**Chrome Dino Game Player Using Reinforcement Learning**

**21K-3153 Bilal Ali**

**21k-3372 Ehaab Tariq**

**21K-3456 Ibrahim Khalil**

BCS-6A

**Introduction (Aim or Motivation)**: The aim of this project is to develop an AI program capable of playing the Chrome Dino Game using reinforcement learning techniques. The motivation behind this project stems from the desire to explore the capabilities of reinforcement learning in solving real-world problems, as well as the challenge of creating an autonomous agent capable of learning and improving its performance in a dynamic gaming environment.

**Background (Research & Project Selection):** Reinforcement learning has gained significant attention in recent years due to its success in various domains, including gaming. Projects such as DeepMind's AlphaGo and OpenAI's Dota 2 bot have demonstrated the potential of reinforcement learning algorithms in achieving superhuman performance. The decision to apply reinforcement learning to the Chrome Dino Game was influenced by the game's simple yet challenging nature, making it an ideal testbed for studying reinforcement learning techniques.

**Project Specification:**

The project aims to develop an AI agent that can learn to play the Chrome Dino Game by receiving feedback in the form of rewards or penalties based on its actions. The agent will utilize the Deep Q-Networks (DQN) reinforcement learning algorithm to learn an optimal policy for playing the game. The agent's performance will be evaluated based on its ability to achieve high scores and adapt to changes in the game environment.

**Problem Analysis:**

The Chrome Dino Game presents several challenges for an AI agent, including obstacle detection, timing jumps, and adapting to increasing game speed. The agent must learn to navigate these challenges while maximizing its score without colliding with obstacles or falling behind. The dynamic and stochastic nature of the game environment adds complexity to the learning process, requiring the agent to adapt its strategy in real-time.

**Solution Design (Project Detail, Functionality and features):**

**Project Detail:**

The project aims to develop an AI agent capable of playing the Chrome Dino Game autonomously. The agent will utilize reinforcement learning techniques, specifically the Deep Q-Network (DQN) algorithm, to learn and improve its gameplay strategy over time.

**Functionality and Features**:

1. **Environment Setup**:

- The project will create a custom Gym environment, `WebGame`, to interact with the Chrome Dino Game.

- The environment will capture game frames, define action and observation spaces, and implement methods for taking actions, resetting the environment, rendering, and checking game completion.

2. **Training the DQN Model**:

- The DQN model will be instantiated with a convolutional neural network (CNN) policy architecture suitable for processing game frames.

- Training will be conducted using the Stable Baselines3 library, with the model learning to play the game by observing states and taking actions to maximize rewards.

- A custom training callback will save the model periodically to allow for later analysis or continuation of training.

3. **Testing the Trained Model:**

- After training, the performance of the trained DQN model will be evaluated by running multiple episodes in the environment.

- The trained model will make predictions to choose actions based on observed states, and its performance will be measured by the total score achieved in each episode.

4. **Real-time Adaptation:**

- The dynamic and stochastic nature of the game environment requires the agent to adapt its strategy in real-time.

- The agent will continuously learn and update its policy based on new experiences encountered during gameplay, enabling it to improve its performance over successive episodes.

**Solution Thought Process:**

The WebGame class is defined, which represents a custom Gym environment for interacting with the Chrome Dino Game.

This environment captures frames of the game, screenshots at certain places of the screen, defines action and observation spaces, and implements methods for taking actions, resetting the environment, rendering, and checking if the game is done.

After defining the environment, it's tested by initializing an instance of WebGame and visualizing observations and game completion using sample code.

The DQN model is instantiated with the 'CnnPolicy' architecture and trained using the learn() method from Stable Baselines3.

Overall, the solution design encompasses setting up the environment, training the DQN model, testing its performance, adapting to real-time challenges, visualizing training progress, and implementing error handling and optimization strategies to develop an AI agent capable of mastering the Chrome Dino Game.

**Implementation & Testing**

The solution will be implemented using Python and machine learning libraries such as Pytorch and StableBaselines3. The agent will be trained using a combination of simulated gameplay and reinforcement learning techniques. Testing will involve evaluating the agent's performance on unseen game scenarios and assessing its ability to generalize to new environments.

**Project Breakdown Structure (Workload distribution with timeline):**

This project was completed in 5 phases, which encompassed 5 weeks.

Week 1: Define roles and brainstorm ideas. Look for online resources and find the strongest machine capable of running algorithms among us 3.

Week 2: Come up with a method of extracting data from a real time game, feeding it into the agent and noting progress.

Week 3: Configuring reward systems, and starting to train the model.

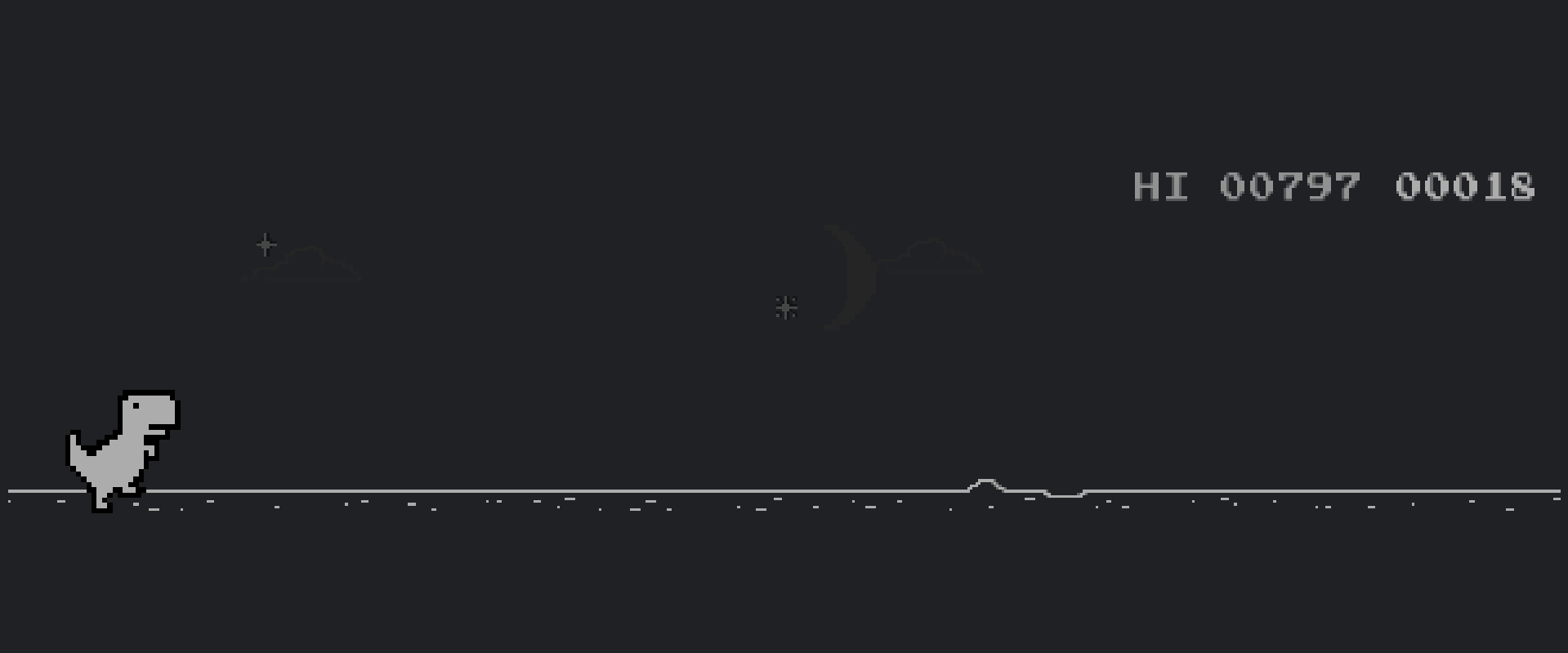
Week 4: Fixing bugs and errors that rose during training of the model.

Week 5: Finishing touches, final training and testing.

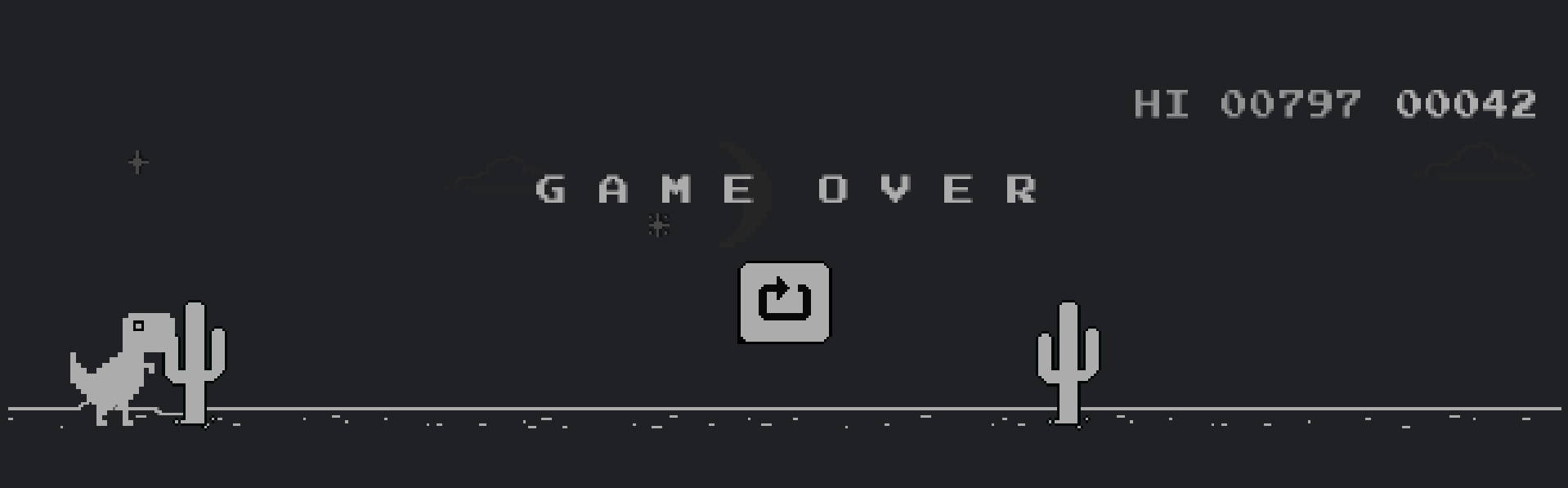
All work was divided equally among the group members.

**Results (Outputs screenshots):**

Before Training

****

After Training:

****

**Conclusion (Summary & Discussion):**

In conclusion, this project is an AI agent capable of playing the Chrome Dino Game using reinforcement learning techniques. By leveraging reinforcement learning algorithms, the agent navigates through the game environment, avoids obstacles, and maximizes its score. The project has the potential to contribute to the advancement of reinforcement learning research and showcase the capabilities of AI in gaming applications.