**Name: Musa Zaheer**

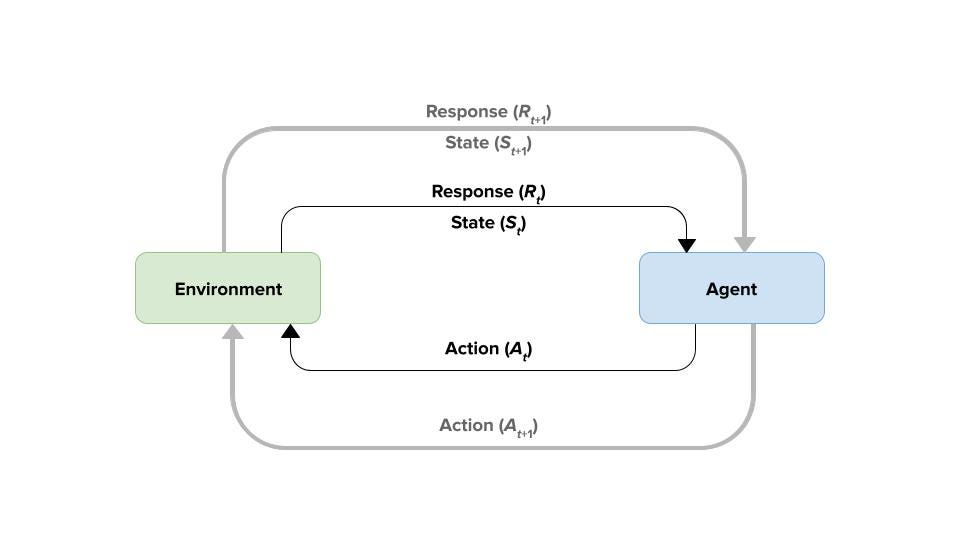
**Roll#BSCE21037**

**Project Module – 2**

**Comprehensive Documentation**

**Super Mario using Reinforcement learning**

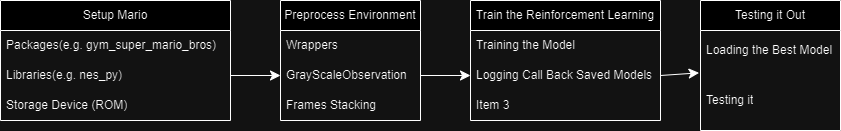
* **System Architecture:**



**Agent (Mario)**

**Environment (Game World)**

rld)



1. **Environment Setup:**

To set up the environment, utilize the **gym\_super\_mario\_bros** package for creating Gym environments tailored for Super Mario Bros. games. This package simplifies interaction with game ROMs and defines observation and action spaces. Additionally, leverage the **nes\_py** package for precise NES hardware emulation within Python environments. Obtain legal copies of Super Mario Bros. **game ROMs** and store them in an accessible directory for seamless integration with Python code.

1. **Reinforcement Learning Components:**

Reinforcement learning components form the core of the system. The agent learns to play Super Mario Bros. games by interacting with the environment, receiving observations, and selecting actions to maximize rewards. A **policy network**, typically a **CNN**, maps observations to action probabilities. Optionally, a value network estimates the value function for better decision-making. A memory buffer stores experiences for efficient training.

1. **Training Pipeline:**

The training pipeline involves collecting gameplay data, storing it in the memory buffer, and training the agent using algorithms like **DQN**. **Hyperparameter** tuning optimizes learning rates, exploration strategies, and network architectures for improved performance.

1. **Evaluation and Monitoring:**

Evaluation metrics measure the agent's performance, while **logging** **tracks** **training** **progress**. Visualizing training curves and behavior aids in understanding and improvement (Tensor Board).

1. **Testing and Validation:**

**Testing** out the model is the closing and the most important part of the project as the step ensures the successful end of the project.

1. **Documentation**:

A **Comprehensive documentation** for the project details and for future reference.

* **Integration Schedule:**

1. Environment Setup (1 - week) - 24/03/2024
2. Reinforcement Learning Components (2,3,4 - week) - 14/04/2024
3. Training Pipeline (4,5 - week) - 28/04/2024
4. Evaluation and Monitoring (6 - week) - 05/05/2024
5. Testing and Validation (7 - week) - 12/05/2024
6. Documentation (Ongoing , till the end of project) - ??/05/2024

The implementation of the "Reinforcement Learning Components" may require a significant amount of time due to the need for research and learning. Additionally, during this timeframe, there are scheduled events such as midterms, last Ashra of Ramadan/Ramzan, and Eid.

* **Finalized Algorithms:**

The finalized algorithms for our project include **Reinforcement Learning (RL)** techniques such as **Deep Q-Networks (DQN)** for training the agent to play Super Mario Bros. games effectively. DQN offers a robust framework for learning optimal policies from high-dimensional inputs such as game screens. Additionally, we will implement a **Policy Network (PN)**, typically based on **Convolutional Neural Networks (CNNs)**, to map observations to action probabilities, enhancing the agent's decision-making capabilities.

* **Dataset Collection:**

In this project’s case, there will be no supervisor or pre-labbelled dataset, as the project requires training data. So for that, the agent(Mario) will get its training from the data stored in the memory buffer which will be collected by the interaction with the game invironment which will include observation, actions, rewards and next state.