Milestone 3 - Reason for Discrepancy & Updated Proposal

Your submission should align with your proposed development plan: Provide a writeup explaining how your milestone aligns with the plan. Explain all discrepancies and submit an updated proposal when such discrepancies occur.

Although we were able to achieve most of our goals for Milestone 3, we had to make some compromises for certain features, precise collisions, advanced pathfinding, extra floors, bosses, dynamic shadows, working cutscenes, and user options. However, we put extra effort into having better sprites, sounds, and feedback, along with implementing damage text and a visual turn queue.

Our group members were especially busy during this time so we had to cut out a fair bit of our planned features

Previous Development plan:

Playability (March 18)

Week 1

- Add a more diverse set of enemies, rooms, actions and items
- Implement basic text pop-up functionality
- Finalize user core user actions and movement options
- Finalize collision physics (ensure everything works precisely)
- Finalize enemy pathfinding and decision making
- Add reloadability functionality

Week 2

- Add more rooms and floors (different theme for each floor)
- Add Final Boss (probably not exactly here but we'll see)
- Implement basic 2D dynamic shadows
- Create tutorial segment
    - Room with 1 door that spawns an enemy to block door when player approaches
    - Teach player how to use EP to move and attack the enemy to unblock the door
- Implement Archive
    - For Notes
    - For Artifacts & Items

Week 3

- Continue adding game content
    - such as items, artifacts, enemies, and rooms
- Create cutscene(s) to play at the beginning and end of the game
- Add parallax scrolling for the background textures
- Create user options

Game Proposal: Adrift In Somnium

CPSC 427 - Video Game Programming

Winter 21/22

Team Members:

Adam Wan - 32962458
Colin Wang - 58520636
Kaiti Mok Rong - 16915143
Jordan Dang - 22739866
Christopher Cliff - 76658699
Sasha Maximovitch - 44548667

Story:

Describe the overall game structure with a possible background story or motivation.

A prospective video game developer takes a course on video game development, and when assigned a large project to create a game from scratch, they immediately start brainstorming and drawing concept sketches in their class notes. After class ends, they work on the project in a nearby internet cafe. However, in their frenzy of motivation, they lose track of time and end up working themselves to exhaustion, passing out.

The protagonist then regains consciousness in a setting resembling the concept sketches they made prior, with no memories of the past. They choose to explore the mysterious labyrinth they ended up in, slowly realizing their purpose. After progressing through a certain number of areas, the protagonist will confront a being who challenges them by asking them what they desire.

What ultimately happens to the protagonist after this point will be based on what the player chooses to do at certain points in the game, meaning there will be multiple endings. The implementation of the alternative endings are optional, however, since the story is already complete with the main ending.

Normal Ending:
Obtained by defeating the Final Boss without answering with a special keyword. The protagonist wakes up from their coma, and although they've been taken out of the video game development course a month ago, they suddenly have an idea and they manage to create a game just in time before their final project is due. The professor is surprised by their sudden return, and praises them for being able to create something despite their unusual circumstances.

Secret Ending:
Obtained by defeating the Final Boss after answering their question with a special keyword, and then clearing the secret area and defeating the Secret Boss. The protagonist wakes up from their coma, and although they've been taken out of the video game development course a month ago, they suddenly have a burst of motivation, walking out of the hospital with great spirit. They work tirelessly for days on end, and create a masterpiece of a game just in time before their final project is due. The professor is absolutely stunned at the sheer magnificence of the game they made, and the protagonist goes on to win a special award.

Incomplete Ending:
Obtained by clearing many rooms in the first area without advancing further into the game, and entering a peculiar corridor in the wall that has light shining through. The protagonist wakes up from their coma, and upon the realization that they've been taken out of the video game development course a month ago, they ponder about their purpose, and what they want to achieve in life, before falling back asleep.

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, physics.

- Rendering:
    - 2D top-down rendering
    - Background elements are rendered using tilesets.
    - Areas of the map may not be visible to the player either due to a “fog of war” effect based on distance, or due to a wall blocking the line of sight.
- Assets:
    - Sprites for players, enemies and actions
    - Tilesets for background (may be third-party assets)
    - Sounds for actions, movement and ambiance
    - Music (third-party assets)
- 2D geometry manipulation:
    - Walls will obstruct movement and the player’s visible zone
    - Movement will be based on distance in pixels, as opposed to a grid system
    - Hit detection for attacks is weapon-dependent, some may use arcs, some may use a ray, etc.
- Gameplay logic:
    - Upon encountering a hostile enemy NPC, a turn order will be determined using the “speed” stat of all characters in the combat encounter
        - During combat encounters, a player will be able to use one of the following actions, or some combination thereof depending on game rules:
            - Move
            - Attack
            - Guard
            - Other action (e.g. use item)
        - See “Unique Mechanics” section for more details on combat
    - Upon meeting the requirements to “clear” a room, several doors will open on the edges of the room that allow the player to advance to the next room
    - Upon clearing a number of rooms, the player will then be able to advance to a boss room, where they can receive healing before and after the boss encounter
    - Clearing a boss encounter will allow the player to advance to the next floor, which will have slightly different aesthetics and encounters
    - Clearing a number of floors will allow the player to reach the final boss, after which the story ending will be determined based on certain player actions
- Unique mechanics:
    - In addition to the standard HP (hitpoints) and MP (mana points) available in most RPGs, Lost In Somnium will have EP (energy points) that will be crucial for surviving combat encounters
    - EP is expended by moving and performing actions such as attacking, and is recovered at the beginning of each player turn
    - Outside of combat, the player is able to move and perform other actions freely without EP expenditure

    - Players may also be allowed to “Prepare” actions to prevent their movement from using more EP than their prepared actions require
    - Players may also “Guard” at the beginning of their turn to immediately end their turn, reducing incoming damage and recovering more EP than usual on the next turn
- Physics:
    - The player and most enemies will not be able to pass through walls or each other due to collision
- Map Generation
    - Room layouts will be generated using pseudo randomization
    - When players enter a room there can be walls and other objects in the room
    - These objects aren’t placed randomly as we will make different room configurations and each room will pick a configuration to use when the map is generated
    - Not only that, but every object in the room has a random chance of appearing, so for instance there can be a room configuration that has two walls in specified places but when generating that room it may choose to place both, one of or none of the walls
    - This will be done for all objects in the room to thus making our approach pseudo randomized

Advanced Technical Elements:

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

- Collectible artifacts (may give player bonus stats or effects that change gameplay style)
    - Impact of skipping: Significantly reduces replayability and variation of playstyles. Skipping this feature would also negatively impact the story as artifacts can be used to hint at the game world and the ending.
    - Alternative: Artifacts implemented as simple collectible images, with no gameplay effects.
- Intro cutscene
    - Impact of skipping: Makes the premise of the game less impactful
    - Alternative: text-based intro
- Varying weapon/attack types (single target, radius, cones, chain attacks, etc.)
    - Impact of skipping: There will likely be less weapons overall, as new weapons would only impact the player’s stats
    - Alternatives: Player only has basic single target and radius-based attacks. (easiest to implement)
- Final boss mechanics
    - Final boss mechanics change based on the player’s answer to its question before the fight
        - text-entry will be used for the player to answer said question
    - Special area for secret boss

        - unlocked if the player answers with a specific keyword that the game will gradually hint at after beating the Final Boss once
    - Impact of Skipping: very little, since the story is already complete at this point
    - Alternative: Only one main ending, which still wraps up the story neatly
- Ending the game if the player does not fight the first boss for a specified number of rooms
    - Will trigger a fully optional "hidden" ending as this is not in the intended gameplay loop
    - Impact of Skipping: very little, since the story is already complete at this point
    - Alternative: unlocks an artifact/hidden achievement

Devices:

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

The game will require a mouse and include some keyboard shortcuts.

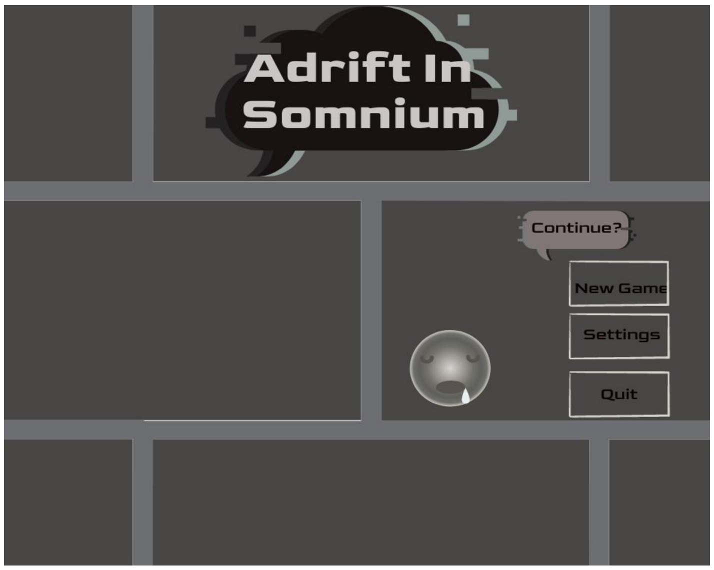
- Mouse Input:
    - Select Options
    - Select where to move on the map (within EP borders)
    - Attack
- Esc to bring up settings/ menu
- W,A,S,D to navigate UI

Concepts:

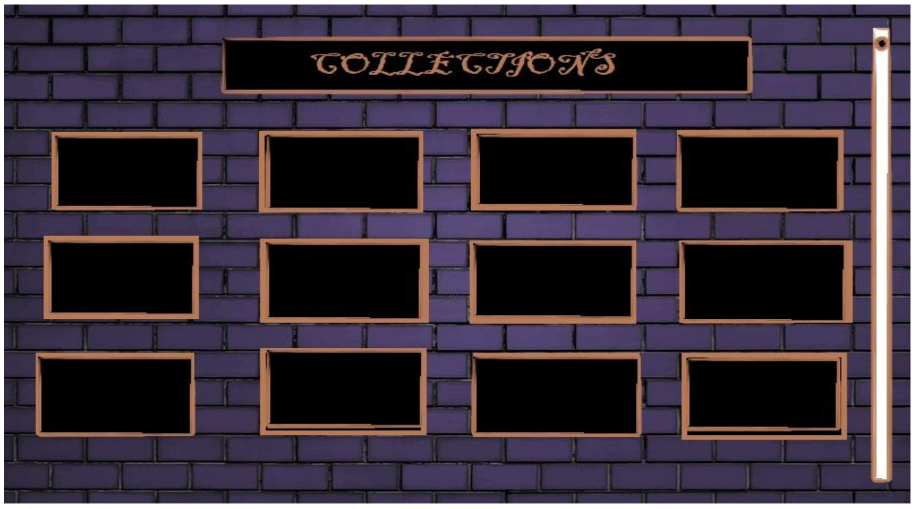
Produce basic, yet descriptive, sketches of the major game states (screens). These should be consistent with the game design elements, and help you assess the amount of work to be done.

Game states

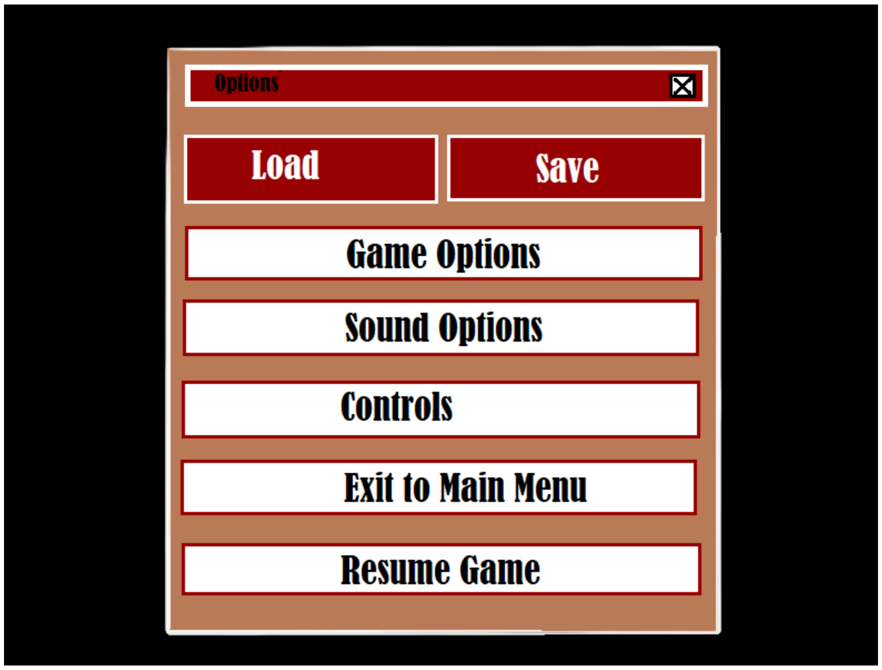
- Main Menu



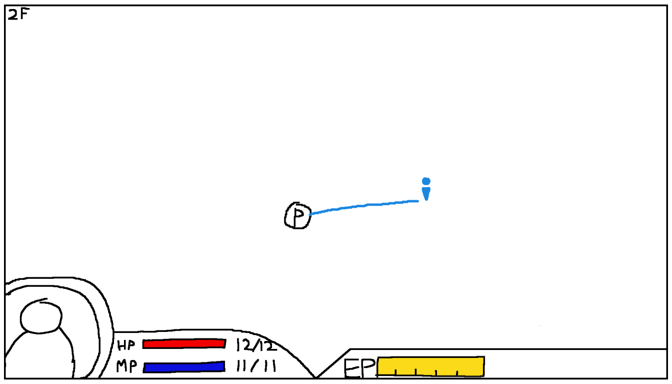
- Artifacts Menu



- Options Menu (Esc)

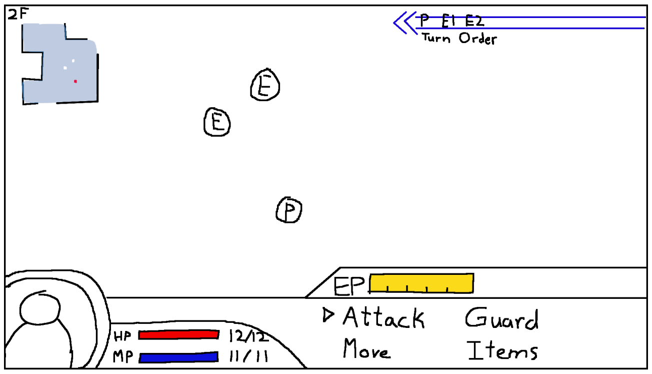


- Free Roam UI



User can move cursor freely and move the player upon click (no EP cost)

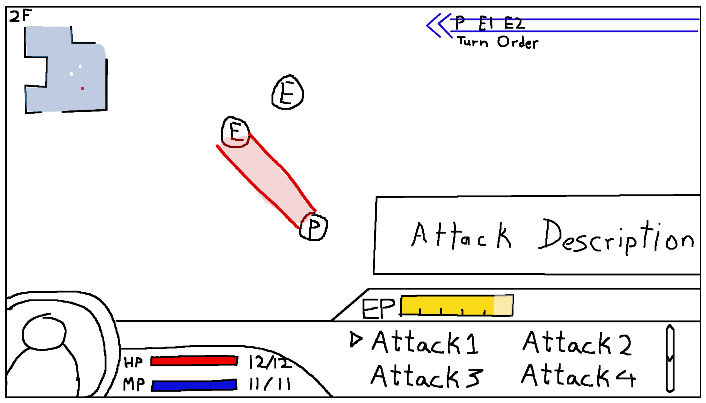
- Combat UI



Turn Order appears and Menu slides up when an encounter with an enemy occurs

User can select which action to take

- Attack UI

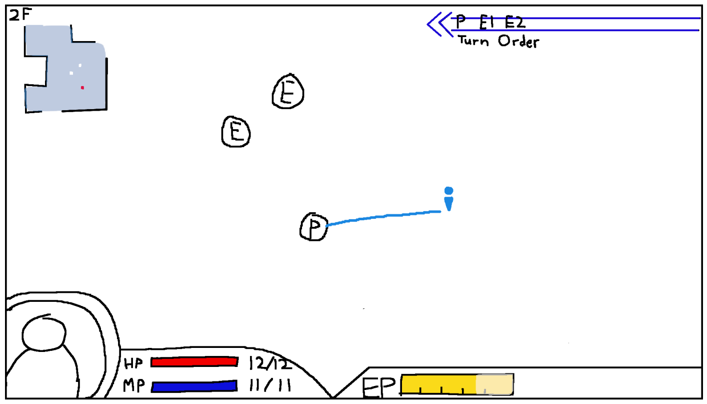


User can select which attack to use

When hovering an attack option, a description of the attack appears and a red outline will display the AOE of the attack relative to the player character

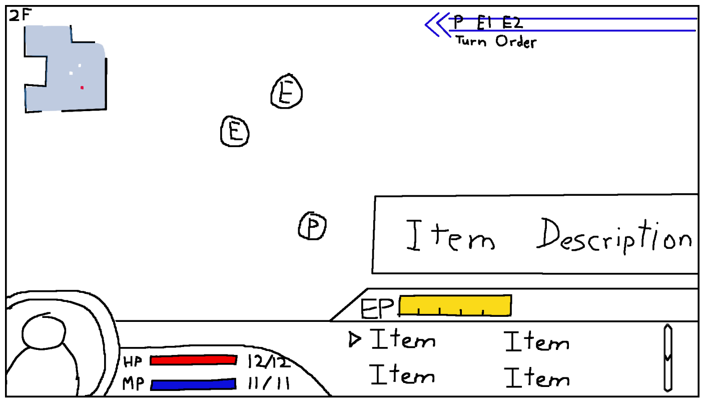
After selecting an attack, user can select the direction of the attack

- In-Combat Movement UI



User can move cursor at the cost of EP, user will confirm movement with a click

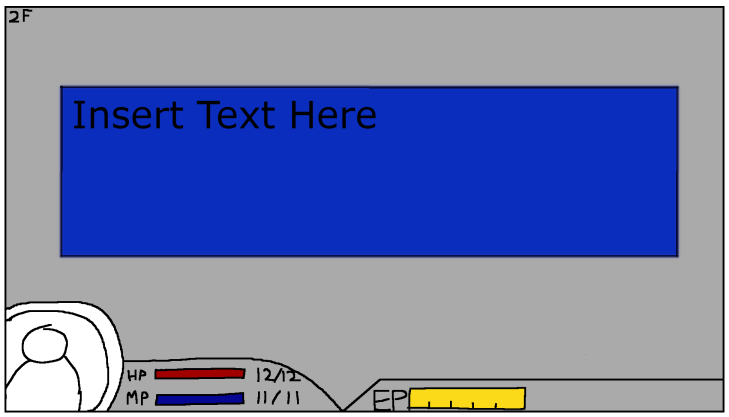
- Items UI



User can select which item to use

When hovering an item option, a description of the item appears

- In-game Text



Pop-up textbox, dimmed background and character expression can change

Tools:

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

Tiled (map creation program)
RapidXML (for parsing .tmx files from Tiled)
FreeType (text rendering)
nlohmann (JSON parsing)

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 using a Kanban board to track tasks and issues, deciding as a group how to split major tasks, and self-assigning more minor tasks. For each week, we will set a number of main tasks according to the roadmap, adding more tasks for issues or extension as needed.

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).

Skeletal Game

Week 1

- Build initial ECS layout
- Create placeholder shapes to represent game elements
- Implement basic movement

Week 2

- Layout fundamental turn order system
- Implement and test collision system
- Develop initial randomization algorithm

Week 3

- Stability and Performance Testing

Minimal Playability (Feb 25)

Week 1

- Create sprites for main character and primary enemies
- Implement an initial set of rooms the player can explore (maybe the first floor?)
- implement basic EP system

- Implement basic enemy pathfinding and decision making
- Create first-draft sprites for main characters and enemies
- Find free sprite assets for background geometry
- Text logging system
- With extra time, look for placeholder music for different game environments

Week 2

- Stability and Performance Testing
- Testing on different devices for consistent resolution
- Start working on story elements

Week 3

- Create refined versions of major sprites and assets
    - Create several animated sprites
- Create visual/sound feedback to user input and actions
- Add placeholder items/artifacts
    - Determine basic game balance afterwards
- Further Stability and Performance Testing after adding a number of game elements

Playability (March 18)

Week 1

- Implement basic text pop-up functionality
- Finalize enemy pathfinding and decision making
- Add reloadability functionality
- Implement artifact system

Week 2

- Create tutorial segment
    - Room with 1 door that spawns an enemy to block door when player approaches
    - Teach player how to use EP to move and attack the enemy to unblock the door
- Implement Archive
    - For Notes
    - For Artifacts & Items
- Implement equipment system

Week 3

- Continue adding game content
    - such as items, artifacts, enemies, and rooms
- Add parallax scrolling for the background textures
- Implement Weapon attacks

Final Game

Week 1

- Add a more diverse set of enemies, rooms, actions and items
- Finalize user core user actions and movement options
- Finalize game balance using input from user testing
- Implement basic 2D dynamic shadows

Week 2

- Finalize previous unfinished features
- Add the parts of the game that require text entry
- Finalize collision physics (ensure everything works precisely)
- Create cutscene(s) to play at the beginning and end of the game
- Major focus on bug fixes discovered during user testing
- Create user options

Week 3

- Add Final floor and Boss
- Wrap up any unfinished features