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.

# High Overview of Main Systems:

The systems that will go into the game and mange something. I will not include small systems such as looking around but whole systems such as player movement and look. I won’t be diving into detail of code here, just outlining what need to do what.

## Main Menu System:

This handles all the UI stuff on the main menu, this will be broken down into smaller modules but overall will be called the main menu system. This will handle creating and loading into lobbies (the rooms the players sit in), adjusting settings and closing the game.

Another “sub” system will be the lobby collection system; this needs to get all the lobbies and keep them up to date on the client.

([Unity doc about their lobby system](https://docs.unity.com/ugs/en-us/manual/lobby/manual/unity-lobby-service)) We will be using the unity lobby package to assist us for finding other lobbies so players don’t need to find workaround or post sensitive information online to play with each other. (not everyone has a secure network)

We also need to think about LAN ([local area network](https://en.wikipedia.org/wiki/Local_area_network)) too, we don’t always have online access. (thanking Solent for this)

## Lobby Manager:

The lobby manager handles the room. This will handle any game related settings such as game time, max round and map. (room settings are not guaranteed to be implemented if not enough settings are in the game or any complications appear).

This will handle creating the game manager and the connecting and disconnecting from clients. (need to take client silent disconnecting e.g. ALT F4 (Closing the application unsafely) and not sending a disconnect message)

## Game Manager:

This is what keeps tacks of the rounds and the stats of the players. It deals with loading the next rounds too. This acts like a persistent object between rounds but not games. We can store the players score here and they will reset when a new game is started because this object will be replaced.

## Round Manager:

This will handle spawning of buff items (the damage, ammo, heal and gas grenade). This will reset every round. A timer is also running. If the timer reaches zero, it will end the round. (I mentioned what happens when the timer runs out earlier).

This will also need to track the player’s body (not the owning entities).

## Player Systems:

It will need to handle a lot of stuff such as movement and look, weapons and throwables, player UI, health and stats. This should be strictly gameplay but can change if any issues occur that need to use the player system “layer” to work.

# Flow charts and UML:

# Refs:

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

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