

# Atari Breakout Reinforcement Learning Environment

**Haider Sajjad**  
1004076251

HAIDER.SAJJAD@MAIL.UTORONTO.CA

**Weiyu Li**  
1003765981

WEIYU.LI@MAIL.UTORONTO.CA

## Abstract

*Atari Breakout environment implementation and training an agent using multiple algorithms over a generated environment (changing brick layouts)*

## 1. Problem Statement

We intend to implement an atari breakout game environment and train a reinforcement learning agent to play the game. The environment will be built over-top a pygame class of the game and we add the necessary environment variables and functions. We will implement the agent to train with different algorithms taught in class and beyond (Q-learning, SARSA, TD(0)/TD(n)-learning, DQN) and compare performance. We will also aim to train the agent generally, to play atari breakout across different levels (by changing brick layout) and achieve a high score, or win the game. We will also compare the performance of the agent over multiple levels by training it over different levels (by generating different brick layouts in the environment), compared to training the agent on a single environment.

## 2. Motivation and Impact

We want to compare the strengths and weaknesses of different RL algorithms in our case of atari breakout, and analyze how well our agent generalizes across different environments (levels) of atari breakout. Our results may motivate strategies on more complex games by projecting our strategies on to them. This may result in more complex and better performing ai components in video games in general.

## 3. Intuition

How we will do this is by implementing a RL environment structure (done, step, rewards, actions, reset) by implementing its variables and functions into an already built breakout game class made using pygame. A game state in our environment will have 7 components: paddle x location, ball (x,y) location, ball (x,y) speed, bricks left, current score.

The general reward mechanisms will be: +10 if ball hits paddle, -10 if ball misses paddle, +20 when ball hits brick, +100 when all bricks are destroyed. We will experiment with these to get best performance.

There are 3 actions the agent can take: stay still, move left, move right.

We will implement all the algorithms we learned in class (that are applicable) and compare there performance. We will attempt to train over both generated environments (changing

YOUR NAMES

brick layouts) and a stationary one, and see how this affects final agent performance. We will also test our trained agent across multiple levels of breakout to analyze how well our agent learned tasks generally.