iTower

Final

Outline

Game/Project Introduction

Game Modification

Communication Design

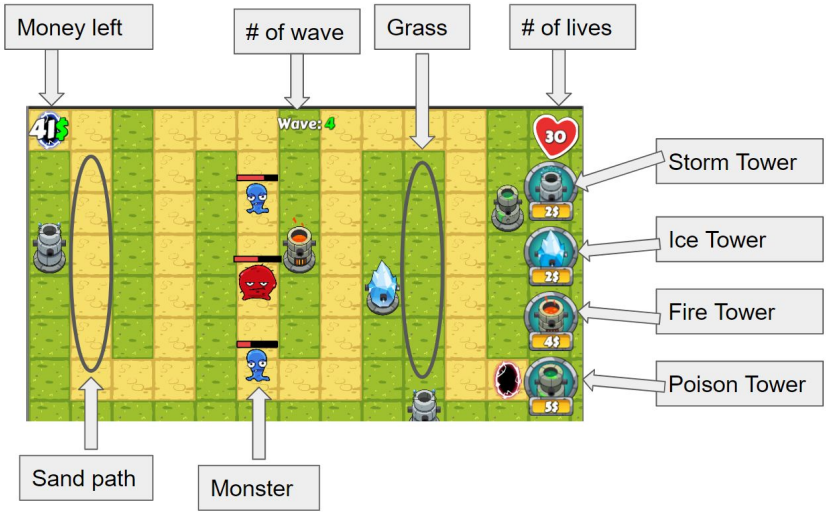
Data Collection/Processing

ML Application

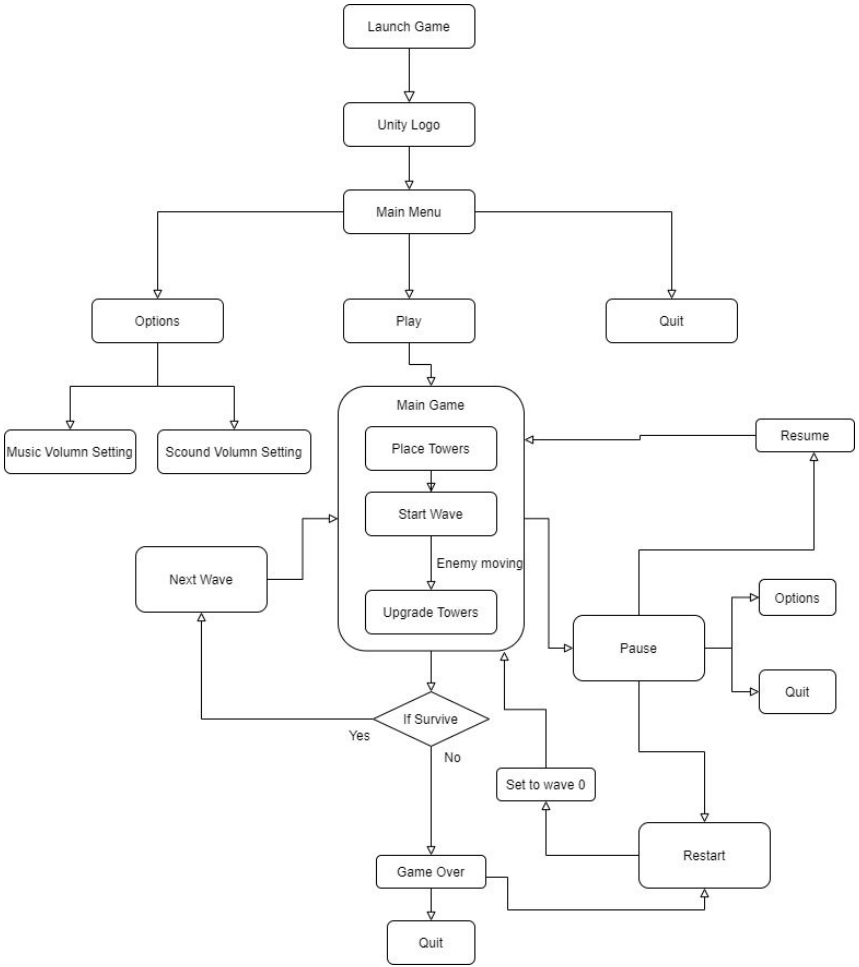
Performance Evaluation

Conclusion

Game Introduction



Game workflow



Project Objective

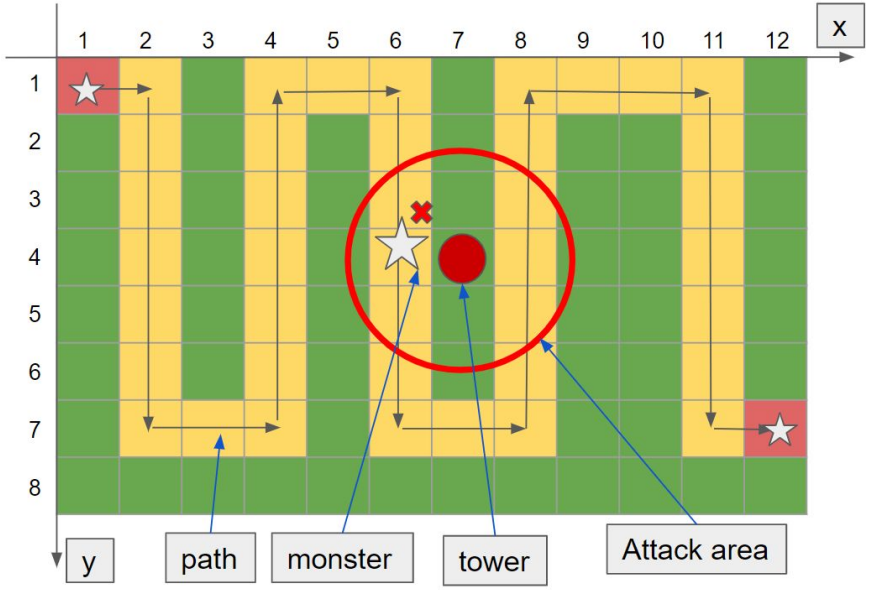
Create an agent that could learn to play the game wisely (higher current money and life at the same time)

Factors to be considered:

1. Current money: numbers
2. Current monsters: numbers and types
3. Towers: costs and types

Expected agent behaviors:

1. Select the proper types of towers
2. Place towers at proper time and location
3. Upgrade properties of the towers.



Prior Research

- Game Research:

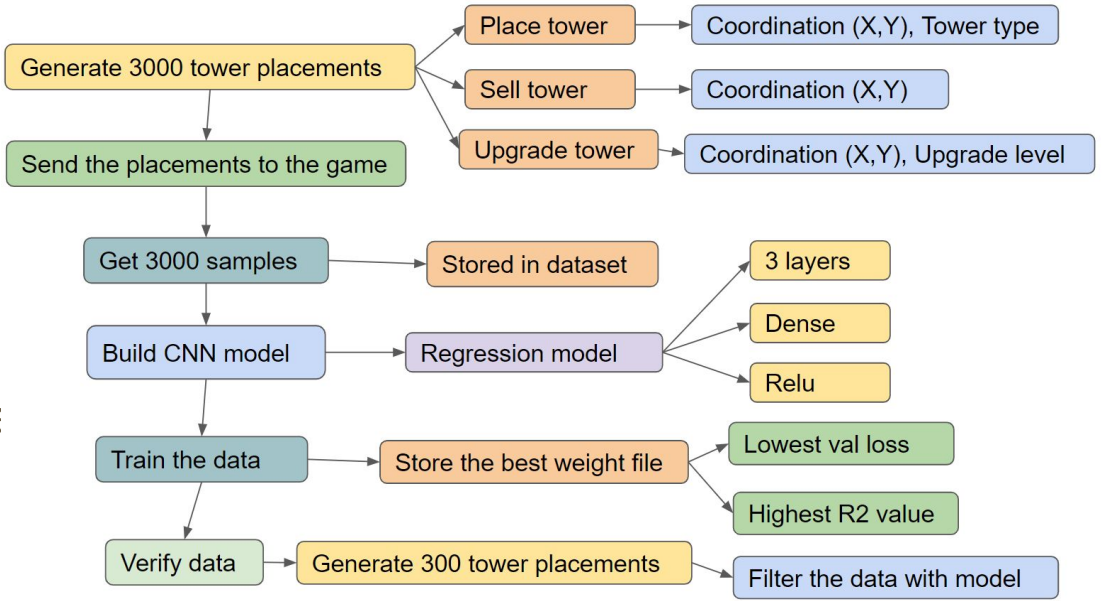
  Jesse Huang, 2D-Tower-Defense-Demo https://github.com/JessHua159, 2017

- Machine Learning Mechanism:

1. CNN classification model
2. NN regression model

Project Details

1. Game Modification
2. Communication Design
3. Raw Data Collection
4. Data Pre-processing
5. Machine Learning Applica



Game Modification - Dependency Removal

- CNN Classification Model:

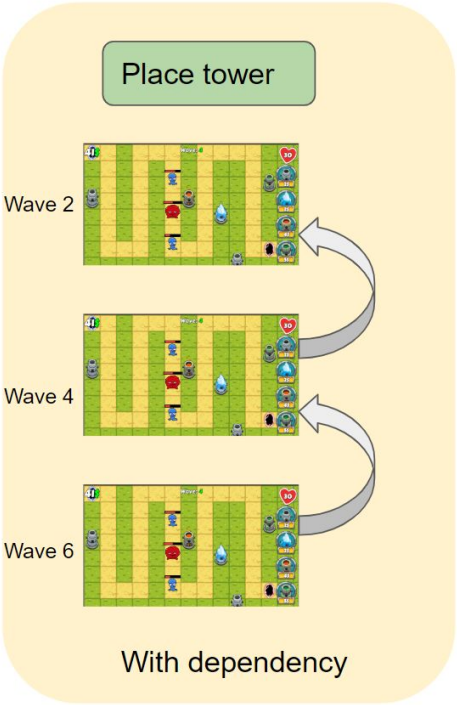
  Doesn't care about dependency

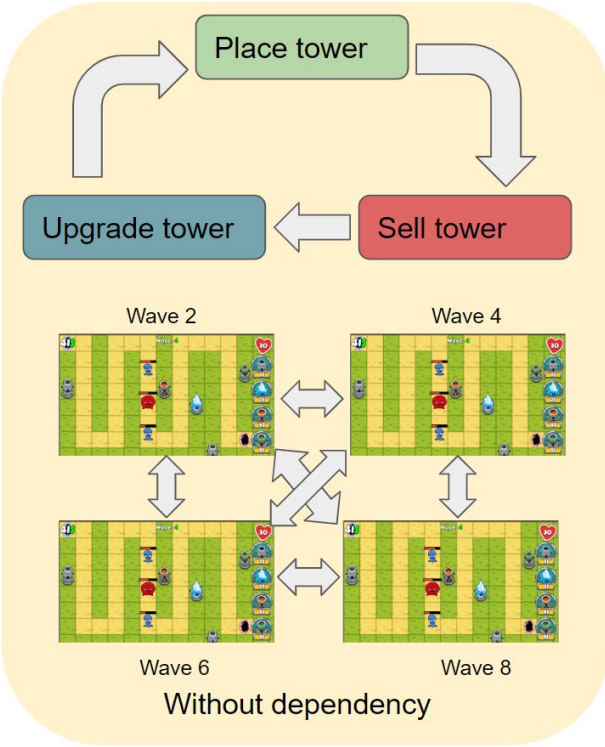
- NN Regression Model:

  Must remove dependency

- Current sequence:

1. Place tower
2. Sell tower
3. Upgrade tower

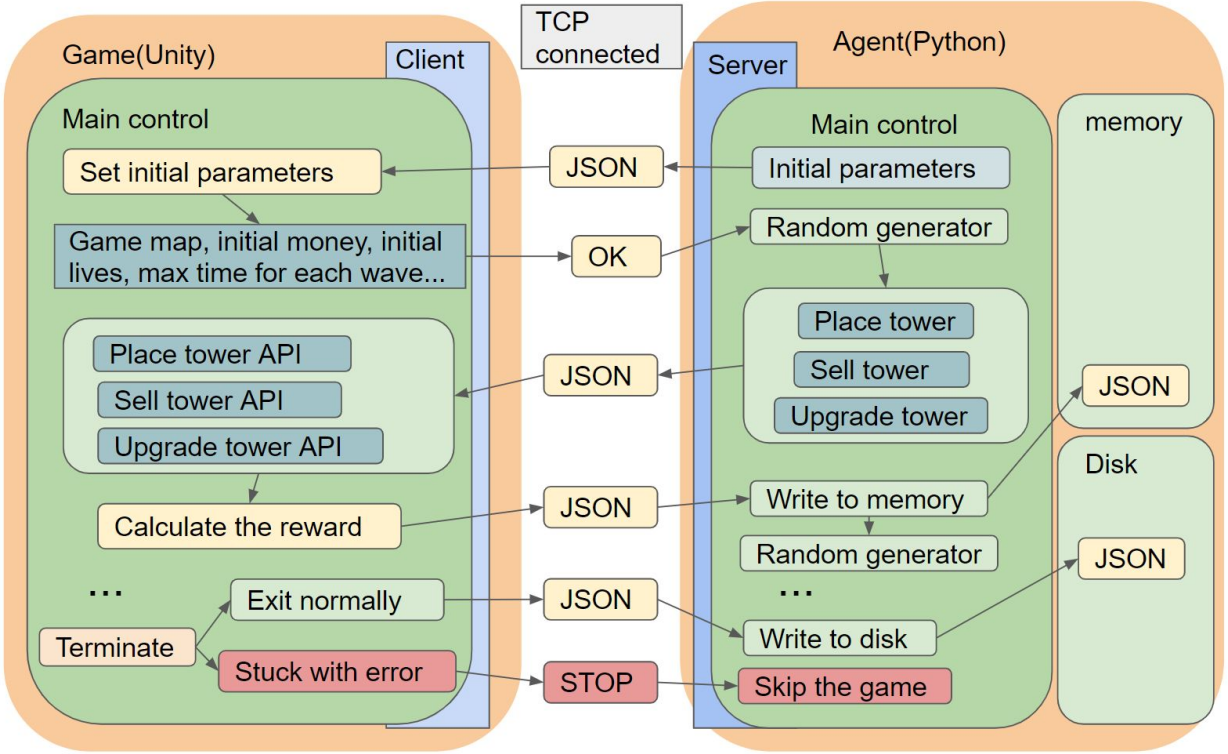




Game Modification - Others

Events	Before	Adjustments	After
Error handling	Monsters stuck at birthplace sometimes	Set a max threshold time for each wave to auto restart	Game and ML process go smoothly
Game termination	Game terminated when no money or lives left. Max survived wave number is 10	Increase initial money and lives dramatically	Max survived wave number is 22

Communication Design

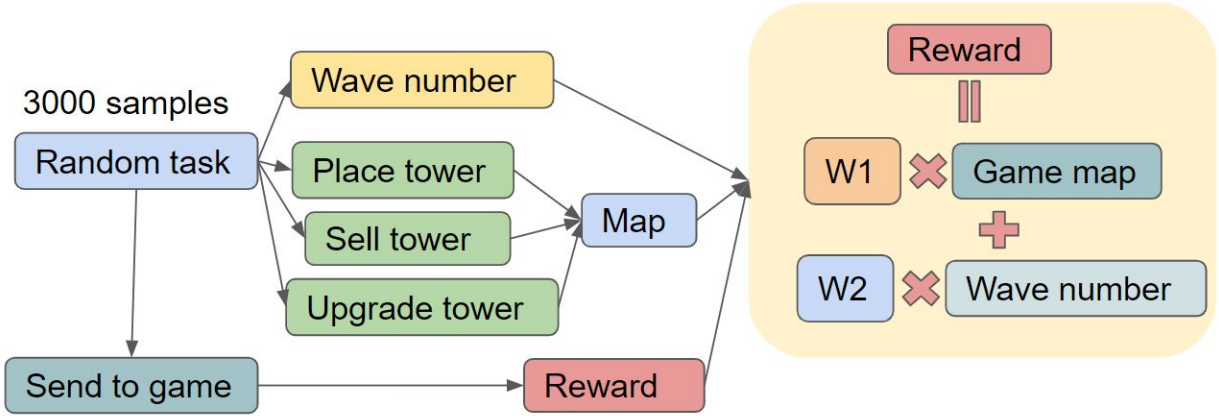


Raw Data Collection

Over 3,000 raw dataset for neural network regression training

Data Contents:

1. Game map: 8*12*4 3D matrix containing tower type and coordinates
2. Wave number to store the state of the game
3. Total life remaining



Data Pre-processing

3,000 groups of carefully selected data based on the following criteria:

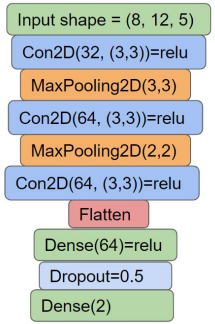
1. Eliminate the incorrect data due to the game bug
2. Normalized the tower placement data using minmax scaler normalization
3. Computed the reward for each wave

Reward = Total life remaining + Monsters' shortest remaining distance to the end

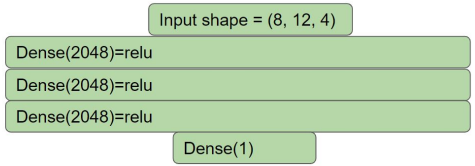
= F(game map, wave number)

ML Application

CNN Classification Model (Before Midterm)	Deep Neural Network Regression Model
• Use # of monsters killed & money earned as reward • Game map as a image • Different tower as different channels • Binary classification • Performance falls after waves	• Use life remain & distance of monsters passed as reward • Game map as a image • Different tower as different channels • Regression to predict the reward • Performance significantly improved



Classify model



Regression model

Reward = Total life remaining + Monsters' min(distance)

= F(game_map, wave_number)

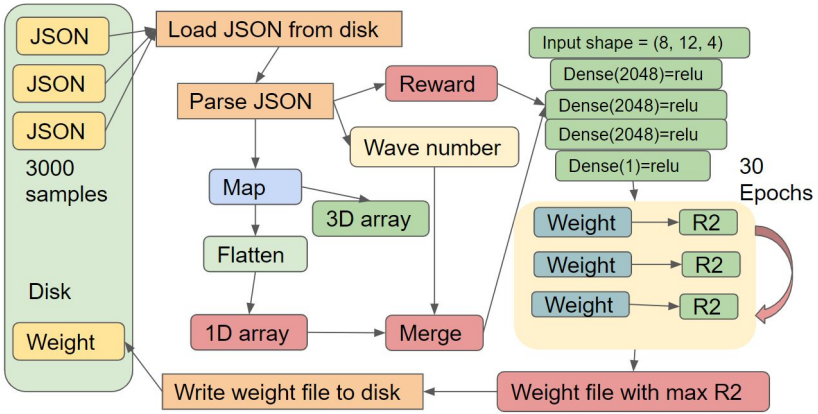
ML Application

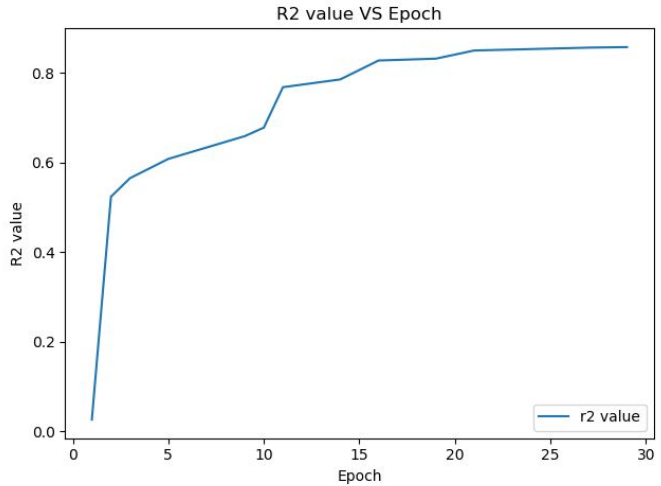
Hyperparameter Tuning for the Deep Neural Network Regression Model

R-squared: It provides a measure of how well observed outcomes are replicated by the model

Hyperparameters:

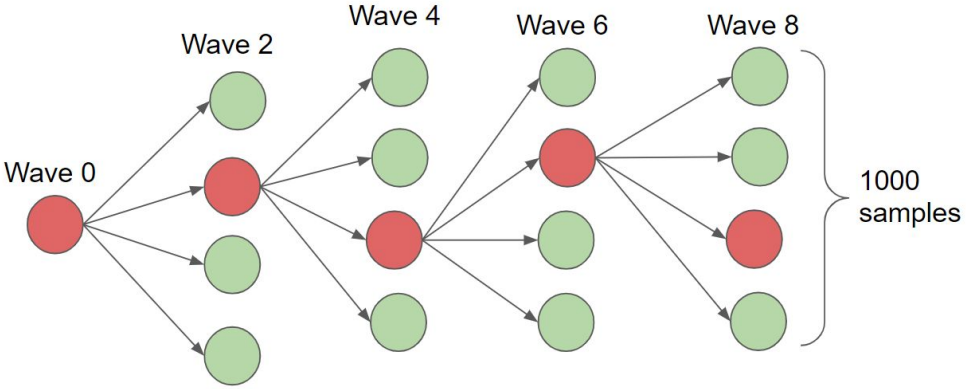
- # of neurons per layer
- # of layers
- Drop out rate
- Activation function
- Optimizer
- # of epoch

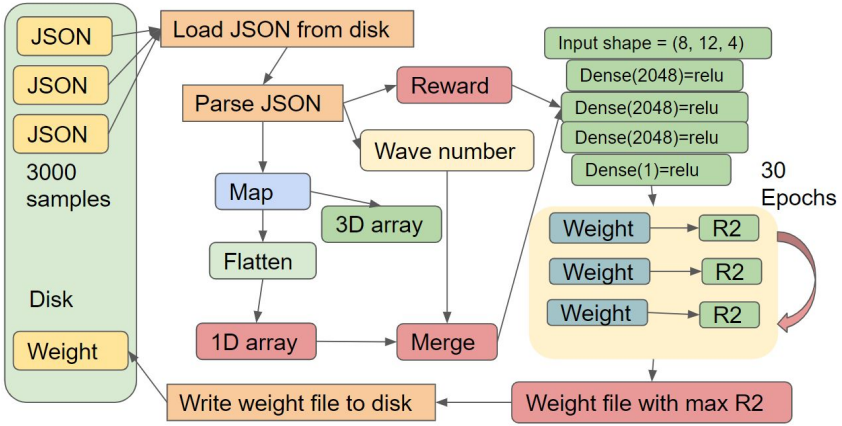


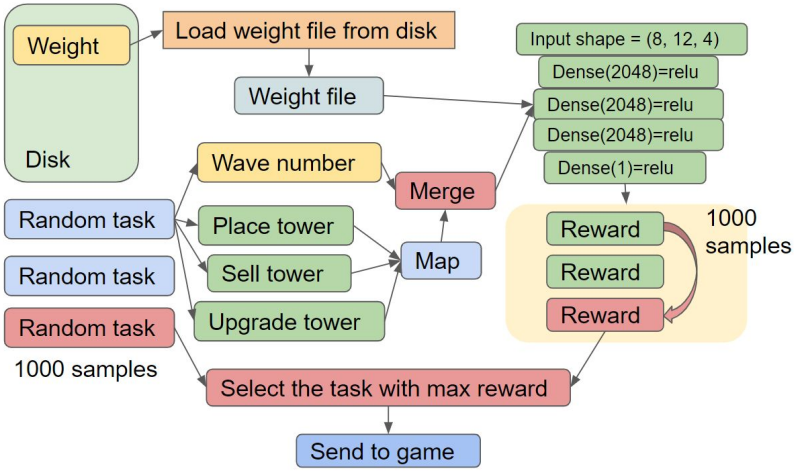


ML Application

Game Strategy Generation

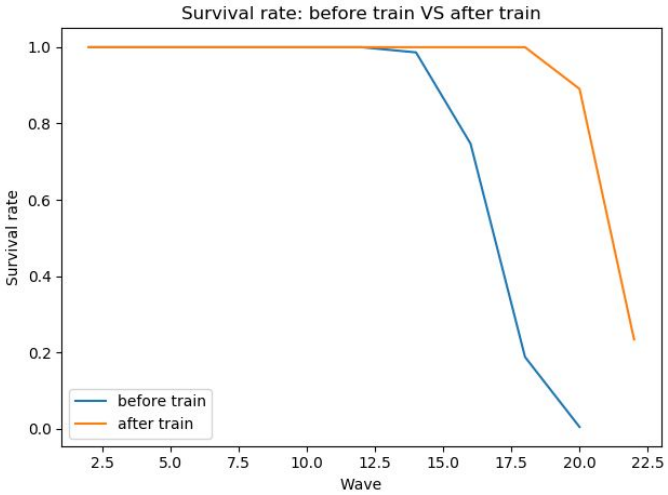


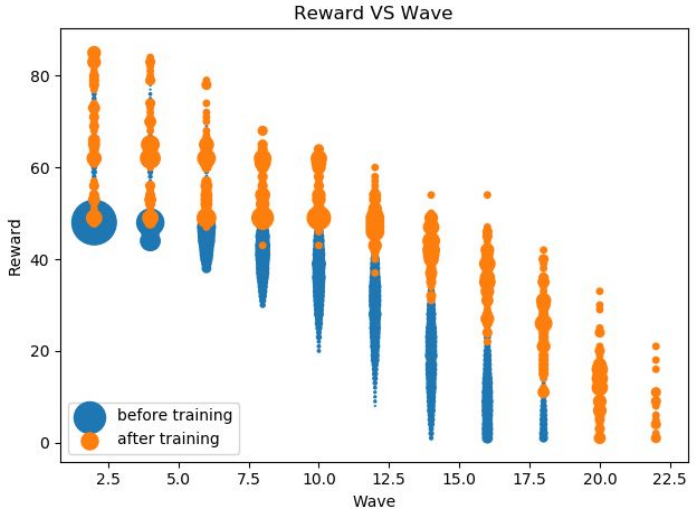




Performance Evaluation and Conclusion

	Maximum survivable wave
CNN Model	20
Deep Neural Network Regression Model	22





Limitations & Future Work

- Fix the bug caused the game to freeze.
- Collect more training data after the bug has been fixed.
- Explore to improve the model performance using newly collected bug-free training data.
- Extend our model to more maps.

Demo Video

Thanks