**Defense Documentation for Lost Plateau**

**Statement of Purpose**

The purpose of this project is to create an engaging and accessible video game that balances enjoyment with ease of access. The gaming industry often requires significant financial or time investments, limiting accessibility and contributing to boredom. This project addresses this by developing Lost Plateau, a horror casual game with rogue-lite elements. Research highlights the cognitive and mental health benefits of video gaming, emphasizing its positive impact on well-being (Godman; Natl Inst of Health). The game will provide a fun, low-barrier experience that enhances well-being while avoiding common industry pitfalls such as data collection and paywalls.

**Research & Background**

The concept of Lost Plateau is grounded in research indicating the mental health benefits of hobbies, including video games. Studies have shown that games improve cognitive skills, impulse control, and working memory, particularly in younger audiences. Drawing inspiration from Vampire Survivors, Lost Plateau aims to replicate its addictive, progression-based gameplay while introducing unique elements through the Godot engine. The rogue-lite genre, characterized by procedural generation and permanent progression mechanics, ensures replayability and continuous engagement.

**Project Language(s), Software, and Hardware**

Languages: GDScript, C#

Software: Godot Engine (version 3.5 or later), Git for version control

Hardware: Desktop PC with at least Intel i5 processor, 8GB RAM, dedicated GPU with 2GB VRAM, 10GB of storage

Operating Systems: Windows 10, Windows 11

**Project Requirements**

* Player Movement: The player can navigate using W, A, S, D keys.
* Enemy Spawning: Enemies spawn outside the player's viewbox.
* Enemy Movement: Enemies move toward the player every frame.
* Health Loss: Player loses health upon contact with enemies.
* Health Bar: Health bar appears and updates when the player takes damage.
* Collision Detection: Characters cannot occupy the same space.
* Skill Activation: Player periodically launches equipped skills.
* Enemy Damage: Enemies lose health when hit by projectiles.
* Knockback: Enemies experience knockback when hit by certain projectiles.
* Projectile Despawn: Projectiles despawn after missing or leaving the screen.
* Enemy Despawn: Enemies despawn when their health reaches zero.
* Gem Drops: Enemies drop gems upon despawn.
* Gem Collection: Player can collect gems by coming into contact with them.
* Gem Despawn: Gems despawn after being collected.
* Progress Bar: Updates as the player collects gems.
* Upgrade System: After collecting enough gems, an upgrade window appears.
* Equip Upgrades: Selected upgrades are applied to the player character.
* Increasing Difficulty: More enemies spawn as the game progresses.
* Return to Menu: A button appears to return to the menu upon player death.
* Victory Screen: Displays if the player survives for five minutes.
* Privacy: No user data is collected, and inputs are restricted to movement keys and mouse clicks.

**Project Implementation Description & Explanation**

Upon launching Lost Plateau, players are greeted with a main menu offering options to start the game or quit (Fig 1. The main menu). Starting the game transitions the player to a procedurally generated grasslands environment where kobolds aggressively pursue the player(Fig 1. The grasslands). Movement is controlled using the W, A, S, D keys, and the player automatically launches equipped skills periodically to fend off enemies.

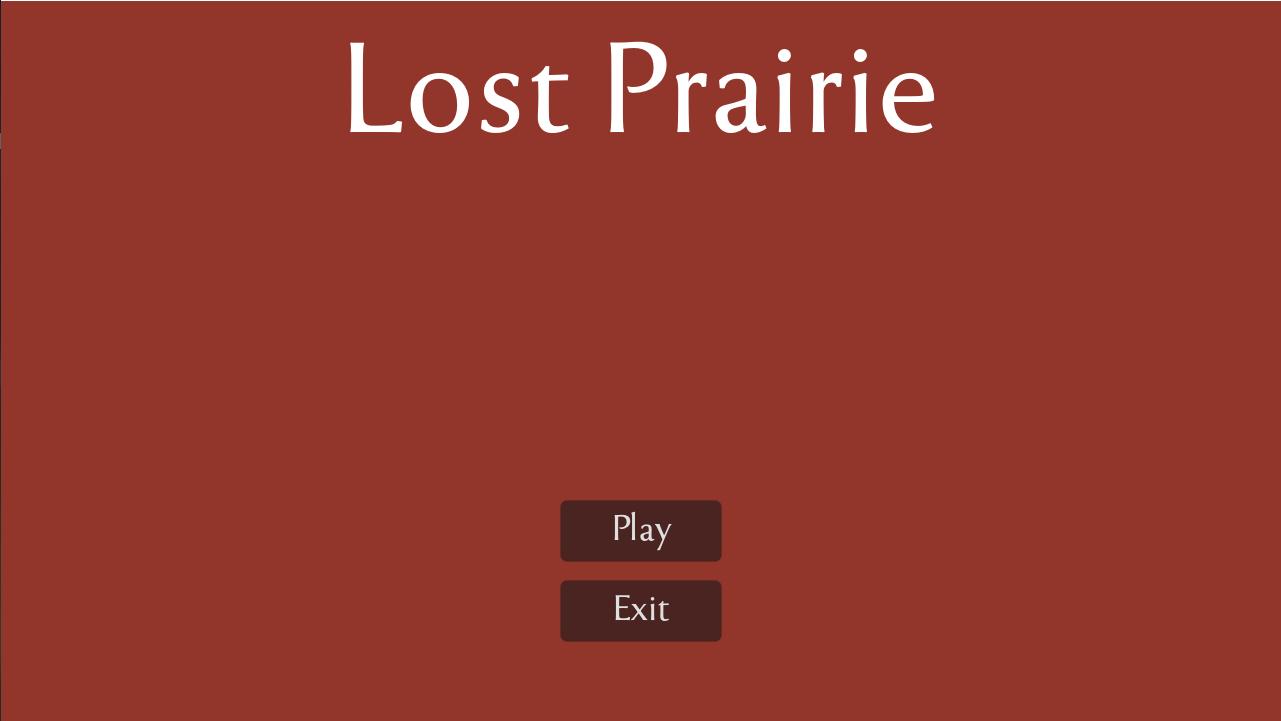


Fig 1



Fig 2

As players defeat kobolds, they drop gems, which the player collects to fill a progress bar. Once enough gems are collected, an upgrade window appears, allowing players to select enhancements to better survive the increasing onslaught of enemies (Fig 3. The upgrade screen). Enemies spawn at random locations outside the player's immediate view, ensuring immersion. If the player survives for five minutes, a victory screen is displayed; otherwise, death triggers a return-to-menu button (Fig 4. The end game screen).



Fig 3



Fig 4

The source code for Lost Plateau is available in the following repository: https://github.com/MackWessels/Senior-Project

**Test Plan**

The test plan outlines strategies to ensure the game meets usability, performance, and functional requirements. Testing focuses on:

Player movement responsiveness.

Enemy spawning behavior and movement.

Collision detection between characters and projectiles.

Health system functionality and UI responsiveness.

Performance scalability under stress conditions.

| **Test Case** | **Result** |
| --- | --- |
| Verify Player Movement with Directional Keys: The player moves correctly using W, A, S, D keys. | Player character responds accurately and smoothly to directional inputs. |
| Verify Enemies Spawn Outside Player's View: Enemies should spawn outside the immediate viewbox. | Enemies consistently spawn outside of view, maintaining immersion. |
| Ensure Enemies Move Toward Player Each Frame: Enemies should pursue the player continuously. | Enemies track and move toward the player in real-time. |
| Ensure Player Loses Health Upon Contact with Enemy: Player health should decrease on enemy contact. | Player loses health when hit, and health bar updates dynamically. |
| Verify Health Bar Appears on Health Loss: Health bar should become visible and update as needed. | Health bar appears and reflects accurate health loss. |
| Ensure Characters Can't Occupy Same Space: Collision detection prevents overlap. | Proper collision detection prevents overlap of characters. |
| Verify Automatic Skill Launching: Player's skills should activate periodically. | Skills auto-activate at regular intervals without manual input. |
| Ensure Enemies Lose Health When Hit by Projectiles: Projectile hits reduce enemy health. | Enemies take damage and display health reduction. |
| Validate Enemy Knockback on Projectile Hit: Enemies should be pushed back when hit. | Knockback effect successfully applied on hit. |
| Ensure Projectiles Despawn Off-Screen or on Miss: Projectiles should not persist indefinitely. | Projectiles despawn correctly. |
| Ensure Enemies Despawn on Zero Health: Enemies should disappear when defeated. | Enemies despawn as expected upon reaching zero health. |
| Ensure Enemies Drop Gems on Despawn: Defeated enemies should drop gems. | Gems consistently dropped on enemy death. |
| Ensure Player Can Pick Up Gems: Gems collected when touched. | Gems collected correctly when player touches them. |
| Ensure Collected Gems Despawn from Game World: Gems should disappear after pickup. | Gems despawn after collection. |
| Verify Progress Bar Updates on Gem Collection: Progress should reflect collected gems. | Progress bar updates properly. |
| Ensure Upgrade Window Appears After Gem Threshold: Upgrades shown after enough gems. | Upgrade window triggers appropriately. |
| Validate Equipped Upgrade Functionality: Chosen upgrade affects gameplay. | Upgrades applied correctly to player. |
| Ensure Enemy Count Increases as Game Progresses: Enemy spawns should scale over time. | Enemy numbers increase logically. |
| Check for Return-to-Menu Button on Death: Player can return to menu after dying. | Button appears on player death. |

All tests met their expected outcomes, and no significant performance issues, bugs, or unexpected behaviors were encountered during the testing phase.

**Professional Practices**

Throughout the project, professional tools and methodologies were applied to ensure an efficient development process:

Industry Tools: The game was developed using the Godot Engine, with GitHub for version control, and various debugging utilities to track and fix issues.

Project Management: The project followed an iterative workflow, ensuring tasks were completed in a structured and timely manner. Milestones included setting up core mechanics, implementing gameplay features, testing, and refinement.

Skill Development: This project allowed for hands-on experience in procedural game design, AI behavior scripting, and performance optimization, all of which were essential for a successful implementation.

**Challenges Overcome**

One major challenge was learning the Godot engine, as it differed significantly from prior experience with other game development tools. Adapting to its workflow required extensive research and practice. Another difficulty was implementing procedural generation while maintaining balanced gameplay. Ensuring that enemy placement and movement remained fair yet challenging required iterative adjustments. Additionally, performance optimization posed a challenge as managing large numbers of entities like enemies and projectiles had to be carefully controlled to prevent lag or slowdowns. Strategies such as entity pooling and efficient collision detection were used to solve these issues.

**Future Enhancements**

Future plans for Lost Plateau include expanding content with additional levels, new enemy types, and improved gameplay mechanics. Mobile compatibility is also a key focus, requiring optimization for touch controls and performance on lower-powered devices. Another potential enhancement is multiplayer support, which would introduce cooperative gameplay and broaden the player experience. Lastly, AI behavior improvements are planned to make enemy movement and decision-making more dynamic, adding depth to the gameplay.

**References:**

Godman, Heidi. “Having a Hobby Tied to Happiness and Well-Being.” Harvard Health, 1 Jan. 2024, www.health.harvard.edu/mind-and-mood/having-a-hobby-tied-to-happiness-and-well-being.

“Video Gaming May Be Associated with Better Cognitive Performance in Children.” National Institutes of Health, U.S. Department of Health and Human Services, 20 Apr. 2023, www.nih.gov/news-events/news-releases/video-gaming-may-be-associated-better-cognitive-performance-children.

“What Is Gamification (GF)?” The Interaction Design Foundation, 17 Jan. 2024, www.interaction-design.org/literature/topics/gamification.

Project Advisor: Michael O'Neill

Developer: Mack Wessels