Game Proposal: Aria

CPSC 427 - Video Game Programming

Team Members:

Jason Nguyen 11170081
Avinav Dhakal 81919490
Frank Liu 69162840
Leo Wang 18416891
Elizabeth McDonald 50906841
Lallit Narang 47975511

Story:

In a mystical realm known as "Aetheria," an ancient prophecy foretells of a legendary hero who will rise to protect the land from the encroaching darkness. You step into the shoes of the chosen hero, a young mage named Aria, who is gifted with an extraordinary power to manipulate matter. The game begins with Aria's awakening in her remote village, where she accidentally discovers her power to control nearby matter, turning it into elemental projectiles. She is visited by a lost soul of a wise elder who reveals her true identity as the prophesized hero. The lost soul offers to help Aria develop her powers and fulfill her destiny in exchange for helping find the elusive and ancient Soulstone to bring itself back to life. Along their journey, Aria faces not only the twisted monstrosities lurking in the depths of Aetheria's dungeons but also her own inner demons, as the line between good and evil blurs in this eerie, otherworldly adventure. In this dungeon crawler, players will be able to strategically defeat their foes with the use of elemental powers that they gain along their quest for the Soulstone. Every dungeon floor presents a new opportunity for exploration, gaining new abilities, and/or fighting waves of enemies. All of this leads up to the final boss who you must defeat in order to win the game.

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

- OpenGL
- 2D 8-bit style - Top down game occurring within a 2D plane

Assets

- Collecting assets from open source databases. (e.g. freesounds.org)
- Creating custom sprites

2D geometry manipulation

- Enemy collision with the player will cause the player to take damage.
- Players cannot move past certain boundaries like walls for example.

Gameplay Logic

- Enemies

    - Enemy AI will vary based on the type of enemy, some will randomly move within a set boundary and others will react to the player. Enemies damage the player upon collision.
    - There will be a couple different types of enemies, each with their own weakness to certain elements. There will be visual hints to allow the player to intuit what a given enemy's weakness is.
    - Mini-bosses and bosses can attack with projectiles.
    - Enemies will have HP which can only go down if the player damages it.
- Player
    - The Player can move around with WASD and shoot different elemental projectiles with J, K, L, etc.
    - The Player will have a set amount of HP which goes down upon collision with enemies or from being hit by projectiles.

Physics

- Basic kinematic physics for player/enemy movement.
- Projectiles experience friction as it travels through the air (slows down). Projectiles diminish when hitting a wall.

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.

Unique weapons

- The player can choose to use certain elements to defeat their enemies. The effectiveness of the elements depends on the enemy.

Audio Feedback

- Opening doors, taking damage, slaying monsters will make noise.

Randomness

- Mini-bosses and bosses will have cycling weaknesses to different elements at 'random' time intervals.

Dungeon Levels and Rooms

- Dungeon rooms make up an entire dungeon floor. Upon clearing all the rooms, the player must defeat the dungeon floor boss. Once all dungeon floor bosses are defeated, the player must defeat the final boss.

Damage Knockback

- Player and enemies who are damaged will be slightly knocked back in position.

Devices:

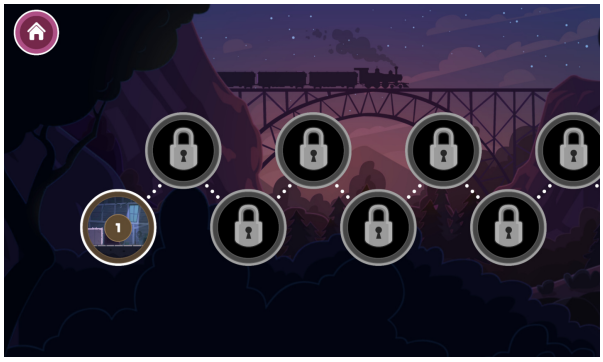
We will support keyboard and mouse (or trackpad). For example, the WASD keys will be used to move the character, and the cursor will be used to navigate in-game.

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.

Main Menu

Level Select



Using rock projectile against ghost

Using fire projectile against stump

Fighting boss with cycling damage weakness

Selecting power ups

Tools:

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

ChatGPT (for ideas)
Adobe Photoshop
Audacity
Freesound.org

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 have a kanban board.
    - Weekly meetings => Demo day
    - Tasks will be distributed weekly ensuring that everyone has something to do for the week.
    - GitHub Projects: https://github.students.cs.ubc.ca/orgs/CPSC427-2023W-T1/projects/18

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 (Sept 22 - Sept 28)

- Begin drafting sprite assets (player, enemies, map).
- Start README.md.
- Think of ECS design pattern to use.
    - Figure out all components and under which systems should the components belong.
- Get idea of controls: keyboard/mouse inputs to use.

Week 2 (Sept 29 - Oct 5)

- Begin drafting sprite assets (player, enemies, map).
- Gray-box completed
- Figure dungeon room layouts.
- Implement visuals for dungeon rooms and render rooms with the player in the room.
- Keyboard control for the player including both movement and shooting a basic projectile.
- Implement a few basic enemies with AI movement.
- Orientation of characters (player/enemies) change orientation when moving.
- Collision with walls.

Week 3 (Oct 6 - Oct 12; Milestone 1 due on Oct 12)

- Start finalizing sprite assets.
- Ensure major bugs are fixed.
- Ensure the game is stable (play testing).
- Check for performance (lag, stuttering, etc.).
- Creative and rendering component:
    - player character has a sprite
    - walls and floors of first level have textures
    - elemental projectiles are distinguished by color
    - enemies still look like simple shape
- Game should be playable by the following criteria without stutter/lag:
    - Moving around smoothly and orientation updates when moving up/down and left/right
    - Collision detection between players and walls, and players and enemies.
    - Collision detection between projectile and enemies.

Minimal Playability

Week 1 (Oct 13 - Oct 19)

- Start decision tree implementation for enemy entities.
    - Enemy follows player if player is within a certain range of that enemy.
- Draft sprites for enemies and bosses.
    - First element levels.
- Draft UI sprites.
    - Main menu, level selection, health bar, etc.
- Draft Player sprite animations:
    - Shooting
    - Walking
- Start implementing a tutorial screen.
- Player should be able to kill enemies now.
- Add at least one or more different rooms and introduce new elemental mechanics.

Week 2 (Oct 20 - Oct 26; Milestone 2 due Oct 30)

- Finish player sprite animations.
- Add the first power-up selection that the player can choose.
- Finish UI sprites.
- Add weaknesses to elements for enemies.
- Incorporate the use of mana when using projectiles and implement the regeneration of mana overtime.
- Add audio feedback to firing projectiles, opening doors on level completion, taking damage from enemies, and killing enemies.
- Play-test to test stability and ensure no lag or stutter.

Playability

Week 1 (Oct 31 - Nov 6)

- Implement new levels until the 5 minute mark.
- Implement new elements for the levels.
- Implement mini-bosses.
- Begin implementation of save/load feature (creative).
- Implement complex motion on main menu ui elements (creative).

Week 2 (Nov 7 - Nov 13)

- Play test new levels for stability.
- Ensure robustness by checking for memory leaks or memory hoards by leaving the game running for a while.
- Add knockback to enemies (physics creative).

Week 3 (Nov 14 - Nov 20; Milestone 3 due Nov 20)

- Complete implementation for save/load feature.
- Continue testing for stability especially for new features.
- Flickering torch light that changes shadow orientation.
- Begin implementation for final levels.

Final Game

Week 1 (Nov 21 - Nov 27)

- Identify and fix bugs from prior milestones.
- Implement a tutorial for introduction to game mechanics.
    - Intuitive visual indicators
- Begin user feedback process.
- Finish implementation for final levels for the 10 minute mark.
- Create and add audio components to game entities (creative).
    - Add background music.
    - Projectile sound effects.
    - etc.
- Add 2D dynamic shadows (creative).
    - Fire shadow
    - Projectile shadows

Week 2 (Nov 28 - Dec 4; Milestone 4 due Dec 4)

- Implement changes from user feedback.
- Test game stability.