Game Proposal: World Odyssey

CPSC 427 – Video Game Programming

Team: MI427

Manvinder 68393826

Deep Gill 20872941

Daman Sohi 24879397

Denis Kovalenko 36915940

Story:

All of a sudden, a powerful force has wreaked havoc upon the world, corrupting each biome and endangering harmony. As the chosen hero, you are provided with a mystical weapon and the responsibility to restore harmony. Go on an adventure through diverse biomes, battling corrupted creatures, and uncovering the dark secrets behind this danger.

Storyline:

The game begins in a forest biome, where corruption has already taken hold. Strange mutations and hostile creatures roam freely, threatening the ecosystem. As the hero, you receive a vision revealing that this corruption is spreading worldwide, and it's your destiny to stop it.

Act 1: The Forest Menace

- Navigate through dense forests, facing corrupted animals and battling a colossal corrupted tree as the first boss.
- Discover clues about the origin of the corruption and receive guidance to the next biome.

Act 2: Desert of Desolation

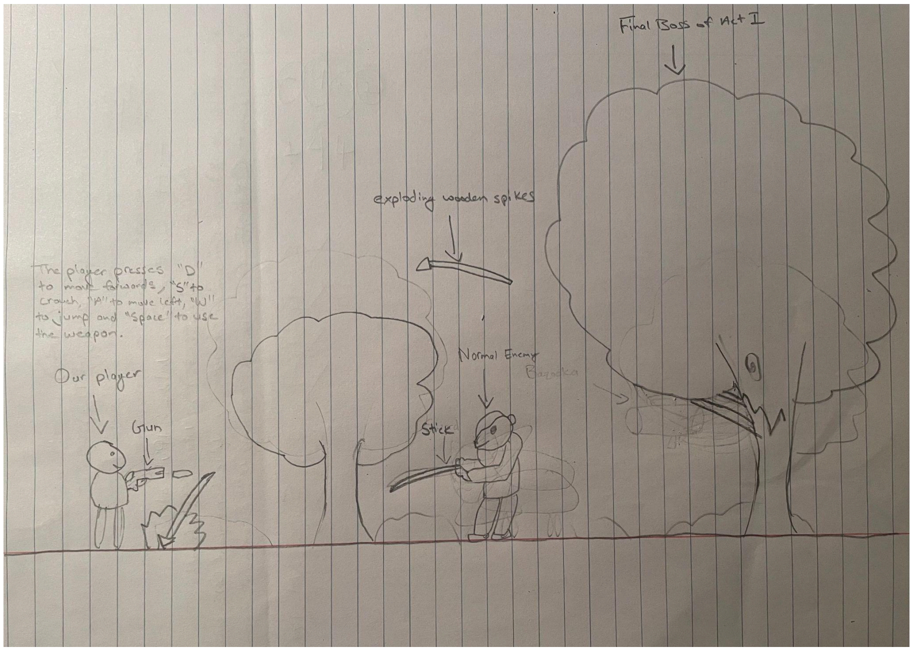
- Traverse a scorching desert biome filled with sandstorms and corrupted desert creatures.
- Confront a corrupted elemental being that controls the desert's destructive forces as the second boss.
- Uncover ancient ruins that hint at a forgotten civilization linked to the corruption.

Conclusion:
With the corruption eradicated, the hero becomes a symbol of hope and a guardian of the restored world. The game ends with the protagonist watching over the rejuvenated biomes, now thriving and coexisting in harmony once again.

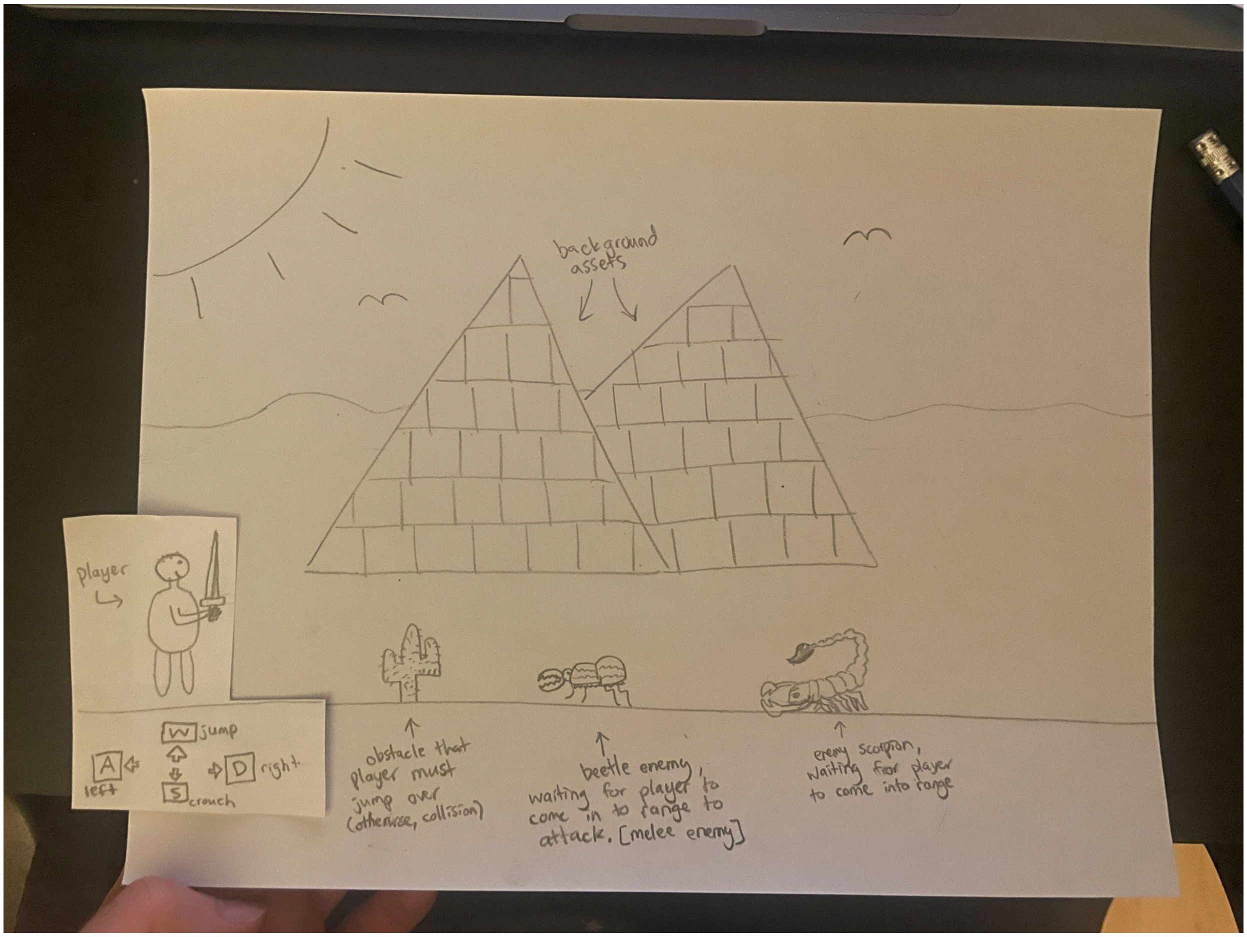
Scenes:

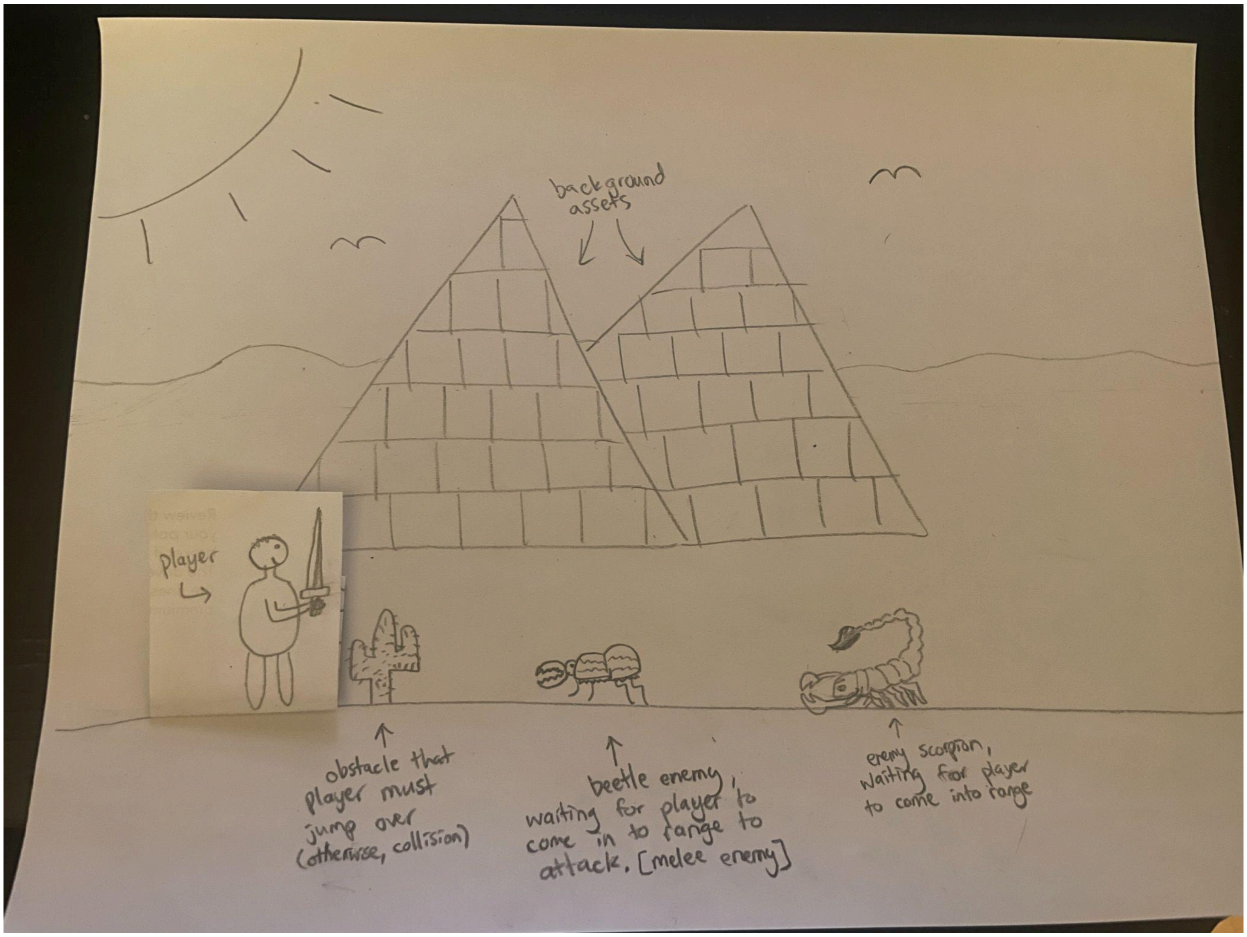
Produce basic, yet descriptive, sketches of the major game states (screens or scenes). These should be consistent with the game design elements, and help you assess the amount of work to be *done. These should clearly show how players will interact with the game and what the outcomes of their interactions 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.*

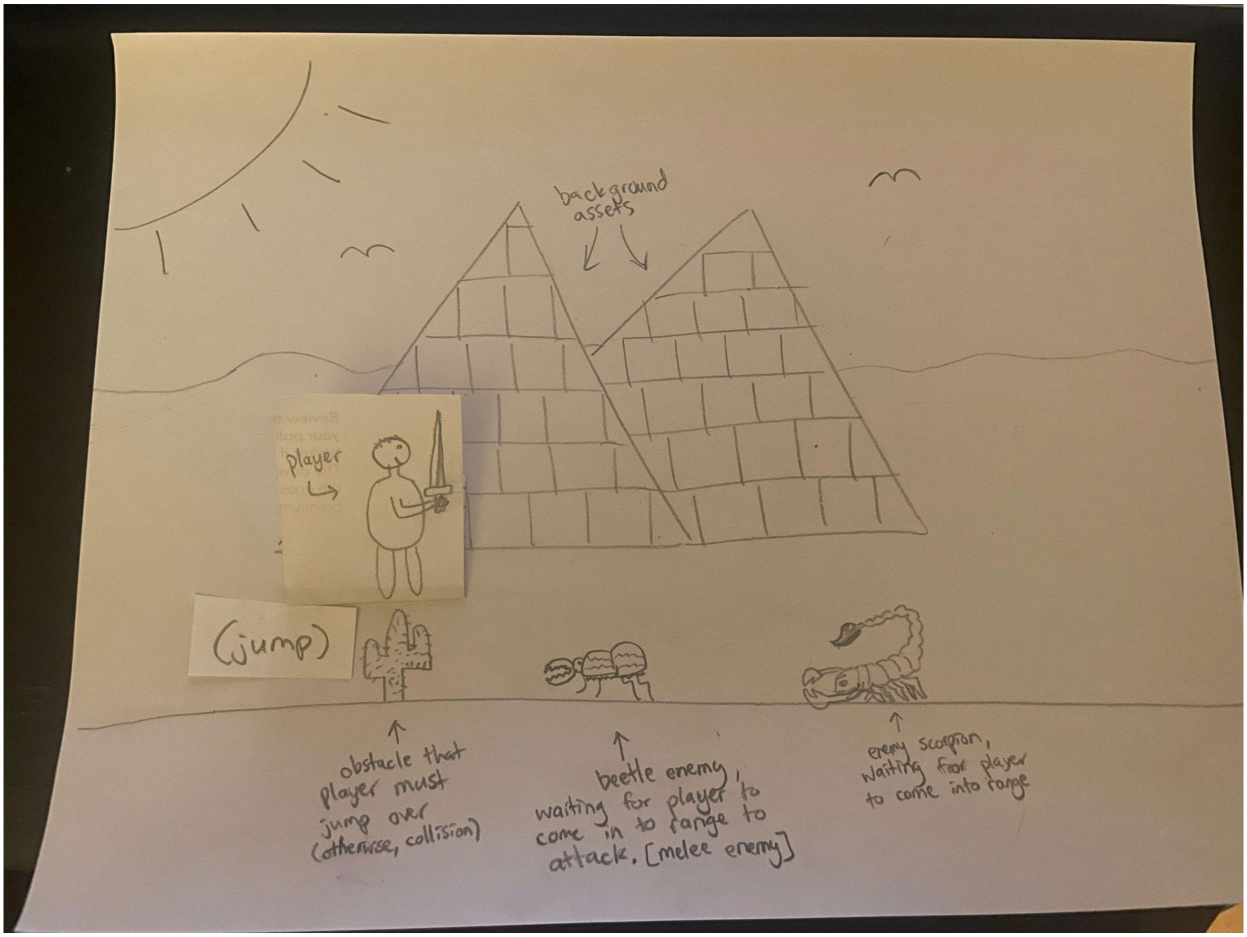
Forest scene:

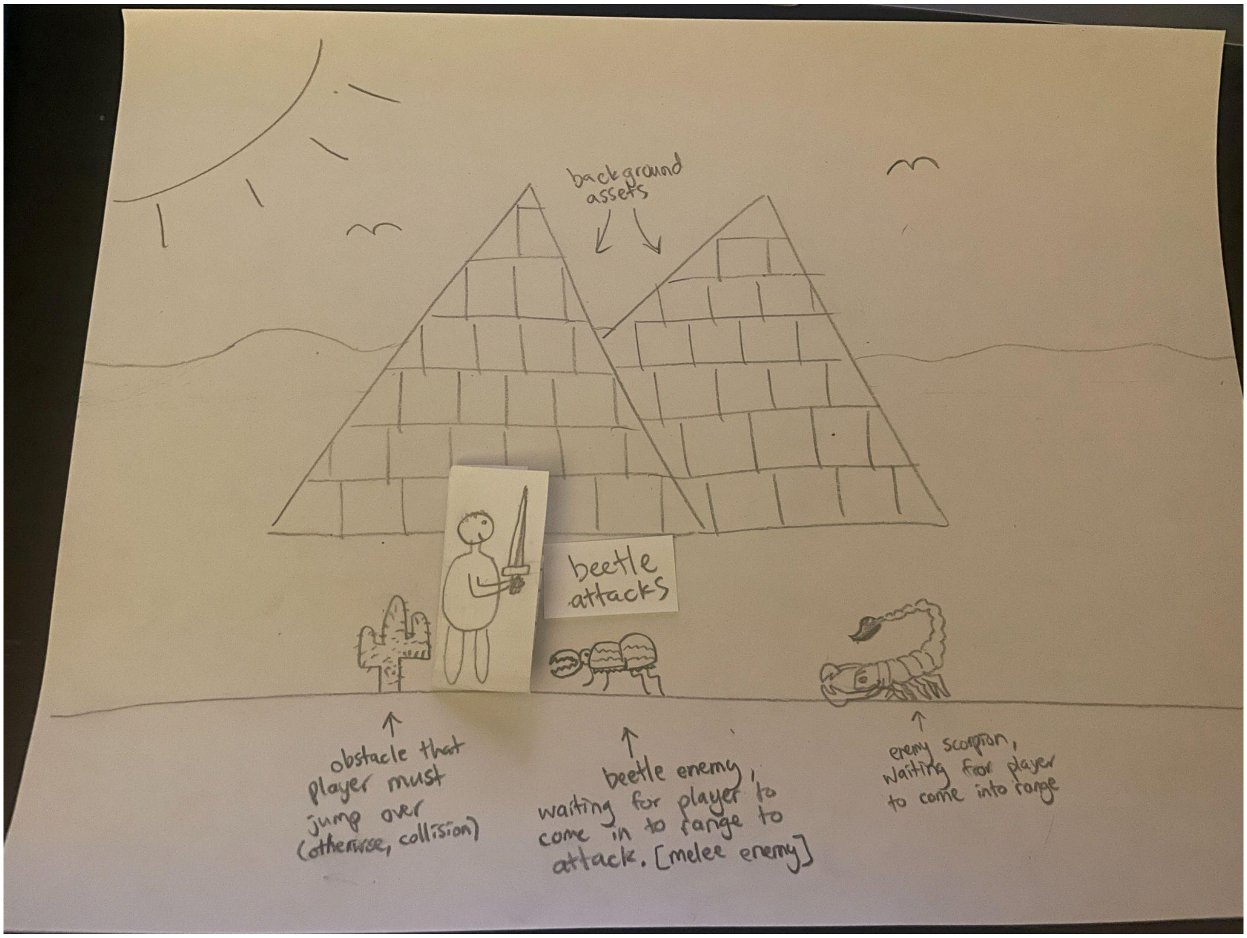

Final Boss of Act I
exploding wooden spikes
The player presses "D" to move forwards, "S" to crouch, "A" to move left, "W" to jump and "Space" to use the weapon.
Our player
Gun
Normal Enemy
Bazooka
Stick

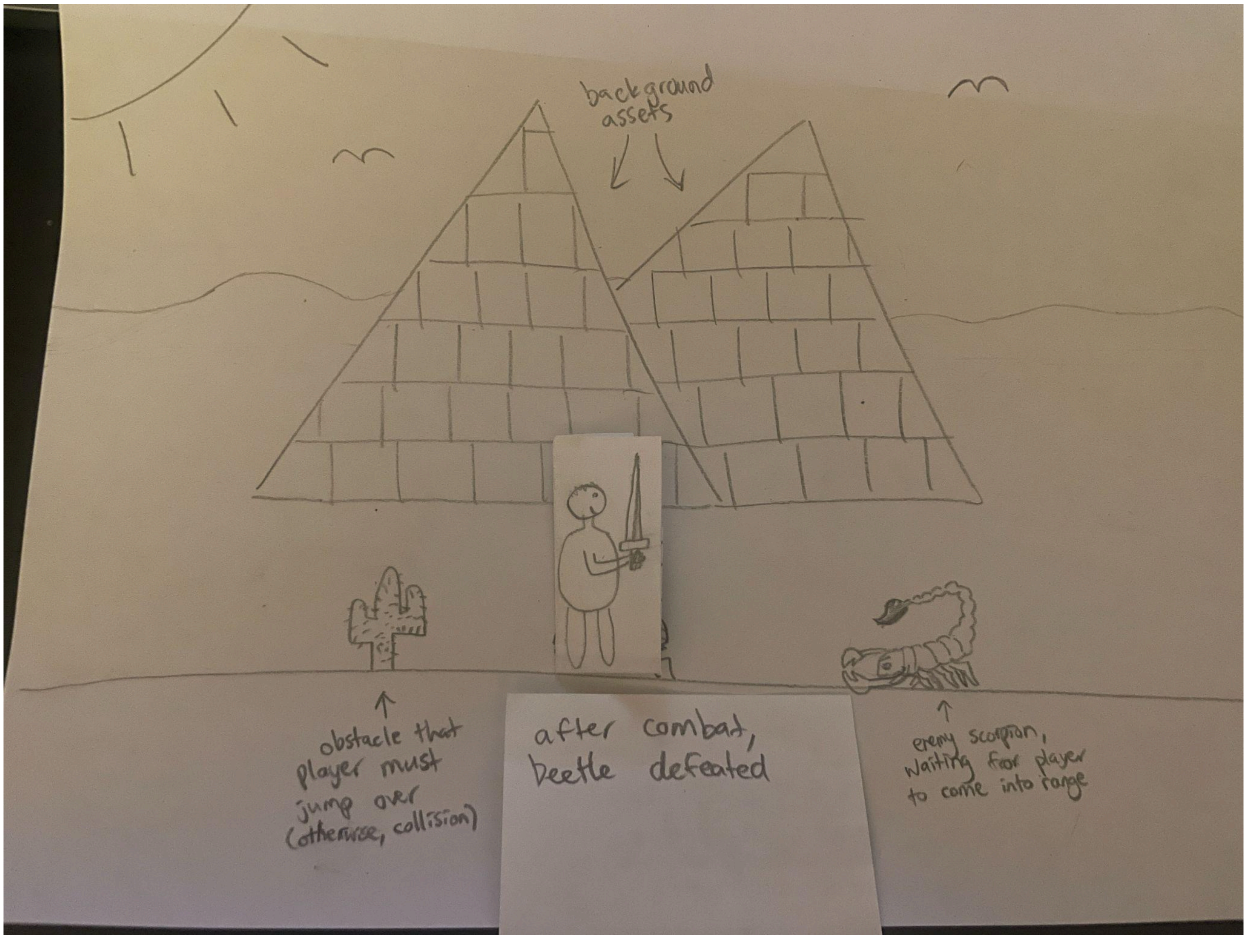
Desert scene:


background assets
player
W jump
A
left
S crouch
D right.
obstacle that player must jump over (otherwise, collision)
beetle enemy, waiting for player to come in to range to attack. [melee enemy]
enemy scorpion, waiting for player to come into range

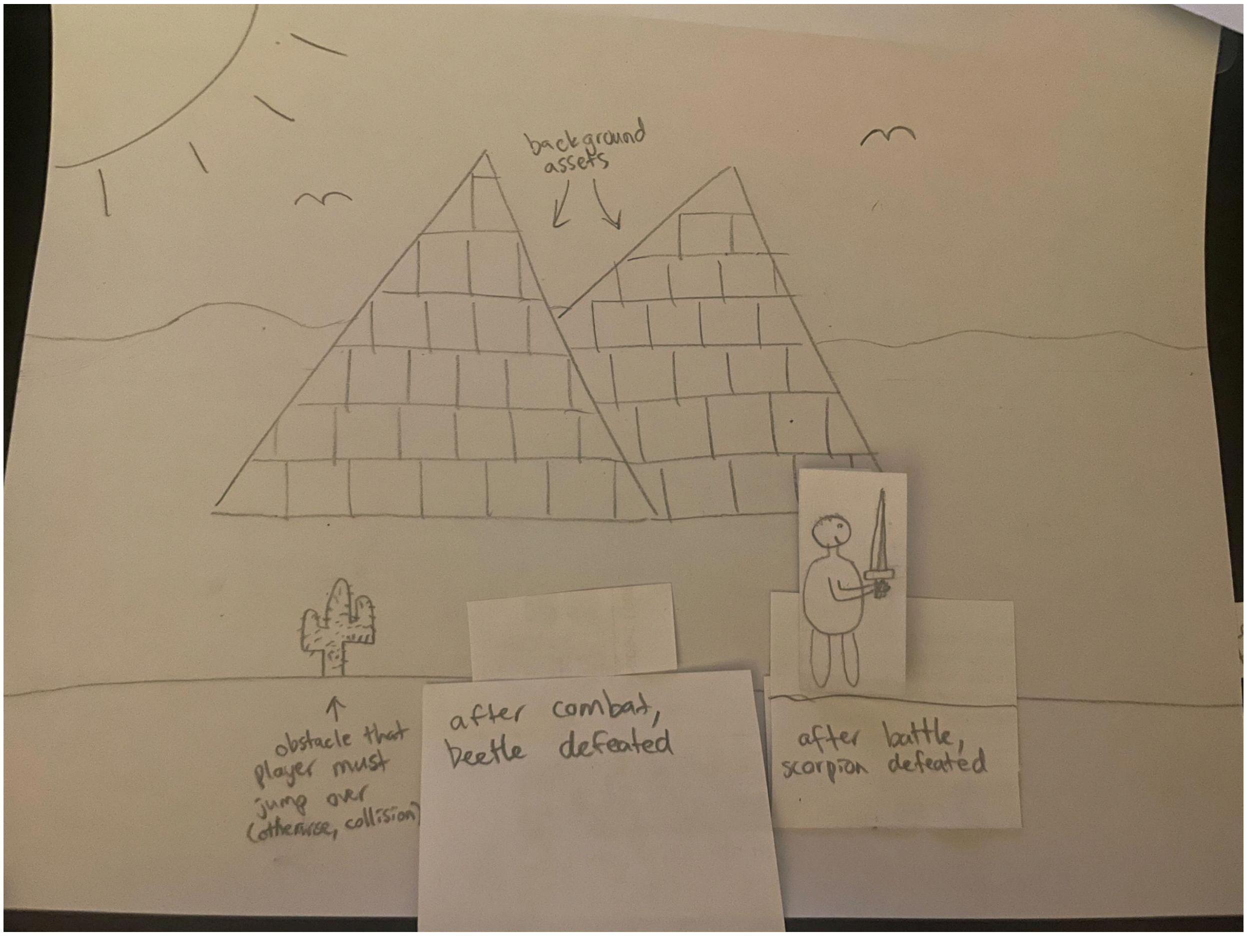

background assets
player
obstacle that player must jump over (otherwise, collision)
beetle enemy, waiting for player to come in to range to attack. [melee enemy]
enemy scorpion, waiting for player to come into range

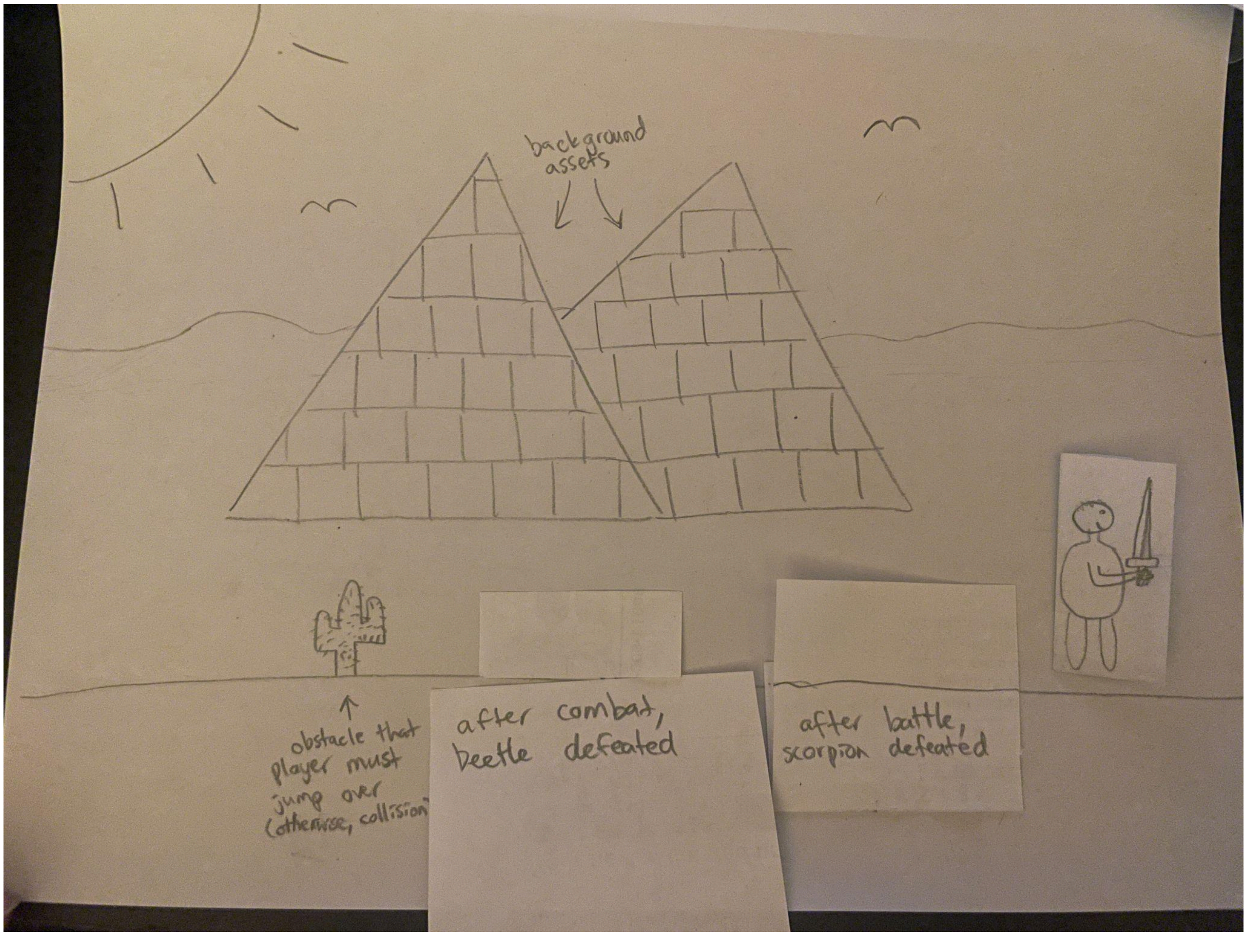

background assets
player
(jump)
obstacle that player must jump over (otherwise, collision)
beetle enemy, waiting for player to come in to range to attack. [melee enemy]
enemy scorpion, waiting for player to come into range

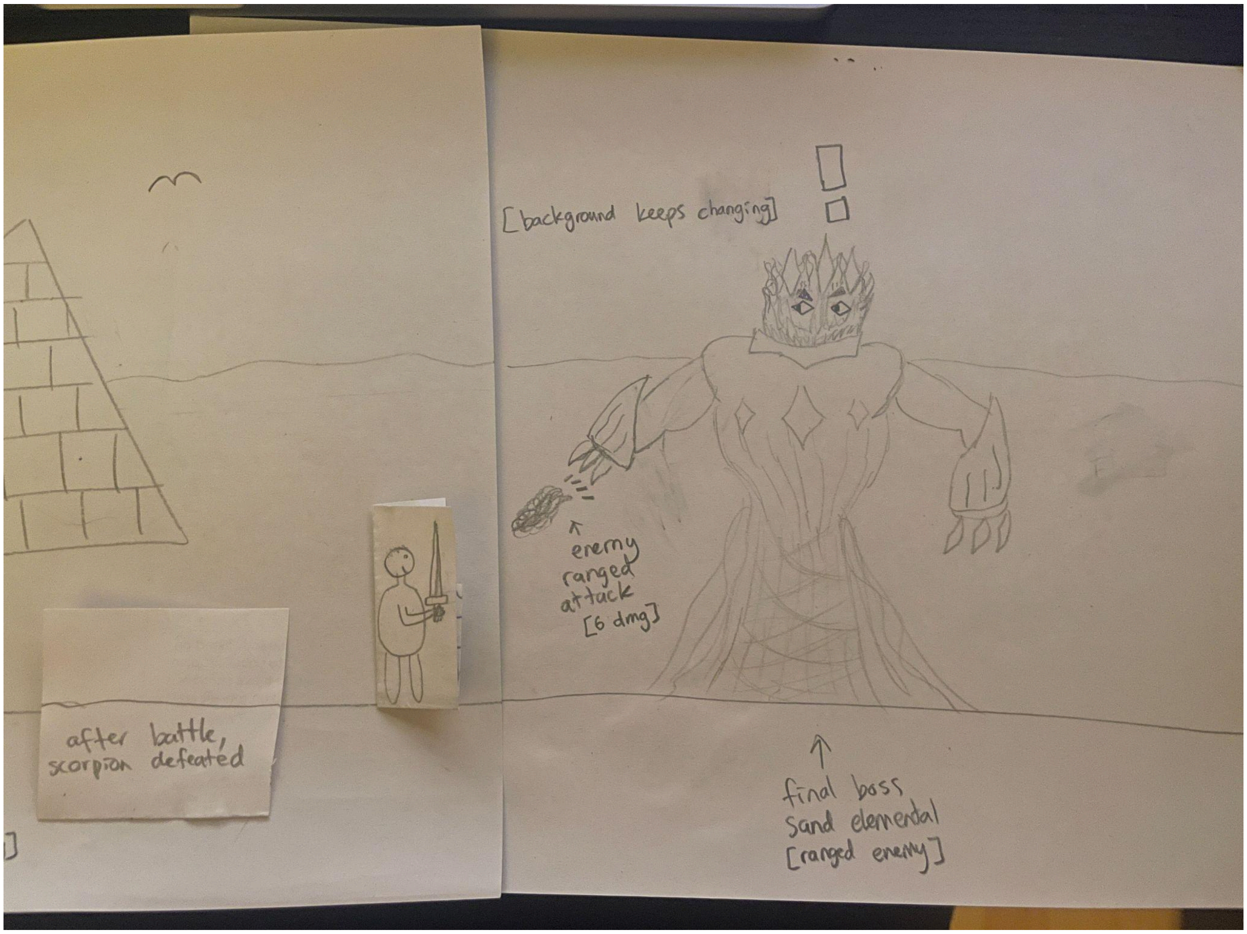

background assets
beetle attacks
obstacle that player must jump over (otherwise, collision)
beetle enemy, waiting for player to come in to range to attack. [melee enemy]
enemy scorpion, waiting for player to come into range


background assets
obstacle that player must jump over (otherwise, collision)
enemy scorpion, waiting for player to come into range

after combat, beetle defeated


background assets
obstacle that player must jump over (otherwise, collision)
after combat, beetle defeated
after battle, scorpion defeated


background assets
obstacle that player must jump over (otherwise, collision)
after combat, beetle defeated
after battle, scorpion defeated


[background keeps changing]
enemy ranged attack [6 dmg]
after battle, scorpion defeated
final boss sand elemental [ranged enemy]

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 (visual effects, parallaxing, etc.):

- The game environments will have parallaxing
- The combat will have several different visual effects, like sparks when your weapon hits enemies or they hit you and projectiles attacks that fly from enemies or the player
- OpenGL will be used to render

Assets (geometry, sprites, audio, etc.):

- Assets involve characters like the player character itself and the enemy characters, some small enemies and two bosses
- The different biomes will look different, the forest will have plants and trees, the desert will have sand dunes
- The weapon (sword with melee and ranged attacks), audio for attacks, collisions, etc.

2D Geometry Manipulation (transformation, collisions, etc.):

- Enemies disappear when killed by attacks

- Boss flinches when hit by attack and disappears when killed
- Background, creatures and boss change when in new biome
- Collisions with walls/obstacles which can't be passed through

Gameplay Logic (world interaction, character interaction, player controls, etc.):

- The player will travel through the 2D world moving from left to right, fighting enemies and overcoming obstacles
- Creatures chase and attack the player with melee and ranged projectile attacks
- Player can attack creatures to kill them, the enemy creatures will die in 1-5 hits depending on the creature
- Bosses will take more hits to defeat, at least 10
- Player can move and attack, their sword will start with melee attacks only and later they unlock ranged attacks where the sword can shoot energy projectiles which travel straight and damage enemies
- There may be obstacles or walls in the biomes which the player must jump over or they have to defeat the enemies in the area to make the wall come down

AI (pathfinding, entity behavior, etc.):

- Pathfinding algorithm to get the creatures to chase the player when they are nearby
- Entity behavior to attack the player once close enough, some enemies will only have melee attacks, others will also have ranged attacks

Physics (entity interactions, particle effects, kinematics, etc.):

- Entities can damage each other with attacks, melee or ranged
- Attacks will cause particle effects, like sparks
- Player can jump and move left and right, enemies will run on the ground
- There will be collisions with obstacles, walls, or enemies

Sound (effects, music, etc.):

- Use non-copyrighted sound effects such as background music
- Player collision sound effects and player weapon attacks sound effects
- Sound effects when enemies or player are hit and damaged by attacks
- Background music changes per level
- Play audio when player passes a level

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.

- The player character can fly around as part of their movement. The impact of skipping this would be that the player has to be on the ground the entire time. An alternative would be that the player can hold the jump key (W) to jump higher than normal, giving them another way to be in the air for a few seconds.
- Enemies can fly around as part of their movement while attacking the player. The impact of skipping this would be that enemies are stuck on the ground the entire time. An alternative

would be enemies that are in the air in fixed positions or which travel a fixed path (rather than flying freely) and can attack the player with ranged attacks from these positions.

- The player character can have different weapons, like a gun or bombs. The impact of skipping this would be that the player can only use the sword as their weapon. An alternative is to implement ranged attacks that the sword can do which mimic bullets and bombs

Devices:

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

We plan on supporting keyboard controls. The keys W, A, S, and D would be used to move the player forwards (D), backwards (A), jump (W) and crouch (S). The key SPACE would be used to use the weapon and P would be used to pause the game. If flying is implemented, hold W to start flying and you would move around with W, A, S, and D in the air. Use 1, 2, and 3 to change weapons if multiple weapons are added.

Tools:

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

- **Box2D** for the Physics Engine
- **FMOD** for the audio

Team management:

We will be utilizing the GitHub Project board to create tasks. Then, each task will have an assigned member with a short description of the task at hand. We will utilize the status column of the project board to track whether the task is yet to be started, in progress, or completed. Once tasks have been finished we will move forward with assigning and completing further tasks until all tasks on the project board have been completed. We will have weekly recurring check-ins to ensure everyone is on the right path and whether they need some adjusting or assistance.

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

- Render the background of the game, the player, one enemy, a boss and a weapon.
- Add keyboard controls that change the position of the player and fire weapons (likely without allowing the player to move past the borders of the initial “frame”, ie. a static background for now.)

Week 2

- Allow player to move past borders of initial “frame”, ie. dynamic background
- Add collision detection between player and weapon, player and enemy and enemy and weapon.
- Add boundary detection
- Create Test plan and bug list

Milestone 2: Minimal Playability

Week 1

- Render more enemies, weapons and other details
- Add the AI that controls enemy movements and enemy weapon firing

Week 2

- Add the gameplay logic that determines when the game is lost, game is won, the player is dead, an enemy is dead, taking damage, etc

Milestone 3: Playability

Week 1

- Add audio to the game, add character battle movement (swinging sword, taking damage, etc)

Week 2

- Add more enemies, different biome(s) with different background and enemy visuals
- Alternatives: add new enemies that are the same as old ones but look different or have different health, damage, etc.

Milestone 4: Final Game

Week 1:

- Equipping new items. Add any final enemies, weapons, character interactions
- Alternatives: no new items, instead the sword can do new kinds of attacks

Week 2

- Polishing game, fixing any bugs, playtesting