Project SOW

Project Title:

Tic-Tac-Toe Multiplayer Game

Team:

Emmanuel Koomson

Project Objective:

The objective of this project is to develop a Python-based multiplayer Tic-Tac-Toe game using a client-server architecture. This project will demonstrate fundamental concepts of socket programming and networking by allowing multiple players to connect and play the game in real time.

Scope:

Inclusions:

- Develop a server to manage client connections and handle game state.
- Create a client that connects to the server, allowing players to take turns and display game progress.
- Implement game logic for Tic-Tac-Toe, including win conditions, draw conditions, and error handling.
- Design a simple command-line interface for players to input moves and view game status.

Exclusions:

- No graphical user interface (GUI) will be included.
- No player authentication or persistent storage of game history.
- No network encryption or advanced security features.

Deliverables:

- Python scripts for both the server and client.
- Documentation describing the server-client architecture, game logic, and setup instructions.
- A GitHub repository containing the code and README file.

Timeline:

Key Milestones:

1. **Sprint 0** (Sept 08 - Sept 22): Form teams, set up tools, and submit SOW.
2. **Sprint 1** (Sept 22 - Oct 06): Implement basic socket programming for TCP client-server communication.
3. **Sprint 2** (Oct 06 - Oct 20): Develop game message protocol and manage client connections.
4. **Sprint 3** (Oct 20 - Nov 03): Add multi-player functionality and synchronize game state.
5. **Sprint 4** (Nov 03 - Nov 17): Implement full gameplay logic and win/draw detection.
6. **Sprint 5** (Nov 17 - Dec 06): Add error handling, perform testing, and submit the final project.

Task Breakdown:

- Server setup and socket programming: 10 hours
- Client implementation and game logic: 15 hours
- Error handling and testing: 8 hours
- Documentation and README: 3 hours

Technical Requirements:

Hardware:

- No special hardware required; standard laptops with internet connectivity.

Software:

- Python 3.x, Git, and an IDE (VS Code or PyCharm).
- Libraries: Python socket and threading.

Assumptions:

- All players will have a stable internet connection.
- The server will be hosted on a machine with adequate uptime and network access.

Roles and Responsibilities:

- **Emmanuel**: Responsible for the full development of both client and server, as well as testing and documentation.

Communication Plan:

- Weekly updates will be shared on the project's GitHub repository.
- All communications regarding issues or questions will take place via email or instant messaging.

Additional Notes:

- The project will focus on demonstrating core networking principles rather than providing an advanced gaming experience.