Group_22 - AI Project (DS251)
Multi Agent Hide and Seek using Reinforcement Learning

Group Members:

K. Aditya Vardhan	- 12040740
Vetsa Tarun	- 12041720
NBS Bharath	- 12040940
Velluru Shyam	- 12041700
Imdadulla Khan	- 12040530

1. Introduction:

Welcome to "Hide and Seek," a dynamic project where strategy and fun converge. Experience the thrill as our intelligent hider and seeker navigate the game arena, joined by additional agents for extra excitement. This project explores the realm of reinforcement learning, creating a delightful blend of playfulness and strategic decision-making. Get ready for a captivating journey where every move unveils new possibilities and challenges.

2. State Space :

Defining the State as is in the form of tuples (x_start_coordinate, y_start_coordinate, Distance from Initial coordinates, Cumalative rewards till now) .

3. Algorithm:

We are using Q-learning Algorithm to perfectly determine which state we have to move onto next. In Q-learning, an agent learns to make decisions by updating Q-values for state-action pairs based on observed rewards. The Q-value for a state-action pair represents the expected cumulative reward.

The agent explores the environment, updating Q-values using the equation:

$$\boxed{Q(s,a) \leftarrow (1-\alpha) \,.\, Q(s,a) + \alpha \,.\, (R + \gamma \,.\, \max_a Q(s',a')\,)}$$

- Q(s,a): Q-value for state-action pair (s,a).
- α : Learning Rate
- R : Immediate Reward
- γ : Discount factor.
- s´ : Next state
- a´ : Action in the next state

4. Tech stack used:

- ***Programming Language:*** Python
- ***GUI Development:*** Tkinter for creating a user-friendly graphical interface
- ***3D Visualization:*** Raycast library for rendering visual elements in a simplified manner
- ***Data Storage:*** Pickle files for saving and loading trained game states
- ***Algorithm:*** *Q-learning*

5. Training Process:

The training process involves iteratively refining the behavior of seekers and hiders through reinforcement learning.

1. *Policies and Rules:*

   The policy.py code defines a reinforcement learning-based policy for players in a hide-and-seek game. The Policy class represents the decision-making strategy for both seekers and hiders. Seekers aim to optimize detection strategies, while hiders focus on successful evasion and hiding.

   The Q-learning algorithm is employed to update the Q-values for state-action pairs, guiding the players to make informed decisions based on past experiences. The exploration-exploitation trade-off is managed through the epsilon-greedy strategy, balancing random exploration with exploitation of learned Q-values. The code captures the learning and decision-making process of players in a dynamic hide-and-seek environment.

2. *Reward System:*

   The reward system in the given code is straightforward. Positive actions, like successful hiding or catching a hider, add to the agent's total reward. On the other hand, collisions with walls or undesirable actions lead to penalties, reducing the total reward. This approach guides the Q-learning algorithm to update values based on the cumulative impact of rewards and penalties, helping the agent learn effective hide-and-seek strategies.

3. *Iterative Improvement:*

   Over approximately 10,000 to 12,000 iterations, agents undergo iterative improvement by learning from experiences. The Q-learning algorithm updates strategies based on observed rewards and policies. The trained data, including states, actions, and rewards, is stored in a pickle file, allowing the agents to retain learned knowledge across iterations and enhancing their hide-and-seek strategies over time.

4. *State-Action Exploration:*

    Agents explore different state-action pairs to find the best strategies, updating Q-values in pickle files. The Q-learning algorithm focuses more on using learned values when the learning rate (α) exceeds 0.1 . This helps the agents improve their hide-and-seek behaviors, making their decision-making smarter and more effective as they learn over time.

6. Pickle Files:

Explain the purpose of the pickle files, how they store initial states, and how the seeker's moves are updated and saved in each iteration.

1. *Initial Pickle File Content:*

    The initial content of the hider pickle file includes state-action pairs with their corresponding Q-values. For example:

    Key(state) :: (384, 184, 0.0, 0.0)
    Value(action) :: {'LEFT': 0, 'RIGHT': 0, 'UP': 0.06015678618675692, 'DOWN':0.13186272464634627}

    This represents the initial Q-values for specific states and actions, providing a foundation for the hider's decision-making process.

2. *Training Process and Update:*

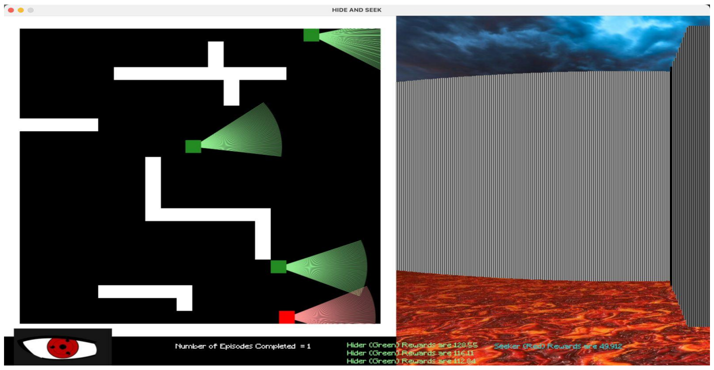
    Detail how the pickle file is updated after each iteration of training. Explain how the Q-values are adjusted based on the hider's experiences during the game. For example, if the hider successfully avoids the seeker, the Q-value for the chosen action in that state might be increased.
    Similar approach is taken for the seeker, with a separate pickle file storing state-action pairs and associated Q-values for the seeker's decision-making.

3. *Iterative Improvement:*

    Highlight the significance of the iterative process in refining the Q-values. With each iteration, the hider and seeker become more adept at making decisions based on learned experiences, contributing to improved gameplay.

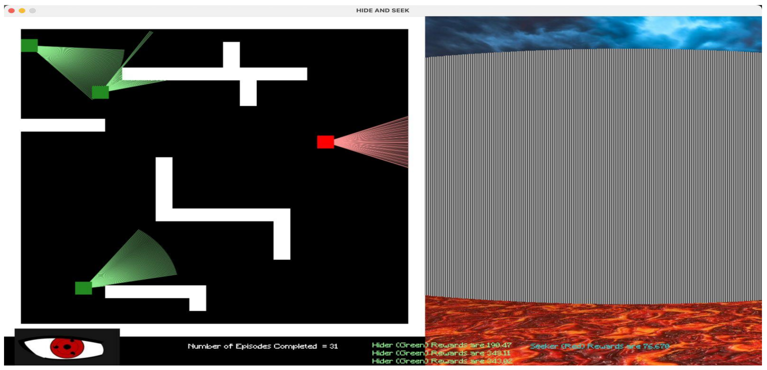
7.Results:

1. *Untrained Game:*
   In the initial phase without training, the hide and seek agents exhibit random and unoptimized behavior. The seekers may struggle to locate the hiders effectively, and hiders might not employ strategic hiding patterns.



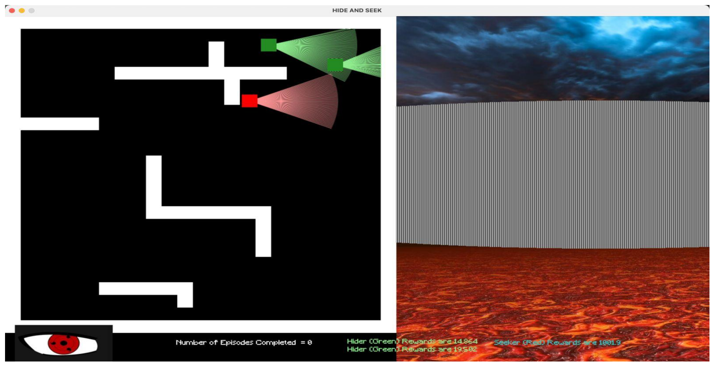
```
Number of Episodes Completed = 1    Hider (Green) Rewards are 128.55    Seeker (Red) Rewards are 49.912
                                    Hider (Green) Rewards are 116.11
                                    Hider (Green) Rewards are 112.84
```

2. *Trained Game:*
   After approximately 900 to 1000 iterations of training, the hide and seek agents demonstrate improved and strategic behavior. Seekers show enhanced pursuit techniques, and hiders display effective hiding strategies.



```
Number of Episodes Completed = 31    Hider (Green) Rewards are 190.47    Seeker (Red) Rewards are 76.670
                                     Hider (Green) Rewards are 348.11
                                     Hider (Green) Rewards are 343.02
```

(i) Here the hiders are using effective strategies to hide



```
Number of Episodes Completed = 0    Hider (Green) Rewards are 14.864    Seeker (Red) Rewards are 1001.9
                                    Hider (Green) Rewards are 19.502
```

(ii) Here the seeker enhancing his pursuit

The visual comparison between the untrained and trained games highlights the progress achieved through the reinforcement learning process. This evolution in agent behavior is a testament to the effectiveness of our training approach and the potential of the developed hide and seek game.

8. Experiments:

We can also train the game with a measure of the time taken to catch the hider, the number of successful hides by the hider, or something like these, so we could train the game with the following metrics:

1. *Heuristic Based Approach:*
   We used heuristics to guide the behavior of hiders and seekers. For instance, hiders may move toward areas with more obstacles, and seekers may move toward areas with fewer obstacles.

2. *Deep Q-Networks:*
   We used Deep Q-Networks to learn optimal strategies for hiders and seekers in a hide and seek game by providing a state representation as input and training the network to estimate Q-values for actions, optimizing for cumulative rewards..

3. *Monte Carlo Methods:*
   We applied Monte Carlo methods to a hide and seek game by simulating multiple episodes, estimating action values through reward averaging, and using the results to improve strategies, guiding hiders and seekers in the game.

9. References:

https://openai.com/research/emergent-tool-use
https://dohyeongkim.medium.com/implementation-of-the-hide-and-seek-of-the-openai-part-1-3e1b9eeb86cf