CS698R: Deep Reinforcement Learning Project

FORAGING IN THE FIELD

Mentors: Adithya, Ajita

Group-5:

Tabish Ahmad - 21111060

Prakhar Srivastava - 170486

Areeb Ahmad -180135

Mohd Niyas P -170394

Samarth Mehrotra -20128408

PROBLEM DESCRIPTION:

- There is a field with certain number of patches, and each patch has certain number of berries of various sizes.
- Reward is proportional to the size of the berry.
- There is a pre-defined variable draining rate proportional to distance travelled which depletes the total reward accumulated.
- Continuous movement or having the option to stay?
- Field constant or changing ?

AIM:

- Maximize the reward remaining at the end?
- Maximize the total cumulative reward collected during 5 mins?
- Both?

MOTIVATION:

- Although most decision research concerns choice between simultaneously presented options, in many situations options are encountered sequentially.
- The decision is whether to exploit an option or explore for a better one.
- All animals must forage food for survival but doing so is energetically costly.
- But we know little about process involved in decision making.

LITERATURE SURVEY:

- **Information foraging : -** Application of optimal foraging theory to understand human behaviour while searching for information on internet.
- **Marginal Value Theorem :** It describes the behaviour of an optimally foraging individual in an environment where resources (berries) are located in discrete patches separated by areas with no resources.
- **Observation of Animals : -** Species like great tits and screaming hairy armadillos have been studied and their foraging patterns have been tested for optimality.
- **NEAT:** Neural Evolution of Augmented Topologies. A genetic algorithm which evolves the neural networks over generations.

ENVIRONMENT DETAILS: (RECAP)

Overall **• Field - 20,000 x 20,000 pixels** **• Screen - 1920 x 1080** **• Time - 5 minutes**	**Patch** **• Size - 2600 x 2600** **• Number - 10** **• Patches completely inside field** **• Interpatch distance - >=5000** **• Berry sizes - 10, 20, 30, 40** **• Berry number - 20 each/patch**
Reward • Initial = 0.5 (out of 1) • Drain rate = 0.5d/(120*400) (d distance in pixels) • Berry reward = diameter/10,000	Player • Size = 10 • Speed - 400 pixels/seconds • Arrow key movement (8 directions)

ASSUMPTIONS

- ✓ Agent knows that there are four types of berries(by size).
- ✓ Agent's vision is restricted to visible screen.
- ✓ The berries don't replenish after they are consumed.
- ✓ Patches has same densities.

IMPLEMENTATION:

- Environment implemented with OpenAi gym integrated with PyGame for rendering.
- Environment has the option human and computer(agent) play.
- Agent is trained by the algorithm Double DQN.
- Agent is provided two options: Continuous Action or Have an additional option to stay.
- Continuous action helps is maximizing the cumulative reward.
- Non-continuous action allows the agent to maximize the reward remaining at the end.

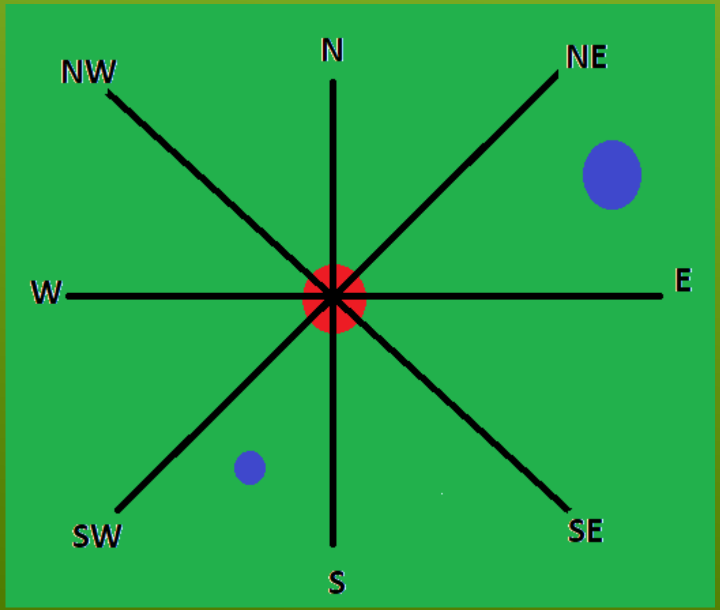
Agents

- Continuous Movement
- Non-Continuous Movement
- Random Environment

AGENT:

- State-Size = 35, Action-Size = 8 or 9 (depending on continuous or non-continuous movement).
- The input vector to our neural network is a 35 length vector. The output is a vector of size 8 or 9.
- Since we can't use CNN because of **memory issues** and **FPS** drops we took the information from the image as features and provided it to the network.
- The features are:

Consider the vector: [N, S, E, W, NE, NW, SE, SW]



- The agent scans in the 8 contained directions as far as it can see or until it sees a berry .
- The vector is filled as: [0, 0, size1/distance, 0,size1/distance, 0, 0, 0]
- Another example: [0, size2/distance,0, 0,0, 0, 0,size2/distance]
- There is a different vector for each size of berry. Therefore we have 8*4 = 32 states.
- **Intuition:** Is it worth travelling the distance for the corresponding size?

- 33rd state is the density. Meaning how much berries are occupying the screen at the current moment.

- 34th state is the current health remaining.

- 35th state is the current time remaining.

- **Intuition:**
  - Density helps in determining if it is a good time to leave the patch.

  - Current health and time helps in determining whether to stay in the already depleted patch instead of exploring.

  - Precisely at any point of time the agent should have the health to explore a new patch and if it doesn't the current patch should be able to provide that much resources.

  - Features are scaled so that any feature don't get too big or too small.

- Noise is added to the states to avoid overfitting.
  [States] + ϵ where $\epsilon \sim N(0, \sigma^2)$.

- We don't need to take a action from the agent's q-network every time.
  $if\ random.random < \epsilon_2$:
  take step from the q-network
  $else$:
  continue previous action

- Updated ϵ-Greedy policy:
  $if\ random.random < \epsilon_3$:
  take a random step and continue it for n seconds.
  $else$:
  take a step from the q-network

- This also helps in reducing the stuttering motion of the agent, make the movement more human-like and helps it getting out of oscillations.

- Noise also helps the agent taking some random actions when it doesn't get any input from the visible screen.

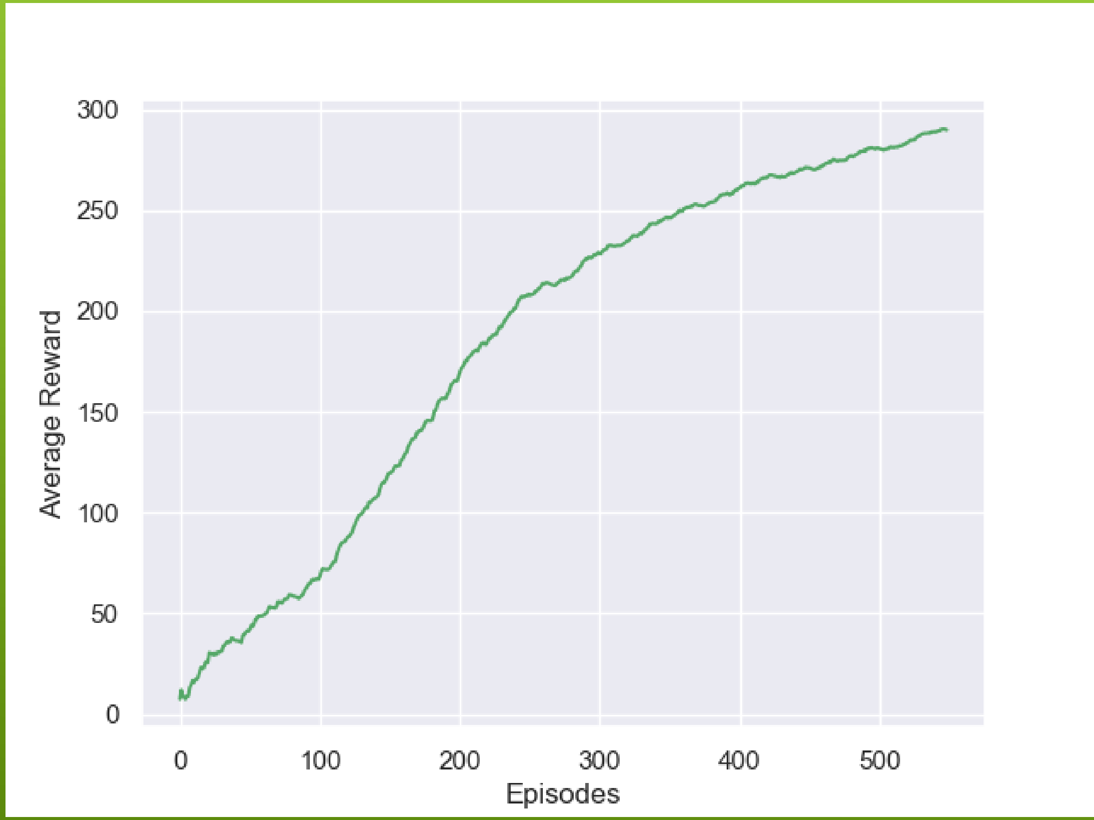
MODEL SUMMARY:

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1            [-1, 1, 35, 64]           2,304
            Linear-2            [-1, 1, 35, 64]           4,160
            Linear-3             [-1, 1, 35, 8]             520
================================================================
Total params: 6,984
Trainable params: 6,984
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.04
Params size (MB): 0.03
Estimated Total Size (MB): 0.07
----------------------------------------------------------------
```

- Size of the model is about 70KB which is very less.
- The model inference time is very fast and doesn't effect or slows down the game.
- This also contributes to agent playing like a human would play.

RESULTS:

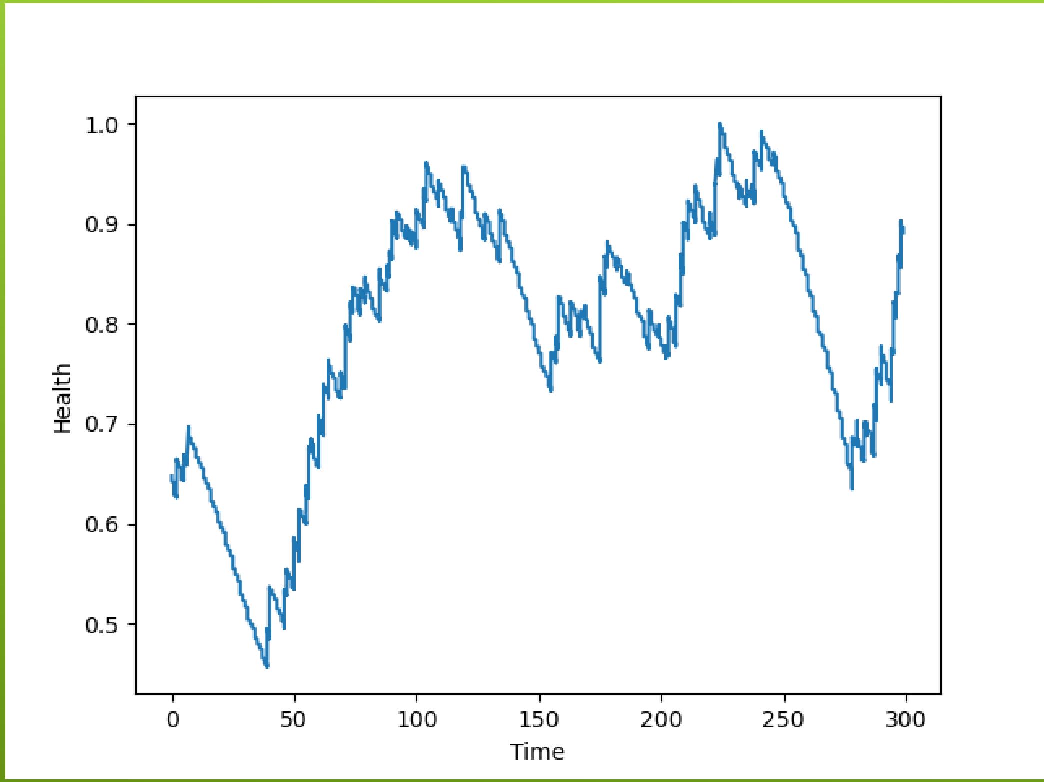
Running Average Reward vs Episodes

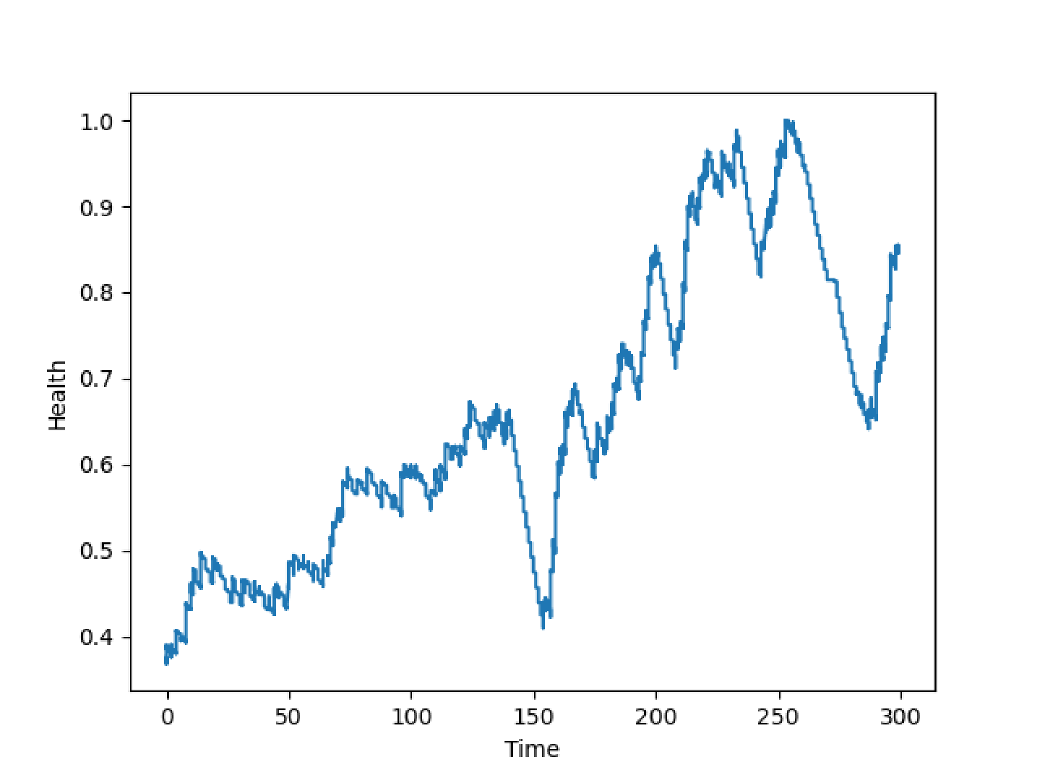


- Time taken for 500 episodes: 26hrs
- The problem's objective starts getting completed after it is able to achieve more than 250 cumulative score.
- The human players average is 344.
- The trained agent average is 416.

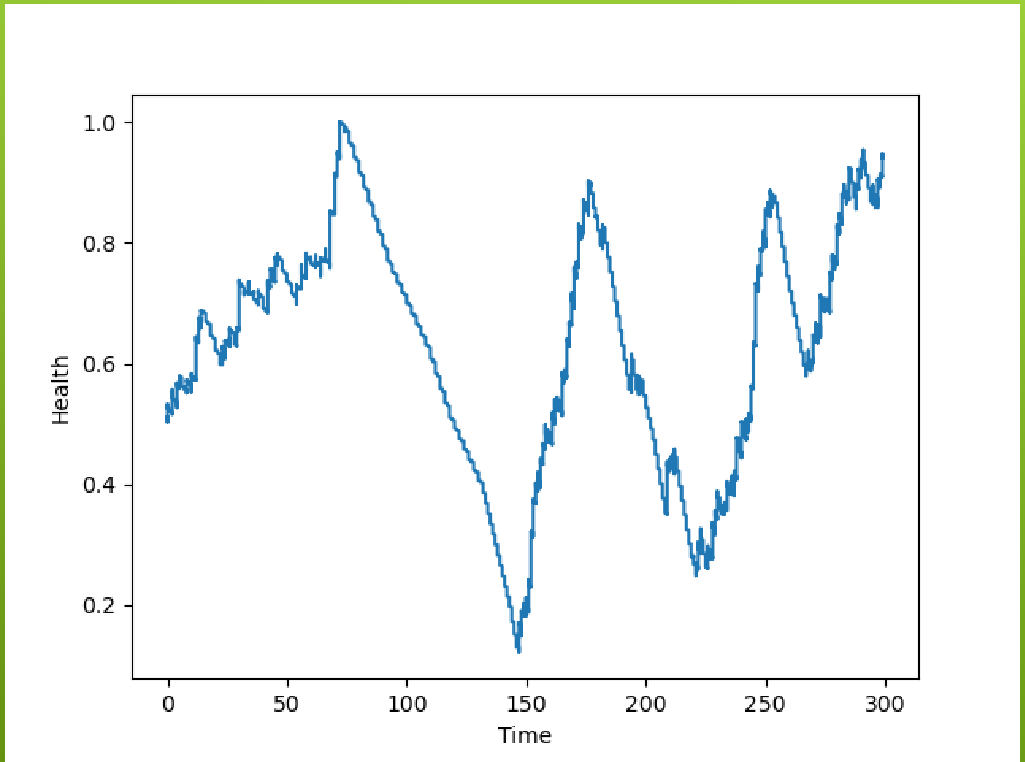
Health vs Time:

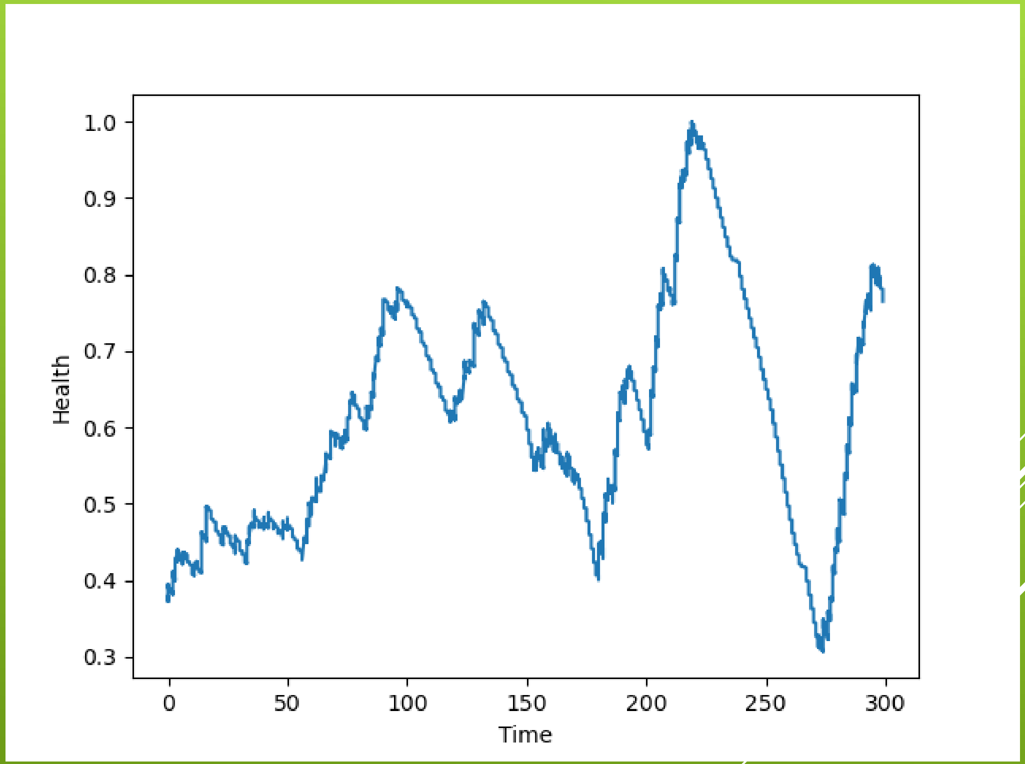
Same Environment



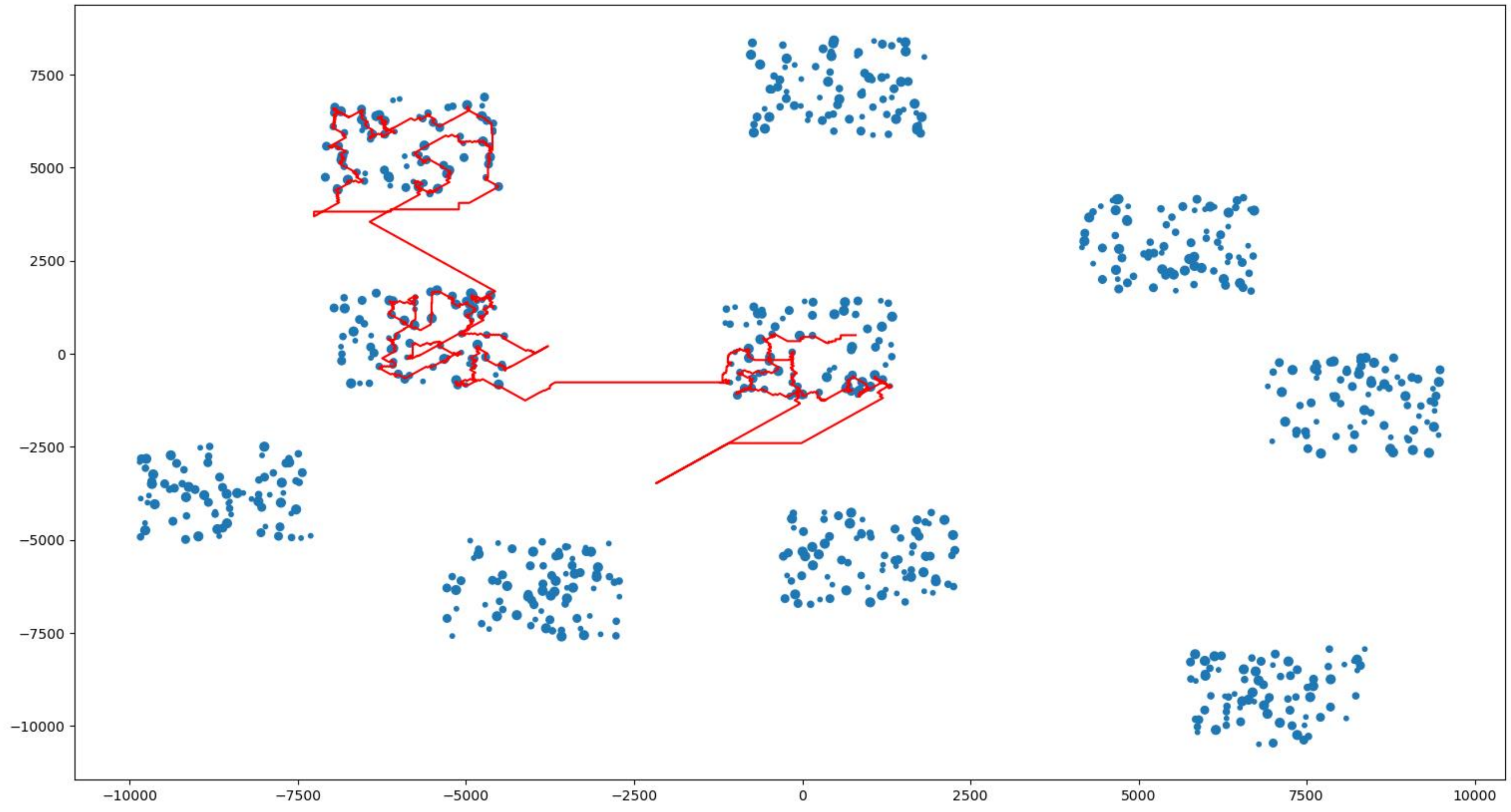


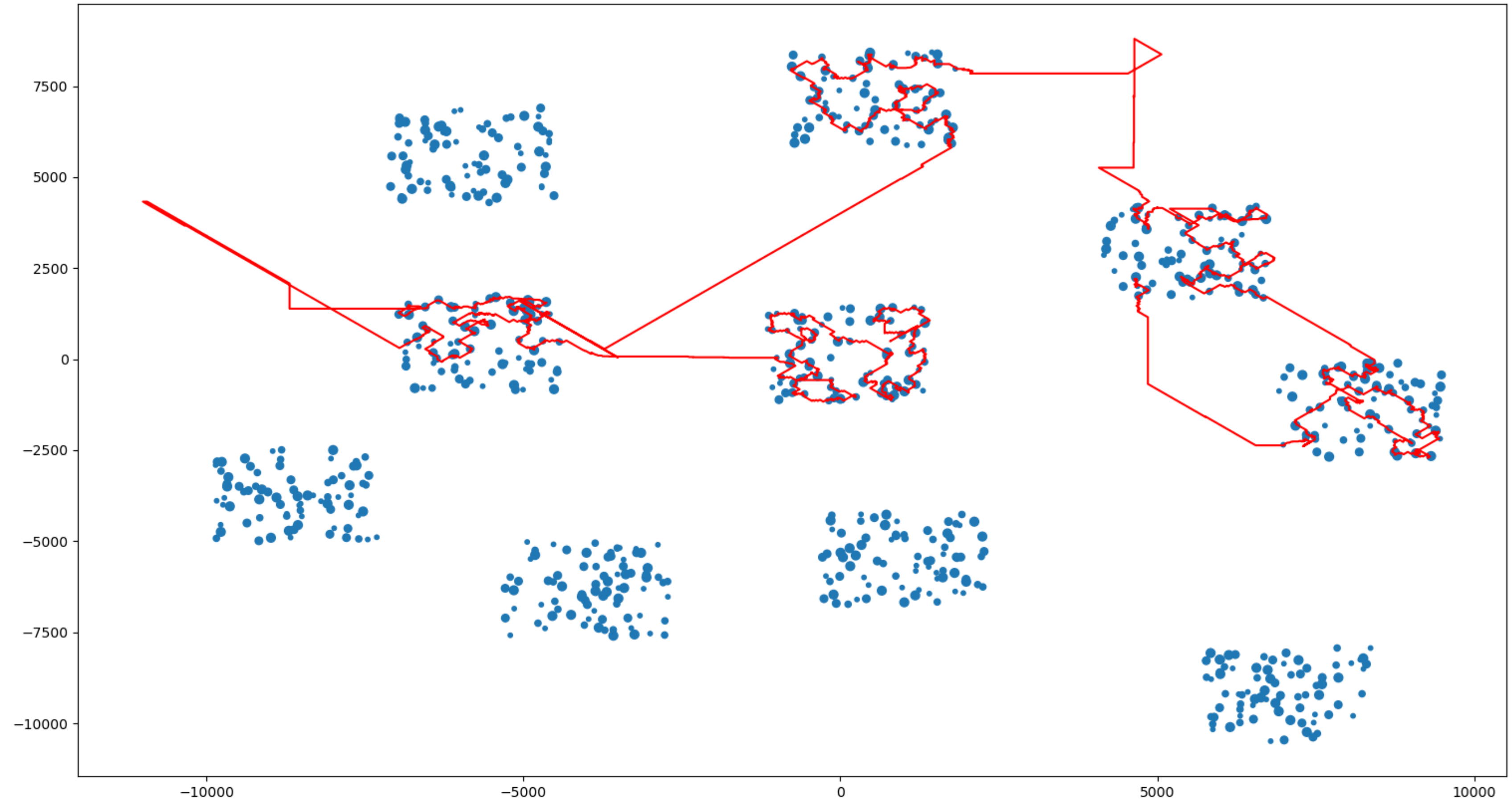
Random Environment



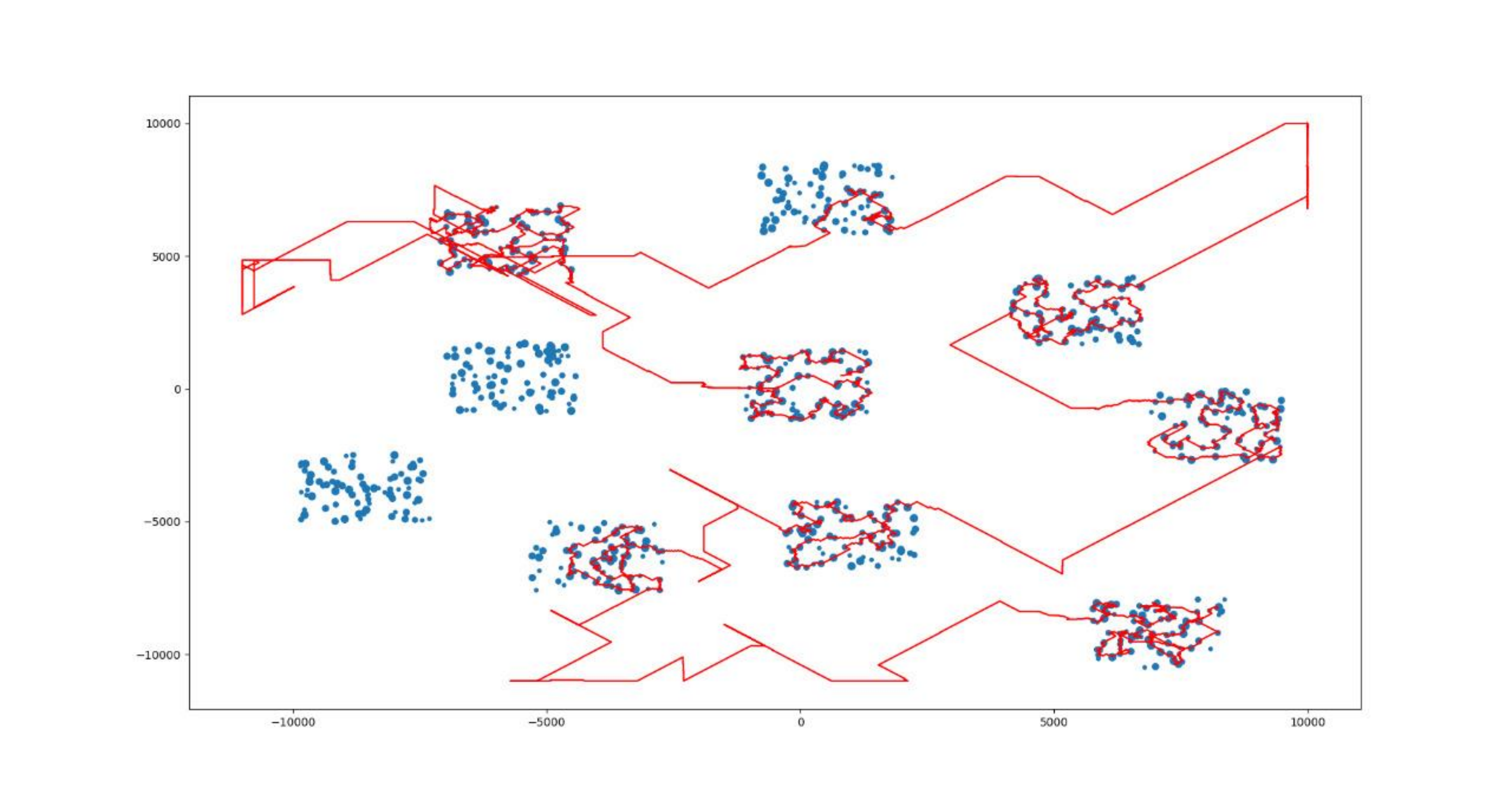


Agent’s Path:

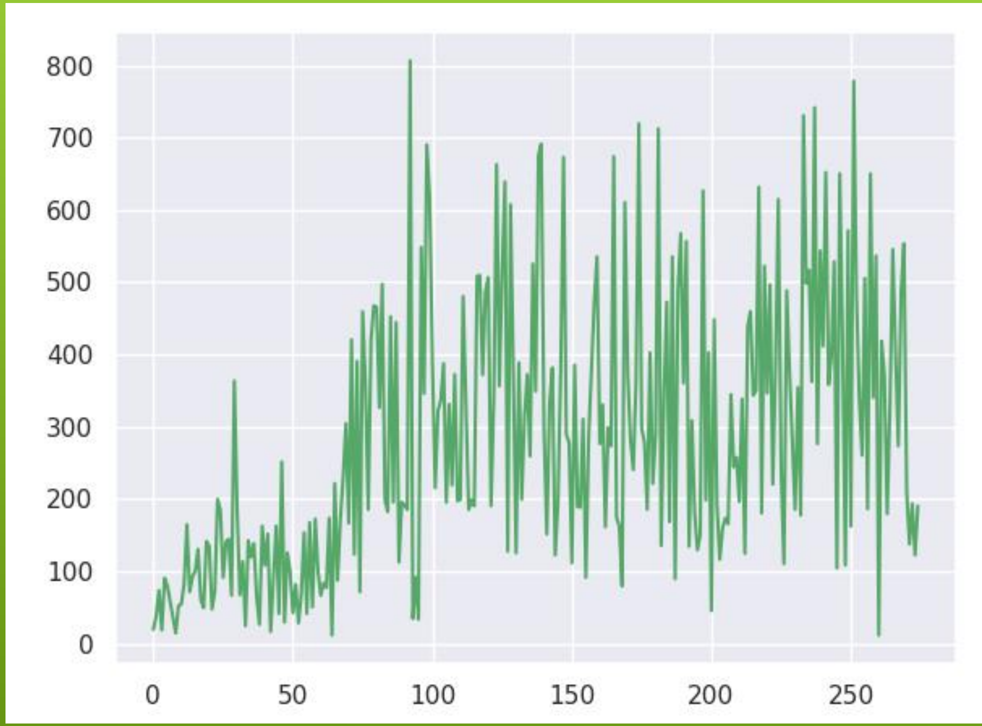

7500
5000
2500
0
−2500
−5000
−7500
−10000
−10000
−7500
−5000
−2500
0
2500
5000
7500
10000


7500
5000
2500
0
−2500
−5000
−7500
−10000
−10000
−5000
0
5000
10000

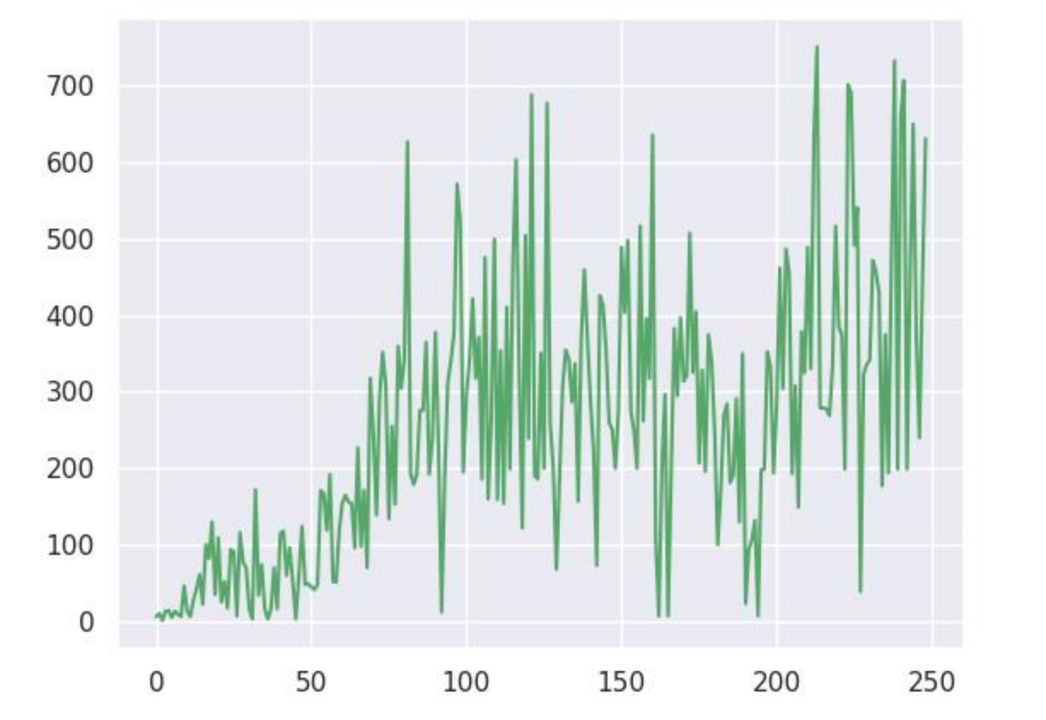
AN OUTLIER !!!!



AGENT'S PERFORMANCE



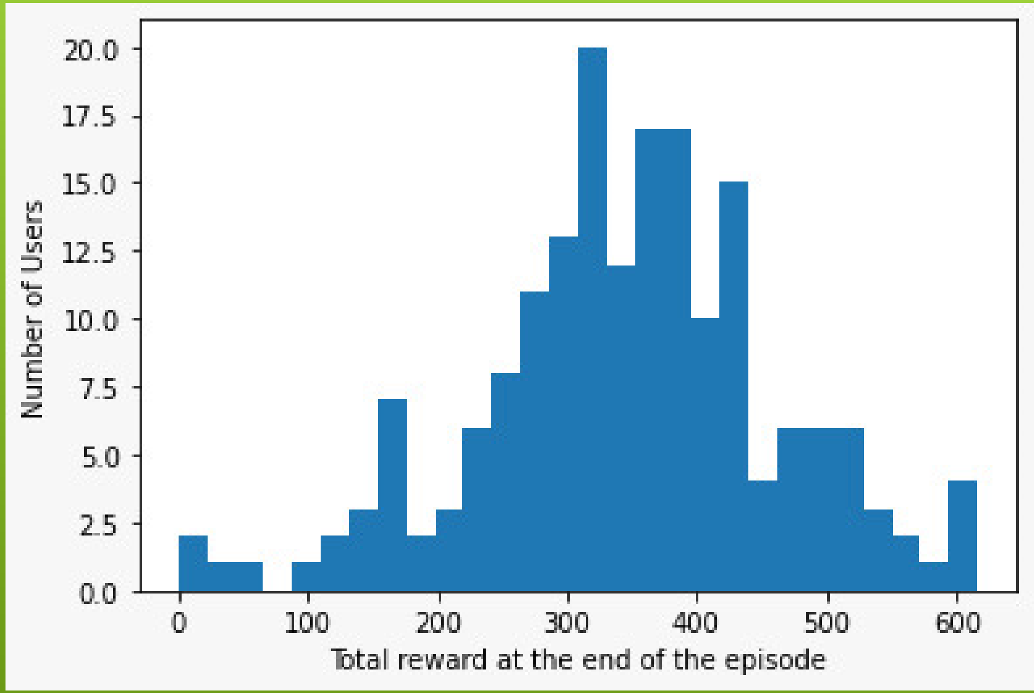
Total rewards vs episodes (Stay option)



Total rewards vs episodes (No-Stay option)

HUMAN PERFORMANCE

The following is the human players performance distribution(data of 183 users.)



CHALLENGES:

- Time to train the agent take about 8-9 hours to reach decent performance.
- As it can be seen, there is lot oscillations ,we believe that it will mitigate as episodes will progress, but it takes lot of time in execution.
- Optimizing the game for smooth performance.
- Resource constraints for using CNN.
- Agents would get stuck into oscillations and the health goes to zero very rapidly.
- Tuning the hyperparameterss of the DDQN.

FUTURE:

- CNN with optimizations of getting the current screen without FPS drop.
  - CNN will help in learning more complicated features.
  - More human-like performance.

- NEAT (Neural Evolution of Augmented Topologies).
  - NEAT is a genetic algorithm: (Simulates the process of natural selection)
    - Individuals with good fitness scores are given preference to pass their genes to successive generations.
    - Cross over of Genes.
    - Mutation in the next generation.
    - Fitness for new population.

FINAL CONTRIBUTION:

NAME	CONTRIBUTION
Tabish Ahmad - 21111060	Environment, Feature selection, Literature survey.
Prakhar Srivastava - 170486	Environment, Pygame and model implementation, Literature Survey, Solution approach, Feature selection, Training agent, getting results, final slides and final report.
Areeb Ahmad -180135	Environment, Solution Approach, Training agent, plotting results, final slides and final report, Data Analysis.
Mohd Niyas P -170394	-
Samarth Mehrotra -20128408	-

DEMO

REFERENCES

- Charnov, E. L. 1976. Optimal foraging: the marginal value theorem. Theoretical Population Biology 9:129–136
- Cowie, R. J. (1977) "Optimal foraging in great tits (Parus Major)" Nature 268:137–139
- Cassini, Marcelo H., Alejandro Kacelnik, and Enrique T. Segura (1990) "The tale of the screaming hairy armadillo, the guinea pig, and the marginal value theorem" Animal Behavior 39(6):1030–1050
- A. Mesquita, Y. Nogueira, C. Vidal, J. Cavalcante-Neto and P. Serafim, "Autonomous Foraging with SARSA-based Deep Reinforcement Learning," 2020 22nd Symposium on Virtual and Augmented Reality (SVR), 2020, pp. 425-433, doi: 10.1109/SVR51698.2020.00070.
- https://neat-python.readthedocs.io/en/latest/neat_overview.html