Assignment Developing a Simple Networked Game Using Unity's NGO

Due 17 January 2025 23:59

Instructions

Objective: The goal of this assignment is to create a simple networked game using Unity's Netcode for GameObjects (NGO). The project will involve setting up a client-server architecture, implementing basic synchronization, and ensuring stable replication of game objects.

Requirements:

1. **Game Design:**
   - Create a simple multiplayer game. Suggestions include:
     - A basic top-down shooter.
     - A simple racing game.
     - A basic platformer.
   - The game should support at least two players (one host and one client).
2. **Networked Game Features:**
   - Implement a client-server model.
   - Ensure player movement is synchronized across the network.
   - Synchronize at least one other game object (e.g., projectiles, NPCs).
   - Implement in game communication (e.g., chat messages, emotes).
   - Implement server authority to handle game state and synchronization.

Steps:

1. **Project Setup:**
   - Create a new Unity project.
   - Add the Netcode for GameObjects package via the Package Manager.
   - Set up a basic scene with a player prefab and necessary game objects.
2. **Network Manager:**
   - Add a NetworkManager to your scene.
   - Configure the NetworkManager to handle player connections and object spawning.

3. **Player Movement Synchronization:**
    - Create a player prefab with a NetworkObject component.
    - Implement a basic movement script that uses Unity's Input system.
    - Synchronize player movement across the network using NetworkTransform or a custom script.
4. **Game Logic:**
    - Add additional game logic such as shooting, scoring, or checkpoints.
    - Ensure all relevant actions are synchronized across the network.
5. **Server Authority:**
    - Implement server authority to manage game state.
    - Ensure all critical game logic (e.g., scoring, player death) is handled on the server.
6. **Testing and Optimization:**
    - Test the game with multiple players.
    - Optimize the network code to ensure smooth performance.
    - Handle common network issues such as latency and packet loss.
7. **Documentation:**
    - Document the development process.
    - Include explanations of how synchronization, server authority, and optimizations were implemented.

Deliverables:

1. A GitHub link with the complete Unity project.
2. A built executable of the game for testing.
3. A written report (2-4 pages) detailing:
    - Overview of the game design.
    - Implementation details of the network features.
    - Challenges faced and solutions implemented.
    - A reflection on the learning experience.

Evaluation Criteria:

- **Functionality:**
    - The game runs smoothly with multiple players.
    - Player movements and game objects are synchronized correctly.
- **Network Implementation:**
    - Proper use of Unity's Netcode for GameObjects.
    - Effective client-server communication and server authority implementation.
- **Optimization:**
    - Efficient network message handling.
    - Minimal latency and smooth gameplay.
- **Documentation:**
    - Clear and comprehensive report including.
        - Struggles and issues.
        - Reflections on the development process.

Submission:

Submit a link to your GitHub repository with the Unity project folder, built executable, and the written report.