University of Nevada, Reno

College of Engineering, Department of Computer Science and Engineering

Project Gemini

Revised Specification & Design

Team 01 - Lauren Davis - Nathan Evans - Allen Ma - Chris Trimble - Eric Valdez
CS 426 Instructors - Dr. David Feil-Seifer, Devrin Lee
External Advisor - Nathan Navarro Griffin (Adjunct Professor/Game Developer for Squanch Games)

February 22, 2023

Table of Contents

1. Abstract

Project Gemini is a 3D puzzle-adventure game developed in the Unity game engine. This game follows the player as they return to their childhood village which has fallen into disrepair due to seemingly supernatural forces. The player can speak and interact with several of the residents while attempting to rebuild the town. The player can then progress toward removing the evil plaguing the land by delving into the nearby dungeon. The game consists of two primary areas: the village and a desert themed puzzle dungeon. The primary goal for this project is to create a fun and entertaining game that will further develop the skills that each of the team members possess.

2. Recent Project Changes

Since assignment one, the team has not made many changes to the main design of the game. Some new content has been added however, including new level designs, an updated character controller, boss mechanics, updated player abilities, an updated dialogue system, and updated UI menus. The team is content with the direction the game is going, and we're focused on completing the remaining content for the game. After that, we'll make some finishing touches to ensure that everything feels good and works properly.

3 Updated Specifications

3.1 Summary of Changes in Project Specifications

The team's dedication to meeting goals and targets has been impressive, with the successful completion of all level one functional requirements from the Fall. Moreover, the team has been able to exceed expectations by achieving several level two goals, expanding the game's functionality even further. Despite the expansion of the project's scope, the team has been able to maintain a high level of focus and intentionality. While there have been changes to the project specifications, they have been relatively minor, as the team has had a solid vision for the game since the start. This has allowed for a smooth and efficient development process, enabling the team to concentrate on delivering a high-quality game that meets and exceeds expectations. Moving forward, the team is committed to continuing this level of excellence, refining and perfecting the game to ensure that it is the best it can possibly be.

3.2 Updated Technical Requirements Specifications

3.2.1 Functional Requirements

FR1. [1] PG will allow for the user to create/delete a game.
FR2. [1] PG will allow for the user to save their progress.
FR3. [1] PG will allow for the user to load a saved game.
FR4. [1] PG will implement settings and option menus.
FR5. [1] PG will implement a journal for tracking quests.
FR6. [1] PG will implement character movement.
FR7. [1] PG will implement character abilities.
FR8. [1] PG will implement AI/interactable non-player characters (NPCs).
FR9. [1] PG will implement a dungeon level with puzzles.
FR10. [1] PG will implement a boss for the dungeon level.
FR11. [1] PG will implement a music system.
FR12. [1] PG will implement dialogue.
FR13. [1] PG will implement quests for the user.

FR14. [2] PG will implement refined character mechanics.
FR15. [2] PG will implement refined Dungeon level and puzzle mechanics.
FR16. [2] PG will implement improved menu screens.

FR17. [3] PG will implement four more dungeon levels with individual bosses.
FR18. [3] PG will implement a finished music system.
FR19. [3] PG will implement multi-platform support.
FR20. [3] PG will implement more character abilities.
FR21. [3] PG will implement an achievement system.

Table 1: Functional Requirements for the video game Project Gemini (PG).

3.2.2 Non-Functional Requirements

NFR1. [1] PG will make sure the game does not crash.
NFR2. [1] PG will implement intuitive gameplay.
NFR3. [1] PG will implement simple to use menus.
NFR4. [1] PG will implement challenging yet intriguing puzzles.
NFR5. [1] PG will implement stylized graphics/animations.
NFR6. [1] PG will be compatible with Windows OS.
NFR7. [1] PG will introduce a storyline.

NFR8. [2] PG will implement refined stylized graphics and aesthetics.
NFR9. [2] PG will implement refined puzzle challenges.
NFR10. [2] PG will implement different types of optional settings to improve performance.
NFR11. [2] PG will make the game fun.

NFR13. [3] PG will implement multi-platform support.
NFR14. [3] PG will implement scalability to allow for more content.
NFR15. [3] PG will implement a complete story and game.

Table 2: Non-Functional Requirements for the videogame Project Gemini (PG).

3.3 Updated Use Case Modeling

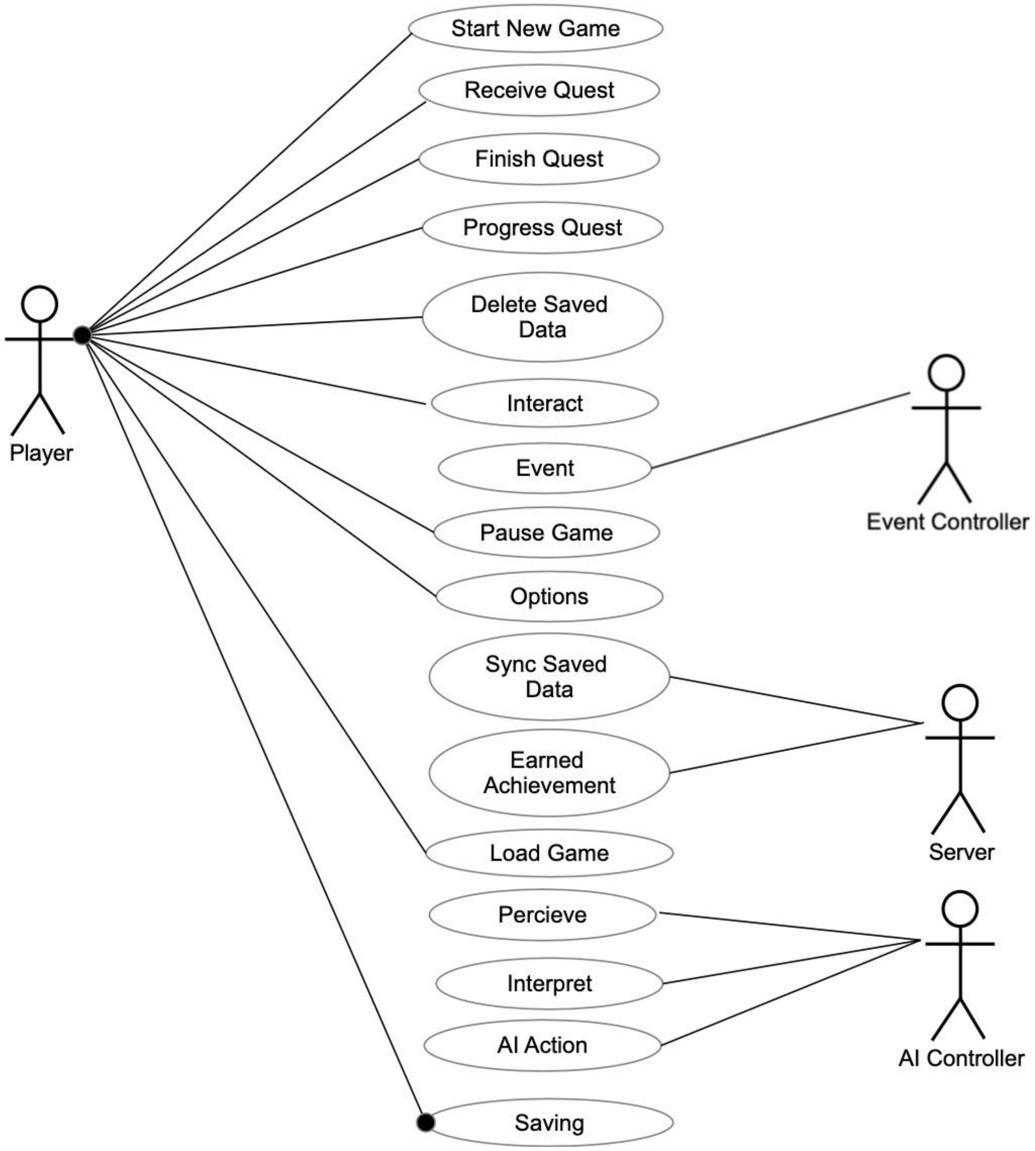


Figure 1: Diagram displaying the use cases and actors for Project Gemini.

3.3.1 Detailed Use Cases

ID	Use Case	Description
UC01	Start New Game	User can select from the main menu to begin a newly created data.
UC02	Receive Quest	User can talk to an NPC with a quest to offer and can choose to accept the quest which will be added to the journal.
UC03	Finish Quest	The user can finish a quest with the NPC who gave it once the quest's conditions are met.
UC04	Progress Quest	Progressing through the stages of a quest adds information to the journal and updates the journal image.
UC05	Delete Saved Data	From the main menu the user can delete any existing saved data.
UC06	Interact	The user can interact with NPCs, the environment, and more if those objects are designated as interactable.
UC07	Event	The event controller can take control of the game when the user has met the required conditions for the event to trigger.
UC08	Pause Game	The user can pause the game at any time.
UC09	Options	The user can adjust the options from the pause menu at any time to adjust the experience to each user's preference.
UC10	Sync Saved Data	The platform servers will sync cloud data and local data based on user preferences to allow progress to be tracked across multiple devices.
UC11	Earned Achievement	The platform servers will track achievement progress and award an achievement once the conditions are met.
UC12	Load Game	The user can from the main menu continue any of their prior saved data
UC13	Perceive	The AI entities will update on a per-frame basis evaluating the world and conditions around them.
UC14	Interpret	The AI entities will use the perception data to make choices.

UC15	AI Action	Based on the choices made the AI will perform appropriate actions.
UC 16	Save Game	The user can, from the pause menu, save the current player data.

Table 3: Detailed descriptions of the use cases for Project Gemini.

4 Updated Design

4.1 Summary of Changes in Project Design

When it comes to the design of Project Gemini, the team has been equally focused and deliberate. From the outset, the team had a clear vision for the game's aesthetics, mechanics and narrative, and the team worked tirelessly to bring that vision to life. The initial design concept was well-defined, with a consistent and cohesive art style that helped to establish the game's identity. We were able to meet our initial design goals of a completely unique and custom art set created by us. As development has progressed, the team has continued to refine and improve the game's design, adding new elements and features that enhance the user experience, and adjusting the narrative superficially. As the project moves forward, the team will continue to prioritize design as a key element of the game's success, ensuring that every aspect of the game's look and feel is polished and professional.

4.2 Updated High-Level and Medium-Level Design

Project Gemini has undergone significant development in recent months, and this evolution is illustrated in Figure 1, which displays the System Context Diagram. The project comprises six key components that are integral to its success. The first is the ScriptableObject Event System, which is a framework for creating event-driven scripts that can be customized to fit the needs of the project. The second component is JSON Data Management, which is a system for storing and retrieving data in JSON format. The third component is Audio, which is responsible for handling all of the game's sound effects and music. The fourth component is Player Interaction, which encompasses all of the interactions that the player can have with the game, such as movement. The fifth component is Feedback, which provides the player with visual and auditory cues to help them understand what is happening in the game. Finally, the sixth component is AI, which governs the behavior of non-player characters and enemies within the game. Each of these components plays a critical role in the success of Project Gemini and has been carefully designed and implemented to ensure that the game provides an engaging and immersive experience for players.

4.2.1 Data Structures and Storage

In the development of Gemini, the team has employed a variety of data structures to support the game's functionality and performance. One such structure is JSON, which is used to store dialogue trees and save data information. By using this lightweight, text-based format, the team has been able to efficiently manage large amounts of data without sacrificing speed or flexibility. In addition to JSON, the team has also utilized scriptable objects to keep track of game state and server as the local access, eliminating the need to constantly read from JSON. This has helped to streamline the game's performance, reducing load times and improving overall efficiency.

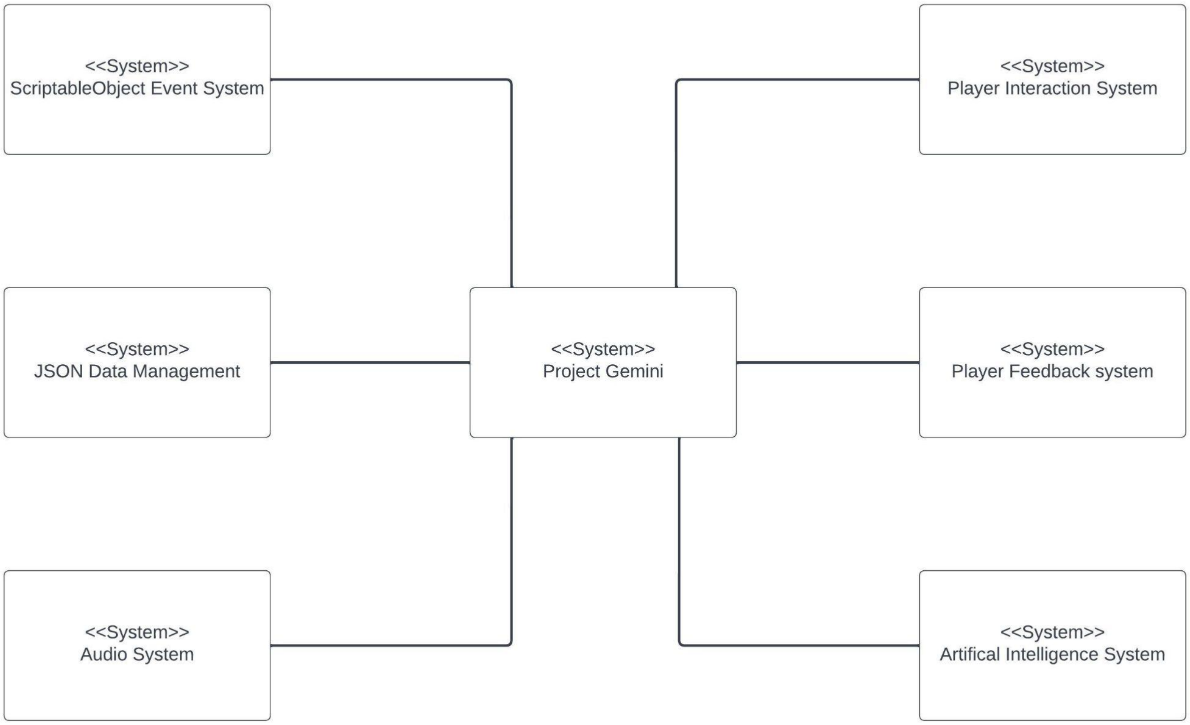


Figure 2: Project Gemini High Level Systems Diagram

4.3 Updated Hardware Design

As this project has been developed, there have been no hardware components necessary. As such, the hardware design of the team has not changed since the fall of 2022.

4.4 Updated User Interface Design

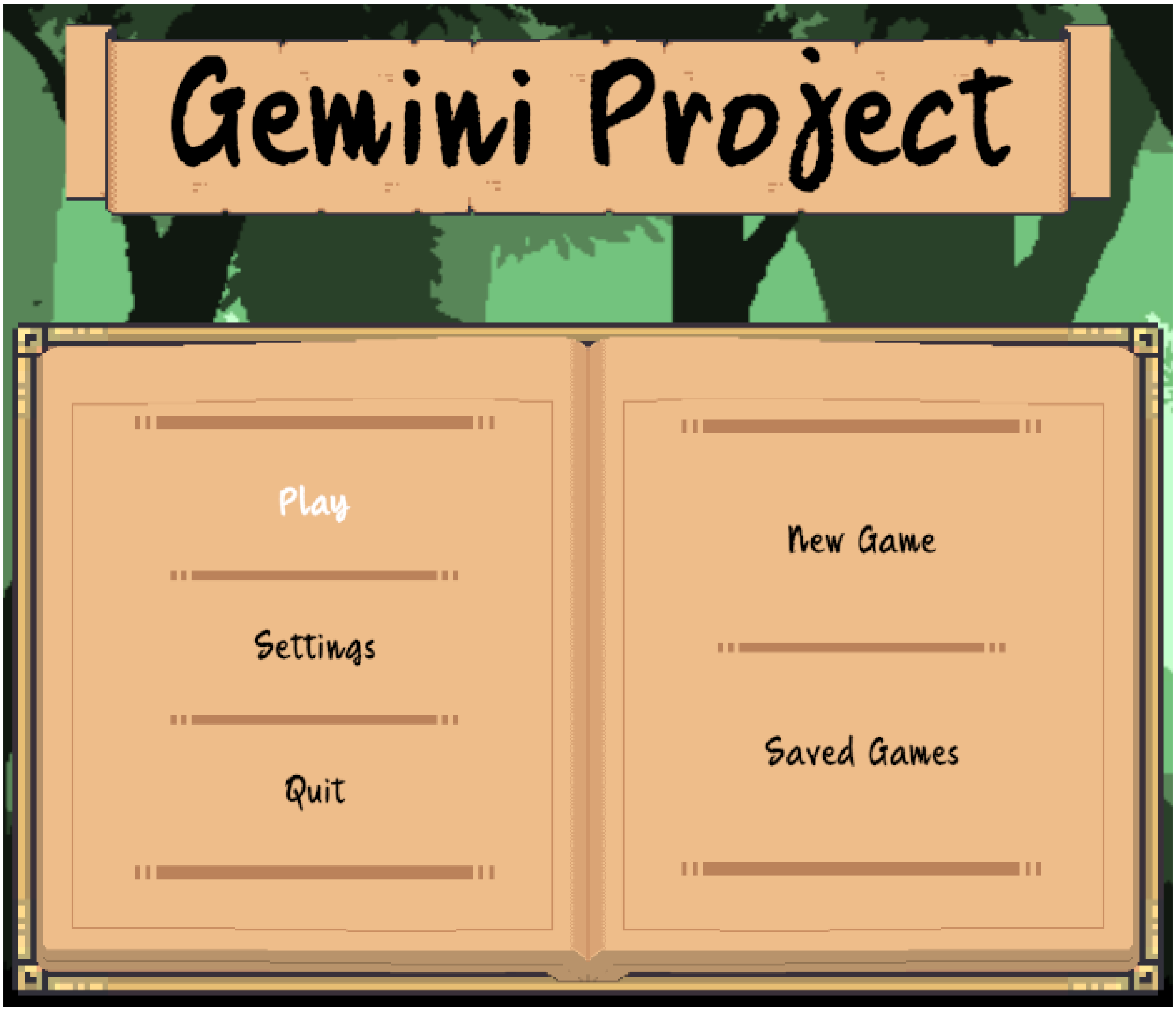


Figure 3: Main menu of the program.

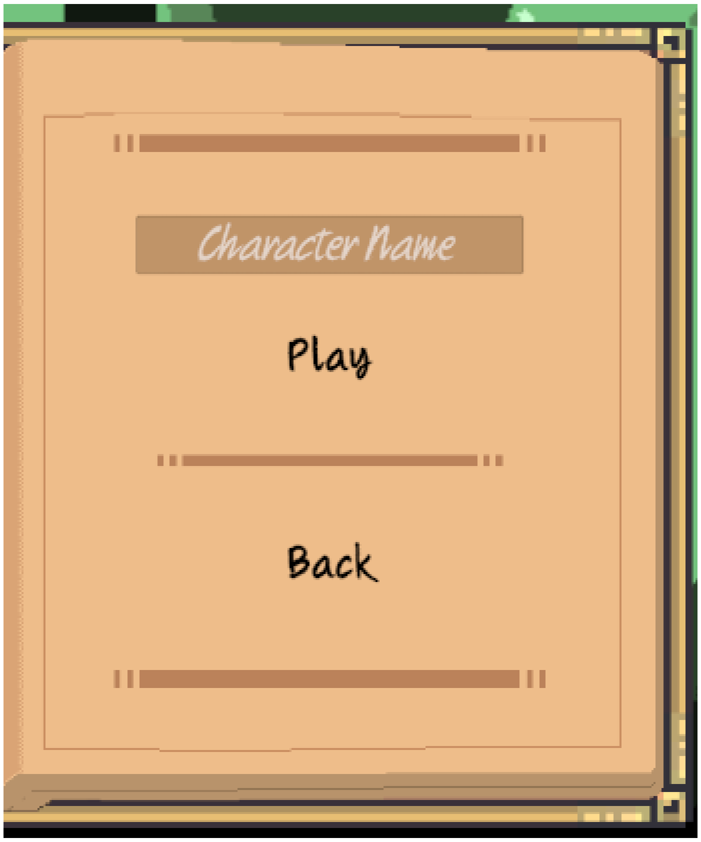


Figure 4: “New Game” menu for creating a new save.

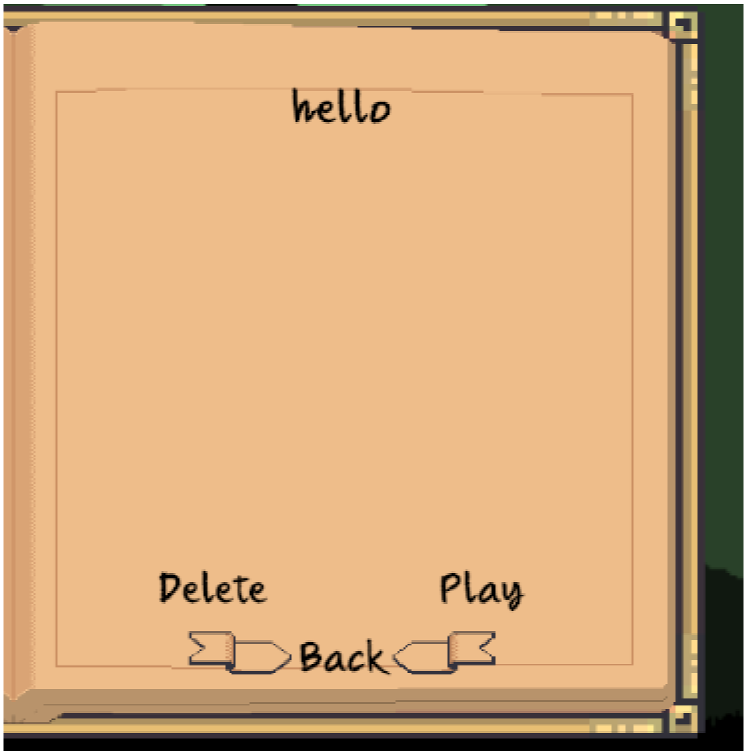


Figure 5: Saved games menu for loading previous saves.

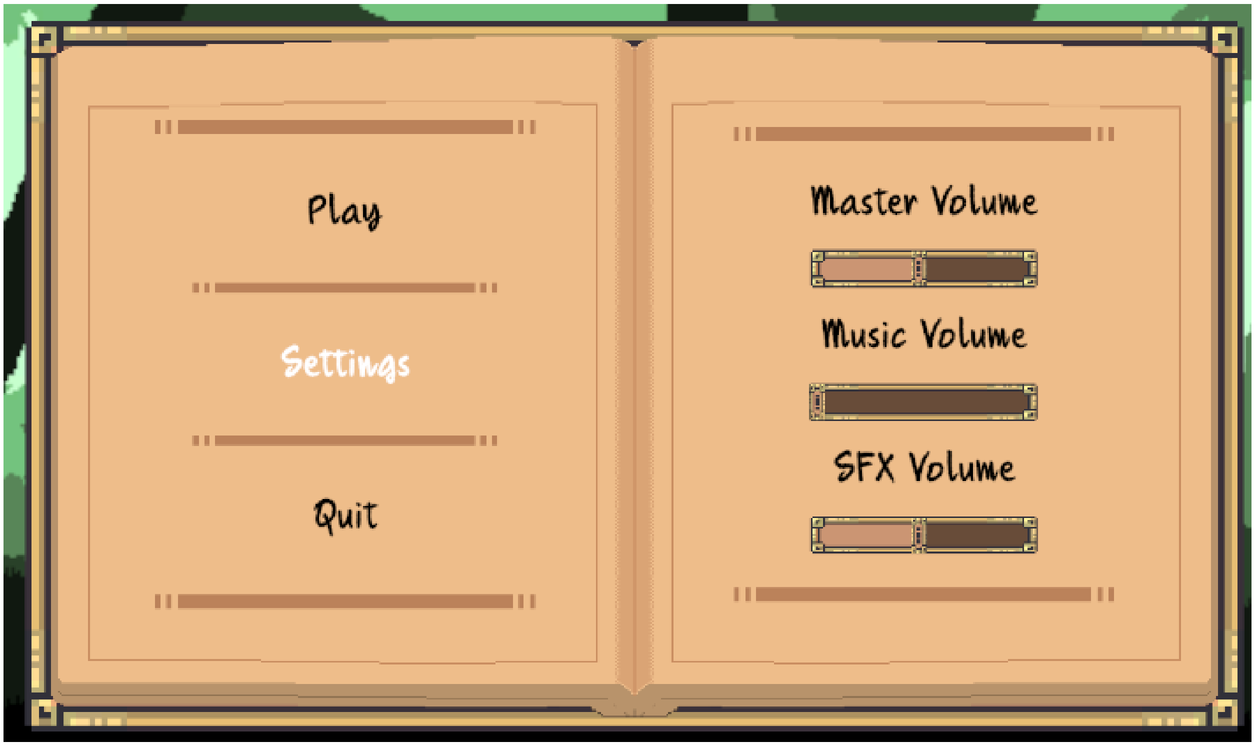


Figure 6: Main menu’s “Settings Menu”

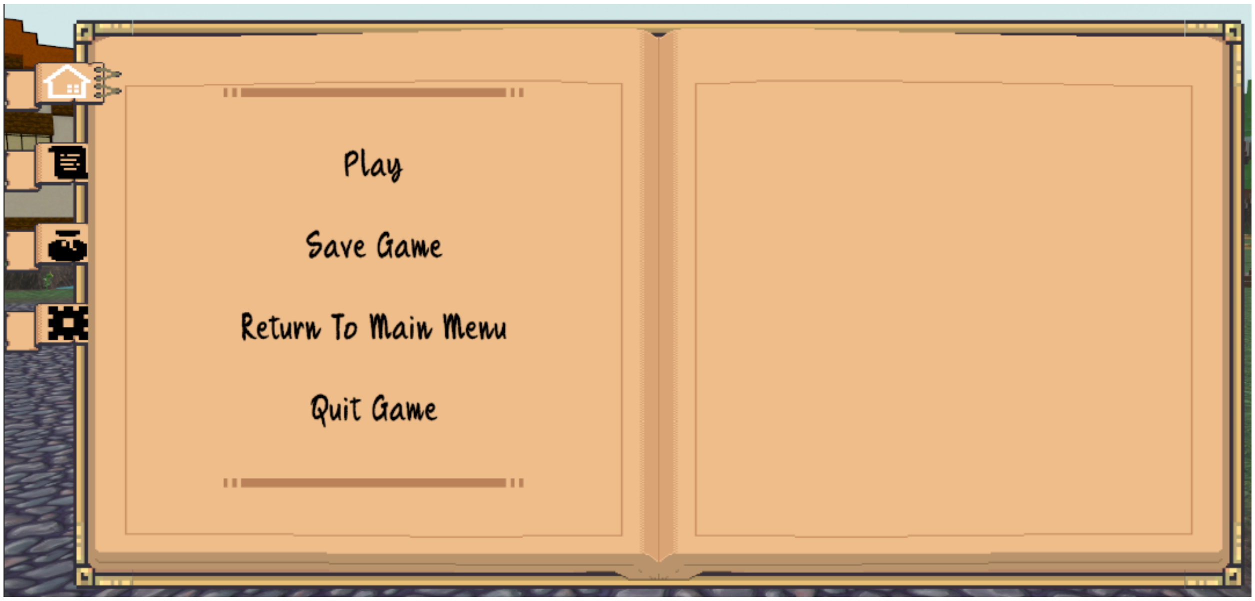
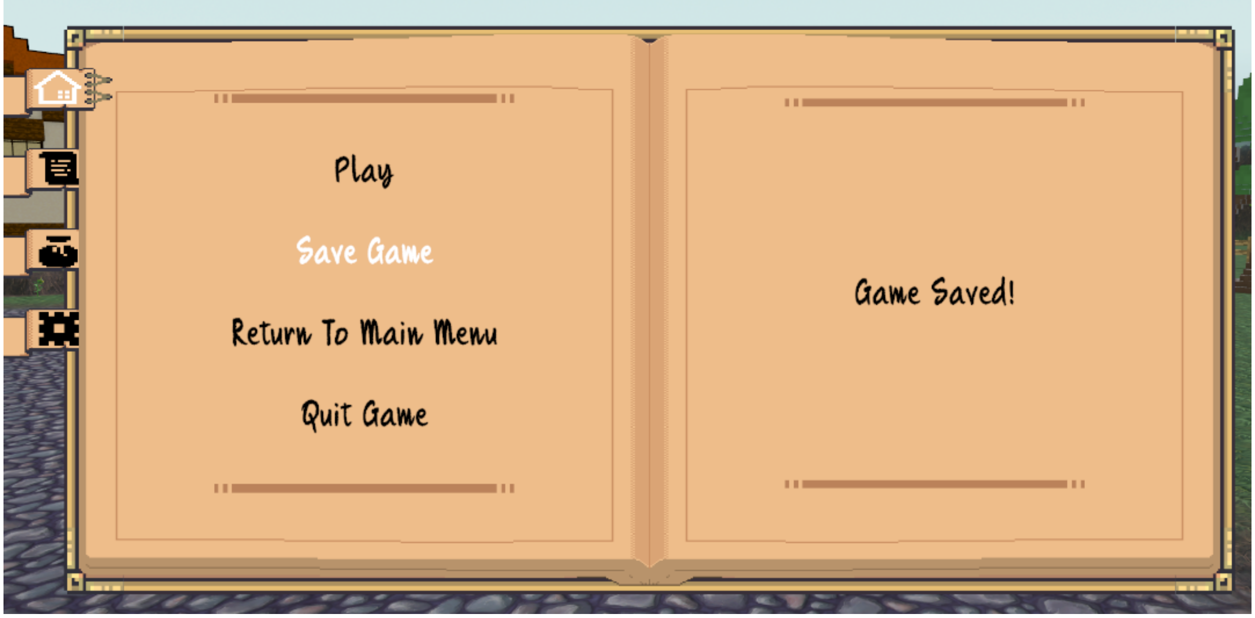


Figure 7: In-game pause menu



Game Saved!

Figure 8: In-Game Saving Menu

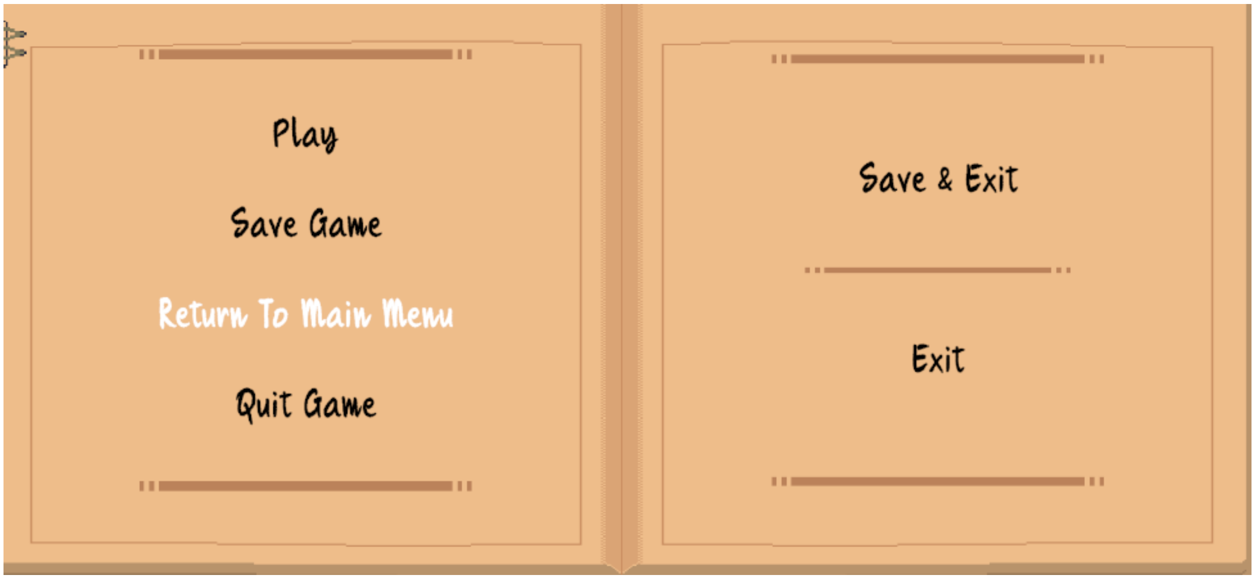


Figure 9: In-Game Return To Main Menu Menu

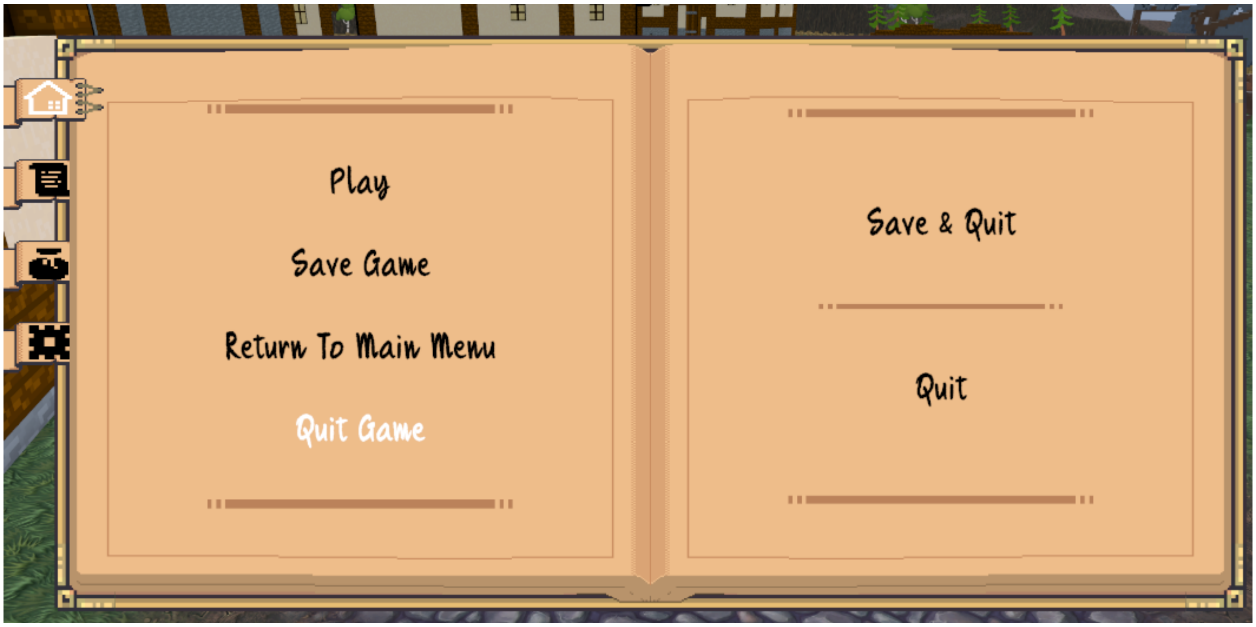
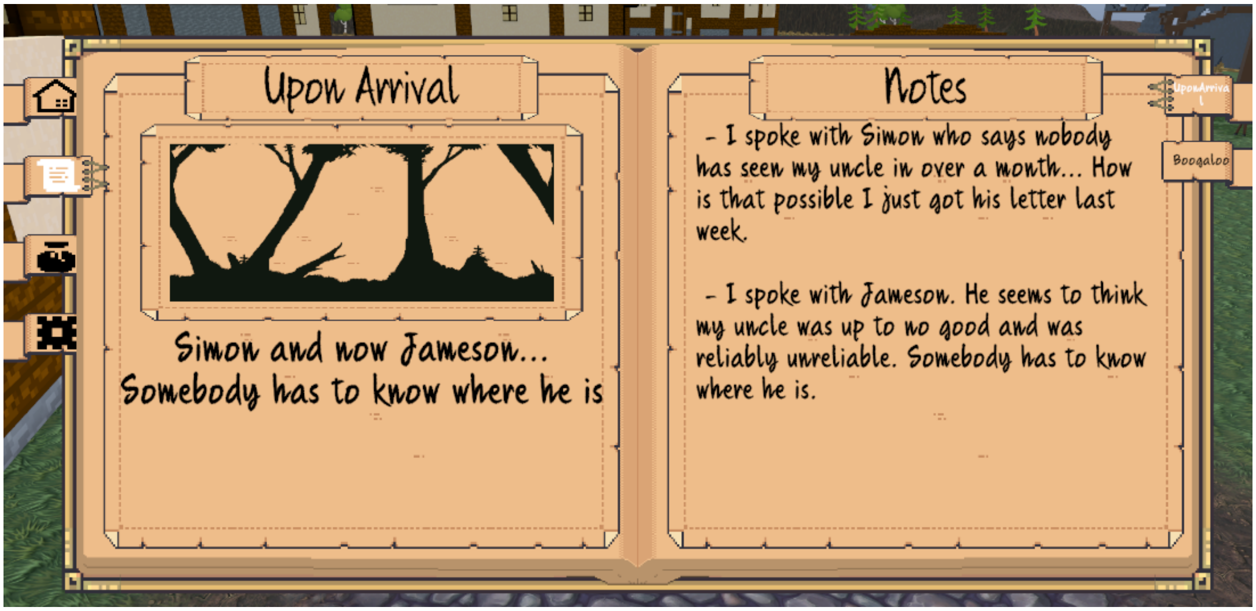


Figure 10: In-Game Quit Game Menu

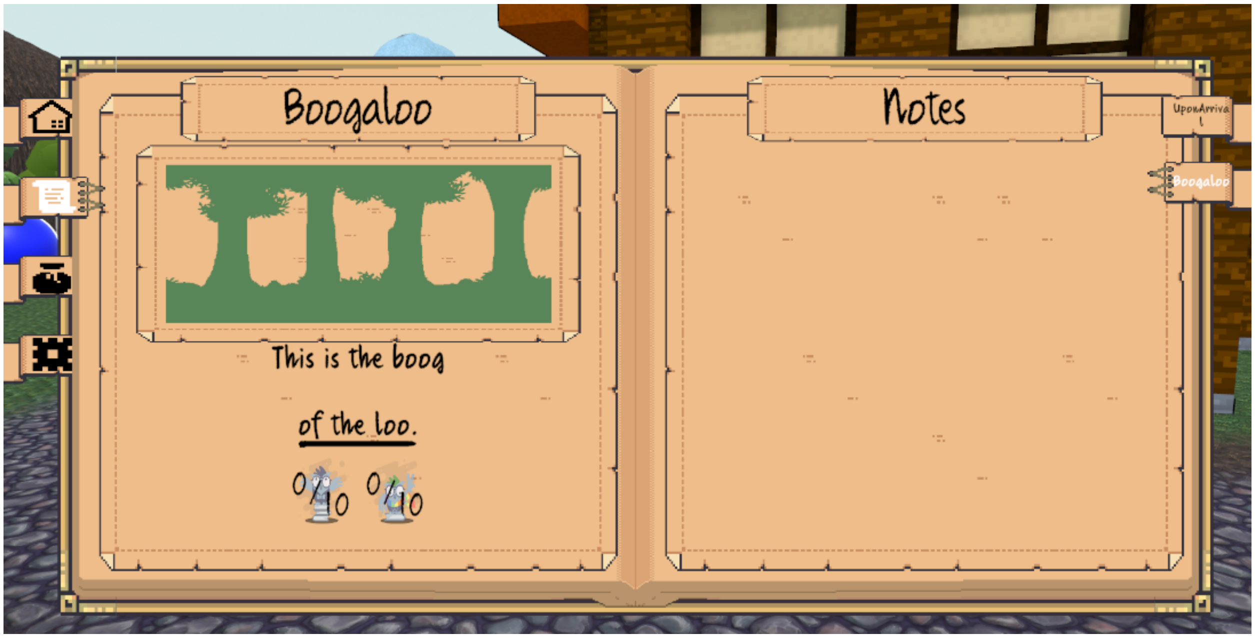


Simon and now Jameson... Somebody has to know where he is

- I spoke with Simon who says nobody has seen my uncle in over a month... How is that possible I just got his letter last week.

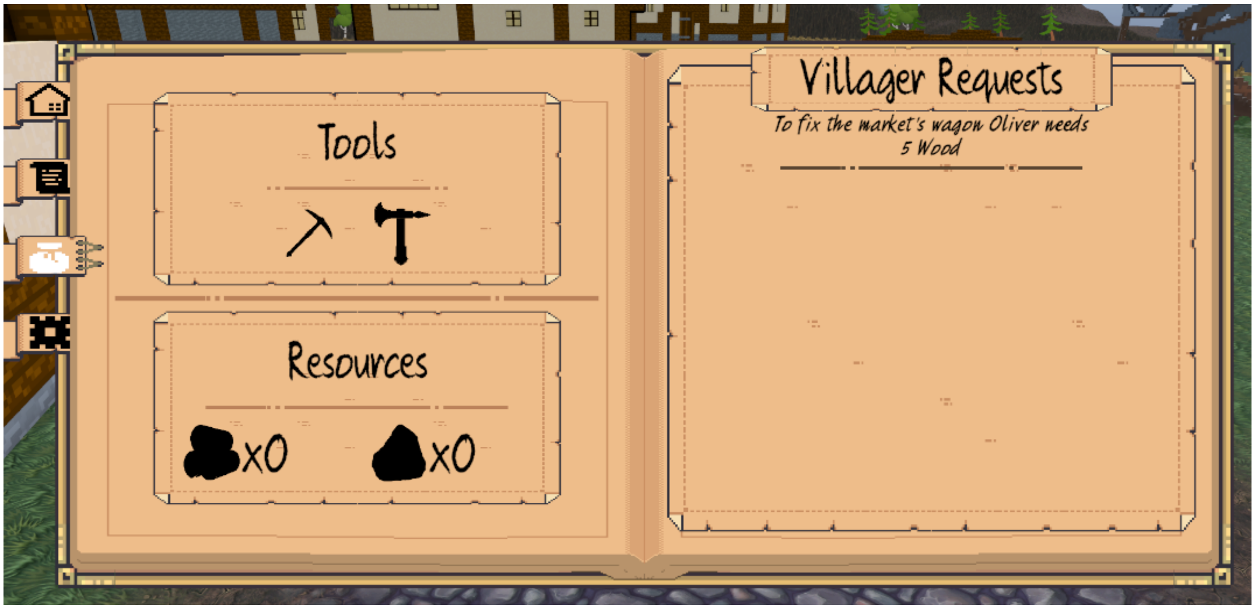
- I spoke with Jameson. He seems to think my uncle was up to no good and was reliably unreliable. Somebody has to know where he is.

Figure 11: Quest menu showing a quest at a later stage with notes



This is the boog of the loo.

Figure 12: Quest Menu of a quest in an earlier stage that needs quest items.



To fix the market's wagon Oliver needs
5 Wood

Figure 13: Resources menu that shows the player's tools, resources and requests given by the villagers.

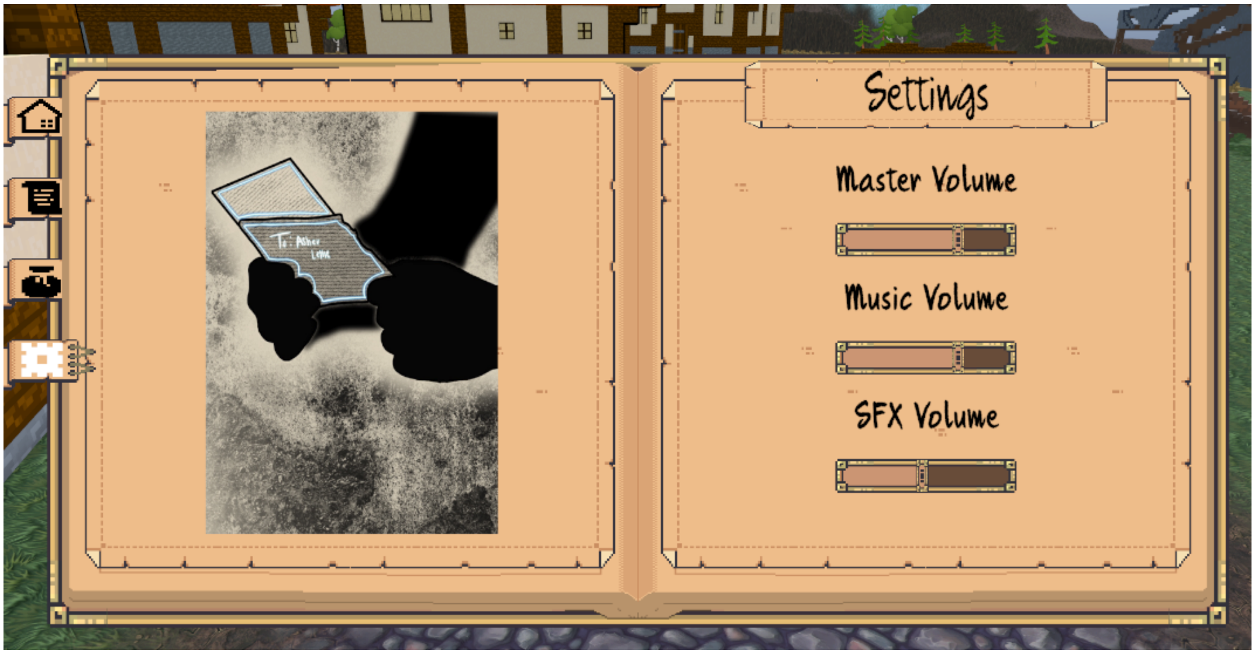


Figure 14: In-game “Settings Menu.”

5. Updated Glossary of Terms

1. Adventurer: A genre of video games that focuses on player exploration to fuel the narrative.
2. AI: An abbreviation that stands for artificial intelligence which is a game entity using coded computer algorithms to simulate human activity.
3. Balance: A method of creating an enjoyable experience for the player by manipulating mechanics.
4. Build: A term used to describe the compiling of game components together into an executable version of the game.
5. Clipping: A certain area of the game scene where the game components are rendered; everything outside the specified area is not rendered.
6. Collison: A component of a game entity that allows another entity to strike into it without the two entities passing through each other.
7. Cutscene: A visual where the player watches a depiction of the narrative and is unable to interact with any components within the game.
8. Demo: A playable version of the game, typically not for the official release but instead for receiving feedback and finding bugs.
9. Entity: A Unity term used to describe an object in the game that involves some interaction or functionality.
10. Game engine: A software that simplifies the key components of game development (graphics, physics, sound, networking, AI, and input processing) into usable tools to build games.
11. Grayboxing: A method used in designing blueprints of game levels for scaling, testing, and referencing so that it can be easily changed before polishing and adding assets.

12. Indie game development: A video game development by a small group of developers with personal aspirations to create the game idea.
13. Mechanics: The rules for the player to interact and progress through the game.
14. Multiplatform: A term used to describe the compatibility of the game build to be played on multiple hardware devices.
15. Playtest: A term used to describe the action of simulating realistic gameplay as a player to improve the overall game.
16. Prefab: A Unity term that is a consistent and reusable game object used throughout multiple different scenes.
17. Publisher Demo: A version of the game, typically a vertical slice, to show potential publishers in order to get the game published.
18. Puzzler: A genre of video games that focuses on solving various puzzles to progress through the game.
19. Release: A version of the game to be officially published and released to the public.
20. Render: The action of computer processing that constantly generates the graphics to the computer screen.
21. Scene: A Unity term that refers to the editable area for placing game entities or game objects to build the game environment.
22. UI: An abbreviation for the user interface which is a set of interfaces that interact with the player to display game information.
23. Unity: A free game engine developed by Unity Technologies used to make various games for a variety of different platforms.
24. UX: An abbreviation for user experience which refers to the player's experience with the UI.
25. Vertical Slice: A term used to describe a playable version of the game that represents a chunk of the game's concept; this chunk includes the majority of the main components and functionality related to the final version of the game.

6. Engineering Standards and/or Technologies

6.1 Engineering Standards

Accessibility Standards

Accessibility standards in video games refer to the measures taken to ensure that players with disabilities can enjoy the game on an equal footing with other players. Some common disabilities that can affect gameplay include visual impairments, hearing impairments, motor impairments, and cognitive impairments. For our game, we plan to implement sound cues for clicks, adjustable volume, adjustable dialogue sizes, and as a stretch goal, being able to have alternative control schemes.

Industry-Specific Standards:

The video game industry has several industry-specific standards. For example, the Entertainment Software Rating Board (ESRB) is a self-regulatory organization that assigns ratings to video games and other interactive media. The ratings are designed to inform consumers about the age-appropriateness and content of the game, so they can make informed decisions about whether or not to purchase or play it. The ESRB rating system consists of several different categories, including Early Childhood (EC), Everyone (E), Everyone 10+ (E10+), Teen (T), Mature (M), and Adults Only (AO). Each category has specific guidelines for content, and games are assigned a rating based on their level of violence, language, sexual content, and other factors. For our game we plan to use an ESRB rating system to rate our game for teenagers, which means content that may be suitable for ages 13 and older.

6.2 Technologies

Unity 3D

Unity is a cross-platform game engine that is being used to create Project Gemini. The engine itself can facilitate the creation of both two-dimensional as well as three-dimensional games. Project Gemini has been developed using the three-dimensional project template. Unity provides several benefits to the team, primarily with its built-in version control software that will be spoken about later which is known as PlasticSCM.

C#

C# is a modern object-oriented programming language that is very similar to other C based languages such as C++. C# is used for a wide variety of projects covering web development, Windows apps, and most importantly to this team, game development. C# is used as the built-in scripting language for Unity, which allows the team to create all scripts for Project Gemini under the same language constraints.

PlasticSCM

PlasticSCM is a commercial version control tool developed by Codice Software Inc. Plastic is a full version control stack which, while it can communicate with Git, PlasticSCM is not based on it. In 2020, Unity Technologies had acquired Codice Software and introduced the version control software into Unity itself. The Project Gemini team has been using this software as the primary version control for the project due to its ease of use with merging and distributing changes.

Blender

Blender is a free computer graphics tool which has many uses such as animations, creating models, and more. The primary use that the team has for using blender is for the creation of animations, specifically those relating to the characters. While the team has not currently used this software for the creation of any models, several animations have been created and implemented into the game using this software.

MagicaVoxel

MagicaVoxel is a free voxel-based art editor that gives artists the ability to easily create various three-dimensional models with voxels. This software is easy to use and easily allows for exporting models into other software such as Unity and Blender. Due to the ease of use with this tool, the Project Gemini team has been using this software to create the vast majority of art assets for the game. All characters and most of the scenery have been created with the help of this tool.

Procreate

Procreate is a raster graphics editor application developed by Savage Interactive that is available for download on iOS. This tool offers a wide variety of digital painting tools and has helped the team create many of the journal scenes that will be used as cutscenes within the game. Lauren completed several of these drawings previously and the team may be using this tool in the future to create more of these journal drawings.

7. Project Impact and Context Considerations

Project Gemini plans on impacting the video gaming community by engaging players through our gameplay, showcasing a unique narrative, and providing a source of entertainment for varying individuals. Socially, our game may offer players a sense of community as they deal with their personal struggles with their inner vices. After completing this game, they can have some relief by realizing they are not alone and might even have the strength to conquer their own vices in real life. Project Gemini's ability to inspire the breaking of mental health stereotypes also pertains to our cultural impact on society's perspective of mental health. Our environmental impact will be relatively small since we plan on releasing this game only electronically which reduces electronic waste correlated with physical products. Both economic and global impacts will be seen if the game is released to the public marketplace like Steam because it will have a global outreach and the ability to generate revenue in a multi-billion dollar industry.

Project Gemini's current accessibility is slightly limited between the UI and functionality since the game's current state requires a lot of mobility but with the addition of volume fluctuation some individuals with an auditory disability are able to adjust the volume higher. Due to a lack of accessibility, the team has added plans to improve color contrast, text size and scaling, keyboard accessibility, and potentially screen reader compatibility. For example, the team plans on researching color contrast filters through the collaborative asset store, offering customizable controls, use speech-to-text features, and text-to-speech features as well. In order to ensure accessibility, we plan on testing these features with keyboard-only testing, screen reader testing, color contrast testing, font size, and type testing, and hopefully using user testing with various individuals with disabilities.

8. Updated List of References

8.1 Problem Domain Book

Schell, Jesse. *The Art of Game Design*. Boca Raton: CRC Press, 2008. Print.

> This book goes into great detail about all the fundamental elements of game design. Each chapter analyzes a different decision made in order to form a game from game design to team formation. The book explains the thought process of game development like the designer makes an experience that creates a game for the player to experience. The designer, experiences, game, and player are described by the author to make successful design decisions to make a successful game.

8.2 Reference Articles

Engstrom, Henrik, et al. “Game Development From a Software and Creative Product Perspective: A quantitative literature review approach.” *Entertainment Computing,* vol. 27, 2018, pp.10–22. https://doi.org/10.1016/j.entcom.2018.02.008.

> This article discusses the problems and balancing issues related to creating a software product and a creative product. This duality of game development causes issues with the management of these projects since software development works agilely and creative development finds agile production too restrictive for their creativity. The article promotes more research focused on this balance between an engineering mindset and a creative mindset in order to make game development more effective and efficient.

Freeman, Guo, and McNeese, Nathan J. “Exploring Indie Game Development: Team Practices and Social Experiences in A Creativity-Centric Technology Community.” *Computer Supported Cooperative Work*, vol. 28, 2019, pp. 723–748. https://doi.org/10.1007/s10606-019-09348-x

> This article uses indie game development in their research to conduct an analysis of team collaboration and social experiences. Since indie game development teams are small, self-made, and promote individual creativity alongside teamwork. The research found that shared aspirations were foundational in the formation of the team, core team practices were essential to creating a collective vision for the game, and tensions arise from the complex social experiences between individual work and teamwork.

Perez-Liebana, Diego, et al. "General Video Game AI: A Multitrack Framework for Evaluating Agents, Games, and Content Generation Algorithms," *IEEE Transactions on Games*, vol. 11, no. 3, 2019, pp. 195–214. http://doi.org/10.1109/TG.2019.2901021.

> This article discusses the General Video Game AI (GVGAI) which is an AI competition focused on making general AI algorithms to be used in any game. The advancements in this competition provide academic tools for teaching AI algorithms to newer game developers as well as new tools for current game developers to use. For example, a game designer could use AI to play test their game instead of doing it manually.

Sushra, Tulasi, et al. “Judging a Book by its Cover: Significance of UX Design in Gamification and Computing Systems.” *Iran Journal of Computer Science,* vol. 5, 2022, pp. 117–126. https://doi.org/10.1007/s42044-021-00096-0.

> This article provides the differences between UI and UX development within the gaming industry and analyzes the importance of UX design through human-computer interaction. The study claims a successful UX design can be developed through gamification, hardware computing systems, and software computing systems. In general, the development of UX should focus on the aesthetics of the UI being pleasing to the human eye, emotional outreach to the player, and creating usable and accessible computing and software systems.

8.3 Project Related Websites

https://docs.unity.com/

> The Unity documentation provides information on how to use the multiple Unity components. The Unity Editor manual details how to use the Unity Editor to make a 2D or 3D game. The scripting reference refers to the documentation on making Unity scripts for coding. Since we plan on using the Unity game engine to build our game the documentation is a great resource to help our development.

https://www.voxelmade.com/magicavoxel/

> This website provides resources for the software used in developing voxels, a 3D box art. We plan on using Magicavoxel to develop customized art for our game. Along with the ability to download the software, the website provides information on different voxel software. The site also includes links to how to use the software functionalities, understand the software’s interface, and online tutorials to help assist in voxel creations.

https://trello.com/tour

> Trello is project management software that our team plans on using to organize the development process. Trello allows users to visualize tasks in different areas of development. For example, our team sections processes into 6 categories: backlog, bugs, current to-dos, in development, needs testing and finished. By assigning team members tasks and associated due dates, Trello helps manage and organize the workload.

https://www.blender.org/support/

> Blender is a free and open-source software which can be used for creating animations, models and visual effects. Blender has already been used by the team in order to create several animations for the characters. This website provides documentation on the software as well as links to helpful tutorials which could prove useful further into development of Project Gemini.

9. Contributions of Team Members

Lauren Davis

1.5 hours - Updated References, Updated Glossary, Project Impact and Context Considerations

Nathan Evans

1.5 hour - Updated Specifications, Updated Design

Allen Ma

1 hours - Project Changes, Project Standards

Christopher Trimble

1 hours - Updated User Interface

Eric Valdez

1.5 hours - Abstract, Project Technologies, and General Formatting