**COMP4003 / COMP4031 / COMP4026 Project Plan**

**Student:**

**Student ID:**

**Course:**

**1. Project title.**

Development of an Interactive Online UNO Game Server.

**2. Statement of the research problem.**

The project aims to develop an interactive UNO game server that allows multiple players to play the game online. The research problem addressed is the need for a reliable and efficient platform that enables players to participate in UNO games remotely, while facilitating real-time communication and score tracking. This research problem is important as it meets the growing demand for online multiplayer gaming experiences and explores the challenges associated with implementing a multiplayer UNO game server.

**3. Related work.**

While several online gaming platforms offer UNO as a multiplayer game option, this project builds upon existing work by focusing on the development of a customized UNO game server with a specific emphasis on real-time multiplayer interaction and score management. Existing solutions often lack customizability or the ability to integrate seamlessly with other systems. The approach taken in this project differentiates by implementing a server-client architecture using Python's socket programming capabilities and developing a graphical user interface (GUI) to enhance the gaming experience.

**4. Methodology.**

The methodology for this project includes technology development, algorithm design, and system implementation. The following steps will be followed:

Server Architecture and Setup: Design and implement the server architecture using Python's socket programming, allowing multiple clients to connect and communicate with the server.

Game Mechanics and Rules Implementation: Develop classes and methods to represent UNO game components, including the deck, cards, players, and game rules. Implement the necessary algorithms to handle card plays, turns, and win conditions.

Server-Client Communication: Design and implement a communication protocol that ensures smooth and secure data exchange between the server and connected clients, enabling real-time gameplay.

Player Management and Score Tracking: Implement mechanisms for player registration, turn-based gameplay, and real-time score tracking. Manage player interactions, including joining and leaving game sessions, ensuring fair gameplay.

Graphical User Interface (GUI) Development: Select a suitable Python GUI library and create an intuitive and visually appealing GUI that displays game elements, allows players to interact with the game, and provides a seamless user experience.

Testing and Evaluation: Conduct rigorous testing to validate the server's stability, responsiveness, and scalability. Evaluate the gameplay experience, identify and fix bugs or issues, and optimize performance.

Documentation and Reporting: Prepare comprehensive documentation that includes project specifications, design choices, implementation details, and instructions for setting up and running the UNO game server. Write a project report that summarizes the achieved goals, challenges faced, and lessons learned.

**5. Programme of work.**

**Work Package 1 (WP1) - Server Architecture and Setup**

Activity: Design and implement the server architecture using Python's socket programming.

Milestone: Successful establishment of socket connections and basic server functionality.

**Work Package 2 (WP2) - Game Mechanics and Rules Implementation**

Activity: Develop classes and methods to represent UNO game components and implement game rules.

Milestone: Functional single-player UNO game with proper game flow.

**Work Package 3 (WP3) - Server-Client Communication**

Activity: Design and implement a communication protocol for seamless communication between the server and clients.

Milestone: Successful communication between the server and multiple clients.

**Work Package 4 (WP4) - Player Management and Score Tracking**

Activity: Implement mechanisms for player registration, turn-based gameplay, and real-time score tracking.

Milestone: Proper management of game sessions and accurate score tracking.

**Work Package 5 (WP5) - Graphical User Interface (GUI) Development**

Activity: Choose a suitable Python GUI library and design an interactive and visually appealing game interface.

Milestone: Functional GUI displaying game elements and allowing player actions.

**Work Package 6 (WP6) - Testing and Evaluation**

Activity: Conduct comprehensive testing to ensure stability, responsiveness, and scalability of the game server.

Milestone: Stable and well-tested game server with satisfactory performance.

**Work Package 7 (WP7) - Documentation and Reporting**

Activity: Prepare comprehensive documentation, including project specifications, design choices, and implementation details.

Milestone: Submission of the project report and sharing access to the version control repository.

**6. Time 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.

**Enter word count out of /1000 words**

**Assessment criteria**

Is there a clear statement of what the project is about?

Is there an explanation of why this problem needs to be addressed / solved?

Is the novelty of the proposed research substantiated with reference to related work?

Is there a clear statement of how the problem is going to be addressed / solved?

Are the proposed methods for addressing / solving the problem appropriate?

Is there a clear statement of how the results are going to be evaluated?

Is the proposed evaluation appropriate?

Is there a breakdown of the research into work packages?

Is this breakdown appropriate?

Is there a time plan?

Is the time plan realistic?

*Affirmative answers to each of these questions will result in a first class mark.* *Markers should briefly describe how the project plan diverges from these desired outcomes to justify their mark.*