|  |  |  |
| --- | --- | --- |
| GitHub Repository | [alemaki/super\_mario\_bros\_AI](https://github.com/alemaki/super_mario_bros_AI) | |
|  | | |
| Project author | | |
| Name | | Faculty Number |
| Alexander Makedonski | | 8MI0800114 |

|  |  |
| --- | --- |
| Project name | Super Mario Bros AI Player |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| The goal of this project is to develop an AI agent capable of playing and completing levels of the classic game *Super Mario Bros*, including unseen levels, using Reinforcement Learning (RL) techniques. The OpenAI Gymnasium ([Gymnasium Documentation](https://gymnasium.farama.org/index.html)) framework and the gym-super-mario-bros ([gym-super-mario-bros · PyPI](https://pypi.org/project/gym-super-mario-bros/)) environment will serve as the foundation for training and evaluating the agent.  The project's core objectives include:   1. **Custom Reward Functions:** Experimenting with different reward strategies to determine their impact on training performance. For example:    * Rewarding the agent for progressing further and faster in the level.    * Rewarding based on points scored or specific in-game actions. 2. **Movement Set Comparison:** Comparing the agent's performance across the three distinct movement sets provided by the environment, exploring how the complexity of available actions affects the learning outcome. 3. **Generalization and Transfer Learning:** Training the AI on a subset of levels (~80%) and testing its ability to generalize to unseen levels (~20%), analyzing its performance and adaptability. 4. **Exploratory Research Component:**    * Investigating the impact of various RL model architectures and hyperparameters.    * For example: Q – learning, Deep Q-learning, Deep Q-Network, A2C algorithm (advantage critic), A3C algorithm (Asynchronous Advantage Actor Critic)    * More: investigation with Evolution algorithm.   This project concentrates on experimentation and analysis to draw insights into the effectiveness of different reward functions, movement sets, and training approaches for solving platformer games. The focus is on customizing the training process, interpreting results, and enhancing the agent's ability to generalize beyond its training data. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. ML/Agent System Description using PEAS [<https://aima.cs.berkeley.edu/4th-ed/pdfs/newchap02.pdf>] | | | | |
| **Agent name** | **Performance Measure** | **Environment** | **Actuators/Outputs** | **Sensors/Inputs** |
| * 1. **KoopaCrusher** | Maximize the distance traveled within each level.  Minimize the time taken to complete a level.  Maximize the points scored during gameplay.  Ensure the agent can complete unseen levels successfully, demonstrating generalization. | *Game Environment:* gym-super-mario-bros based on the NES *Super Mario Bros*.  Training Levels: ~80% of the game levels for reinforcement learning.  Testing Levels: ~20% of unseen game levels to evaluate generalization.  Movement Sets: Different predefined movement sets (simplified, complex, or full movement options) provided by the environment. | Action Commands: The agent generates discrete actions to control Mario’s behavior, including:   * Move Left/Right * Jump * Duck (if available) * Run/Sprint   The specific set of available actions depends on the chosen movement set. | Visual Input: The agent receives pixel-based observations (screen frames) of the game environment.  Game State: The environment provides game state information, such as:  Mario's position (X, Y coordinates)  Level progress  Points scored  Number of lives (if applicable) |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Test AI Generalization** | The trained agent is evaluated on unseen levels to measure its ability to generalize beyond training data. | User, **KoopaCrusher** |
| * 1. **Compare Movement Sets** | The agent's performance is analyzed across the three different movement sets available in the environment. | User, **KoopaCrusher** |
| * 1. **Optimize Reward Function** | Experiment with various reward functions (e.g., progress, points, time) to determine the most effective training strategy. | User, **KoopaCrusher** |
| * 1. **Evaluate Performance** | Measure the agent's performance using metrics such as level completion, distance traveled, time efficiency, and scores. | User, **KoopaCrusher** |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (REST/SSE/WebSocket Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Game Environment** | Provides access to the *Super Mario Bros* environment for training and testing the agent. | *gym\_super\_mario\_bros.make()* |