

# Game Proposal: Sky Seeker

CPSC 427 – Video Game Programming

## Team: Falene (Team 7)

Emma Park #33281130;  
Elton Tiet #22039192;  
Stephane Gallant #72846678;  
Benson Wu #76200195;  
Cecilia Liu #75597831;

## Story:

As the player, you start outside your house in the Abyss, a large city steeped in darkness. You will fight your way through different levels to try and reach the top so you can escape from the Abyss. You will progress through different rooms and fight many enemies. After defeating all enemies in a room, you may choose one of the offered items that will augment your combat abilities before proceeding to the next room of your choice. Throughout your journey, you will gain a multitude of items that will help you face the boss enemy that blocks your way to your goal. And if you die, well, there's always another person yearning to leave the abyss.

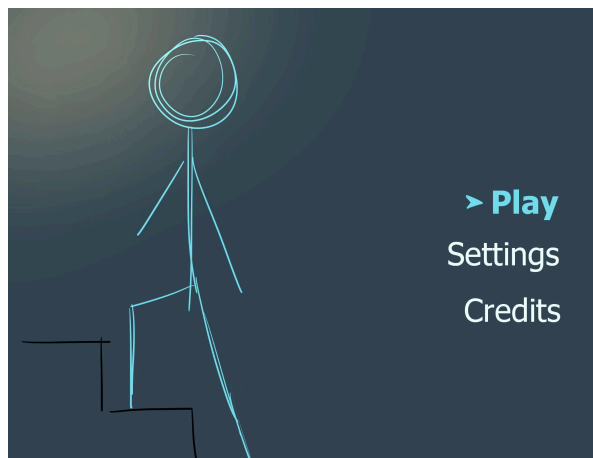
## Scenes:

The major game states include menus, game progression, gameplay mechanics, and game levels. Below are the sketches for each section.

## Menus:

The start screen will have menus for play, settings, and credits. The settings will include audio volume control, resolution, and key bindings feature.

*Start screen:*



*Setting screen:*

### Settings

Volume

Resolution

### Key Bindings

Move up	w
Move down	d
Move left	a
Move right	s

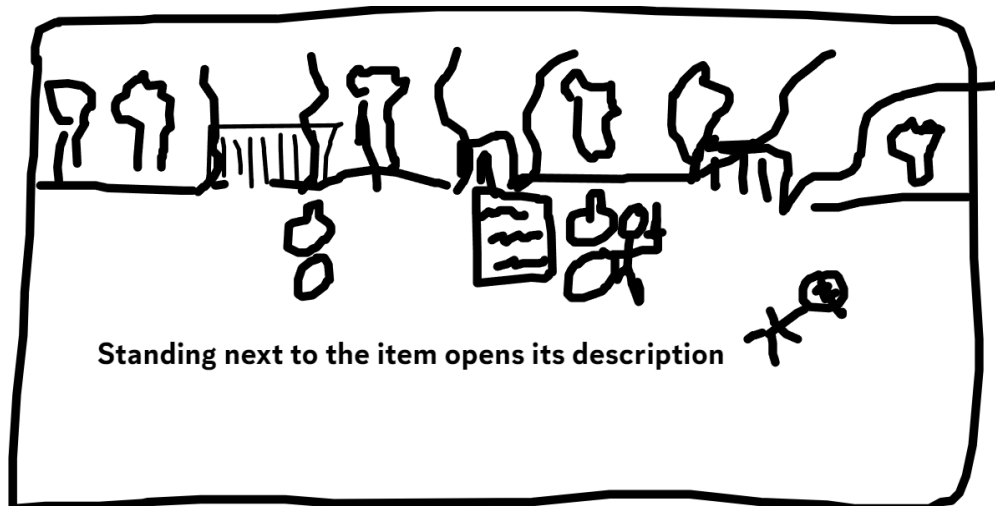
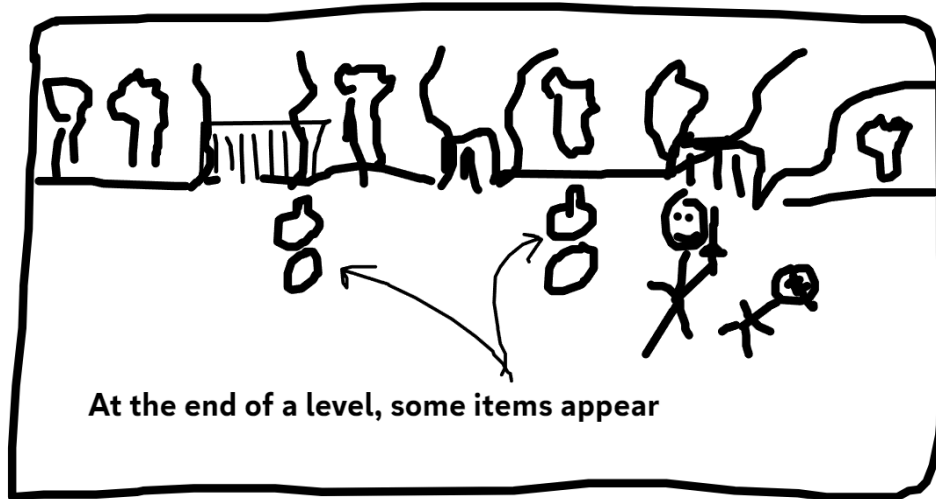
*Credit screen:*

Credits
Role
Names here
Role
Names here
Sources
Names of sources here

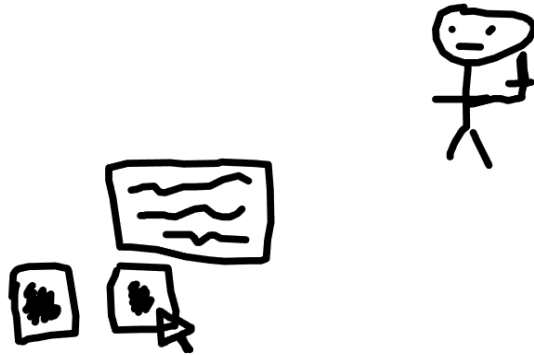
## Progression:

Game progression includes how the player acquires items and how upgrades function. The currency for upgrades will be souls, which the player earns for each kill.

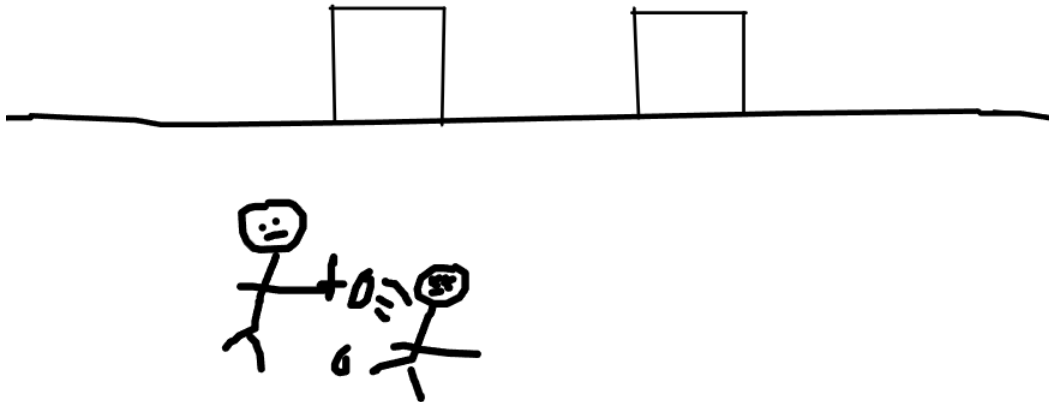
### *Items:*



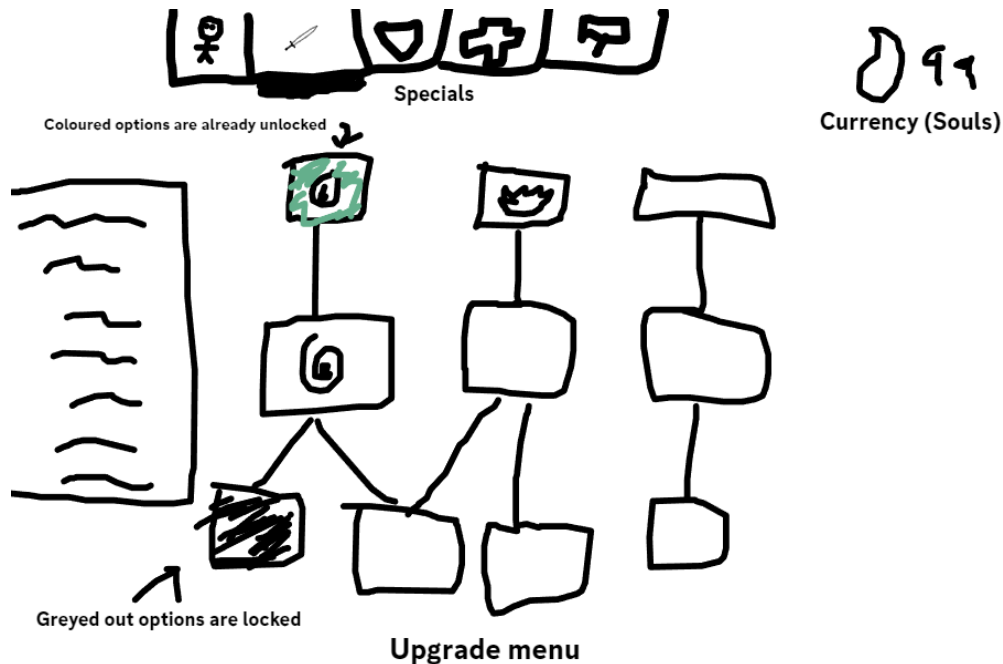
Hovering over an item in the inventory should display the description of the item  
There is no inventory cap



*Upgrades:*



Killing enemies grants souls



The menu is accessed from the lobby

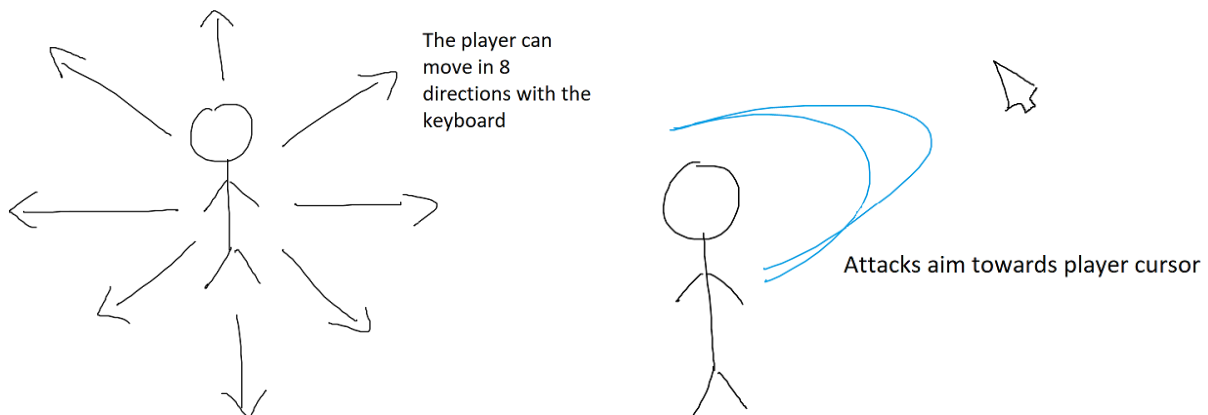
## Gameplay - Player Mechanics:

The player mechanics include the basic and advanced mechanisms.

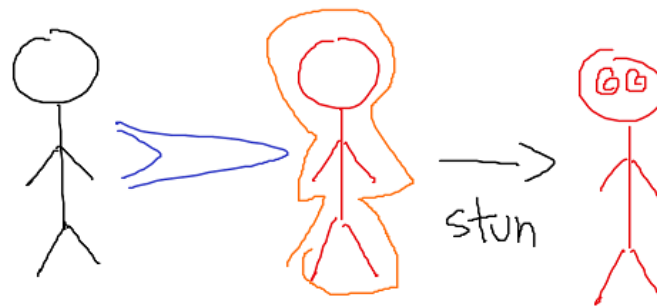
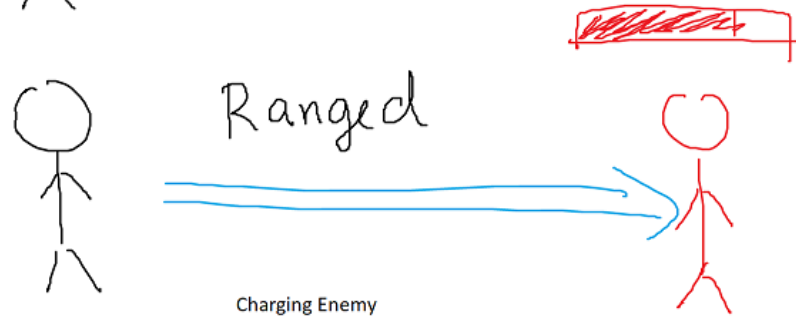
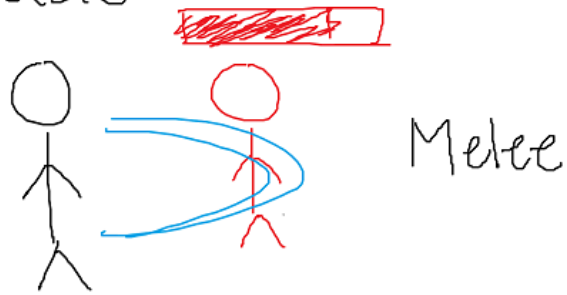
For basic player mechanics, the player can perform one form of melee attack, a ranged attack and a parry (parry comes with a cooldown). The player can also dash to dodge.

For advanced player mechanics, our priority is to implement the four attack-string for melee attack. Then, we will implement additional special abilities: heal (heals lost health), stun (stuns enemies), and shield (blocks damage).

### *Basic Player Mechanics:*



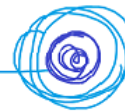
Basic



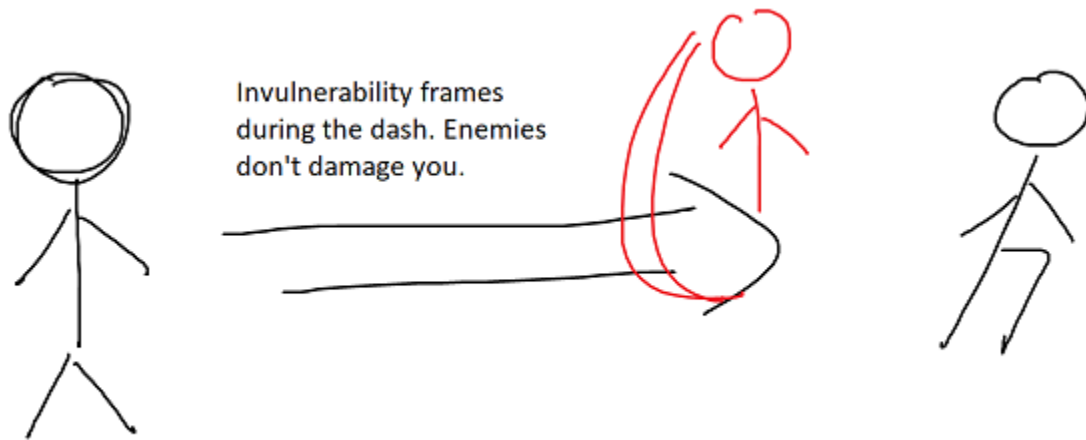
Parrying a charging enemy interrupts and stuns them.  
Parrying an attack blocks the attack and stuns the enemy



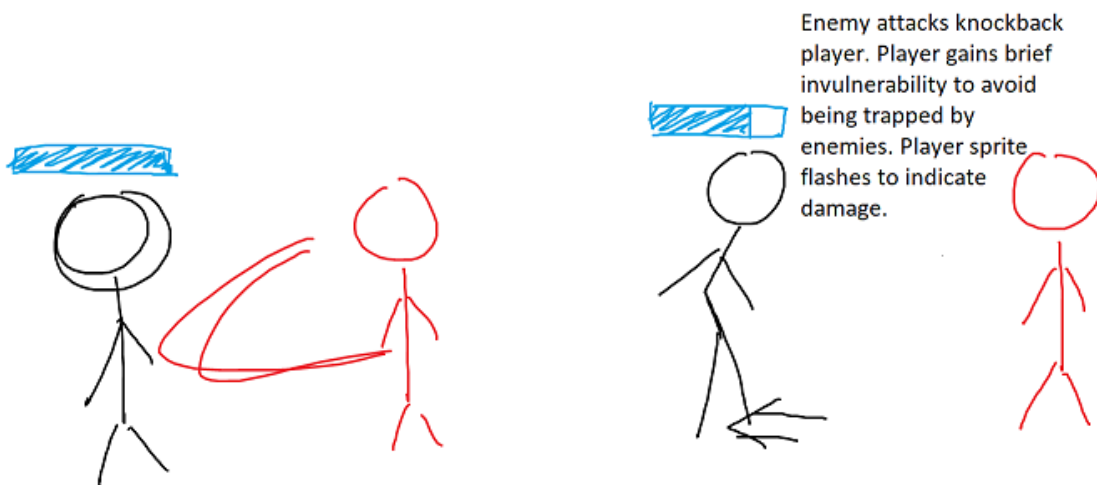
Ranged Attack - Magic Orb/Missile. Charging it makes it bigger and faster



### *Dash (Basic):*



### *Getting Hit by Enemies (Basic):*

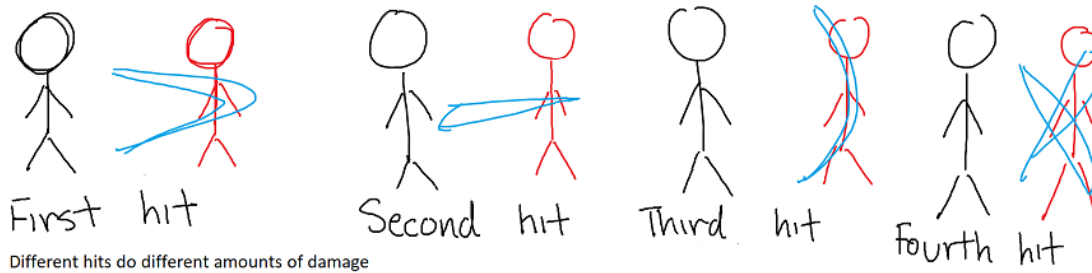


### *Player Death (Basic):*



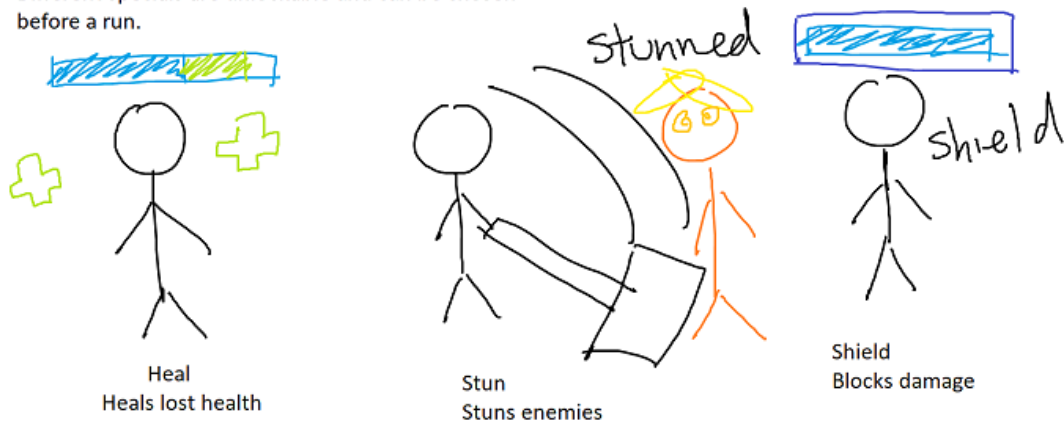
*Advanced Mechanics (melee attack):*

## Advanced Combat System (Combo)



*Specials (advanced):*

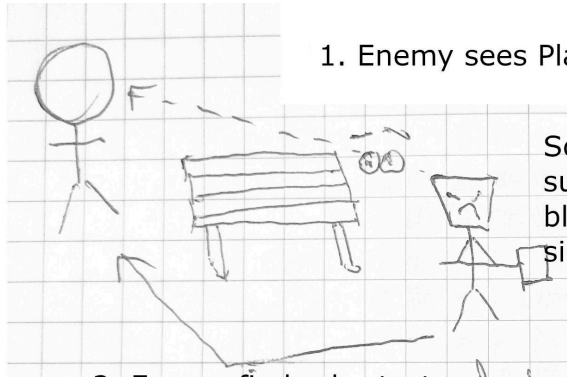
Different specials are unlockable and can be chosen before a run.





## Gameplay - Enemy Mechanics/AI:

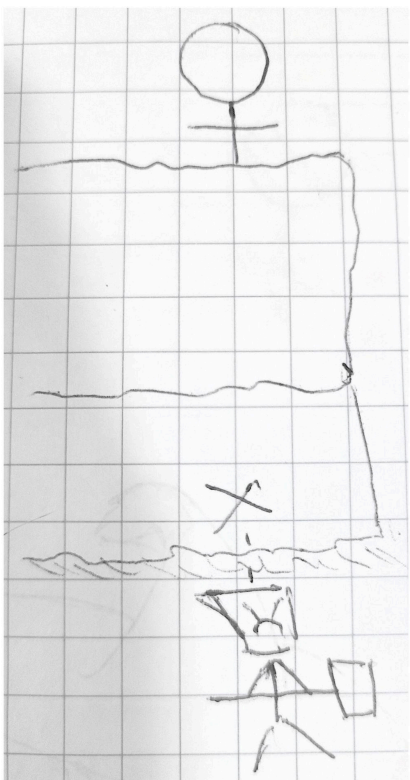
Enemy mechanics include chasing the player, performing melee and ranged attacks, charging, and executing boss-specific attacks.



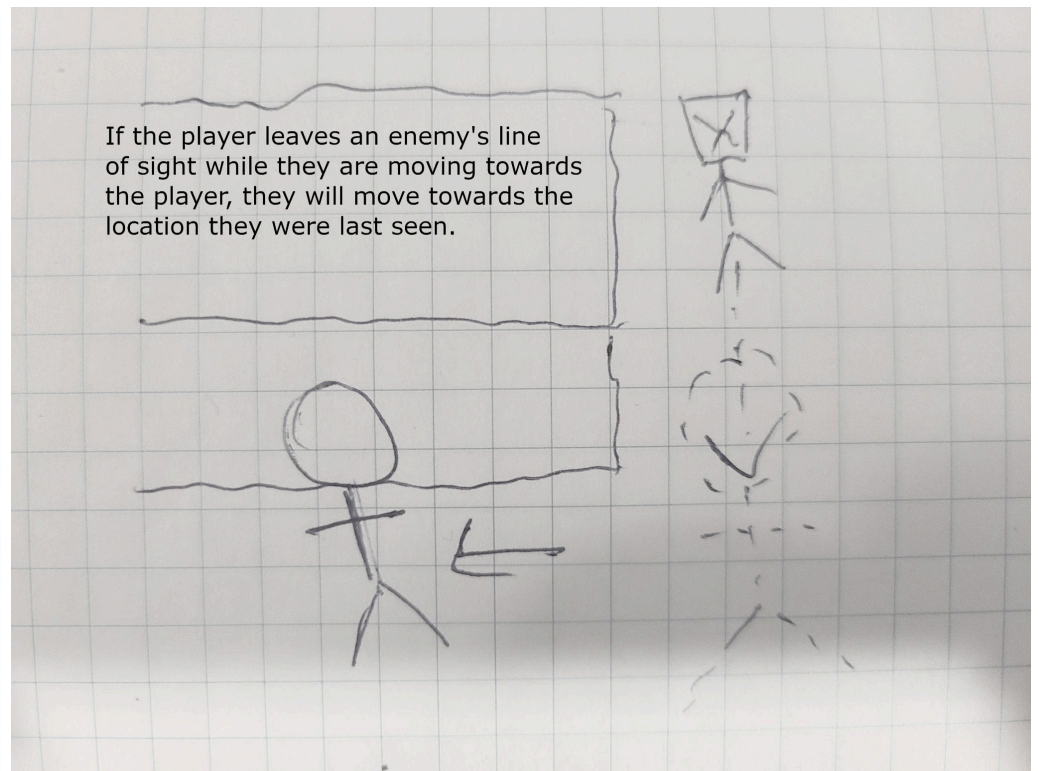
1. Enemy sees Player

Some objects on the map such as rubble or furniture blocks movement but not sight

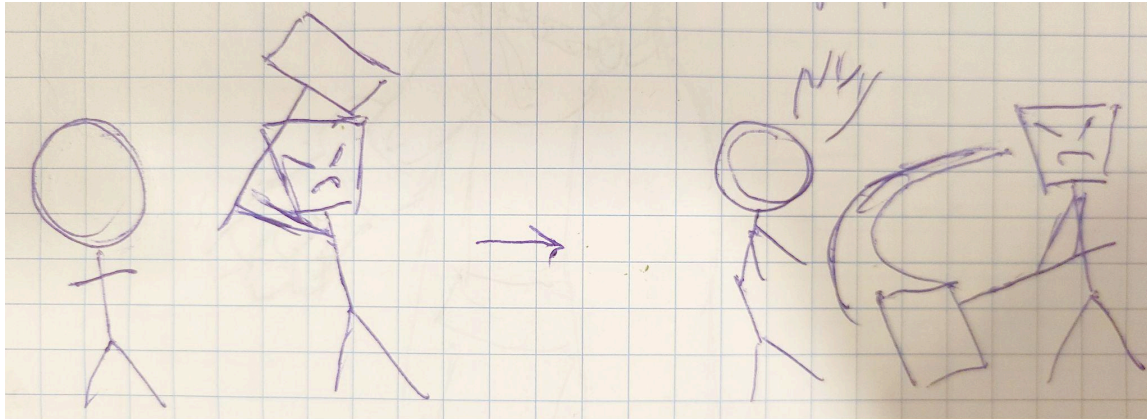
2. Enemy finds shortest path to location within attack range of the player and begins moving



Walls block vision.

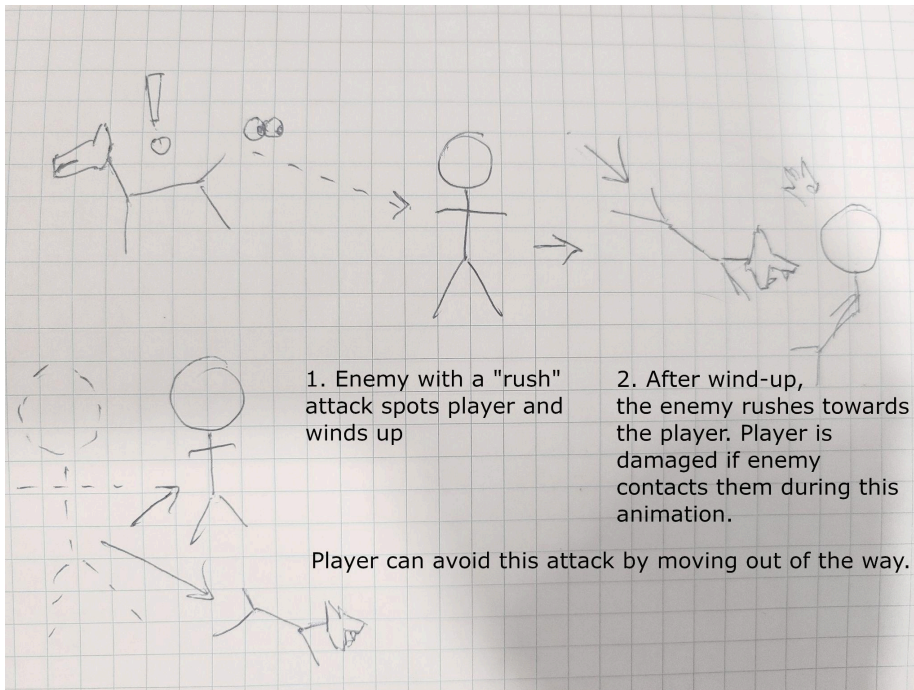


If the player leaves an enemy's line of sight while they are moving towards the player, they will move towards the location they were last seen.



1. Enemy is within attack range of the player and does a wind-up animation. It can be parried during this time.

2. After wind-up, the attack animation plays. If player is hit, they take damage and are knocked back.

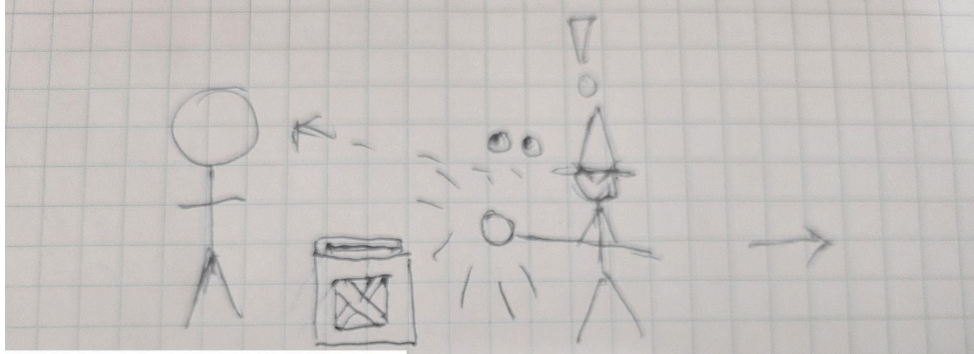


1. Enemy with a "rush" attack spots player and winds up

2. After wind-up, the enemy rushes towards the player. Player is damaged if enemy contacts them during this animation.

Player can avoid this attack by moving out of the way.

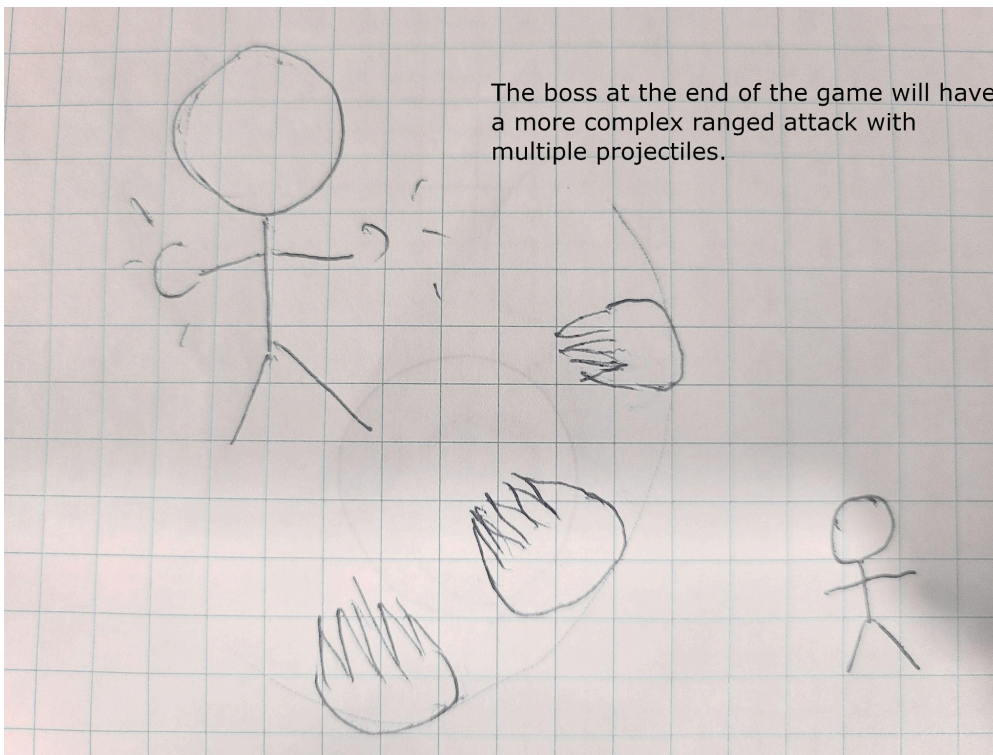
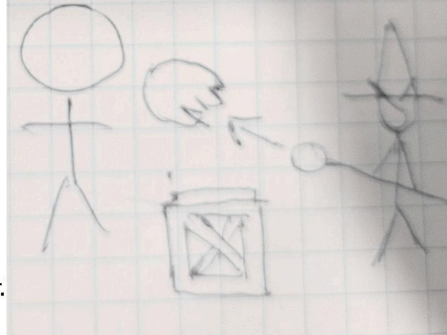




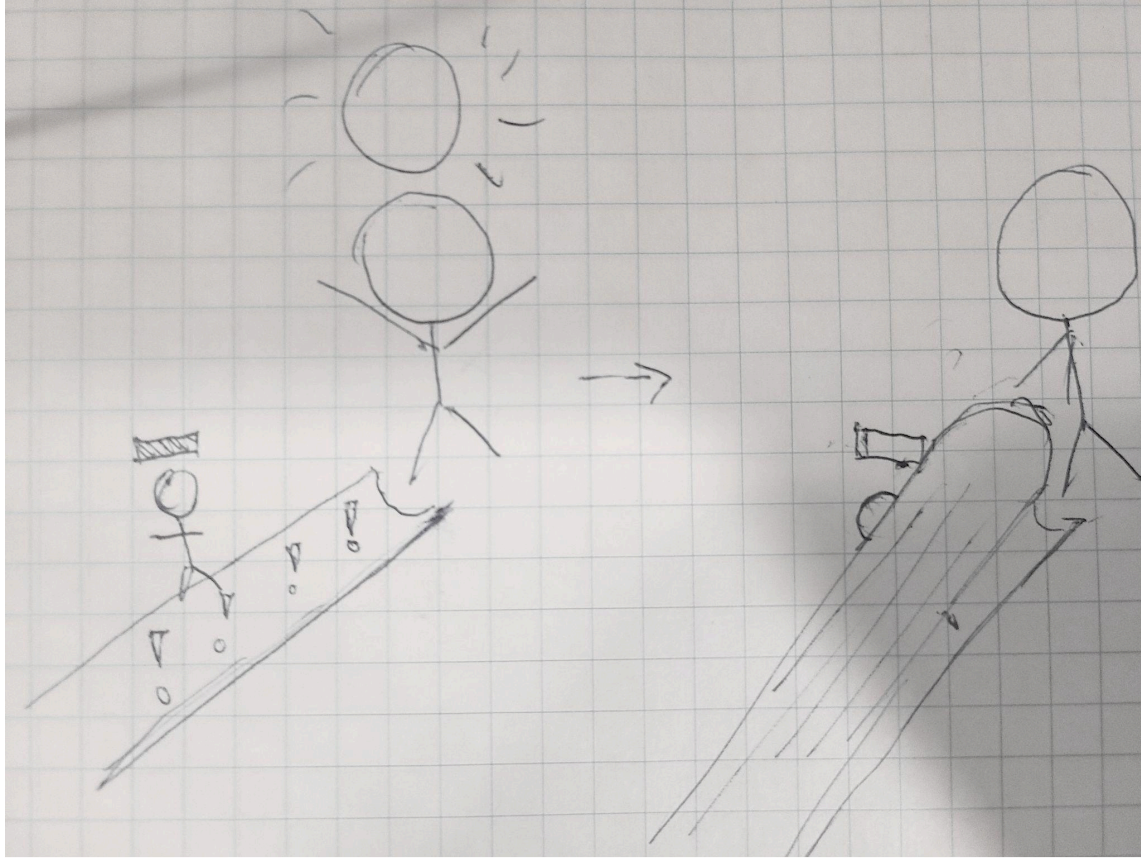
1. Enemy with ranged attack spots the player and begins a wind-up animation

If an object can be seen over, it can be shot over.

2. Wind-up finishes and a projectile goes towards player.



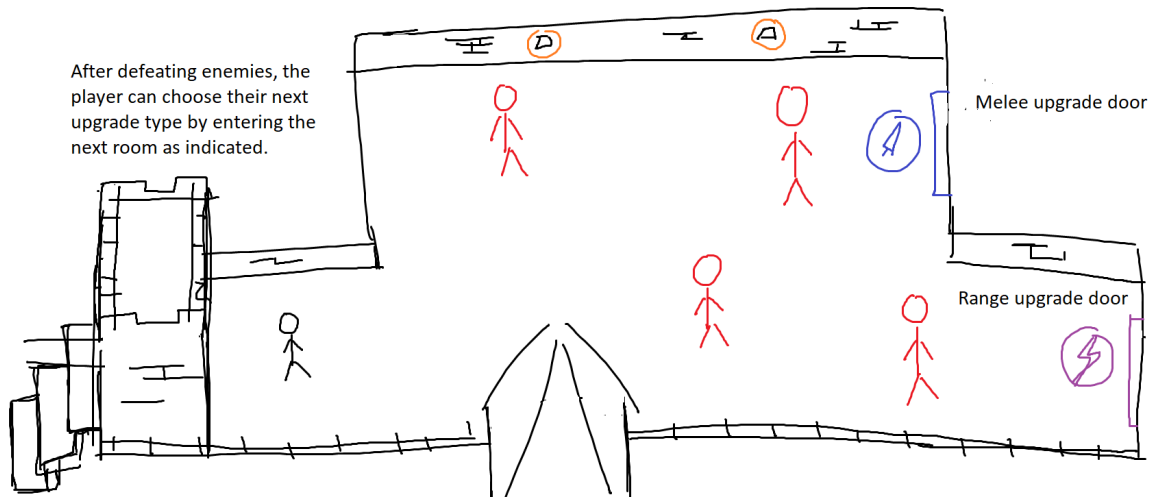
The boss at the end of the game will have a more complex ranged attack with multiple projectiles.



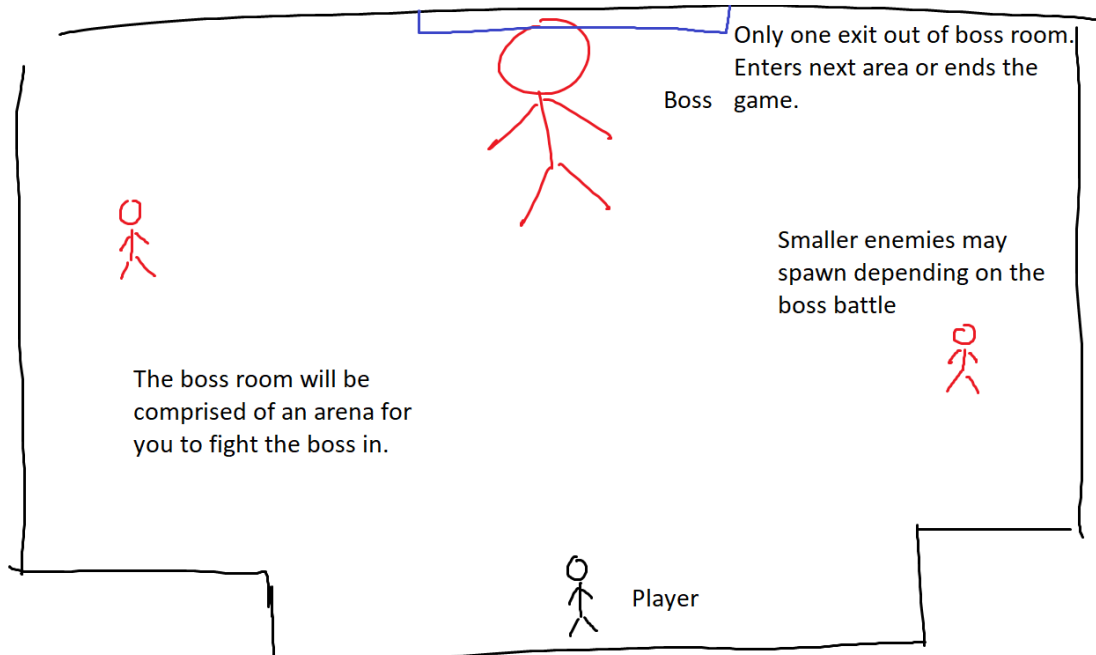
If time permits, the boss will also be given an attack that will deal damage to the player if they stand within an area that is indicated on the ground after the attack finishes charging up.

## Story - Levels:

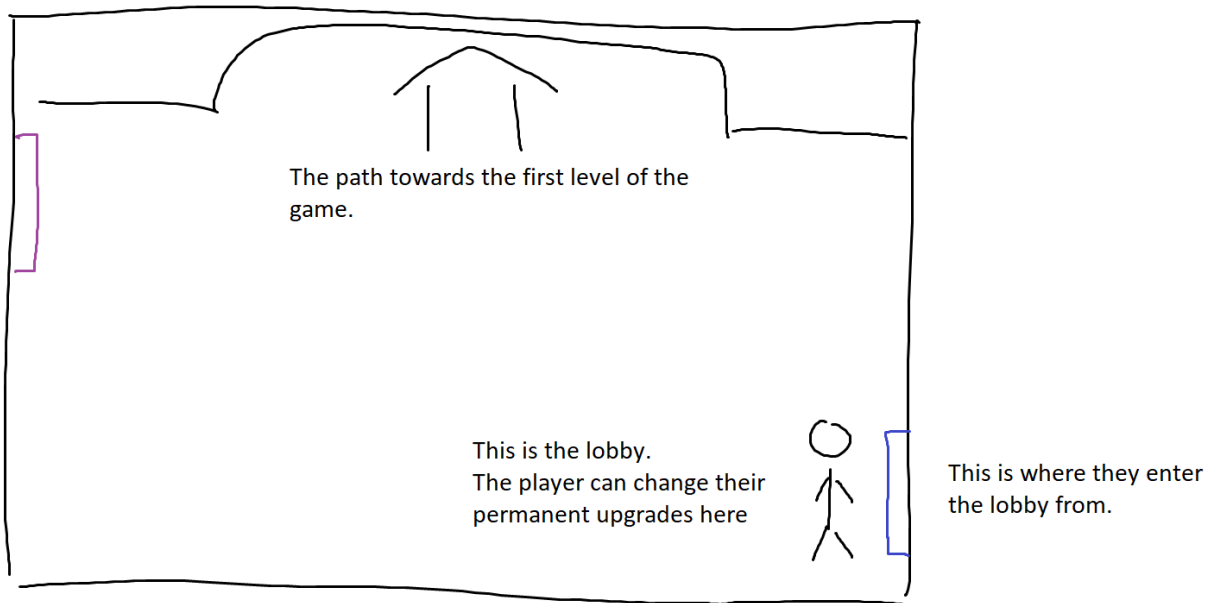
### Basic Level



### Boss Room



### *Lobby (Starting Area)*



## Technical Elements:

For **rendering**, we will have simple rendering effects using fragment shaders and OpenGL uniforms to enhance the visual experience.

For **assets**, we will have sprites for characters, enemies, and terrain. Player animations include walking, idle, dashing, knock backs, death, parry, melee and ranged attacks. Enemy/boss animations include walking/chasing, attacks (charge, melee, ranged, stun, and being attacked). For sound assets, we will include sound effects for attacks, special abilities, and action like opening the doors. We will also include ambient sound, and background music.

For **2D geometry manipulation**, the player sprite will be translated to move around levels and involve dashing mechanics. We will have collision detection between sprites, terrain, objects, and enemies.

For **gameplay logic**, the player will be able to move in 8 directions and dash to dodge attacks. The enemies will chase the character and engage in attacks. The players must defeat enemies, collecting souls for each kill. Upon completing each level, the player collects items, then chooses which abilities to upgrade/unlock. The player wins the game by defeating the boss at the final level.

**Enemy AI** will choose the shortest path to the player, as long as the enemy has line of sight. Movement, but not vision, can be obstructed by rubble or furniture. Both vision and movement will be obstructed by large terrain such as walls. Until the player is spotted, the enemy AI will randomly roam, and start attacking with melee and ranged attacks when the player is spotted.

Basic **physics** will be included in effects on hit. This will be in the form of knockback from a player being hit by an enemy, and vice versa. Enemies will likely experience less knockback than the player.

We will implement a save for the souls and upgrades gained between runs. (**re-loadability**).

For **UI and IO**, the camera will follow the player's movements, and there will be audio feedback for actions like attacking and taking damage.

## Advanced Technical Elements:

For advanced **graphics** and **rendering**, we will implement particle systems for attack effects, boss effects, item interactions, and 2.5D lighting effects to enhance the visuals.

For the melee attack effects, the swing trails follow the weapon's movement and the spark effects to indicate an enemy has been hit. For ranged attack effects, charge effects, projectile trails, and spark/explosion effects will be included. We will include projectile trails and explosion effects for boss attacks, and have a dramatic particle effect (explosion) to signify the final boss death.



The items will have glowing particle auras around them and collecting them will have particle trails leading to the player. Collecting the souls will have shadow particles trailing towards the player.

We will also include a 2.5D lighting effect to show shadows for the moving and static objects to create depth and dimensionality.

Advanced **gameplay** features will include more specials, melee combo attacks (as seen above), and more bosses with complex behaviour.

More specials will give the player more choices on how they wish to play the game. Fewer specials will limit the player's choices in how they approach the game.

Melee combo attacks will be done in a chain of four melee attacks, each doing a different amount of damage with different speeds which enables a way for more skilled players to get rewarded for understanding the game. A more simplistic game will be the result of skipping this feature.

Complex behaviour for bosses includes an area of effect indicator, and would have AI that works with this. Omitting this will lead to less complex boss behaviour.

In terms of advanced story elements, item descriptions will provide lore. If not implemented, the worldbuilding of the game will be reduced.

## Devices:

For game control, the mouse and keyboard will be mapped as follows:

WASD - Movement

LMB - Melee Attack

RMB - Ranged Attack

Mouse - Aiming

Space - Dash

Q/E - Parry

F - Select Item

Escape - Pause Menu

## Tools:

This isn't an exhaustive tool list as there may be tools that we will add in the future or tools that will not be used.

**Audacity** will be used to edit music and sound effects.



**Clip Studio Paint** will be used to create any 2D illustrations needed for the game that are not pixel art. This will most likely be used to create the splash screen.

**Tiled** will be used to create custom maps that can be exported into a JSON file.

**Tilesetter** will be used to create tilesets that will be used in conjunction with Tiled.

**Krita/Aseprite** will be used to create custom sprites and pixel art for assets we cannot find online.

**EnTT** will be used for our ECS system.

## Team management:

We will use a kanban board on Github to keep track of our tasks and goals. Tasks will be assigned at each sprint planning. Sprint planning happens once a week. Everytime we meet up for any reason outside of sprint planning, we will do a check up to see everyone's status and if they need assistance. If a person has finished their assignment, they should pick up other tasks based on priority in the backlog.

The assigned tasks will generally align with our roles, but may branch out a little. For example, someone working mainly on assets may be assigned tasks for OpenGL and level designing, as well as miscellaneous tasks like bugs.

We have reassigned our roles as follows:

Benson: Gameplay + Level design

Cecilia: Game Assets (2D artist) + Game Design

Elton: Game Engine + AI

Emma: Game Engine + QA + Rendering

Stephane: Game Assets (Sound) + Gameplay

## Development Plan:

We will attempt to complete each task on the week that it is placed in. If it proves to be unfeasible, milestone 3 has a lot of advanced mechanics that we can drop, so we can push less important tasks back. We will also be working on future tasks in advance if we have extra hands.

### Milestone 1: Skeletal Game (Feb 9)

Week 1: Jan 27th - Feb 2nd

- Basic character movement (wasd)
- Basic collision detection (player with walls/objects)
- Minimal assets for 1 level (parsing and formatting)
- Basic rendering

Week 2: Feb 3rd - Feb 9th

- Add one type of enemy (not attacking) + minimal enemy AI
- Testing map environment
- Player Melee attack + Collision with enemies
- Connecting rendering system with parsed tile assets

This milestone will be setting up fundamental systems and describing templates for implementing things in the future.

If we cannot finish everything in time, we will push back the minimal enemy AI as well as parsing the assets and connecting tile assets to the rendering system as they are not as important to the base game.

## Milestone 2: Minimal Playability (Mar 2)

Week 1: Feb 10th - Feb 16th

- Sprites for key elements (player, enemies, common objects)
- Enemy melee attack logic
- Player ranged attack
- Dashing mechanics

Week 2: Feb 17th - Feb 23rd

- Player parry
- Basic Item system - Stat changes
- Simple enemy pathfinding
- Start and menu screen
- Sprites for dashing, ranged attack

Week 3: Feb 24th - Mar 2nd

- Sprites for boss
- Boss room
- More levels
- Test the completed level
- Background music / Minimal sound effects

This milestone will focus on adding the main mechanics of the game and should allow a playtester to get a feeling of the core gameplay. We will focus on getting the enemy AI, item system, dash, and parry (first special) completed.

The ranged attack can be pushed back to milestone 3 if it is too much of a challenge, and extra levels are dependent on the amount of time remaining. Extra sprites, the boss room, and sounds are also not needed and can be pushed into milestone 3.

## Milestone 3: Playability (Mar 23)

Week 1: Mar 3rd - Mar 9th

- Add more levels (Ongoing task)

- Game tutorial (showing controls)
- Enemy ranged attack
- Advanced enemy AI (will go to player's last seen position)
- Boss attacks and AI
- Sprites for items

Week 2: Mar 10th - Mar 16th

- Upgrade system
- Reloading from a save
- Melee combo (advanced feature)
- Advanced Items - Items that augment attacks (DoTs)
- Particle systems (advanced feature)
- Sound effects for all player interactions

Week 3: Mar 16th - Mar 23rd

- User testing
- Finalize art & music
- Final touches to fix any bugs

This milestone focuses on creating game mechanics that improve gameplay and having more dynamic combat. It will also finish peripheral parts of the game, not part of the main mechanics such as music and art. There are a few advanced features here that can be dropped with minimal effect on the final game if time is short.

## Milestone 4: Final Game (Apr 6)

Week 1: Mar 24th - Mar 30th

- Item flavor text
- Additional specials (advanced feature)
- 2.5D lighting (advanced feature) (not strictly required, "nice-to-have")
- More complex boss behavior / multiple bosses (advanced feature)

Week 2: Mar 31st - Apr 6th

- Background music and ambient sound effects
- Add prologue screen before the first level
- Add epilogue screen after defeating the boss

This milestone focuses on polish and refinement. We intend to use this time to handle any tasks from previous milestones that had to be delayed, or to flesh out game content if time permits. Features such as complex boss behavior or more content like flavour text and more specials will be cut if development takes too long. Since these features are not a part of the core game, we do not have a plan B to replace them.