



RUTGERS THE STATE UNIVERSITY OF NEW JERSEY - 12/18/2023

# RL SPACE INVADERS

ECE - 472 Final Project

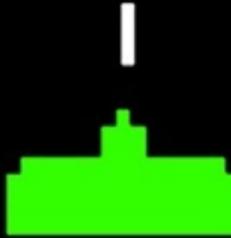
---

**Georgiy Aleksanyan**

ECE - 472

**Professor: Dr. Kristin Dana (ECE)**

SCORE 1,337



[Back to Overview](#)

# Introduction

- Space Invaders is an extension of the atari game suite. The idea is too build a custom game and train an rl agent to win the game.
- Goal: The trained agent performs better than the random agent

# Can AI Play Space Invaders?

Kind Of - Let's Dig Deeper

[Back to Overview](#)

## Objectives

- Create The Basic Game
- Test The Game
- Create A Custom OpenAI – GYM Environment
- Test that the Environment is Functional
- Create a Training Script
- Train a Model
- Compare Trained Model Wit a Random Agent

# Hypothesis

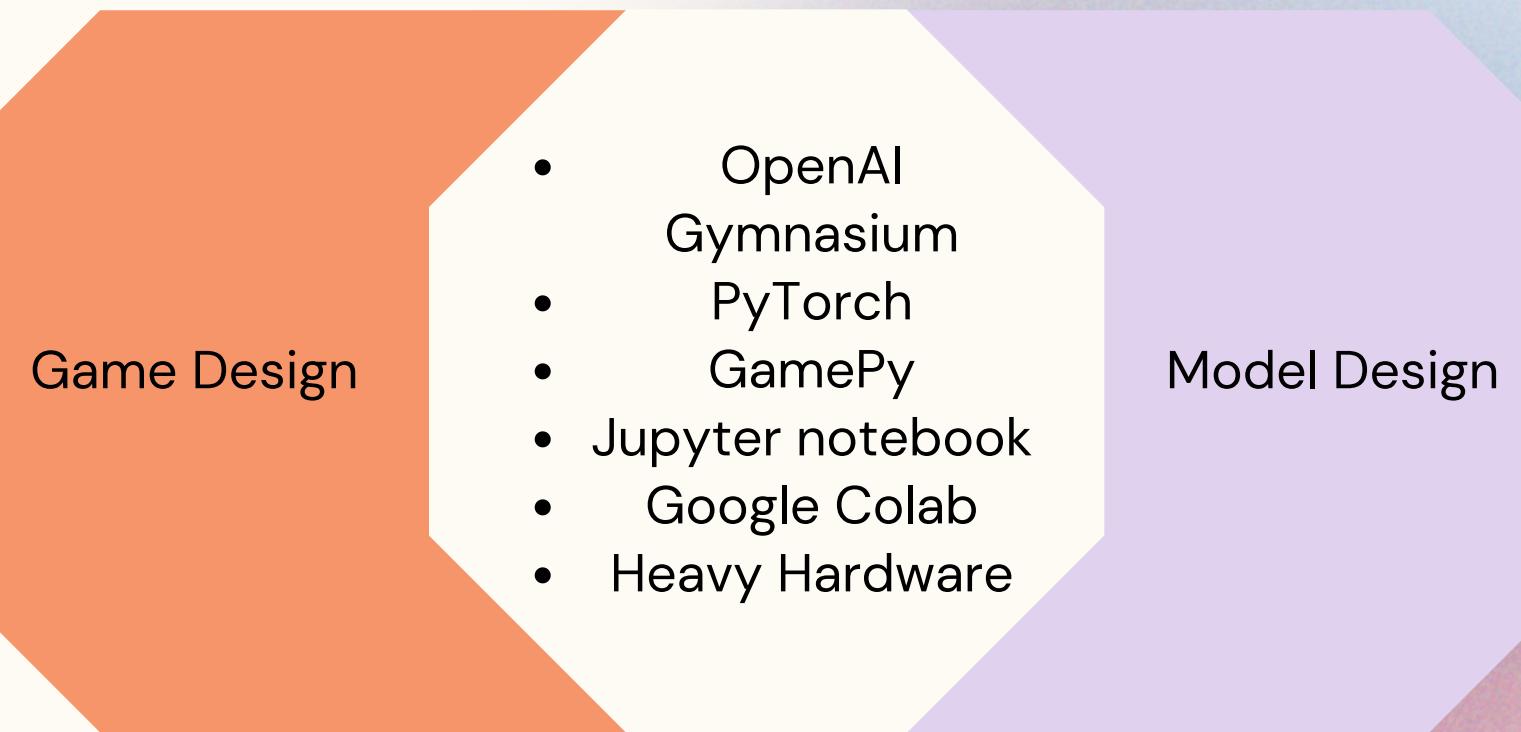
---

[Back to Overview](#)

**TRAINED AI > RANDOM**

# Framework

[Back to Overview](#)



```
(space_defenders_rl) C:\Users\georg>cd "C:\Users\georg\space_defenders_rl\spaceInvadersEnvV7"  
(space_defenders_rl) C:\Users\georg\space_defenders_rl\spaceInvadersEnvV7>python exampleUsage.py  
Episode: 1, Total Reward: 40.8949999999957  
Episode: 2, Total Reward: 59.8969999999951  
Episode: 3, Total Reward: 11.88400000000034  
Episode: 4, Total Reward: 68.356000000003  
Episode: 5, Total Reward: 57.3670000000024  
Episode: 6, Total Reward: 78.880000000011  
Episode: 7, Total Reward: 46.3999999999764  
Episode: 8, Total Reward: 61.380000000004  
Episode: 9, Total Reward: 57.383000000008  
Episode: 10, Total Reward: 101.878000000061  
Episode: 11, Total Reward: 26.89199999999697  
Episode: 12, Total Reward: 65.365000000004  
Episode: 13, Total Reward: 11.9640000000196  
Episode: 14, Total Reward: 11.8840000000219  
Episode: 15, Total Reward: 119.946000000009  
Episode: 16, Total Reward: 71.4500000000115  
Episode: 17, Total Reward: 49.4469999999974  
Episode: 18, Total Reward: 73.3690000000043
```

- Adjusting Parameters in the environment such as:
  - game speed
  - observation space
  - action space
  - reward
  - rendering
- Adjusting Parameters in the training script such as:
  - BATCH\_SIZE
  - GAMMA
  - EPS\_START
  - EPS\_END
  - EPS\_DECAY
  - TAU
  - LR
  - ReplayMemory()
  - DQN Layers

# Methodology

[Back to Overview](#)

## Design and Methods

- Optimise Reward To Make Sure That The Agent Isn't Overly Constrained

## Fine Tuning

- Conduct Multiple Trainings With Varied Parameters To Dile In The Model

## USE DQN

- Using Images to process states and gather observations for DQN in the Training

## Data Gathering

- Torch Tensors that Store the Trained Model Weights

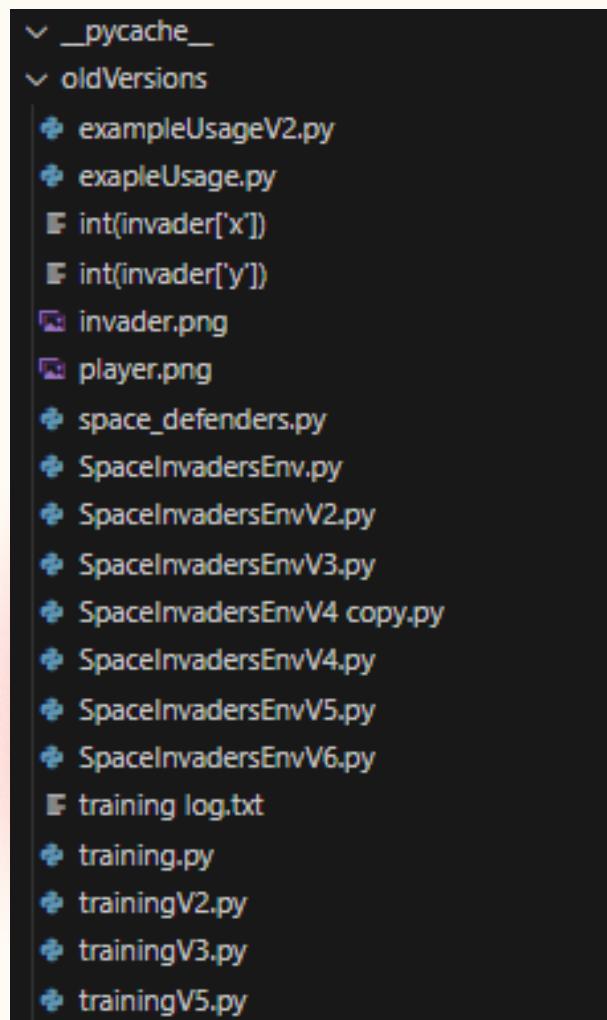
## Optimization

- Greedy Approach, Maximize Reward, Markov Decision Model, Bellman Equations

## Data Analysis

- Mean Values, Episode VS Duration and Reward Graphs, Trained VS Untrained

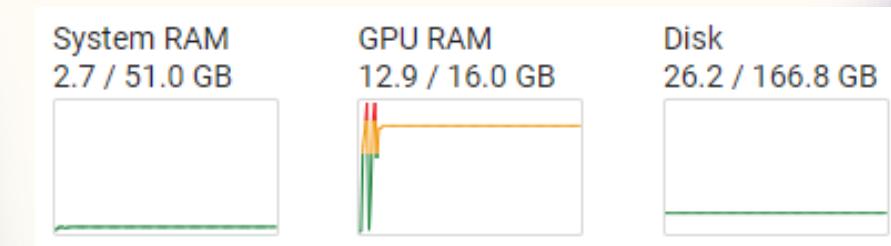
# Limitations



[Back to Overview](#)

## HARDWARE

- Training Especially with Rendering Required A Lot Of Resources



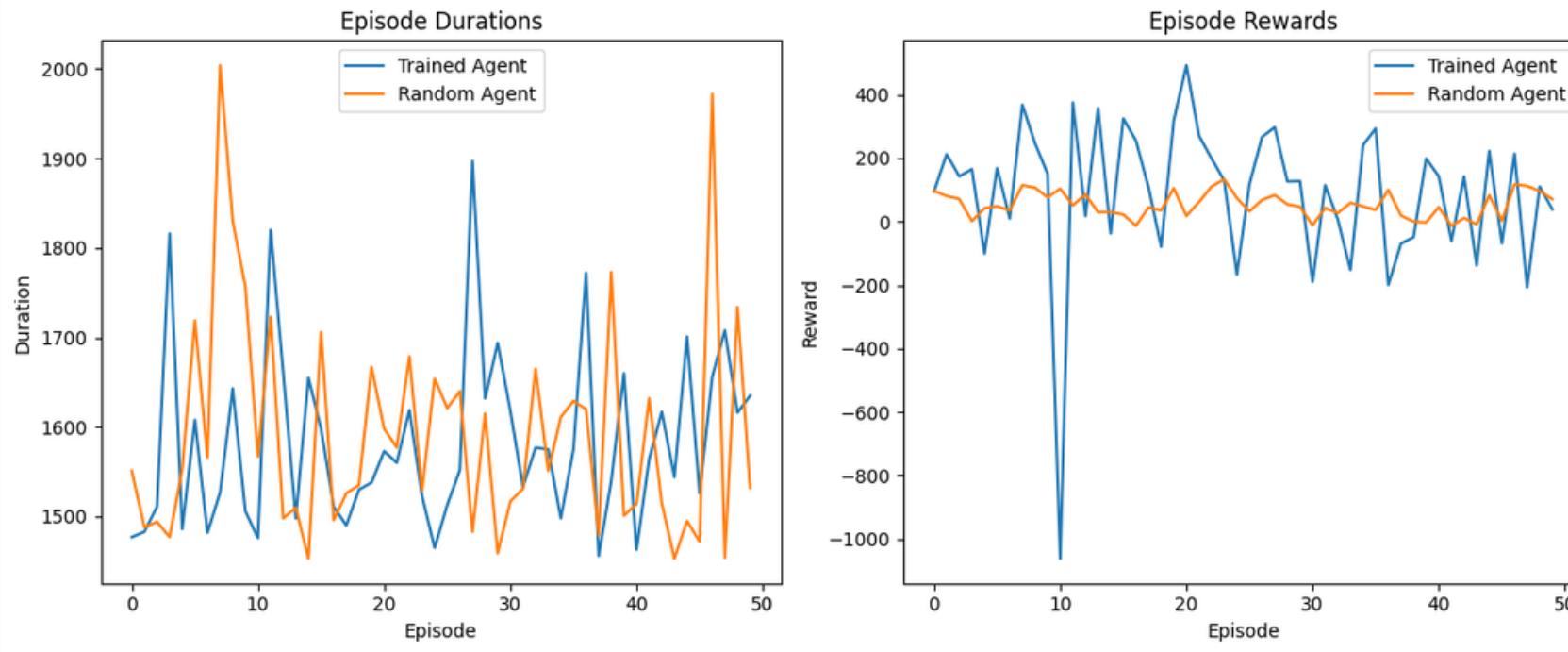
## KNOWLEDGE – Bit Off More Than I Could Chew

- I had a hard time adjusting convolutional layers to fit the inputs and outputs. The DQN tutorial from PyTorch That we covered in class Was Helpful Wrote Many (MANY) Versions of Code

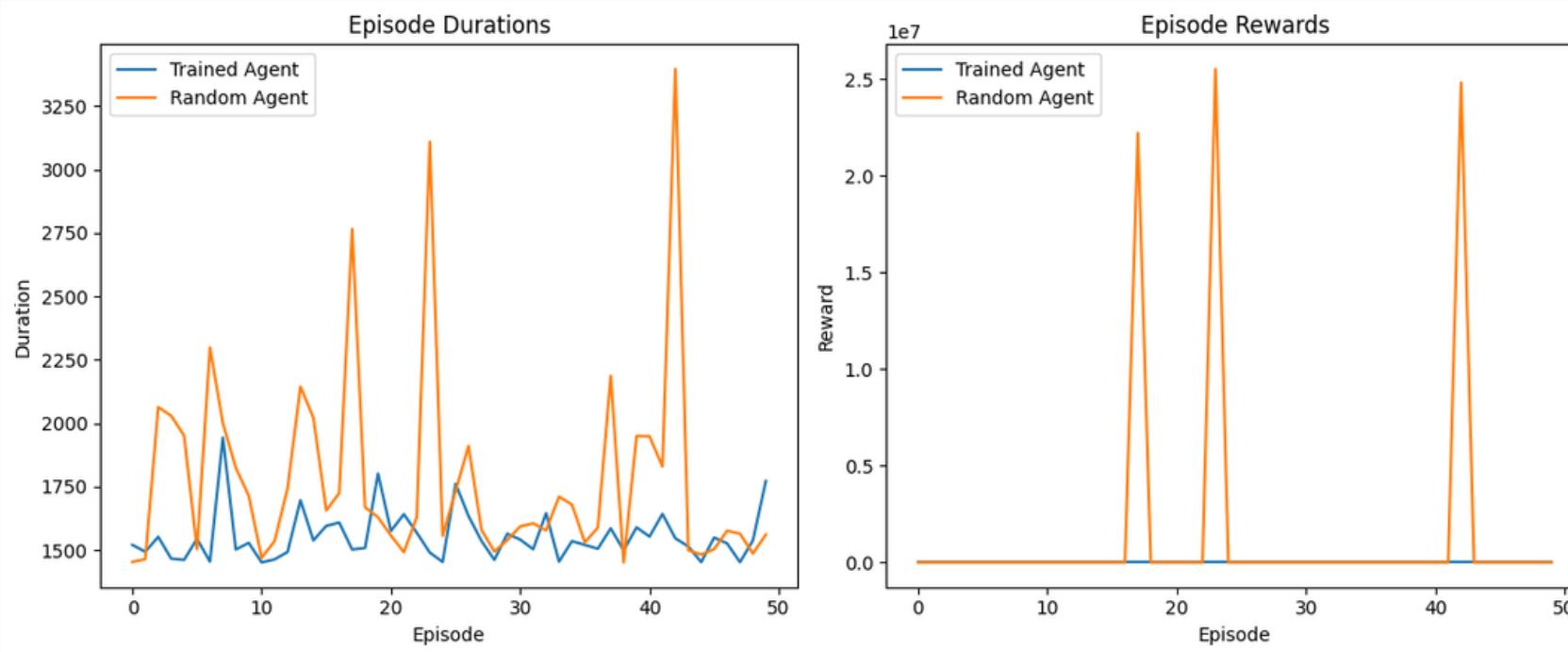
```
RuntimeError: mat1 and mat2 shapes cannot be multiplied (900x200 and 300x128)
```

## Suboptimal Observation Space/Reward Structure/Model Fitment

## Training For 50 Episodes/No Soft Update



## Training For 600 Episodes/ + Soft Update



# Results

## AI Can Indeed Play Space Invaders

- Results Show Higher Max reward in trained model performance than in a random agent for 50 episodes of training.
- Training for 600 episodes, increasing model parameters Like Batch Size, and Replay Memory, and Changing Reward Structure cause the model to be overly constrained.
- ALL Code Is Available On [Github](#)
- Hypothesis Disproven

# Summary and Conclusion

[Back to Overview](#)



## Fine Tuning Proved To be a length (VERY LENGTHY) Process

Too many factors to consider when fine tuning parameters to train a solid model

## A Lot Of Uncertainty Still Present In The Model

Having Images Processed via DQN Rather than specific game parameters (locations of player etc.)

## More Layers == More Problems

Using more convolutional layers caused mismatched tensor sizes and in some cases memory issues

# Implications and Recommendations

---

[Back to Overview](#)

## Use A Different Model

Using two or three other models to compare results would have been essential to building a successful model.

## Training Is Not Simple

Space Invaders though did not seem like so in the beginning is a complex environment and training a model that can play a game with a ton of uncertainty is an expensive task.

7 hours turned into 50

# References

---

[Back to Overview](#)

- [Nicholas Renotte](#)
- [sentdex](#)
- [PyTorch](#)
- [Geeks For Geeks](#)
- [OpenAI Gym](#)