

Project Name
ML-BREAKOUT

Team Members
Jared Connor
Tingting Fang
I Chun Hurng
Jacob Silverberg

Course
CS467 Online Capstone Project

Date
August 2022

Introduction:

Our team built a 3D Unity-based game similar to Atari's Breakout. We trained a neural net with reinforcement learning algorithms from Unity ML-Agents to play the game. There is a two-player side-by-side mode to let humans compete with the trained neural net. As a team, we had no experience in Unity, C# and Machine Learning. What drew the four of us to this project was the opportunity for us to develop software skills in game development and machine learning which are both rapidly growing industries.

User's Perspective:

To the end user, our Breakout clone brings a refined of a classic game and provides the exciting challenge of competing against a trained machine learning agent. The core gameplay has the user controlling a lateral-moving paddle at the bottom of the screen to bounce the ball in order to break the bricks arranged in a grid at the top of the screen in order to increase their score. There are multiple levels of increasing difficulty to keep the player challenged while facing off against the trained neural network.

Compared With Project Plan:

Our team developed the game and trained the agent by sticking closely to our initial project plan. Looking back at our plan and the work we completed, there was almost no deviation from what the team agreed upon in the first week. We believe that our following of the plan helped us stay ahead of schedule and reduced the amount of potential issues between the different branches of our game. It led to an increase in teamwork, as all team members handled their responsibilities well and were confident in the other team members' progress through the project. The exact distribution of work can be seen in the Team Member Accomplishment section below.

Tools and Resources for Software:

- Unity Game Development Engine
- Unity ML agent package
- C# for scripting
- TensorBoard of TensorFlow

Team Member Accomplishment:

Jared Connor

- Set up machine learning agent training environment
- Designed agent observation, reward system for agent training
- Implemented parallel training strategy
- Managed version control system (git)

Tingting Fang

- Built one-player game
- Debugged for two-player game

- Set up machine learning agent training environment
- Designed agent observation, reward system for agent training
- Integrated the agent into two-player mode

I Chun Hurng

- Set up machine learning agent training environment
- Designed agent observation, reward system for agent training
- Implemented parallel training strategy
- Organized project demonstration video

Jacob Silverberg

- Built one-player game
- Added features to one-player game (score, levels, 3D effects etc.)
- Implemented two-player game feature
- Tested machine learning
- Designed project poster

Conclusion:

Our team is thrilled with our recreated version of Atari's Breakout. Utilizing Unity's game engine and C# scripts, we have developed a game which allows a human competitor to play against a trained Unity ML Agent opponent. Our team trained and tested this agent to ensure a challenging and fun experience for the human player. Some future additions to this game will be to dig into different reinforcement learning algorithms such as PPO (Proximal Policy Optimization) which is used by Unity ML agent. The game can continue to be improved upon with a mode to play against ML Agents which have been trained for different lengths of time and a leaderboard to show who has performed the best against our trained neural network.

References:

1. U. Technologies, "Machine learning agents," Unity. [Online]. Available: <https://unity.com/products/machine-learning-agents>. [Accessed: 03-Aug-2022].
2. Unity-Technologies, "Unity-Technologies/ML-Agents: The Unity Machine Learning Agents Toolkit (ML-agents) is an open-source project that enables games and simulations to serve as environments for training intelligent agents using deep reinforcement learning and imitation learning.," GitHub. [Online]. Available: <https://github.com/Unity-Technologies/ml-agents>. [Accessed: 03-Aug-2022].
3. T. Simonini, "An introduction to unity ML-agents," Medium, 25-Mar-2020. [Online]. Available: <https://towardsdatascience.com/an-introduction-to-unity-ml-agents-6238452fcf4c>. [Accessed: 03-Aug-2022].