



GAME DESIGN DOCUMENT (GDD)

Game Name: Neighborhood Defense

Genre :

First-Person Shooter (FPS). The game focuses on wave-based combat, enemy survival mechanics, and first-person shooting, set in a familiar residential environment that gradually becomes hostile.

Game Elements:

Action / Shooter

Real-time first-person combat based on precision shooting using raycasting, requiring fast reactions and accurate aiming.

Survival

The player must survive successive waves of enemies, managing positioning, movement, and combat efficiency to stay alive as difficulty increases.

Strategy

Although primarily action-based, the player must choose where to position themselves, how to move through the environment, and how to handle increasing enemy pressure as rounds progress.

Player:

The game is a single-player experience, presented from a first-person perspective. The body of the playable character is not shown, to avoid unnecessary complexities in the animations and preserve immersion. The player's identity is deliberately kept neutral, avoiding any kind of gender stereotype.

TECHNICAL SPECS

Technical Form:

The game uses 3D graphics throughout, with a fully three-dimensional environment. The settings combine specifically modelled structures with imported resources. Lighting is calculated in real time and incorporates post-processing effects to enhance visual quality. In addition, shooting interactions, enemies and the environment are governed by a physics system that brings realism and consistency to the experience.

View:

The main camera is a first-person camera linked directly to the player controller. This camera is used as the origin point for the raycasting-based shooting system, ensuring accuracy and consistency. No additional cameras are used, keeping the system simple, efficient, and performing well.

Platform:

PC (Windows)

Keyboard and mouse input

Built and tested as a standalone executable

Language:

C# scripting

Developed using Unity

GAME PLAY

Locations:

Entire map. A structure modeled, serving as the central defensive location.



Main House. A structure modeled, serving as the central defensive location.



Other buildings. Different structures surrounding the house.



Park Area. Additional combat zone that expands the playable space.



Characters:

Player. First-person controlled character

Enemies (“The Glitches”). Hostile entities that spawn in waves and pursue the player using NavMesh-based AI

Objects:

- Player weapon
- Enemy entities
- Environmental structures (walls, doors, windows)
- Particle effects for bullet impacts
- Spawn points for enemy generation

Graphics:

Visual design prioritizes clarity and performance over excessive realism:

- Simple but readable materials
- Clear contrast between indoor and outdoor spaces
- Uniform interior lighting for combat visibility
- Directional exterior lighting simulating sunlight
- Particle effects for impact feedback

Game Play Outline:

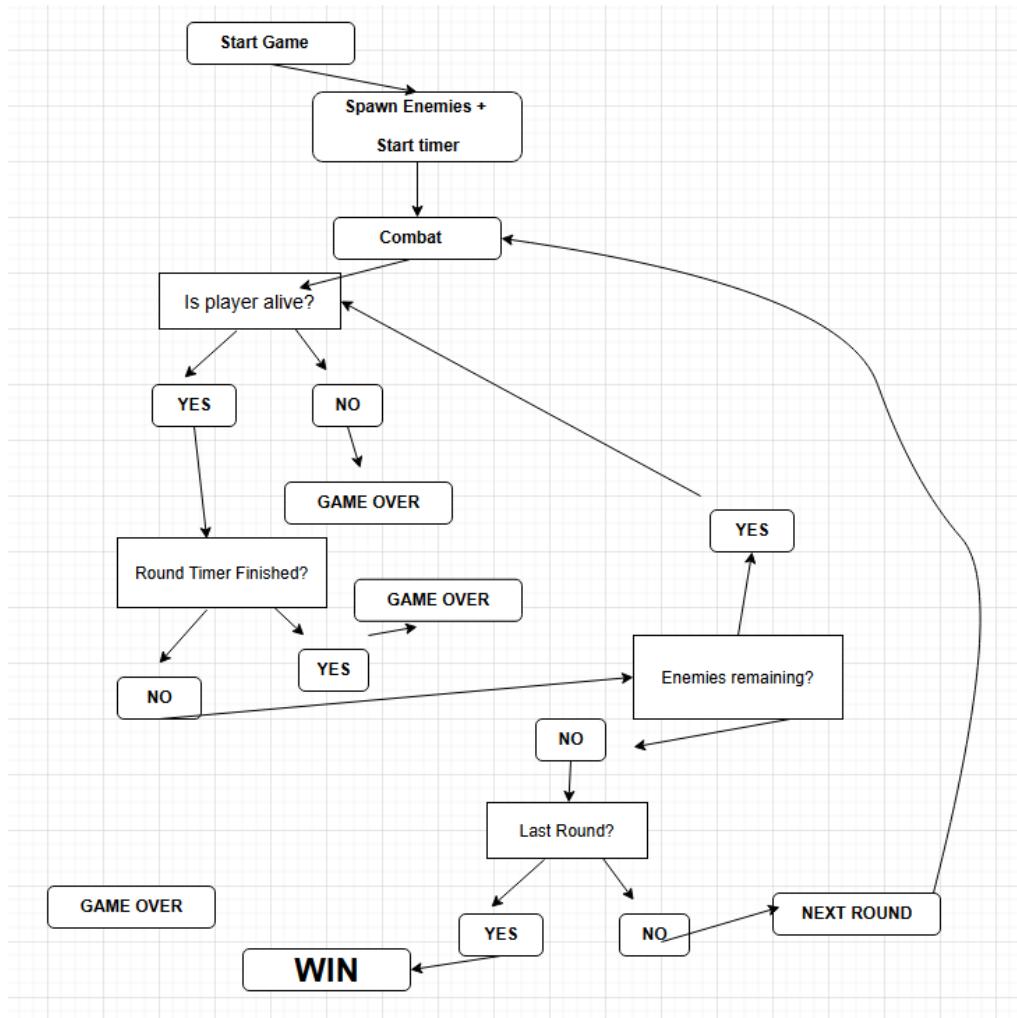
When the game starts, the player is placed in a neighborhood environment. The first round begins automatically, kicking off the action without the need for additional intervention. Enemies appear at predefined spawn points and move towards the player using a NavMesh-based navigation system. The player can eliminate enemies using a raycasting-based shooting system. Once all enemies in a round have been defeated, the next round begins automatically, progressively increasing the difficulty by generating a greater number of enemies. If the player's health reaches zero, the game ends.

Game Flowchart

The game flowchart represents the logical structure of the gameplay loop in Neighborhood Defense. The player starts the game and progresses through successive rounds. Each round spawns enemies and activates a timer.

The game continuously checks whether the player is alive and whether the round timer has expired. If either condition fails, the game ends with a Game Over state. If all enemies are defeated before the timer ends, the game progresses to the next round with increased difficulty.

The game is won when the player successfully completes all available rounds.



Player Definition:

The game is designed for:

- University-level students learning game development concepts
- Players interested in first-person shooters
- Players who enjoy wave-based survival challenges
- Learners exploring AI navigation, physics-based combat, and 3D environments

User Interface (UI):

- Minimal HUD
- Crosshair for aiming
- Player health display
- Visual hit feedback via particles
- Future expansion may include:
 - Score display
 - Round counter
 - Game Over and Victory screens