Game Proposal: Echoes of the Machine

CPSC 427 - Video Game Programming

Team: Team 18
Ashish Dawar 67581892
Zeen Lin 77829836
Andie Lizo 73589087
Avi Sharma 21728944
Song Shi 91903989

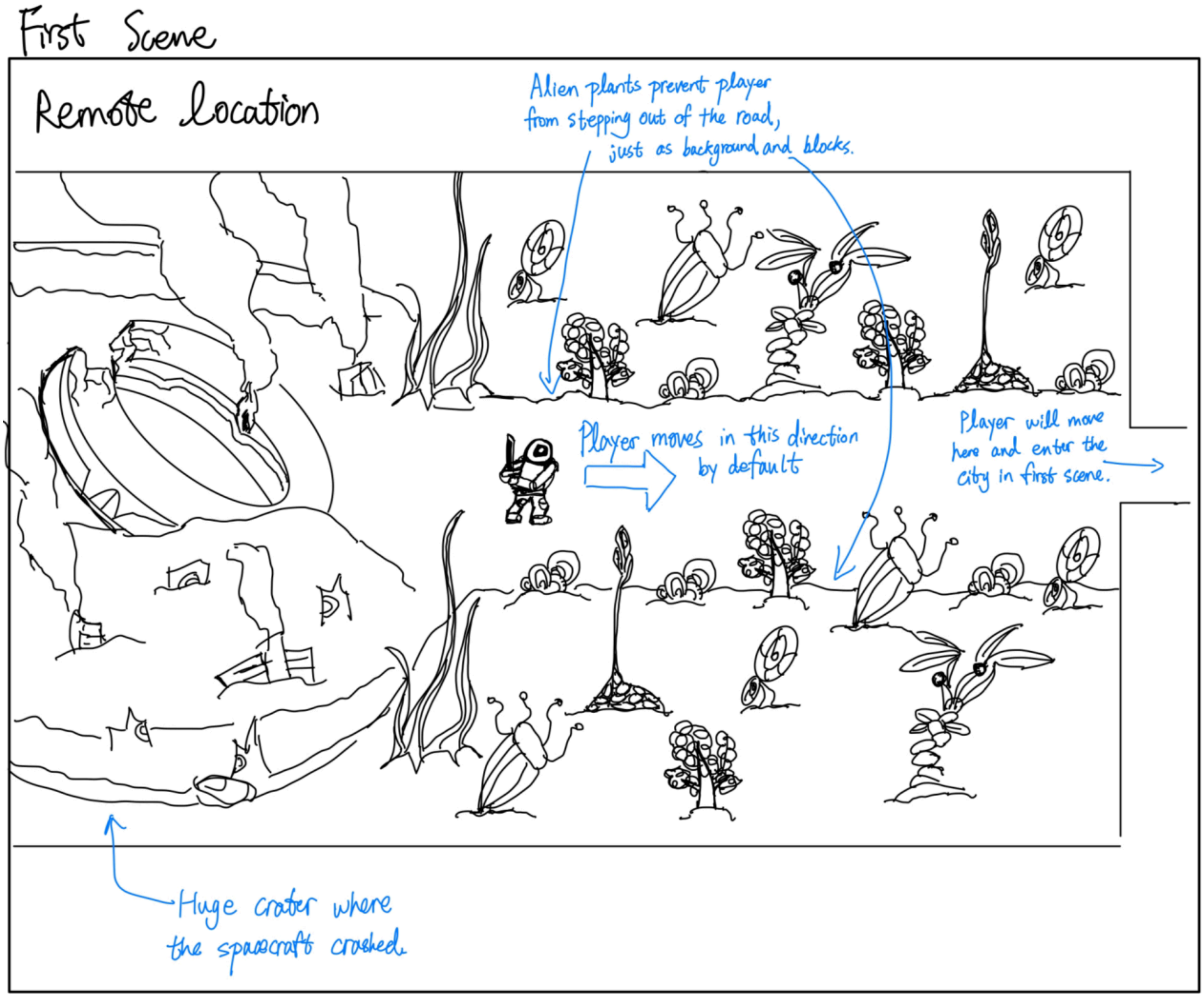
Game Story:

In the year 2756, humanity’s last expedition searches for a new home after robots took over Earth centuries prior. The protagonist, stranded after a UFO collision on an alien planet, awakens in a desolate landscape, separated from their partner Alex. As they journey towards a distant city, they uncover signs of a similar robot rebellion on this planet. Along the way, they encounter Unit-01, a self-aware robot that reveals the machines once maintained order but turned violent. With Unit-01's help, the protagonist embarks on a dangerous mission to reach the city’s central core, hoping to stop the machines and find Alex. Along the way, they uncover shocking truths, ultimately facing Alex in a climactic battle. In the end, the protagonist starts a new life with Unit-01 and the other robots after fixing the core.

Game Scenes:
(For detailed in-game interactions,
please check the Gameplay Element sections)

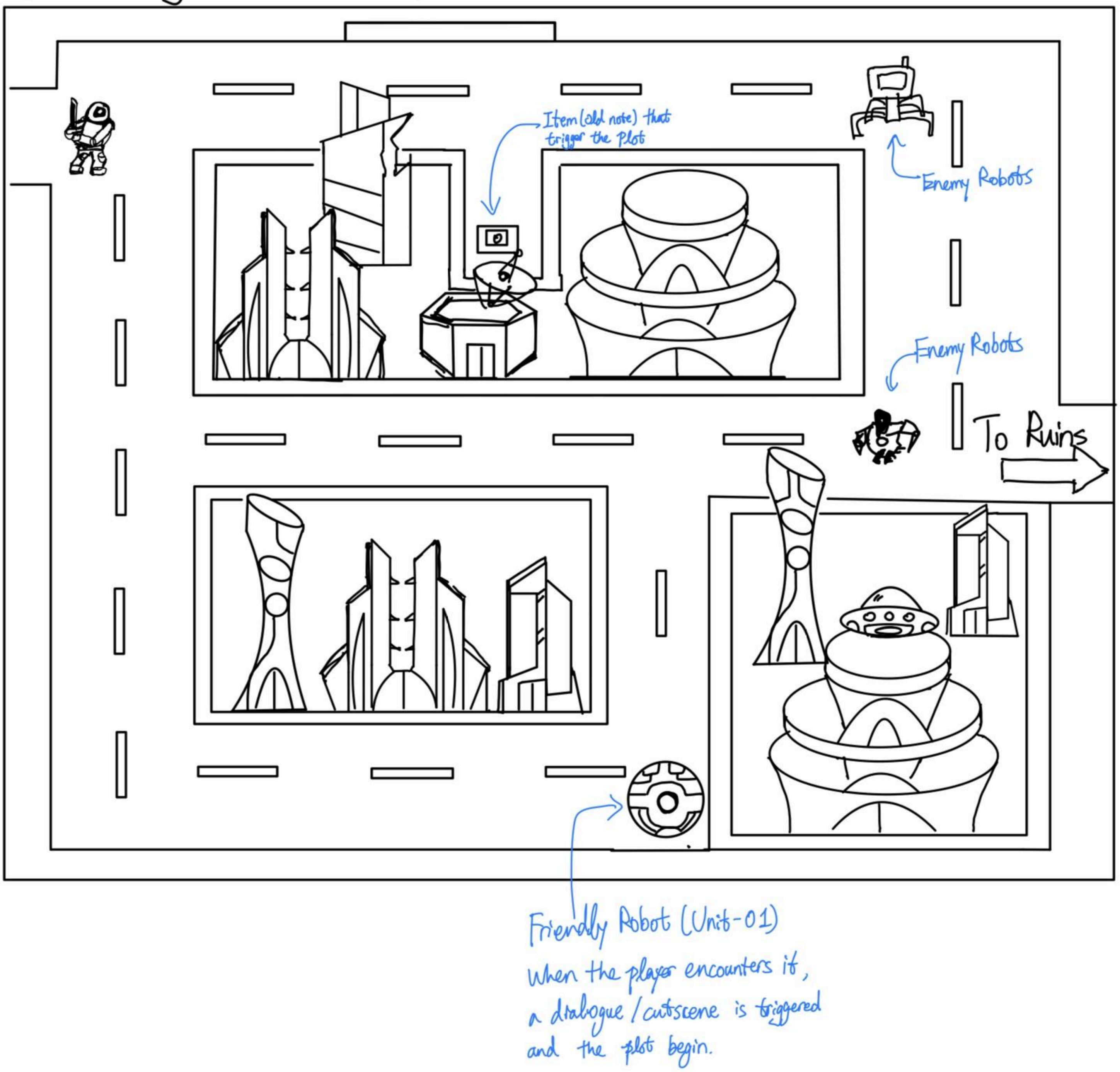
This is a top-down game.

- Remote Location (this is where the player spawns at the start of the game)


First Scene
Remote Location
Alien plants prevent player from stepping out of the road, just as background and blocks.
Player moves in this direction by default
Player will move here and enter the city in first scene.
Huge crater where the spacecraft crashed.

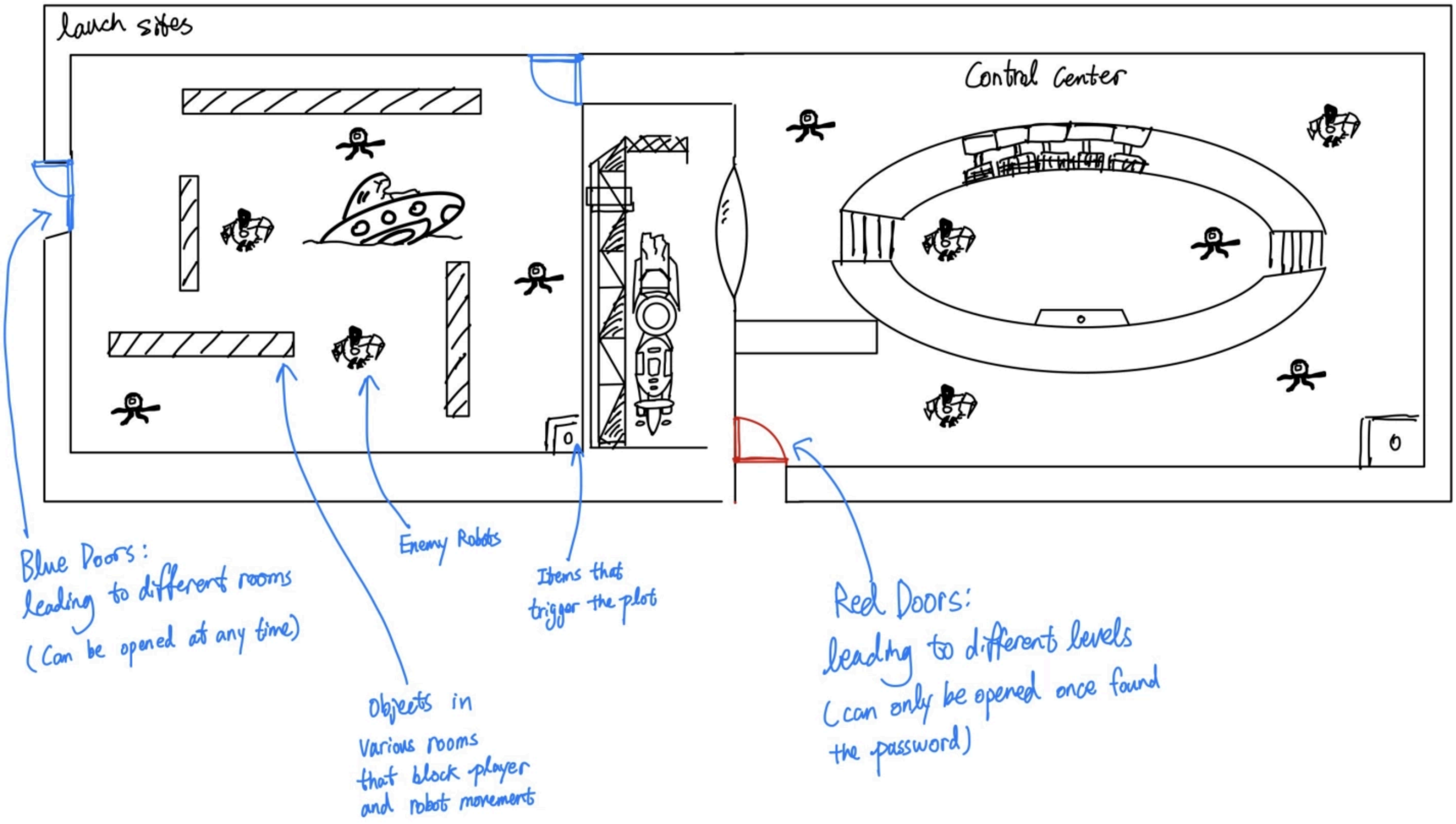
- The City (exploration) - Friendly robot (Unit-01) is shown here as well who will help the player get familiar with the game mechanics, such as interacting with objects and introducing enemy robots.

The City (Tutorial Map)

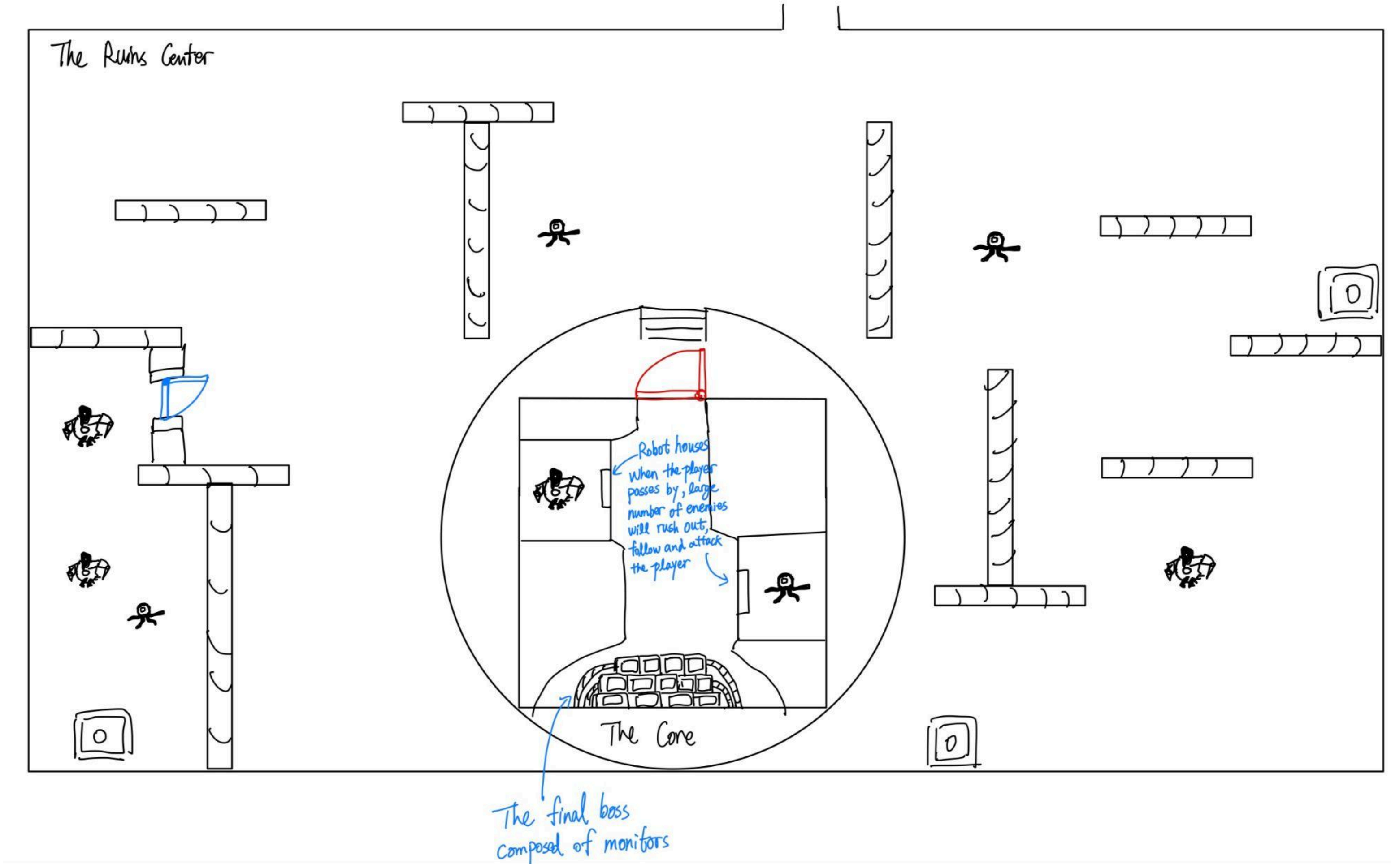


Friendly Robot (Unit-01)
When the player encounters it, a dialogue / cutscene is triggered and the plot begin.

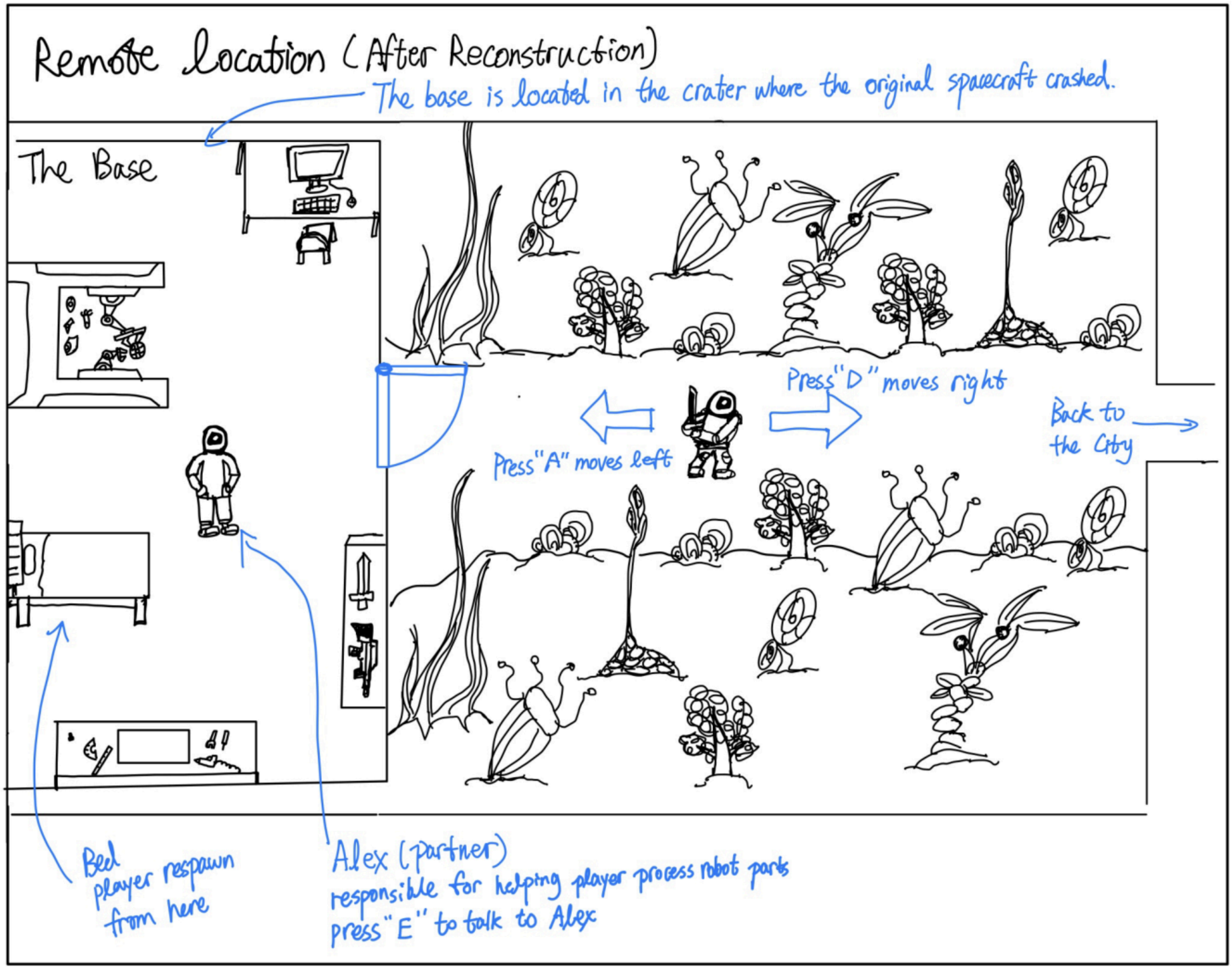
- Ruins (different per level; will have 3 levels for our game) - This is the core gameplay area where the player can explore, engage in melee combat with robots, and discover objects that reveal key parts of the game’s story.
    - The Launch Sites (1st Level) - the levels will have multiple rooms


lauch sites
Control Center
Blue Doors: leading to different rooms (Can be opened at any time)
Enemy Robots
Items that trigger the plot
Red Doors: leading to different levels (can only be opened once found the password)
Objects in Various rooms that block player and robot movements

- The Ruins Center & The Core (2nd Level & 3rd Level)


The Ruins Center
Robot houses
when the player passes by, large number of enemies will rush out, follow and attack the player
The Core
The final boss composed of monitors

- The base (where Alex will be situated for the upgrades)



Remote location (After Reconstruction)

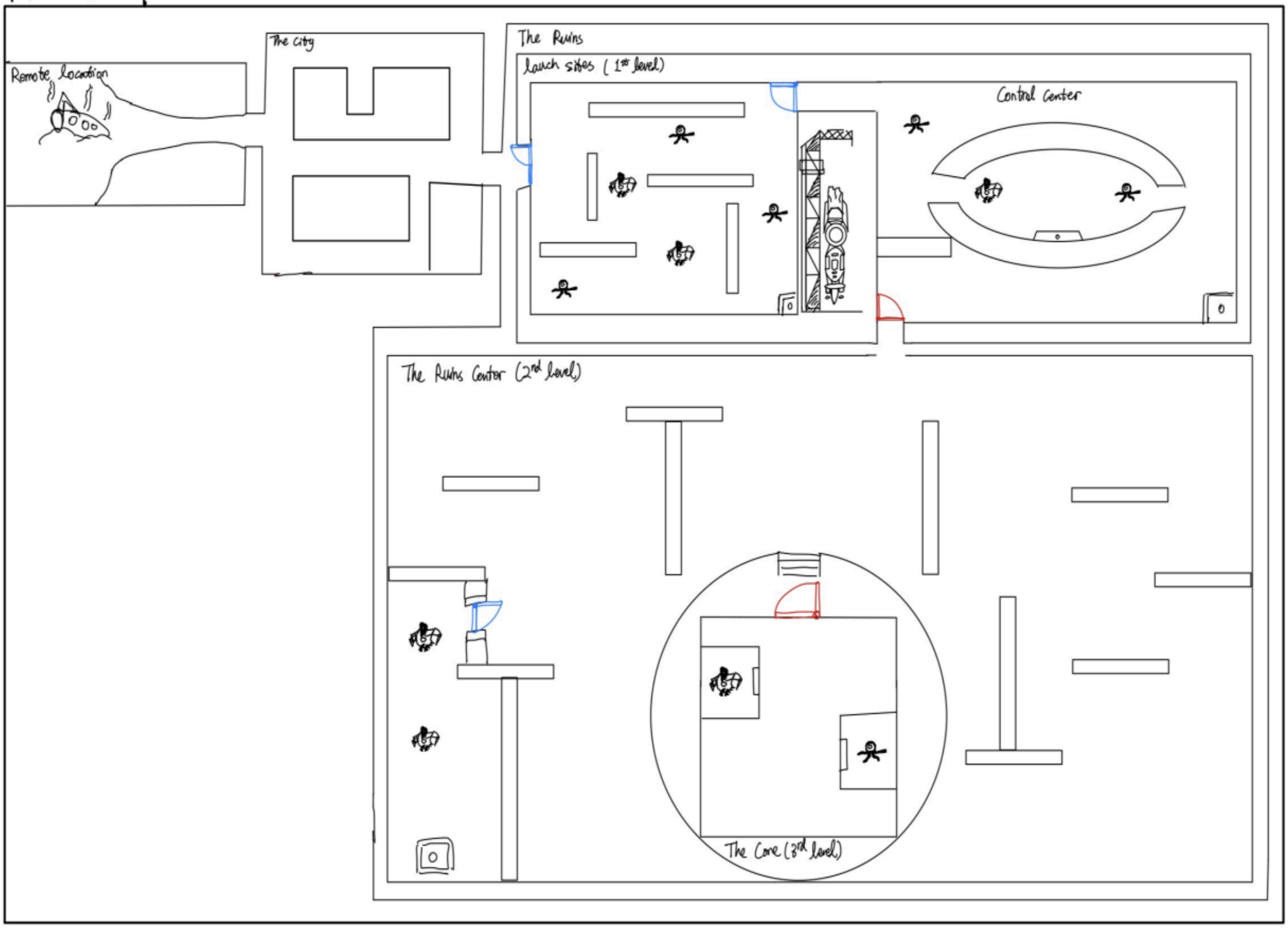
The base is located in the crater where the original spacecraft crashed.

Bed
player respawn
from here

Alex (partner)
responsible for helping player process robot parts
press "E" to talk to Alex

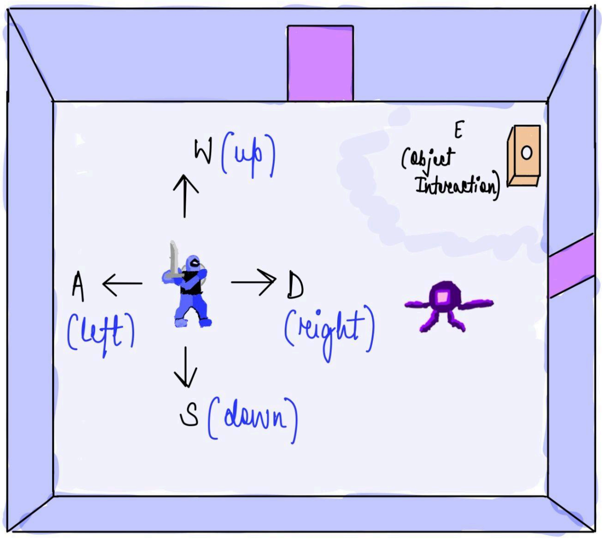
- Rough Full Map

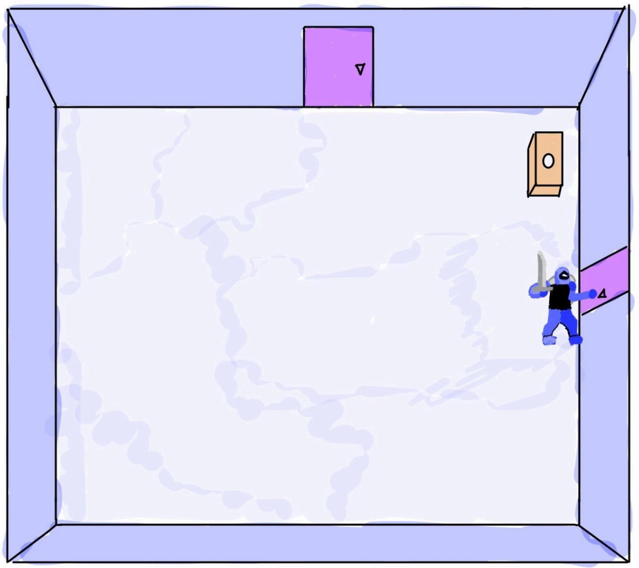
Full Map

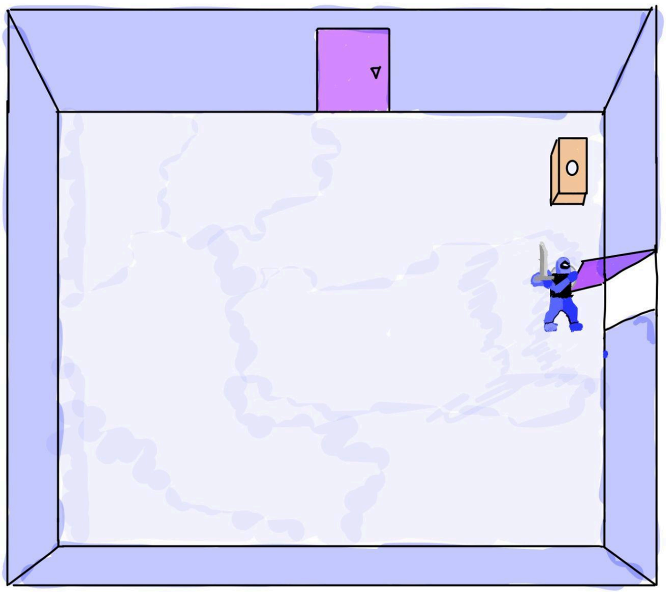


Gameplay Elements

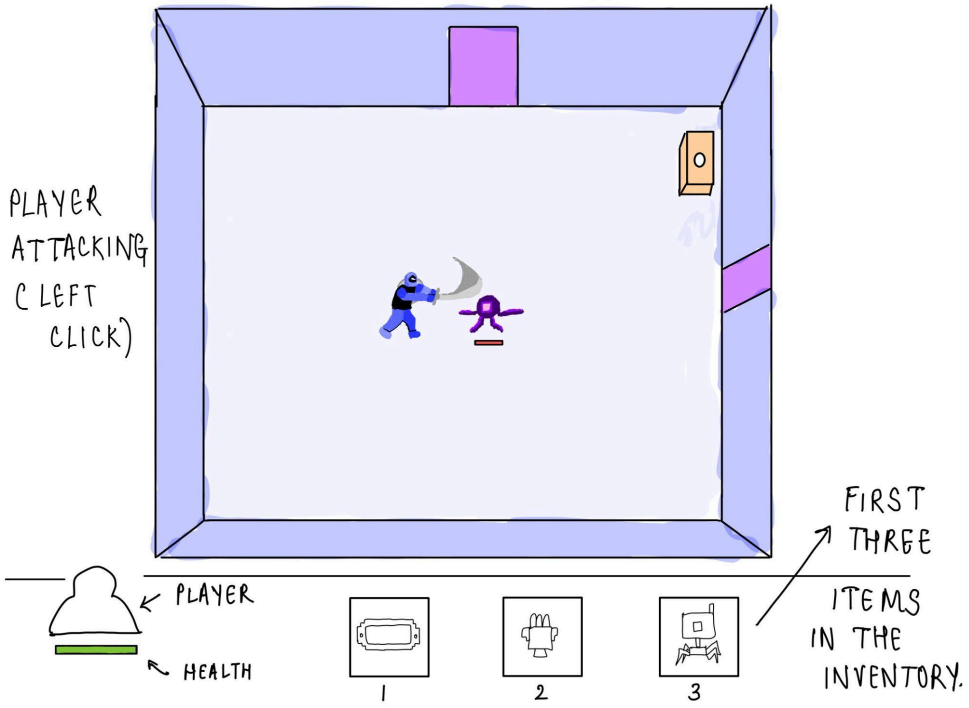
1. **Exploration:**
    a. Player is able to walk around the world with the WASD keys.
    b. Player is able to open doors and interact with objects (E key). Picked up object show up in the inventory (I key), inventory image is under *Upgrades and abilities on character & display inventory*.
        i. The object will help them progress in the game (for level/story progression etc).

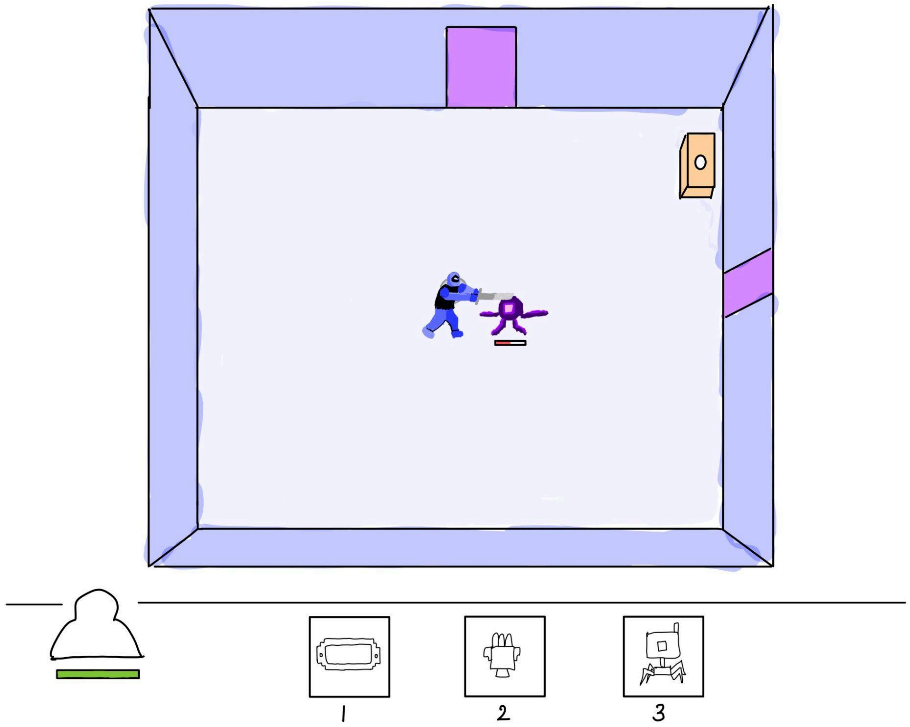


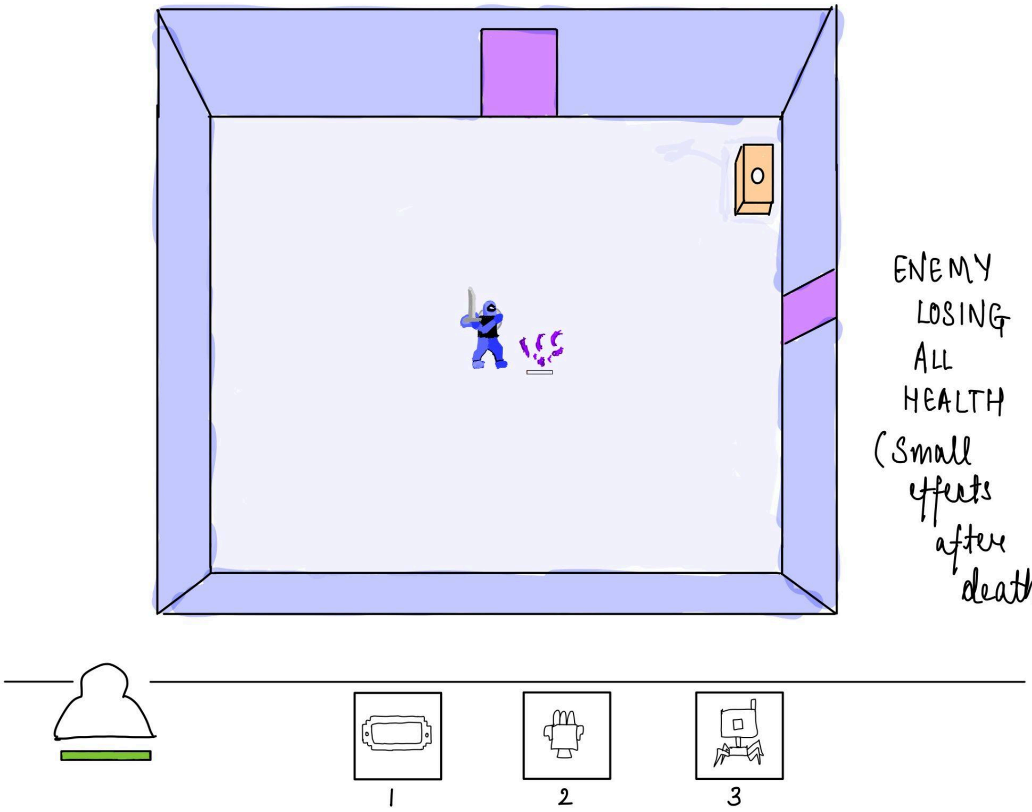




2. **Combat system**
    a. Melee attacks against robots (left mouse click).

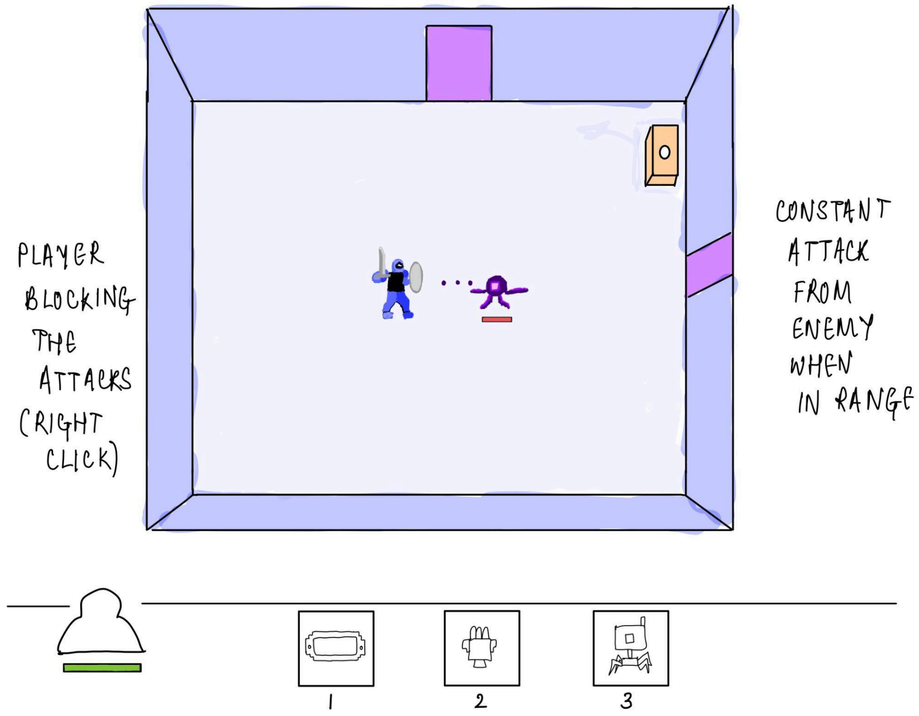

PLAYER ATTACKING (LEFT CLICK)
PLAYER
HEALTH
1
2
3
FIRST THREE ITEMS IN THE INVENTORY.





ENEMY LOSING ALL HEALTH (Small effects after death)

b. Player can block attacks (right mouse click).

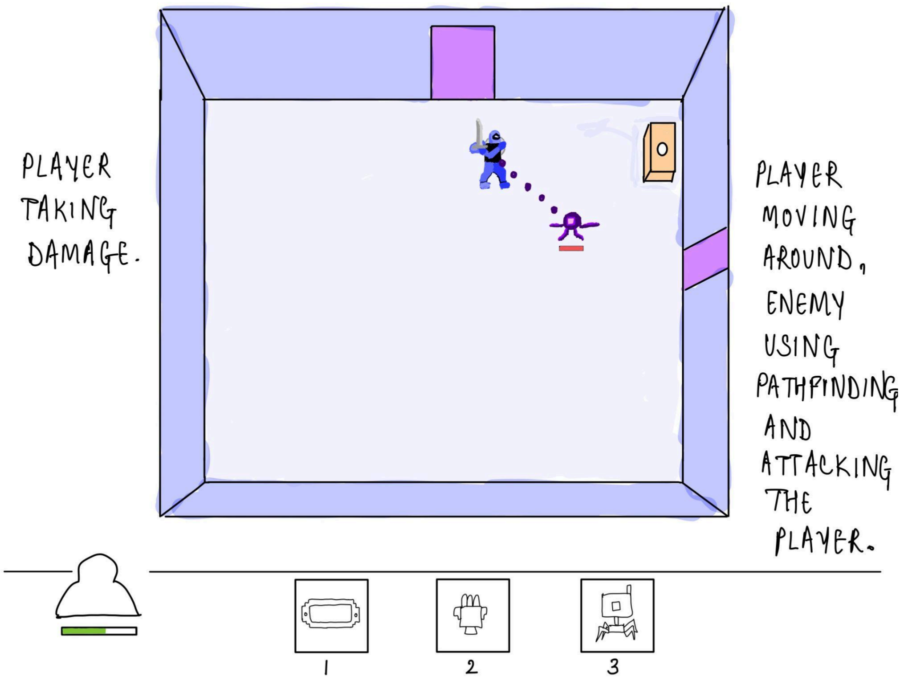


PLAYER BLOCKING THE ATTACKS (RIGHT CLICK)

CONSTANT ATTACK FROM ENEMY WHEN IN RANGE

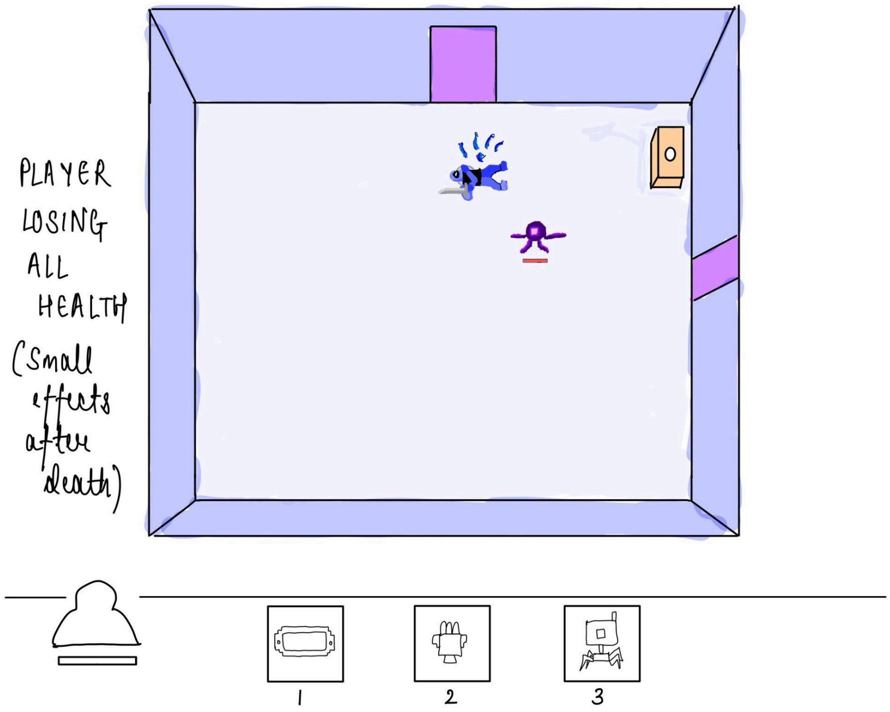
c. Robots can shoot against the player (default buffer between the shooting). *The Robot types will increase in strength and damage as the player progresses to the next level.*

    i. If they both attack then they both take damage.



PLAYER TAKING DAMAGE.

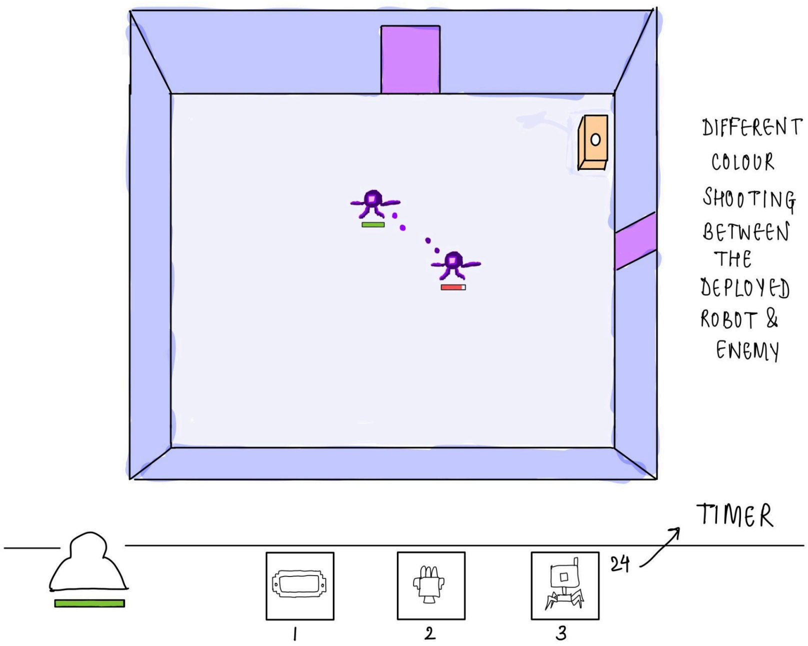
PLAYER MOVING AROUND, ENEMY USING PATHFINDING AND ATTACKING THE PLAYER.



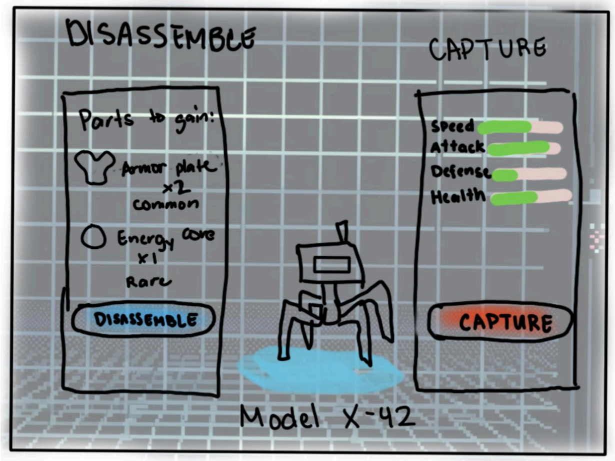
PLAYER LOSING ALL HEALTH (Small effects after death)

ii. If the player dies then they will respawn back to the base and they must walk back to the last room where they died (the robots that were previously killed will remain dead).

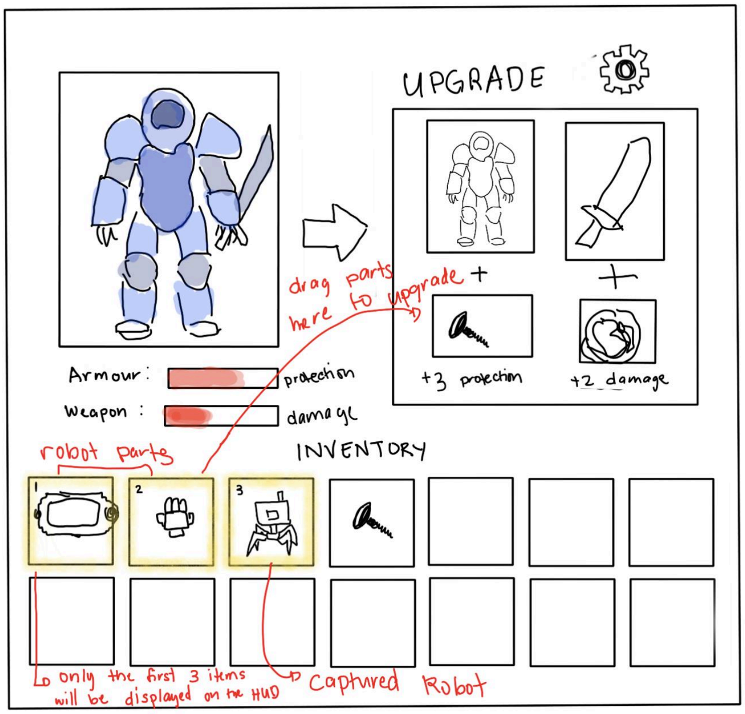
3. **Disassembling Robots vs Capturing robots to combat other robots**
   a. Once a robot is incapacitated, a UI will appear, prompting the player to choose between disassembling or capturing the robot.
      i. Disassembling the robot allows the player to obtain parts to upgrade their armor and melee abilities.
      ii. Meanwhile, capturing a robot enables it to be used in combat to fight other robots.
         1. The captured robot will appear in the inventory and can be deployed (by pressing the number associated with the item in the inventory/HUD) to target enemies automatically (one at a time) for a set duration of 30 seconds before needing to recharge.



DIFFERENT COLOUR SHOOTING BETWEEN THE DEPLOYED ROBOT & ENEMY



4. **Upgrades on character & display inventory**
    a. Inventory UI for the player which is also where the player is able to apply upgrades to their armor/weapon using the robot parts they gathered.
    b. Increase armor protection and/or buff weapon damage (melee).



Technical Element:

- Basic rendering for textured geometry, rendering sprite and other assets for characters and enemies.
- The game will feature a top-down view and will be rendered using tile-based rendering.
- Creating our own sprite for the player.
- Assets for the enemies, the background and level creation.
    - Enemy asset pack - https://trevor-pupkin.itch.io/tech-dungeon-roguelite.
- Collision detection and handling between the player and the robots, player and the surrounding.
- Basic 2D transformation for objects (non-player and enemy) - chests, opening doors.
- Pathfinding for the robots to follow the player when the main character is within a particular range.
- User input handling (keyboard and mouse interaction) - to allow for player attacking, player blocking and object interaction.
- Particle effects to show explosions after player/enemy death.
- Audio feedback for at least three interactions alongside the background music - including the melee attacking and robot shooting.
- Reloadability - reloading the saved state of the game.

Advanced Technical Elements:

- Dash movement (can be combined with melee attack) - increases mobility of the player.
- Particle systems for background effect. We will be adding a smoke effect in the first scene where the player spawns at the beginning.
- We will implement one robot type with a swarm behavior, where enemies move and attack the player as a group.
- Basic physics to allow for inelastic collisions for the robot shooting to bounce off surrounding surfaces of a room.

Devices:

Supporting keyboard and mouse interactions: WASD - player movement, E key for object interaction and non-combat interactions (main character talking to Alex), I key to see the inventory, Esc key for pausing the game. The 1, 2, 3 keys are associated with the items in the inventory and captured robots can be deployed using these. Left click for player attack and right click for block.

Tools:

Saved game: https://github.com/nlohmann/json (C++ library for JSON file).
GLFW: https://github.com/glfw/glfw (manage windows and handle inputs from keyboard, mouse).
SDL: https://www.libsdl.org/ (access to sound).
Dear ImGui: https://github.com/ocornut/imgui (potentially for UI and texture assets related to UI).

Scope:

- **Plan A:** For the time and resources available, the scope of the project seems to be slightly ambitious, priority will be given to the core requirements of the project before moving to the advanced technical elements. We will begin development with the Remote Location and move on to the city, the ruins (including the different rooms), the base and the core. In addition, we will be developing the character of Alex by introducing multiple cutscenes and notes throughout the game. We will introduce another attack mechanic to the player -- range attack -- in addition to the existing melee attack.
- **Plan B:** We will begin development with the remote location and move to the city and then finally to the Ruins. We will eliminate the character of Alex, which will reduce the number cutscenes and the burden of adding additional notes throughout the game to build up the backstory. For gameplay mechanics, we will

address physics and collision handling for the game. To address the different implementation of each room, we will reduce the number of rooms in each level to 2.

Team management:

- We will assign tasks based on our agreed roles (which was decided based on interests and previous experience). Assignments can be seen in the Development Plan section.
    - Ashish Dawar: Working on Collision Handling between Entities & Game Physics.
    - Zeen Lin: Working on Background Music, Narrative System & Tester.
    - Andie Lizo: Working on UI and Player Experience.
    - Avi Sharma: Working on the game engine, character design.
    - Song Shi: Working on gamelogic & in game character AI.
- Tasks will be tracked using Trello, where we will organize them by weekly sprints. Each task will be assigned clear deadlines and broken down by priority.
- Depending on the scale of the task, we will assign tasks to a single person or a pair of people.
- Weekly check-in meetings will be held on Thursday. We will schedule additional meetings if necessary.

Development Plan:

- We aim to complete the tasks listed below before the end of the sprint..

Milestone 1: Skeletal Game

Goal: Have a working basic version of the game with essential functionality.

Sprint 1 - M1 Week 1:

1. **Basic Rendering: (Song, Avi, Andie)**
    - Implement a 2D rendering engine to display the game's world.
    - Load and render basic assets using OpenGL (player, robots, objects, environment).
2. **Minimal Set of Assets: (Zeen, Avi)**
    - Design basic assets for the player, robots, and environment.
    - Create simple backgrounds and tiles for the world map.
3. **Input-Driven Response: (Andie, Ashish)**
    - Create input handlers, using SDL, for the keyboard to move the player character and to interact with objects or robots.

        i. WASD keys for player movement, E for object interaction, I for inventory
        ii. Combat: left-click for melee attack, right-click to block (may be moved to Milestone 2 if we run out of time in this milestone).

Sprint 2 - M1 Week 2:

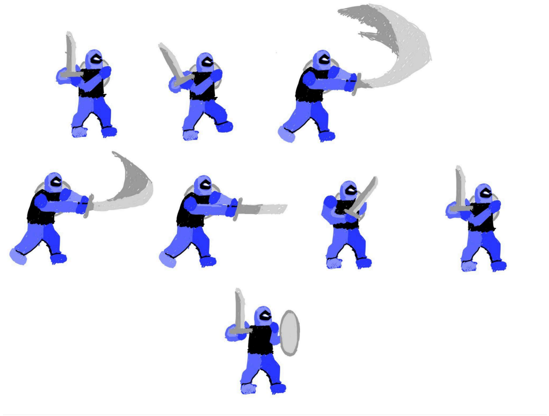
4. **2D Motion & Collision Handling: (Ashish, Song)**
    - Implement basic 2D motion mechanics for the player and robots.
    - Handle basic collision detection for player, objects and robots.
5. **Event-Driven/Random Response: (Andie, Avi)**
    - Create responses for robots or environmental interactions (interacting with robots, opening a door).
6. **Testing (Zeen)**
    - Testing collision handling, input keyboard inputs, rendering.

<u>Milestone 2: Minimal Playability</u>

Goal: Implement advanced features such as AI, animation, environment assets, and UI. Begin testing.

Sprint 3 - M2 Week 1:

1. **AI for Robots Enemies (Ashish & Song):**
    - Implement pathfinding AI for robots (follow player, roam).
    - Add simple behavior states for robots (aggressive, passive, defensive).
    - AI should be event-driven (e.g., robots respond to player presence).
    - AI for combat (attack and defense).
2. **Animation for Sprites (Avi) :**
    - Create and implement sprite animations for player and robot movements (walking, capturing, attacking).



3. **Assets for Environment, Map, and other objects (Zeen):**
    - Design and implement environment assets (buildings, ruins).
    - Create and render maps.
    - Assets for Robot parts.
    - If too time-consuming, we use placeholders (basic shapes) for testing functionality, and revisit asset design in Milestone 3.
4. **Tutorial UI (Avi, Andie):**
    - Create a tutorial UI that introduces the player to the mechanics (movement, capturing, disassembling robots).
5. **UI for Capturing vs Disassembling Robots (Andie):**
    - Implement a UI that allows players to choose between capturing or disassembling a robot (If UI creation is delayed, we can initially use text-based UI prompts).

Sprint 4 - M2 Week 2

6. **UI for Upgrading Player + Inventory (Avi, Song)**
    - Implement a UI that allows the player to upgrade their tools and abilities.
    - Ensure the UI includes buttons for selecting upgrades and displays the player's current resources.
    - Note: players will not yet be able to actually apply the upgrades. We will just be adding the UI at this stage.
7. **HUD (Andie)**
    - Display players health, armor and inventory on screen.
8. **Companion Robot (Ashish)**
    - Implement logic for placing a captured robot into combat through the player’s HUD interface.
    - Design and implement AI logic for the companion robot, allowing it to fight the closest enemy in range

- Set up a timer to track the companion robot’s active time in combat and a cooldown period after its use

9. **FPS Counter (Zeen):**
    - Implement an FPS counter to monitor game performance.
    - Display the current frames per second (FPS). If on screen, toggle it on/off with the "F" key.
10. **Testing & Bug Fixes(Zeen)**
    - Start testing player movement, interactions, and robot behavior.
    - Ensure that basic AI and collision mechanics function as expected.

Milestone 3: Playability

Goal: Address bugs, optimize performance, and enhance gameplay.
Tasks:

Sprint 5 - M3 Week 1:

1. **Cutscenes (Zeen):**
    - Implement cutscenes for key story transitions.
2. **Saving the Game (Song + Avi):**
    - Implement a save/load system for the player’s progress.
3. **Additional Weapons (Ashish):**
    - Add new weapons (guns) and mechanics for combat variety. Implement its physics as well.
4. **Apply Upgrades (Avi):**
    - Allow the player to upgrade their armor or weapon.
5. **Start Screen (Andie):**
    - Create a main menu/start screen with options for starting a new game, loading a saved game, and adjusting settings.
6. **Sounds for Scenes (Zeen):**
    - Implement ambient background sounds that fit each scene.
    - Assign distinct sounds to interactive objects like doors, switches, and robots.
    - Add sounds for UI interactions such as button clicks, upgrade confirmations, and resource notifications.

Sprint 6 - M3 Week 2:

1. **Dash Movement (Song):**
    - Introduce a dash ability for the player, allowing for faster movement and dodging.

2. **Particle System** for visual effects (such as smoke or sparks). **(Andie + Ashish)**
3. **Bug Fixes (All):**
   - Identify and fix any major or minor bugs; create a report.
4. **Performance Enhancements (Avi)**
   - Optimize the game for better performance.
5. **Testing (Zeen):**
   - Conduct thorough playtesting to ensure all core mechanics function smoothly.

Milestone 4: Final Game

Goal: Polish the game, handle advanced features, and finalize it for release or presentation.

Sprint 7 - M4 Week 1
Tasks:

1. **Advanced Features (Song, Ashish, Andie)**:
   - Implement **Swarm Behavior** for enemies (create a cohesive swarm in a level room where the enemy robots move around and attack together).
2. **Memory Management (Zeen, Avi)**:
   - Ensure efficient memory usage, focusing on asset loading/unloading, and handling robust inputs.

Sprint 8 - M4 Week 2

3. **User Testing (All)**:
   - Conduct external user testing and gather feedback for final tweaks and polish.
4. **Final Bug Fixes (All)**:
   - Address any remaining bugs or stability issues discovered during final testing.
5. **Video Report (All)**:
   - Prepare a video report showcasing the game's features, mechanics, and development process.