Making of Document GD S3

MAKING OF GAME DESIGN SEMESter 4

Damian van Veldhoven

2018

Inhoud

[Fase 1 – Concept 2](#_Toc4090209)

[Mechanics 2](#_Toc4090210)

[Dynamics 2](#_Toc4090211)

[Aesthetics 2](#_Toc4090212)

[Fase 2 – Design 2](#_Toc4090213)

[Mechanics 3](#_Toc4090214)

[Dynamics 8](#_Toc4090215)

[Aesthetics 9](#_Toc4090216)

[Fase 3 – Deliverable 10](#_Toc4090217)

[Mechanics 10](#_Toc4090218)

[Dynamics 10](#_Toc4090219)

[Aesthetics 10](#_Toc4090220)

[Sources 10](#_Toc4090221)

# Phase 1 – Concept

We decided on making a bullet-hell roguelike, where the premise of the game is based around the frantic shooting of mobs whilst clearing rooms and dodging lots of bullets. With the addition of smooth player and camera movement, the game will feel intuitive and finetuned, with a silver lining of strategy in the approach of attack. The inspiration of the game is “Binding of Isaac”.

## Mechanics

We wanted to have a deep and extensive combat system that would allow us to really challenge ourselves in our approach and design. For this we decided to make a game where we could

* Use movement abilities, such as dashing, jumps.
* Have multiple equipable weapons, each with different attacks.
* Have multiple mobs, each with different behavior
* Have random floors with easily customizable textures
* Have shop-rooms where players can buy items, with a well thought-out money system

It was also important for us that players are able to pick up weapons that are found throughout the dungeons, and that the floors are made up of rooms that lock themselves up after the player has entered, so that players are forced to clear the room before they can continue.

## Dynamics

We want to make monsters that have special behavior(s) depending on their kind, for example ranged or close-range monsters. This forced a strategic attempt from the player to effectively kill the monsters. The monsters have an AI script that allows them to make decisions based on their surroundings, which in turn gives the player a bigger challenge in killing them.

The weapon system will have a lot of guns such as pistols, rifles, shotguns and lasers, and because of this the player has a wide variety of methods to kill the monsters. These weapons all have their own use in a specific situation and provide the option to use strategy.

The rooms have purposes, such as a regular killing room, a shop room, or a boss room. Depending on what room the player is going to enter, a choice on what do has to be made. This forces the player to think about their next step.

## Aesthetics

The rooms and monsters are in a fantasy-dungeon type style, but do not necessarily adhere to this theme. Guns and odd monsters are going to be in the dungeons although they are not found there in the real world. This allows us to be creative in the process of creating monsters, weapons and rooms.

The drawing style of the game will be much like the drawing style from the Binding of Isaac, giving the player the feeling that they are in a dungeon where something is not quite right. For reference: 

# Phase 2 – Design

All the content shown in the figures is (possible partly) created by me.

## Mechanics

We made a class diagram (figure 1) for our game, which quickly explained how the weapons, player and monster work together in the code. Because we did this, we felt like we would be better prepared for the coding challenges up ahead.

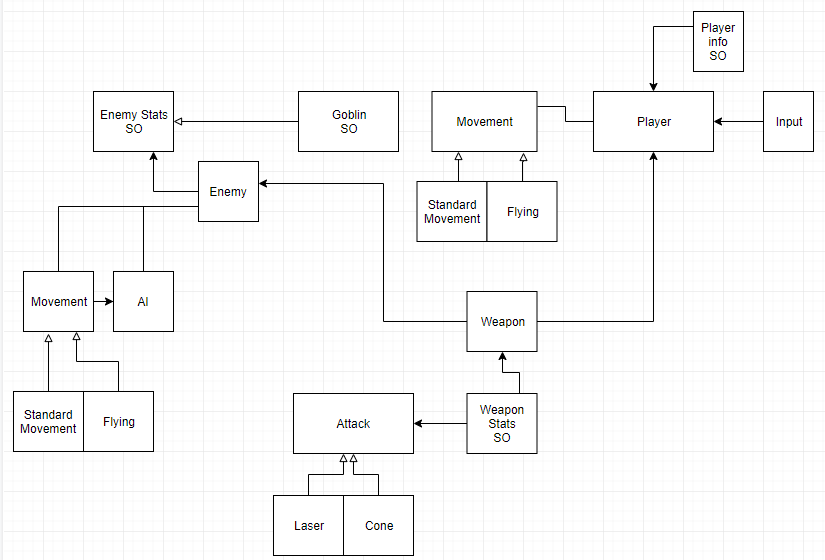


Figure 1

With this we started out making the player, which has the ability to look at the mouse cursor (which will later be the Xbox controller) (Figure 2). We chose to start out with the mouse cursor as a tool for aiming because this allowed us to easily debug the game, because we didn’t need a controller plugged in, and it was easier to implement.



Figure 2

After this is started looking for a script that allows AI to be created, and once I found a suitable script I ended up with a working AI (figure 3). The AI follows the target, but goes around the designated walls, which is perfect for our project. We needed this AI script so we can make responsive enemies for our rooms, and since the killing of the enemies is a major part of the game we wanted the enemies to have good pathing.

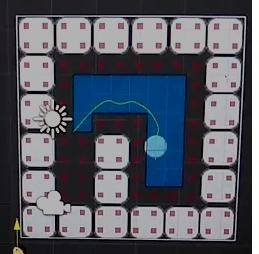
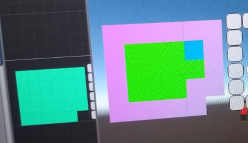
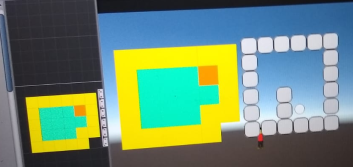


Figure 3

After this is started out with the rooms, they had to be made in such a way so that the textures are easily replaceable, so that we can recycle them for separate floors. I made this as a starter (figure 4, 5).



After creating the rooms I started with the generator, because this seemed like the biggest challenge which we could do right of the bat. The algorithm I wrote placed the rooms in a random manner, but they still had to be connected, so for this I made a node system which placed the exits of rooms only where they were supposed to be, and replaced the exits which were unused with walls. (figure 6).

The floors don’t have a counter which keeps track of how many times a room is placed, but instead the algorithm relies on the fact that once there are a lot of floors, the chance of a room being spawned twice next to each other becomes very small. This results in us spending less time on the generator algorithm, and more on other things.

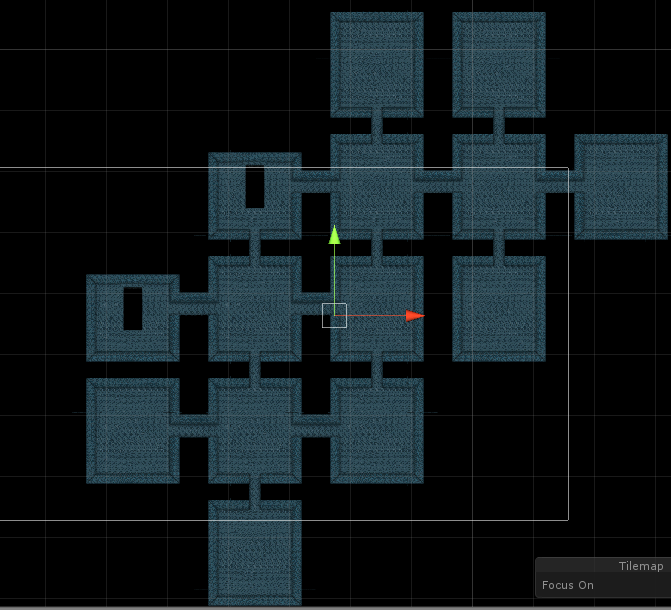


Figure 6

After this I worked on making rooms which could only be spawned in with a specific configuration. See the room marked in red (Figure 8). Doing this allowed me to make rooms with very special features, which will never look weird because I decide in what configuration they spawn.

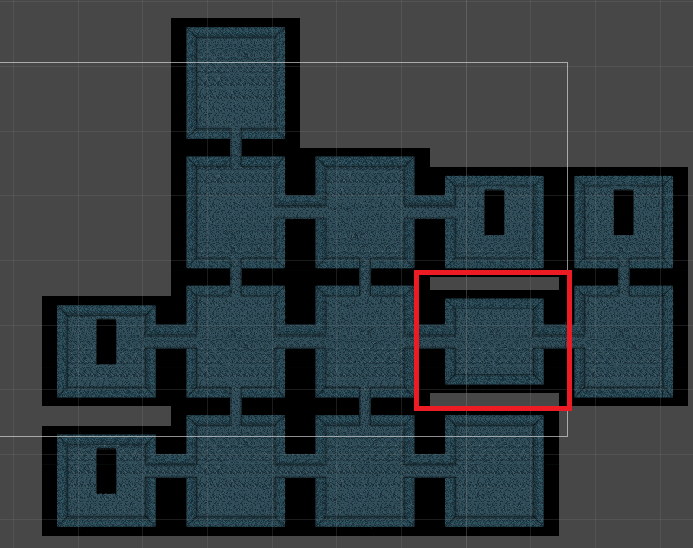


Figure 8

After this came the spawning of the mobs and the creating of the gates for the separation of the rooms. Because Lucas had already provided me with mobs to spawn in, it was easy for me to add them to the rooms (figure 9). The mobs have preset spawn-points in the rooms, and therefore will never spawn in a weird location. It also allows me to make special formations in rooms so that the player has to behave a certain way in a room. The mobs are enabled and visible once the player enters the room because this is better for the optimization. They are spawned in together with the floor though.

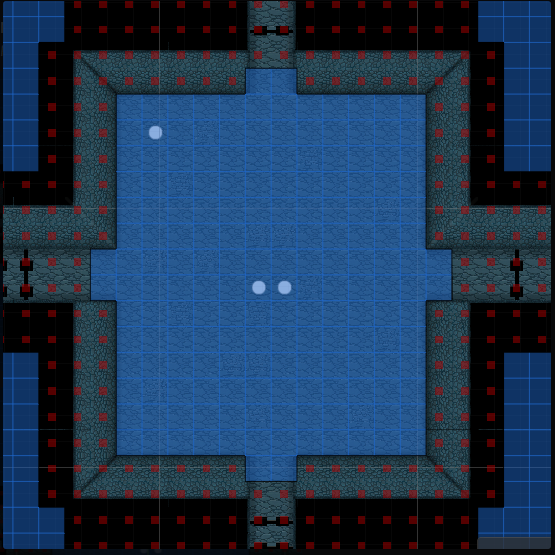


Figure 9

The placing and creating of the gates was finished, and after a lot of debugging I also got them to close when a player entered the room (Figure 10). This causes the player to be unable to leave the room which forces the player to clear the room. This also results in the player getting more gold, taking more damage and receiving more healing and/or items, which all help with the overall aesthetic of the game as well.

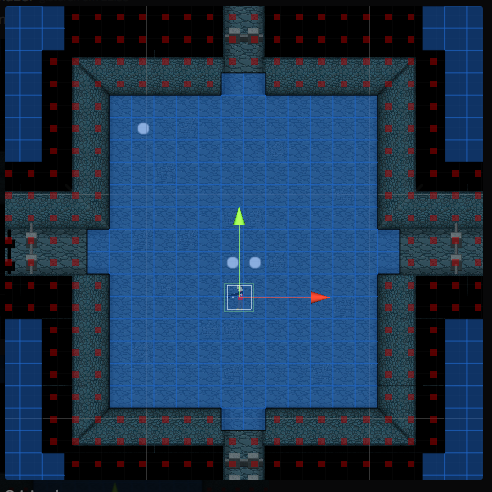


Figure 10

## Dynamics

My parts of the game did not include a lot of work on the dynamics of the game, but I did do a little AI tweaking which causes the mobs to move more naturally. The design of the rooms was also my responsibility and because of the addition of walls, so there is a forced manner in which the rooms have to be cleared (figure 12, 13).

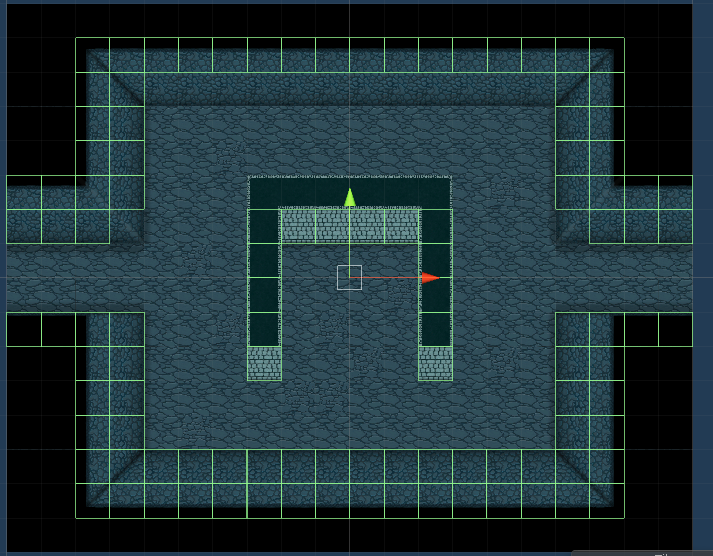


Figure 4 Figure 13

## Aesthetics

Because of the addition of the health and gold in the game, the player has to be careful with the hp that is given. This results in careful yet chaotic gameplay in which the player is forced to put themselves in difficult situations, but also has to be careful to not take damage.

Every time the player enters a room, AI spawn into the room which each have different behavior. This results in a different approach for each AI to kill the player, but also vice versa, and because of this, the rooms stay unpredictable, unlike the mobs. This allows the player to get used to the mobs’ patterns, but not to the rooms “layout”.

# Phase 3 – Deliverable

## Mechanics

## Dynamics

## Aesthetics

# Sources