Game Proposal: Touhou: Twilight Dungeons

CPSC 427 – Video Game Programming

Team: Project Zero (Team 1)

James Hoang (96347992)

Shaolun Huang (91732875)

Kevin Huang (34934810)

William Qiu (28966216)

Sam Sun (25137324)

Lin Su (54329743)

Story/Gameplay:

Briefly describe the overall game structure with a possible background story or motivation. Focus on the gameplay elements of the game over the background story.

<u>Background story:</u>

This game takes place in Gensokyo, a place secluded from Japan by a magical barrier and is home to various life forms such as fairies, and gods. One day when Reimu, a shrine maiden, continues her routine to fetch water at a nearby well at her shrine when she discovers that after reeling up the bucket, it wasn't full of water as usual! Many thoughts came rushing into her mind, what could be the cause of this? To find the cause of this situation she decided to slip down into the well. Then she saw a faint light wiggling at the bottom of the tunnel, it seemed there was a bigger area for her to explore. She found a dungeon! With this great discovery, her main motivation for water shifted. Her eyes sparkled, who put this dungeon here?

(Note: In Gensokyo, all conflicts are resolved by Danmaku, which means "barrage" and another name for bullet hell, thus, this is why everyone can shoot bullets)

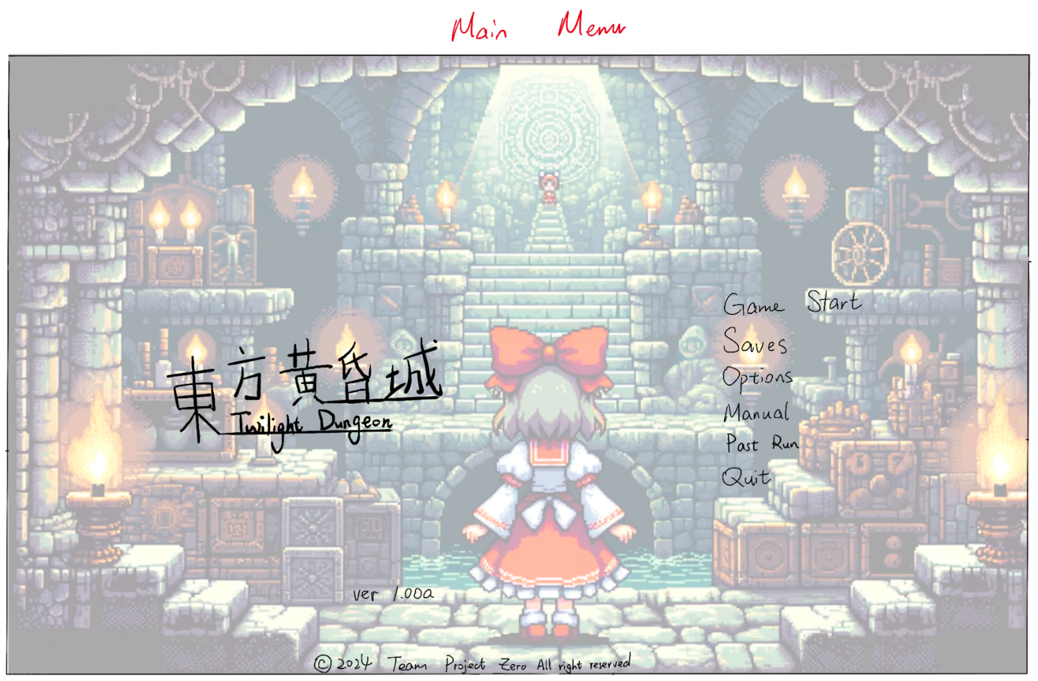
<u>Overall Gameplay Structure/Elements:</u>

1. The gameplay progression goes as follows:
   The player spawns on the first floor in an empty room with doors on each adjacent room.
2. The player moves to a door which brings them to another room starting at a door.
3. All doors become closed if there are any enemies in the room. The player must defeat all enemies for doors to become unlocked and for the player to continue.
4. Clearing rooms while not getting hit will add to the players combo multiplier. The multiplier will directly increase game speed while increasing enemy drop amount for money.
5. The player will start with a basic projectile attack they can fire to deal dmg and kill enemies. Items and buffs will alter the player's base attack. There are no other attack options.
6. Items can either be stat buffs to attack, hp, speed, etc or transformative items such as bullets that ignite enemies doing damage over time.
7. Players can press [Shift] to enter focus mode where for the cost of combo multiplier/second, the player's hitbox will become visible, drastically smaller, and the sprite will become more transparent. The player movement will become slower to allow precise movement.
8. There will be special rooms that can spawn next to any room such as shops or item rooms.
9. Shops will allow players to spend money dropped by enemies to buy permanent upgrades in the form of items, health to heal, keys to open treasure chests, or temporary buffs that last 1 floor.
10. There is always a boss room on each floor containing a strong unique enemy who must be defeated to progress to the next floor.
11. Each preceding floor will get more difficult with harder enemies. There are no current plans to alter the theme of the floors drastically other than different enemy types.
12. After clearing 5 floors and defeating the dungeon master on the 5th floor, the game is won. The run is saved and the player can reset for a new run. If the player dies, their entire progress gets reset and they need to start from floor 1 with no items.

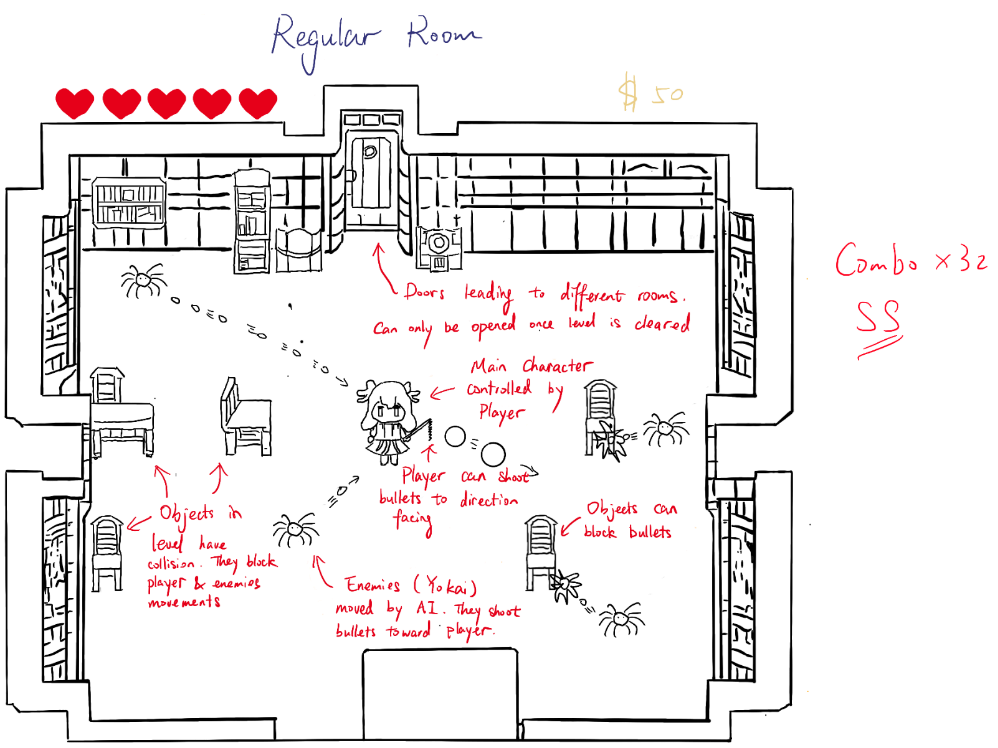
Scenes:

Produce basic, yet descriptive, sketches of the major game states (screens or scenes). These should be consistent with the game design elements, and help you assess the amount of work to be done. These should clearly show how players will interact with the game and what the outcomes of their interactions will be. For example, jumping onto platforms, shooting projectiles, enemy pathfinding, or 'seeing' the player. This section is meant to demonstrate how the game will play and feed into the technical and advanced technical element sections below. If taking inspiration from other games, you can include annotated screenshots that capture the gameplay elements you are planning to copy.

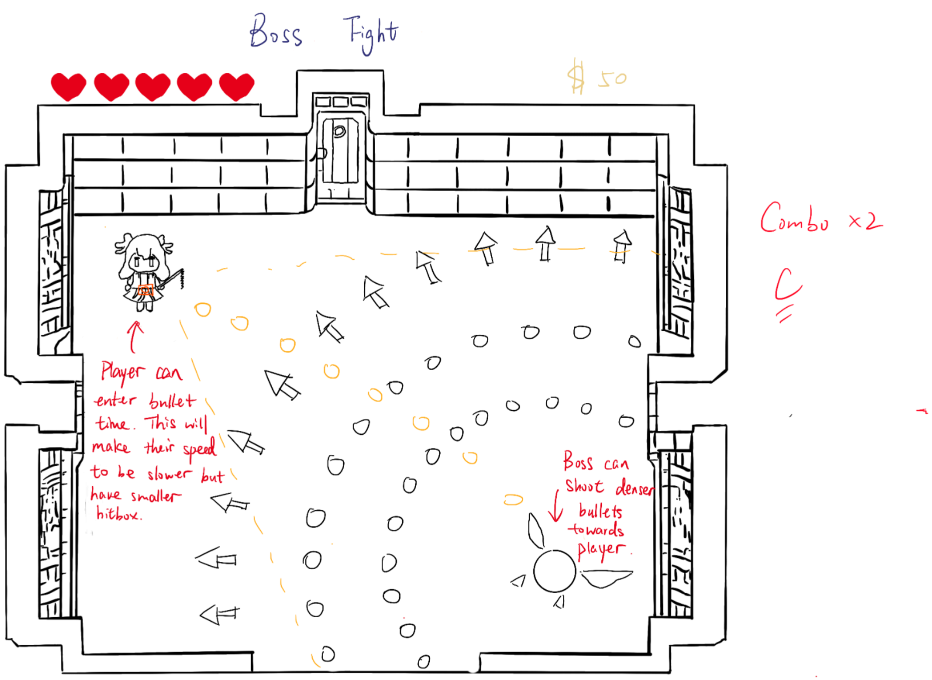
Main title screen: Game start and Saves -> Start the game; Options -> Game/Audio/Key settings; Manual -> Display game background/how to play, Past Run -> Display previous completed dungeons, Quit -> quit game.



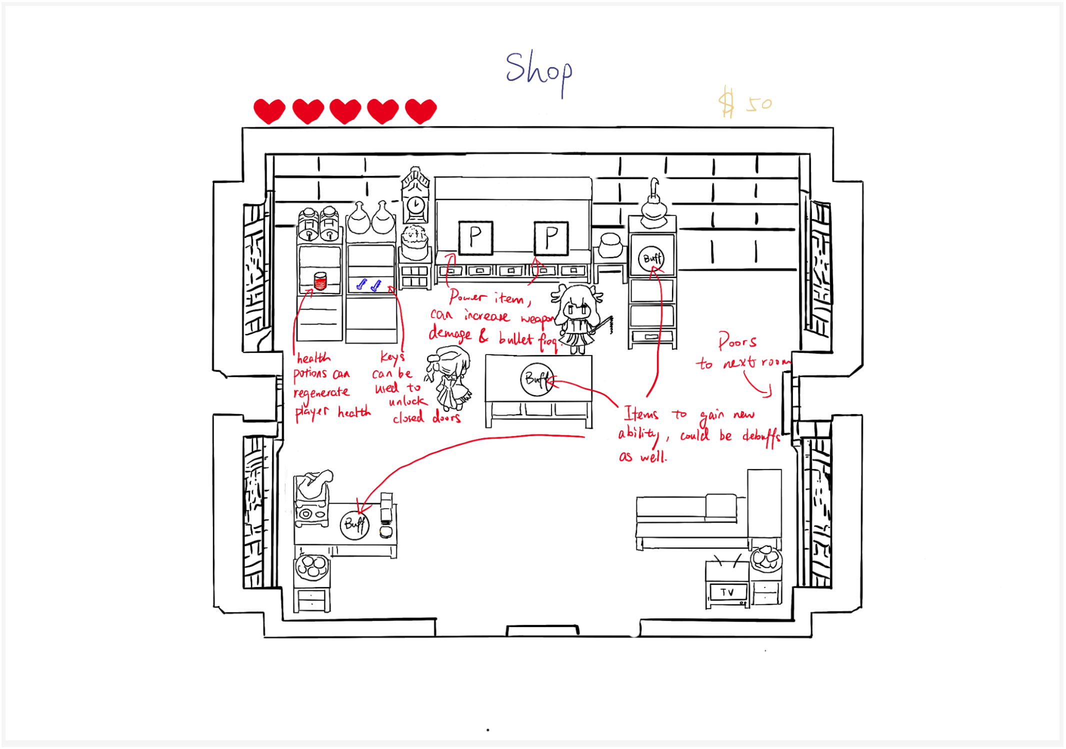
Standard Game Play: Left top corner displays health and Right top corner displays money held. The combo meter is shown directly on the right. There are objects in the scene to block any movements (e.g. player, enemies, and projectiles. Doors will be locked until every enemy in the room is cleared. Players can shoot in the direction they are facing, which will be controlled through mouse and keyboards.


Regular Room
$ 50
Combo ×32
SS
Doors leading to different rooms. Can only be opened once level is cleared
Main Character controlled by Player
Player can shoot bullets to direction facing
Objects can block bullets
Objects in level have collision. They block player & enemies movements
Enemies (Yokai) moved by AI. They shoot bullets toward player.

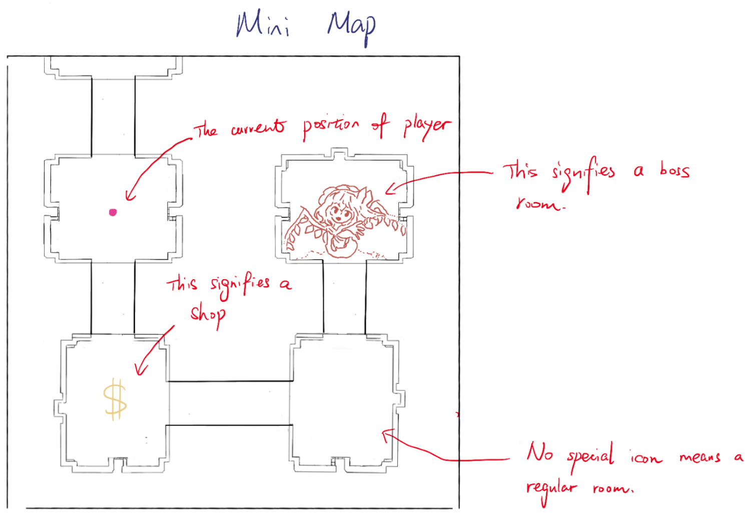
Boss fight:


Boss Fight
$ 50
Combo ×2
C
Player can enter bullet time. This will make their speed to be slower but have smaller hitbox.
Boss can shoot denser bullets towards player.

Shop: Players can buy health potions in the shop. The health of the player is clamped by the maximum hp of the player. The key is used to unlock any rooms that have closed doors. The player can also choose to buy buffs and debuffs that can be applied in game, as well as upgrade their weapon.


Shop
$ 50
P
P
Buff
Power item, can increase weapon demage & bullet freq.
health potions can regenerate player health
Keys can be used to unlock closed doors
Buff
Doors to next room
Items to gain new ability, could be debuffs as well.
Buff
TV

Minimap: Minimap is displayed on the top right corner of the scene. Player's location will be marked by a red dot. Rooms are connected through corridors and doors.


Mini Map
The current position of player
This signifies a boss room.
This signifies a shop
$
No special icon means a regular room.

Technical Elements:

Identify how the game satisfies the core technical requirements: rendering; geometric/sprite/other assets; 2D geometry manipulation (transformation, collisions, etc.); gameplay logic/AI, and physics.

- Rendering
    - Uncapped FPS
    - Rendering the sprites, walls, tiles, from png source image.
    - Player will have 8 different sprites based on directions NESW and diagonal direction -> will face in the direction of the mouse cursor
    - The player has an idle animation where the sprite is moving a tiny bit like bobbing up/down
    - Design decision to render everything at the beginning to prevent lag when transitioning to another room
- Geometric/sprite/other assets
    - We will also utilize AI-generated art if possible (Menu screen splash screen)
    - Static simple 2D sprite sheet - For player, enemies, items, bullets
    - Tiles - Rock, dirt, water, wood, grass floor tiles
    - Audio - Touhou style music, firing/hit sound effect
- 2D geometry manipulation (transformation, collisions, etc.)
    - Transformations
        - Translations - enemy/player/projectile movement - mostly used to move objects around the room
        - Bullet/Enemy entity size scaling
    - Collisions - Will be mostly box to box collisions, circle to circle collisions for bullets
        - Player/enemies colliding with a wall cannot move off the screen
        - Bullet objects colliding with the wall are deleted
        - Player colliding with enemies, the player will lose a heart
        - Enemy bullets colliding with other enemies have no effect
        - Player/enemy bullet collisions deal damage and are deleted afterwards
        - Door collisions - if no enemies, Player can pass through it to the next room. Enemies cannot collide with doors
        - Player collides with item, show a prompt to press a button to pick up
- Gameplay logic/AI
    - Enemies AI - Pathfinding behavior
        - Given the map, player, and enemy position, all the algorithm does is return a direction the enemy should move. They could also move in a straight line.
        - Some enemies/bosses shoot bullets straight at Player. Bosses also have a “bullet pattern” (e.g. shooting bullets in random directions) rather than direct shots - does not need AI.
        - Use A* Pathfinding algorithm or another algorithm for enemies to chase down the Player in case of an obstacle
    - Enemies spawn locations - Pseudo-Random (picking from a set of locations)
        - Every room type has a spawn locations (e.g. 5 enemies can spawn in the middle, or 4 enemies can spawn in each of the corners of a room)
        - Enemies should already be generated at the beginning (can’t spawn when the player is in room). They should also only spawn once (not spawning repeatedly like in Minecraft).
    - Currency system - Enemies/bosses drop money of which the Player can collect and spend them on items in the shop.
- Physics

- On-hit knockback for enemies based on their mass attribute, heavier enemies will have little impact whereas lighter enemies will be pushed further
- Bullets will be spawned in the center of the Player/enemy model with an amount of velocity

Advanced Technical Elements:

List the more advanced and additional technical elements you intend to include in the game prioritized on the likelihood of inclusion. Describe the impact on the gameplay in the event of skipping each of the features and propose an alternative.

Prioritized by likelihood of inclusion to not as likely:

- Enemies drop money (common), keys to open chests/special rooms (rare)
  - Removal of this feature means the Player cannot buy anything in the shop and renders the shop useless. This also means they can't get stronger, and so enemies will slowly power creep the player.
  - An alternative would be that the shops and chests can be opened for free
- Combo meter - the larger the meter, the faster the game (e.g. everything is faster such as bullets, enemies, and players), increase meter by defeating enemies.
  - Core difficulty scaling which rewards better players and allows for a more dynamic playthrough
  - Removal of this feature will then make difficulty more linear and require difficulty scaling to be stat increases for the enemies and increases to enemy amount.
- Invulnerability for a few seconds (~0.5-1 sec) when Player gets hit with a visual cue (flashing white)
  - Allows for Players to plan and rebound from a mistake. This prevents instant death from continuous knockback if Player touches multiple bullets at once.
  - Skipping this will make the game much harder. An alternative is to increase enemies' health points.
- Focus Mode: (previously called Bullet Time)
  - The player's hitbox will become a small circle and visible to the player. The smaller hitbox will only check against the bullet hitbox however the original hitbox will check against terrain and enemy collision. This is to prevent players entering focus mode, walking to the corner of the room, then exiting focus mode to get stuck in a wall.
  - Without focus mode, the gameplay will be more difficult due to not having the option to shrink the player's hitbox and less engaging combat.
  - The Base player hit box can be reduced if this feature is skipped
- Procedurally generated maps
  - Allows for unique runs though the game and lower repetition. Removal of the feature will force us to set prebuilt maps which can make each run feel less diverse.
  - An easier method would be to randomly choose from a set of pre built maps to have the diversity without implementing a complex procedural generation system
- Mini-Map
  - Helps players orient themselves in the labyrinth like dungeon.
  - Without the map players may spend time exploring areas they have already been in and so out of combat movement speed increase is an alternative
- Fog of War
  - Adding to the labyrinth exploration elements of the game, covering up non explored rooms with pitch black darkness creates a clear direction to inform the player where to go next
  - Without fog of war, the only way to tell if a room has been explored is if there are enemies in them which can be hard to tell quickly.

    - The alternative will be to simply have the players check for enemies manually.
- Status Effects
    - Conditions imposed on enemies that has different effects based on the type of status, ie fire and poison deals continuous damage over time
    - Sources of status effects will primarily be from a player item adding a status effect to their attack.
    - Without status effects, the item designs will become more limited and player combat can become less interesting.
- Advanced Item Types
    - Such as Attack altering Items or Cursed Items
    - These items will change the player's attack pattern in ways beyond simple status increases such as status effect items, items that make the player bullet grow/shrink over time, items that have the player shoot in random directions.
    - These items should make the player actively decide if they want to pick it up or not which adds to both gameplay variety but also player strategy.
    - Absence of advanced items will have all items simply be status increases.

Devices:

Explain which input devices you plan on supporting and how they map to in-game controls.

We plan on supporting a keyboard and mouse (or trackpad)

WASD - to move up, left, right, and down respectively, holding two keys allows the player to travel diagonally.

Mouse cursor - the direction at which Reimu will be facing or where she will shoot

Mouse1/space - navigate the menu, when in the game - press to fire one bullet in the direction of the cursor, and hold to continue firing

Shift - to trigger the Focus Mode mechanic.

E - Interact, most commonly used to pick up the item on the ground.

F - to read the description of the item.

ESC - to pause the game, or exit a menu screen

Tools:

Specify and motivate the libraries and tools that you plan on using except for C/C++ and OpenGL.

https://www.scenario.com/ for AI model

https://hotpot.ai/game-asset-generator Create backgrounds, sprites, gems, weapons, cards, characters, and other game assets with AI.

https://www.mapeditor.org/ tile editor for level design (maybe - depends on lecture)

More to come as we learn…

Team management:

Identify how you will assign and track tasks and describe the internal deadlines and policies you will use to meet the goals of each milestone.

We will be utilizing the GitHub Projects in the repository. This will allow us to track and assign tasks while being able to create tasks in the backlog, to-do, doing, and done. For every task, we will open an Issue and a new branch on github that addresses that task. This issue will then be put into the GitHub Project where we can easily track them.

We will also hold bi-weekly online scrum meetings on discord. In these meetings, we will share and discuss what we have done, problems we have faced, and additional help to resolve issues. This will put us on track to meet the goals of each milestone.

Below are the main tasks, and each team member can choose and assign the one that suits their interest, preference, or expertise best. This is so they can enjoy their work more.

Main tasks (coding):

- User Stories
- Create unit tests
- Set up OpenGL (render that red triangle)
- Art rendering (sprites, tiles)
- Game engine (game loop)
- Entity spawning system (method to spawn enemies, items, bullets)
- Tile collision system (player-enemy, player-bullet, enemy-bullet, player-walls, etc)
- Combo system (global multiplier value)
- EnemyEntity system (template for different enemy designs)
- Item system (inventory, items that affect player stats and abilities, enemies drop items from inventory)
- Movement (all entities)
- Player controls (player movement, player attack, player camera)
- Physics (enemy knockback)
- UX/UI (menu, pause screen, loading screen)
- Design map rooms (shop room, empty room, enemy room 1, boss room, etc)
- Map generation (manual map creator / procedural generator)
- Enemy ai (pathfinding, attack patterns)
- Get an image loading system and render PNG
- Reading animation sprite sheets and playing them on demand
- Sound system (music and sound effects)
- Console commands for debugging

Development Plan:

Provide a list of tasks that your team will work on for each of the weekly deadlines. Account for some testing time and potential delays, as well as describing alternative options (plan B). Include all the major features you plan on implementing (no code).

Plan B:

For **core features (Bolded)**: We will shift that task to the next week

For *non-core features (Italicised)*: We will scrap the feature.

Milestone 1: Skeletal Game

Due: Feb 16th

Week 1 (Week of Feb 5)

- **Set up OpenGL**
- **Game Engine (game loop)**
- **Art rendering (sprites, tiles)**
- **Entity Spawning System (Create entities and components, make ECS)**
- *Get an image loading system and render and move PNG*

Week 2 (Week of Feb 12)

- **Player Entity in Simple Room with single enemy**
- **Player controls (player movement, player attack, player camera)**
- **Camera controls (mouse influences camera) (basic #21)**
- **Audio feedback (basic #23)**
- **Simple Player-wall collision / player-enemy**
- *Combo system (global multiplier value)*

Milestone 2: Minimal Playability

Due: March 4th

Week 1 (Week of Feb 19 - Reading week)

- **Tile collision system (player-enemy, player-bullet, enemy-bullet, player-walls, etc)**
- **Movement (all entities)**
- *Enemy Entity system(template for different enemy designs)*
- *Physics(enemy knockback)*

Week 2 (Week of Feb 26)

- **Map generation basic(manual map creator)**
- **Enemy ai (pathfinding, attack patterns)**
- *Design map rooms(shop room, empty room, enemy room 1, boss room, etc)*

Milestone 3: Playability

Due: March 25th

Week 1 (Week of March 5 + March 12)

- **Reading animation sprite sheets and playing them on demand**
- **UX/UI (menu, pause screen, loading screen)**
- *Item system basic(inventory, money and keys ,enemies drop items from inventory)*

Week 2 (Week of March 19)

- *Map generation advanced(procedural map generation)*
- *Console commands for debugging*
- *Item system advanced (items that affect player stats)*

Milestone 4: Final Game

Due: April 8th

Week 1 (Week of March 26)

- *Special items (items affecting player abilities, debuff items, cursed items, etc)*
- *Special event rooms (rooms that heal, random chance to get an item or get hit, etc)*
- *Add additional items, enemy designs, boss designs*

Week 2 (Week of April 4)

- *Polish*
- *Edge case testing*
- *Optimization*