A

REPORT

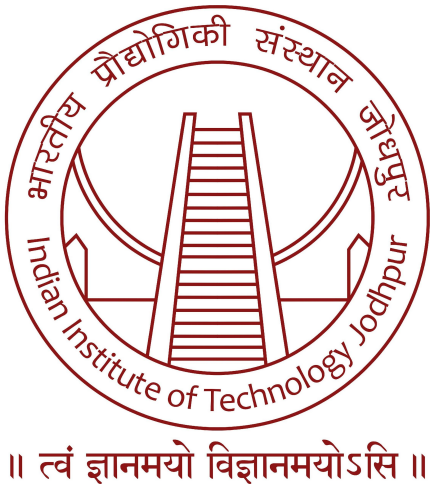
ON

“ADVANCE ARTIFICIAL INTELLIGENCE: ASSIGNMENT 4”

Submitted To

DR. GAURAV HARIT

IIT JODHPUR



Submitted By

Alok Kumar Vinay Kumar Shukla (M22CS051)
Bikas Dutta(D22CS051)

AAI Assignment-4 Report

Task: To implement time Deep-RL Algorthim.

Objectives: Part 1: Describe the game

- Describe this game in terms of agent, environment, actions, and observations. Describe an appropriate reward function for this game, and briefly explain why it is valid for this game.
- Hint: You can try playing the game or look at its source code in game_state.py.

Procedure:

- Agent: The player-controlled entity and the enemy entities.
- Environment: A rectangular game board with a player-controlled entity, enemy entities, and a goal location.
- Actions: The player-controlled entity can move up, down, left, or right. Enemy entities move randomly.
- Observations: The agent can observe the current state of the game, including the locations of the player-controlled entity, enemy entities, and goal locations.

common > game_constants.py > ...

```python
GAME_WIDTH = 800
GAME_HEIGHT = 600
PLAYER_SIZE = 20

GAME_FRICTION = 0.05
GAME_ACC_FACTOR = 0.6

GOAL_SIZE = 10

ENEMY_SIZE = 25
ENEMY_COUNT = 7
ENEMY_SPEED = 4

BACKGROUND = (0x00, 0x50, 0x50)
ENEMY_COLOR = (0xfc, 0x20, 0x20)
PLAYER_COLOR = (0x20, 0xfc, 0x20)
GOAL_COLOR = (0x20, 0xfc, 0xfc)

FPS = 20

# You can play around with these constants, but for evaluation
# of your code, the original values will be used.


# The fixed game seed that your game must be based on.
# It must be unique among all class members.
GAME_SEED = 422319051#'Assign your integer unique seed value here'

```

Fig: Default Game Constraints.

Objectives: Part 2: Deep RL algorithm

- Select a suitable Deep RL algorithm and implement it in the game. You must add this implementation in the file game_controller.py , within the class functions of AIController .
- Notice the comments that explain which portion of the file you are allowed to change, and complete the implementation accordingly. On running the game from the file game.py , enter the prompt to use your AI Model, and the code will automatically call the Train() function of the controller and run a custom evaluation on it before launching the game. You are required to mention this evaluated result in your report.
- Note: The game uses the library pygame to render the game. Use pip install pygame to ensure before running the game.

Procedure:

- The GetAction function of the KeyboardController class returns the action that the player should take in response to the key they have pressed. It takes the game state as input, gets the pressed key, and returns the corresponding action.

- The AIController class has the same function GetAction, but it returns the action that the AI agent should take based on the current state of the game. It uses an epsilon-greedy policy to select the action, where epsilon is the exploration rate.

- The controller class also has a function TrainModel, which is not implemented yet but is expected to train the AI agent in the game environment using a suitable reinforcement learning algorithm.

- The AIController class has a Q-table that keeps track of the expected reward for taking each action at each state. The Q-table is initialized with zeros.

- The TrainModel function implements the Q-learning algorithm to update the Q-table by exploring the game environment. It updates the Q-table using the current state, the action taken, the reward obtained, and the next state.

- The TrainModel function sets hyperparameters like the learning rate (alpha), discount factor (gamma), and the number of episodes to run the training (epochs).

- The TrainModel function updates the Q-table for each episode until the goal is reached or the enemy attacks the player. The function also updates the exploration rate (epsilon) for each successful episode.

- The reward system is designed in a way that encourages the AI agent to move toward the goal and penalizes it for moving away from it. If the AI agent reaches the goal, it gets a high reward, but if the enemy attacks the player, it gets a negative reward.

Fig: Hyperparms for training and penalties for the agent.

- The Epsilon is self.epsilon = 1, because we want the agent to explore at the starting phases.
- Alpha is 0.55 because we want the learning algo to take bigger steps, and also tried learning rate decay but didn't get good results.
- Gama is 0.99, i.e., discounting factor.
- Epochs are 100, i.e., training iterations for the model.
- Penalties:

    - Distance-Based Penalties:
        - Distance b/w agent and goal, if increased ⬇️; otherwise, ⬆️.
        - (experimented) Distance b/w agents and enemies and apply rewards accordingly.
    - Goal-Based Penalties:
        - If you reach the goal, 👍
        - If colliding with the enemy, 👎
    - Note: also modifying the epsilon such that, agent do less exploration and more exploitation.
- Result:

```
~/IIT-J/AAI/Assignments/Assignment4 17s ❯ python3 game.py
pygame 2.3.0 (SDL 2.24.2, Python 3.10.8)
Hello from the pygame community. https://www.pygame.org/contribute.html
Welcome to AI Assignment 4 program!

1. Run game on your AI model
2. Try the game by controlling with keyboard
Enter your choice: 1
AI controller initialized!
Now training...
AI controller is trained!
Now evaluating...
On Evaluation, player died 2844 times, while reaching the goal 13 times
Would you like to see how this model performs on the game (y/n)?y
```

Fig: Training the model with the above Hyper-Parms.

Objectives: Part 3:

- On the basis of your selected algorithm, briefly explain if after training your model, would it still score well in this game (without further retraining) on changing the following game constants (defined in the file game_constants.py)?
- If GAME_SEED is selected randomly, rather than keeping it a constant number.
- If the dimensions of the game, i.e GAME_WIDTH and GAME_HEIGHT are changed.
- If the variable GOAL_SIZE is changed.
- If the variable ENEMY_COUNT is changed.
- If the variable GAME_FRICTION is changed.
- If the variable FPS is changed.

Procedure:

- If GAME_SEED is selected randomly, rather than keeping it a constant number:
    - The random selection of GAME_SEED would result in a different game environment in each run.
    - The model's performance may vary depending on the randomly generated environment, and may not perform as well as it did during training.
- Result:

```
~/IIT-J/AAI/Assignments/Assignment4 17s ❯ python3 game.py
pygame 2.3.0 (SDL 2.24.2, Python 3.10.8)
Hello from the pygame community. https://www.pygame.org/contribute.html
Welcome to AI Assignment 4 program!

1. Run game on your AI model
2. Try the game by controlling with keyboard
Enter your choice: 1
AI controller initialized!
Now training...
AI controller is trained!
Now evaluating...
On Evaluation, player died 2844 times, while reaching the goal 13 times
Would you like to see how this model performs on the game (y/n)?y
```

- If the dimensions of the game, i.e GAME_WIDTH and GAME_HEIGHT are changed:
    - Changing the dimensions of the game would result in a different game environment with different input features for the model.
    - If the model was trained on a specific size of game, it may not perform as well on different sized games.
- Result

```
~/IIT-J/AAI/Assignments/Assignment4 52s ❯ python3 game.py
pygame 2.3.0 (SDL 2.24.2, Python 3.10.8)
Hello from the pygame community. https://www.pygame.org/contribute.html
Welcome to AI Assignment 4 program!

1. Run game on your AI model
2. Try the game by controlling with keyboard
Enter your choice: 1
AI controller initialized!
Now training...
AI controller is trained!
Now evaluating...
On Evaluation, player died 3235 times, while reaching the goal 16 times
Would you like to see how this model performs on the game (y/n)?n
```

Fig: Game Window Size 700x600

- If the variable GOAL_SIZE is changed:
    - Changing the size of the goal would affect the difficulty of the game.
    - If the goal is made smaller, the game would become harder and the model may not perform as well as it did during training.
- Result:

```
~/IIT-J/AAI/Assignments/Assignment4 27s ❯ python3 game.py
pygame 2.3.0 (SDL 2.24.2, Python 3.10.8)
Hello from the pygame community. https://www.pygame.org/contribute.html
Welcome to AI Assignment 4 program!

1. Run game on your AI model
2. Try the game by controlling with keyboard
Enter your choice: 1
AI controller initialized!
Now training...
AI controller is trained!
Now evaluating...
On Evaluation, player died 3158 times, while reaching the goal 15 times
Would you like to see how this model performs on the game (y/n)?n
```

Fig: Goal Size Changed to 15

- If the variable ENEMY_COUNT is changed:
    - Changing the number of enemies in the game would affect the difficulty of the game.
    - If the number of enemies is increased, the game would become harder and the model may not perform as well as it did during training.
- Result:

```
~/IIT-J/AAI/Assignments/Assignment4 16s ❯ python3 game.py
pygame 2.3.0 (SDL 2.24.2, Python 3.10.8)
Hello from the pygame community. https://www.pygame.org/contribute.html
Welcome to AI Assignment 4 program!

1. Run game on your AI model
2. Try the game by controlling with keyboard
Enter your choice: 1
AI controller initialized!
Now training...
AI controller is trained!
Now evaluating...
On Evaluation, player died 2167 times, while reaching the goal 15 times
Would you like to see how this model performs on the game (y/n)?n
```

Fig: Enemy Count Changed to 5

```
~/IIT-J/AAI/Assignments/Assignment4 15s ❯ python3 game.py
pygame 2.3.0 (SDL 2.24.2, Python 3.10.8)
Hello from the pygame community. https://www.pygame.org/contribute.html
Welcome to AI Assignment 4 program!

1. Run game on your AI model
2. Try the game by controlling with keyboard
Enter your choice: 1
AI controller initialized!
Now training...
AI controller is trained!
Now evaluating...
On Evaluation, player died 4220 times, while reaching the goal 10 times
Would you like to see how this model performs on the game (y/n)?n
```

Fig: Enemy Count Changed to 10

- If the variable GAME_FRICTION is changed:
    - Changing the friction of the game would affect the movement of the player and enemies.
    - If the friction is increased, the game would become slower and the model may not perform as well as it did during training.
- Result:

```
~/IIT-J/AAI/Assignments/Assignment4 14s ❯ python3 game.py
pygame 2.3.0 (SDL 2.24.2, Python 3.10.8)
Hello from the pygame community. https://www.pygame.org/contribute.html
Welcome to AI Assignment 4 program!

1. Run game on your AI model
2. Try the game by controlling with keyboard
Enter your choice: 1
AI controller initialized!
Now training...
AI controller is trained!
Now evaluating...
On Evaluation, player died 3167 times, while reaching the goal 19 times
Would you like to see how this model performs on the game (y/n)?n
```

Fig: Game Friction Changed to 0.3

```
~/IIT-J/AAI/Assignments/Assignment4 ❯ python3 game.py
pygame 2.3.0 (SDL 2.24.2, Python 3.10.8)
Hello from the pygame community. https://www.pygame.org/contribute.html
Welcome to AI Assignment 4 program!

1. Run game on your AI model
2. Try the game by controlling with keyboard
Enter your choice: 1
AI controller initialized!
Now training...
AI controller is trained!
Now evaluating...
On Evaluation, player died 2727 times, while reaching the goal 5 times
Would you like to see how this model performs on the game (y/n)?n
```

Fig: Game Friction Changed to 0.7

- If the variable FPS is changed:
    - Changing the FPS (frames per second) would affect the speed at which the game is played.
    - If the FPS is decreased, the game would become slower and the model may not perform as well as it did during training.
- Result:

```
~/IIT-J/AAI/Assignments/Assignment4 12s ❯ python3 game.py
pygame 2.3.0 (SDL 2.24.2, Python 3.10.8)
Hello from the pygame community. https://www.pygame.org/contribute.html
Welcome to AI Assignment 4 program!

1. Run game on your AI model
2. Try the game by controlling with keyboard
Enter your choice: 1
AI controller initialized!
Now training...
AI controller is trained!
Now evaluating...
On Evaluation, player died 3214 times, while reaching the goal 11 times
Would you like to see how this model performs on the game (y/n)?n
```

Fig: FPS changed to 20

```
18
19    FPS = 40
20
21    # You can play around with these constants, but for evaluation
22    # of your code, the original values will be used.
23
24
25    # The fixed game seed that your game must be based on.
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
~/IIT-J/AAI/Assignments/Assignment4 ❯ python3 game.py
pygame 2.3.0 (SDL 2.24.2, Python 3.10.8)
Hello from the pygame community. https://www.pygame.org/contribute.html
Welcome to AI Assignment 4 program!

1. Run game on your AI model
2. Try the game by controlling with keyboard
Enter your choice: 1
AI controller initialized!
Now training...
AI controller is trained!
Now evaluating...
On Evaluation, player died 2904 times, while reaching the goal 9 times
Would you like to see how this model performs on the game (y/n)?n
```

Fig: FPS changed to 40

- If the Enemy Count is changed:

- Changing the enemy count would affect the safe position to move inside the game window; hence, increasing the number would result in more deaths.
- If the enemy count is decrease,d then the window becomes sparse and the agent can more more freely without the collision, hence increasing the reaches to goal state.

- Result:

```
~/IIT-J/AAI/Assignments/Assignment4 16s ❯ python3 game.py
pygame 2.3.0 (SDL 2.24.2, Python 3.10.8)
Hello from the pygame community. https://www.pygame.org/contribute.html
Welcome to AI Assignment 4 program!

1. Run game on your AI model
2. Try the game by controlling with keyboard
Enter your choice: 1
AI controller initialized!
Now training...
AI controller is trained!
Now evaluating...
On Evaluation, player died 2167 times, while reaching the goal 15 times
Would you like to see how this model performs on the game (y/n)?n
```

Fig: Enemy Count Decreased to 5

```
~/IIT-J/AAI/Assignments/Assignment4 15s ❯ python3 game.py
pygame 2.3.0 (SDL 2.24.2, Python 3.10.8)
Hello from the pygame community. https://www.pygame.org/contribute.html
Welcome to AI Assignment 4 program!

1. Run game on your AI model
2. Try the game by controlling with keyboard
Enter your choice: 1
AI controller initialized!
Now training...
AI controller is trained!
Now evaluating...
On Evaluation, player died 4220 times, while reaching the goal 10 times
Would you like to see how this model performs on the game (y/n)?n
```

Fig: Enemy Count Increased to 10

- Note: All the above runs are executed separately.