**National University of Computer & Emerging Sciences**

**Karachi Campus**



**AI-Based Checkers Game Using Reinforcement Learning**

**Project Proposal**

**Artificial Intelligence**

**Section: 6-E**

**Group Members:**

**22k-4409 Rida Qazi**

**22k-4552 Mashal Jawed**

**22k-4172 Abdul Wasey**

Project Proposal

* **Introduction**

This project aims to develop an **AI-powered Checkers game** that utilizes **Reinforcement Learning (Q-Learning)** to improve its gameplay over time. The AI will be able to learn from self-play, adapt to different strategies, and compete against human players. A **graphical user interface (GUI)** will be implemented using pygame to allow interactive gameplay.

* **Existing System**

Traditional Checkers AI implementations rely on **rule-based approaches** or **Minimax with Alpha-Beta Pruning**, which requires hand-crafted heuristics. These methods do not allow the AI to **learn and improve over time**. Most existing Checkers engines follow predefined strategies, limiting adaptability against dynamic human players.

* **Problem Statement**

The primary limitation of existing Checkers AI systems is their **lack of learning capabilities**. Rule-based approaches do not adapt, and Minimax requires significant computational power for deep search trees. Our project introduces an **adaptive AI that learns through self-play**, enabling it to enhance its strategies dynamically without human intervention.

* **Proposed Solution**

We propose an **AI-driven Checkers system** where the AI improves over time using **Reinforcement Learning techniques**. Initially, **Q-learning** will be implemented to develop a basic AI model. The system will also include **game visualization** using pygame, allowing human players to compete against the AI.

* **Salient Features**

✔ **AI that learns from self-play** using Q-learning  
✔ **Graphical User Interface (GUI)** for player interaction  
✔ **Multiple difficulty levels**, with AI adapting over time  
✔ **Support for AI vs AI and Human vs AI modes**  
✔ **Dynamic reinforcement learning** for improving AI strategy

* **Tools & Technologies**

🖥 **Programming Language:** Python  
📚 **Libraries:** pygame, numpy, TensorFlow / PyTorch, matplotlib  
🛠 **Frameworks:** OpenAI Gym (for reinforcement learning environment setup)  
🖥 **Operating System:** Windows / Linux  
📊 **Machine Learning Techniques:** Q-learning