Dominic McNeill

2mcned00@solent.ac.uk

Abstract

A networked FPS game that must synchronize two clients with minimal latency whist having some game features such as buffs and debuffs.

Network game programming   
cgp503

Dominic McNeill - 102061800

Contents

[Introduction / Game Idea: 2](#_Toc193663865)

[Game Actions: 3](#_Toc193663866)

[Main Menu: 3](#_Toc193663867)

[In Game Menu: 3](#_Toc193663868)

[In Game Actions: 4](#_Toc193663869)

[High Overview of Main Systems: 5](#_Toc193663870)

[Main Menu System: 5](#_Toc193663871)

[Lobby Manager: 5](#_Toc193663872)

[Game Manager: 5](#_Toc193663873)

[Round Manager: 6](#_Toc193663874)

[Player Manager: 6](#_Toc193663875)

[Flow charts and UML: 7](#_Toc193663876)

[Refs: 8](#_Toc193663877)

# Introduction / Game Idea:

A simple arena shooter that is in first person, it’s a one vs one game mode. Both players get one simple AR rifle that will be full auto, the weapons will be simple recasting weapons. They will both start with limited ammo.

Around the arena there will be items the players can pick up. One will give ammo; one will heal; the third one will give double damage, and the fourth gives another gas grenade. A maximum of 4 items must be present and between 6 to 12 spawn points.

The arena will be asymmetrical.

The players will have 2 hitboxes, one for the head that will kill the player instantly and the other one is the body and will deal damage from the attack like normal. (normal weapon damage unless there is a buff).

The players will also get a set amount of gas grenades, if any of the players enters the gas, they will get limited vision for a period and audio making it harder to hear. (like a concussion grenade).

Particles and SFX should be included.

This is a round system

Best of 5 round system. (rainbow six siege quick match), first to 3 wins. A timer of 5 minutes (300 seconds) is set per round. If the timer runs out the health of both are compared and the one with the most health wins. In case of a draw, no one gets a point (point as in round).

# Game Actions:

## Main Menu:

* Load to UI.
* Lobby menu
  + List all open games
  + Click to join games
  + Favourite lobbies
  + Direct connection
* Settings
  + Audio
  + Sensitivity
* Quit Game

## In Game Menu:

* Tab menu
  + Kick player for host only.
  + List kills and deaths, maybe KD too and score?
  + Player names
* “Pause” menu (“you can’t pause a multiplayer game”)
  + Close menu aka resume.
  + Settings?
    - Audio
    - Sensitivity
  + Leave match
  + Quit
* Hud
  + Cross hair
  + Ammo counter and grenade counter
  + Health bar
  + Timer displayed
  + Points displayed
  + Current round displayed

## In Game Actions:

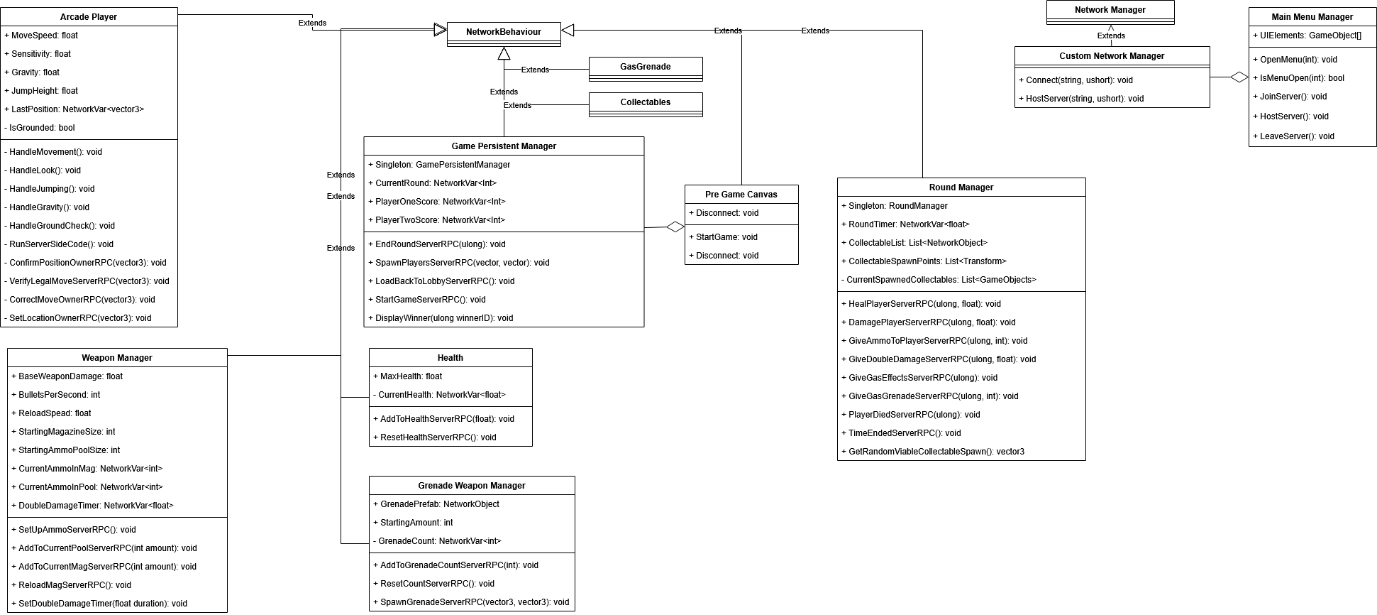
* Shoot (recast hit)
  + Detect hitbox, deal expected damage.
* Take damage
* Collect buff
  + Heal player
  + Grant double damage.
  + Give ammo
* Throw physics grenade. (server owned, need to be syncs to all clients accurately). (R6 ragdoll moments)
  + Gas cloud. Deals damage to everyone, can just be server owned environmental hazard.
* Handle player death.
* Round system.

# Flow charts and UML:

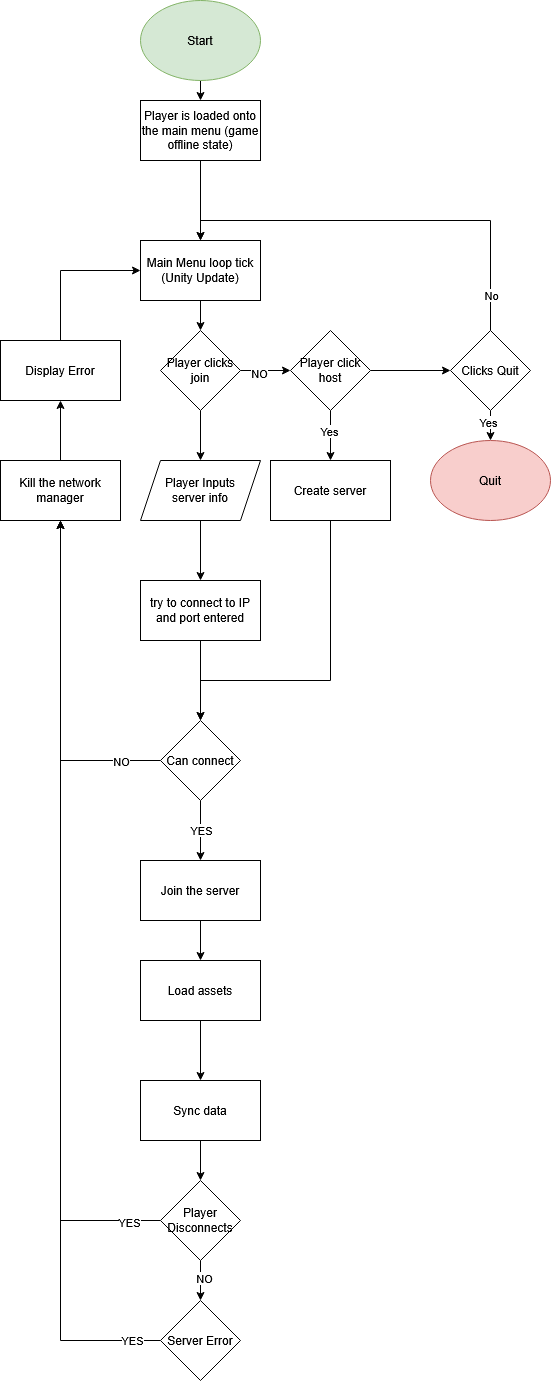
<https://viewer.diagrams.net/?tags=%7B%7D&lightbox=1&highlight=0000ff&edit=_blank&layers=1&nav=1&title=Networking%20Flow%20and%20UML.drawio&dark=auto#Uhttps%3A%2F%2Fdrive.google.com%2Fuc%3Fid%3D1696fGM5Qh_1YszE9zf2ONVLErNG_NzK_%26export%3Ddownload>

A link to the diagrams above.

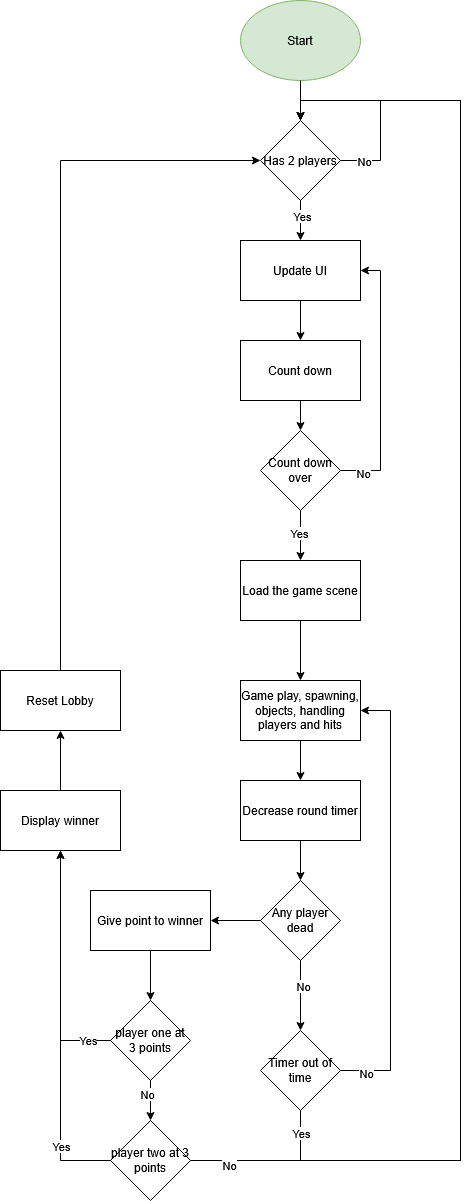
## UML diagram:



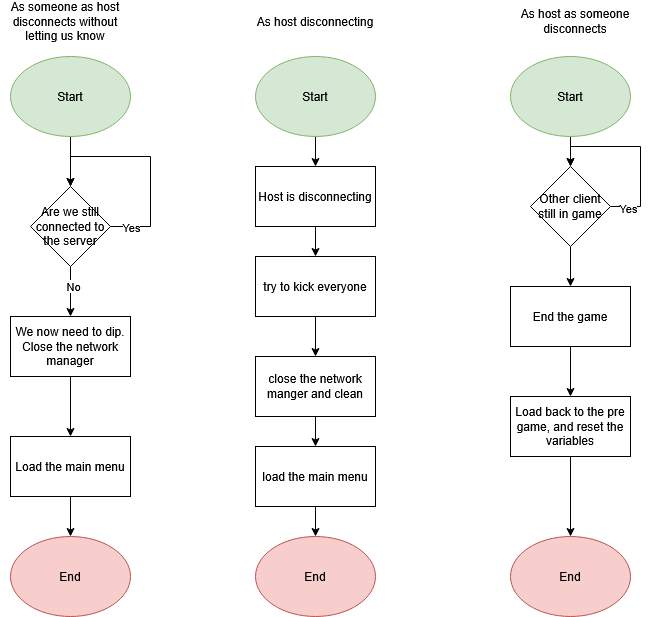
## Flow of main menu:



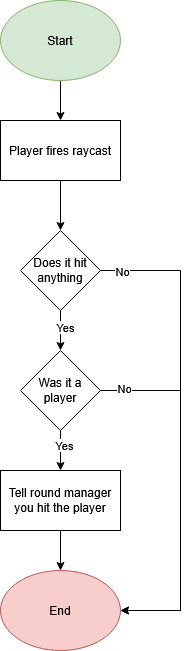
## Flow of pregame and game:



## Flow of disconnect handling:



## Flow of hit reg:



# Reflection:

At the beginning designing the systems, I had limited practical knowledge of NGO. I have experience with Photon Pun in the past but that is a very different architecture. I feel like I should had put more faith into the designs of the code instead of starting with a very high-level overview.

The coding standard in the project is subpar to my current standards. I will say adjusting from working from studio constantly did make it difficult adjust but I focused more on the implementation than the look of the code. I prioritised the functionality when I should have spent equal time on all aspects of the code.

The time frame of working on the project has forced me into a crush to get the systems to work and operated fully. I have had snags where I planned to do something would not work and required a creative solution.

I feel like that code is secure running more on the server than on the clients reducing how much control the clients have, but there are aspects that can be improved upon to make it more secure.

I would have liked to have more authoritative player network transform, but I faced issues where I couldn’t get around and used the base version and implemented syncing with the server. I also wanted to use a custom extrapolation algorithm to help smooth motions between clients too.

Now that I have a good foundation understanding of Unity NGO, I feel more confident on designing and creating network games now. With the newfound tools and the freedom that NGO provides compared to Photon is very freeing.

I do want to point out burnout has affected my capabilities on the project, and I barely pushed through this at a cost. Taking proper breaks is very important and I will take this summer to take a full break.

Overall, a fun and interesting project, I managed to achieve the goals I have set and some extra (except animations as they broke). It was fun playing this game with a friend and will use what I have learned here in the future.

DM signing off. ~

# Refs:

<https://en.wikipedia.org/wiki/Local_area_network>

<https://docs.unity.com/ugs/en-us/manual/lobby/manual/unity-lobby-service>