CSD2161/CSD2160/CS260/MET3302 2025 Spring *Assignment 4*

Program RTIS/METS Name _____

DigiPen Institute of Technology

Due Date:	3rd April 2025
Topics covered:	Network for games
Grouping :	Group project: group size: 1 - 6
Deliverable:	Moodle submission: A compilable Visual Studio solution (Release mode, version 2022), a README and a design document in a zipped file.
Objectives:	To demonstrate abilities to add networking to a game engine, with synchronization of passive and active objects in a game.

This assignment requires an implement of a ***LAN network engine over UDP***.

1 Specifications

You will submit a Visual Studio Project (VS 2022) for a game called *Spaceships*, akin to the classic asteroids game. The main task is to change the game into a multi-player game played over LAN. The requirements are:

1. YOU MUST USE UDP. (Any usage of TCP connection for game data transfer means 0 for the entire assignment for the whole team.)
2. 4 players to play the game over a LAN connection. (Failure to meet this requirement definitely no full credit.)
3. The asteroids creation, destruction and movement on all screens should be (somewhat) synchronized. The reason why "somewhat" is put in there is that full synchronization is hard to achieve in practice. However, there should not be huge discrepancies between the screens that affect the gameplay.
4. The movement of the ships, again, should be in sync across the screens. Again, there is a subjective element to this - but in my experience, groups have been able to achieve synchronization to a high degree.
5. There must be some win condition and at the end of each round I should be able to see the scores of each player so that I know who won and how I did. Needless to say, the score for each player should be the consistent across all screens.
6. Single player must still work (you don't need to enhance it at all from what it currently does however).
7. Some kind of a rough lock-step for synchronization. The round-trip time of the LAN in the labs should be around 2 - 4 ms i.e., get the clients to perform some acknowledgments and do not render the event until the Ack has been received.

2 Implementation Hints

1. Game lobby is not needed.

2. Each player should be able to read the server location from a config file and have the server start the game whenever it has enough players.
   Alternatively, you may try broadcast on the LAN to find the server so that you do not require any config files.

3. Each player should be able to launch spaceships.

4. Although Client-Server architecture is strongly recommended, either Client-Server or Peer-To-Peer are accepted.

3 Rubrics

It should be noted that due to the nature of the assignment, the grading cannot be 100% objective. Overall, the aim is to provide a satisfactory gameplay experience over the network. Here is the breakdown of the scoring:

1. Synchronization of Asteroids
   (a) Creation - this should be synced very precisely i.e., all clients should be creating asteroids at the same spot at almost the same time.
   (b) Movement - the asteroids should be moving in sync, especially given their passive and deterministic behaviour.
   (c) Destruction - this must be synced. (Hint: Client must ack this event!).

2. Synchronization of ships
   (a) Movement should be synced, especially given that the ship has essentially only 2 degrees of freedom (Rotate and go forward).
   (b) Firing - a firing event should be pretty much synced, though a little delay is tolerable. (i.e., the firing cannot disappear, but it should happen!)
   (c) Bullets - again, should be very much synced given that this is a deterministic behavior and a passive object.

3. Scoring system
   - Each player's score should be dynamically displayed.
   - The highest 5 player's names, scores and play date/time should be saved and display at the end of game.

4. Durable communication
   - Minor communication disruption can be introduced.
   - If a player lost connection, other players should not be affected.
   - If a player lost connection, and resumes the connection back, this player should be allowed to continue the game.

5. Fun Factor
    - Some win condition MUST be implemented i.e., we need a game!
6. A README: indicates how to run.
7. A design report (named `Assignment_4_Design_Report_SIT-ID.pdf`) with the following contents:
    (a) Game design introduction
    (b) Implementation block diagram and introduction.
    (c) Individual contribution: including all members' full names, SIT IDs, and DigiPen IDs, and contribution percentage.