



GLASS GEAR GAMES

Design Document For:

LIGHT

Written by:

Lars Joar Bjørkeland

Eivind Hobrad Naasen

Marika Sofie Grøsfjeld

Fritjof Jackwitz Berger

Version 1.0

TABLE OF CONTENT

1. Goals and Questions	Page 3
<i>Our game design goals</i>	<i>Page 3</i>
<i>Common questions</i>	<i>Page 3</i>
2. Descriptions	Page 4
<i>Project Description</i>	<i>Page 4</i>
<i>Technical description</i>	<i>Page 4</i>
3. Story	Page 5
<i>Features</i>	<i>Page 5</i>
<i>Characters</i>	<i>Page 5</i>
<i>Storyline</i>	<i>Page 5</i>
<i>Theme</i>	<i>Page 6</i>
4. Gameplay and Core Mechanics	Page 7
4.01. <i>Goals</i>	<i>Page 7</i>
4.02. <i>Actions that the player is able to perform</i>	<i>Page 7</i>
4.03. <i>Gameplay Loop</i>	<i>Page 7</i>
4.04. <i>User Skills</i>	<i>Page 7</i>
4.05. <i>Weapon: Flashlight</i>	<i>Page 8</i>
4.06. <i>Pickups</i>	<i>Page 10</i>
4.07. <i>User Interface</i>	<i>Page 12</i>
4.08. <i>Losing</i>	<i>Page 12</i>
4.09. <i>Enemies</i>	<i>Page 13</i>
4.10. <i>Gameplay variables</i>	<i>Page 16</i>

5. Art style	Page 18
<i>Design of Hero</i>	<i>Page 18</i>
<i>Weapon Design</i>	<i>Page 18</i>
<i>Design of Enemies</i>	<i>Page 19</i>
6. Environment	Page 20
<i>General Idea</i>	<i>Page 20</i>
<i>Environment of Level 1</i>	<i>Page 21</i>
<i>Environment of Level 2</i>	<i>Page 22</i>
7. Level Design	Page 23
<i>Level 1</i>	<i>Page 23</i>
<i>Level 2</i>	<i>Page 24</i>
8. Music and sound	Page 25
9. Marketing and funding	Page 25
<i>Demographic</i>	<i>Page 25</i>
<i>Platforms and Monetizations</i>	<i>Page 25</i>
10. Glossary	Page 26

1. Goals and Questions

Our Game Design Goals

What we wanted with this game was to make something out of the ordinary. We could go the easier route and make a run n' gun or a simple platformer, however what we went for was similar, with a more unique mechanic.

The Game's focus is around avoiding the enemies attacks whilst weakening them with your flashlight and then killing them with a laser, quite a violent version of tag. An added hazard is the water, as the main character is a cat, water would be the #1 enemy. So, a violent game of tag combined with the floor is lava and you got our game.

Common Questions

What is this game?

A shooter/platformer where you kill the creatures of the abyss with your trusted flashlight. This flashlight will weaken the enemy, for you to then kill them with a laser.

Why create this game?

We want people to recognize the clear inspiration from mid 90s to early 2000s platformers, but also feel like this is a whole new world for them to explore. We also wanted to explore the flashlight mechanic.

Where does this game take place?

The setting of the game is a ruined world in the wake of a cataclysmic event caused by the villain of the story; "The Abyss Mother". She wanted to create a world for her spawn, so she corrupted the Beacon of Light, the surface world's only defense against the monsters of the abyss.

The Beacon's location is in a dark forest surrounded by a dense swamp filled with ruins of a lost civilization, the builders of the Beacon.

What do I control?

You are the "Hero", a feline creature sporting a tribal mask. Her weapon of choice is a flashlight.

What is the main focus?

The player must destroy the evil abyss creatures and restore the Beacon of Light.

2. Descriptions

Project Description

The world's shield has been corrupted; the Beacon of Light. The earth has been plunged into darkness. Albino monsters are coming up from the ground and want to take over the world. You are the world's last "light bearer". Using your trusty flashlight, you set out to stop the albino monster invasion, and restore the light of the world.

The game is a dark isometric shooter, where you kill enemies by weakening them using your flashlight and then finishing them with your laser. It is a simple shooter with a spooky/scary tone. Graphically it has a simple 3D, environment covered in fog. The highlights are the "Hero" and her enemies.

Technical Description

Game should be produced in the "Unreal Engine 4" version 4.24. As a 3D application, we are going to use "Maya 2018". Gantt chart is made using "Monday.com". The integrated development environment used for programming is "Visual Studio 2019". Our target platform is PC with no additional specific requirements.

3. Story

Features

- Linear game world
- Unique weapon mechanics
- 3D graphics with simplistic textures
- Platforming

Characters

Hero

Humanoid cat using a flashlight as a weapon. Is white/beige in colour and sports a tribal mask.

Enemies

Albino-crystal monsters with red eyes flashing in the darkness. Their skin burns when exposed to light making them vulnerable to killing blows.

Storyline

The leader of the subterranean race, The Abyss Mother, has corrupted the Beacon of Light, plunging the world into darkness. This enabled her albino-crystal people to ascend to the surface world. The time of their reign is nigh. Only but a single speck of light remains in the world. A small flashlight held by a Hero stands between the subterraneans and total world domination.

The hero fights through hordes of creatures of the abyss. She, with the help of her trusty flashlight, forms a path to the Beacon of Light which is surrounded by the dark lord's red fortress. A battle between the avatar of light and the lord of darkness ensues. With the power of her flashlight at her side, she defeats the creature of terror, and re-assumes control of the world's light.

The earth has suffered greatly and needs time to heal after the conflict. With the night creatures back underground where they belong, the earth moves towards better times yet again.



Picture depicting the darkness of the world

Theme

This game is about “good” versus “evil”. It is about facing adversity. A seemingly small “girl” can make a big impact. It needs to give the player a sense of progression and impact in the world.

4. Gameplay and Core Mechanics

4.01. Goals

Short term

To defeat enemies by weakening them with your flashlight and then shooting them with a high powered laser to kill them.

Long term

Restore the beacon of light and expel the albino creatures back to the abyss.

4.02. Actions that the player is able to perform

Due to *Light* being a short game, the actions that the player is able to perform are kept at a minimum;

- Running is controlled with the “WASD” keys
- Jumping is controlled by hitting the “Spacebar”
- Aiming the flashlight is controlled by moving the mouse
- Shooting is controlled by holding down “Left Mouse Button”

Core mechanics for these actions are found under *4.11.Gameplay variables*.

4.03. Gameplay loop

The player does some simple platforming until she reaches an encounter arena. Here enemies will spawn. Enemies are killed by pointing the flashlight at them until they are fully stunned and then shooting them with a laser attack. Enemies will occasionally drop pickups that makes you stronger for a limited time enabling you to kill enemies more effectively. After all enemies in an encounter has been killed, you unlock a checkpoint at this location. Should you die, you will return here. You continue to do some more platforming until you reach another “arena”. Rinse and repeat until you reach the end of the level.

4.04. User skills

- Accuracy by aiming
- Spatial awareness
- Being able to do some simple platforming

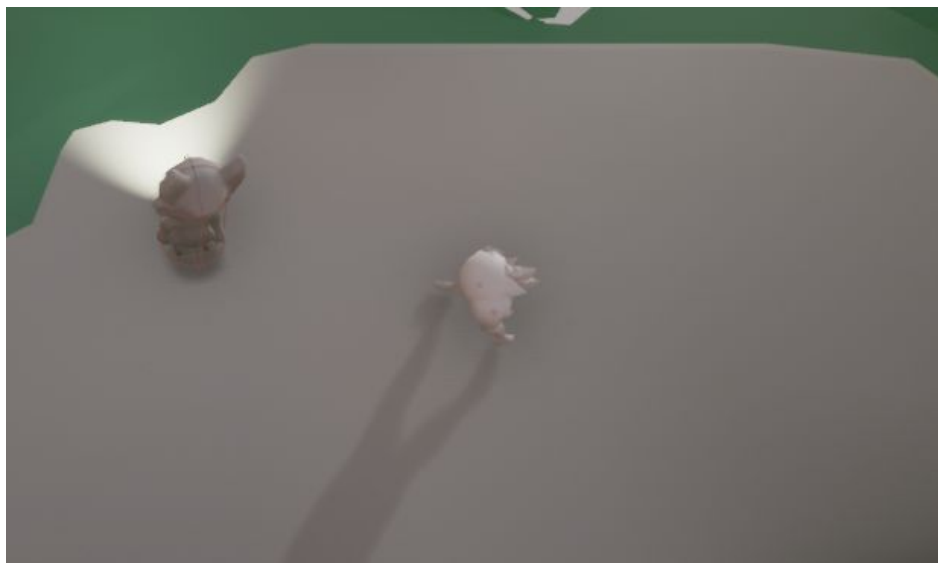
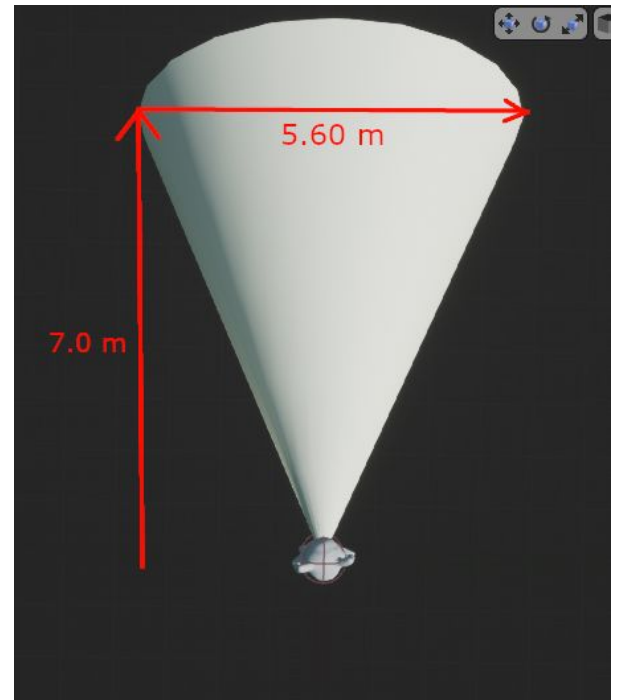
4.05. Weapon: Flashlight

The flashlight serves as both a way to light up your path, and as your main weapon. While an enemy is located within the flashlight's cone, it goes into a state of **"Weakening"**. The weakened state flows between 0 - 2 and is referred to as the **"Weakening Scale"** (0 = **"Rallied"**, and 2 = **"stunned"**). As long as an enemy is in the state **"Weakening"**, this scale increases based on time the enemy spends inside the flashlight cone.

Enemies who are in the state of **"Weakening"**, will be slowed by a certain percentage, based on its **"Weakening Scale"** value (0 = 0% slowed, 2 = 100% slowed).

When outside of the flashlight's cone, the enemy starts **"rallying"**. This reduces the **"Weakening Scale"** at the same rate as it increases, moving back towards 0. The light hitbox has a length of **7.0 meters** and a width of **5.60 meters**.

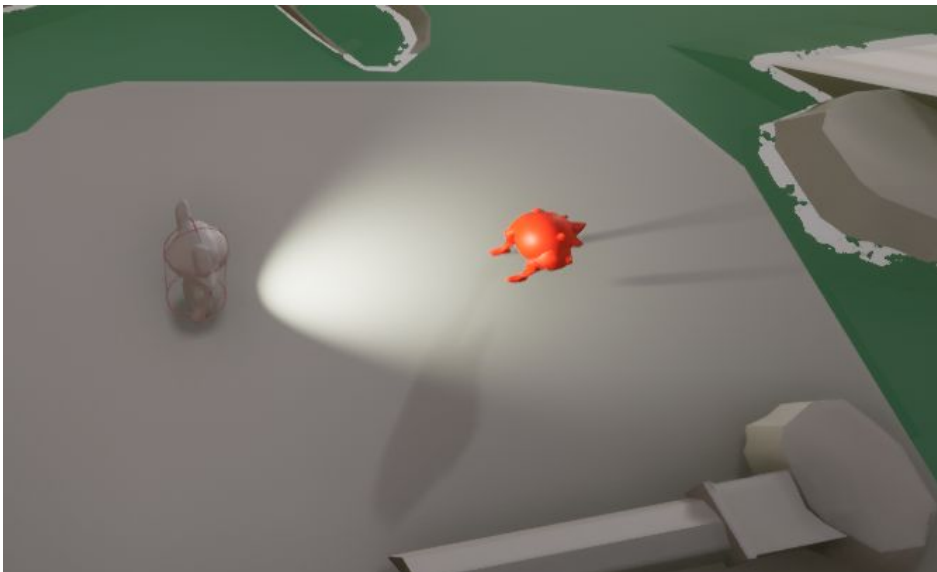
While **"stunned"** the enemy is unable to move until it becomes **"rallied"** again. Also when **"stunned"**, the enemy is susceptible to a laser (your flashlight becomes a powerful laser that kills all **"stunned"** enemies it hits). The laser is **14.0 meters** long and **0.5 meters** wide.



*Example of an enemy in **"Rallied"** state*



*Example of an enemy in “**Weakening**” state*



*Example of an enemy in “**Stunned**” state*

4.06. Pickups

All pickups will drop from enemies with different drop chances respectively. They will spawn at the location of the enemy, at the moment it is killed.

Power-ups

There are three 3 different power-ups. These will help the player defeat enemies by adding different functionalities to the flashlight. Power-ups are dropped by enemies through a power-up dropping process. Whenever an enemy dies, this process, consisting out of two stages, is executed.

First stage: one out of the possible three power-ups will be prepared for dropping. There are an equal chance for all power-ups to be prepared ($1 / 3 = 33\%$).

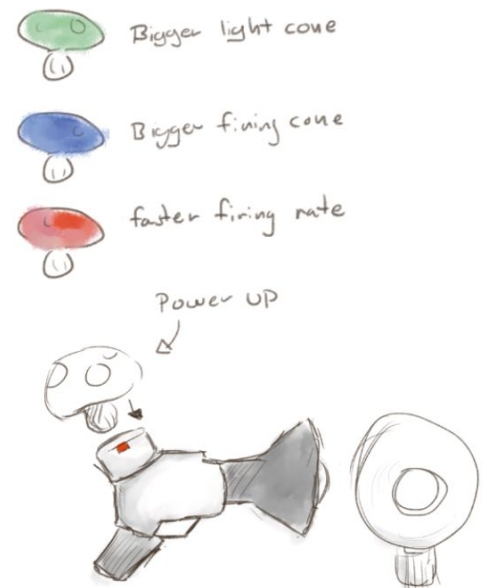
Second stage: the prepared power-up, in stage one, will be dropped. The chance of it dropping is set to 10%.

Only one power-up can be active at a time. After pickup, the powerup lasts for 20 seconds. If another power-up is picked up while already possessing one, the new power-up will override the already active one.

Visually, the power-ups come in the form of mushrooms. These will be slotted into the back of the flashlight. The current active power-up will be displayed through the user interface. The duration of the current power-up is displayed through the power-up bar in the user interface.

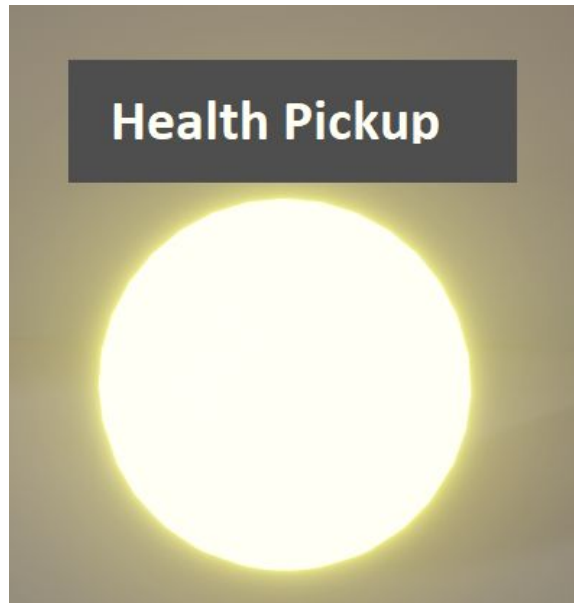
These are the three different power-ups:

1. Wider Flashlight: Increases the radius of the flashlight's cone by 2x its size. Rendered as a green mushroom.
2. Wider Laser: Increases the size of the "Laser" by 4x its size. Rendered as a blue mushroom.
3. Faster Reload: Increases the charge up speed of the "Laser" by 2x. Rendered as a red mushroom.



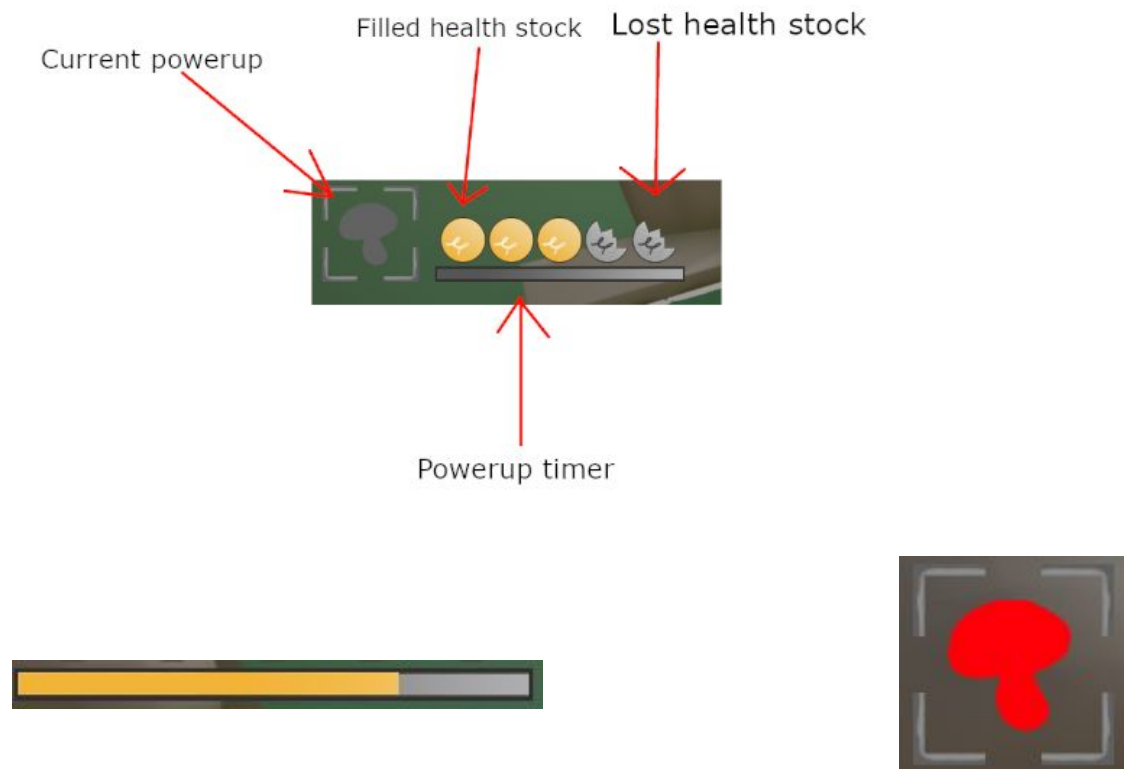
Health pickup

The “health pickup” will heal the player to full health. It will repair all broken light bulbs represented through the HUD, and have a drop chance of 4%. The “health pickup” will restore player’s health as soon as it is picked up.



Visually, the “health pickup” is rendered as a glowing orb

4.07. User Interface



(How the powerup timer looks while having a powerup)

(How the "Current powerup" case looks with a powerup)

We are keeping the user interface simple so that it does not draw away too much attention from the game world. It will be placed in the top left corner of the screen. The UI consists of three main components; health, power-up bar and current active power-up. The power-up bar keeps track of the remaining duration of your current power-up. If there are no active power-up, the current active power-up should be displayed as a silhouette of a mushroom. Your health is displayed by 5 light bulbs. For each health point you have lost, a light bulb will shatter. Shattered health stocks will be returned to normal if you restore your health.

4.08. Losing

You lose when you run out of hit points or fall into the water. Losing will result in you being returned to a checkpoint. The checkpoint is determined by which encounter you last completed.

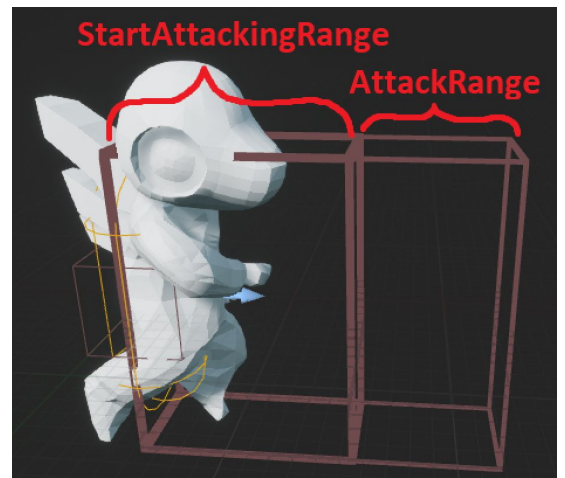
4.09. Enemies

All enemies are controlled by artificial intelligence (AI). When enemies are close enough to the player, they will start their attack animation which will deal one damage if hit, consequently shattering one light bulb.

The range of an attack differs between enemy types. To increase the amount of times enemies will land an attack, enemies will start their attack animation whenever present within sufficient range of the player. This range is different from the actual range of an attack. To illustrate this further: an enemy is equipped with two different colliders; the *StartAttackingRange* collider and the *AttackRange* collider.

If the player starts overlapping with the *StartAttackingRange* collider, the enemy's attack animation will start playing. If the player is located within the *AttackRange* collider after 0.85 seconds since the start of the animation, it should count as a hit (deal one damage to player).

There are two different enemies; the *Anklebiter* (the weaker one) and the *Herder* (the stronger one).



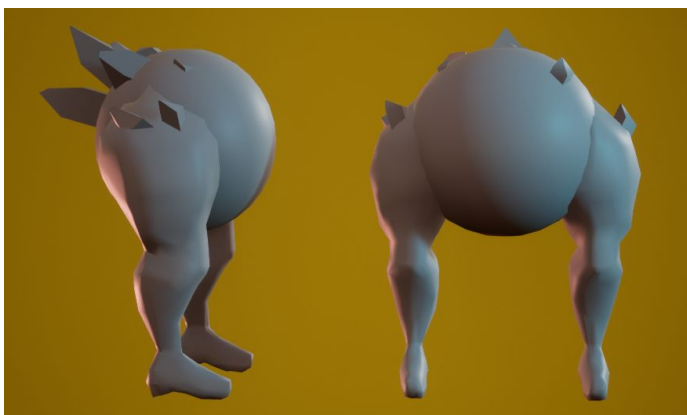
Anklebiter

The anklebiter can be thought of as a “goomba” from the “Super Mario” series. The AI is designed to make the anklebiter always run straight towards the player. Its movement speed is approximately 70% off player's movement speed.

Length of attack colliders (measured from enemy's center and straight forward):

StartAttackingRange = 0.48 meters

AttackRange = 0.64 meters



Herder

Being the more powerful one, the Herder has more advanced AI. Its movement speed is the same as player's movement speed. Due to herders being equipped with a spear, its attack range is increased.

Length of attack colliders (measured from enemy's center and straight forward):

StartAttackingRange = 0.72 meters

AttackRange = 1.20 meters



The herder's AI is designed to flank the player. This is achieved through two prioritization stages:

Stage One: prioritize the location of a point behind player's back.

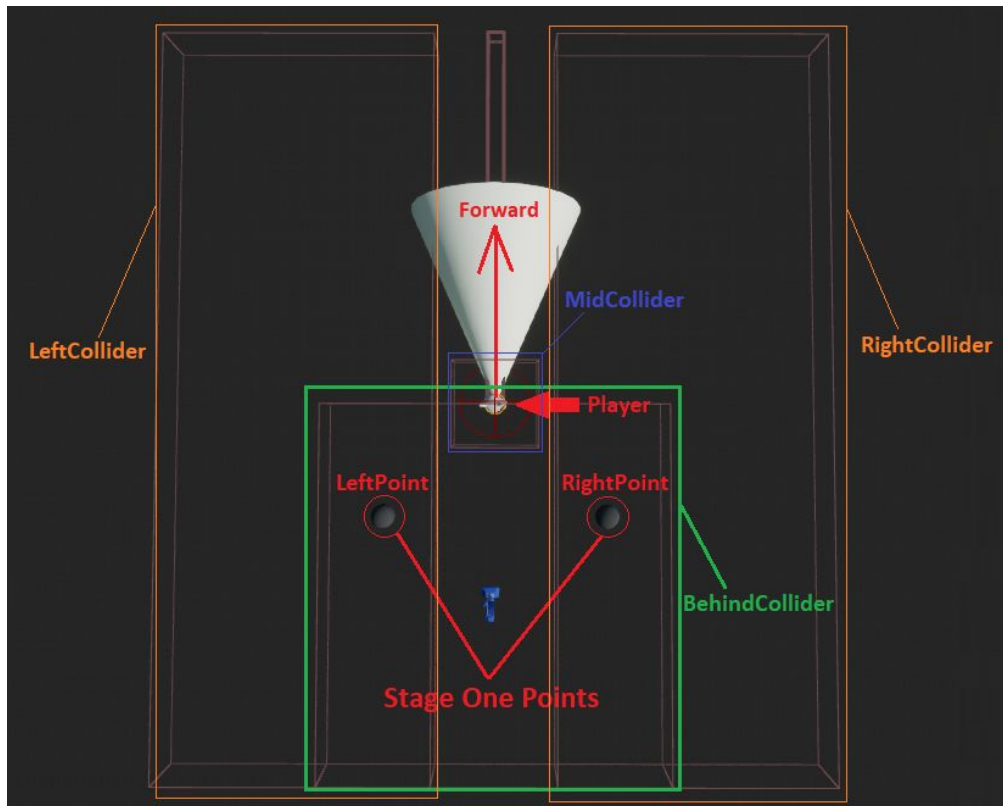
Stage Two: prioritize player's location.

When a location is prioritized, the herder will try to navigate through the game world, with its goal being the prioritized location. These two stages can only be completed with the help of these components: *LeftCollider*, *RightCollider*, *MidCollider*, *BehindCollider*, *LeftPoint* and *RightPoint* (look at picture below for placement reference). All components are children of player.

To describe this further, here is a more in depth look at these two stages:

Stage One: At this stage, the herder will prioritize the closest "Stage One Point". Both *LeftCollider* and *RightCollider* is used to determine the prioritization. If herder is located within the *LeftCollider*, the *LeftPoint* will be prioritized. If herder is located within the *RightCollider*, the *RightPoint* will be prioritized. Stage one will be considered complete whenever the herder is located within a 1.50 meter radius of either *LeftPoint* or *RightPoint*.

Stage Two: this is the last stage. Stage two will be initiated as soon as stage one is completed. The main goal for this stage is to move to the player. The herder will now be located inside the *BehindCollider* (either at *LeftPoint*'s location or *RightPoint*'s location). As long as the herder is located within the *BehindCollider*, it will prioritize the player. If the herder happens to exit the *BehindCollider*, then return to stage one.



If the herder happens to ever collide with *MidCollider*, regardless of what stage is being executed, it will prioritize the player over everything else, until it exits the *MidCollider*.

For the herder to behave as intended in stage one, *LeftPoint* and *RightPoint* is required to always be located above walkable area. This is guaranteed by having the distance from the player to the point being a variable, and not a constant. This will open up for a dynamic distance, instead of a static one. Both points are assigned their own distance variables. To adjust these values, each point will cast its own raycast aimed directly downwards. These raycasts are used to scan the surface below each point. The information gathered from these raycasts will determine whether the distance is considered *valid* or *invalid*;

Valid = if raycast detects walkable area (for example a platform).

Invalid = if raycast detects unwalkable area (for example water)

Both of these distance variables will be assigned a max value of 7.00 meters, which will be their preferred distance. This max value will always be set as the distance, as long as there has not been detected any *invalid* distances during the calculation of the current frame. If a distance is considered *invalid*, the distance will be reduced until distance is considered *valid*. The reduction consists of decrements of 0.40 meters.

4.10. Gameplay variables

Player

PowerUpTime

Decides how long power-ups will last in seconds

PowerUpTimeLeft

A timer that checks how much time is left of a power-up

ReloadSpeedDefault

A scalar that decides how fast you charge up your laser without a powerup

ReloadSpeedUpgraded

A scalar that decides how fast you charge up your laser with a powerup

ReloadSpeedCurrent

Value that holds the current reload speed of player's flashlight

PowerUpLightScale

A scalar that decides how much bigger your flashlight's cone becomes with a powerup

PowerUpLaserScale

A scalar that decides how much bigger player's laser hitbox becomes with powerup

PlayerHealth

Decides the amount of health points of the player

DamageTaken

How much health the player loses when damage is taken

ShootingTime

How long the laser attack should be active after fired in seconds.

CurrentPowerUp

Checks what your current powerup, used for display on the UI

Enemy

TimeUntilStunned

How long an enemy needs to stay in the flashlight cone for it to become stunned

TimeStunned

How long an enemy should be stunned in seconds

TimeInFlashlight

How long this enemy has been inside the flashlight cone

HerderTopSpeed

How fast a Herder is able to move

AnklebiterTopSpeed

How fast an Anklebiter is able to move

MovementSpeedReduction

How much speed the enemy should lose by being inside the flashlight cone

Movement

Movement speeds:

Player = 350.0

The Anklebiters = 250.0

The Herders = 350.0

These are values gotten from the Unreal Engine character movement component and they are totally “linear”. This means that e.g 200 is double of 100. Having the Anklebiters be slower than the player allows the player to run away from them and easily dispose of them. The Herders have the same speed as the player making them more of a threat.

Player jump:

The player has a jump height that allows her to jump past certain encounters. This is balanced by checkpoints only being obtained by clearing monster encounters.

5. Art style



Hero color palette

Design of Hero

The hero of the story has a lively colour palette, with a friendly look. The green is to symbolise life and growth.

Weapon design

It is an ancient weapon carved out of a tree. A powerful crystal is made to fit the weapon and will exterminate any impurity from the unknown. The weapon is designed to fit in the universe of the game with its nature-like appearance.





Enemy colour palette.

Design of Enemies

The enemies have a more cold dead appearance to show that they are not friendly. The use of the colour red and black makes it easier to spot who the enemy is. They consist of red crystals which symbolizes their corruption.

6. Environment

General Idea

The design of the environment was a decision made after the design of the character was finalized. We went through our older designs to see if anything could work with the current theme of the game.

What we needed to put into consideration was how it would work with the gameplay. Looking back we found a bathtub filled with toys, which the main character used as platforms to get from A -> B.

We wanted to employ this method of level design due to two reasons.

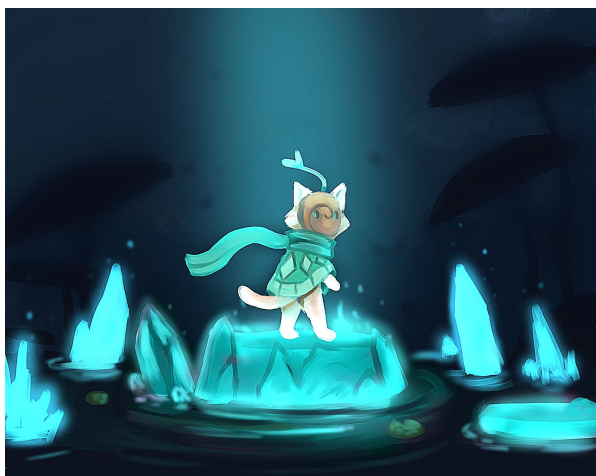
1. The player would have little to no difficulty finding their way through, when most of the map was dangerous water.
2. With our limited time, simplifying the number of assets and their complexity would help us amplify their quality.

From here we could take to form the design of said platforms and backdrops.

When doing the design it was clear from the beginning that we wanted a desolated, Mystical world. The Beacon of light, the ruined city and the old forts are all made to emphasize the age of this world. This is the holy land of some ancient beings, and one shouldn't tread carelessly.

To show the invasion of the monsters of the abyss, we added a new addition to the old landscape, namely huge, red, glowing crystals.

This effect is used again in the second level, but with mushrooms. These glowing mushrooms are there to add a alienation the player might feel, these lands are not exactly your home. The glowing effect was added to balance out the red glow of the crystals, to show that even nature is fighting against these intruders.



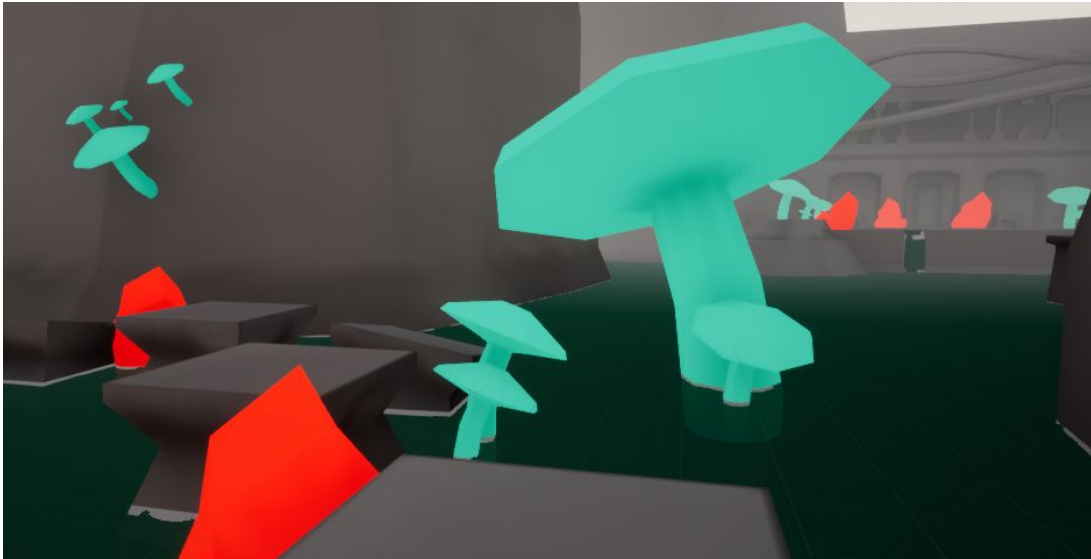
Environment of Level 1

The First level features a myriad of buildings, towering over the player character, but sadly sunken into the ocean below. To reflect this, we decided to give everything a slight mossy tint, and show clear signs of decay. The building style of this ancient civilization would be out of a beige/gray stone, showing the withering of time. Throughout the level glowing red crystals will light up the level with a red hue



Environment of Level 2

The second level builds upon the architecture of level 1, however with the denser jungle surrounding the level, the greenish tint is turned even darker. The mushrooms in this level will add a hue of light blue, whilst the crystals, like in level 1, a red hue.



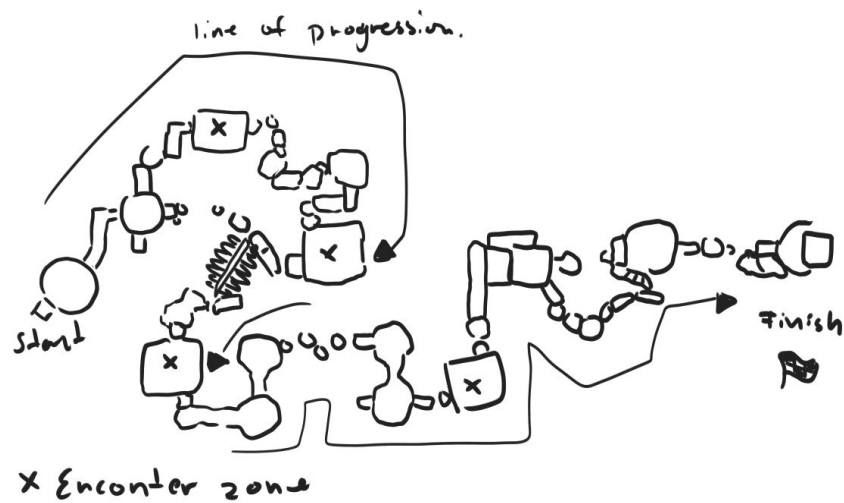
7. Level design

Level 1

There will be a total of 2 levels. They should have a linear layout with a clear path of progression.

The first level is on a set of water platforms. It is a place where the hero meets less enemies and has more of a focus on learning movement and fighting.

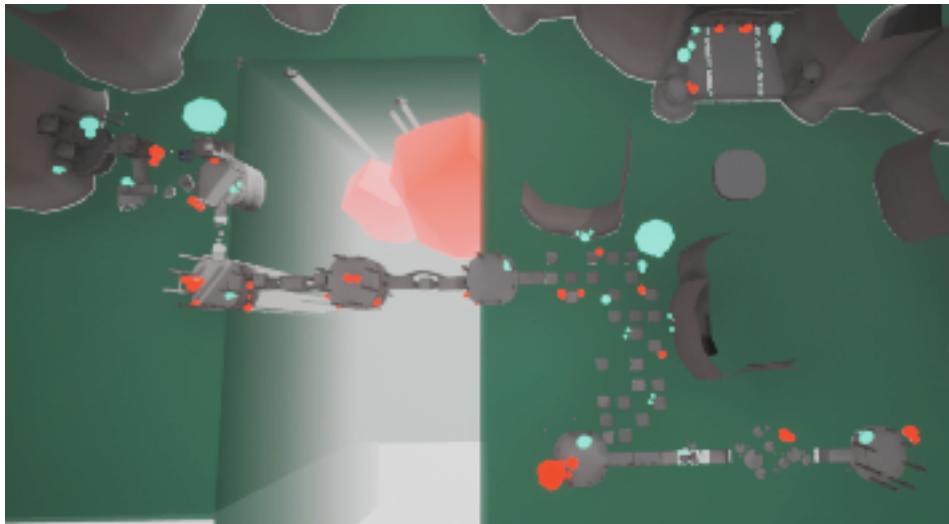
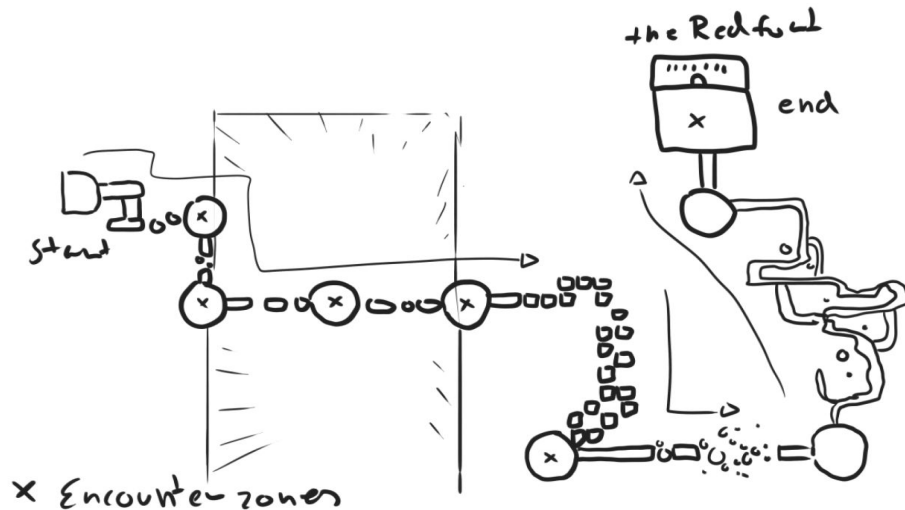
Level 1: water themed



Level 2

The second level is a forest. It begins after the player reaches the end of the first level. This forest is filled with mushrooms that glow. This place has more enemies and ups the difficulty of battle greatly. The game ends when the player arrives at the Red Fort and defeats its guards.

Level 2: forest/ swampy theme



8. Music and sounds

With limited time, comes limited possibilities. Looking back in time to an area of video game music which thrived in simplicity, due to the lack in processing power. However walking into this area expecting something easy is fool errand.

Trying to limit oneself to only a couple of instruments to make the music feel free and open, much to emphasise the areas in-which the music would be played.

The first level, set by the ocean, would feature a set of slow synth build-ups, so one could feel the eeriness of the land you trespass.

The second level would feature a set of chimes and bells, to give of this magical fea feeling. This is their realm and you might not be as welcome as you'd think.

Both Levels have a underlying rhythmic track, both comprised of maracas and drums with brushes.

Inspirations:

https://www.youtube.com/watch?v=4Qp0_NE8M1c (Donkey Kong Country 2)

<https://www.youtube.com/watch?v=QbBvh192RRs> (Hyper Light Drifter)

<https://www.youtube.com/watch?v=amlR7AKL9jl> (Chrono Trigger)

9. Marketing and Funding

Demographic

The game is aimed at people who grew up in the "Nintendo 64" era. The simple 3D graphics with very memorable character design really gives nostalgia for old N64 classics like "Super Mario 64" and "Legend of Zelda". It appeals to people who wants a simple game to just relax.

Platforms and Monetizations

"Steam" is a prime contender for monetization of the game although with the increasing popularity of the "Epic games store" it would be great to have it released there as well since we are using their engine to make it. Taking the scope and quality of the game into mind we would not sell it for more than a dollar.

10. Glossary

“Weakening” = Enemy is in the process of being “**stunned**”. “**Weakening Scale**” will be set to a value between 0 and 180.

“Stunned” = Enemy is completely weakened and susceptible to the “**Laser**”. “**Weakening Scale**” = 180. Enemy will stay “**Stunned**” until its “**Weakening Scale**” returns back to 0.

“Weakening scale” = Determines the progress of how weakened an enemy is. Scale floats between 0 and 180.

“Rallied” = Enemy who is not weakened at all. “**Weakening Scale**” = 0.

“Laser” = A powerful beam that kills enemies who are “**stunned**”. It also stuns enemies who are not currently “**stunned**”.