# 1 Introduction

This project involves developing an Uno game server where multiple players can play Uno online using a Python application with a GUI interface and track their scores.

By implementing a multiplayer online Uno game server, we can provide an engaging gaming platform that attracts players to interact online and enjoy the game of Uno.

We will Utilize Python programming language to implement Uno game rules and logic and create a server using networking techniques to allow multiple players to connect via clients and play the game.

# 2 Overview of Existing Work in the Field

* **Uno Game:** Uno is a classic card game where players match the color, number, or special function of the current card to get rid of their cards as quickly as possible.
* **GUI Development in Python:** Python offers various libraries like Tkinter, Pygame, etc., for creating graphical user interfaces and building interactive applications.

# 3 Work Plan

## Phase 1 (1 week): Design the game logic

* Study Uno game rules and mechanics.
* Design and implement classes to represent the deck, cards, players, and game state.
* Develop methods for shuffling and dealing cards, playing cards, drawing cards, etc.
* Implement the game logic for determining valid moves, checking win conditions, etc.

## Phase 2 (1 week): Implement a single player game

* Create a GUI interface using a Python library like Tkinter for a single-player Uno game.
* Display the game state, player’s hand, and available actions.
* Implement basic functionalities of a single-player game, such as playing cards, drawing cards, etc.
* Implement the necessary event handling and update the game state accordingly.

## Phase 3 (1 week):

* Research Python networking programming, understand how to establish communication between a server and clients.
* Design and implement a multiplayer online server for Uno game, supporting multiple client connections and game interactions.
* Define message formats and communication protocols between the game server and clients.

## Phase 4 (1 week):

* Update the game logic to accommodate multiple players and manage their interactions.
* Implement basic user interface and operations, such as joining the game, starting the game, viewing scores, etc.
* Test the communication between the server and clients, address any potential issues and risks.

## Phase 5 (1 week):

* Enhance the GUI, allowing it to handle interactions with multiple players.
* Add visual effects and animations to enhance the user experience.
* Perform comprehensive testing, optimize the code, and fix any vulnerabilities.

## Stretch Goals (if additional time permits):

* Implement AI players, enabling single-player mode to play against computer opponents.
* Add sound effects and background music to enhance the game’s entertainment value.
* Design and implement game recording and playback features, allowing players to review and share their gameplay.

## Project Evaluation:

* Conduct unit testing and integration testing throughout the development process to ensure the correctness of game logic and stability of the server-client communication.
* Conduct user testing, collect user feedback, and make necessary modifications and improvements based on the feedback.
* Perform regular code reviews to ensure code quality and maintainability.