

# CS-467

---

## ML Breakout Online Capstone

### Introduction

The Atari game system was developed in the 1970s and was a pioneer in home video game consoles. Being one of the first game consoles, the games created for this system are well known and nostalgic for many kids and young adults of the 1970s through 1990s. Breakout was a game developed initially for arcade machines, but in 1976 was released in color for the Atari system. The game was a huge success and was among the top five highest grossing arcade games of 1976. The game itself drew inspiration from pong and consists of a paddle, ball, and eight rows of tiles. The goal is to knock all the tiles down without letting the ball travel past the paddle.

In this project, the team will design a clone of the game Breakout in the Unity Game Engine and train a machine learning (ML) algorithm to play the game. The clone needs to allow the trained ML algorithm to interface with the game along with a two player mode that allows a human player to play against the algorithm. The two player mode will allow users to test themselves against trained ML algorithms. This document will discuss the design requirements, architecture, and project plan to build and train an Atari Breakout clone and ML algorithm.

### Problem Statement

The goal of this project is to build an Atari Breakout clone and train a ML algorithm to play the game. The game was developed in the 1970's for arcade machines and the Atari system. The game consists of three elements: balls, paddles, and tiles. The aim of the game is to knock all the tiles down by bouncing the ball into the tiles while not losing the ball by allowing it to touch the bottom of the screen. The clone is to be recreated in the Unity Game Engine. A ML model will be trained to play the game and compete against their human counterparts.

To achieve the project goals, the problem can be split into two sub tasks. The first is to build an Atari Breakout clone to run on modern systems. The second is to build ML models to compete against the user to add player vs environment (computer) or PvE. Including ML models also allows for developers to train, test, and experiment with different algorithms.

The incorporation of trained ML models playing the game offers the user further challenges to test their skill in the game. While this offers a harder challenge, it maintains the game's roots and nostalgia all the while bringing the game to modern systems. The ML model adds the PvE that most games provide today such as games like chess. Developers will also be able to test different models and learn about training costs and performance. The Unity Game Engine will be used to build the Breakout clone. The game engine is cross platform and widely used by many games. An open source unity project, Unity ML-Agents toolkit, will enable and facilitate training of ML models.

### Requirements

The goal of this project is to build a modern version of the arcade game Breakout, where the player controls a paddle to bounce a ball and clear a grid of bricks. The project will be developed in Unity, and just their ML

agent for building a NN and providing a 2nd player AI option. This will involve training a neural that can respond to ball movement and gameplay dynamics, offering an intelligent opponent.

At a high level, the game will feature responsive and intuitive controls, sound effects, a clean and easy to understand user interface, and multiple levels of increasing difficulty. The completed game will be tested and played locally as a standalone desktop application. Important Requirements/Features:

- UI start screen, select number of players
- User controlled paddle - moves left and right
- Ball that bounces off the paddle, walls, and bricks
- Score system - increases when bricks are hit
- Life system - limited lives, goes down when ball is missed
- Sound effects for collision and bricks
- Level system - multiple levels ranging in difficulty
- Unity system
- Creating the clone game
- Training on the NN
- Using the ML agent to control the game
- Implementing 2 player capabilities

## Design Architecture

- Game Manager
  - Controls starting the game, playing, pausing, Points, Lives, Level lost, level won
  - Loads levels, transitions between screens
  - Saves current game state
- Level Manager
  - Loads brick layouts for each level
  - Tracks when all bricks are cleared
  - Handles any change in difficulty
- Player Controller
  - Handles paddle movement based on user input
  - Ensures paddle stays within the boundaries
- Ball Controller
  - Tracks ball movement
  - Detects and responds to collisions with paddle, bricks, and walls
- Brick manager
  - Manages brick objects in the environment
  - Removes bricks upon collision
- Score and Life Manager
  - Updates score on brick destruction
  - Tracks remaining player lives
  - Triggers game over screen when lives reach zero

## Known Bugs:

- Split Screen currently is unavailable.

## Code Location

### Github

1. Open [scripts](#)

### Alternatively

1. Go to [Github Repo](#)
2. Open Assets Directory
3. Open Scripts Directory

### Unity

1. Launch the project
2. Locate the project tab
3. Open the Assets directory
4. Open the scripts directory
5. double-click any script to view in Microsoft Visual Studio

## Installation:

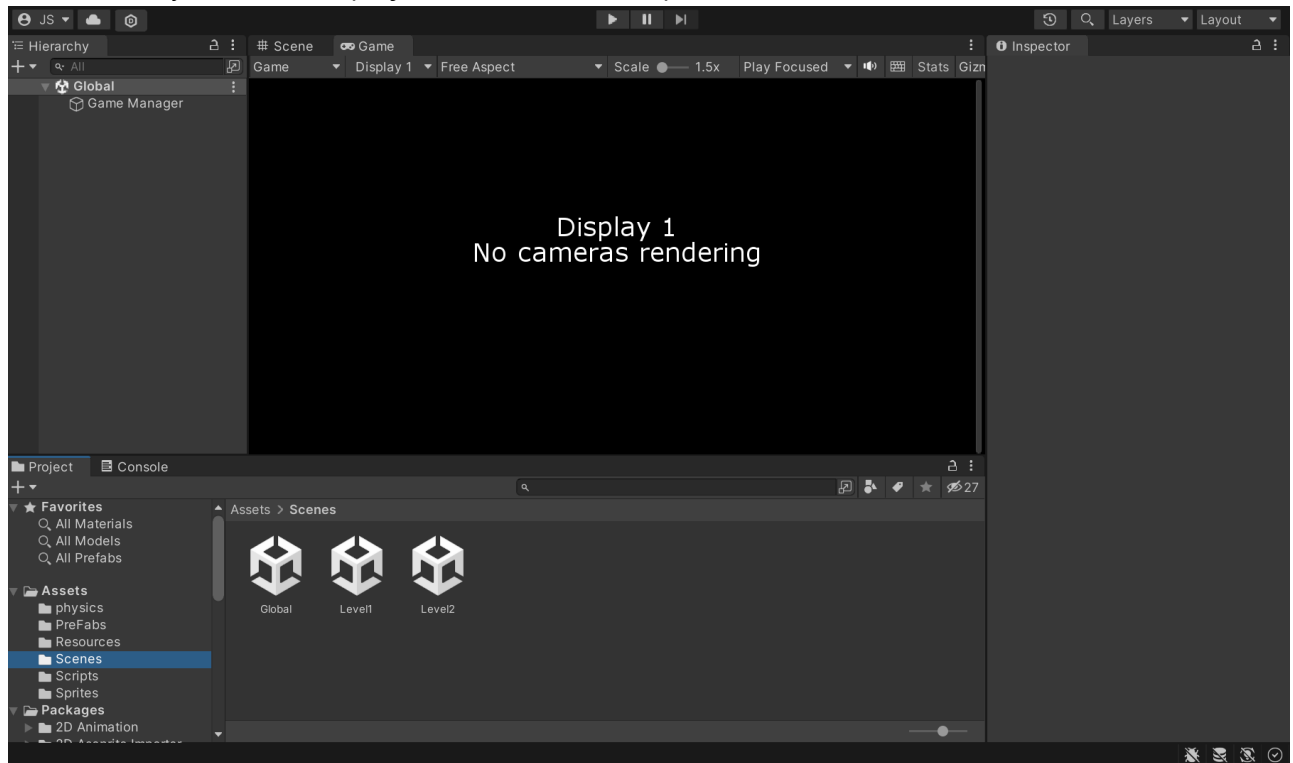
### Release Instructions:

1. Dowload Zip file (ML-Breakout.zip) from [GitHub Releases](#).
2. Navigate to the zip file (Downloads as ML\_Breakout.zip).
3. Extract the zip file (ML\_Breakout.zip).
4. Open the extracted directory.
5. Locate the .exe file (CS-467.exe).
6. Double-click to open the game.

### Developer Instructions:

1. Navigate to the [Github Repository](#).
2. Clone the repository or download the source code as a zip from [GitHub Releases](#).
3. Install [Unity Hub](#)
4. Open Unity Hub
5. Go to installs and install Unity (2022.3.61f1) LTS. Unity may prompt you to install Microsoft Visual Studios. Do so to be able to view code scripts.
6. After downloading or cloning this repository, follow these steps to open and play the game in Unity.
  - a. Open Unity Hub
  - b. Click Open
  - c. Navigate to the folder where you saved the project

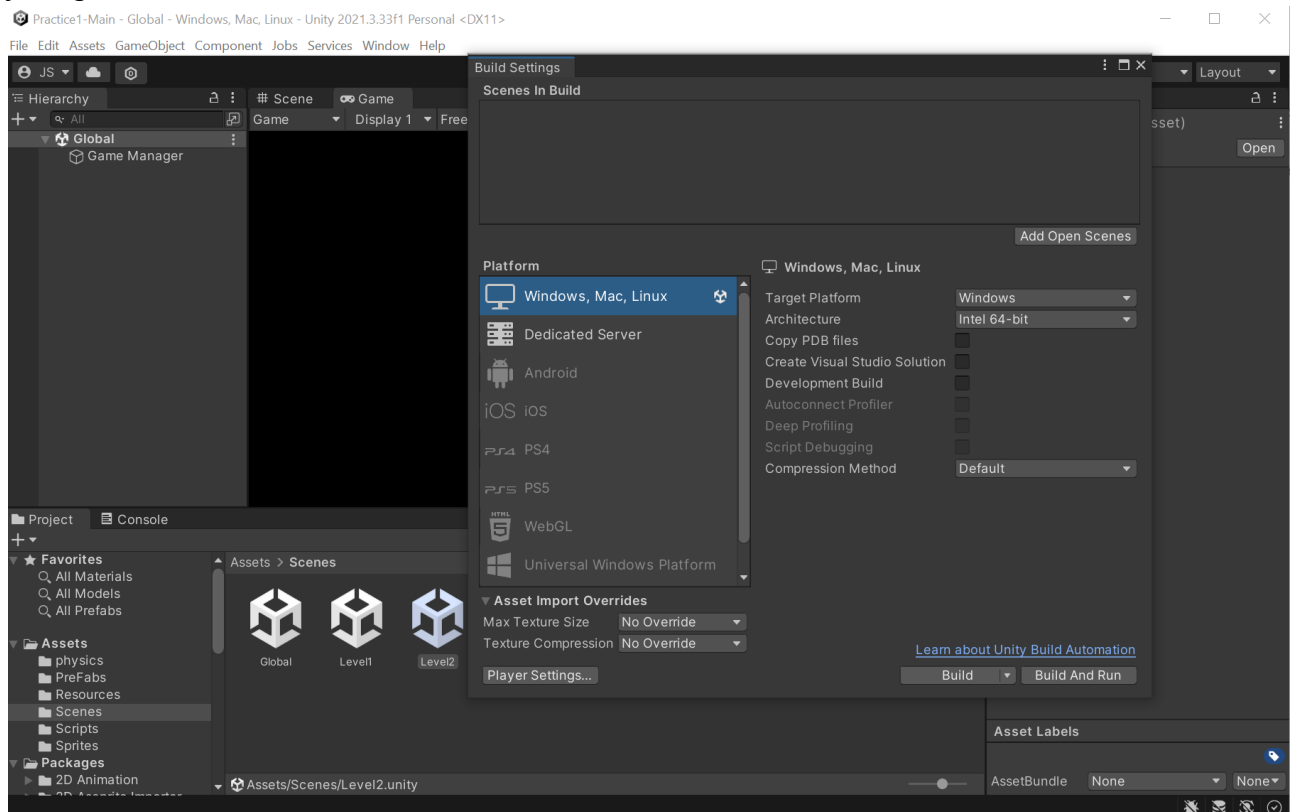
d. Select the root project folder and click Open e. Open Unity Hub and select "Open". Navigate to the folder where you saved the project, then select and open it.

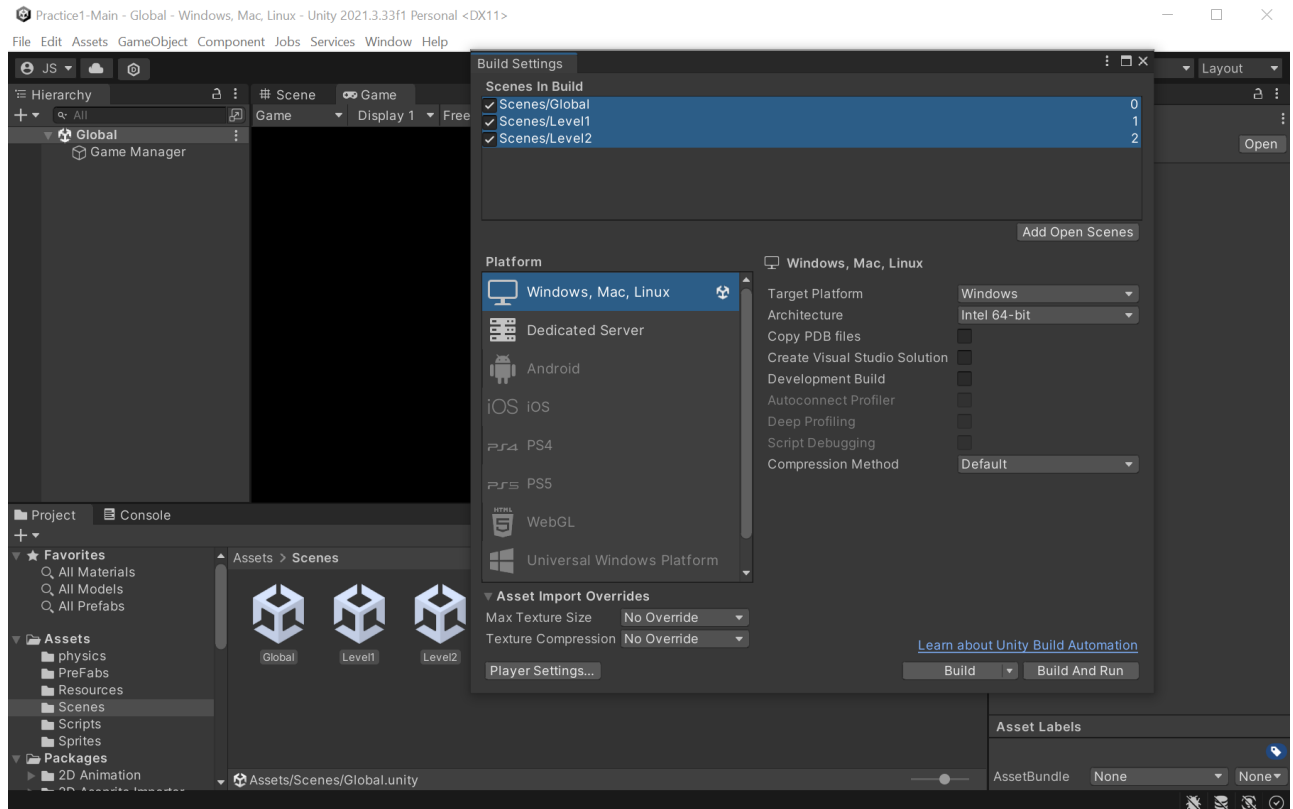


f. Add Scenes to Build Settings g. In Unity, go to File > Build Settings

h. In the Project pane, navigate to: i. Assets > Scenes > Global, Level1, Level2, Menu, GameOver

j. Drag all scenes into the Scenes In Build window





7. Running the Game a. In the Hierarchy panel, click on the Global object

b. Press the Play button (top center in Unity)

c. The game should compile and start!

8. Controls

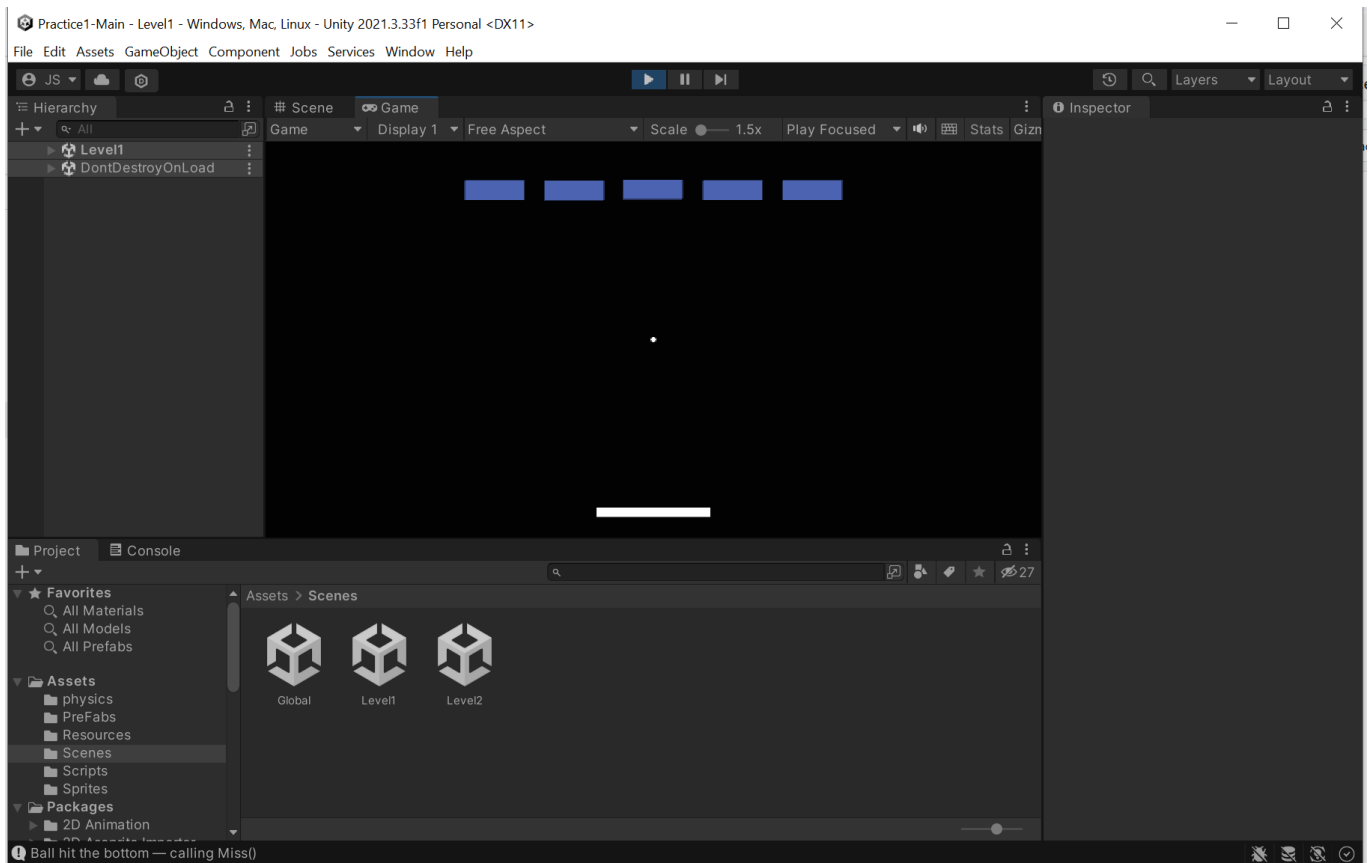
Move the paddle using:

A to move left

D to move right

Launching the ball:

Space



Note:

If the ball is missed, the paddle and ball will reset