

AEM STUDIOS INC.



Video Game Concept and Design Document:

SHOWDOWN

Version #01

Copyright © 2024 AEM STUDIOS INC.
All rights reserved.

AHMET FURKAN AYLAC
MUHAMMED MUSTAFA AY
MUHAMMET EMİN TUNÇEL

21.08.2024

Document Contents

Table Of Contents

1	<u>TEMPLATE INTRODUCTION.....</u>	<u>1</u>
2	<u>CONCEPT DOCUMENT.....</u>	<u>2</u>
2.1	TITLE PAGE	2
2.2	CREDIT PAGE	2
2.3	SIGN-OFF	3
2.4	INTRODUCTION.....	3
2.5	GAME ANALYSIS	4
2.6	GAME ATMOSPHERE.....	6
2.7	GAME PLAY	8
2.8	KEY FEATURES.....	9
2.9	SELLING FEATURES	10
3	<u>DESIGN DOCUMENT.....</u>	<u>12</u>
3.1	DESIGN VERSION.....	12
3.2	DESIGN GUIDELINES	12
3.3	GAME DESIGN DEFINITIONS	12
3.4	GAME MATRIX	13
3.5	GAME FLOW CHART	14
3.6	PLAYER ELEMENTS.....	15
3.6.1	PLAYER DEFINITION	16
3.6.2	PLAYER PROPERTIES	17
3.6.3	PLAYER REWARDS (POWER-UPS & PICK-UPS)	18
3.6.4	USER INTERFACE (UI)	18
3.6.5	HEADS UP DISPLAY (HUD)	19
3.6.6	PLAYER VIEW	20
3.7	ANTAGONISTIC ELEMENTS	20
3.7.1	ANTAGONISTIC DEFINITIONS	23
3.7.2	ANTAGONISTIC PROPERTIES.....	23
3.7.3	ANTAGONISTIC LIST	23
3.7.4	ARTIFICIAL INTELLIGENCE (AI)	23
3.8	GLOBAL GAME ELEMENTS	25
3.9	THE STORY	25
3.9.1	THE STORY COPY	26
3.10	CONCEPT ART	26
3.11	LEVEL DESIGN.....	26
3.11.1	LEVEL COPY	26
3.12	AUDIO & SOUND F/X.....	26
3.13	GAME ARCHITECTURE	27
3.13.1	GAME ARCHITECTURE OVERVIEW	28
3.13.2	ARCHITECTURE COPY.....	28
3.13.3	HOW TO PLAY COPY	28

<u>4</u>	<u>TECHNICAL DOCUMENT</u>	<u>29</u>
4.1	SYSTEM REQUIREMENTS	31
4.2	VISUAL CONTENT.....	31
4.3	AUDIO CONTENT	33
4.4	PROGRAMMING CONTENT.....	33
4.5	CODE STRUCTURE.....	34
4.6	CONCERNS AND ALTERNATIVES	35
4.7	RESOURCES	36
4.8	TECHNICAL MATRIX.....	36
<u>5</u>	<u>APPENDIX A</u>	<u>39</u>
5.1	TAXONOMY	39
5.2	GENRE.....	39
5.3	GAME ELEMENTS.....	40
5.4	CONTENT	40
5.5	THEME	40
5.6	STYLE.....	40
<u>6</u>	<u>APPENDIX B.....</u>	<u>41</u>
<u>7</u>	<u>APPENDIX C</u>	<u>42</u>
<u>8</u>	<u>APPENDIX D</u>	<u>43</u>

1 Template Introduction

This document is based on the research done during an internship at Overloaded Pocket Media. The document itself has been updated several times based on new experiences and insight into game design. In this version the intention is to describe a broader range of game than just mobile games, as well as provide a document for fellow students and game designers to use.

This document has been created with the intention of making a clear game design document. To do this the document has been divided into 3 different documents, the game concept document, game mechanics document and game tech document. This is done with the intention of making a modular document that can be updated and split into different versions easily.

Documents created with this game design template will focus upon creation of a well devised game. What this template does not do is predict the playability of the game and define a plan for the overall game development. It is therefore important to consider this is only one phase in the Game Development process. It is recommended that any game design be tested with tangible means such as pencil and paper, cards, board, etc, before any major resources are allotted to it. It is also important to understand that this document will act as a starting point and can not replace hands on interactivity and game testing.

This document is modified w.r.t. needs of Istanbul Sabahattin Zaim University YAM 415 Game Programming Course by Assist. Prof. Dr. Abdullah Sönmez.

2 Concept Document

The concept document serves the purpose as a way to present a game concept.

A general overview of the game, with the idea anyone can read and understand what the game is like.

This part of the document is one that will change very little once the concept is accepted.

2.1 Title Page

Game Title: Showdown

Game Logo:



Game Catch Phrase: Fight, Survive and Win.

Document Type: Game Concept & Design Document

Document Version: 1.0

2.2 Credit Page

Document Purpose: To outline the core gameplay mechanics, atmosphere, and unique elements of the battle royale game Showdown.

Document Version: 1.0

Working Title: Project_Showdown

Game Concept: Showdown is a survival-based battle royale game where players or teams fight to be the last standing on a shrinking map, with unique character abilities enhancing both offense and defense tactics.

Game Document Author:

Ahmet Furkan AYLAÇ

Contributors:

Muhammed Mustafa AY

Muhammet Emin TUNÇEL

2.3 Sign-Off

This section is here to confirm that each team leader has read through the design and agrees with the current game design.

Game Design Sign-off

CEO:

Lead Designer:

Lead Programmer:

2.4 Introduction

Showdown is a battle-royale survival-action game that designed for solo players or teams up to 4. That game is focusing on tactical decision-making and skillful combat. The map is isolated arena that either takes the form of island or the simulation area, the gamers have loot, survive, and compete each other until to become the last player or last group. Players begin the game from underground, ascending into the arena like style of *Hunger Games* and then battle within a shrinking safe zone as they search for resources and eliminate opponents.

The Showdown introduces additional layer of strategy through with special character abilities, before the game starts, players can choose from one of the three characters, each offering special skills tailored to offense, defensive, or tactical advantages. This character selection scheme effects the gameplay style, allowing players to adapt different scenarios based on their team composition or strategy.

With immersive visuals, a dynamic arena environment, and player abilities that support both groups play styles and solo play styles, Showdown aims to offer an exciting and strategic battle-royal experience.

Genre: Battle Royale, Action-Adventure, Shooter

Player Type: Multi-player and team based (1, 2 or 4 players per team)

Technical Form: 3D, Multi-player, Abilities

References: By games like PUBG and *Valorant* for gameplay mechanism and abilities.

Theme: Survival, Modern, Tactical Strategy

Design Intentions: The game offers exciting action and strategy, allowing players to adapt their playstyle with unique character abilities, whether playing solo or team. Set in dynamic environments, like an island or simulation zone, players face intense challenges as the safe zone shrinks. Character skills and tactical depth, enhancing replayability as players explore various combinations and strategies. Both teamwork and individual skill are rewarded, making for a balanced, competitive experience until only one player or team remains.

2.5 Game Analysis

This is a general overview of the game.

Game Description	
Genre:	<ul style="list-style-type: none">▪ Battle Royale▪ Survival▪ Action
Game Elements:	<ul style="list-style-type: none">▪ Combat Players fight each other with different weapons and character skills.▪ Collecting Looting special items, armor and etc. across the map to improve win chances.▪ Strategy Choosing character abilities for team strategy and gain a competitive edge.▪ Exploration Moving around the map, find good items and locate other players to be winner.

Game Content:	<ul style="list-style-type: none"> ▪ Action Fast-paced combat and survival challenges. ▪ Thriller The shrinking safe zone and the unpredictability of enemy movement add a constant layer of suspense. ▪ Realism A modern-combat atmosphere and supported with modern-realistic weapons, detailed maps with modern structures and environmental effects. ▪ Survival Players must collect items, develop strategies and use tactics to stay alive and to be winner.
Theme	<ul style="list-style-type: none"> ▪ Modern Warfare and Survival ▪ Tactical Competition
Style:	<ul style="list-style-type: none"> ▪ Realistic ▪ Modern-time Zone
Game Sequence:	<ul style="list-style-type: none"> ▪ Simulation ▪ Battle Royale Sequence
Player:	<ul style="list-style-type: none"> • Single-player (with bots) and Multi-player (up to 1-4 players per team) (with bots)
Game Reference	
Game Taxonomy:	<ul style="list-style-type: none"> • Fiction Simulation / Competitive Game • Subcategories: <ul style="list-style-type: none"> - Simulation: To emulate survival experiences with realistic map details, weapon mechanisms and dynamic safe-zone. - Battle Royale: Player and teams compete in survival format, with only one person or one group winner. - Action&Strategy: Features of combat and strategic decisions based on character skills.
Player Immersion:	<ul style="list-style-type: none"> • Tactical Gameplay: Players can experience survival atmosphere with character abilities and strategic locations on map. Winner must play with good tactic. • Survival Pressure: The shrinking safe zone force the players enable decision-making and engage in combat, adding tension. • Solo and Team Play: The game supports every kind playstyle that solo or team. Immersing players in a strong competitive arena. • Realistic Environment: The island or simulation zone, filled with modern-realistic details, enhances the feelings of being in fight.
Reference:	<ul style="list-style-type: none"> • The story and competitive way in Showdown inspired by concept of dystopian tournaments, like The Hunger Games. Where participants compete for life and a chance of redemption. • The gameplay and battle-royale format are inspired from PUBG, with players dropping into arena or zone, looting for items, and competing to be last one standing. • The character abilities draw inspiration from games like Valorant, which use unique character skills to add depth of strategy to gameplay, enabling players to choose different approaches based on their chosen character.

Game Technical	
Technical From:	<ul style="list-style-type: none"> • 3D • Solo or Multiplayer
View:	<ul style="list-style-type: none"> • First Person Shooter and Third Person Shooter
Platform:	<ul style="list-style-type: none"> • C#
Device:	<ul style="list-style-type: none"> • PC
Game Sales	
Consumer Group:	<ul style="list-style-type: none"> • Aimed at players that 16+ and who enjoy action-packed battle-royale game.
Payment:	<ul style="list-style-type: none"> • Initial purchase with optional in-game purchases for cosmetic items and character customizations. Enhancing player experience without affecting core gameplay.
Estimated Price:	<ul style="list-style-type: none"> • The game will be sold for around \$10-\$20, with optional in-game purchases that like season pay, cosmetics and character customization. This game can be service game.
Device Support List	<ul style="list-style-type: none"> • PC

2.6 Game Atmosphere

The players are spawned in a simulated zone that contains a mix of realistic nature and modern, abandoned structures. This combination of untouched natural landscapes, mountains, and decaying buildings creates a tense, post-apocalyptic atmosphere.

• 2.6.1 Character/Units Sketch & Description

- *Character Types:* Players can choose from three class character, each with unique abilities that offensive, tactical, and defensive.
- *Description:* Characters are designed to blend with environment's gritty, realistic and survival theme, equipped with army gears for both natural and urban settings within in dome.

- *Characters Sketches:* (Handmade by Ahmet Furkan Aylaç)



Sniper:
This character's playstyle is based on secret.

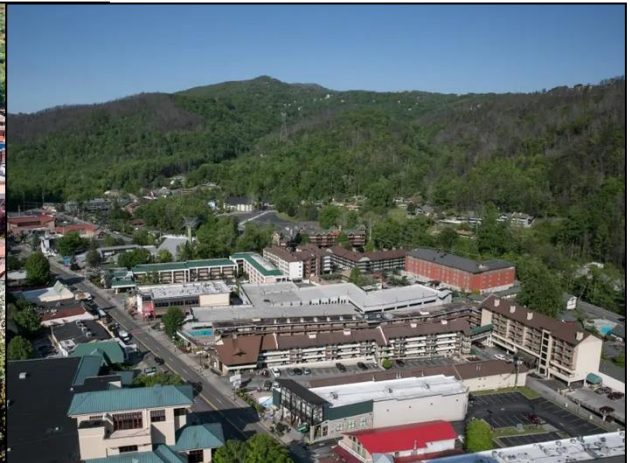
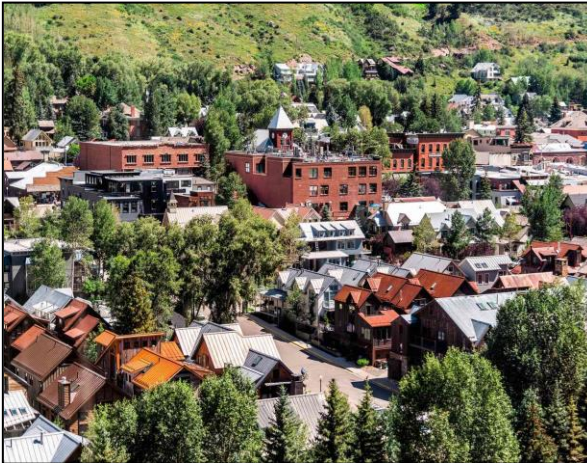
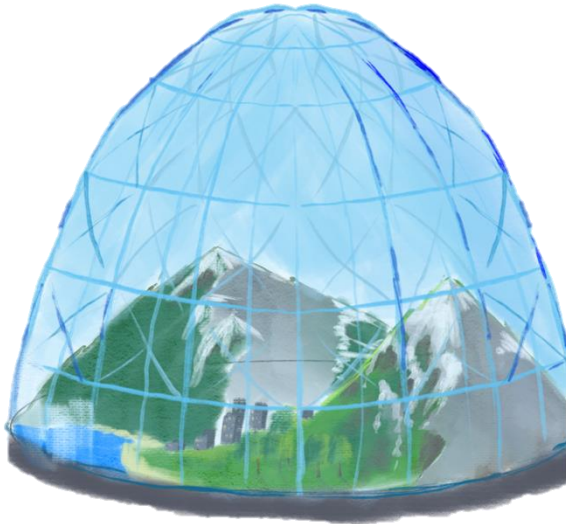


Attacker:
This character's playstyle is based on attack, this class can be mid-class. Suitable for the average action gamer

- **2.6.2 Level (Locations) Sketch & Description**

- *Location Sketch:* The arena map within in dome features a mix of dense forests, mountain terrains, flat areas, modern-towns (1.1) and abandoned urban areas, with resources for players to loot. The dome structure encapsulated the whole map, creating a boundary that restrict players for survival experience. The game area, that dome, will be designed as a circular map because of allowing for a randomly shrinking safe zone. The toxic gas will progressively close in, pushing players closer together to more action and create more action scene.
- *Description:* This environment challenges gamers to navigate and survive in terrain, from open natural spaces to confined abandoned ruins, with the dome's structure constantly reminding them of the border.
- *Map Sketch:* (Handmade by Ahmet Furkan Aylaç)

The map:
This sketch is done for understanding how the idea.



- **2.6.3 Audio Description**
 - *Ambient Sounds:* The game has quiet electronic hums from the dome, distant animal sounds, and wind through trees.
 - *Combat and Action Sounds:* Players can experience the survival environment by listening to realistic gunfire, footsteps on various surfaces, echoes in cities, and the silence in woodlands.
 - *Dynamic Sound spaces:* The sounds blend in with the atmosphere of the dome, whether it's the gentle hum of the barrier or the silent silence of abandoned buildings.

2.7 Game Play

- **2.7.1 Opening the Game Application**
Gamers launch the game and enter the main menu, players have to be chosen how to play (solo or team) after that, players need to choose one of them characters. After the matchmaking query, player can play.
- **2.7.2 Game Options**
Players can customize setting such as graphics, sound, and basic controls.
- **2.7.3 Story Synopsis**
In the year 2100, skilled individuals with special abilities, marked by their criminal pasts, compete in a high-stakes sci-fi dome. The winner earns a chance to redeem themselves by fighting for humanity's future.
- **2.7.4 Modes**
 - *Solo Mode:*
Players fight alone to be winner.
 - *Team Mode:*
Players can join or can create a group of up to four, working together and to be last group.
- **2.7.5 Game Elements**
 - *Battle Royale:*
The core gameplay is about to survival and competitive style within a shrinking safe zone.
 - *Character Abilities:*
Each player can select specific characters, each with unique abilities. (offensive, defensive and tactical)
 - *Looting and Resources:*
Players must gather modern weapons, armor, and healing items across the map.
 - *Toxic Gas:*
A deadly gas closes in, reducing the safe zone and pushing gamers closer for action, which intensifies combat and increases the stakes as the game progress.
- **2.7.6 Game Levels**
The game takes place on a single, big map enclosed in zone, with diverse terrains such as forest, mountains, abandoned rains, flat areas and modern American towns.
- **2.7.7 Player's Controls**
Players use standard controls for movement, aiming, and using abilities, with additional keys for accessing the inventory, map, and team communications.
 - Example Controls:
 1. W – Forward

Figure 1.1

4. D – Move Right
5. Space – Jump
6. Shift – Sprint
7. Ctrl – Crouch
8. C – Toggle between crouch and stand
9. Left Mouse Button – Fire
10. Right Mouse Button – Aim
11. R – Reload
12. 1,2,3 or Scroll – Switch Weapons
13. G – Throw Grenade
14. F – Interact / Pick up Item
15. Q – Use Character Abilities 1
16. E – Use Character Abilities 2
17. M – Open Map
18. Tab – Open Inventory

- 19. H – Use Healing Item
- 20. V – Toggle Camera View

- **2.7.8 Winning**
The game ends when the last player or group remains alive within the shrinking safe zone, to win.
- **2.7.9 Losing**
Players or groups are eliminated if they're defeated by opponents or if they're outside the safe zone when the toxic gas closes in.
If play mode is group, the whole member of group has to be dead for losing game.
- **2.7.10 End**
The last one or last team is declared the winner, earning rewards (this can be in-game coin), achievements, experience (for level system) based on their performance.
- **2.7.11 Why is all this fun?**
An exciting, strategic experience is produced by the combination of challenging survival gameplay, special character skills, and a steadily decreasing safe zone. Finding resources, outwitting rivals, and adjusting to the surroundings create suspense that keeps players interested and challenged in every game.

2.8 Key Features

- **2.8.1 Number of Levels**
The game features an alone, expansive map designed as a high-tech dome that randomly shrinks during each match. This large map is diverse in terrain, including forests, mountains, and abandoned places, making every match feel distinct.
- **2.8.2 Number of Enemies/Characters**
Players can choose from three specific characters, each with special abilities that focus on offense, defense, or tactical advantages. The game allows for up to 25 players (*it can be change*) (*with bots*) in a single match, competing either solo or in teams.
- **2.8.3 Time of Gameplay**
Each match lasts roughly 20-30 minutes, depending on how quickly players are eliminated and the safe zone shrinks. This keeps gameplay fierce and time-efficient for short play sessions.
- **2.8.4 Replayability**
Every match offers a good experience because to the varied terrain on the map, randomized diminishing zones, and various character abilities, which entices players to come back and try out new tactics.
- **2.8.4 Audio Specifications**
Immersive sound design with ambient noise (wind, electronic dome hum [*if player is too close to border.*]) realistic combat sounds, and dynamic audio [*5.1 Surround Audio can be added.*])
- **2.8.5 Graphic Specifications**
High-quality 3D graphics with realistic light and textures, creating a visually engaging experience. The dome's transparent blue shield, natural terrain structures, and decaying modern-style buildings contribute to the game's intense, modern-sci-fi atmosphere.
- **2.8.6 Device Combability**
Current available for only PC.
- **2.8.7 Number of Players**
Up to 25 players (*it can be change*) (*with bots*) can be in a match, either a solo player or group. (*1 to 4*)
- **2.8.9 Online Activities**
Players can track their level and achievements. A global leaderboard adds a competitive element, showing the top players and teams based on KDA and Win Rate.
- **2.8.10 Number/Type Modes**
 - Solo Mode: For players who want to compete just only.
 - Team Mode: Players can be team with 1,2 or 3 players, encouraging cooperation and strategic play.

2.9 Selling Features

- **2.9.1 Marketing Ideas**

- Promote the game as an exciting, modern-sci-fi battle royale inside a dome. Show the game's survival and competitive style with unique character abilities and intense action with shrinking safe area.
- Sharing the gameplay clips on social media or stream platforms to show exciting and action moments, character skills, lucky scenes, and teamwork.
- Work with popular streamers (*Twitch, Kick and YouTube*) to help reach more people who like to play battle royale games.
- Pricing can be optimized for different countries to increase accessibility and attract global players.
- Players who join Beta Test Version, can receive rewards. Increasing the loyalty and stronger connection with game.

- **2.9.2 Consumer Group**

Aimed to players ages 16+ and who enjoy action, battle royale and survival games. Ideal for players, looking for exciting and strategic multiplayer game in PC. Many players today, try to find best game that can play with friends. The *Showdown* offers both solo and team modes for shadier experience with friends.

- **2.9.3 Unique Features**

- *Special Character Abilities*
Players can choose one of three characters with special abilities (*offensive, defensive and tactical*) adding more strategy and exciting mechanic to game.
- *Futuristic Safe Zone*
The game's map mixed natural areas, old buildings and also some modern buildings inside a dome, creating unique look and feel.

- **2.9.4 Merchandising**

- In-game items like character outfits, weapon designs and special skins to match the game's style.
- Branded items like t-shirt, cup, posters, characters model and accessories with game's logo, character designs and some fan-art.
- Giving early-access code to famous YouTubers and streamers can generate excitement and hype. By sharing their gameplay experience, they can build good feelings to game before the release date.

This document describes how game objects behave, controlled and the properties they have.

The enemy game object moves towards the player when in range and performs an attack action. If the player is out of range, it patrols a predefined path.

The player character is controlled using keyboard inputs: WASD for movement and the spacebar for jumping. Alt for focus-mode. Non-player characters (NPCs) are controlled by an AI system that uses decision trees to determine behavior.

This documentation is primarily concerned with the game its self.

The game is TPS Shooter – Battle Royale. There is some enemy objects and a map. Map on a island and on the island we have one big city. Non-player characters(NPCs) are controlling by ai that move them randomly and you have to find the Non-player characters and shoot them if you can take down eleven of them you will win, if you can't you lose.

This part of the document is meant to be modular. Meaning you could have several Game Mechanic documents attached to the Concept Document.

W: Move Forward A: Move Left ALT: Focus-Mode SHIFT: Run

S: Move Backward D: Move Right SPACE: Jumping

MOUSE SCROLL: Changing Weapons M: MAP R: Reload

LEFT CLICK: Aiming RIGHT CLICK: Shoot

3.1 Design Version

(U.): (UNITY) Developed for game. Version: unity 6000.0.29f1

(UT): UNITY Technologies.

(VSC) Visual Studio Code for c# scripting

3.2 Design Guidelines

This is an important statement about any creative restrictions that need to be regarded and a general over all goal of the design.

Game is about to playin TPS-Shooter and mindset of Battle-Royale so we need to have a limited map. So we have to make a island for our perception in Battle-Royale. Also we have a city which Non-player characters will move randomly so we made the city on island. We have a view limiter for our character. Paths, buildings, mountains, forests and sea everywhere except the island for our goal design..

3.3 Game Design Definitions

This is a section where the definition of the game play is established.

Game playing is actually so easy you have some bullets and eleven enemy every enemy NPCs has hundreds hp and characters have weapons dmg 20 per hit so you need to hit 5 times for each bot to take down. NPCs are spawning on city but moving randomly so you need to take them down quickly if you can't they can go everywhere of the island. NPCs have their own attack animation and hit the damage to playin characters hp bar and if heath points runs out then you will lose the game.

Definitions should include how a player wins, loses, passes levels and the main focus of the game play.

Win: You have to take down all eleven NPCs.

Lose: Your health points need to runs out from attacks of NPCs.

Level: There is no level for Battle-Royale games.

Main Focus of the Gameplay: Typical Battle-Royale but offline version. Take down all enemies(NPCs) and win.

Issues that should be addressed here are:

- Starting screen with two character option, start and exit options.
- Battle-Royale game with eleven NPCs if you can take down before hp bar runs out you win.
- TPS angle Shooter game with two different weapon and character you chosen at first screen.
- W A S D for movement SPACE for jumping ALT for focus mode Mouse scroll for weapon switch LEFT CLİCK for shoot RİGHT CLİCK for aiming R for reload.

- If you can take down eleven npcs you win if your hp bar runs out you lose.

3.4 Game Matrix

Game Objects Table

Name	Type	Health	Speed	Abilities	Behavior	Notes
Player	Playable Character	100	Walk 2 units/s Run 4 units/s	Jumping, Shoot, Focus Mode	Controlled by player input	Main Character
Solider	Enemy(NPC)	100	Walk 2 units/s Run 4 units/s	Melee Attack	Patrols, attack on sight	Common enemy

The table provides an easy cross-reference for game elements that have specific values, such as health, speed, abilities, and behavior patterns. For instance, developers or designers can quickly identify the properties of the "Player" or "NPC" without searching through individual files or documents. This ensures consistency and helps in if we want to balance the game mechanics by comparing values like health or damage output across main character.

It is essential to consult with the programmer to define the technical implementation of a game object's properties. For example, the Player Character's "Health Points(HP)" might need to be implemented as an integer, while the "Speed" property could be a float. Additionally, we need to discuss how abilities such as "Focus Mode" will function programmatically, including triggers and cooldowns. This collaboration ensures that the design and technical implementation are aligned.

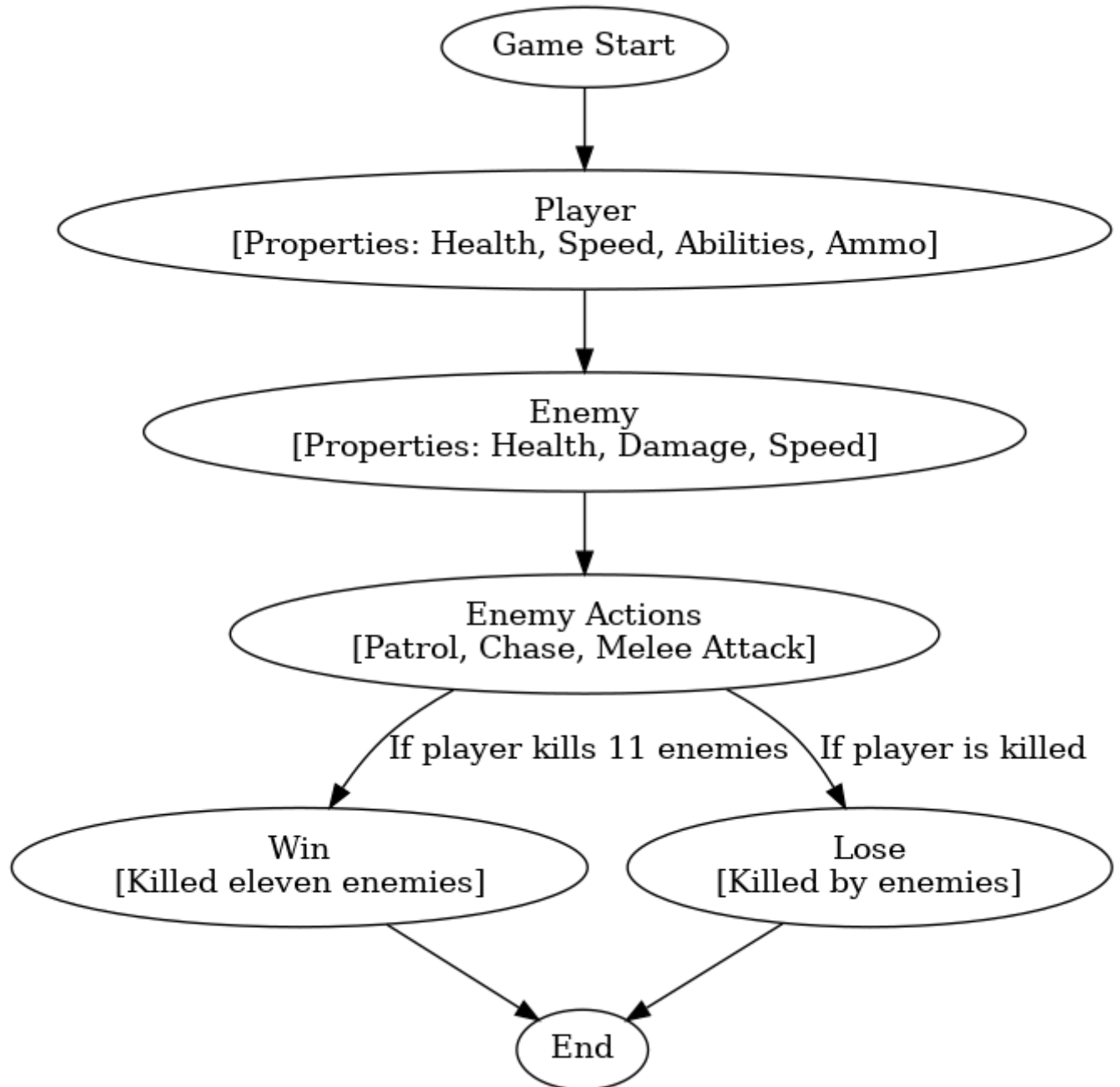
See Appendix B

3.5 Game Flow Chart



In this picture we can see the Field Of View(FoV) and our map crossing each other. Mountain has Friction Force so character can not fall from there(we will reduce Friction and character will slip.).

Game Flow chart



Game Start -> Player [1.1, 2.1, 3.1] -> [Properties: Health, Speed, Abilities, Ammo] Enemy [1.2, 2.2, 3.2] -> [Properties: Health, Damage, Speed] Enemy [1.2] -> Actions: Patrol, Chase, Melee Attack [3.2] Killed eleven enemies -> Win -> End Killed by enemies -> Lose -> End

See Appendix C

3.6 Player Elements

- HP bar
- Speed
- Weapons(switching with mouse scroll)
- Movement(w:move forward a: move left d:move right s:move backward space:jump)
- Focus-Mode(ALT)
- Map (you can attach a point from map for minimap)

MainCharacter1	Psyops
MainCharacter2	Cyborg
Weapon1	AK-47
Weapon2	Scar L

- **Health:**
Definition: Represents the amount of damage the player can take before they lose a life or the game ends.
Unit: Numeric value (100)
- **Speed:**
Definition: Player moving speed.
Unit: Distance per second.
- **Ability:**
Definition: Special actions the player can perform, such as jumping, shooting, or activating a unique skill.
Jumping: Allows the player to overcome obstacles.
Shooting: Enables the player to attack enemies.
Focus Mode: Temporarily enhances player performance (for make good shoot).



We don't have multi-player section for game.

3.6.1 Player Definition

The main characters, controlled directly by the player. Characters have unique appearance between each other. Player can shoot, jump, move, use Focus-Mode for aiming. Goal of the game is takedown npcs before hp bar runs out.

Default settings are 100 hp bots and weapons damage 20 to hp bar for npcs.

A suggested list of player definitions:

- Actions: W A S D for movement forward left back right. Space for jump. ALT for Focus-Mode.
R: Reload
MOUSE ACTIONS: LEFT CLICK: Shoot MOUSE SCROLL: Weapon switch
RIGHT CLICK: Aiming.
- Information (Status): HP bar for your character and its runs out with every hit from npcs. Npcs have hp bar (100HP). Weapons hitting 20 damage for each shoot each bullet.
- Default Properties: Easily, just run the exe file and press start button on Main menu.
- Winning: Game has eleven NPC on island player has to takedown all npcs for win.

- Loosing: Player have hp bar if your hp bar runs out before takedown all npcs you lose.

3.6.2 Player Properties

- **Health:**
Definition: Represents the amount of damage the player can take before they lose a life or the game ends.
Unit: Numeric value (100)
- **Speed:**
Definition: Player moving speed.
Unit: Distance per second.
- **Ability:**
Definition: Special actions the player can perform, such as jumping, shooting, or activating a unique skill.
Jumping: Allows the player to overcome obstacles.
Shooting: Enables the player to attack enemies.
Focus Mode: Temporarily enhances player performance (for make good shoot).
- **Health** affects the player's survival.
- **Speed** influences how quickly the player can navigate the world and evade threats.
- **Ammo** is crucial for combat. Limited ammo forces the player to think strategically, using weapons only when necessary.
- **Abilities** give the player the tools needed to overcome obstacles or enemies. The availability and usage of abilities will directly impact the player's approach to each situation.

A suggested list of player definitions:

- Health: 100
- Weapons: Scar L AK-47
- Actions: W A S D for movement forward left back right. Space for jump. ALT for Focus-Mode.
MOUSE ACTIONS: LEFT CLICK: Shoot MOUSE SCROLL: Weapon switch
RIGHT CLICK: Aiming.

Each property should mention a feedback as a result of the property changing

- **Health:** Player can HP bar so player can do whatever wants to do with knowin hp.
- **Speed:** The player can avoid enemy attacks faster.
- **Ammo:** Ammo counter shows how many rounds are left.
- **Abilities:** The player gets a temporary better aim.

3.6.3 Player Rewards (Power-ups & Pick-ups)

The player takedown all NPCs WINS the game.

3.6.4 User Interface (UI)

User Control of the Game

In this game, the player controls their character using the following inputs:

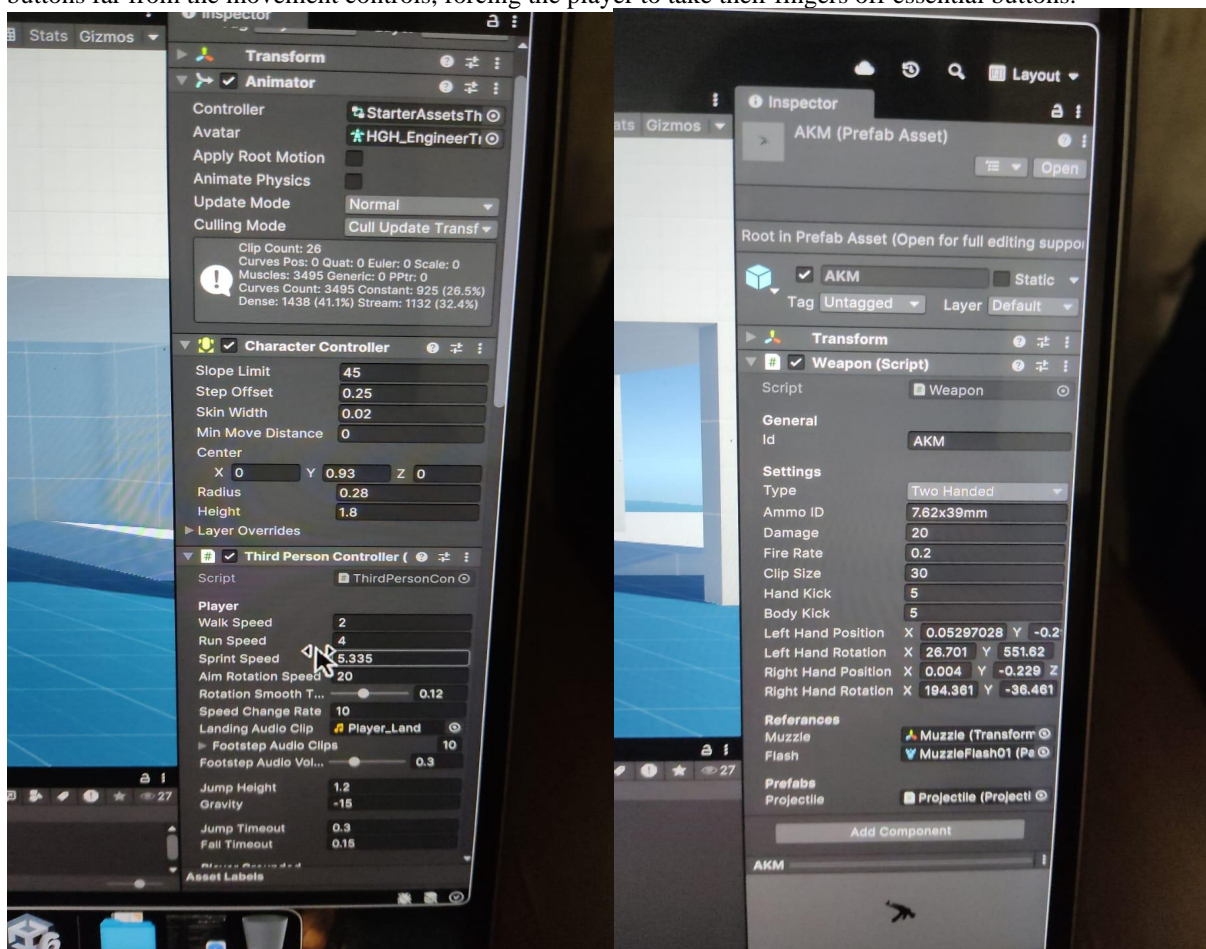
- **Movement:** WASD for movement
- **Jumping:** Spacebar
- **Attacking:** Left mouse click
- **Aiming:** Right Mouse click
- **Special Abilities:** Shift key (For run) & ALT(Focus-Mode)
- **Reload:** R

Ideal Control Layout

To ensure an optimal user experience, the layout should be intuitive and easy to access. On a keyboard, movement is controlled by WASD keys, while actions like shooting or jumping are mapped to easily reachable buttons

Worst Layout Evaluation

The worst layout would be one where essential actions are difficult to reach, such as placing the jump or attack buttons far from the movement controls, forcing the player to take their fingers off essential buttons.



Quality Control Feedback

When designing the UI and control scheme, we research the TPS Shooter games controllers for the most efficient use of the keyboard to the players.

3.6.5 Heads up Display (HUD)

This is where a description of any graphics that will represent information during game play should be described.

Health Bar

- **Description:** A visual bar that represents the player's remaining health. Provides immediate feedback to the player about their remaining health.

Ammo Counter

- **Description:** A display that shows the number of bullets or ammunition left for the player's weapon. This counter positioned at the bottom-left corner of the screen for easy access.

Score Display

- **Description:** A visual element that tracks and displays the player's current enemies defeated. This located at the top-right corner of the screen.

Minimap

- **Description:** A small, real-time map showing the player's position in the game world. The minimap located at the top-left corner.



3.6.6 Player View



Camera moving freely in normal game but if you aim(right click) camera will be turned to character's shooting angle.

3.7 Antagonistic Elements

Solider

- **Description:**
The **Solider** is a common enemy type in the game. He is armed with his fists. When the player comes within a certain proximity, the Solider will begin chasing the player and will attempt to attack them with powerful punches. The Solider is relentless and will follow the player's movements until they are in range for an attack. If the player is hit, they will take damage, so it is important to avoid the Solider's punches.
- **Behavior :**
 - **Chasing:** The Solider will patrol a certain area but will start chasing the player if they enter a predefined range.
 - **Melee Attack:** When close enough, the Solider will use a punch to damage the player.
 - **Patrol:** The Solider will walk randomly, but once the player enters their detection range, they will try to chase and attack player.
 - **Attack on Sight:** The Solider will attack immediately once the player comes into the attack range.

- **Properties:**
 - **Health:** 100
 - **Damage:** 10 (each punch)
 - **Speed:** 2 units/s (walking), 4 units/s (chasing)
 - **Detection Range:** 10 units (radius for chasing the player)
 - **Attack Range:** 2 units (for melee punch)
- **Visual Representation (Görsel Temsil):**



Health:

- **Description:** Represents the enemy's capacity to survive. The enemy loses health points as they take damage, and when health reaches zero, they are defeated.
- **Example:** "Health: 100"

Damage:

- **Description:** Represents the amount of harm the enemy deals to the player or other targets. Each attack inflicts a certain amount of damage.
- **Example:** "Damage: 10"

Speed:

- **Description:** Defines the movement speed of the enemy. It determines how fast the enemy moves and how quickly they can approach the player.
- **Example:** "Speed: 2 units/s"
- **Description:** Defines the distance at which the enemy detects the player or target and begins to chase them.
- **Example:** "Detection Range: 10 units".

Attack Range:

- **Description:** The distance at which the enemy can initiate an attack. It determines how close the player needs to be for the enemy to successfully hit them.
- **Example:** "Attack Range: 2 units"

Behavior:

- **Description:** Refers to the general actions and movement style of the enemy. It defines how the enemy moves and interacts with the player.
- **Examples:**
 - **Patrol:** The enemy moves randomly
 - **Chase:** The enemy starts chasing the player once detected.
 - **Attack on Sight:** The enemy will attack immediately when it sees the player.

Generic Name (Code)	Game Name
NPC1	Solider
NPCHealth	Enemy Health
NPCSpeed	Enemy Speed
NPCAttack	Punch Attack
NPCBehavior3	Patrol
NPCBehavior3	Chase
NPCBehavior3	Attack on Sight

3.7.1 Antagonistic Definitions

Antagonistic elements are designed to create conflict, challenge, and adversity for the player. Without antagonistic elements, a game would lack the core tension and challenge that motivates players to engage and progress.

3.7.2 Antagonistic Properties

- **Health:** 100
- **Damage:** 10 (each punch)
- **Speed:** 2 units/s (walking), 4 units/s (chasing)
- **Detection Range:** 10 units (radius for chasing the player)
- **Attack Range:** 2 units (for melee punch)

3.7.3 Antagonistic List

The Game has only 1 Antagonistic in game.

- **Soldier:** Soldiers spawn at a random location near the city on the map and moves randomly. When the player enters its detection range, the soldier will begin to follow and attempt to attack the player.

3.7.4 Artificial Intelligence (AI)

- **Normal State:** The bot uses a script that enables random movement to move randomly on the NavMesh when it cannot see the player. The AI-driven bot moves aimlessly through the surroundings.



- **Detection State:** The script that controls the bot's random movement is turned off once the player gets within ten meters of it; instead, a different script is triggered to cause the bot to approach the player. The player can get away because this bot moves a little more slowly than the player. The script for random movement is reactivated once the player leaves the radius.



- **Reaction State:** The Animator controls a punch animation that starts when the bot gets near 0.5 meters of the player. This animation mimics the player being punched by the bot. The player loses ten health points if the bot's collider makes contact with theirs. Until either the player or the bot kills the other, this behavior keeps happening.



- End State: The bot has 100 health points, much like the player. The bot loses 20 health points for each hit when bullets from the player's weapon strike its collider. A background script deactivates the random movement script and the Animator that controls the animations, destroying the bot if its health falls to zero. The bot is eliminated from the game and its death animation is activated upon annihilation.



3.8 Global Game Elements

Boundaries: Map is on island so our boundaries are water parts.

Neutral Objects: Mountains Buildings Forests Ride-Paths.

Camera Views: The game has TPS camera view.

Scale of the World: All island is our Scale of World.



3.9 The Story

The story takes place on an island where the main character is trying to kill 11 different NPCs to survive. The island is filled with harsh environmental conditions and various dangers. When the main character first arrives on the island, they must eliminate these enemies in order to survive and find a way to escape.

Each NPC is a soldier trained in the same way; if they see the player, they will chase and attack. Throughout the story, the main character will have to face each NPC one by one, but this journey will be not only a physical challenge but also a psychological struggle.

The island becomes not just a hunting ground, but also a place where the character must confront their inner fears and weaknesses.

In the end, the main character's hope of escaping the island remains, but the challenges they face will test them both physically and mentally.

3.9.1 The Story Copy

Player has to eliminate all eleven soldiers for survive. But this journey will be not only a physical challenge but also a psychological struggle.

3.10 Concept Art



3.11 Level Design

The Game has Battle-Royale playstyle so do not have any leveling system.

3.11.1 Level Copy

The Game has Battle-Royale playstyle so do not have any leveling system.

3.12 Audio & Sound F/X

Game has environmental sounds because player needs to hear soldiers footsteps.

3.13 Game Architecture

Game Architecture Flowchart (Custom Screens)



See Appendix D

3.13.1 Game Architecture Overview

Splash Screens and Video Clips: Since the game does not feature cutscenes, the splash screen has characters images and Game Logo is the background of Main Menu screen.

Menu Design: The game menu should be simple, intuitive, and focus on key options such as:

- **Start Game**
- **Options**
- **Exit**
- **Character choose screen**

Player can start the game with one click.

- **Movement and Controls:** WASD for walk shift for run and space for jump
- **Combat:** Right click for aim Left click for shoot Mouse Scroll for switch weapons and R for reload
- **Objectives:** Takedown 11 solider for win.

3.13.2 Architecture Copy

Splash Screens and Video Clips: Since the game does not feature cutscenes, the splash screen has characters images and Game Logo is the background of Main Menu screen.

Menu Design:

- **Start Game**
- **Options**
- **Exit**
- **Character choose screen**

Game Instructions:

Eliminate 11 solider before hp bar runs out.

3.13.3 How to play Copy

W: Move Forward A: Move Left ALT: Focus-Mode

S: Move Backward D: Move Right SPACE: Jumping

MOUSE SCROLL: Changing Weapons M: MAP R: Reload

LEFT CLICK: Aiming RIGHT CLICK: Shoot

Programming Aspects: In UNITY version of unity 6000.0.29f1. VsCODE for scripts c# Artificial Intelligence (AI)

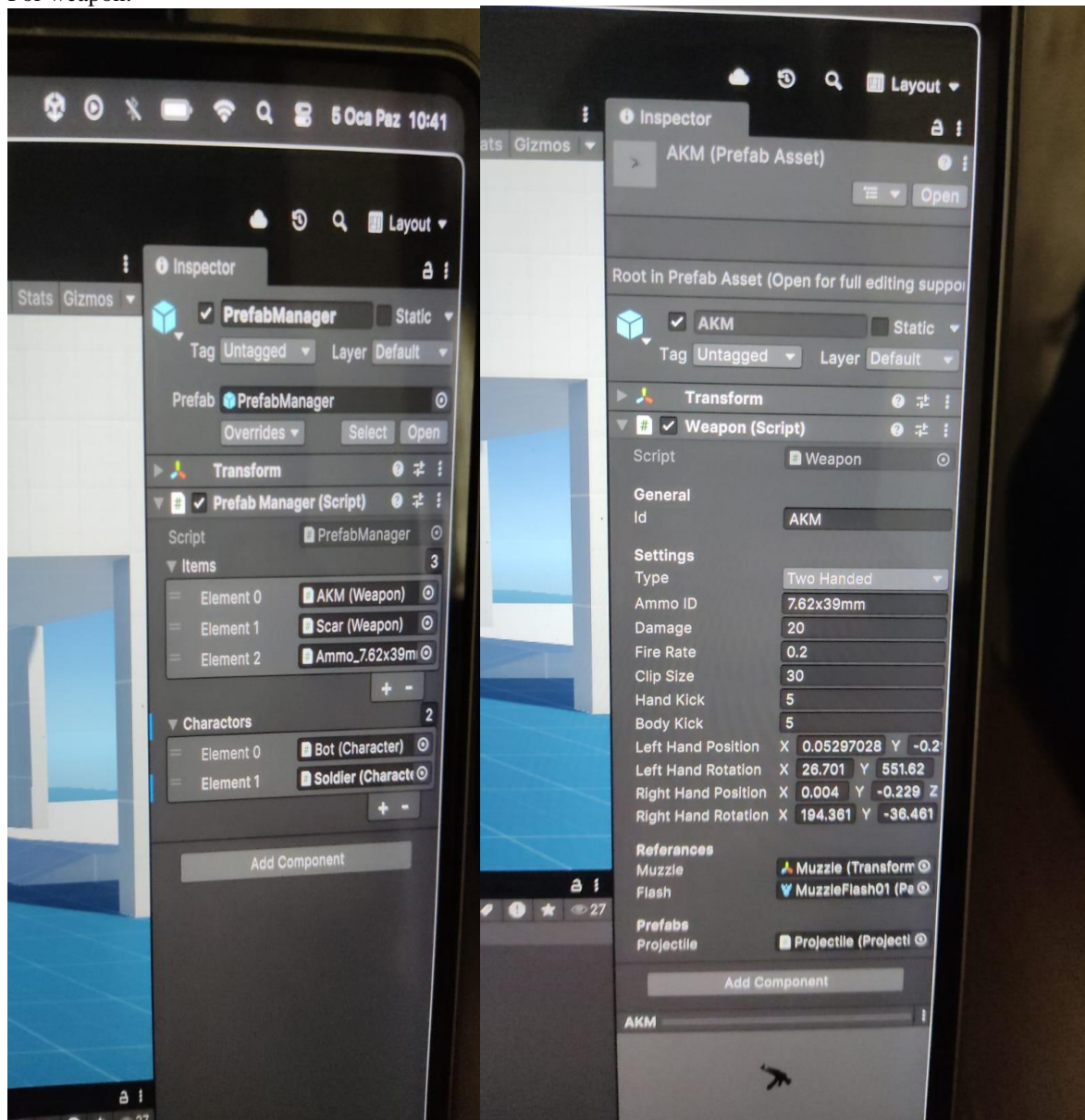
- Normal State: The bot uses a script that enables random movement to move randomly on the NavMesh when it cannot see the player. The AI-driven bot moves aimlessly through the surroundings.

Detection State: The script that controls the bot's random movement is turned off once the player gets within ten meters of it; instead, a different script is triggered to cause the bot to approach the player. The player can get away because this bot moves a little more slowly than the player. The script for random movement is reactivated once the player leaves the radius.

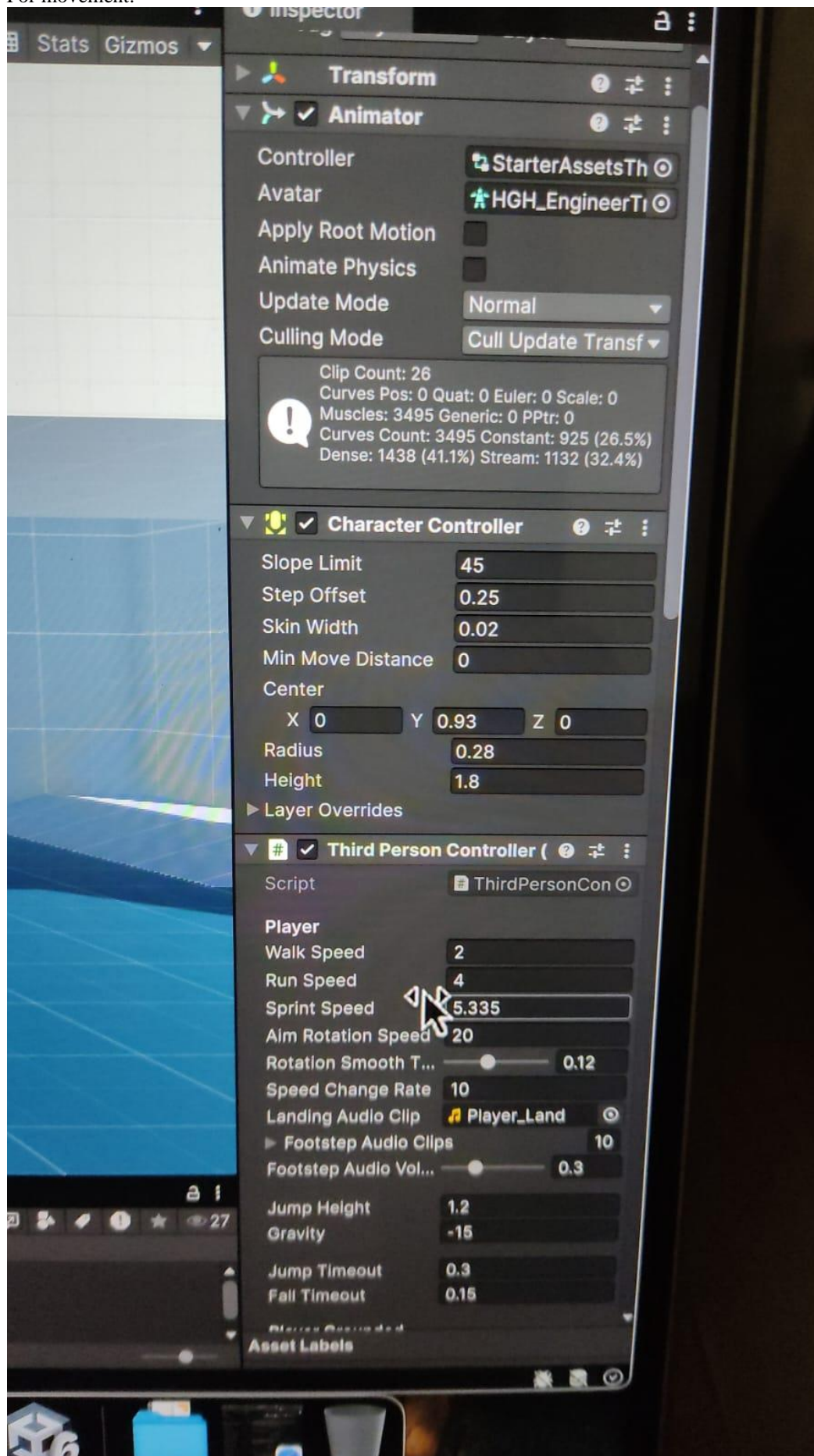
Reaction State: The Animator controls a punch animation that starts when the bot gets near 0.5 meters of the player. This animation mimics the player being punched by the bot. The player loses ten health points if the bot's collider makes contact with theirs. Until either the player or the bot kills the other, this behavior keeps happening.

End State: The bot has 100 health points, much like the player. The bot loses 20 health points for each hit when bullets from the player's weapon strike its collider. A background script deactivates the random movement script and the Animator that controls the animations, destroying the bot if its health falls to zero. The bot is eliminated from the game and its death animation is activated upon annihilation.

For weapon:



For movement:



Visual Aspects

For the visual aspects of the game, the following tools and assets were used to create the environment, characters, and overall look:

- **Heightmap:** A heightmap was used to create the basic terrain for the game. This allowed for the generation of realistic landscapes, which were then refined using Unity's built-in **Terrain Editor**.
- **Assets:** All the necessary assets were sourced from the **Unity Asset Store**. This includes a variety of models and textures that fit the visual theme of the game.
- **Roads:** The **EasyRoads** asset was used to design and implement roads within the game world. This tool allows for the creation of smooth, dynamic road paths, enhancing the realism of the environment.
- **Skybox:** A custom skybox was added to create realistic environmental lighting and atmosphere. This adds depth to the game world and complements the terrain.
- **Ocean:** A **Creent Asset** was used for the water in the game. This asset provides realistic water simulation, enhancing the visual appeal of the game's coastline and water-based areas.
- **Fog and Effects:** For the fog effect (FOG), custom designs were created in **Photoshop**. These were then integrated into the game with scripts to ensure dynamic fog generation based on the player's location or environmental changes.
- **Main Menu:** The main menu and UI elements were designed using **Unity's built-in resources**. This included creating the layout and interactions within the Unity Editor, ensuring a seamless user experience.

This combination of Unity's Terrain tools, Asset Store resources, and custom-made designs creates a visually immersive game world that aligns with the overall artistic direction of the game.

4.1 System Requirements

Minimum System Requirements:

- **OS:** Windows 10
- **Processor:** Intel Core i5 or AMD Ryzen 5 (Quad-core, 2.5 GHz or faster)
- **Memory:** 8 GB RAM
- **Graphics:** NVIDIA GeForce GTX 660 / AMD Radeon HD 7870 or equivalent (2GB VRAM)
- **Storage:** 20 GB available space (SSD recommended)
- **DirectX:** Version 11 or higher
- **Input:** Keyboard and Mouse

Recommended System Requirements:

- **OS:** Windows 10
- **Processor:** Intel Core i7 or AMD Ryzen 7 (Hexa-core, 3.0 GHz or faster)
- **Memory:** 16 GB RAM
- **Graphics:** NVIDIA GeForce GTX 1060 / AMD Radeon RX 580 or equivalent (6GB VRAM)
- **Storage:** 20 GB available space (SSD recommended)
- **DirectX:** Version 12 or higher
- **Input:** Keyboard and Mouse

4.2 Visual Content

A list of technical requirements from those in concerned with the visual aspects of the game.

This is a section that will require extensive meetings with a graphic designer.

All objects should be listed with their generic names.

- **General**
 - **File Size Restrictions:**
Each visual asset optimized for performance, with texture sizes ranging from 512x512px for smaller assets to 2048x2048px for more detailed textures. Larger assets should be compressed to balance quality and performance, ensuring they run smoothly on the target platforms.

- File Format Type

Textures: PNG is used for 2D textures and TGA when transparency is needed.

Models: 3D models are saved as FBX files, ensuring compatibility with Unity and other game engines.

Animations: Animation data is stored in FBX or Unity's native format for ease of integration.

Audio/Video: Audio files are in MP3 or WAV formats, while video clips (if applicable) are in MP4 or AVI format.
- File Quality Type

Assets created in high resolution for clarity and detail but optimized for performance. For textures, the balance between quality and performance is key, using higher resolutions only when necessary. 3D models should be low-poly where possible to maintain efficiency.
- Visual Scale

All visual elements, including textures, models, and UI elements, consistently scaled. Characters and environment objects should maintain proportionate sizes relative to one another. Unity's built-in tools can help standardize scale and prevent discrepancies between assets.
- Player Elements
 - Type of States

Default: The standard idle state of the player character.

Damage: The visual state when the player takes damage player moving little bit backward

 - Amount Animation Frames
 - Movement Animation:** 8-12 frames for walking/running to ensure fluid transitions.
 - Jump Animation:** 5-7 frames, showing the player character's leap, mid-air, and landing.
 - Damage Animation:** 3-4 frames to show the player character recoiling or flinching.
- Heads Up Display (HUD)
 - Type Icons

Health, ammo, and ability icons. These icons are clear and distinct, with different visual representations for each state (health decreasing as the player loses health, ammo count decreasing when shooting, when out of ammo not shooting).
 - States

Normal: Standard icon appearance during gameplay.

Damage: Standard icon appearance during gameplay.
 - Font Type

A clean and simple sans-serif font like **Roboto** or **Arial** is chosen for readability. Font weight and size should adjust dynamically based on screen size to ensure legibility across devices.
- Antagonistic Elements
 - Type of States (Default, Damage, Destroyed, ect.)

Default: The basic state in which the antagonist patrols or idles.

Destroyed: Once the antagonist is defeated, they will play a dying animation.
 - Amount Animation Frames

Idle/Patrol Animation: 4-6 frames to maintain fluidity during movement.

Attack Animation: 6-8 frames to showcase the enemy's attack sequence.

Damage Animation: 3-4 frames showing the enemy flinching or reacting to damage.
- Global Elements
 - Background/Texture/Tiles

Backgrounds are static or dynamic 2D images, depending on the scene. They are designed to seamlessly blend into the environment. Tiles for terrain and floors are also created to seamlessly tile, ensuring no visible seams during gameplay.
 - Font Type

For in-game text and menus, a consistent sans-serif font like **Arial** or **Roboto** will be used for clarity. This ensures that all text is easily legible across different screen sizes and resolutions.

4.3 Audio Content

General:

- **File Size Restrictions:** Audio files are optimized for game performance.
- **File Format Type:** Audio files are in **WAV** format for sound effects and **MP3** format for music.
- **File Quality Type:** Sound effects are high-quality to match the game's visual standard.

Player Elements:

- **Type of Sound Effects:**
 - In-game sound effects for the player include footsteps, jumping.
- **Device Vibration:** Not specifically mentioned.

Antagonistic Elements:

- **Type of Sound Effects:**
 - The game's antagonistic elements, such as enemies, will have footstep sounds when they are close to the player.
- **Device Vibration:** Not specified.

Global Elements:

- **Ambient Music:** Not specified.

Splash Screens:

- **Ambient Music:** Not specified.

Menus:

- **Type of Sound Effects:** The menu features background music with variations, and there is an option to adjust the sound settings.

4.4 Programming Content

- **General:**
 - **Requirements:** The game is developed using **C#** scripting language, running within the Unity game engine.
 - **File Size Restrictions:** 1.14GB.
 - **File Format Type:** exe.
 - **Specify Coding Conventions:** Follow Unity's **C#** coding standards, with clear naming conventions for variables and functions. Use comments to document key parts of the code.
 - **Language/Device Restrictions:** The game is developed using **C#** in Unity and is intended for platforms supported by Unity, including PC.
 - **Screen Type:** The game supports only full screen.
- **Player Elements:**
 - **Type of Event:**
 - The player's ammo count is tracked and updated based on actions, such as shooting and reloading. This will trigger events to decrease ammo and update UI elements accordingly.
- **Antagonistic Elements:**
 - **Type of Event:**
 - Bots are programmed to follow certain behaviors, such as chasing the player when within range and attacking. When a bot is killed, it is counted, and once 11 bots are eliminated, the game ends.
 - If the bot kills the player, the game will end as well.
- **Global Elements:**
 - **Type of Event:** The game keeps track of the player's progress, such as how many bots have been killed and whether the player survives or is defeated. Events will trigger the game's end when conditions are met (e.g., 11 bots killed or player killed by bots).
- **Splash Screens:**
 - **Type of Event:** Not specified.
- **Menus:**
 - **Type of Event:**
 - The menu includes interactive options, such as starting the game, adjusting settings and exit button.
 - **Type of Options:** The options menu includes sound settings

4.5 Code Structure

Player Object:

The player character is the central object in the game, with its health, speed, abilities, and status being tracked in real-time. Player inputs (movement, jumping, shooting) trigger changes to these properties. The player interacts with enemies, neutral objects, and the environment.

Enemy Objects:

Each enemy object (soldier, NPC) has its own set of properties, including health, damage, and behavior (patrol, attack). Enemies track the player's position and perform actions based on proximity (chasing and attacking). The enemies' status can change when they take damage or are dead.

Environment Objects:

These include static and interactive objects in the game world, such as obstacles, terrain. These objects may affect the player's movement or abilities.

Game Systems:

Several core systems manage interactions:

- **Physics Engine:** Responsible for player and enemy movement, collisions, gravity, etc.
- **Combat System:** Manages interactions between the player and enemies, including damage calculation, health updates, and effects from weapons.
- **AI System:** Controls enemy behavior, such as patrolling, chasing, and attacking the player.
- **HUD System:** Displays vital game information, such as health, ammo count, and remaining enemies, on the player's screen.

List of Specified Functions/Routines:

1. **Player Movement:**
 - **Function:** Moves the player based on input.
 - **Routine:** Calculates new position based on speed, checks for collisions, and updates the position in the game world.
2. **Player Health System:**
 - **Function:** Tracks and updates the player's health.
 - **Routine:** Reduces health when the player is damaged, checks if health reaches 0 (player death), triggers appropriate game over screen.
3. **Enemy AI (Chase and Attack):**
 - **Function:** Detects the player's presence and decides actions.
 - **Routine:** Uses distance checks to determine if the player is within range, then chases and attacks.
4. **Combat System:**
 - **Function:** Handles interactions between weapons, enemies, and players.
 - **Routine:** Detects when the player's weapon hits an enemy, calculates damage, and updates health accordingly.
5. **HUD/Interface Updates:**
 - **Function:** Updates the display of health, ammo, etc., on the screen.
 - **Routine:** Constantly checks for changes in the player's health, minimap, remain target, and ammo's information updates the on-screen elements accordingly.

6. Sound/Audio System:

- **Function:** Plays sound effects based on game events.
- **Routine:** Triggers appropriate sound effects for actions (footstep, environmental sounds) based on game events and user actions.

Order of Module Development:

Core Game Engine Setup:

W: Move Forward A: Move Left ALT: Focus-Mode SHIFT: Run

S: Move Backward D: Move Right SPACE: Jumping

MOUSE SCROLL: Changing Weapons M: MAP R: Reload

LEFT CLICK: Aiming RIGHT CLICK: Shoot

Camera Setup:

- **Free Camera Mode:** By default, the camera is in a free-roaming mode allowing the player to view the environment from various angles.
- **Focus Camera Mode:** When the player right-clicks to aim, the camera shifts to focus on the player's aim direction, keeping the character in the center of the screen. This mode helps in targeting and shooting enemies with precision.

Player and Enemy Interaction:

Once the basic gameplay mechanics are in place, develop the combat system, including the health and damage mechanics for both the player and enemies. Also, implement enemy AI behaviors, such as patrols and attacks.

HUD and User Interface:

Implement the HUD, showing the player's stats, ammo, minimap and information about enemy counts.

Audio/Visual Effects:

After gameplay mechanics are solid, integrate sound effects that helps the player to hear footsteps from Solider.

Final Polish and Debugging:

After all core systems are implemented need to do Final test and find bugs and something like that errors. After solving all problems. Game ready to launch.

4.6 Concerns and Alternatives

1. Concern : Limited Environmental Interaction

Issue: The game currently lacks direct interaction with the environment, which may reduce immersion and player engagement.

Alternative: Introduce simple interactive elements such as opening doors, picking up objects, or activating switches. These interactions can be gradually added in future updates to enrich the game world without overwhelming the core gameplay.

2. Concern: Limited Combat AI Behavior

Issue: The AI's combat behavior is currently basic, with limited interaction beyond attacking the player when close.

Alternative: Enhance AI tactics by adding behaviors like taking cover, flanking, or retreating when low on health. This would improve the challenge and make encounters more dynamic.

4.7 Resources

Applications and Equipment

This section outlines the acceptable applications and assets used during the development of the game, with a focus on legal considerations to ensure compliance with software licenses.

1. **Unity Asset Store:**
 - A variety of pre-made assets, including models, textures, and animations, were sourced from the Unity Asset Store. These assets are licensed for use within the game and are fully compatible with Unity's development environment.
2. **EasyRoads Asset:**
 - The EasyRoads asset was utilized to create roadways within the game environment. It allows for the efficient generation and modification of roads, enhancing the world's realism and navigability.
3. **Skybox:**
 - A skybox was added to the game to create dynamic and immersive environments. The asset is licensed for use within Unity and helps in establishing various weather conditions and time of day in the game world.
4. **Ocean Asset:**
 - A "Crent" ocean asset was used to simulate water within the game. This asset provides realistic water effects, including waves, reflections, and underwater visuals. It is sourced from Unity's asset library and complies with its terms of use.
5. **Photoshop:**
 - Photoshop was used for creating and editing custom FOV (Field of View) designs and other textures for the game. All designs were created in-house and are fully owned by the development team.
6. **Unity's Built-in Tools:**
 - The main menu and various in-game UI elements were created using Unity's native UI system, ensuring compatibility and efficiency in integrating with the overall game structure.

This list ensures that all tools and assets used during the game's development are legally compliant and properly licensed for their intended purpose.

4.8 Technical Matrix

Category	Subcategory	Details
Visual Content	General	
	File Size Restrictions	Texture sizes: 512x512px to 2048x2048px. Larger assets compressed for balance between quality and performance.
	File Format Type	Textures: PNG/TGA (transparency); Models: FBX; Animations: FBX/Unity native; Audio: MP3/WAV; Video: MP4/AVI.
	File Quality Type	High resolution, optimized for performance. 3D models should be low-poly where possible.
	Visual Scale	All elements scaled

Player Elements	Type of States	consistently, using Unity's tools for standardization.
	Animation Frames	Default (idle), Damage (recoiling/flinching). Movement: 8-12 frames; Jump: 5-7 frames; Damage: 3-4 frames.
HUD	Type Icons	Health, ammo, ability icons with dynamic states (e.g., health decreasing).
	Font Type	Sans-serif fonts like Roboto or Arial, dynamically adjusting size for readability.
Antagonistic Elements	Type of States	Default (idle/patrol), Damage (flinching), Destroyed (dying animation).
	Animation Frames	Idle/Patrol: 4-6 frames; Attack: 6-8 frames; Damage: 3-4 frames.
Global Elements	Backgrounds/Textures	Seamless 2D static or dynamic backgrounds and tiles for terrains/floors.
	Font Type	Consistent sans-serif fonts like Arial or Roboto for in-game text and menus.
Audio Content	General	
	File Size Restrictions	Audio files optimized for performance.
	File Format Type	WAV for sound effects, MP3 for music.
	File Quality Type	High-quality sound effects consistent with visual standards.
Player Elements	Type of Sound Effects	Footsteps and jumping sounds.
Antagonistic Elements	Type of Sound Effects	Enemy footsteps near the player, attack sounds.
Menus	Type of Sound Effects	Background music with adjustable sound settings.
Programming Content	General	
	Requirements	Developed in Unity using C#.
	File Size Restrictions	Final build size is 1.14GB.
	File Format Type	Executable (.exe).
Player Elements	Coding Conventions	Unity C# standards, clear naming conventions, and comments for documentation.
	Language/Device Restrictions	Unity-supported platforms using C#.
	Screen Type	Full-screen only.
	Type of Event	Ammo count tracked and

Antagonistic Elements	Type of Event	updated during shooting/reloading, with dynamic UI updates. Bots chase and attack players when in range. Game ends when either the player dies or 11 bots are killed.
Global Elements	Type of Event	Tracks player progress and triggers game end based on conditions (e.g., bots killed or player defeated).
Menus	Type of Event	Interactive options for starting the game, adjusting settings, and exiting.
	Type of Options	Includes sound settings adjustment.

5 Appendix A

In this appendix there are lists for a games Taxonomy, Genre, Elements, Contents, Theme and Style. These lists are far from complete but are useful for further examples.

5.1 Taxonomy

- Simulation
 - Story
 - Play
 - Chance
 - Fiction
 - Non Fiction
- Story
 - Simulation
 - Play
 - Chance
 - Fiction
 - Non Fiction
- Play
 - Story
 - Chance
 - Simulation
 - Fiction
 - Non Fiction

5.2 Genre

- Basic
 - Adventure
 - Arcade (any "twitch" element)
 - Construction & Management
 - Puzzle
 - Role-play
 - Simulator
 - Strategy
- More Specific
 - Abstract (i.e., Arkanoid)
 - Adventure
 - Arcade (very generalised category)
 - Beat-em-up
 - Flight Sim/Space Sim
 - FPS (first-person shooter)
 - MMOG(Massive Multi-player Online)
 - MMORPG (role playing game)
 - MMORTS (real time strategy)
 - MMOTBS (turn base strategy)
 - Platform
 - Puzzle
 - Racing
 - RPG (role-playing game)
 - RTS (real-time strategy)
 - Shoot-em-up (scrolling shooter)
 - TBS (turn based strategy)
 - Trading

5.3 Game Elements

- Alignment
- Catch
- Chase
- Collecting
- Combat
- Cooperation
- Dodging
- Escape
- Fighting
- Forbidden Act
- Hiding
- Jump
- Luck
- Maze
- Nursing
- Obstacles
- Puzzle
- React
- Resource Management
- Seeking
- Shooting
- Story Comprehension
- Target
- Trading
- Trivia
- Timing

5.4 Content

- Action
- Drama
- Erotic
- Horror
- Humor
- Pure Play
- Realism
- Thriller

5.5 Theme

- Abstract
- Crime
- Fantasy
- Noir
- Porn
- Sci-Fi
- Spy
- War
- Western

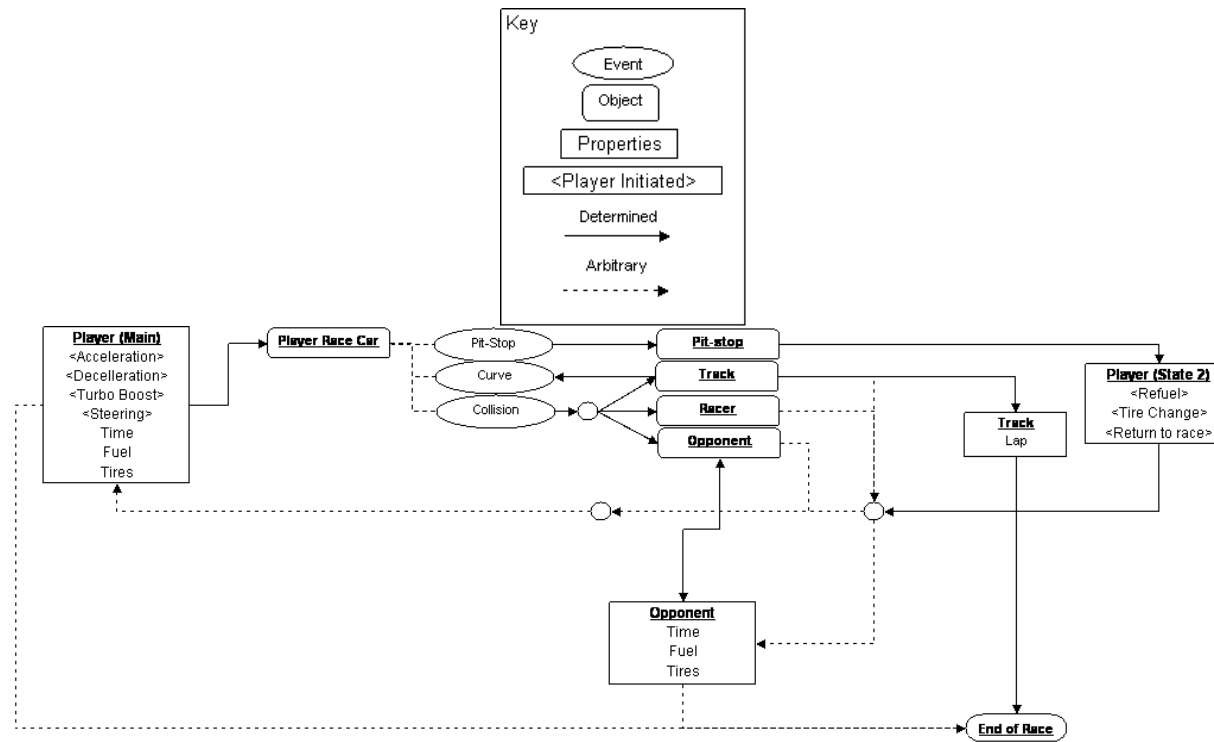
5.6 Style

- Abstract
- Cartoon
- Manga
- Old School
- Realism

6 Appendix B

Object	Properties			
	<i>Unit type</i>	<i>Hit Points</i>	<i>Damage</i>	<i>Bullet speed</i> <i>Movement</i>
<i>Basic Enemies</i>				
	Cannon -01	30	100	Medium NA
	Cannon -02	60	150	Slow NA
	Static rocket launcher	70	700	Medium NA
	Flock 1	15	500	NA Medium
	Flock 2	15	35	Slow Medium
	Medium 1	80	350	Fast Slow
	Medium 2	90	400	Medium Slow
	Rail	120	600	NA Slow
	Small 1	25	200	Fast Fast
	Small 2	30	210	Fast Fast

7 Appendix C



8 Appendix D

