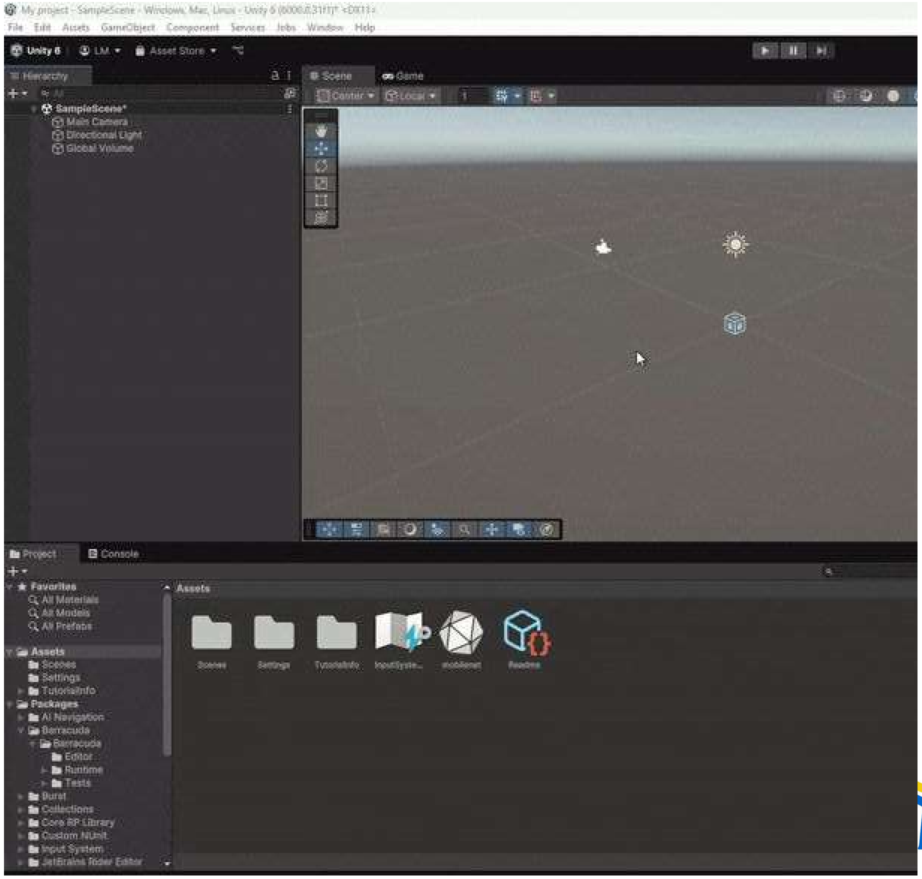
AI Navigation

- Surfaces
  - Show Only Selected □
  - Show NavMesh ☑
  - Show HeightMesh □
- Agents
  - Show Path Polygons ☑
  - Show Path Query Nodes □
  - Show Neighbours □
  - Show Walls □
  - Show Avoidance □
- Obstacles
  - Show Carve Hull □

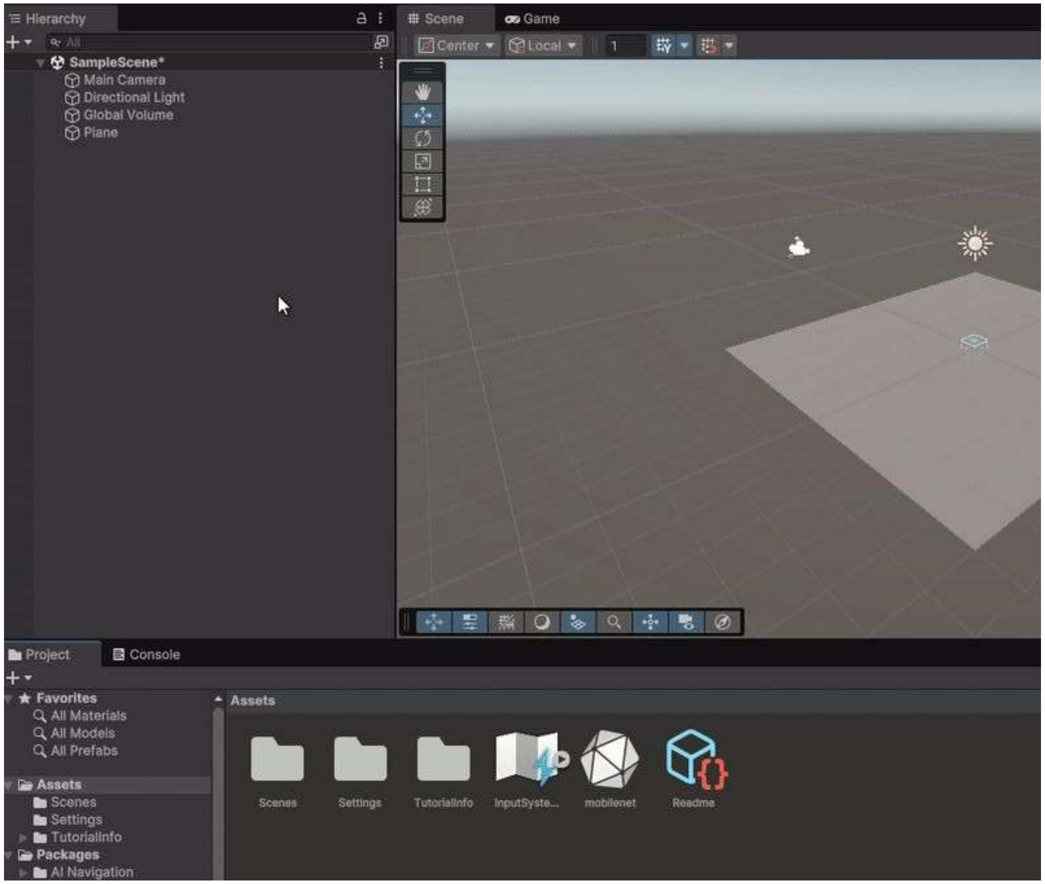
- Download and Import a Pre-Trained Model
  - Download a pre-trained MobileNet model in ONNX format from TensorFlow Hub. We've done this for you, and it is available on the GitHub.
  - Drag and drop the .onnx file into Unity's Assets folder.
  - In real life, you would need to convert the keras, tf2, or HD5 model to the onnx format (Tutorial). We have already done this, and it is on the Github.

Create the Scene - Add a Plane

- Add a Plane for the Camera Feed
- In the Hierarchy window:
  - Right-click
  - 3D Object
  - Plane.
- Position the plane so it's visible to the main camera.



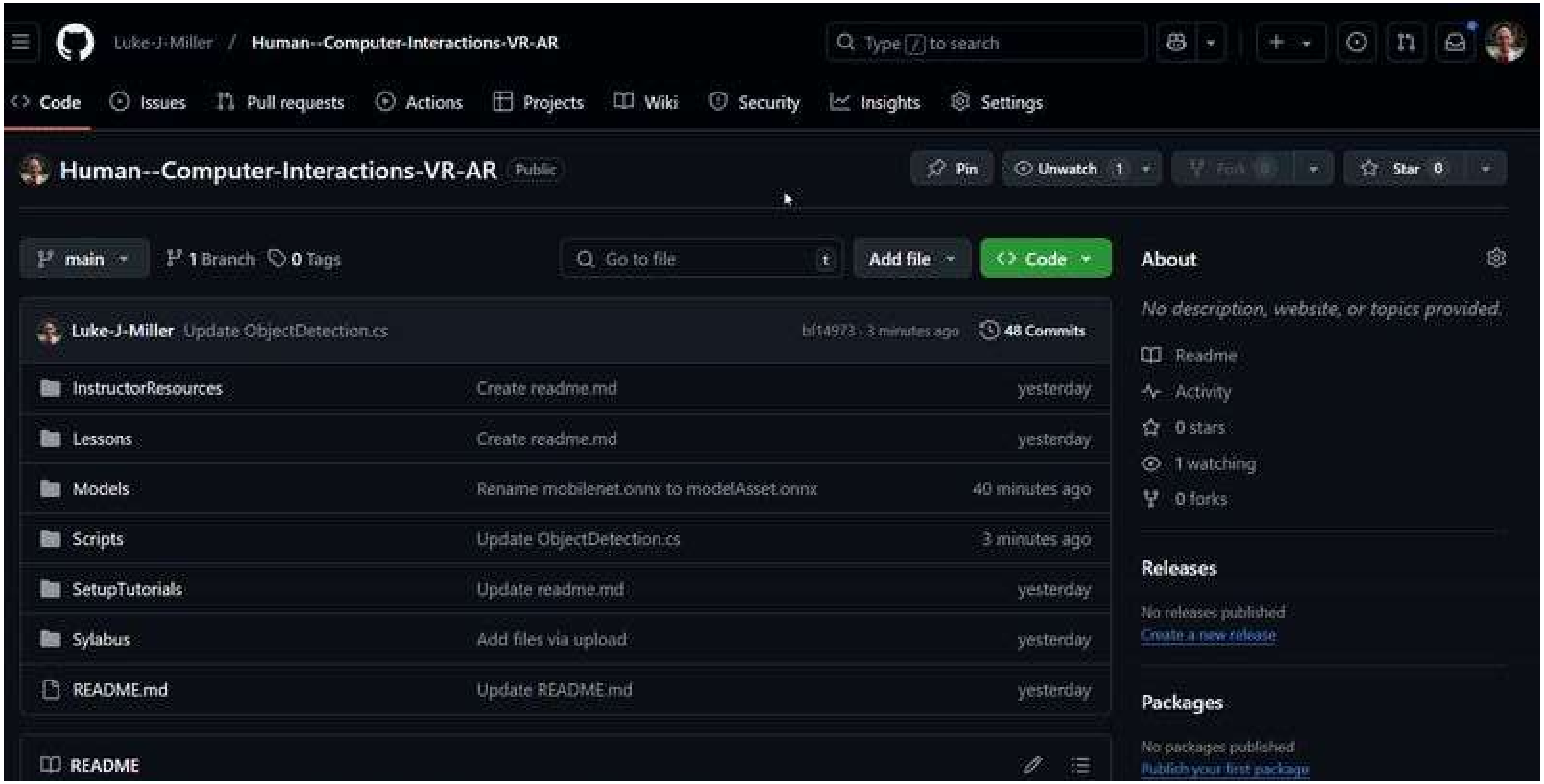
Create the Scene - Configure Camera



- Select the Main Camera in the Hierarchy.
- Create a material for displaying the live webcam feed:
  - Right-click in Assets
  - Create
  - Material.
- Name it 'webcamMaterial'.

Integrate the ML Model - Model Processor Script

- Create a new script called ObjectDetection.cs and attach it to an empty GameObject. Code available on the Github



Luke-J-Miller Update ObjectDetection.cs bf14973 · 3 minutes ago 48 Commits

InstructorResources	Create readme.md	yesterday
Lessons	Create readme.md	yesterday
Models	Rename mobilenet.onnx to modelAsset.onnx	40 minutes ago
Scripts	Update ObjectDetection.cs	3 minutes ago
SetupTutorials	Update readme.md	yesterday
Sylabus	Add files via upload	yesterday
README.md	Update README.md	yesterday

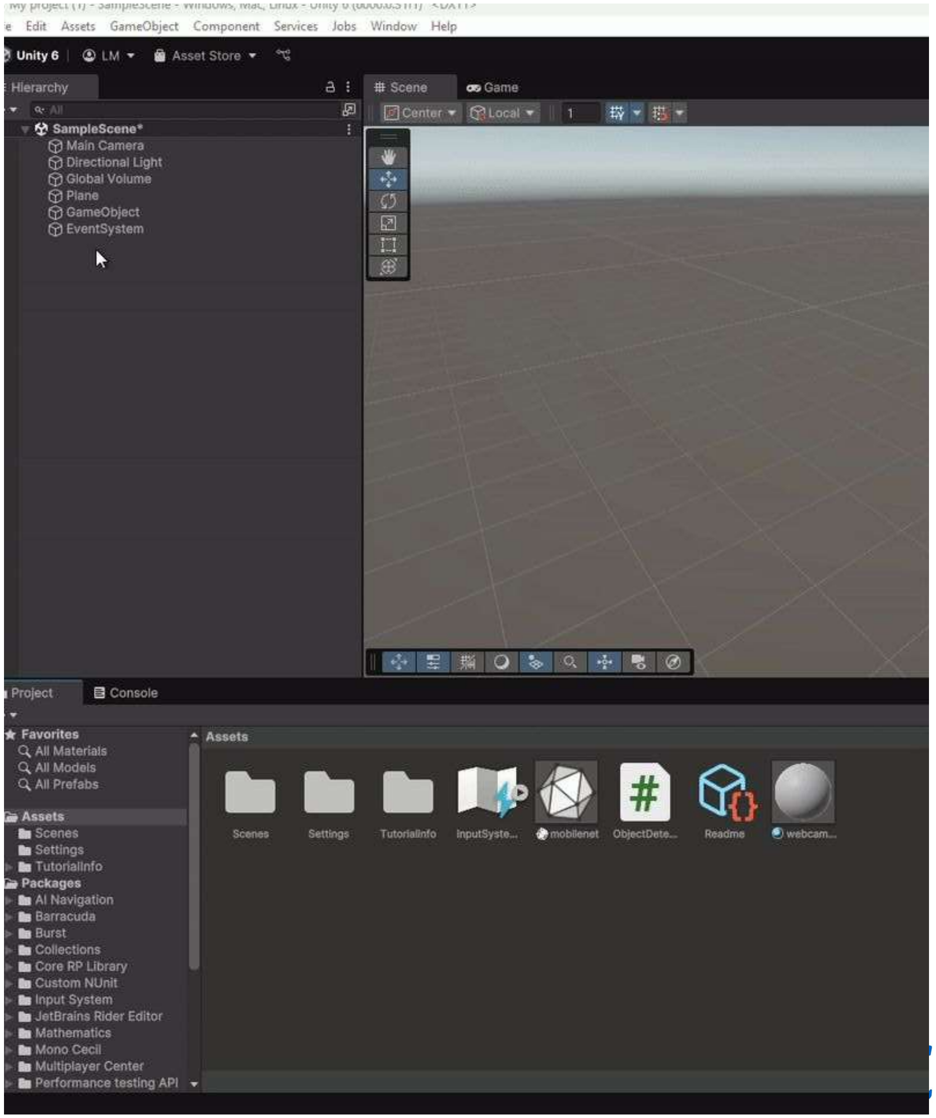
No description, website, or topics provided.

No releases published
Create a new release

No packages published
Publish your first package

Display Object Detection Results

- Add a Canvas to the scene:
  - Right-click in the Hierarchy
  - UI
  - Canvas.
- Set the Render Mode of the Canvas to:
  Screen Space - Overlay.
- Add a placeholder for bounding boxes:
  - Right-click the Canvas
  - Create Empty
  - Rename to "BoundingBoxesContainer".



Create a UI Prefab for Bounding Boxes

- Create a new Panel:
  - Right-click in the Canvas
  - UI
  - Panel.
- Resize the panel to a small rectangle (this will represent a bounding box).
- Add a Text element as a child of the Panel:
  - Right-click the Panel
  - UI
  - Text.
  - Style the text to show labels and confidence scores (adjust font size and color).
- Convert the Panel to a Prefab:
  - Drag it from the Hierarchy to the Assets folder.
- Delete the original Panel from the scene.

UMKC