

Concept: 4v4 3rd person shooter with 2 other player called masterminds who help their allies during combat from a 2d interface. The goal of the game is to destroy the mastermind opponent who is located at the end of their base. The mastermind works like a support hacker closing/opening door for its allies dropping loot on them such as extra ammunition.

Core aspects:

third person shooter

Online

Mastermind requirement:

- Good ui layout for the mini map interface (Really important it needs to be good for it to be playable)
- Code to switch from normal 3rd person shooter to rts-like gameplay
- Have everything important in the map be rendered into

Map: every thing on the map need to appear in the mini map of the mastermind and anything that could be interacted with need to show up in the mastermind ui

Work:

IMPORTANT STUFF TO KNOW:

We gotta make the file organisation professional meaning no Felix_stuff file names

Any asset use needs to be consistent with the rest of the asset meaning if we are doing a low poly all assets need to be low poly. This applies to ui where the ui need to be consistent with the feel of the game.

1. Gameplay Mechanics

Third-Person Shooter (TPS)

- **Player Character Controls:**
 - Implement movement (WASD or joystick).
 - Implement aiming, shooting, and reloading mechanics.
 - Add crouch, sprint, and jump functionality.
- **Weapons:**
 - Create different weapons (e.g., rifles, shotguns, grenades).
 - Add weapon switching and ammo systems.
 - Implement projectile physics for weapons (hitscan vs. projectile-based).
- **Health System:**
 - Create health and armor systems for players.
 - Add health regeneration or health pack pickups.
- **Death and Respawn:**
 - Implement player death mechanics and respawn timers.
 - Display kill/death notifications and scoreboard updates.

Online Multiplayer

- **Matchmaking and Lobby System:**
 - Allow players to form teams (4v4) with matchmaking.
 - Create team-based spawning and balancing systems.
 - **Replication:**
 - Ensure all critical systems (movement, shooting, health, abilities) are properly replicated for online play.
 - **Chat and Voice:**
 - Add in-game voice chat or text chat for teams.
 - **Game Session Management:**
 - Create game start, pause, and end conditions.
-

2. Mastermind Gameplay

RTS-like Interface

- **Camera System:**
 - Add a top-down camera mode for Mastermind players.
 - Implement zoom, pan, and rotate controls.
- **UI Layout for Mastermind:**
 - Design an intuitive and responsive UI layout for the Mastermind role, including:
 - A detailed minimap showing all players, objectives, and interactive elements.
 - Clear team management tools (e.g., selecting and assigning orders to players).
 - Quick access to abilities, building options, or commands.
- **Ability to Switch Roles:**
 - Allow players to switch seamlessly between the third-person shooter and RTS gameplay modes.
 - Ensure smooth transition between camera perspectives.

Interactive Map

- **Rendering Map Elements:**
 - Render all important map features (e.g., cover areas, objectives, pickups) into the minimap.
 - Use icons and color-coding to indicate different entities (players, interactables, obstacles).
 - **Interactivity:**
 - Allow the Mastermind to click on elements in the UI to:
 - Assign orders to players (e.g., attack, defend, move).
 - Trigger map-based abilities (e.g., airstrikes, healing zones).
 - Interact with specific map features (e.g., doors, traps).
-

3. Map Design

- **Main Map Layout:**
 - Design a symmetrical map with two distinct bases.
 - Add a clear pathway to the Mastermind's location in each base.
 - Include areas for combat, cover, and strategic positioning.
 - **Interactive Elements:**
 - Place interactable objects such as doors, traps, or terminals.
 - Implement objectives that can influence gameplay (e.g., control points or resource nodes).
 - **Minimap Integration:**
 - Ensure all map elements are accurately reflected on the minimap.
 - Add dynamic updates to the minimap for real-time events (e.g., explosions, player movements).
-

4. Mastermind Player Features

Base Defense

- Add options for the Mastermind to:
 - Deploy defensive structures (e.g., turrets, barriers).
 - Set traps or automated systems to protect the base.

Abilities

- Create abilities exclusive to the Mastermind, such as:
 - Buffs or debuffs for allied/enemy players.
 - Map-based support actions (e.g., healing zones, artillery strikes).

Resource Management

- Implement a resource system to limit the Mastermind's abilities and defenses.
 - Add resource generation points on the map.
-

5. Visuals and UX

UI Design

- **TPS Players**
 - Design HUD elements for health, ammo, abilities, and objectives.
- **Mastermind UI:**
 - Use clear visuals for the minimap, player controls, and resource management.
- **Cross-Role Communication:**

- Implement a system for TPS players and Mastermind players to coordinate effectively (e.g., ping system, quick messages):

Visual Feedback

- Add effects like screen shake, hit markers, or UI alerts for critical events (e.g., Mastermind abilities or base damage).

Who does what?:

Gameplay

Player Gameplay: Felix

Mastermind gameplay: