# **Game Design Introduction**

This is a multiplayer adaptation of the classic arcade game known as “Asteroids”. The objective is for players to navigate their spaceship, destroy asteroids, and compete against other players in a synchronised multiplayer environment. The game ensures that gameplay remains engaging.

**Game Mechanics**

Players can rotate their spaceship with the left and right arrow keys and propel it forward. Players will also be able to shoot bullets to destroy asteroids and gain points to go towards the win condition of having the most points of 2500.

Asteroids move randomly across the game space. When hit by bullets, these asteroids will disappear.

Up to four players can connect over a Local Area Network (LAN) using a User Datagram Protocol (UDP) based network system.

All players should see the same asteroid movements, spaceship actions, and bullet trajectories, ensuring a fair and consistent gameplay experience.

**Objectives**

Players score points by destroying asteroids. The game session ends when a win condition is met, when a player has obtained a total of 2500 points.

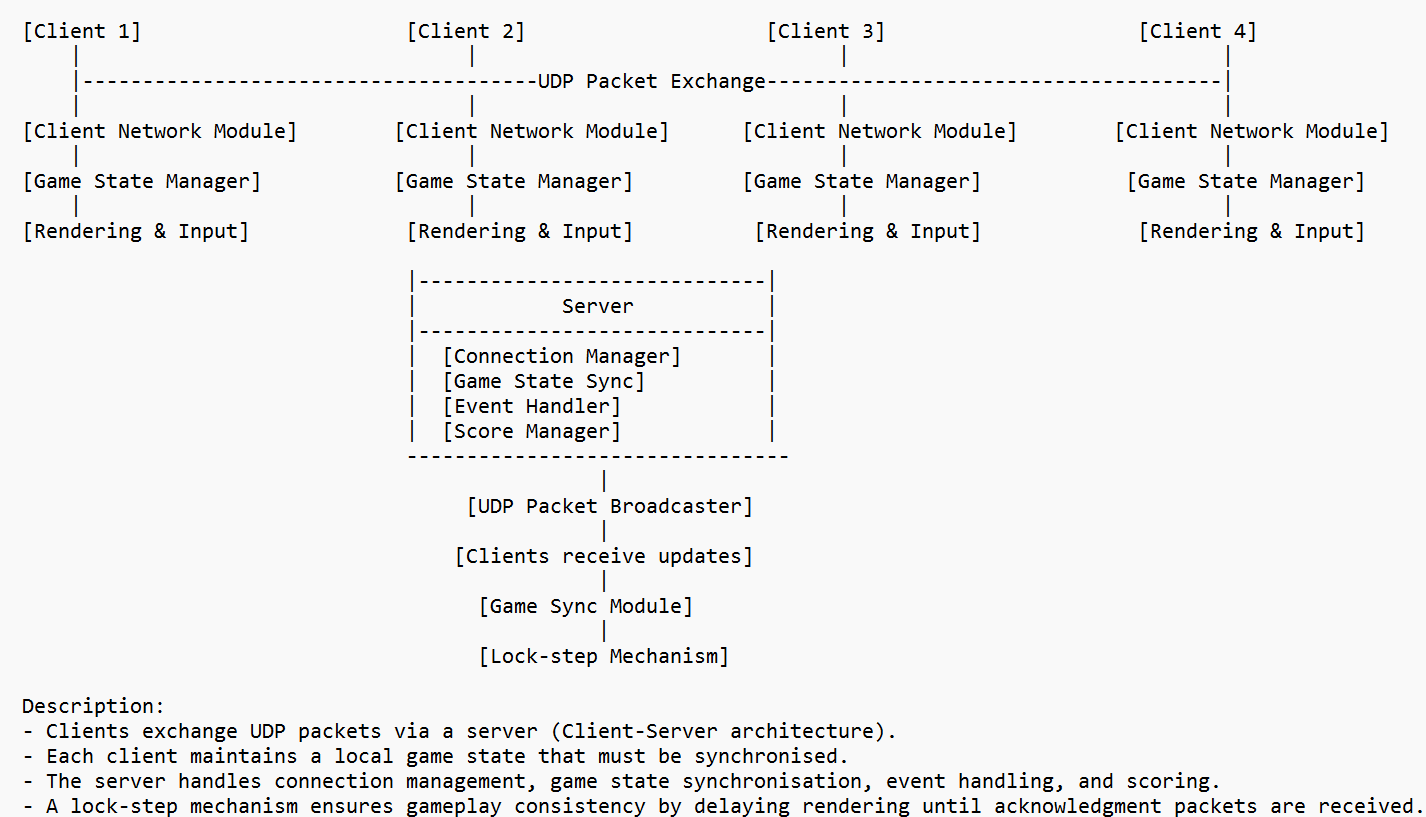
Playing as multiplayer requires players to connect to a server, which synchronises game events and ensures consistency in movement and actions.

**Challenges and Considerations**

The program will have to ensure that asteroids, bullets, and ships move identically across all clients while minimising delays and ensuring actions like shooting and movement are responsive as well as preventing desynchronisation issues that could give players an unfair advantage.

# **Implementation Block Diagram and Introduction**

**Block Diagram**



**Implementation Introduction**

The game’s network model follows a Client-Server architecture, where the server maintains the authoritative game state, synchronising all connected clients. Upon successful connection between the server and each client, the server assigns each client a unique Player ID. Each client sends player input data to the server, receives updates, and renders the game accordingly. Data sent includes the ship’s positions, rotations and bullet data. A UDP protocol is used for communication due to its low-latency advantages over Transmission Control Protocol (TCP).

The game flow follows an initialisation where players connect to the server, which assigns a unique identifier. The server then constantly updates the inputs sent from the clients and updates the game state. The clients confirm important actions such as asteroids being destroyed to prevent inconsistencies. Once the win condition is met, the session ends.

**Key Features**

Players' ship positions and rotations are kept in sync with minimal lag between the server and each of the clients.

Asteroids are generated in the same positions across all clients, ensuring a fair and consistent gameplay.

Firing and collision detection are confirmed by the server to ensure accuracy.

Each client updates based on server validation, ensuring fairness.

# **Individual Contribution**

**Member 1**

Full name: Ho Jing Rui

SIT ID: 2301375

DigiPen ID: jingrui.ho

Jing Rui used his strengths in the comprehension of the code and concepts to write the backbone of the project. He pieced together and amended the code used in Assignment 2 and Assignment 3 to meet the requirements of this project. He also contributed to the administrative documents required such as the ReadMe text document and this Design Report.

**Member 2**

Full name: Saminathan Aaron Nicholas

SIT ID: 2301206

DigiPen ID: s.aaronnicholas

Aaron tested the functionalities of the code written with the different use cases as specified in this Assignment’s specifications and helped with the debugging of the code to meet the requirements of the project. He mainly contributed to the administrative documents required such as the ReadMe text document and this Design Report.

**Member 3**

Full name: Jay Lim Jun Xiang

SIT ID: 2301335

DigiPen ID: jayjunxiang.lim

Jay helped with debugging and making the code more robust. He also contributed to the administrative documents required such as the ReadMe text document and this Design Report.