

University of Nevada, Reno

**College of Engineering, Department of Computer Science and
Engineering**

Project Gemini

Progress Demo

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)

March 13, 2023

Table of Contents

Table of Contents	1
1. Use Cases & Requirements Implemented	2
1.1 Functional Requirements	2
1.2 Non-Functional Requirements	2
1.3 Use Cases Implemented	2
<i>Example Use Cases Created By Nathan:</i>	2
<i>Example Use Cases Created By Allen:</i>	3
<i>Example Use Cases Created By Lauren:</i>	3
<i>Example Use Cases Created By Chris:</i>	4
<i>Example Use Cases Created By Eric:</i>	4
2. Use Cases & Requirements Not Yet Implemented	5
2.1 Functional Requirements	5
2.2 Non-Functional Requirements	5
2.3 Use Cases Yet To Be Implemented	5
3. Summary of Project Status	6
4. Contributions of Team Members	7
Contributions on Project Part 4	7
<i>Lauren Davis</i>	7
<i>Nathan Evans</i>	7
<i>Allen Ma</i>	7
<i>Christopher Trimble</i>	7
<i>Eric Valdez</i>	7
Contributions Since CS 425 Demo	7
<i>Lauren Davis</i>	7
<i>Nathan Evans</i>	7
<i>Allen Ma</i>	7
<i>Christopher Trimble</i>	7
<i>Eric Valdez</i>	7

1. Use Cases & Requirements Implemented

1.1 Functional Requirements

FR1.	[1] PG will allow for the user to create/delete a game. (Lauren)
FR2.	[1] PG will allow for the user to save their progress. (Lauren)
FR3.	[1] PG will allow for the user to load a saved game. (Lauren)
FR4.	[1] PG will implement settings and option menus. (Chris)
FR5.	[1] PG will implement a journal for tracking quests. (Chris)
FR6.	[1] PG will implement character movement. (Allen)
FR7.	[1] PG will implement character abilities. (Eric)
FR8.	[1] PG will implement AI/interactable non-player characters (NPCs). (Nathan)
FR9.	[1] PG will implement a dungeon level with puzzles. (Lauren & Eric)
FR10.	[1] PG will implement a boss for the dungeon level. (Chris)
FR11.	[1] PG will implement a music system. (Chris)
FR12.	[1] PG will implement dialogue. (Nathan)
FR13.	[1] PG will implement quests for the user. (Nathan)
FR14.	[1] PG will implement challenging yet intriguing puzzles. (Lauren & Eric)
FR15.	[1] PG will introduce a storyline. (Nathan)
FR16.	[2] PG will implement refined character mechanics. (Eric)
FR17.	[2] PG will implement refined Dungeon level and puzzle mechanics. (Lauren & Eric)
FR18.	[2] PG will implement improved menu screens. (Chris)

Table 1: Functional Requirements Implemented in Project Gemini.

1.2 Non-Functional Requirements

NFR1.	[1] PG will make sure the game does not crash. (Team)
NFR2.	[1] PG will implement intuitive gameplay. (Team)
NFR3.	[1] PG will implement simple to use menus. (Chris)
NFR4.	[1] PG will implement stylized graphics/animations. (Allen & Eric)
NFR5.	[1] PG will be compatible with Windows OS. (Team)

Table 2: Non-Functional Requirements Implemented in Project Gemini

1.3 Use Cases Implemented

Example Use Cases Created By Nathan:

As a player I want to be able to interact with the story through NPC conversations.

- A player can initiate conversations with the “E” key when in range.
- A player can skip dialogue animations with the “RMB”
- A player can select dialogue choices with the mouse.
- A player can get different conversations based on world state.
- A player can finish a quest goal by turning in required quest materials.

As a player I want to be able to interact with the quest system through NPC conversations.

- A player can receive a quest through NPC Conversation
- A player can receive a request though NPC conversation
- The world state changes as a result of NPC Conversation
- A player can finish a quest though NPC Conversation
- A player can finish a request through NPC Conversation.

Example Use Cases Created By Allen:

As a player, I want to be able to move my character around the world to explore different things and progress the story.

- A player can press “WASD” keys to move.
- A player can move their mouse around to look around.
- A player can press “Space” to jump.
- A player can press “Left Shift” to sprint.
- A player can utilize a jump buffer and coyote time when jumping that makes gameplay more fluid

As a player, I want to see my character’s movement corresponding to the animations I am inputting.

- The animation for walking plays when a player presses “WASD”.
- The animation for jumping plays when a player presses “Space”.
- The animation for running plays when a player hold’s “Left shift” and uses “WASD” to run.

Example Use Cases Created By Lauren:

As a player, I want to create a new character so I can play the new game without any previous gameplay experience.

- A player is in the main menu
- The player has already selected “Play” from the UI.
- The player presses “New Game” from the UI to create a new character.
- The player types in the name they would like to use for the new game.
- The player presses “Play”.
- After the player presses “Play” for the second time, a new player is developed and saved to JSON files.

As a player, I want to continue playing the game so that I can play the game with my previous character’s information and experience.

- A player is in the main menu.
- The has already selected “Play” from the UI.
- The player presses “Saved Game” from the UI to view a list of all previously saved characters.
- The player selects the character name of which we would like to continue playing by pressing the name and then pressing “Play” from the UI.
- After pressing “Play”, the specified character information is located within the JSON files and reads all previous information into the player object.

As a player, I want to save my character's progress so that I can continue playing the game later.

- The player has already begun playing the game.
- The player presses “P” on the keyboard to bring up the pause menu.
- The player presses “Save Game” from the UI.
- After pressing “Save Game”, the current player information (e.g position, amount of materials, etc) is saved to the JSON file corresponding to the character’s name by replacing the previous information.
- The player may now close the game or press “Quit” from the UI.

Example Use Cases Created By Chris:

As a player, I want to check what quest items I need for a certain quest

- A player is loaded into a level of the game.
- The player presses “P” on the keyboard to bring up the pause menu.
- The player clicks on the quest icon tab using the mouse left button.
- The player then clicks on the specified quest names tab using the mouse left button.
- The player will now see what quest items they need for the quest and which ones they have already gathered on the left page.

As a player, I want to change the sound level of the music.

- A player is loaded into a level of the game.
- The player presses “P” on the keyboard to bring up the pause menu.
- The player clicks on the settings “cog” icon using the mouse left button.
- The player then clicks on the music levels bar and drags to where they want the volume.

As a player, I want to check which NPC requests I have left.

- A player is loaded into a level of the game.
- The player presses “P” on the keyboard to bring up the pause menu.
- The player clicks on the resources tab “bag” icon using the mouse left button.
- The player is now able to identify which NPC requests are incomplete, as they will not have a strikethrough indicating completion status.

Example Use Cases Created By Eric:

As a player, I want to create boxes so that I can activate pressure plates to open doors and activate traps.

- A player can press a button to create a crate.
- If the button is held, the player will see an indicator of where that object will appear.
- The crate is created based on the position of the player character and the camera’s rotation.
- After a crate is created it can be pushed around by the player.
- If a crate is created or pushed onto a pressure plate a door or trap will activate.
- Creating a crate while one exists will cause the other crate to fade out and be destroyed.

As a player, I want to create an elevator so that I can reach areas that would otherwise be out of reach.

- A player can press a button to create an elevator.
- If the button is held, the player will see an indicator of where that object will appear.
- The elevator is created based on the position of the player character and the camera's rotation.
- After an elevator is created, the object will raise to a specified height.
- Once the elevator has reached its peak, it will be destroyed after a short pause.

As a player, I want to throw a hook so that I can interact with objects that are far away and pull objects towards me.

- A player can press a button to instantiate a hook that will progress until it hits an object.
- If the object that is hit by the hook is "grabbable," the object will be brought back to the player.
- If the object that is hit by the hook is a lever, the lever will be flipped which can activate doors, traps, or reveal hidden objects.
- If the hook hits any other objects the hook will return to the player without any other interaction.

2. Use Cases & Requirements Not Yet Implemented

2.1 Functional Requirements

<p>FR19. [2] PG will implement refined puzzle challenges. (Lauren & Eric)</p> <p>FR20. [2] PG will make the game fun. (Team)</p>
--

Table 3: Functional Requirements That Are Not Yet Implemented

2.2 Non-Functional Requirements

<p>NFR6. [2] PG will implement refined stylized graphics and aesthetics. (Allen & Eric)</p> <p>NFR7. [2] PG will implement different types of optional settings to improve performance. (Chris)</p>

Table 4: Non-Functional Requirements That Are Not Yet Implemented

2.3 Use Cases Yet To Be Implemented

As a player, I want to be able to jump at different heights in order to make platforming more fluid. (Allen)

- A player can press and hold the jump button to add different heights.
- Hang time on player jumps will be increased to provide a slightly less restrictive gravity to the player.

As a player, I want to run through a complete storyline so that I can be immersed within the world of Project Gemini. (Nathan)

- A player will be able to speak with NPCs in order to learn more about the world from their perspectives.
- A player will be able to get more quests directly tied to the storyline
- A player can partake in options side requests.

As a player, I would like to experience more levels within the dungeon with more varied content. (Lauren, Allen, & Eric)

- A player will be able to traverse new level(s) that include more content.
- These new levels will include similar traps and puzzles, but with different ways to interact with the levels.
- New arrow traps may be added, more platforming segments, etc.

As a player, I would like to see more details while fighting the boss to better understand how to overcome the challenge. (Chris)

- The player will be able to see which column the boss will be jumping to and from.
- The attacks from the boss will contain custom art.

As a player, I would like to be able to gather materials in order to help rebuild the town. (Eric)

- A player can press a button in order to chop down trees and split rocks.
- Once these material nodes are destroyed a collectible will spawn.
- The player will be able to collect these materials and bring them back to an NPC that requested them.

3. Summary of Project Status

Our project's current status is in progress and on schedule to be completed before final demos. As seen above in the requirements section, our team has fulfilled all of our level one functional and non-functional requirements and has commenced implementing level two requirements. Since December, we have made significant progress with the development of the dialogue system and creation of the narrative design. The dungeon levels have also been fully developed, complete with challenging puzzles and an AI boss framework. We've added new environmental assets to the dungeon levels and character assets for the boss, and updated the village with environmental assets that better reflect the game's current atmosphere. Our UI and menus now have new art assets and animations. In order to enhance the game's overall ambience, we've developed an audio system that effectively manages different audio sources. We've also updated the player controller to improve the feel, reduce object clipping with the camera, refine animation graphics, and enhance light abilities with directional input and indicators. These recent efforts have brought us very close to content completion, allowing some of us to focus on reducing player friction. Going forward, our plans include continuing to address player friction and improving the game's overall quality.

4. Contributions of Team Members

Contributions on Project Part 4

Lauren Davis

1 hour - Use cases, Functional requirements, Project Summary.

Nathan Evans

0.5 hours - Use cases, Functional Requirements.

Allen Ma

0.5 hours - Use cases, Functional Requirements.

Christopher Trimble

0.5 hours - Use cases, Functional Requirements.

Eric Valdez

0.5 hours - Use Cases, Requirements and General Formatting

Contributions Since CS 425 Demo

Lauren Davis

100 hours - Dungeon Levels, and Puzzles

Nathan Evans

120 hours - Dialogue, Narrative, Ai Behavior Tree, Project Management

Allen Ma

95 hours - Movement, animations, character design.

Christopher Trimble

135 hours - Menu system, Boss mechanics, Audio system.

Eric Valdez

110 hours - Character Abilities, Resource Gathering System, Environment Art, Tutorial Level