***THIS IS A WORKING DOCUMENT***

# General Information

Name of Game: **The Day After**

Intended Game Systems: PC

Target Age of Players: 17+

Intended ESRB Rating: T

The Day Dun
After Dun
DUMM

# Game Flow/Story

The extent of this world is in the city of ***Syptort*** located on the coast. It is an independent place with a small government but it is mostly just people living their lives. Theoretically, other places exist that they trade with. This world is a steampunk-style place with ~19th-century tech and some sci-fi aspects. The city has different areas to explore with different specializations/themes.

The player starts the game around midday in their mech shop in the outer suburbs which also acts as their home. This area is separated from the rest of the city by a river and drawbridge which starts up to keep the player in this area. The player is a mech builder who takes commissions from other people. One of these commissions is what requires the player to leave the house to shop for the required materials. The player does not have any other significant lore.

When stepping out of their home the player is met with the “house” area, a currently peaceful place that’s more suburban with fields in the distance. They walk through the town to a shop area with NPCs who have set up stalls to sell their goods. This acts as a tutorial for the shop system. When a list of items has been bought, the player is directed to go back to their shop. At this point, it is starting to get late in the day and the sun is setting. When the player gets back, they are notified of one forgotten part, but the shops are closed at this point. The player is then directed to go to the junkyard to search for the part instead. This begins the tutorial for the mech, as the player gets in the mech to start going to the junkyard.

By the time the player gets to the junkyard, it is dark. The player sneaks into the junkyard through a side entrance and avoids searchlights as they progress through the area looking for this part. At the end of this path, they stumble upon a broken-down mech that has a glowing gem in it. When the player picks it up and takes the item, the junk guard catches it. This starts the first mini-boss fight which gives a tutorial on fighting mechanics. Right before the player defeats the junkyard guard, a cutscene plays in which the guard calls for reinforcements, and the player can see the construction mechs from ***Syptort’s Industrial District*** coming to assist.

Suddenly, the player sees everything slowly stop, and then begins reversing. The mechs walk backward, the junkyard guard moves back to the tower, and the stars move and the sun rises from the west. However, the gem the player picks up makes them immune to these effects. The player then returns to their home to sleep. The next day, the player tries to ask NPCs in the house area about the reverse effect they saw, trying to figure out what happened. Most of the NPCs are clueless and greet with the same dialogue as the previous day, having no memory of their interactions. At this point, the rest of the map opens up for the player to investigate the time loop that continues each day that is spent, as well as to look for a way to stop it.

After the first day, the player is then directed to go to the city center. They reach the edge of the home village and cross a bridge over the 90° river. The city center is full of NPCs who share facts about their lives, and offer quests to the player in return for a small reward. There are additional shops that offer higher-tiered goods, but also for a higher price. At the edge of the city lies ***Syptort's Lab***, an old and abandoned lab once used by a team of brilliant scientists who experimented with a strange material that could affect the fourth dimension: time. These scientists have supposedly long since disappeared. The lab is locked with an ancient security system, and the key to disarm it has been split into pieces and lost to time.

The city also acts as a transition to the two other areas of the game: ***Syptort's Industrial District*** and the ***Docks***. The player is allowed to choose which area they explore next. The entrances to both areas are heavily guarded, and the player must find alternate routes into the areas.

If the player chooses to go to the main entrance of the dock area, they find that it is guarded by a security system designed by someone known as the Portmaster. The Portmaster can be seen on a TV watching everyone who tries to enter and does not let the player in. If the player does not leave, the portmaster summons mechs and guns to scare them away. The player is then prompted to search for another entrance to the area. They eventually find an alleyway they can use to sneak in.

In the Docks, loader robots named Yeeters are moving crates from loading areas to boats that will ship the cargo elsewhere. The yeeters are controlled by the Portmaster Lennard in the control room, so the yeeters are hostile to the player on site. When the yeeters see the player, they will throw their cargo at the player, and attempt to melee the player. The player will have to fight their way through the dock area to make it to the control room, where they can fight Lennard.

In this fight,

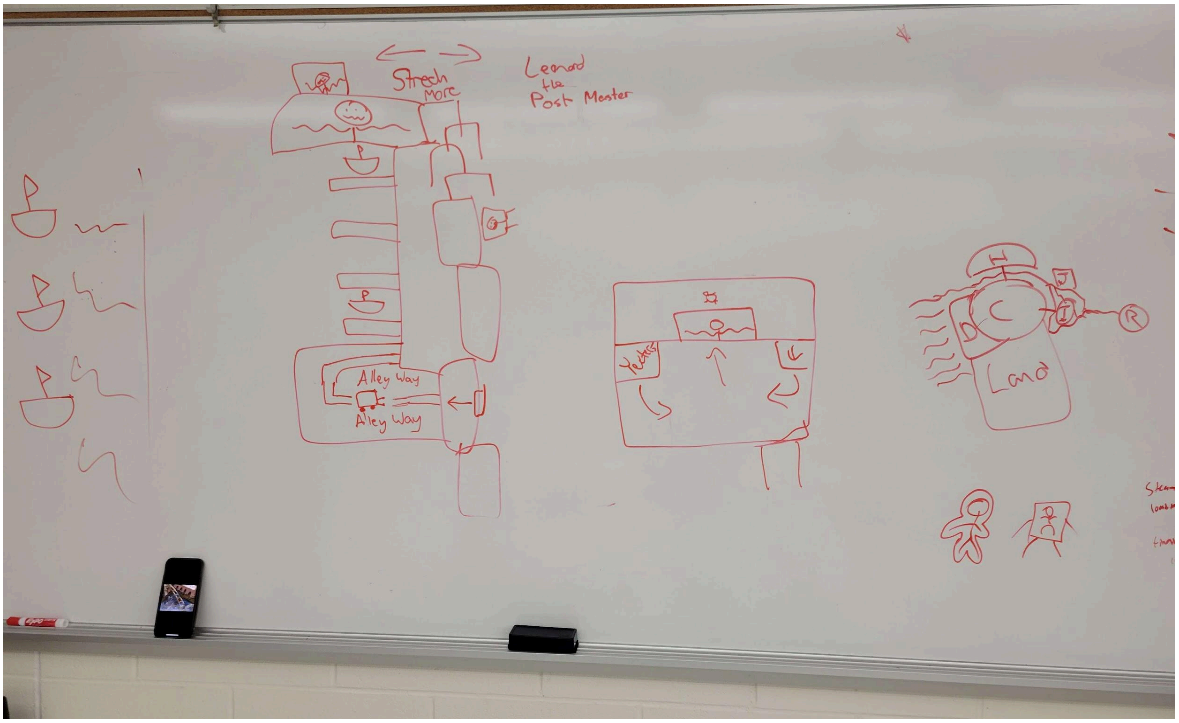
Once both the Portmaster and MA boss have been defeated, the player has both parts of the key to be able to access the lab. The player is directed to go back to the city to the lab entrance and insert the key. This area looks very sterile, with unfinished experiments on the tables, papers strewn about, and dust everywhere implying the lab has been long abandoned. There is no one inside, but in the center is a large trapdoor which can be opened to find an old mine elevator which leads underground. When the player enters the elevator it starts to go down, but at some point begins to shake and stop, with the floor of the elevator eventually giving out and leaving the player to free fall to the bottom. At the bottom the environment becomes a narrow cave passageway that leads to a larger open cave area with time gem deposits on the walls. The "time lord" who is controlling the time loop can be found in this cave, waiting for the player in his mech, which has a large time gem in it. When the player enters the arena, a cutscene plays where the player character can be seen attempting to punch Rekab, but Rekab uses a device on his arm to invoke the time powers he has to revert time a couple seconds to just before the punch, and block against the player. Periodically, Rekab will use time powers to defend against the player, but this only makes him weaker. In becoming weaker, he gets more mentally unstable and arrogant, launching harder hitting attacks and attacking more frequently, but is less able to react to the player. After a certain amount of rewinds, the player can defeat him which starts the final cutscene. The player character can be seen knocking Rekab to the ground, and smashing the time manipulation device. The player character then rips off the helmet of the mech, revealing his identity. Credits play.

# Gameplay

Are there multiple chapters/acts to the game? What is included in those acts? Is the game broken up into levels or rounds? Are there any cool scenarios like driving while shooting or jumping from a building to run away from an enemy? If so call attention to those. Include the big set pieces, such as the big cities/villages. Outline any minigames if we have them and give a short description of them. Now go into details about the platform-specific features. What game features capitalize on the hardware we have? Do we utilize a memory card or a hard drive? Or is it downloadable? Does it use a camera or a motion controller? Is it multiplayer, with a split screen? Cover those details as it will be important for readers to understand what tech requirements will be needed to produce your game.

There are 6 different sections in the game. This includes the Home Base, the Junkyard, the City, the Docks, the Industrial District, and the Lab. In those different areas we have various bosses, enemies, missions, lore, etc.

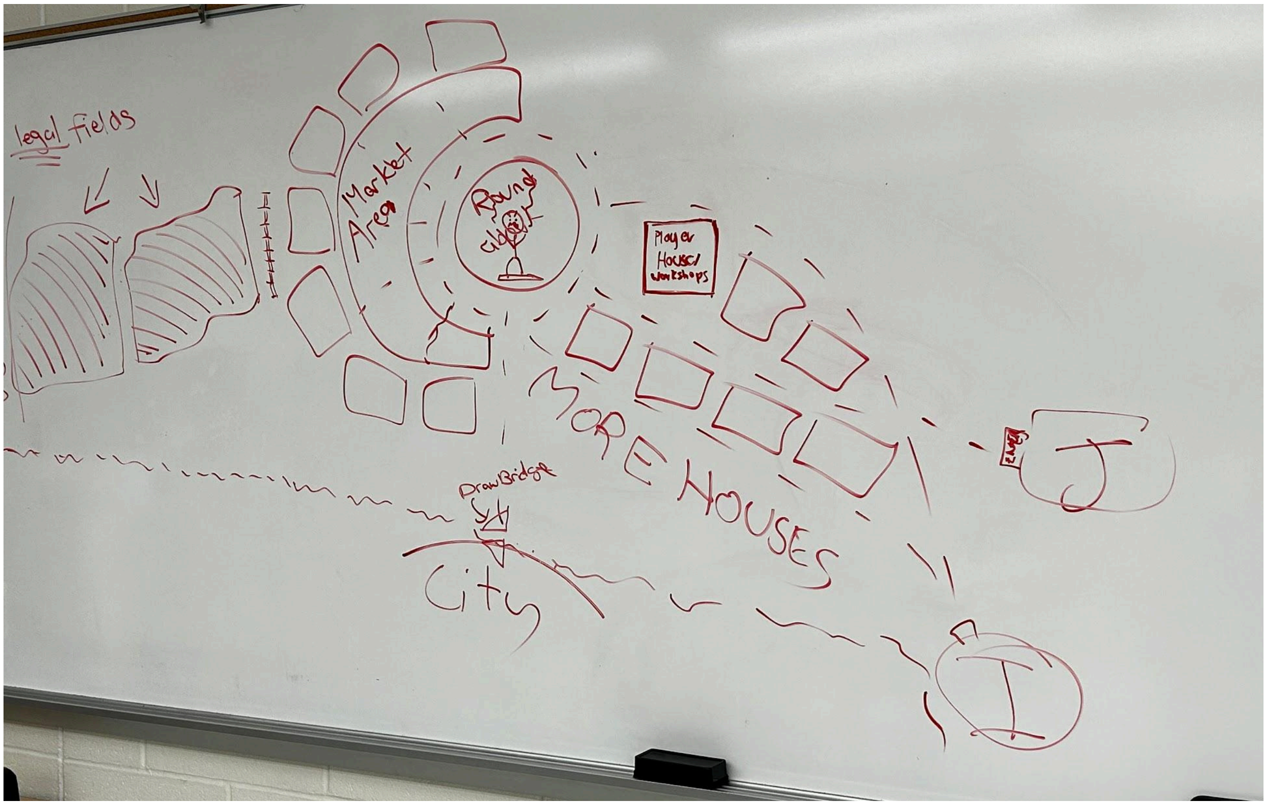
# Game World

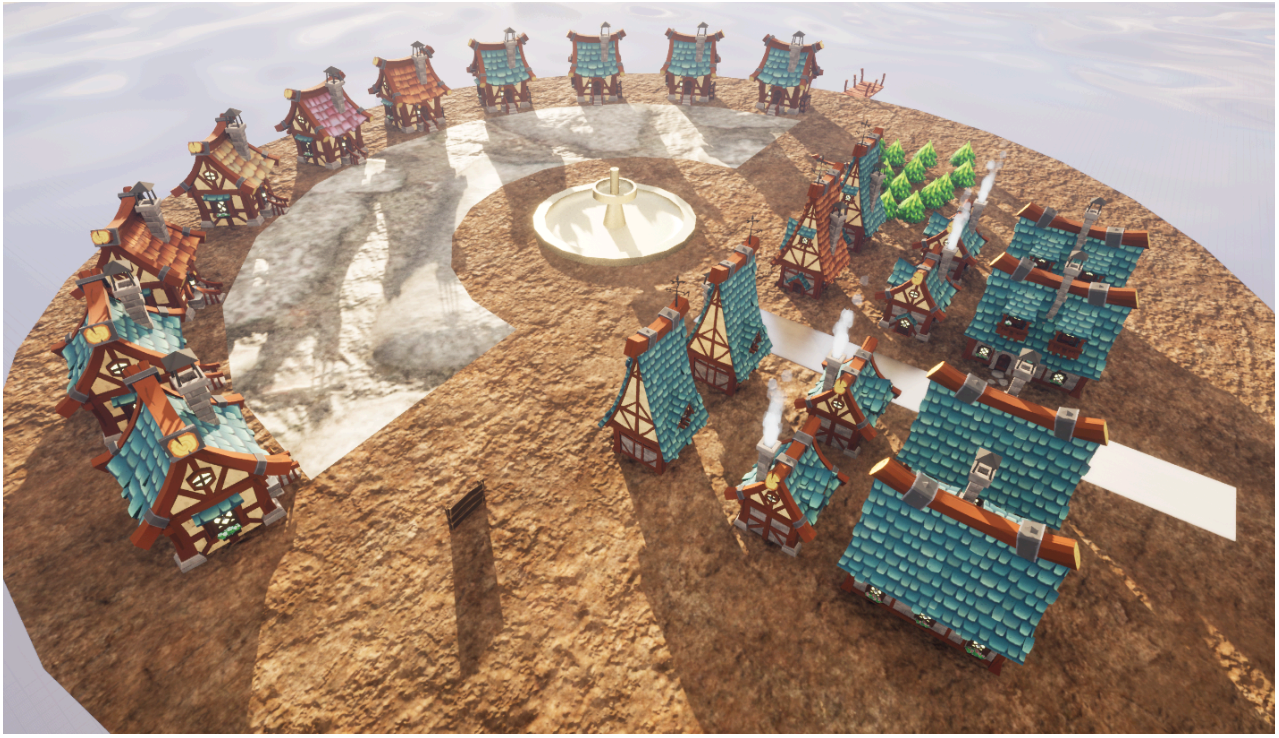

Strech
More
Leonard
the
Post Master
Alley Way
Alley Way
Yeetus
H
J
C
D
I
R
Land

## Housing Area

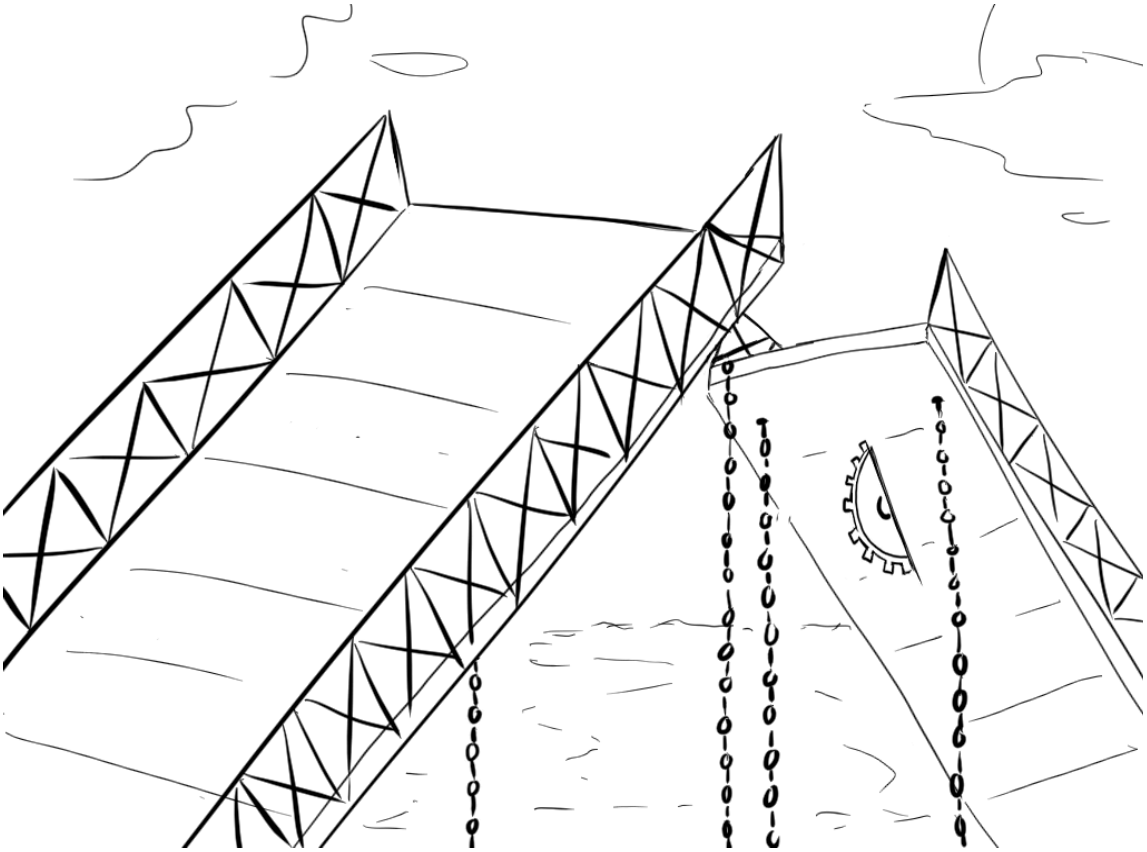
The housing area is where the player, along with the majority of the city's citizens, lives and works. The player's workshop and house are both located here. The houses  look very similar to one another, with brick and wood facing and mainly steam powered technology inside.

In the player's workshop, the player makes and repairs mechs for the citizens, which requires scrap parts. These parts can most often be found in a separate area, the Junkyard. There is a drawbridge, also steam powered, which lowers from the housing area to the city to allow for commute.



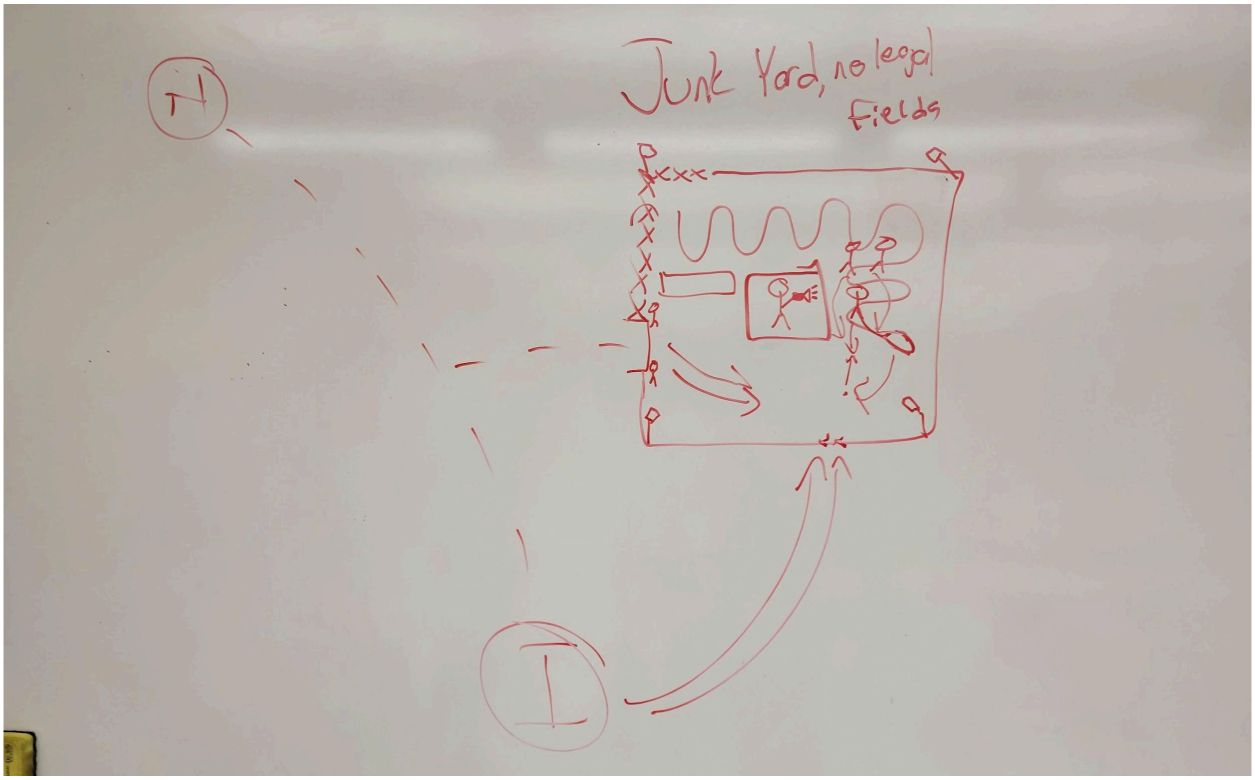


CLOCK
fountain
market



## Junkyard

After beginning the first key quest directing you to go retrieve an unavailable part necessary for your mech, the player ends up on the track to the dark and shrouded Junkyard. The Junkyard serves as both a trash heap for scrap and unused parts as well as as a secure containment center for a vital technology hidden from the public - a time crystal. Given the defensive nature of the map, The Junkard is surrounded by a tall wall, and contains a lookout tower at the center with two manned spotlights crowning it. The map is designed at night time to give the experience of an illegal break-in; the player inadvertently enters a high stakes stealth mission. The thick fog, narrow visibility, metal fences, and patrolling guards all further add to the feeling of a dangerous, restricted area (the amount of obstacles put in the way here should raise alarms in the player. Why is there so much security?).After discovering a secret entrance on the side of the map, the player must navigate through a maze of scrap and metal to find the part - all while avoiding the positioned searchlights. The map is relatively small and easy to navigate and, given its order in the game, is supposed to serve as a tutorial map that teaches the player the basic mechanics before the player begins mech-action gameplay. However, If the searchlights do catch you, alarms will be alerted and red lights will activate. All nearby guards are directed to your position, to which the player is supposed to lose a fight against and restart the level - the goal is to take down the guards individually without alerting the alarms. When the player reaches the boss arena, a fenced off hexagon shape that contains a broken down mech in the center, the player notices a glowing gem in the body of the mech. Curiosity prompts the player to take it, which alerts the guard in the lookout tower to jump down to engage the player. After the bossfight, reinforcements are called and break through the wall. Fortunately, at this moment the time loop resets and the player is put back in his house while retaining his memory and the crystal sphere part.

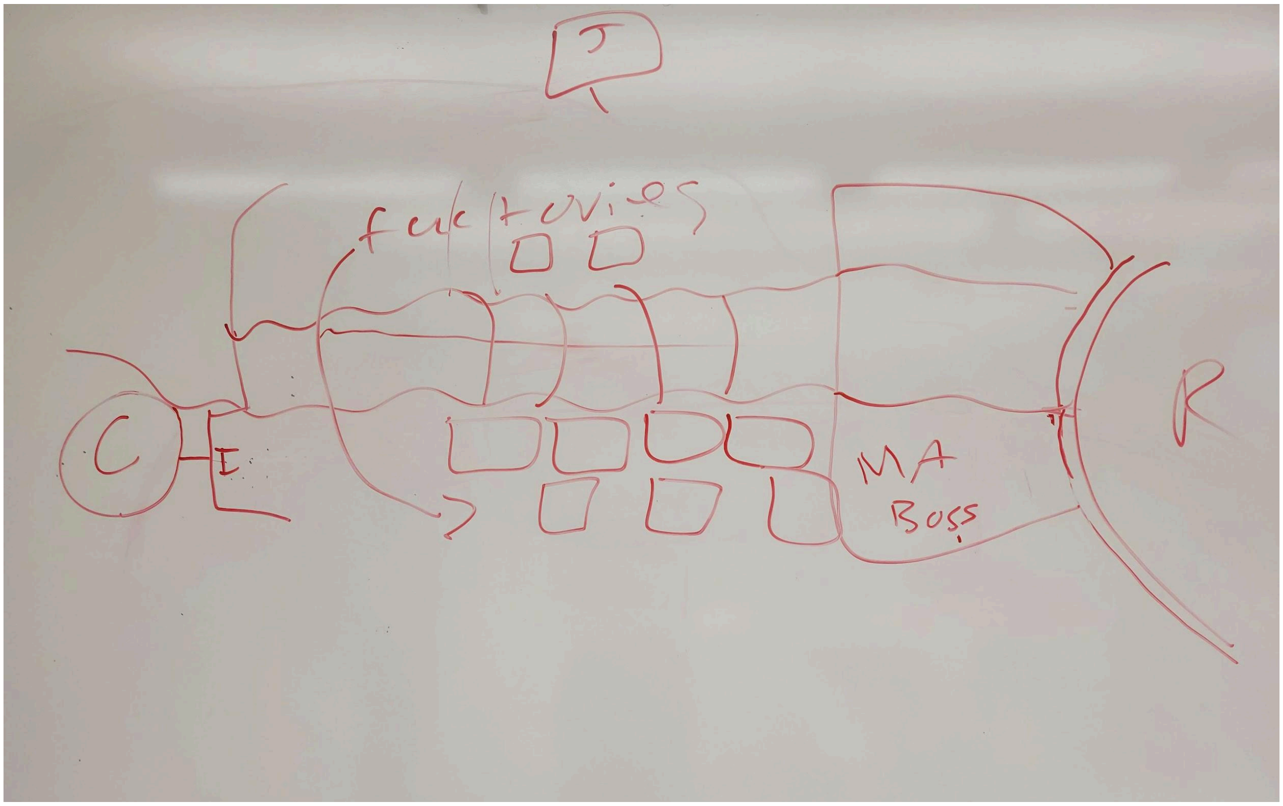

H
Junk Yard, no legal Fields
I

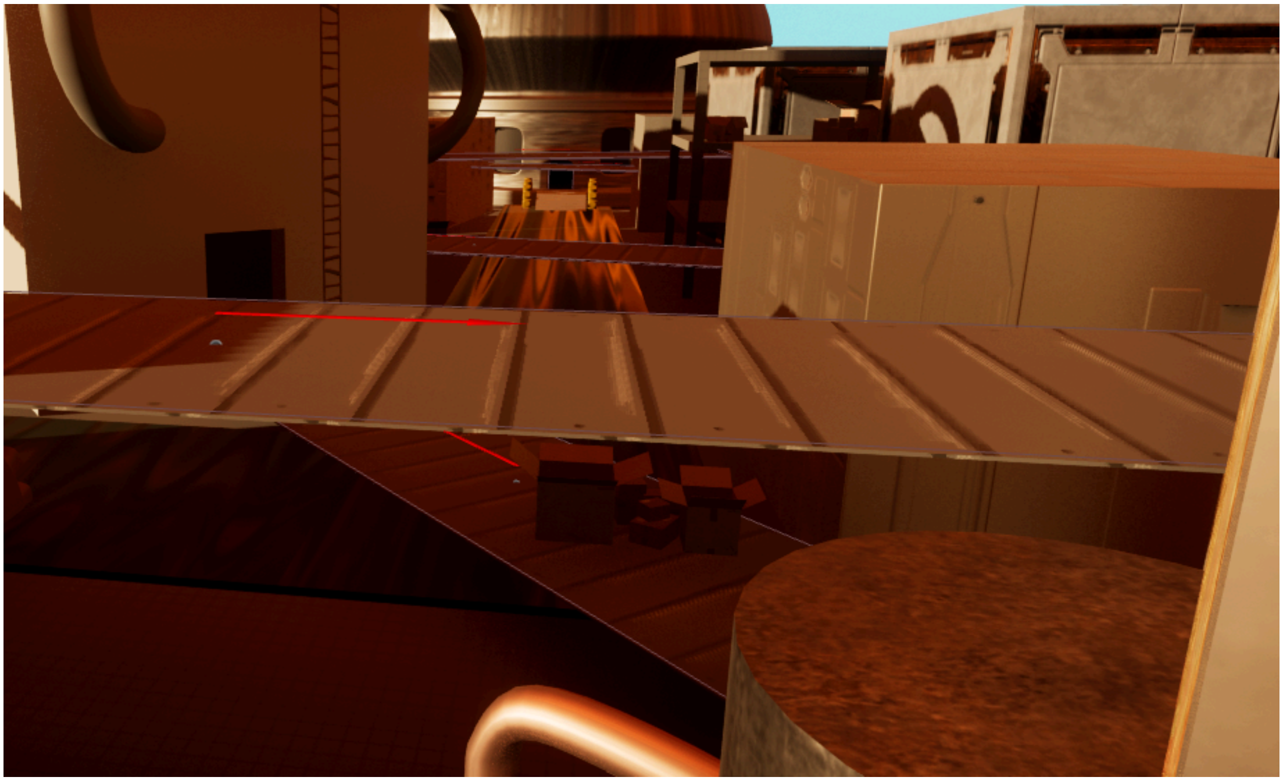
## Industry

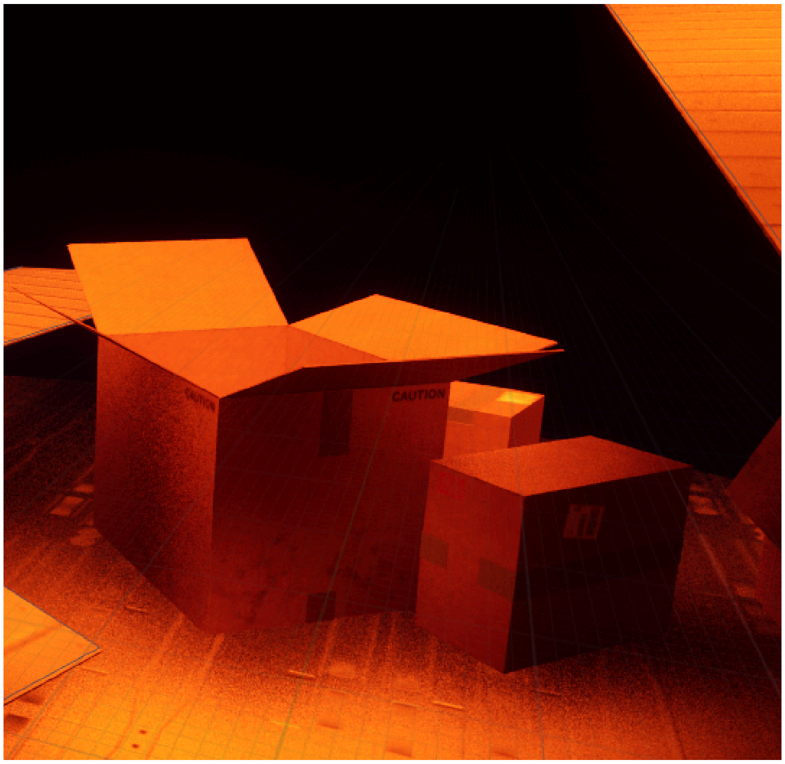
The industrial area of the city is where most mech parts are manufactured, and is filled with factories. They occupy the sides of the river, using steam and water generated power in order to perform all of their functions. Conveyor belts travel between the buildings, allowing for transferral of materials and goods, and for easy transport as the player works their way through the level. The final building is an oversized factory seated atop the river, taking in most of the power of the district and housing a boss fight.

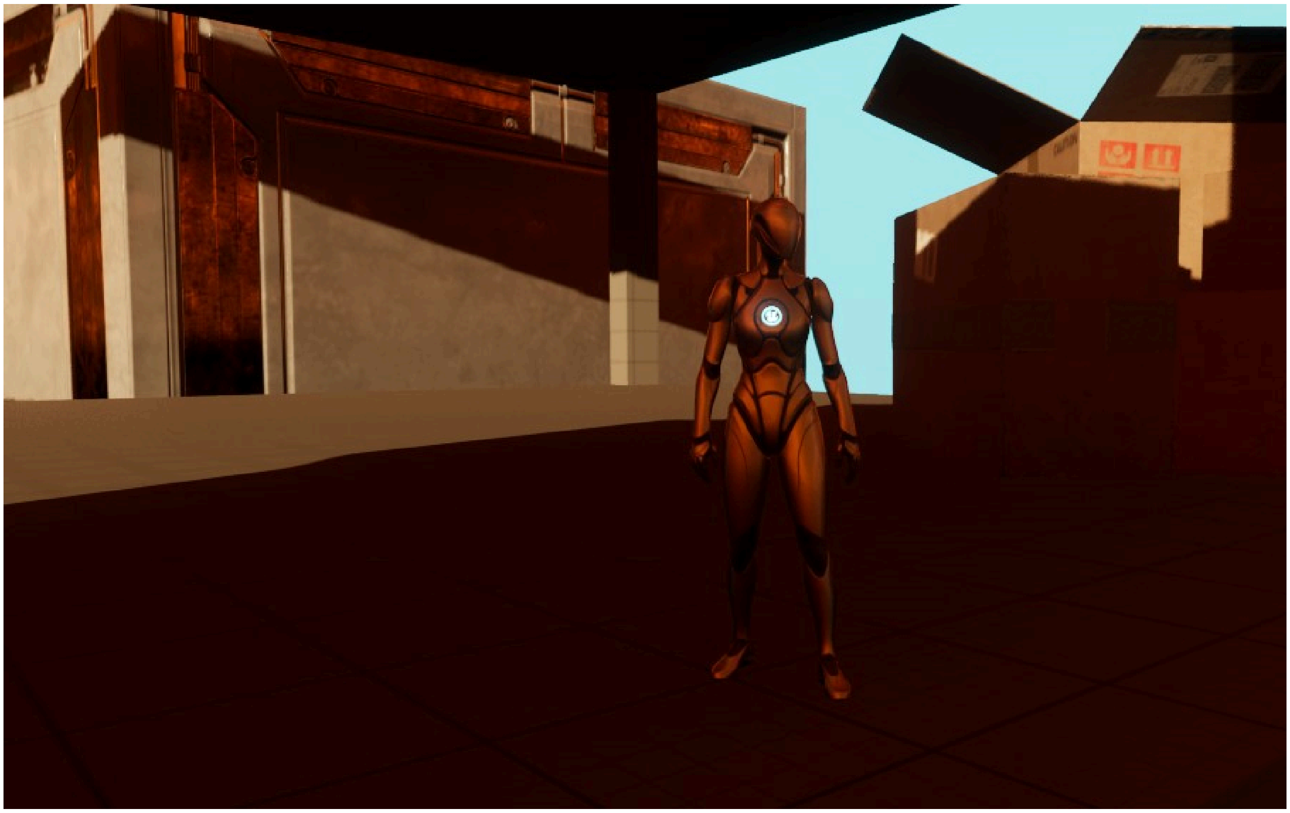
All around the district are workers from the industrial area in their labor mechs, who will attack the player on sight, since the industrial area is meant to be restricted. The map is large despite being uncomplicated, and so while easy for them to navigate, the presence of the river in the center means that the player must scale pipes and ladders, hop on conveyor belts, and leap across the tops of smaller factories in order to reach this final destination, which then transports them into the circular interior where they will fight the boss.

The arena is dark except for red and orange lighting which pulses as the player fights. Cardboard boxes, presumably those used to transport goods across the industry and to the city's citizens, are scattered throughout the area.


J
factories
C
E
MA
Boss
R

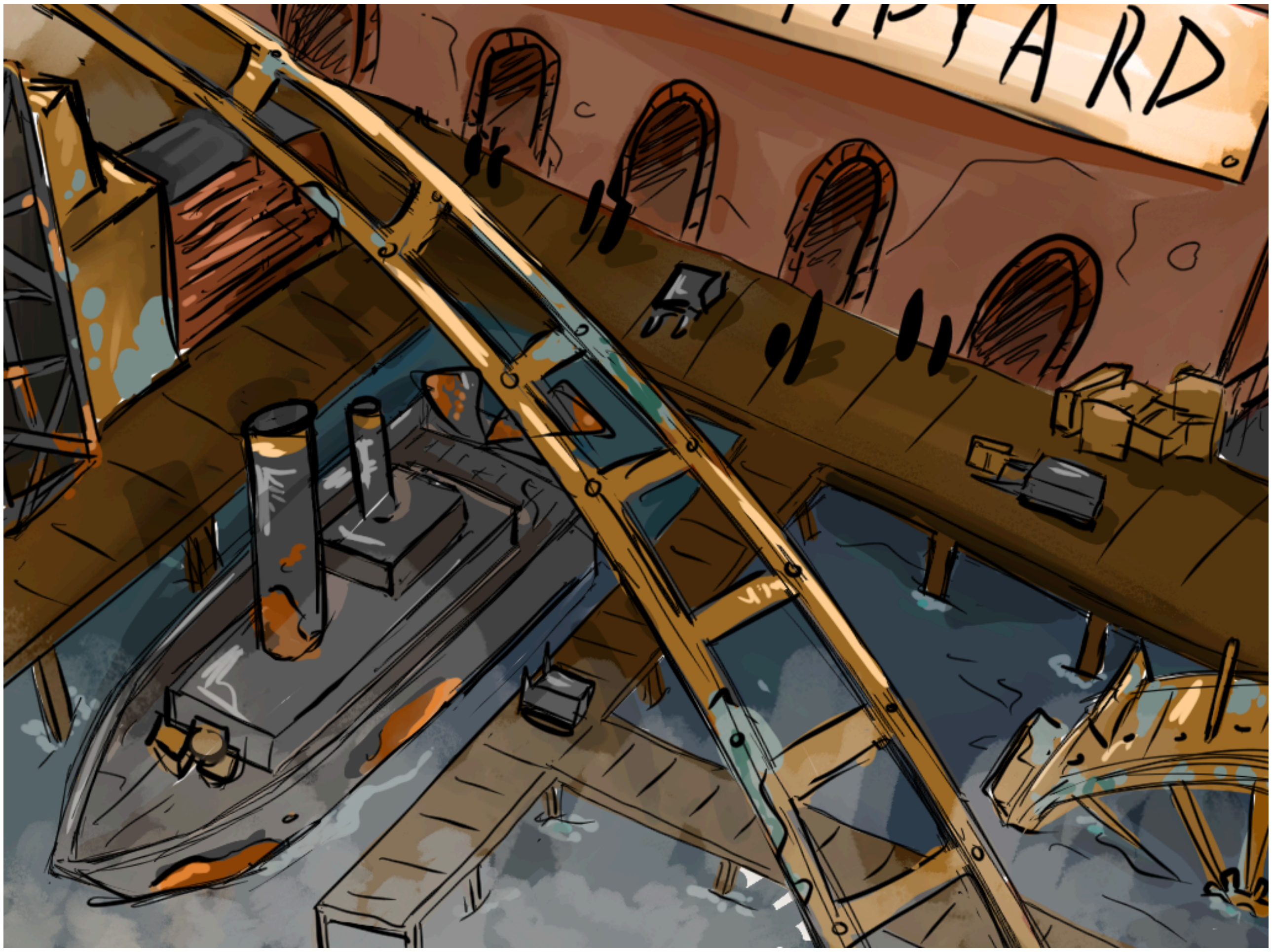



CAUTION
CAUTION



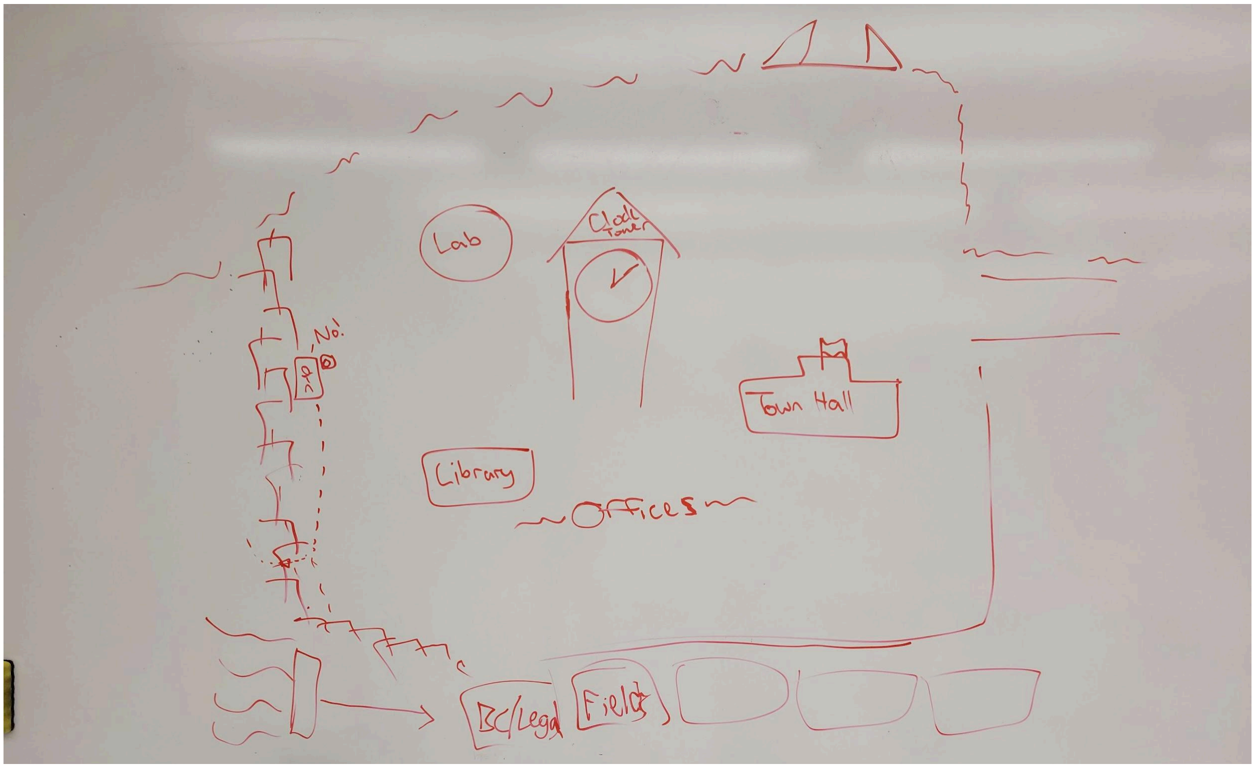
## Docks

esiufhj



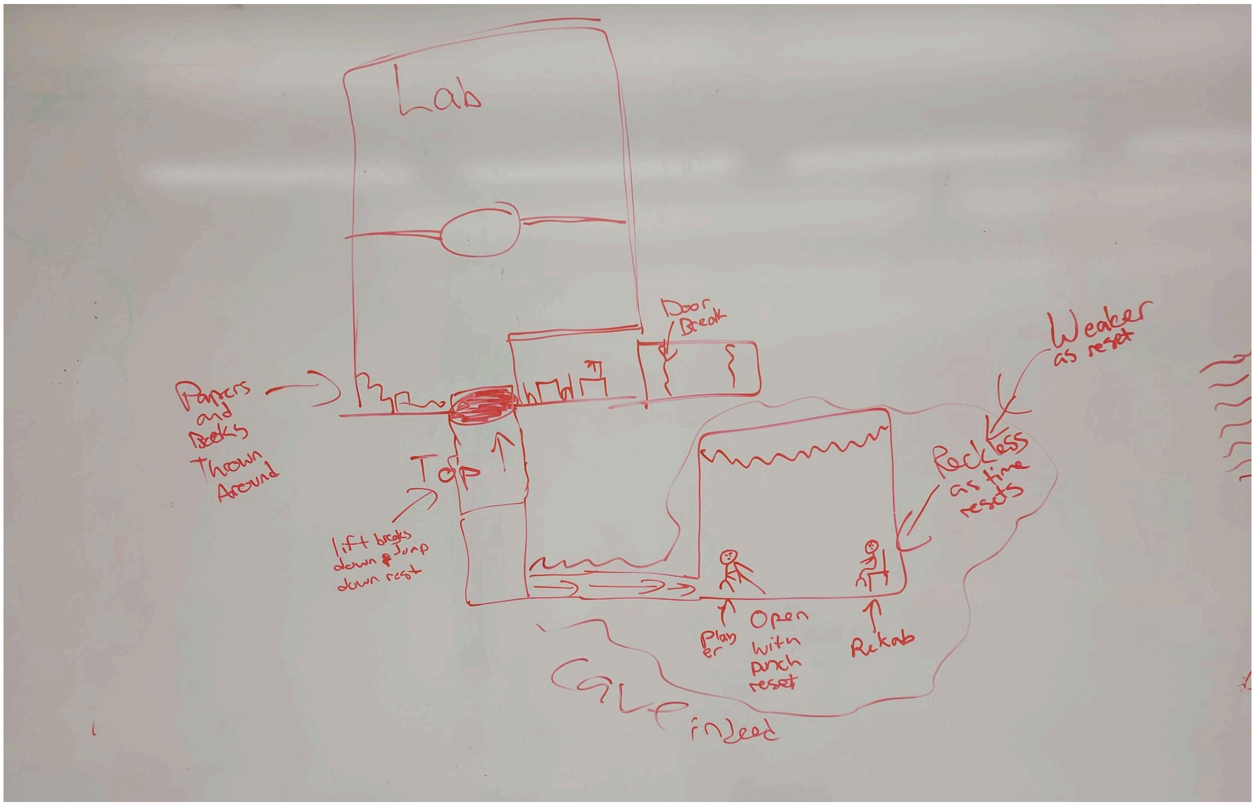
## City Center

srktgyhjfd

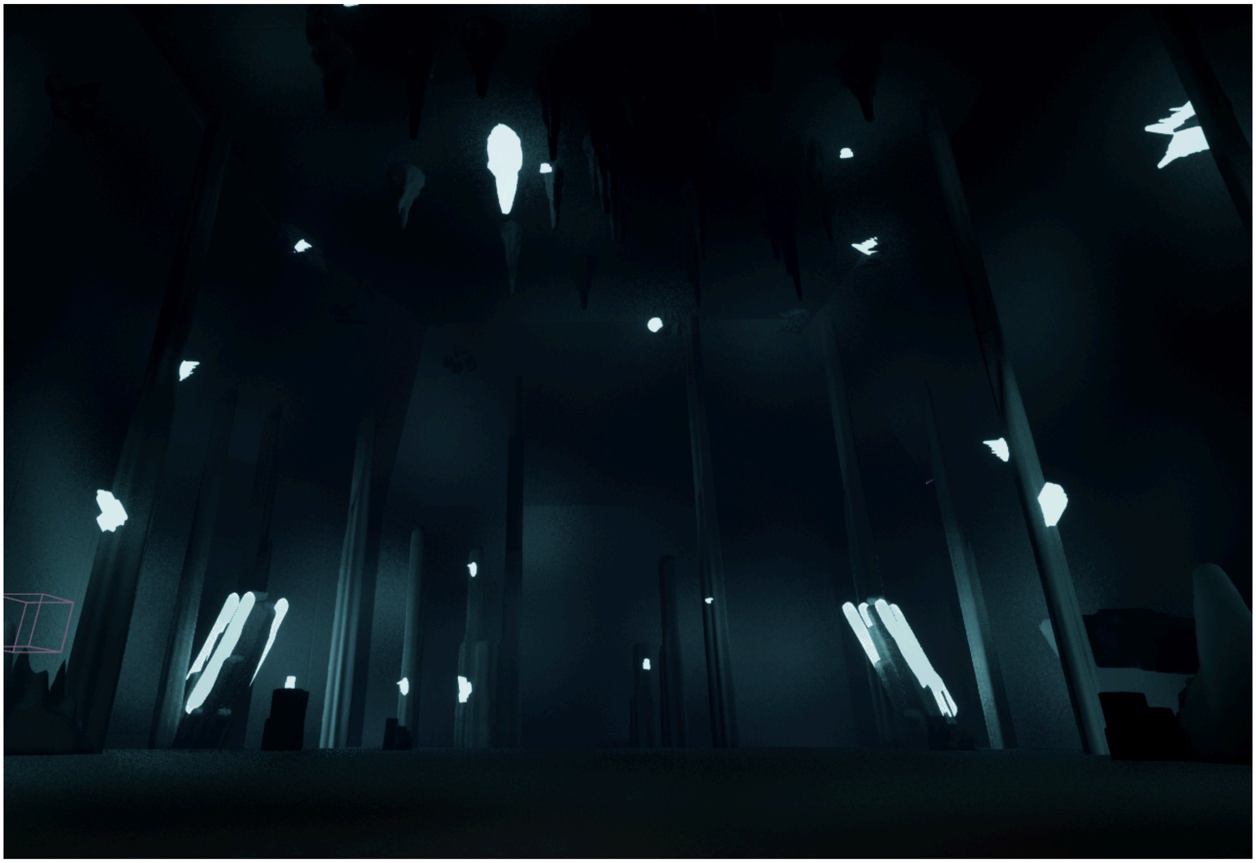


## Lab

hjghgrj


Lab
Door Break
Weaker as reset
Papers and Books Thrown Around
Top
lift breaks down & Jump down rest
Reckless as time resets
Player
Open with punch reset
Rekab
Cave indeed

# Secret Cavern

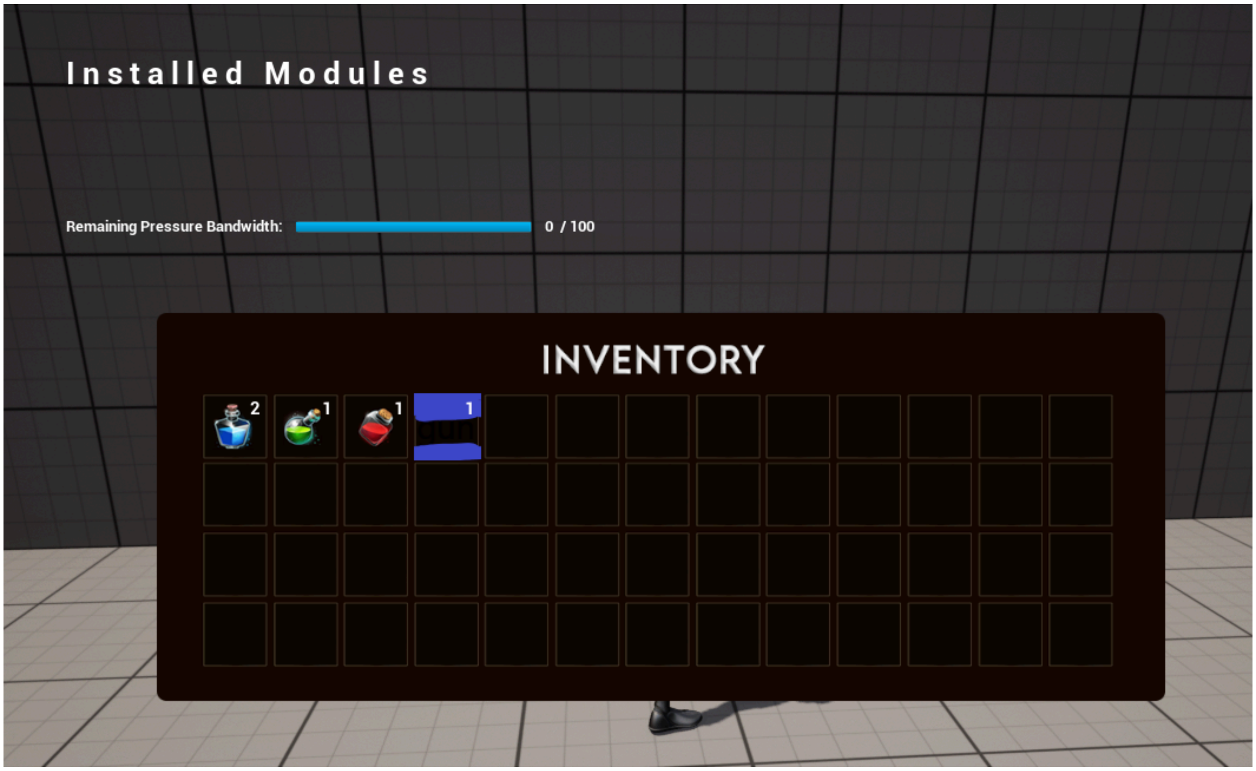


# Game Experience

To make a game feel like a complete experience, you need to account for the feel of the starting screens, your cinematics, your music, your sound design, and the camera. In other words, the whole of the game. What is the “whole of the game”? What is the theme? How do we include the theme in the game? Music? Cutscenes? Fonts? Visuals? Characters? What does the player first see when they open the game? What emotions/moods is the game trying to create? How are sound and music used to convey the game’s feel? How does the player navigate the shell/menus in the game? Show a flow chart about how the menu flows and show each part.

# UI Elements

## Inventory



***<THIS IS PLACEHOLDER>***

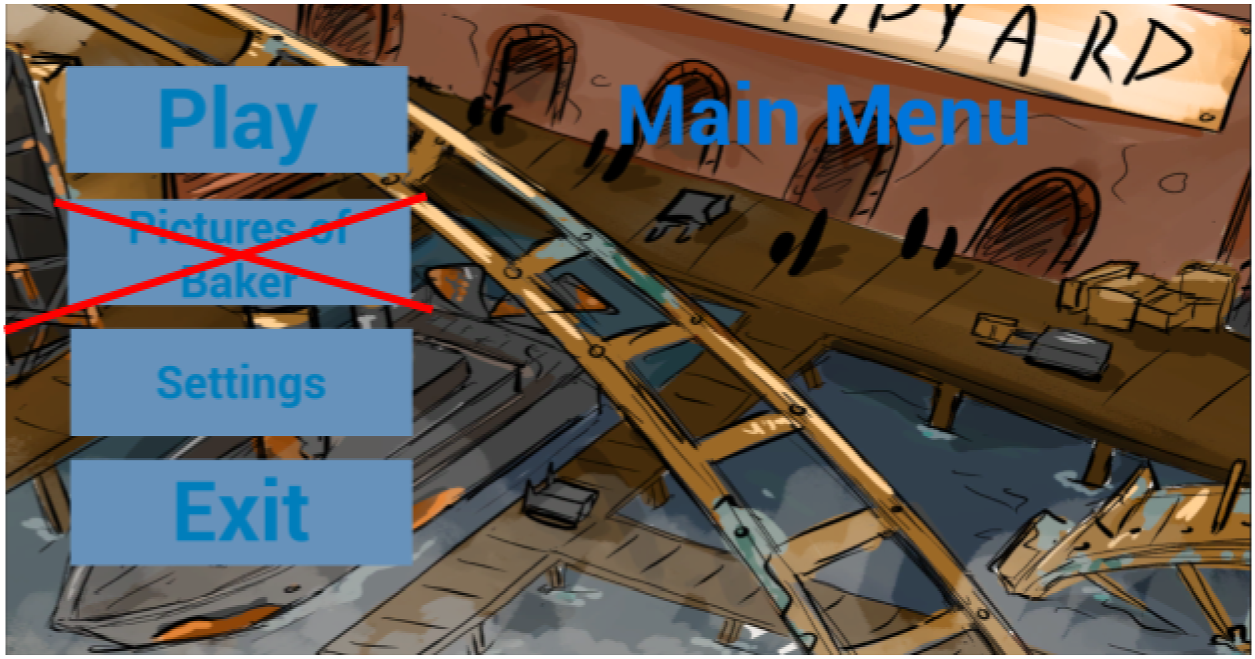
The inventory is organized into multiple parts. In the middle is where all the items the player has picked up are displayed. This includes things like consumable items, lore items, and upgrades, referred to as modules, are stored. The inventory is not sortable by the player as all items start at the zeroth slot, and items will automatically sort themselves to start at the first slot.

When the mouse is hovered over an item, a small description will appear. When clicked, a hover menu appears with the options to drop the item, or use the item. Dropping an item will remove it from the inventory and place it back into the world, and the use button will do different things depending on the item.

When a consumable item is used, it will apply its effect to the player character, and disappear from the inventory. The rest of the inventory will shift down so no empty spots are left over.

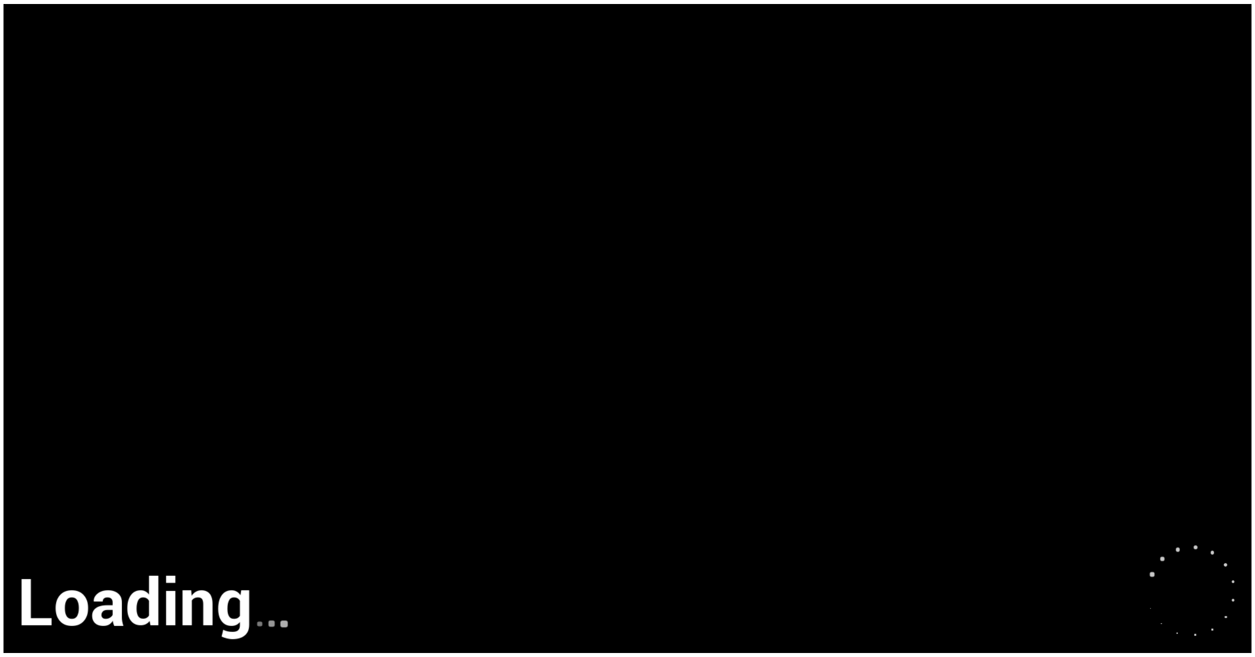
In the modules UI, modules will stack horizontally as the player continues to apply them. The Pressure Bandwidth bar will fill as models are applied, and a visual effect will be shown when a player is extra moist or overheated.

## Main Menu



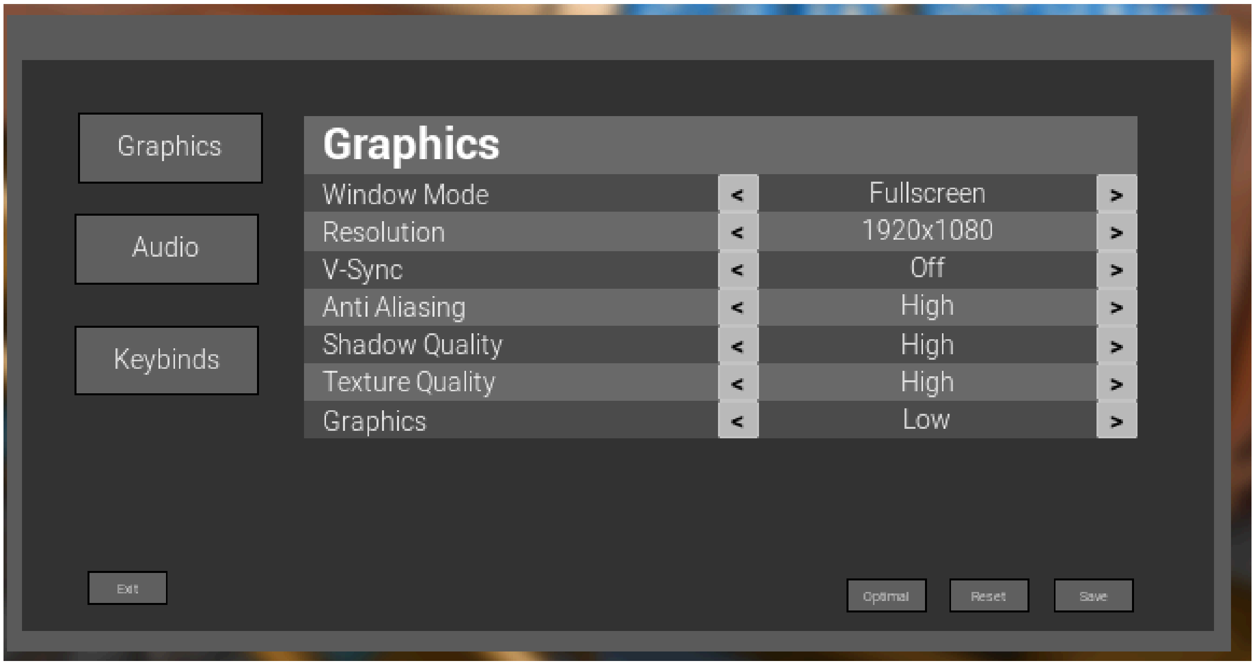
The Main Menu is the metaphorical health to any good game out in circulation. It stands as the gateway to the culmination of tons of effort. Everyone understands the importance of a first impression in real life and what is important to remember is that the relationship aforementioned is also applicable to the relationship between the player and the UI. Now that you understand the intricacies behind the psychology of a good Main Menu, we have decided to implement ~~4 key slots~~ 3 key slots. The play button is the most important button of all as it serves as the portal between the first impression and the actual game. The next key button is the settings menu which is elaborated on below. Lastly, we have the exit button that is fairly self explanatory. This button is meant to simply exit from the game, allowing the player to resume at a later date.

## Loading Screen



A loading screen is meant to signal the movement of a player to a new level. Our transition is made up of a trigger box. When the player chances upon the hitboxes of the trigger box, the player is then shown the transition UI element while their character and the map load in the new level. The loading screen will also have different elements such as the loading text, a progress wheel, tooltips about the game, and very important elements to enhance the experience of loading between levels. All of these combined make the experience of loading more believable and trustworthy than the level is loading.

# Settings Menu

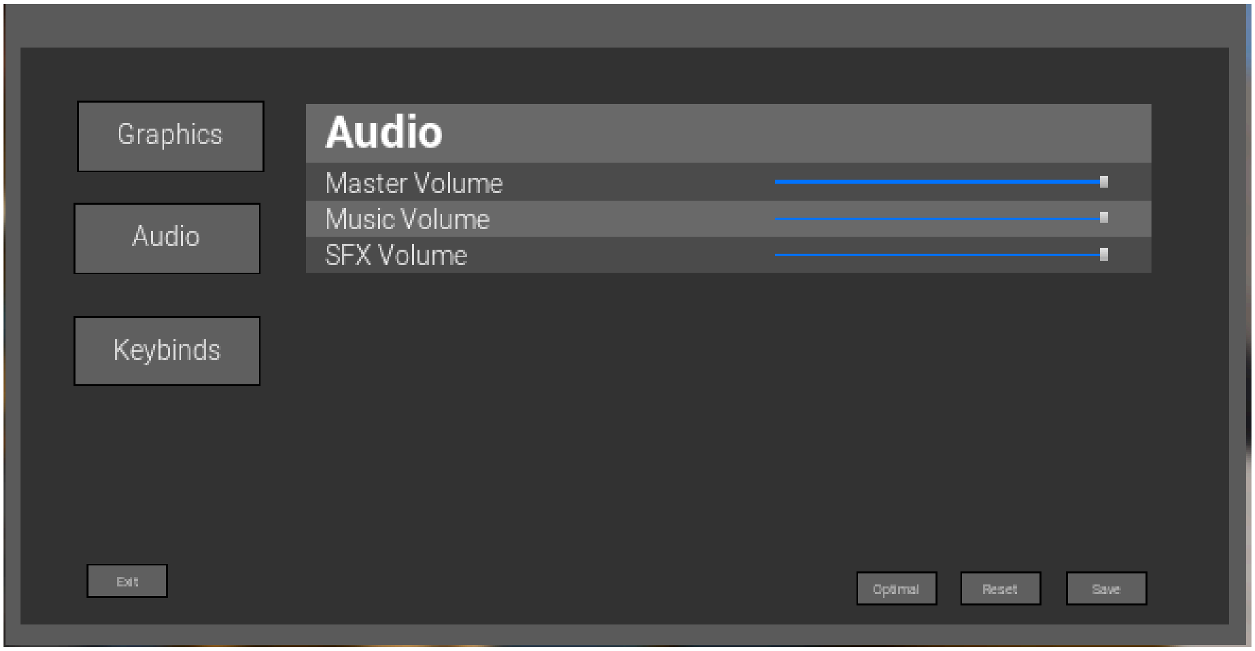


## Graphics

| | |
|---|---|
| Window Mode | Fullscreen |
| Resolution | 1920x1080 |
| V-Sync | Off |
| Anti Aliasing | High |
| Shadow Quality | High |
| Texture Quality | High |
| Graphics | Low |

The settings menu has three different menus that you can choose from when pausing the game (or while in the main menu). Using content switches, you can cycle between each of the windows based on whatever you want to change. Different tabs at the bottom of the menu will change slightly depending on what content browser you are on. The two options that will remain consistent will be the save button and the exit button which will save the settings changes and exit the menu accordingly. The other potential buttons that can be utilized are optimal and reset. Optimal will run a hardware benchmark on your computer and will adjust all graphics settings based on whatever your computer can handle. The reset button will do as the name states, it will reset settings back to what they were before changes were made.

- Graphics Menu (Photo is shown above)
    - The graphics menu features a total of seven different options that can be changed by the user. These options are Window Mode, Resolution, V-Sync, Anti-Aliasing, Shadow Quality, Texture Quality, and Graphics. Settings Anti-Aliasing, Shadow Quality, Texture Quality, and Graphics can be adjusted from a scale of low to ultra.VSync is switched from on and off, Window Mode has three options, Full Screen, Windowed, and Windowed Fullscreen, and Resolution has a total of 5 different options to choose from. The Graphics option is slightly different from the other options as it will adjust any engine scalability setting that is not specified in the list of options shown in the current graphics menu.
- Audio Menu

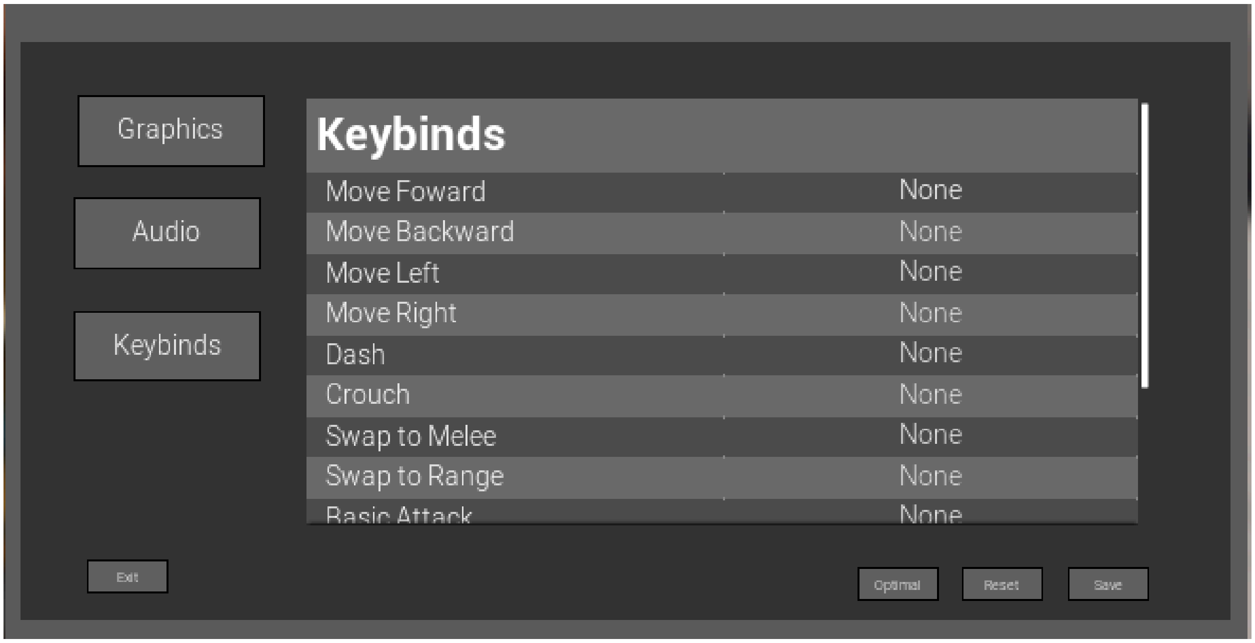
- 

## Audio

Master Volume

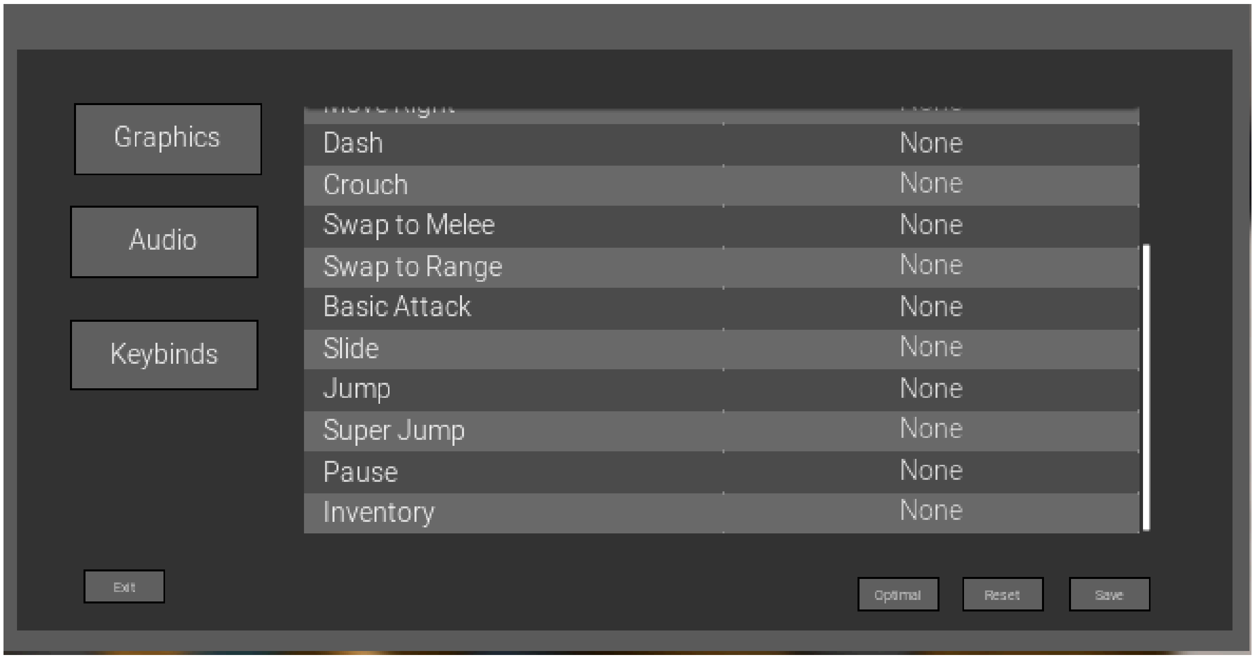
Music Volume

SFX Volume

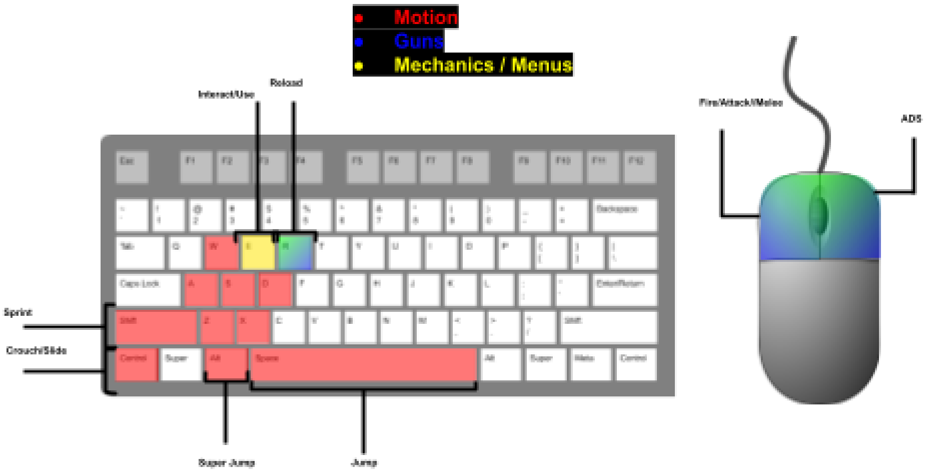
  - The Audio Menu has the same UI elements as the graphics menu, and now has 3 sliders to adjust various audio options. The master volume, as its name suggests, will control all other audio settings. Therefore, if the master volume is turned down, for example, the music and SFX volumes will follow suit and decrease as well. The music and SFX volume settings also feature the slider that will change volume accordingly for each setting.
- Keybind Menu
- 

## Keybinds

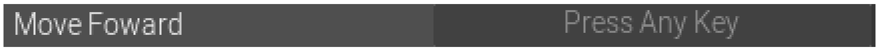
| | |
|---|---|
| Move Foward | None |
| Move Backward | None |
| Move Left | None |
| Move Right | None |
| Dash | None |
| Crouch | None |
| Swap to Melee | None |
| Swap to Range | None |
| Basic Attack | None |

- 

| | |
|---|---|
| Move Right | None |
| Dash | None |
| Crouch | None |
| Swap to Melee | None |
| Swap to Range | None |
| Basic Attack | None |
| Slide | None |
| Jump | None |
| Super Jump | None |
| Pause | None |
| Inventory | None |

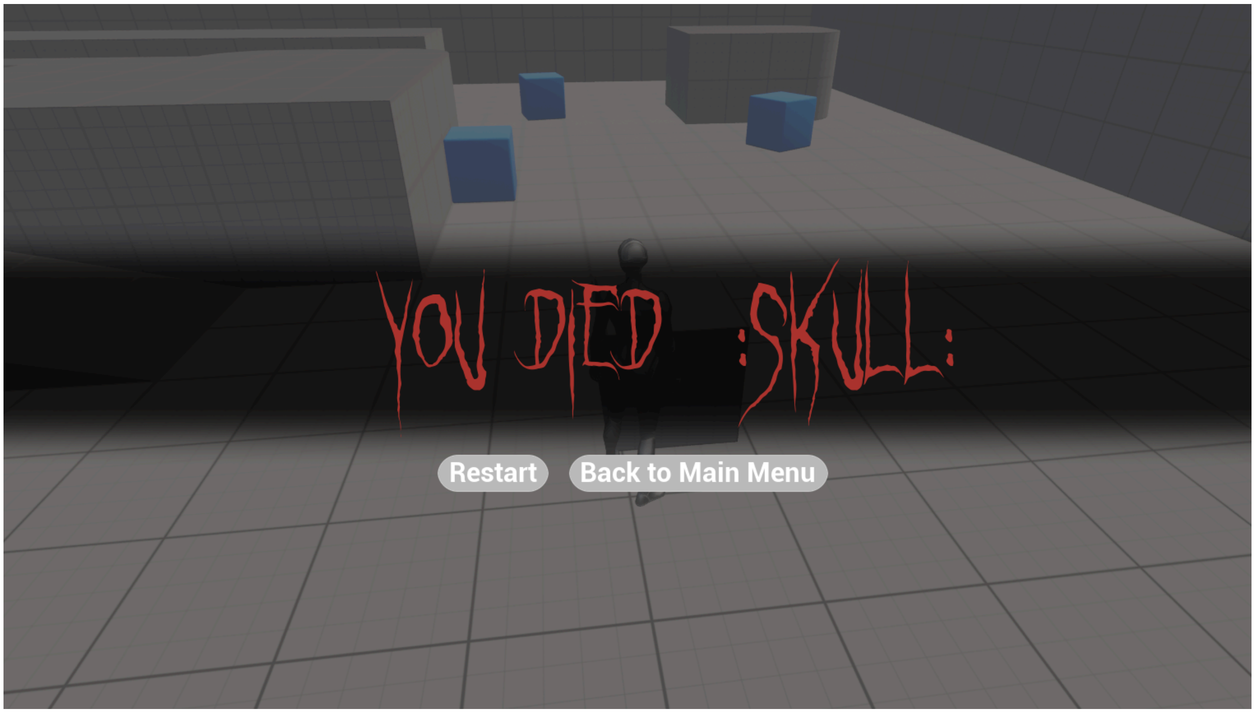
- 

    ○ The keybinds menu, once again, features the same UI elements as the other content browsers. To accommodate the vast amount of keybindings that the user may modify, a slider is used to scroll down to settings not currently visible. This browser currently supports 11 different keybind settings. These settings are Move Forward, Backward, Left and Right, Dash, Crouch, Switch to Melee/Range, Basic Attack, Jump, Super Jump, Slide, Pause, and Inventory. Once the user clicks inside the space that says "None", the menu will prompt the user to click a button on their keyboard to change that setting.



| | |
|---|---|
| Move Foward | Press Any Key |

## Death Screen

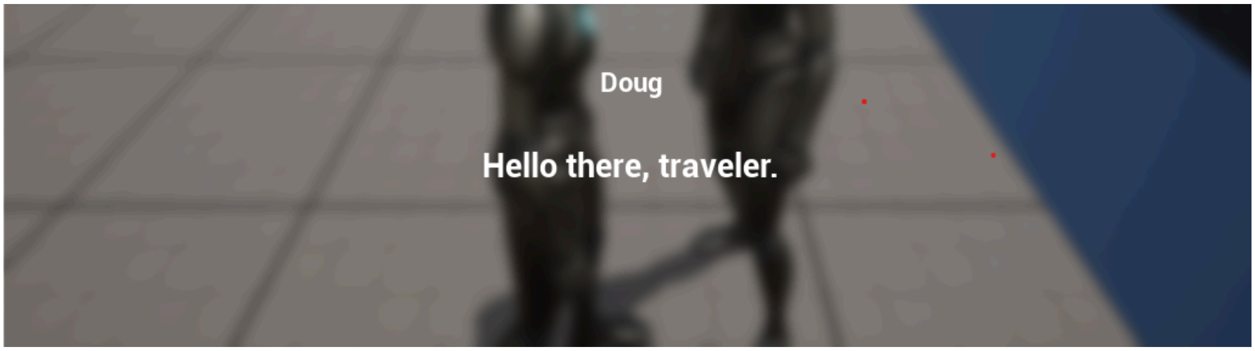


The Death Screen, as the name suggests, will play on player Death. The Death Screen features a fade in motion that will cause the Death Screen to slowly enter the viewport of the player. This screen will also lock the player screen so that no movement will be allowed while dead, and two buttons will be available for the user to press once the scene has fully been displayed. These buttons will be one to return the user to the main menu screen, and another option to restart. The restart option will bring the user back to their previous save point before they die (subject to change).

## Quest Menu

## NPC Dialogue

When the player talks to NPCs, the NPC dialogue text box will appear on the screen. The NPC dialogue text box consists of a simple gray square that hovers on the lower half of the screen. The NPC's name can also be found at the top of the text box. The NPC dialogue appears below the name and is also animated, meaning letters and characters in the dialogue appear over time.



Doug

Hello there, traveler.

## Soundtrack

1. Scrapyard Shuffle
   - Plays during the sneaking section of the junkyard. There are two versions.
     - The flight version plays while the player is being stealthy. It’s quiet and empty, which is important when you’re trying to not be seen. The fight version plays while an enemy is aware of the player.
     - This version is louder, as you’re distracted from guards.
2. COGPILE
   - Plays when you get caught by the lights in the junkyard and enemies start swarming toward you. It is a play on *dogpile* and *cogwheel*.
3. Keyboard Kalamity
   - Boss music. Uses keyboard instruments.

# Gameplay Mechanics

## Movement Abilities

The character will have basic movement abilities to walk, run, jump, and crouch, as well as more advanced movement such as the ability to slide, vault over ledges, and mantle ledges. By defeating bosses, The player can unlock the ability to use a “Super Jump”. This jump triples the player’s jump velocity so that you can jump three times as high, this ability uses double the amount of pressure as a normal jump. The ability can be upgraded to include a shockwave area of effect attack that damages everything around the player’s landing area. The shockwave attack by itself can be triggered by pressing the Super Jump key while airborne.

Additionally, different areas and levels of the game feature unique challenges the player would have to overcome. For example, once the player leaves the starting area and enters the junkyard, the game transforms into a stealth-like operation, with the player needing to cross the maze-like junkyard while avoiding enemies who are patrolling defined routes along the setting. Apart from that, the junkyard features large sweeping searchlights constantly scanning the park for signs of intrusion. Upon sensing a player, the spotlight will turn red, sirens situated along every light post will start blaring and glowing, and all enemies in a somewhat nearby vicinity will immediately be drawn to investigate the location of the spotlight. At this point in the game, the player will not have a weapon to fight back against enemies, and since a spotlight locked on a player never locks off, the player will most likely be met with certain death. Throughout the game the player experiences various interactable elements of the map such as; a fractured wall that can be destroyed upon impact, a wall for the player to vault over, areas to crouch and slide under, and places to avoid sight from enemies - all functions designed to teach the player mechanics.

The Industrial area is filled with buildings producing mechanical parts for the city to function and is guarded by patrolling mechs. Goods are transported between factories through a system of conveyor belts and bridges which all center around a river that provides the power necessary for the industrial sector to function via waterwheels. The final boss of the area, the twins, feature unique mechanics different from other bosses, in the sense that 2 bosses are working together to defeat the player. One enemy strafes around the player and dives in for melee attacks, while the other shoots ranged projectiles while simultaneously teleports around the arena. These bosses can both heal the other boss, but not themselves, adding new challenges for the player to overcome.

## Scrap

Scrap is the general currency in the game. Scrap is represented in the game world by bits of scrap metal - pipes, nuts, screws, gears, etc. and can be obtained in a few ways. The most common method of obtaining scrap is by killing enemies. Most common enemies the player will find around the world will drop a fair amount of scrap when they're killed, representing the pieces of their mech suits falling off. After the scrap is dropped, the player can go collect it - standing close to a piece of scrap will add it to the player's inventory. Scrap can also be obtained from NPCs, as rewards for completing quests, or may be hidden around the world.

Once the player collects the scrap, it has a couple of uses. First and foremost, it's a currency. The player can exchange their collected scrap for upgrades to their armor, weaponry, or other items they may need. It can also be used as ammunition for the scrap spitter - a unique weapon that flings scrap at the enemies.

## Pressure

Many of the more advanced player abilities are limited by the amount of Pressure they have available. The amount of pressure available is viewable in the bottom left of the UI as a bar. Pressure will regenerate passively as the player's mech suit "boils" more water. Pressure will regenerate at a base rate when the player is moving or walking, and slightly slower when the player is sprinting.

When an ability is used, a set amount of pressure is used. If not enough pressure is available, the ability will not be activated. Any changes in the player's pressure will be reflected in the status bar. When the player is at maximum capacity for pressure, a visual effect will appear, and a higher-power attack will be available.

## Quests

The player is also able to receive quests as they advance in the game. Quests can be given to the player from NPCs that can be found dotted around different locations. After speaking to an NPC that can give the player a quest, the player will receive that quest and will be tasked to complete it. Quests can give the player a multitude of tasks to complete such as visiting a location, defeating several enemies, or a specific item, and meeting other specific requirements. Upon completing a quest, the player must report back to the NPC that has given them their quest. Upon talking to the NPC, the player will be given a reward for completing the quest. Rewards can vary depending on the difficulty or length of the quest but they mainly consist of scrap (in-game currency), story items, weapons, upgrades, and other items that serve a benefit to the player.

## Single Time Consumables

## Modules

The module system acts as a way for the player to apply upgrades to their character, and is inspired from the charms system from *Hollow Knight*. There will be a variety of modules available in the map for the player to find, which will be stored in the inventory as a normal item. Modules can be installed and swapped at repair stations. Modules each take up a specified amount of “Steam bandwidth” the player has available, so they must carefully choose which modules they want installed. Armor is also included as a module, and can be installed similarly. Similar modules can stack their effects, but many modules have unique functions that do not overlap.

There are two special effects a player’s module load out can apply. The first, called *overheated*, is when all the player’s steam bandwidth is used. There will be a visual effect indicating this, and in this state, the player’s maximum pressure will decrease to half the base capacity, and pressure regeneration will be extremely slow.

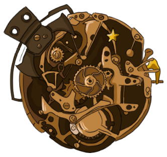
Alternatively, the player may have the *extra moist* effect. *Extra moist* is applied to the player when no modules are installed and zero steam bandwidth is consumed. While *extra moist*, there will be a visual effect indicating this, and the player’s maximum pressure capacity is doubled from the base capacity, and pressure regenerates significantly quicker.

Even while not *overheated* or *extra moist*, the player’s pressure regeneration speed will depend on how much steam bandwidth is used. As more modules are installed and less bandwidth is available, pressure regenerates slower.

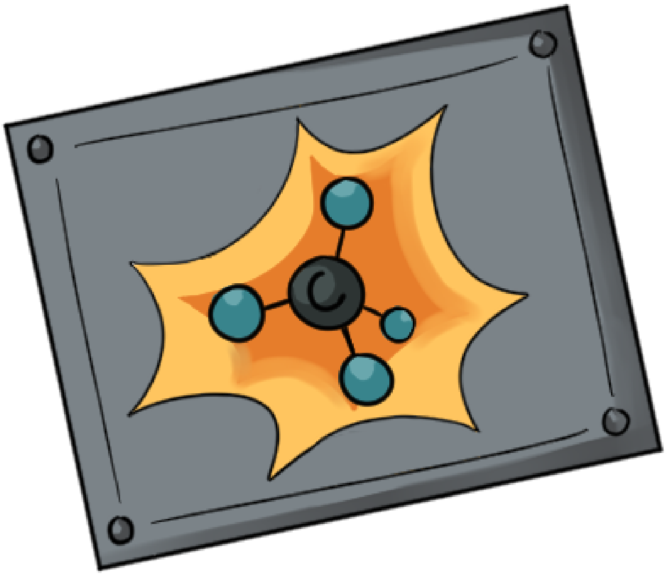
### Available Modules:

- Armor modules come in several varieties and rarities, and can be found around the map

- Gear Replication Device: small percentage bonus to any scrap picked up

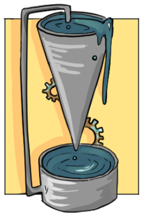


- High Carbon Steel: Increases the health capacity of the player

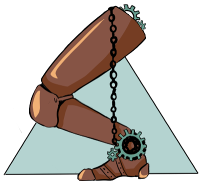


- More Moist Machines: Increases the rate at which the player generates pressure

- Wetter Water: Expands the player’s pressure capacity



- Overclocked Legs: Increases the base walk and sprint speed of the player



- Health Hound: Heals player by 5% of any damage dealt

## Saving

Save files are fully supported by the game. Saves are split into several different files based on what is contained in them. There is a file for the player’s inventory and map items that have been picked up, and a file for all questing data and game progression.

Items are serialized through the use of a prebuilt registry of all items, that associates a string key with the runtime type of an item. Items are loaded into the game by reading the list of strings and finding their matching runtime value.

# Weapons

The game will have both melee and ranged weapons. Melee weapons will do more DPS but require the player to go up close to the enemy. Basic weapons will be available for purchase or automatically granted at a certain part of the story. Advanced weapons will be a random enemy or boss drop.

## Melee

Melee weapons are attached to the right hand.

Punch- the basic melee attack that the character starts with

### Sword

The sword is in the middle ground of both attack and

### Spear

The spear has the longest range but also the smallest

### Axe

The axe has the shortest range but has the added

| | | |
|---|---|---|
| range, with more attack than the spear but less than the axe and more range than the axe but less than the spear. | point of penetration meaning it has the least damage | benefit of blunt damage, making it the hardest hitting of all the |

The player gets to decide which weapon is best for them and fits their playstyle the best.

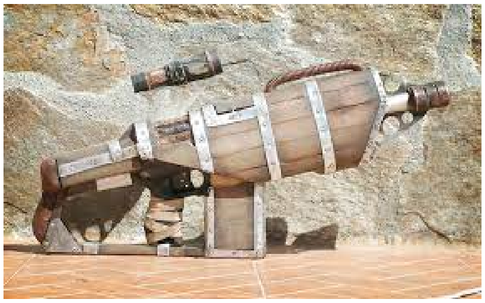
## Ranged

Ranged weapons are attached to the player’s left hand.

### Scrap Spitter

The Scrap Spitter is an old abandoned gadget found in the junkyard. It shoots scrap from your inventory at a pretty high rate. After the player collects scrap around the junkyard, it will be used as ammo for the scrap spitter to defeat the junkyard boss. The Scrap Spitter can be found in a small room right before the junkyard boss fight.

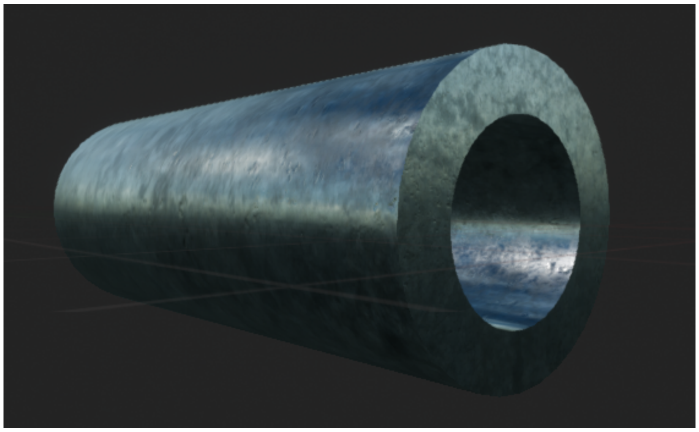
Scrap Spitter Reference Image:



### Cannon

The Cannon can be bought at the shop for a certain amount of scrap. It is a small cannon modified to be able to mount on a mech. It shoots cannonballs at a decent speed with a good explosion radius and decent damage. It should deal slightly less damage than the scrap spitter but should not require any scrap as ammo.

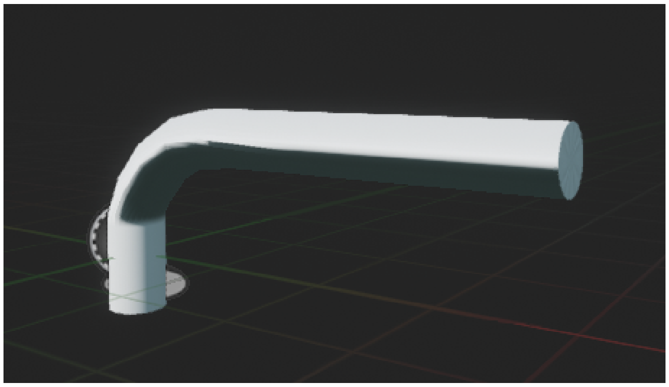
Cannon Concept Model:



## Flintlock

The Flintlock can be bought at the shop and it is more expensive than the Cannon. It is a mechanical gun that shoots a fast bullet and can hit targets from a long range. It uses a hitscan projectile and a small explosion. Additionally, It deals a lot of damage but has a slow fire rate.

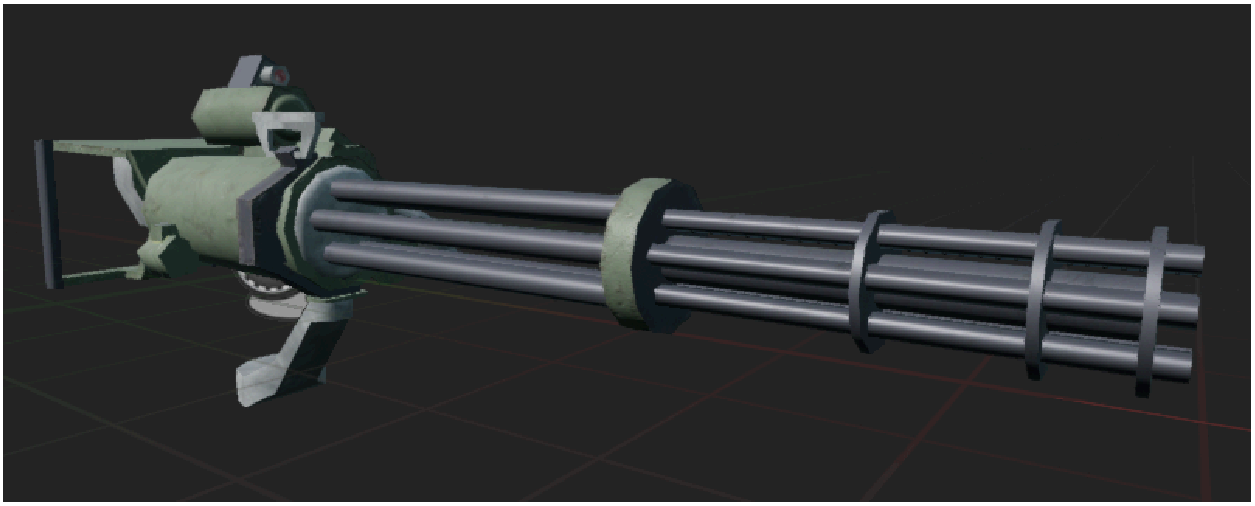
Flintlock Concept Model:



## Minigun

The Minigun can be bought at the shop and requires a lot of scrap to be purchased (enough to require farming). It is the best ranged weapon in the game as long as you can aim. It is an advanced minigun that shoots very fast but has some spread. The DPS will be very high up close, but the spread will lower its damage output at range.

Minigun Model:



## Nail Gun

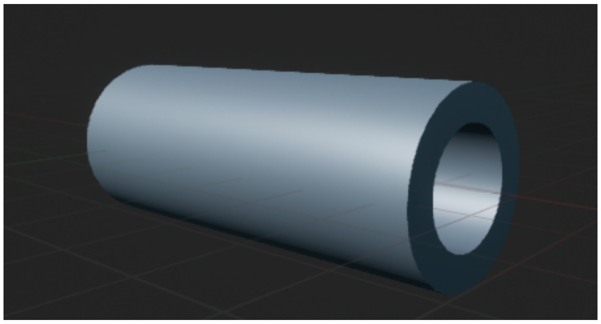
The Nail Gun is a random drop from armed industry workers in the industry. The Nail Gun is also used by industry workers to defend against any intruders. It shoots fast projectiles that deal pretty high damage. It should have higher DPS than the cannon and flintlock but slightly less than Minigun. Nails will deal good damage, have a small explosion and gravity, and be somewhat hard to hit.

Nail gun reference images:

## Missile Launcher

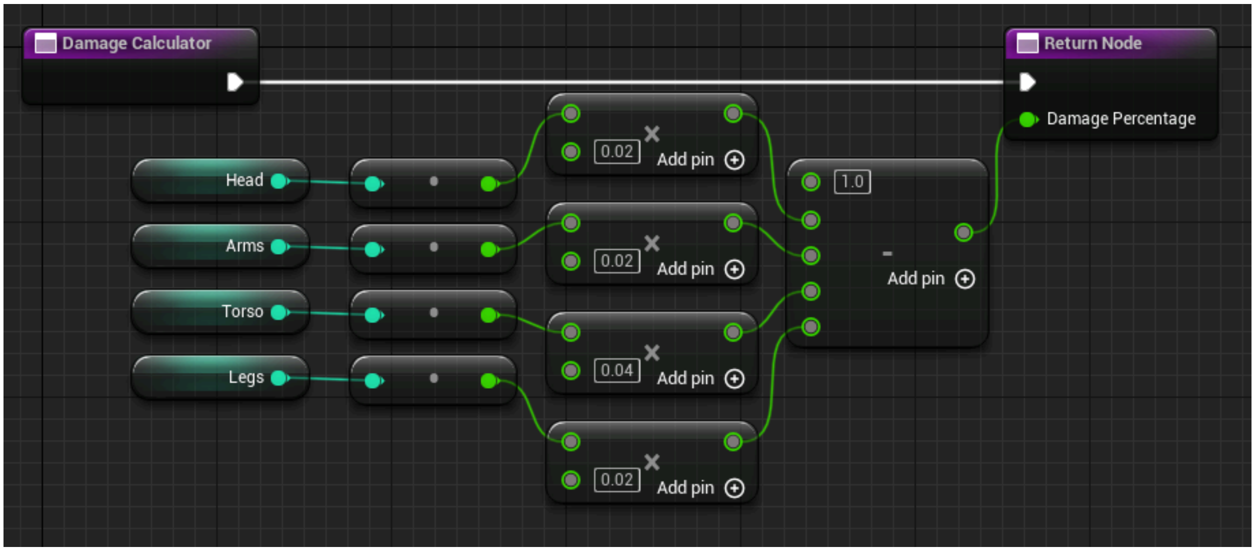
The Missile Launcher is a guaranteed one-time drop from Leonard (dockmaster). It is a tube that launches explosive-seeking missiles at enemies. Missiles have slow projectile speed but do great damage and seek enemies. Additionally, their explosion size will be the largest of any weapon. The fire rate will be the same as cannon, and overall DPS will be less than the nail gun but more than the flintlock.

Missile Launcher concept model:



# Armor

The level of the armor of each piece is averaged together to determine what percentage of damage will be translated to actual health being taken away. The arms, legs, and heads constitute 20 percent of armor each while the torso is 40.



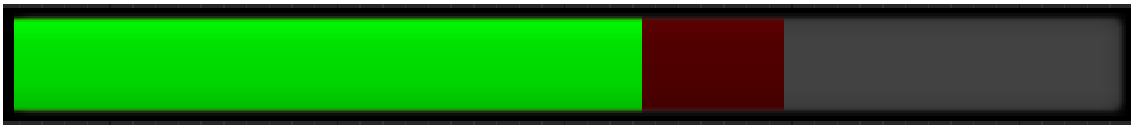
The armor will be equipable in the inventory through the module system.

PLACEHOLDER

There will be multiple levels of rarity that will impact the damage reduction. A full kit of the best armor will reduce damage by 50%

# Enemies

All enemies have their respective health which is conveyed to the player with the use of a health bar. To prevent extensive clustering in the case of large swarms of enemies, health bars are naturally invisible. When an enemy takes damage from any source, their respective health bar becomes visible again for a few seconds, shows (in red) how much damage they just took, and then proceeds to fade out. Have a health bar that appears after the enemy takes damage.



When enemies run out of health, whether through natural causes or from player damage, their AI gets voided and their model begins to ragdoll on the floor in the direction of the lethal damage that killed them. Additionally, when an enemy dies, scrap in random quantities will explode out of them in all directions, resting on the floor for a while before despawning. This scrap is the base reward for enemies and can be picked up by the player and used to acquire upgrades for their mech. Some enemies that contain external equipment such as the Junkyard Guards

Most enemies share the same functionality, such as sight and sound systems with slightly tweaked values. The majority of enemies have a given range of vision, where they are allowed to see the player if their head makes a line of sight with the player. However, the vision of enemies is limited to a vision cone extending out of the front of their head, up to the limit of their range of vision. Upon seeing a player through the line of sight, enemies, unless they are spawned to immediately go after the player (see the Dock Master boss fight), the enemy will leave their idle/patrolling state and enter an attacking state where they will begin to chase the player and attack them. Ranged enemies, however, stay at a distance and shoot the player down from afar. Enemies additionally have a hearing system that allows them to hear sounds made by the player. Upon sensing this sound, enemies that are not already in an attacking state enter a change from their idle/patrolling state to an investigating one. When investigating, enemies will walk up to the point of investigation, where the sound was heard, and then proceed to look around for the player. If no player is found, the enemy will either return to patrolling if they have a patrol route, or return to their idle state.

# Humanoid Enemies

## Junkyard Guards

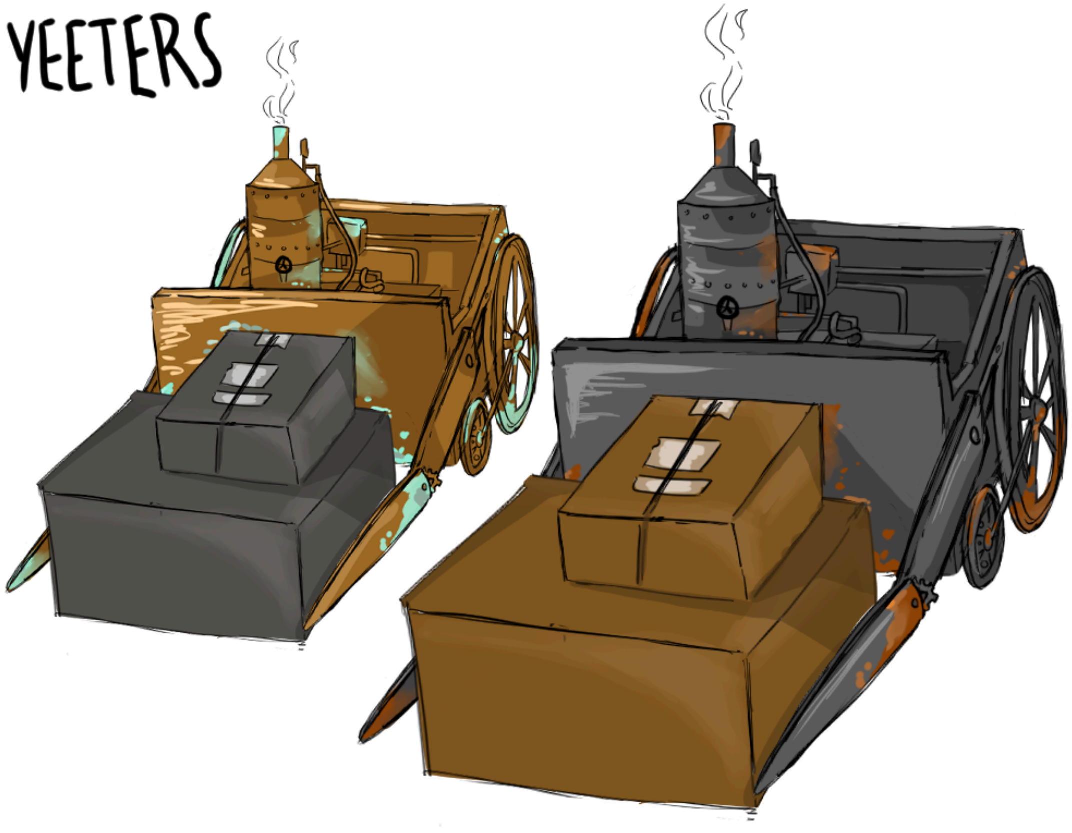
They patrol the junkyard, moving along a predetermined path. When a player is spotted … TODO

## Industry Worker

While the player is traversing through the industrial district, industry workers fire their nail guns at the player. There are several that exist in different parts of the district.

# Other Enemies

## Yeeters



# Bosses

## Junkyard Boss (Junkyard Joe)

- The Junk Guard is the first boss of the game. When the player reaches the end of the maze in the junkyard, and grabs the time stone, they will be suddenly met with the junk guard coming down from the watchtower (where he was spying on the player) and have to beat him. This boss mainly serves as a way to further introduce the core mechanics of the game, and also the lore in general.

- Description of the fight: Junk guard has 2 phases.

- Attacks (Phase 1):

  - Scrap whip: the boss will wind up into a swing and swirl his weapon by the ground.

  - Scrap rain: the boss will erupt and eject some scrap particles into the sky that will rain down onto the ground.

  - Smashdown: the boss jumps to a certain area around the arena, dealing area of effect damage.

  - Metal mash: the boss conjures a big metal fist and attempts to throw a punch

  - Metallic Pike: the boss strikes his weapon into the ground and scrap spikes come out from the ground at random places

- Attacks (Phase 2):

  - Phase 1 attacks (scrap whip, scrap rain, smash down, metal mash, metallic pike) but slightly faster

  - Carbon shockwave: the boss emits 3 shockwaves each spat out in short intervals

  - Rusted Waterfall: around the arena, some areas will be indicated with a marker before a bunch of metal scraps fall onto those areas.

  - Aluminum outburst: the boss will repeatedly fire many scrap metals in 5 directions

## Dock Master (Lennard)

DOCK
MASTER

## MA Boss

- MA Boss, also known as the “twins”, is a boss found at the end of the industrial district, inside the large circular dome by the northernmost end of the factory.
- The boss has a singular phase, but manifests as two separate entities, both of which the player needs to eliminate to beat the boss. Apart from the boss, the arena has additional hazards for the player, with two extremely hot fire tornadoes circling around the boss arena. These tornados, which the boss is completely immune to, force the player to constantly be on the move to avoid burning, and a large central pillar exists in the center of the arena to eliminate a central safe spot from the spinning tornados.
- One of the entities, or twins, that MA Boss consists of is the melee twin, holding a large metal sword. This boss will constantly attempt to run at the player, and when in close proximity, will begin strafing around them. At a random interval while they are strafing, the boss will come in for a quick attack (a randomized attack out of 4) and go back to strafing mode. If the player runs away, the boss will run back to the player’s position and begin strafing once more. Additionally, the boss has extremely high damage negation from all sources when they are not attacking. This means that for the player to deal meaningful damage to the

boss, they have to wait for the boss to strafe and commit an attack, and in the interval where the boss is swinging his sword lies an opportunity for the player to damage the boss.

- The other entity is the ranged twin, who utilizes magic coming from the palms of her hands. This boss will try to stay a set distance away from the player, and will make sure it isn't too close or too far from the player. The boss also has the ability to teleport around the arena, indicated by the boss during into a purple misty orb and quickly zipping to another spot before rematerializing back into reality. The boss will teleport in two specific cases:
    - 1) If the player moves to a spot where the ranged twin cannot see the player, the boss has a chance, instead of sprinting to a spot where it can see the player, to teleport to a spot where it can see the player.
    - 2) Every time the boss does an attack, it will always do 2 teleports. First, it will attempt to do a quick teleport to a ranged location where the boss can see the player, but the player cannot see the boss. This could be directly behind the player, or far to the left or right, making the boss harder to track. Immediately after this teleport, the boss will begin casting one of their attacks. As soon as the boss casts their spell, they will immediately perform the same type of teleport mentioned previously, teleporting to a spot out of the players vision where the boss can still see the player.
- The boss can cast 3 distinct spells when they attack. The first is a giant, quick, electric ball of energy that it will throw in the direction of the player dealing a significant amount of damage. The second attack the boss could do is one where they summon a flaming orb that will slowly advance and home to the player. This flaming orb will advance to the player infinitely, so the only way to make it go away is for either the player to be hit, or for the player to bait the flaming orb into hitting another enemy or wall. The final attack the boss could do is where they raise their arms in their and a magical sigil appears under them. Immediately after the sigil appears, a normal mage enemy will summoned in to join the fight. This enemy acts similar to the ranged twin but is significantly weaker. It has less health, strafes to locations instead of teleporting, and has only a singular magic attack it can use which flings a low-damage but fast comet of energy to the player.

## Rekab

- Rekab is the final boss in the game. The boss fight occurs inside a cave underneath his lab in the city. He wears a large mech suit with a time crystal encased within the suit's chest cavity. During the player's encounter with Rekab, the player will be tasked with opening Rekab's chest cavity by destroying it. This, in turn, will reveal the time crystal hidden inside that the player will have to destroy as well to defeat Rekab and stop the time loop.

- Rekab has 4 phases. In the first phase, he uses simple attacks such as Smash and Rock Throw. In every subsequent phase, Rekab turns back time, which causes him to lose a bit of health but also makes him more unstable and faster. He also gains a unique style of attacks for each phase, with all of them combined in his 4th phase. Rekab is mostly stationary, attacking from a range (even with his melee attacks). After Rekab reaches his 4th phase, his time crystal becomes fully exposed for the player to destroy.
- Attacks:
    - Smash (Phase 1, 3): Tries to smash player with arms or legs
    - Rock Throw (Phase 1-3): Pick up and throw rocks at player
    - Stalactites (Phase 2+): Makes stalactites rain from the ceiling. A circle on the ground briefly appears to indicate where stalactites will drop.
    - Summon (Phase 2+): Sends smaller enemies out to distract the player
    - Time Beam (Phase 3+): Shoots a bright blue beam out of his chest.
    - Circle Laser (Phase 3+): Places a large circle indicator on the ground at the player location. After 5 seconds, Rekab shoots a laser at an antenna on the top of the cave, refracting towards the circle and splitting into multiple lasers that rotate from the circumference to the center and back out.
- **The story, all names, characters, and incidents portrayed in this production are fictitious. No identification with actual persons (living or deceased), places, buildings, and products is intended or should be inferred and are merely coincidental.**

## Unique Drops

As a reward for defeating difficult enemies, the player may be rewarded with a unique weapon that they can use later on. For example, Lennard (the Dock Master) drops a Missile Launcher that cannot be found anywhere else in the game world. Another boss enemy, MA Boss, drops a unique Jetpack that assists the player's mobility. These weapons and armor pieces cannot be found anywhere else in the game world.

| Unique Item | Type | Location | How to obtain |
|---|---|---|---|
| M100 Missile Launcher | Weapon | Docks | Defeat Leonard |
| Jetpack | Armor Modification | Industry | Defeat Ma Boss |
| Scrap Spitter | Weapon | Junkyard | Found before the entrance to the first Boss arena |

# Cutscenes

Are there any movies/cutscenes in the game? How are they presented to the player? Describe the way that they will be created, using UE5, animation software, or other means. Describe what the player will be seeing, and why it is important to the story. Does the player have control during the cutscenes?

# Bonus Materials

Give some examples of things the players will be able to unlock. What is the player's incentive to play? This is where you would mention things like multiplayer, downloadable content, episodic content, and so on. (We probably won't have a lot about this, but one big one we have is the split story paths)

# Version Control

## Hosting

Version control was a big challenge to set up this year. FriscoISD has blocked access to github over their network, and many of the other hosting services (Gitlab, Bitbucket, etc) had repository size limits that were too small for an Unreal Engine project. So instead, we self hosted the repository initially using free cloud resources from Oracle's Cloud Infrastructure. This server had very little computing resources available to it, but it was functional for our needs at first, since everyone was initially working on their own, independent branches.

However, once we began to combine branches and merge code together, the limitations of the remote server were very restricting. The server had only a single gigabyte of physical RAM available, and even in the initial stages of the repository this was not enough to handle a full clone (during a clone, the entire repository is compressed and stored in RAM, and only then is data transferred over the internet. 1GB was not large enough even for the compressed repository). We needed to add SWAP space to the machine for anyone to even make the initial clone, but this was more of a band aid, as SWAP space is very slow compared to RAM, so full clones and pulls would take upwards of an hour. Partial, branch specific clones did take under a minute.

During the winter break, we attempted to fix some of the issues primarily with transfer speeds of data. Although we were able to get better hardware, the issues we were encountering were more complex than just hardware limitations, and more the fault of the architecture of git itself.

## Software

The git server runs the Ubuntu LTS version 22.04. Repositories were initially accessed through a raw ssh connection into the machine, with basic security measures in place, including a separate, non-sudoer git user with a restricted shell. However, although this provided security to the machine, it did not offer any protection to the repository itself. In this system, anyone was able to make changes to the master branch, and perform force pushes. Additionally, our class helped the other class set up their git repository on the same machine since they did not have members with background in setting up servers, so anyone in either class could clone the other's repository.

Because of these reasons, we switched to an open source project, gitolite, to manage repositories and access control. Gitolite manages access to multiple repositories on the same machine, and includes repository hooks to check reads and writes before they are accepted to reject things like force pushes and changes on master without the correct permissions. This had the added benefit of preventing someone from one class to clone the other classe's repository, as well as a more refined method of managing permissions.

In order to help with setup, several scripts were written to automate the process of setting git up on computers, and simplify the actual usage of git. The primary install script would automatically download all needed data and configure git appropriately. The process required that the git command line was installed on the computer, and from there only required a single command to begin the process.

This initial command downloaded the full script from a webserver also accessed on the machine hosting the git repositories, and would automatically prompt the user for their name and email to configure git, and determine which ssh key to install on their computer. From there, it would download a pre-written ssh configuration file, adjust the file to match to the correct key, and place it in the ~/.ssh directory. It would then download an archive of all ssh keys, pick the corresponding key from the archive and delete the others, and install this to the ~/.ssh directory as well.

The second script added aliases to the ~/.bashrc file that would simplify common git commands to single-word aliases. This included aliases to change directly into the project directory (cdproj), stage, commit, and push changes (commit), and fetch and pull updates from the remote server (pull).

## Usage

Git is not designed to be used with the kind of large binary files generated by Unreal Engine. This created many problems when using git. Although Unreal has an in-beta git plugin available, it is not well made and does not function properly. Additionally, many of the graphical git interfaces relied on github for distributing binaries, even within the installer itself, so we were not able to install any high level interfaces. Instead, we relied solely on the portable, command line version of git available on the git website. Most of the class had no experience on the command line, so we spent several days going over command line use, in addition to how to use git in the first place.

For simple commits and pushes, git worked perfectly fine. Although all the files were in proprietary Unreal formats, Unreal did fragment the data enough to where none of the files were too large. However, problems did appear when we began merging branches. Unreal does not have a function diff editor, so we would have to have two projects open and manually copy paste data from one project to another to resolve conflicts.

To minimize this, we heavily relied on Unreal's "component" classes. This allowed us to encapsulate functionality into their own classes, and apply these components to objects that needed them with a few clicks in the interface. This helped simplify the process of merging branches, as well as keeping the main player class lightweight and modular.

# Item Descriptions

Scrap: The discarded pieces of your enemies' mechs, used as currency.

*What kind of sicko uses the bodies of the fallen as money? Oh, wait, it's you.*

Sword: A quickly constructed one-handed bastard's sword. A staple weapon of mech pilots when outside of their mech. Capable of cutting down many an unarmored foe.

*Calling it a bastard's sword is not an insult to the wielder, it's a real type of sword I swear.*

Axe: A large lumber axe. This works quite well against armored opponents.

*Cuts down enemies and trees alike.*

Spear: A sturdy wooden pole with a dagger attached to the end. This melee weapon's length makes it excellent at keeping others out of reach.

*If only someone figured out that you could throw these. Oh well.*

Time Stone: A strange glowing stone found in a pile of rubble in the junkyard. It gives off some peculiar vibes.

*Junkyard Joe died for this rock*

Scrap Spitter: A basic launcher of various pieces of junk, effective as a long-range weapon. (Costs scrap to fire.)

*This is horrific when you consider what the junk actually is.*

Cannon: Single-fire weapon that, while slow, deals massive damage to any enemy it hits.

*In case “plundering” is on your to-do list.*

Flintlock: A small hand cannon with devastating power and force. High damage and reload time.

*Where did you even find a functioning flintlock?*
*This probably belongs in a museum somewhere.*

Minigun: This weapon’s high rate of fire makes it extremely deadly., however it also chews through ammo like none other. Dodging this is near impossible.

*Say hello to my little friend*
*It costs 400,000 scrap to fire this weapon … for twelve seconds.*

Nail gun: This “weapon” may not do a lot of damage, but ammo is plentiful. Great as a backup weapon.

*Somehow still not as bad as stepping on one.*

Missile Launcher: The Mark 100 Missile launcher deals an incredibly high amount of AoE damage. It boasts an impressive amount of damage, with a slow reload and scarce ammo to boot.

*Don’t try rocket jumping.*
*What happened to the other ninety-nine?*

Key halves
A piece of a key, possibly used to unlock some kind of lab.
*Not very useful on its own.*

# Enemy Descriptions

Yeeter: The mysterious creature simply known as the “*Yeeter*” is some kind of unmanned mech, its primary function is unknown as whenever we get near them they throw things at us.

*Yeet*
Yaaa

Junkyard Employee: Don’t let these guards spot you, they are surprisingly dedicated to their jobs.

*If you don’t get spotted, some of these poor fools might get fired.*

Junkyard Joe: Junkyard Joe was a well respected member of the community. He guarded the junkyard with his life until the very end.

*Until you killed him over a rock.*

MA Boss: the individuals known as “Ani” and “Micheal” ruled over the industrial district of the city with a cast iron fist. Well, “Micheal” did at least, “Ani” was always just kind of there.

*It was satisfying to kick him out that window*

Rekab: The orchestrator of the time loop, all to save himself from a horrific death. How many lives were affected by the actions of this one man? I’ll never know for certain, and since no-one ever knew what was happening, neither will anyone else. But at least he’s dead now.

*In no way is the name of this character connected to any real-world individual, living or deceased.*