Design document

The Gluenadier

# The Gluenadier

A satisfying 3rd person action game, where the player takes control of the nimble gluenadier, surrounded by waves of enemies. Throw grenades with different effects into your enemies and combine their overlapping effects to defeat your enemies, while jumping and dodging their attacks on levels with complex geometry.

# What I plan to Learn

Design skills:

* Creating “juicy” game actions
* Polish and optimization practices
* Basic level design process and good practices
* Game sound mixing

Technical skills:

* Learning to read code in C++ and write small commands
* Working with particle systems
* Getting accustomed to UE5
* Working with Fmod

Soft skills

* Self promotion via game updates
* Networking

# Target Demographic

Players that enjoy skill-based action games would find the game interesting due to its reliance on precision movement and quick decision making due to constant enemy pressure.

Additionally, the colourful and satisfying bomb throwing mechanic would attract people who would enjoy creating chaos and destruction.

The project would also serve to show my design skills to potential recruiters

# Mechanics

## Design Pillars

**Grenade puzzles**Grenades in the game rarely do direct damage but instead have many unique effects on the enemies, the player, and the terrain. Combining these effects between each other, level elements or friendly fire from enemies is the main way to defeat enemies in the game and experimentaiton is highly encouraged.

**Constant Movement**  
The player must always be in action. The player must always think on the fly. Dodging, running, exporing, luring and maneuvering their way around the level setups and groups of enemies, while also quckly coming up with a plan to defeat them all is a big part of the skill appeal of the game.

## **Movement**

The gameplay tempo dictates the player be almost constantly attacked and make decisions, while dodging. The movement options must give the player the ability to dodge incoming fire and enough control to allow the player to bait enemies into specific situations to get the upper hand.

**Player speed**

The player is one of the fastest moving actors in the game. This is necessary to give the player an ability to avoid projectiles and slower enemies, but the speed must not be too great as to make the dodging too easy. The player’s naturally high movement is one their main tool of avoding attacks and setting up great grenade combinations.

**Jump**

The player has an acces to a jump and double jump. This is made to give the player movement some verticality which would assist with setting up grenade throws, interacting with the level architecture and dodging enemy groups.

## **Grenades**

A second part of the gameplay ying-yang, the grenades are the player’s main way of influencing the game world. The grenades have different effects that affect either the enemies or the game world but very few of them deal serious amount of damage. In order for the player to prevail, they will need to combine the grenade effects to set up friendly fire, environmental kills or get damage multipliers. Oh, and another thing: The grenades have an explosioin radius and it **will** affect the player character if caught in the blast. This is one point of synergy with the movement system.

**Sandbox system**

The grenades have many different effects and many of those would overlap with each other by interacting with movement and health components. **Every grenade has multiple effects** and since no grenade deals high amounts of flat damage, the players have to experiement with different grenades to defeat their enemies.

For example:

Explosion grenades deal a small amount of flat damage but also pushes enemies away.  
Flat damage can be used in combination with a grenade that applies a damage modifier  
Pushback can be used to score environmental kills

**Grenade choosing system.**

There are many grenades that can be used by the player, but they can’t simply choose which one they want to use. Instead, the pllyer has 2 grenade slots that can store one type of grenade each.

One slot contains can be chosen and switched at the player’s command, other switches every few greandes. The trick is – the player can see what the upcoming grenade is going to be. This gives the player **a way to plan** spectacular grenade/environment/friendly fire setups that can wipe out scores of enemies.

**Movement grenades**

Grenades are also an essential part of the movement system as some of them provide additional movement abilities.

Teleportation grenade – allows for quick repostioning and croud control

Blast grenade – allows grenade jumping at the cost of some health

## Enemy spawner

Enemy spawner is a blueprint that keeps track of the number of enemies in the level and spawns new ones to keep up the pressure on the player.

It will do so by running a check on a timer and adding new enemies if needed.  
The enemies are not endless (unless specifically chosen to be) and come from an overall bank. Depletion of the enemy bank is one of the victory conditions.

# Visual Style

To make the game interesting for the players, I need to create some sorts of visuals.

Creating assets myself is outside the scope of this project due to my current unfamiliarity with the process; therefore I need to be mindful of what resources I have at my disposal to achieve that goal:

* Asset packs for UE5(I intend to make use of free assets but have some limited monetary resource if necessary)
* Help from VA students and teachers from Buas
* Help Online

# Audio Vision

Goals of game audio:

* Support the “juicy” feeling of game mechanics
* Convey important information
* Create and support a gameplay tempo

To achieve that, I plan to utilize Fmod and FLStudio software.

FL Studio is required for asset creation. It may not be the best choice but I am familiar with it and must focus on other aspects of the project.  
Fmod is a new (for me) piece of software that handles audio mixing in games and gaining experience with it would be very useful in my future career.

# Technical requirements

# Creative Requirements

The sandbox element of the game poses a creative, design and balancing challenge. The core of the design issue stands in striking the balance between giving the player multiple options and space for experimentation and keeping up the high intencity gameplay tempo.

# Proof of concept

The proof of concept’s purpose is to see if the combination of high tempo movement and a grenade combining mechanics work together well enough to create fun gameplay.

To achieve that I would need to set up several elements:

* Enough Grenades to allow for grenade combining mechanic
* Grenade switching mechanic
* Player movement and fire
* At least 1 enemy type for different health variations
* Continious enemy spawner blueprint
* Victory condition
* 2-3 Level design elements that can be combined with grenades
* Health and movemement components that are affected by the grenades and environment

**Reflection at the end of Q1**

Game elements set up (Green – task accomplished, Orange – partial completion, red – not completed)

* Enough Grenades to allow for grenade combining mechanic
* Grenade switching mechanic
* Player movement and fire
* At least 1 enemy type for different health variations
* Continious enemy spawner blueprint
* Victory condition
* 2-3 Level design elements that can be combined with grenades
* Health and movemement components that are affected by the grenades and environment

I did not achieve all the goals I’ve set up for this project. I had to settle for 70% completion due to several reasons:  
1) I have had to settle with compromising on the project’s complexity in order to begin the level design process and have some room left for iterations.

2) I underestimated the time required for me to find direction in my level design process. Additionally, it also exposed the weaknesses of my existing system design, which resulted in re-making already existing work.

**The cause** of those developments was the frequent diversion of focus due to internship preparations. Since my internship requires a work visa for germany it is imperative for me to start the visa application process as soon as possible, therefore, every time I received a document request from the studio it automatically became the highest priority task for that day.

**My mitigation strategy** consisted of developing a more flexible work schedule and keeping several free time slots in the upcoming days in the learning log to be able to push whichever task I was occupied with on the next day.

**Reflection.**The strategy’s flaw was the fact that I could not know when the studio was going to reply, which made further planning difficult. I am dependant on other people in this process and there is a limit to how much I can speed up the process by myself. I tried to deal with this by developing some sort of a mitigation strategy but after a certain point I just had to accept that it achieves nothing but causing frustration. This exposes the degree to which I am unsettled by reliance on others on way to achieving my goals and would need to be adressed.

Action points:

1. Accept the role of other people in this process.

Whether the internship materialises or not would depend largely on the work of the studio and their HR department. I need to accept that there is nothing I can do to make them work faster and that there is a chance that it could fall apart completely through no fault of my own. I find such a prospect severely disheartening but there is nothing I could do, and continued stress only reduces my productivity.

1. Prepare for the worst.

If the internship would fail to materialize, I would need a plan on how to proceed with my year further. I have a great personal project that can be easily scaled up but setting up some basic planning would be worth the time in the upcoming quarter. It would also aliviate some of the stress that the prospect of a failed internhsip creates in me.

1. Test earlly, test more

The unfinished state of my build would hardly impress anyone, but even short testing sessions with just a few people can bring quite a bit of value to the project.

# Block A (Q1)

ILO evidencing highlights:

1.1

A screenshot of a computer

Description automatically generated with medium confidence

A good evidence example would be to create a storyline for several features that demanded research. Like: Components, AI, Level design and simillar ones. I’ll need to arrange the evidence for them in a way that shows that my research process was structured and guided.

Text

Description automatically generated

The second part of the evidencing document that I’ll make for the previous ILO would show how I implemented that knowledge in a sustainable and durable way.

# Changelist

September 20th, 2022 – document created

October 31st, 2022 – “Block A” section added with ILOs 1.1 and 1.2

November 1st, 2022 – Updating the document with an updated vision on the sandbox nature of the genade mechanic, added the creative requirement entry about the sanbox balancing. Added text to the proof of concept. Added enemy spawner text but it’s not structured enough.

November 14th, 2022 – Added a reflection section to the proof-of-concept chapter.