Maze Runner Simulation
COMP 3106
Param Desai: 101263669
William Marcus: 101219545
JC Sevigny: 101219735

Statement of contributions

Our group consists of Param Desai, William Marcus & JC Sevigny. Each member made a significant and equal contribution to this project. The project breakdown of what each member worked on is shown below:

Param Desai

- Collaborated on the project proposal.
- Collaborated and worked on the implementation of the code.
- Specifically worked on the Q-Learning agent, artificial life walls, and rule-based systems for grievers aspects of the code.
- Tested with unique CSV files and validated and debugged from results
- Collaborated on the project report.

William Marcus

- Collaborated on the project proposal.
- Collaborated and worked on the implementation of the code.
- Specifically worked on the GUI representation of the maze runner simulation.
- Tested with unique CSV files and validated and debugged from results
- Collaborated on the project report.

JC Sevigny

- Collaborated on the project proposal.
- Collaborated and worked on the implementation of the code.
- Tested with unique CSV files and validated and debugged from results
- Collaborated on the project report.

Introduction

Background and Motivation:

The Maze Runner simulation project motivation was inspired by the popular movie *"The Maze Runner"* where the protagonist must navigate shifting pathways within a large maze while avoiding deadly threats. This project takes the idea of using artificial intelligence to solve the challenge of surviving and finding a correct path through the dynamic maze while being faced with constant changes and threats. The main motivation comes from the opportunity to use many tools and technologies from artificial intelligence such as reinforcement learning, artificial life, and rule-based systems. This mix of different artificial intelligence concepts raises the opportunity to develop an intelligent agent which makes a very compelling application of problem-solving and finding the most optimal path with AI methods.

Related Prior Work and References:

Below are some prior works to similar types of projects which we analyzed to create this project. Some of this prior work focuses on just one part of the Artificial Intelligence methods that we are implementing in our project.

1. Q-Learning Maze Solver by Samy Zaf (Link)

   Description of the work:

This project reports a walkthrough of Q-Learning applied to a maze environment. The details are about how an agent explores the maze environment, updates the Q-Table, and converges toward an optimal path using a Bellman equation.

Analysis of the work and knowledge gained:

1. Agent Initialization:
    a. The agent starts with no knowledge about the environment and the Q-table is initiated with 0 values for all state action pairs.
        i. This helps with the maze runner project initialization process where the agent begins with an untrained Q-Table and must learn through trial and error. Understanding how initial Q-values influence early exploration can help with where the agent can be stuck in the maze for example.
2. Q-Table update mechanism:
    a. The q values are updated using the Bellman equation:
    $Q(s,a) \leftarrow Q(s,a) + \alpha \times [r + \gamma \times a' max Q(s',a') - Q(s,a)]$
    b. The project explains how this formula helps the agent learn the expected long-term reward for taking an action in a given state.
        i. For the maze runner project, by comparing the update process, we can ensure Q-tables update correctly, especially in complex scenarios with dynamic walls and grievers. For example, we can discover subtle issues where rewards are not propagating backward as expected.
3. Exploration vs. Exploitation:
    a. The project uses an ε-greedy policy to balance exploration and exploitation. With probability ϵ, the agent chooses a random action. With probability 1-ϵ, the agent selects the action with the highest Q-value for the current state.
        i. This resource provides strategies for tuning ϵ. If your agent prematurely converges to suboptimal paths, adopting a similar decay strategy could help balance exploration during training.

2. Solving a Maze with Q-Learning by Mitchel Spryn (Link)

Description of the work:

This article describes the process of implementing Q-Learning to solve a 2D maze, with an emphasis on a grid-based environment. It explains how to simulate rewards and penalties ensuring the agent avoids unnecessary moves or obstacles.

Analysis of the work and knowledge gained:

1. Reward System:
    a. Spyrn assigns rewards as follows:
        - Large Positive rewards for reaching the goal.
        - Small Penalties for each step to encourage efficiency.
        - Larger Penalties for hitting walls or invalid moves.
    b. In relation to the maze runner project, the step penalties help guide the agent toward efficient paths. Applying penalties in the Maze Runner project can prevent the agent from wandering unnecessarily in the maze.
    c. The prior work does not explicitly incorporate griever or dynamic elements, but the reward structure offers a good idea to integrate the grievers and dynamic wall elements.
2. Maze Complexity:
    a. Spyrn tests the Q-learning agent in mazes of varying sizes and complexities, demonstrating the ability to generalize a solution.

b. Testing in larger or more intricate mazes can validate the efficiency of our implementation. Observing how the agent handles additional grievers or dynamic walls could offer insights into scalability.

3. Q-Learning Maze Game (Github Repository) by edvujic (Link)

   Description of the work:

   This repository provides an implementation of a Q-learning-based maze game. The agent learns optimal paths through trial and error while navigating an interactive environment.

   Analysis of the work and knowledge gained:

   1. Learning Process:
      a. The agent explores the maze updating its Q-values based on the rewards received from actions taken. The Q-values are updated using the Bellman equation

      $Q(s,a) \leftarrow Q(s,a) + \alpha \times [r + \gamma \times max a' Q(s',a') - Q(s,a)]$

      Where: Q(s,a): Current Q-value for state-action pair, α: Learning Rate, r: Reward received after taking action a from state s, γ: Discount factor for future rewards, $max a' Q(s',a')$: Maximum Q-value for the next state s′ over all possible actions a′

4. Simulating the Effect of Environmental Change on Evolving Populations (Link)

   Description of the work:

   This study explores how populations evolve strategies to cope with random environmental changes, particularly when lifetime learning is not an option. It examines the impact of change magnitude and persistence on adaption, providing insights into how dynamic environmental factors such as moving walls.

   Analysis of the work and knowledge gained:

   a. This study examines how agents adapt to random environmental changes over time, which helps the maze runner project which uses dynamic walls to introduce unpredictable changes.
   b. Since the dynamic walls are only done when triggered by a condition, the study provides insights into how environment changes can shape agent strategies, such as learning to avoid triggers and optimizing paths in response to dynamic walls.
   c. The study offers a way to evaluate how well the maze runner agent adapts to dynamic wall changes particularly introducing stochastic elements.

5. Fuzzy Cognitive Maps for Decision-Making in Dynamic Environments (Link)

   Description of the work:

   This research introduces a modified Fuzzy Cognitive Map tailored for autonomous agents making decisions in dynamic environments. This categorizes concepts into Needs, Activities, and States, facilitating agents to adapt their behaviors based on environmental triggers. This structure allows for responsive changes in the environment similar to dynamic walls activated by specific conditions.

   Analysis of the work and knowledge gained:

   a. Dynamix walls in the Maze Runner project can be seen as representing states in the environment that change based on the agent's activities (like the triggering condition)
   b. The work introduces a structured way to model environmental changes using Needs, Activities, and State.
   c. This type of organization could enhance dynamic changes, ensuring that they not only respond to triggers but also align with broader goals or constraints in the maze environment.

6. Non-player character decision-making in computer games (Link)

   Description of the work:

This paper surveys current decision-making methods used by NPCs (Non-Playable Characters), categorizing them into distinct approaches. It offers a comprehensive analysis of highlighting applications and effectiveness in various gaming scenarios.
Analysis of the work and knowledge gained:

a. The work categorizes NPC decision-making approaches into five methods, including rule-based systems and hybrids.
b. The griever's left-right movement can be refined using conditional rules such as “move left unless blocked, then move right”
c. This work helps us effectively prioritize and sequence rules to prevent conflicting behaviors, such as simultaneous movement in opposing directions.

7. Biologically-Inspired Gameplay: Movement Algorithms for Artificially Intelligent Non-Player Characters (Link)

Description of the work:
This paper examines movement algorithms inspired by NPC entities such as flocking behaviors, to create realistic group dynamics among NPCs.
Analysis of the work and knowledge gained:

a. This work explores how biological movement principles can enhance NPC behavior realism, even in simple rule-based systems.
b. For the maze runner project, this work suggests implementing pseudo-biological behaviors for Grievers like grievers move left-right normally but avoid proximity if other grievers are nearby and avoid running into walls.

Statement of Objectives:
The Statement of Objectives outlines the key goals and milestones our project aims to achieve, providing a clear framework for its development and evaluation:

- **Developing an Intelligent Agent with Q-Learning for Pathfinding**: Creating an agent capable of navigating from a start point to a goal within a maze. Optimizing the agent’s decision-making using Q-values, rewards for an efficient path and penalizing when interacting with a wall or griever.
- **Incorporating Artificial Life for Dynamic Walls**: Designing the walls to change dynamically when there is an agent-triggered event, simulating a living environment that challenges the agent.
- **Design Rule-Based systems for NPC behavior (Griever)**: Implement grievers with rule-based movement that follows predefined rules such as moving to the left or moving to the right. So ensuring the agent interacts with grievers properly and penalizes collisions with the grievers.
- **Performance Metrics**: To evaluate the success of our project, we will use performance metrics such as completion time, number of steps, best possible path, and adaptability to dynamic changes, comparing these results to those of a naïve agent to measure how effectively our learning agent improves. This also comes with testing the effectiveness of Q-Learning by comparing the agent’s performance with the maze runner agent.

Methods

Methods of Artificial Intelligence:
In this project, we used several different artificial intelligence methods to implement a dynamic and adaptive Maze Runner simulation. The methods used include **Q-learning**, **artificial life,** and **rule-based systems** which were utilized to handle various aspects of the simulation such as agent behavior, griever actions, and dynamic wall changes.

Q-learning
Q-learning was used to train the Maze Runner agent to find the optimal path through the maze. This method was particularly used for this environment because the agent needs to learn through trial and error and adapt to evolving conditions.
Motivation for Using Q-Learning Agent:

For this agent, we were thinking of using A* search, but since the algorithm is for deterministic and fully observable environments, it assumes complete knowledge of the maze and does not adapt to dynamic changes of the walls or griever actions. The project was inspired by the Maze Runner movie where the maze was only partially observable. Q-learning is designed to handle partially observable environments where the agent lacks complete knowledge of the maze and must adapt to changes based on rewards and penalties encountered during exploration.
Implementation:
The maze is represented as a grid with a cell as a state. Rewards include; +100 for reaching the goal, -40 for encountering a griever and -50 for traversing a wall. The agent gets to select actions(up, down, left, and right) based on the Q-value table updated during the training episode (using 10000 episodes for training). The agent chooses actions based on an epsilon-greedy policy, balancing exploration and exploitation. The Q-Table stores the expected reward for each state-action pair during training using the formula
$Q(s,a) \leftarrow Q(s,a) + \alpha \cdot (r + \gamma \cdot a max Q(s',a) - Q(s,a))$. The fine-tuning parameters include learning rate (a = 0.01), discount factor ($\gamma = 0.9$), r is the reward, and Exploration rate (0.2). The exploration rate decreases over training episodes to focus on exploiting learned strategies. The agent does not have prior knowledge of dynamic changes or griever positions beyond immediate surroundings, making the environment partially observable.
Benefits and Drawbacks:
Q-learning enables the agent to adapt dynamically to both static and dynamic obstacles. Also facilitates learning in partially observable settings where new information is revealed as the agent explores. Some drawbacks are that it requires significant training time for convergence, especially in complex environments. The performance depends heavily on fine-tuning parameters like learning rate, discount factor, and exploration rate.

Artificial Life:

The concept of artificial life was applied to simulate dynamic environmental changes through dynamic walls. These walls move to predefined target locations when triggered by the agent's traversal of specific cells. This method enhances the complexity of the path-finding agent.
Motivation for Using Artificial Life for Dynamic Walls:
Dynamic wall behavior brings unpredictability, challenging the agent to adapt its navigation strategy dynamically. This approach provides an interactive environment that forces the agent to plan ahead and react to sudden changes.
Implementation:
Trigger cells and target positions are defined in an input file (dynamic_walls.txt). When the agent visits a trigger cell, the wall moves or appears to its target locations. The wall movements are processed in real time during the agent's traversal. Triggered walls update the maze's layout, forcing the agent to reevaluate its strategy. There is then an integration with the Q-Learning agent where the dynamic wall behavior allows the agent to adapt to its policy based on environmental changes.
Benefits and Drawbacks of Artificial Life for Dynamic Walls
The advantages include how the complexity of the pathfinding challenges the agent to adapt dynamically simulating the case in the movie as well. There are drawbacks as well which include additional computational overhead to manage the dynamic changes making the training slower. The trigger and target configuration have to be carefully balanced to maintain the integrity of the maze.

Rule-Based System:

The movement of the grievers in the maze is governed by a rule-based system. This method defines behavior through deterministic rules to move in the maze for added complexity for the maze runner agent.
Motivation for using rule-based systems for the movement of the grievers:
Rule-based systems provide a perfect NPC behavior creating consistent challenges for the agent without excessive computational complexity. By defining clear rules for griever movement, we ensure proper movement and interactive obstacles for the maze runner agent.
Implementation:
The Grievers follow these simple rules for movement:

- If there is an empty cell to the left, move left.
- If the left cell is blocked by a wall, check the right cell and move right.

- If both directions are blocked by walls, stay in place.

The griever position is updated at each step of the simulation based on the rules. This is then integrated with a Q-Learning agent which accounts for griever movements during training and pathfinding. States with grievers are penalized to discourage the agent from encountering them.
Benefits and Drawbacks:
Benefits include the simplicity ensures manageable NPC behavior but still adds complexity to the maze, allowing the agent to learn effective avoidance strategies. Some drawbacks are that the deterministic nature of griever movement limits the complexity of the interactions. This also might make the environment predictable over time reducing the challenge for the agent.

Datasets:
The maze and dynamic walls were defined using structured datasets in CSV and text file formats:
The maze structure is represented as a 2D grid in a CSV file, where:

- 0 indicates walkable cells.
- 1 indicates walls.
- S marks the start position.
- E marks the goal position.
- G (or -1) indicates griever positions.

A text file defines dynamic walls, with each line specifying a trigger position and its corresponding target positions. For example: (1,3):(1,2), where (1,3) is the trigger, and (1,2) is where the wall will dynamically be changed. This dataset is parsed at runtime, allowing the walls to move dynamically when triggered by the agent.
Example of the CSV and the dynamic walls text file:

Maze.csv:
S,0,0,1,1,1
1,1,0,0,1,1
0,0,0,0,G,0
1,0,1,0,0,1
1,0,0,0,0,0
1,0,0,0,G,0
1,0,E,0,0,1

DynamicWalls.txt:
(2,2):(5,1)
(0,1):(6,1)

Validation Strategies:
To validate the methodologies employed in the Maze Runner simulation project, we adopted a multifaceted approach designed to evaluate the performance and efficiency of the artificial intelligence techniques implemented. Here are the detailed validation strategies:

Q-Learning Validation:
Objective: Validate the effectiveness of Q-Learning in navigating the maze efficiently and avoiding obstacles like grievers and walls.
Strategy:

- Training Convergence: Track the convergence of Q-values over multiple episodes to ensure the learning algorithm stabilizes to an optimal policy.
- Performance Metrics:
    - Path Length: Measure the length of the path taken by the agent in the maze. A shorter path indicates better learning of the maze path.

- Penalty avoidance: Monitor the agent's ability to avoid penalties during navigation.
- Analyze how the agent adapts to dynamic walls and moving grievers, emphasizing its ability to handle environmental changes in a partially observable environment.

Validation of Dynamic Walls (Artificial Life):
Objective: Assess the implementation and impact of dynamic wall movements on the simulation's complexity and agent behavior.
Strategy:

- Trigger Testing: Validate that all dynamic wall triggers function correctly by activating them in controlled scenarios.
- Environment Adaptability: Measure the agent's ability to adapt to pathfinding when walls move and appear unpredictably.
- Assessing the Complexity: Evaluate how the inclusion of dynamic walls increases the complexity of the maze and impacts the agent's decision-making.

Rule-Based System Validation for Grievers:
Objective: Evaluate the impact of rule-based griever movements on the agent's strategy for pathfinding.
Strategy:

- Behavior Testing: Validate the griever's move according to the predefined rules (left or right) and stay within the bounds of the maze.
- Collision Avoidance from the Agent: Monitor the agent's ability to predict griever movements and avoid collisions effectively.
- Path Rerouting: Measure how frequently the agent changes its path to avoid grievers and record how often the agent encounters grievers and the resulting penalties.

General Environment Validation:
Objective: Ensure the overall integrity and functionality of the simulated environment.
Strategy:

- Randomized Testing: Generate Mazes with varying configurations of walls, dynamic triggers, and grievers to test the efficiency of the simulation.
- Test Edge Case Scenarios:
    - Test Scenarios with no grievers or dynamic walls to isolate agent behavior. And also introduces high-density obstacles to test the agent's ability to find paths in the environment.
- Check Consistency: Validate that all the movements in the environment follow the rules and that programmed components work properly.

Results

Quantitative Results

For Qualitative results, we ran 3 different types of mazes as per difficulty to calculate metrics to comment upon the performance of the model
Below are the 3 different types of mazes with the CSV file of the maze and also the text file of the dynamic walls.

Simple Maze (Lowest Difficulty):

Maze:
S,0,1,0,E
1,0,1,0,1
0,0,0,0,1
1,0,0,0,1
1,1,0,G,0

Dynamic Walls:
(2,1): [(2,2)]

Moderate Maze (Mid Difficulty):

Maze:
S,0,0,1,1,1,E
1,0,1,1,0,0,0
1,0,0,0,0,0,0
1,G,0,1,0,0,G
0,0,0,0,0,1,1

Dynamic Walls:
(2,2):[(1,4),(3,4)]
(2,1):[(1,5)]

Complex Maze (Hard Difficulty):

Maze:
1,S,1,1,1,1,1,1,1,1,1
1,0,0,0,0,0,1,1,1,1,1
1,0,1,0,0,0,0,0,0,0,1
1,1,1,1,1,0,0,1,1,0,1
1,G,0,1,0,0,0,1,G,0,1
1,0,0,0,0,1,1,1,1,1,1
1,0,1,1,0,0,1,1,1,1,1
1,0,0,G,1,0,0,0,1,1,1
1,0,0,0,0,1,1,0,1,1,1

Dynamic Walls:
(2,5):(2,6)
(5,4):(5,2)

1,E,1,1,0,0,0,0,1,1,1

Paths of Each Maze:
Simple: [(0, 0), (0, 1), (1, 1), (2, 1), (3, 1), (3, 2), (3, 3), (2, 3), (1, 3), (0, 3), (0, 4)]
Moderate: [(0, 0), (0, 1), (1, 1), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (1, 6), (0, 6)]
Complex: [(0, 1), (1, 1), (1, 2), (1, 3), (2, 3), (2, 4), (2, 5), (3, 5), (4, 5), (4, 4), (5, 4), (6, 4), (6, 5), (7, 5), (7, 6), (7, 7), (8, 7), (9, 7), (9, 6), (9, 5), (9, 4), (8, 4), (8, 3), (8, 2), (8, 1), (9, 1)]
Metrics Calculation Formulas:
Success Rate = ((Success Count)/(Total runs in each episode)) *100
Average Steps = (Total Steps in Successful run)/(Success Count)
Average Rewards = (Total Rewards Across all Runs)/(Total runs in each episode)
Total Penalties = ∑Penalties Across All Runs
Dynamic Wall Triggers = ∑Triggers Across All Runs
Below is the table of the Qualitative Results for the Metrics:

Maze Type	Success Rate	Average Steps	Average Reward	Total Penalties	Dynamic Walls	Grievers
Simple	99.98%	38.293559	275.5115	12448	1	1
Moderate	99.93%	44.643150	325.8886	23604	2	2
Complex	96.62%	130.378597	680.0048	7776	3	3

Analysis of the Qualitative Results:

Simple Maze

The simple maze design featured a relatively straightforward path with minimal obstacles, one dynamic wall, and one griever interaction.

Performance Metrics:

- Success Rate: The nearly perfect success rate indicates that the Q-learning agent had minimal difficulty adapting to the simple maze structure.
- Average Steps: The agent required a low number of steps to navigate the maze due to its simplicity and lack of complex interactions with grievers or walls.
- Average Reward: The high reward score suggests efficient navigation, with the agent consistently choosing optimal actions.
- Total Penalties: The relatively low penalty count reflects limited opportunities for errors, as there were fewer grievers and dynamic wall triggers. The number looks high because the penalties were caught in the training period.

Method Observations:

- The high efficiency in this simple maze environment shows that the Q-Learning approach for a dynamic environment is optimal.

Moderate Maze

The moderate maze introduced more complexity, with 3 dynamic wall triggers, and 2 grievers. The layout also needs to get difficult as more dynamic walls are triggered.

Performance Metrics:

- Success Rate: Still very high, the slight drop in success rate compared to the simple maze reflects the increased difficulty and potential for error after having 2 grievers to go through with most dynamic walls in the 3 test cases.
- Average Steps: The agent took more steps, indicating a longer path due to the maze's increased complexity after the triggering of the dynamic wall.

- Average Reward: Despite the longer navigation, the agent maintained high average rewards, demonstrating effective learning.
- Total Penalties: The higher penalty count correlates with the increased number of grievers and dynamic wall interactions, which challenged the agent to find a better path.

Key Observations:

- The moderate maze highlighted the agent's capacity to adapt to medium-complexity scenarios, balancing reward maximization with error minimization.
- The increased penalties and steps show the trade-off between complexity and performance efficiency.

Complex Maze

The complex maze was the most challenging, featuring multiple valid paths, 3 grievers, and 2 dynamic walls that block multiple paths.

Performance Metrics:

- Success Rate: The drop in success rate compared to the other configurations reflects the challenges posed by the complex maze.
- Average Steps: The high step count indicates the maze's path and the need for frequent replanning due to griever movements and wall triggers.
- Average Reward: The reward was higher due to the longer navigation, but the efficiency slightly decreased due to the environment's complexity.
- Total Penalties: Interestingly, penalties were lower than the moderate maze, likely because the agent avoided direct interaction with grievers more effectively in this setup because of the rewards.

Key Observations:

- The complex maze tested the Q-learning algorithm's robustness and adaptability to its fullest extent.
- The agent's lower success rate and higher step count demonstrate the limitations of the current implementation in extremely challenging environments.
- The balance between exploration and exploitation becomes more critical in such scenarios, as shown by the higher variance in results.

Qualitative results

Agent Q-Learning Behaviour on the Maze for Pathfinding

Observation: The Q-learning agent demonstrated strong adaptability to dynamic environmental changes, such as moving grievers and dynamic wall triggers. It balanced exploration and exploitation, rerouting the path when necessary to avoid penalties.

Insights: The agent prioritized safety (avoiding grievers) and adjusted its path when dynamic walls were triggered, showcasing robust learning. However, minor hesitation in complex scenarios indicates potential for further optimization in real-time decision-making.

Dynamic Wall Interaction

Observation: Dynamic walls added unpredictability, requiring the agent to continuously recalculate paths. This was especially noticeable in the moderate maze with critical junctions affected by wall triggers.

Insights: The dynamic walls demonstrated the effectiveness of artificial life principles in creating a challenging yet solvable environment. The agent’s response highlighted its ability to adapt to changes in the environment while training.

Griever Rule-Based Movement

Observation: The left-right movement of grievers added complexity to the agent's pathfinding. The agent did not visit the griever's left and right movement cells when training for no collision with the grievers.

Insights: Griever rules effectively increased difficulty while maintaining solvability. The agent's success in avoiding grievers reflects optimal training, through the reward function.

Discussion

Limitations of the Work

- Agent Decision Handling and Training Time: In complex scenarios with multiple grievers and dynamic walls, the agent occasionally hesitated which indicates problems in training the model for complex maze pathfinding.

- Simplified Griever Movement: Grievers follow a left-right movement pattern, which may not reflect realistic NPC behavior in more dynamic or competitive environments.

- Reward Optimization: The reward system still seems to need optimization which could maybe solve the problem of long model training time and fewer episodes needed for pathfinding.

Directions for Future Work

- Enhanced Learning Strategies: Incorporate advanced techniques like Deep Q-Learning or Double Q-Learning to improve the agent's decision-making in highly dynamic and complex environments.

- Complex Griever Behaviors: As mentioned above griever behavior is very naive and maybe we can develop more rule-based or learning-based behaviors for grievers, such as patrol patterns or collaborative movement, to increase the challenge for the pathfinding.

Expanded Environmental Dynamics: Explore additional forms of dynamic elements, such as movable goals or checkpoints that add reward locations.

Implications of the Work

- Game Development: This project highlights the effectiveness of Q-learning in adaptive, partially observable environments, which is shown in many 2D games. This project shows many methods for rule-based NPCs and can inform game design to create intelligent and engaging agents.

- The approach can be adapted for real-world navigation problems, such as autonomous robotics or dynamic traffic systems, where environments are constantly changing and are partially observable.

Conclusion

The project successfully achieved its objectives by employing Q-learning, rule-based systems, and artificial life techniques to create a dynamic and adaptive Maze Runner simulation. The results demonstrate that the agent efficiently navigates the maze, handling both static and dynamic obstacles while avoiding grievers. High success rates across all configurations indicate that the methods were well-suited for the problem. While complex scenarios showed some performance drops, the adaptive strategies, including Q-learning and dynamic wall handling, proved effective in addressing the challenges. Overall, the methods met the objectives, providing optimal and efficient solutions for navigating dynamic, partially observable environments.

References

1. Samy Zaf. "Q-learning Maze Solver". GitHub Repository.
2. Mitchell Spryn. "Solving a Maze with Q-Learning". Medium Article.
3. Edvujic. "Q-Learning Maze Game". GitHub Repository.
4. Tan, M. "Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents". AAAI Conference.
5. "The Sims: Rule-Based AI NPC Behavior". ResearchGate Article.
6. Game AI Pro Series. "Behavior Trees and Rule-Based Systems". Gamasutra Article.
7. Reynolds, C. "Flocks, Herds, and Schools: A Distributed Behavioral Model". SIGGRAPH 1987.
8. "Trigger-Based Dynamic Environments in Game Design". GDC Vault Article.
9. Baldassarre, G., & Mirolli, M. "Intrinsically Motivated Learning in Natural and Artificial Systems". SpringerLink.