Game Proposal: Nova

CPSC 427 – Video Game Programming

Team: 5 People

Matthew Smith, 51209682

Ke (Steve) Ren, 91955617

Philip Macau, 25060179

Frank Yu, 70917505

Pranav Laiya, 25663493

Story:

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.

The player begins their journey as a spaceship pilot, transporting important blueprints across the galaxy to their home planet. However, along the way, a mysterious beam shoots up from a nearby planet, striking the ship and causing a crash landing. Upon arrival, the blueprints have been scattered across the planet and their ship is damaged to the point where it cannot leave. The game's primary goal is to recover the blueprints, repair the ship, and find a way to escape the planet.

As the player explores, they can traverse the planet's diverse biomes, uncovering its secrets and gathering resources to upgrade their ship and weapons. Along the way, they encounter both hostile and friendly aliens. Some view the player as a threat, while others are willing to help cautiously. The reasons for this hostility become clear as the player uncovers the planet's hidden history.

Through ancient ruins and alien artifacts, the player learns the truth behind the planet: humanity once invaded it, looting its rich resources and devastating the lives of the aliens living there. Entire alien communities were destroyed, and the ecosystem was thrown into chaos. This explains the beam that was shot up from the planet and hit the players ship.

As the player pieces their ship together and recovers the blueprints, they face moral dilemmas. Are humanity's goals justified or driven purely by the desire for power and resources? is returning home really the right choice, or would staying behind to help rebuild the planet and earn the trust of extraterrestrial species be a better path?

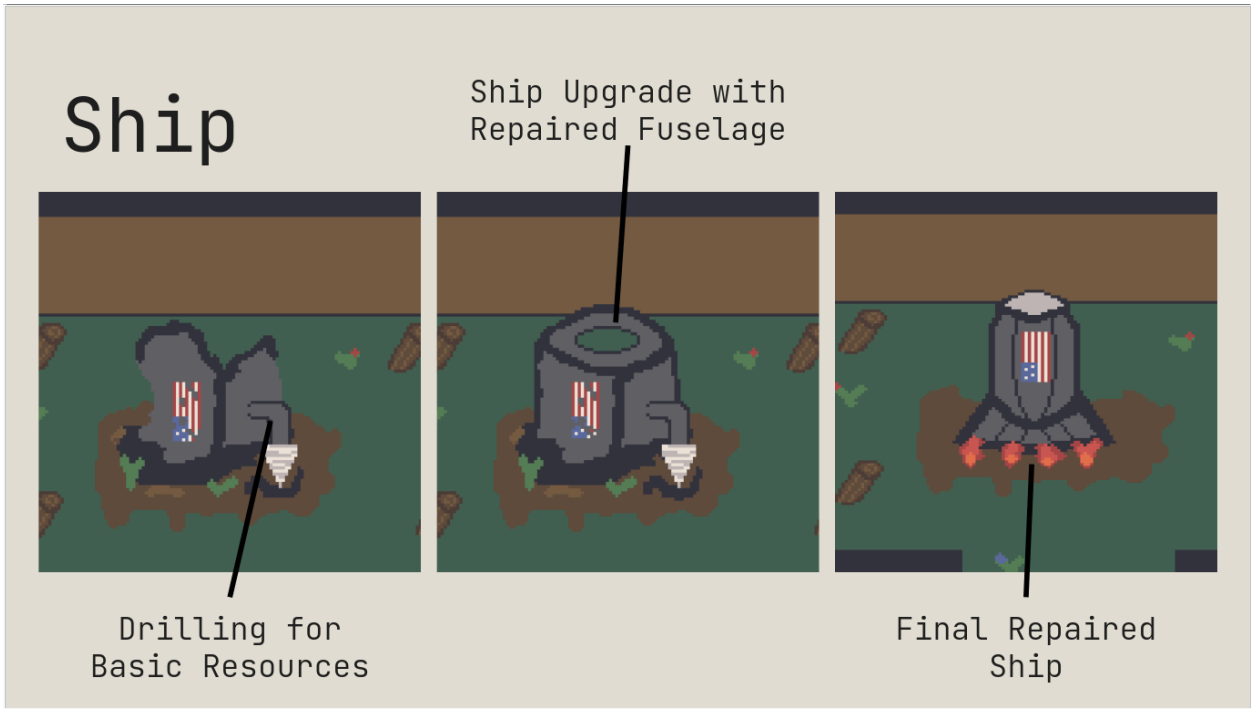
Ultimately, with the ship fully repaired and the blueprints nearly complete, the player faces a final choice: leave the planet and return to humanity to fulfill their original mission or remain behind to aid the aliens and rebuild their planet. The decision is theirs, but it will shape the legacy the player leaves behind.

Scenes:

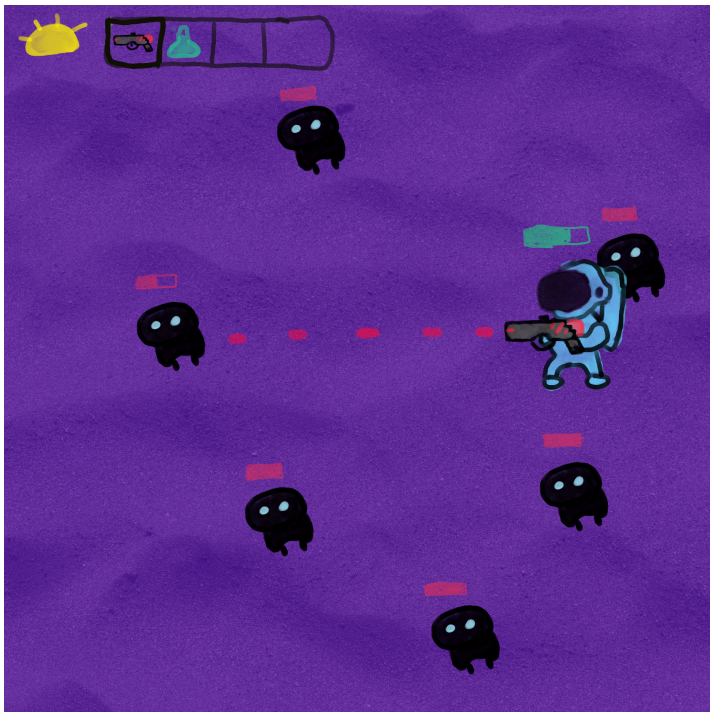
Produce basic, yet descriptive, sketches of the major game states (screens or scenes). These should be consistent with the game design elements, and help you assess the amount of work to be done. These should clearly show how players will interact with the game and what the outcomes of their interactions 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 feeds into the technical and advanced technical element sections below. If taking inspiration from other games, you can include annotated screenshots that capture the game play elements you are planning to copy.

Art Direction and Concept Arts:

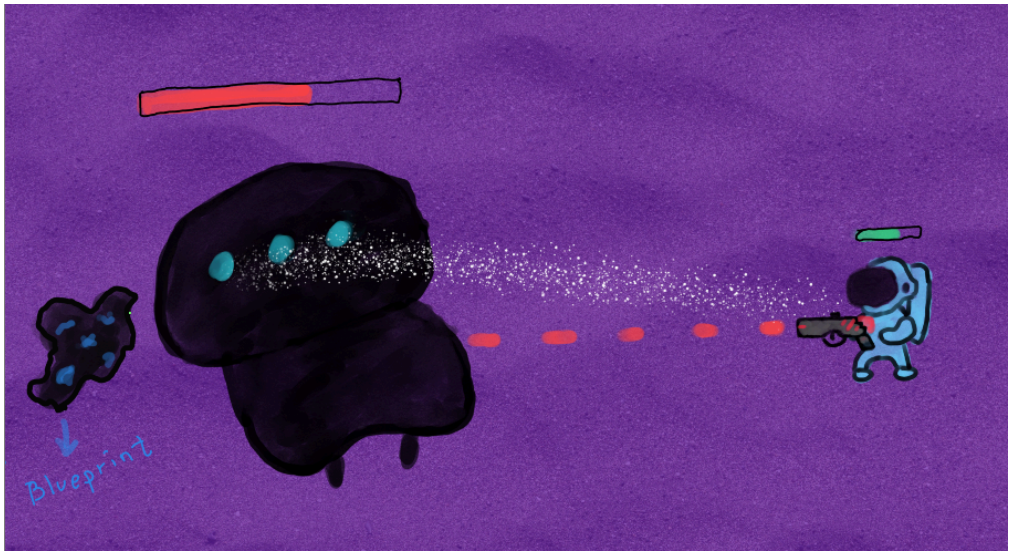
Ship progression:



Combat during day:

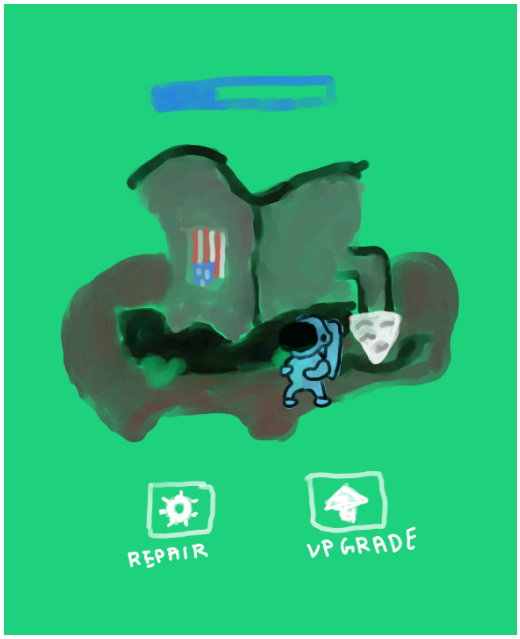


Boss fight:



Blueprints obtained upon defeating the enemy

Combat during night (defending the ship from mob attack):



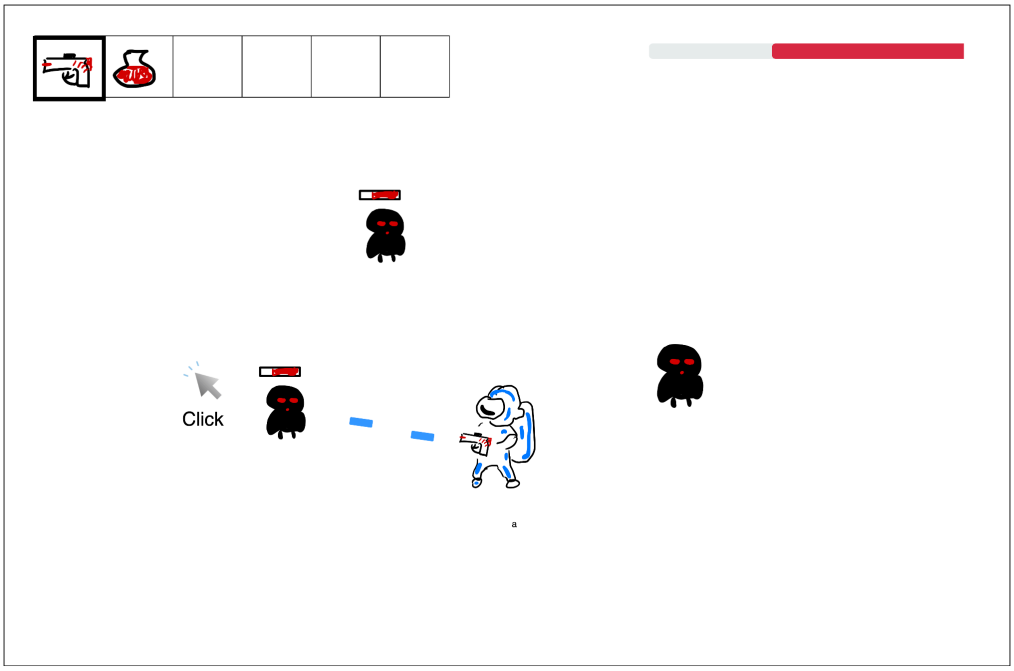
Story telling:

Game Play Scene Demo:

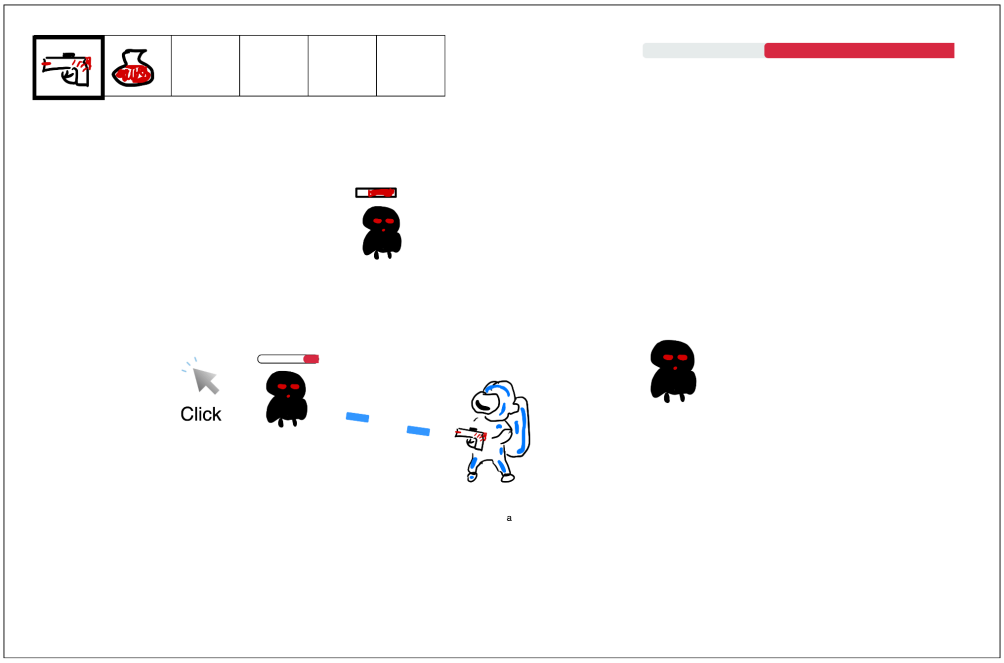
Landing/Menu Page:

Sample gameplay in a biome:

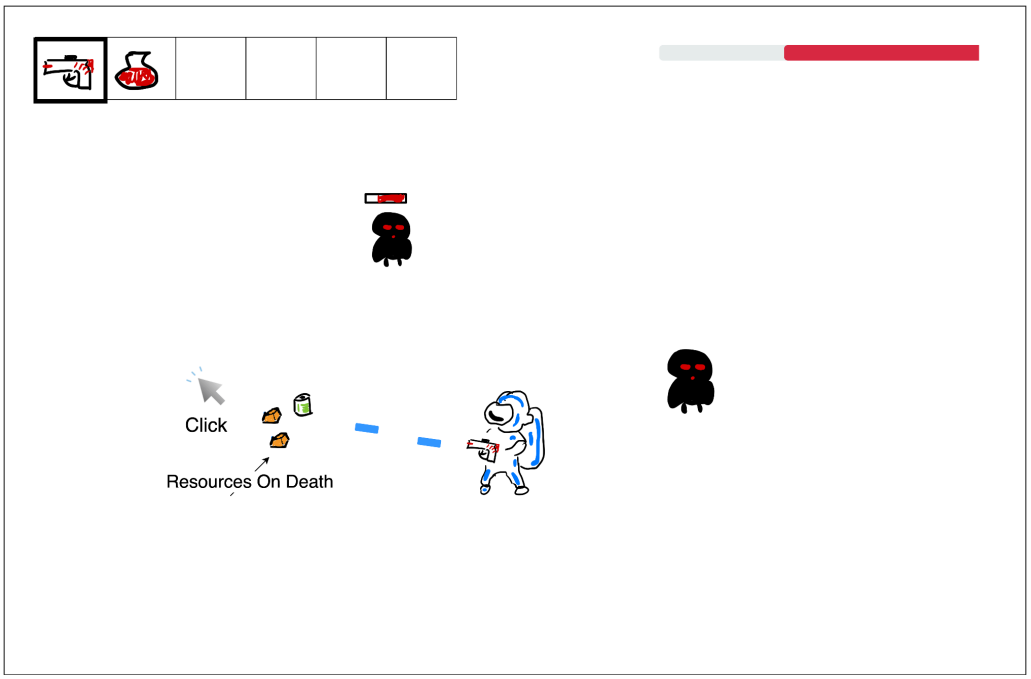
Using a range weapon to attack enemies:



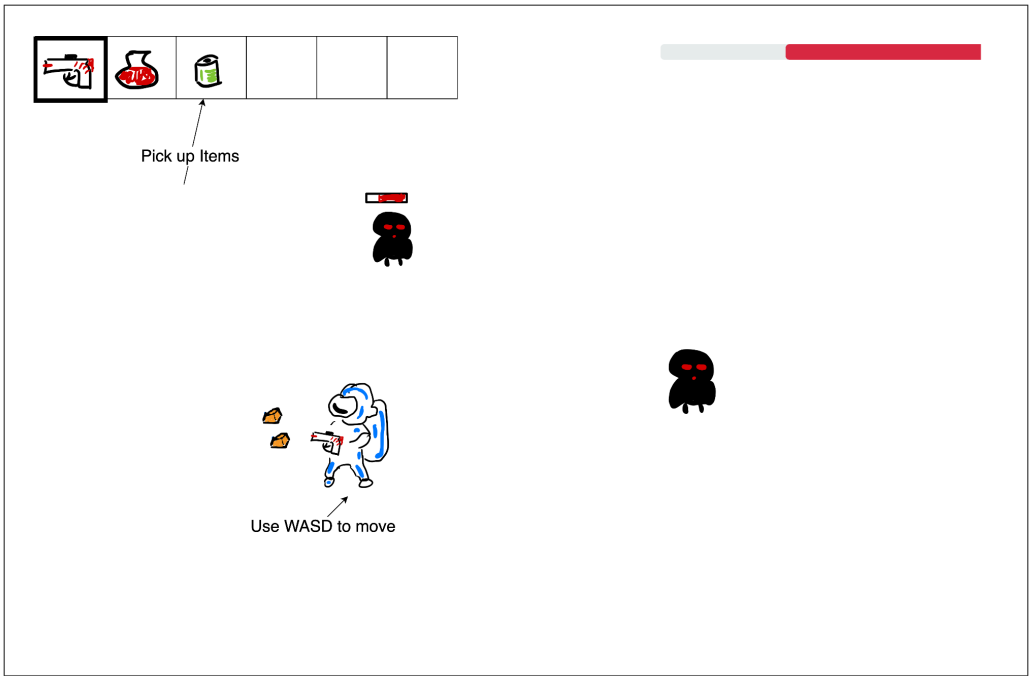
Enemy loses health on hit:



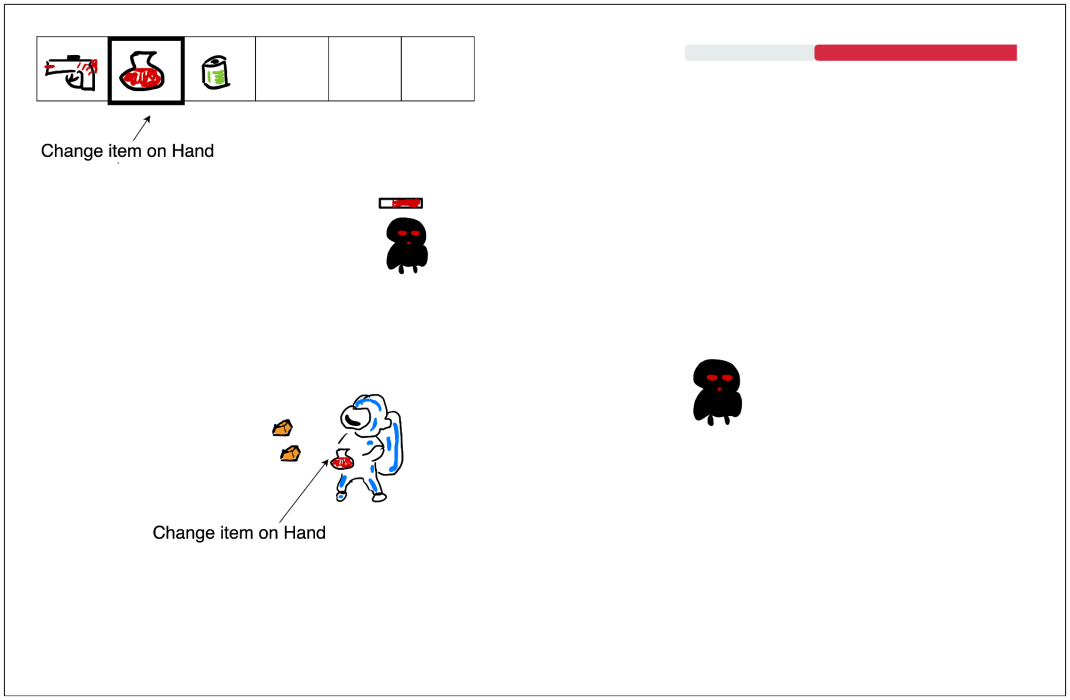
Enemy drops resources on death:



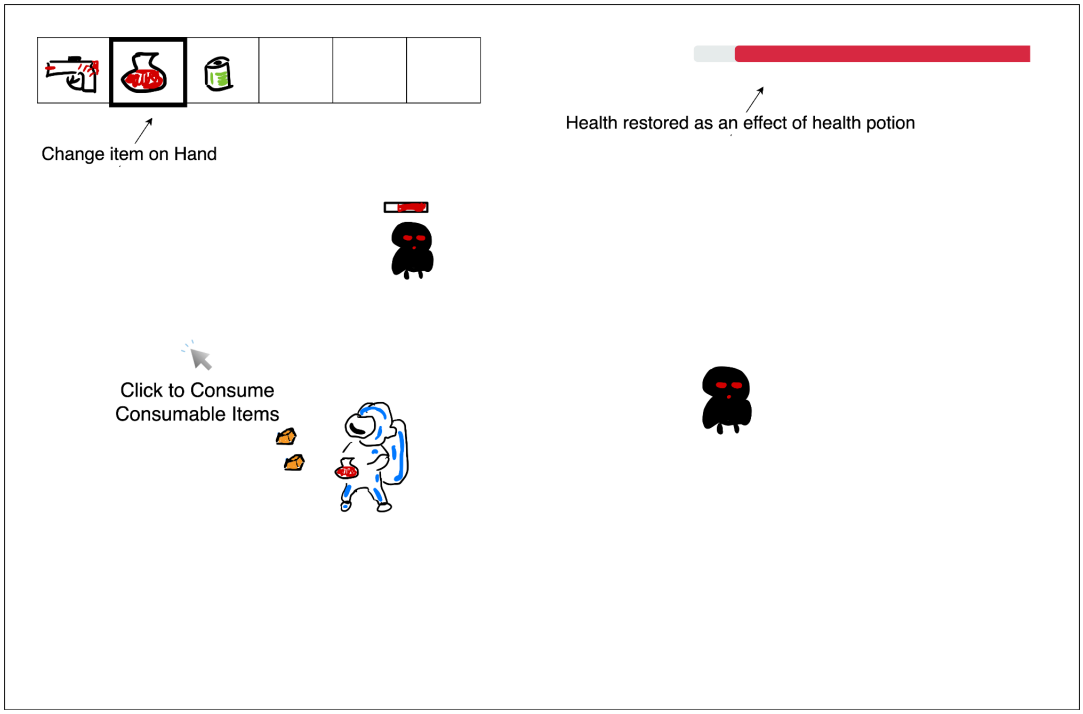
Picking up resources:



Changing items on hand:

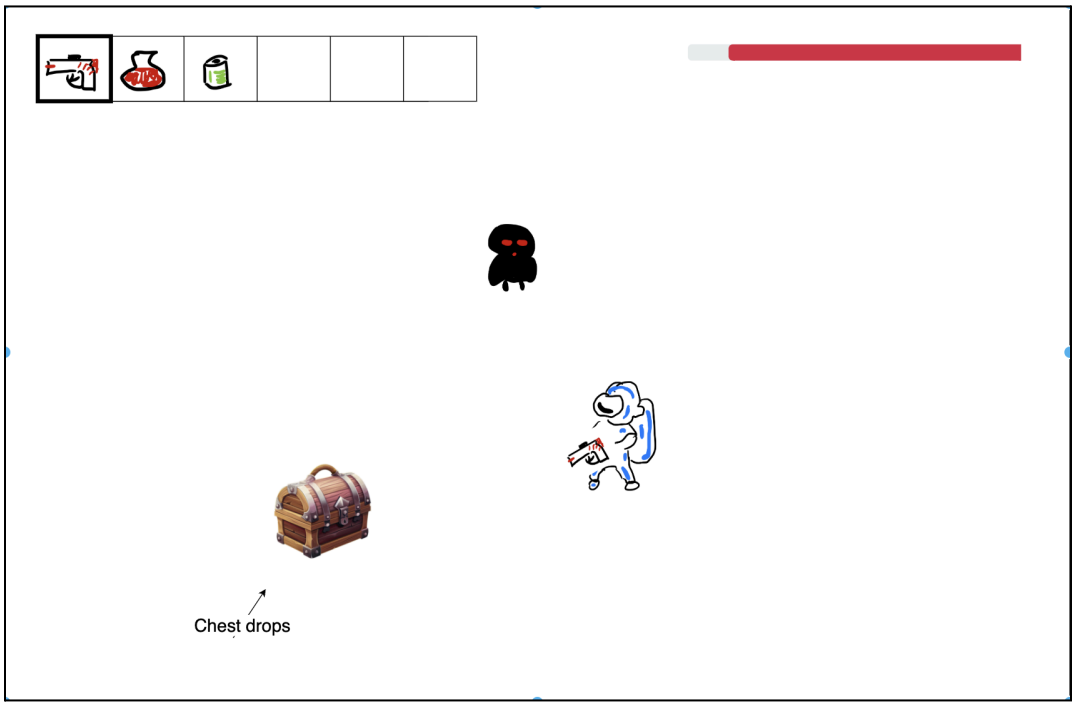


Using health potion:



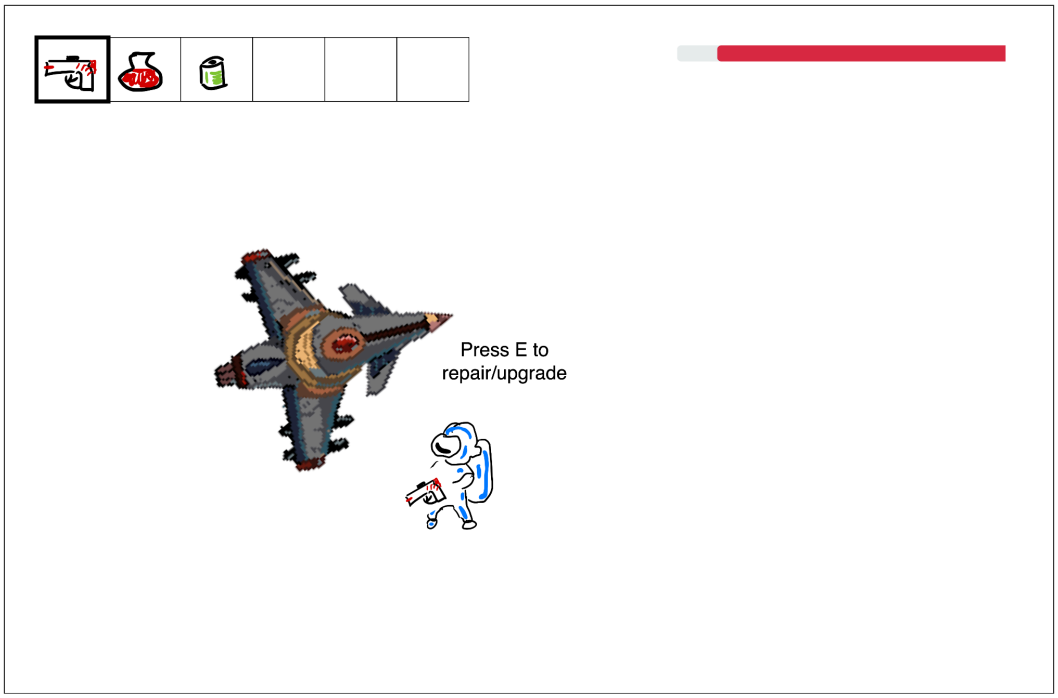
Sample boss fights in a biome:

Resources drop as the boss is killed:

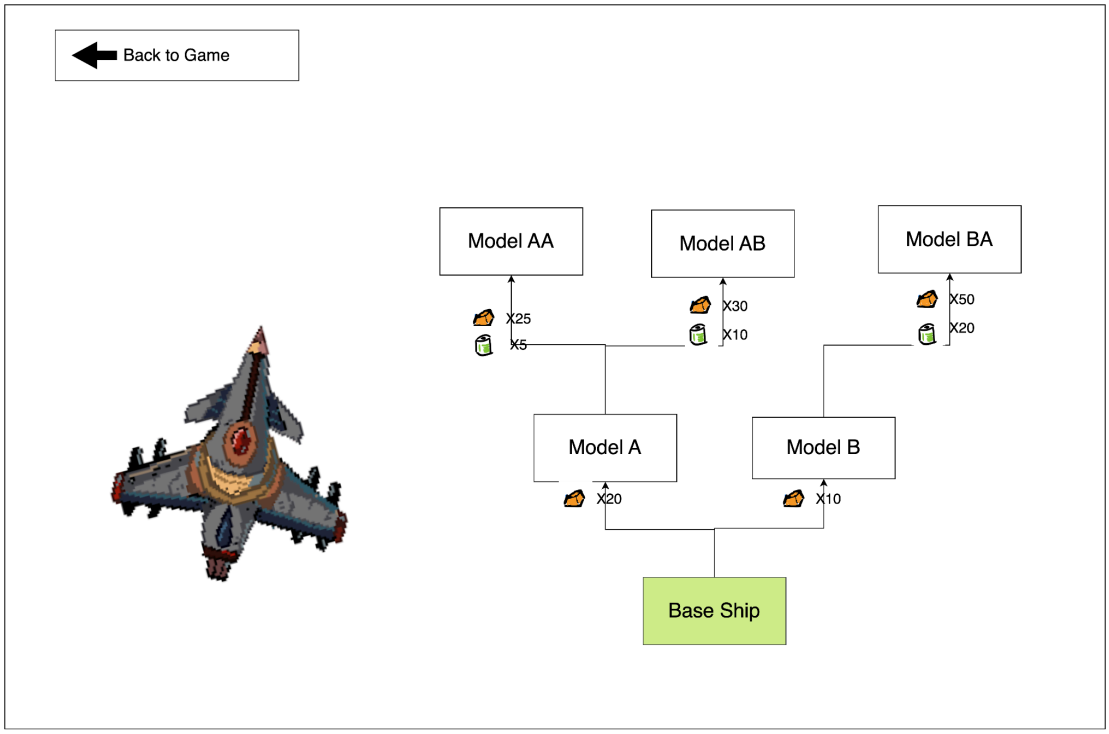


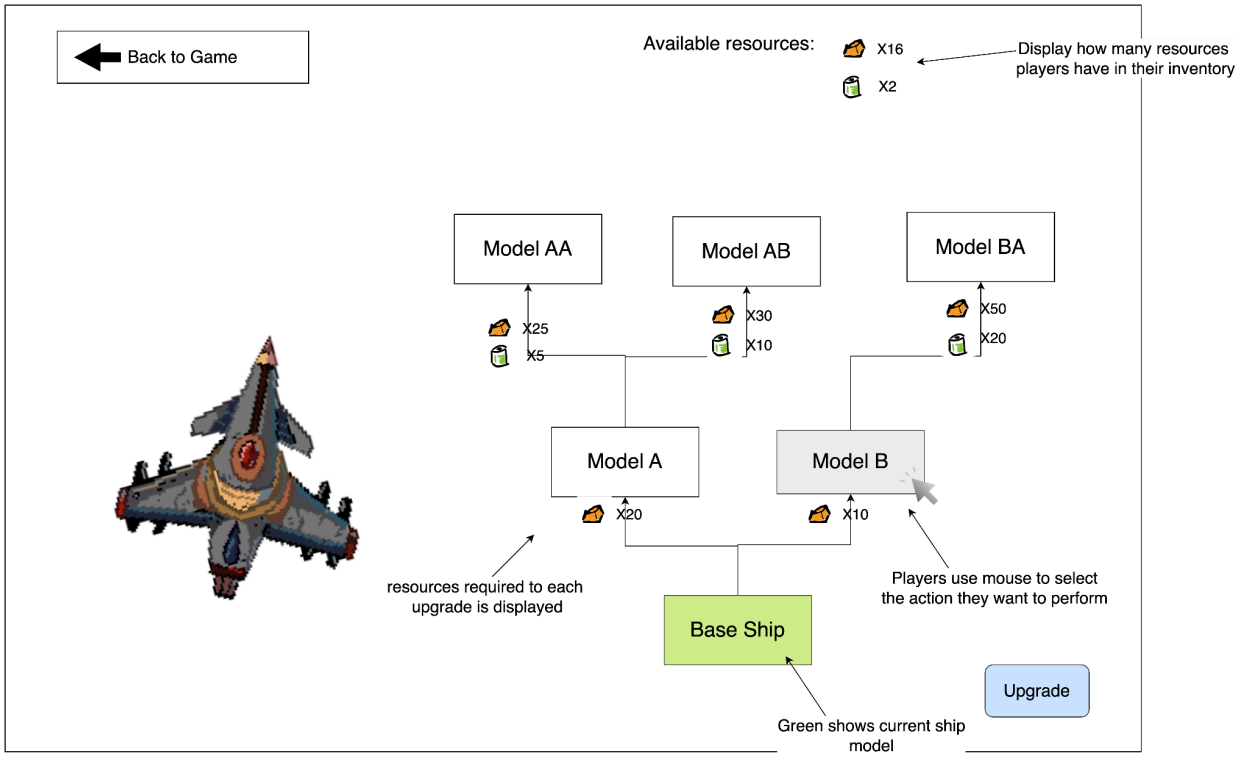
Ship Upgrading System:

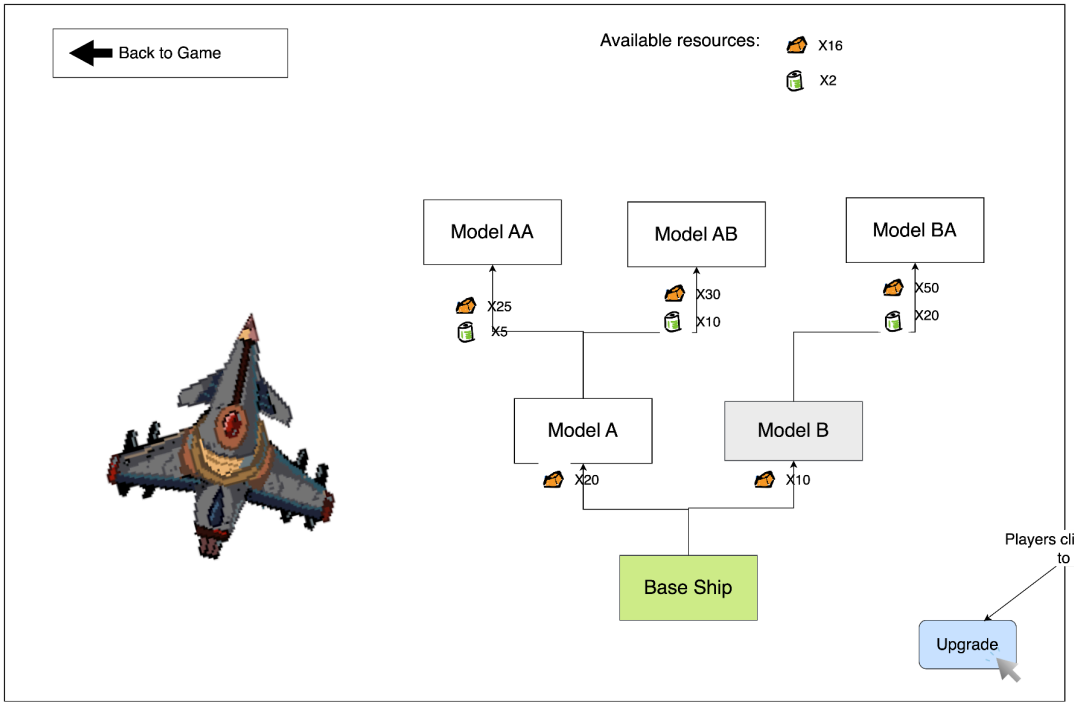
Players can press “E” to repair/upgrade their spaceship when they are near their ship:

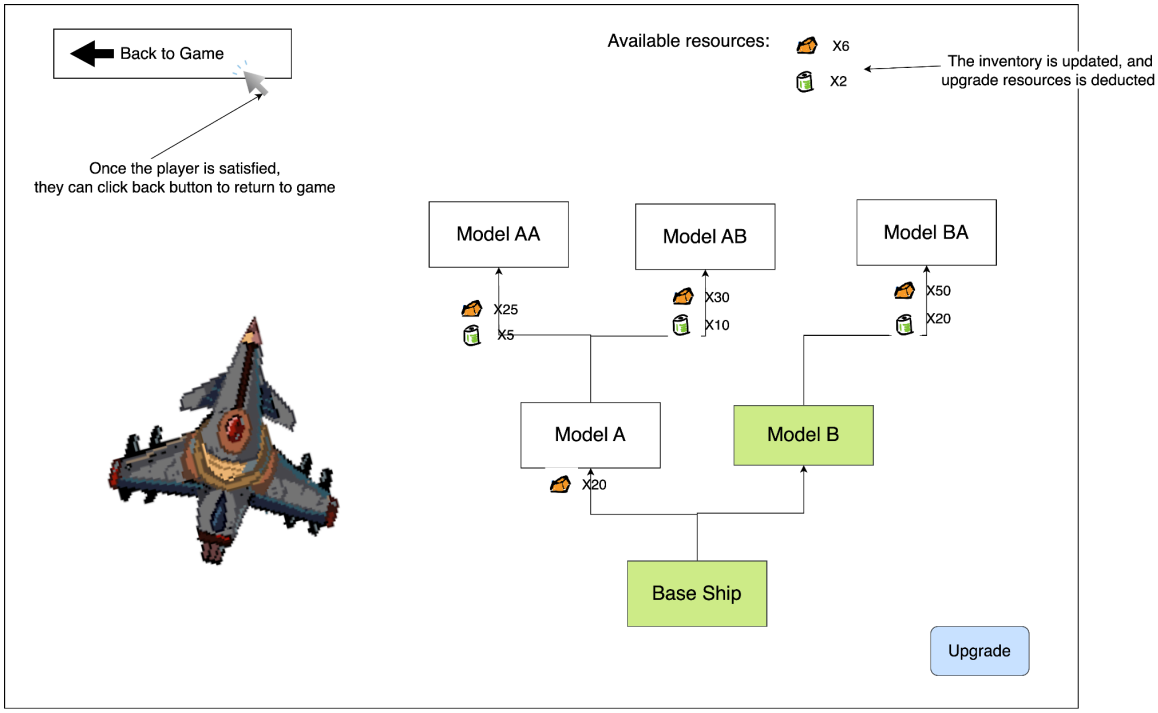


After pressing “E”, a ship upgrade UI will show up. Then player can select the action they want to perform by clicking:


Back to Game
Model AA
Model AB
Model BA
X25
X5
X30
X10
X50
X20
Model A
Model B
X20
X10
Base Ship


Back to Game
Available resources:
X16
X2
Display how many resources players have in their inventory
Model AA
Model AB
Model BA
X25
X5
X30
X10
X50
X20
Model A
Model B
X20
X10
resources required to each upgrade is displayed
Players use mouse to select the action they want to perform
Base Ship
Upgrade
Green shows current ship model


Back to Game
Available resources:
X16
X2
Model AA
Model AB
Model BA
X25
X5
X30
X10
X50
X20
Model A
Model B
X20
X10
Base Ship
Players click on upgrade button
to finish upgrade
Upgrade


Back to Game
Once the player is satisfied,
they can click back button to return to game
Available resources:
X6
X2
The inventory is updated, and
upgrade resources is deducted
Model AA
Model AB
Model BA
X25
X5
X30
X10
X50
X20
Model A
Model B
X20
Base Ship
Upgrade

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:

- Procedurally render tilesets for the floor and walls
- Lighting shaders for the day/night cycle and external light sources
- Ordered rendering sprite layers to maintain a pseudo-3D effect from a top-down perspective
- Animated sprites including idle, attack, and moving for the player and enemies

Sprites and Assets:

- Hand-drawn pixel art sprites for background, foreground, and other entities
- 16 color palette from Lospec
- If time is limited, use publicly licensed sprites from itch.io, reskinned by choosing the nearest neighbor color on the chosen palette

2D Geometry Manipulation:

- Collisions:
    - Players can collide with walls, structures, and enemies
    - Enemies inflict damage upon collision with the player
    - Projectiles and weapons collide with enemies to inflict damage

Gameplay Logic and AI:

- Enemy range and line-of-sight calculations to initiate pathfinding
- A* path-finding algorithm to move enemies towards the player
- Enemies move to be within attack range, meaning ranged enemies will station further away from the player.
- Enemies always move towards the player, because "kiting" enemies would be too difficult to defeat.
- Enemy difficulty (speed, health, and damage) scales with player status and progress

Physics:

- Player, enemy, and projectile movement and speed management
- Particle effects for weapon attack animations

Sound:

- General SFX
    - Menu interactions, the player taking damage, player inflicting damage, using a weapon, picking up items, enemy attacks and grunts
- Biome-specific background music

- If time is limited, use publicly licensed music from itch.io

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.

- Implementing swords into the combat system which would mainly, if we don't implement this we will still have a gun-play system which we plan to flesh out so the game wouldn't lose any core functionality.
- Implementing player movement abilities such as a dash. This would primarily serve to enhance the gameplay and allow for more engaging gunplay with enemies, an alternative we have is to make the gunplay balanced so no special movement is required.
- Boss-music? Ambient sounds/mixing given conditions (i.e. muted background music if it's raining or other scenarios?) Low priority prob, if not we'd likely use one or two soundtracks that are consistent throughout the game.
- Advanced visual effects?
    - Could be something like liquid shaders/physics (super low prio)
    - Weather effects on the player screen, rain, fog, etc, not implementing it would likely not have a large impact on any core mechanics/gameplay.
    - The alternative would be to just not implement these effects as they are mainly planned to be used for aesthetic purposes.
- Being able to go into the ship and have a bed, workbench, and storage system. (Alternative) In our game we likely will still have features similar to this however it will mainly be through a UI which a player can enter near the ship rather than physically separating the features and having a ship interior.

Devices:

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

The input devices we will use for our game are keyboards and mouse.

For Mouse:

- Mouse cursor placement controls player orientation: in our game, we use a mouse to control the direction in which the player is facing depending on where the mouse is pointing on the screen. When players are equipped with weapons, the mouse controls the direction of the aim of the weapons. That is the weapon is going fire in the direction of the cursor
- Left click controls in-game interactions: The left mouse click is used to use items currently active in players' selection. For example, if the player currently has a projectile

weapon selected, a left click will trigger the weapon to fire. If the player currently has a health potion selected, a left click will consume the selected potion.

For keyboard:

- WASD keys for controlling player movement: in our top-down game, we will use WASD to control the player movement for moving up, left, down, and right respectively.
- Number keys for selecting on-hand items: when the player wants to change their items on hand, they will use the number keys. Key 1 corresponds to selecting the first item in the player's inventory, key 2 corresponds to the second item in inventory, and so on.
- "E" key for controlling certain scene interactions: when the player is near their ship, the player can press the "E" key to upgrade their ship. By pressing the "E" key near the ship, a ship upgrade window will be displayed allowing the player to repair or upgrade their ship.

Tools:

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

- SFX and Background: We will use Beepbox.co to create digital background music for our game. We will also use Freesfx.co.uk to search for and use their sound effects (for example, button sound, enemy damage sound, weapon firing sound, etc.).
- Art assets: We will use Aseprite and MS Paint to create our own art assets. We might also use and adopt some fitting free assets from itch.io and kenney.nl/assets depending on our workload.
- Python automation: We might also develop Python scripts for asset reskinning and generation on the loyalty-free assets we found to make them fit our color palette and theme.

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 plan to track and assign tasks using a GitHub projects Kanban board
- As per our Teamwork Agreement, we will have sprint review meetings on Tuesdays at 1:00 PM
    - In these meetings, we will reevaluate requirements, write new user stories for tasks, and move tasks in the Kanban board
    - Our sprints will be one week long, and internal deadlines are on a single-sprint schedule
- Our de-facto task assignment is as follows:
    - Matthew: gameplay, management
    - Pranav: assets, engine

    - Frank: gameplay, UI/UX
    - Philip: engine
    - Steve: engine, UI/UX
- As deadlines and requirements may change, these responsibilities are subject to change

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).

Milestone 1: Skeletal Game

Week 1

Level Design + Rendering map and Player on screen. Create one soundtrack. Implement a skeleton of the ship system that the user can interact with (repair/upgrade system UI).

Week 2

Implementing a collision system with the player and environment.

Creating enemy sprites (no path-finding yet).

Creating a simple weapon that can damage an enemy and has a projectile visible on the screen.

Also creating a death mechanic for the player (can be tested by standing near the stationary enemy and getting slashed to death).

Milestone 2: Minimal Playability

Week 1 Implement resource/blueprint spawns on the map which can be used for the ship system skeleton.

Week 2 Implement path-finding for enemies. And create different classes on enemies that spawn in certain biomes, as well as night/day which impacts enemy spawns.

Milestone 3: Playability

Week 1

Implement enemy abilities (mini-bosses) guarding certain blueprints.

Create a gun upgrade system that is integrated with the ship skeleton.

Implement attachments that may change projectile physics (transform into shotgun with more AoE shorter range, transform into sniper which limits the rate of fire and increases distance and speed).

Have mini-bosses drop valuable resources that can be used for gun/ship upgrades.

Week 2

Implement and flesh out more of the level design, adding ruins, and more advanced structures.

Implement the ship mechanic which ends the game (leaving the planet), or create a choice for helping the aliens.

Create more difficult enemies found in more difficult regions of the game, so more HP, and faster attacks (essentially just the previous enemy classes but enhanced, and distinguishable in terms of design).

Create a dialogue system that can pause the game and serve the story.

Milestone 4: Final Game

Week 1

If any of us are free, start trying to add the advanced technical elements (in the order we have it now).

Flesh out game UI, and make the ship system prettier.

Tweak/balance the combat system if we find that anything is too difficult or too easy.

Flesh out the art, relating to tilesets used, enemy abilities, and projectile appearance.

Week 2

Keep fleshing out the overall aesthetic of the game, and working on any potential advanced technical elements.