**Software Requirements and Design Document**

**For**

**Group 11**

Version 1.0

**Authors**:

Alexis Amoyo

Katelyn Fischer

Hannah Housand

Olivia Mei

Sophia Quinoa

# Overview (5 points)

Dreamscapes ™ is a visual novel game with an exciting and immersive storyline game that puts you in the driver's seat of a dreamscape narrative. As you progress through the dreams, you'll encounter unexpected moments where you must make decisions by answering questions with multiple choices that will directly impact the outcome of the story. This offers a personalized and unique experience every time you play and makes you “never want to wake up.”

The game encompasses 5 chapters featuring unique dreams complete with a diverse array of visual assets, backgrounds, and music. The user plays as the main character of each dream and will be able to save and load progress upon pausing the game. If the user has at least one bad ending by the final chapter, they will receive a bad ending in which the user will not wake up from the dream. If the user has achieved a good ending from each of the chapters, the user will wake up from the dream in the game’s true good ending.

# Functional Requirements (10 points)

Overall

1. The game shall have a playable character. (High)
2. The game shall consist of 5 interactive chapters. (High)
3. Each chapter should contain a unique story. (High)
4. The beginning of each chapter will be clearly marked by a chapter title screen. (Low)
5. The game should have visual assets for characters and backgrounds throughout the entire story. (High)
6. The name of the main character should be taken from the user. (Medium)
7. If there is no name provided by the user, the default character will be “Nameless One”. (Low)
8. Game shall display text related to narration, a character’s thoughts, and a character’s dialogue. (High)

Saves

1. There will be two types of saves in the game: checkpoint saves, and mid-chapter saves. (High)
2. The user shall be able to save their progress at any point in the game as a mid-chapter save. (High)
3. The system will have a single save slot for mid-chapter saves. (Medium)
4. After completing a chapter, the system should save a checkpoint of the user’s progress. (High)
5. The user should be able to return to any saved checkpoint at any point in the game. (Medium)
6. The user should be able to load progress made at their mid-chapter saved point. (Medium)
7. If the user exits the game, they will be prompted to save their progress. (Medium)
8. If the user tries to return to a checkpoint save, they will be prompted to save their progress as a mid-chapter save. (Medium)

Endings

1. The user should be able to make choices that affect the storyline and ending of each chapter. (High)
2. If the user reaches a bad ending to a chapter, they can choose to replay the story or continue to the next chapter. (Medium)
3. The ending of the entire game is unlocked once the user achieves a “good” ending from each chapter. (Low)
4. The ending consists of a cutscene of the player waking up from their sleep. (Low)
5. A “good” or “bad” ending is defined by each individual chapter. (Medium)
6. If the user completes all 5 chapters and does not achieve all good endings, they will be prompted to go back and redo the failed chapters. (Low)

Additional Features

1. The game shall play music throughout its duration. (Low)
2. There shall be minigames throughout the game that affect the storyline. (Low)

Pause Screen

1. The pause screen shall be accessible at any point in the game. (High)
2. Pressing the escape key will open the pause screen during the game. (High)
3. The user will be able to access the main menu, exit, settings, and save options from the pause menu. (Medium)

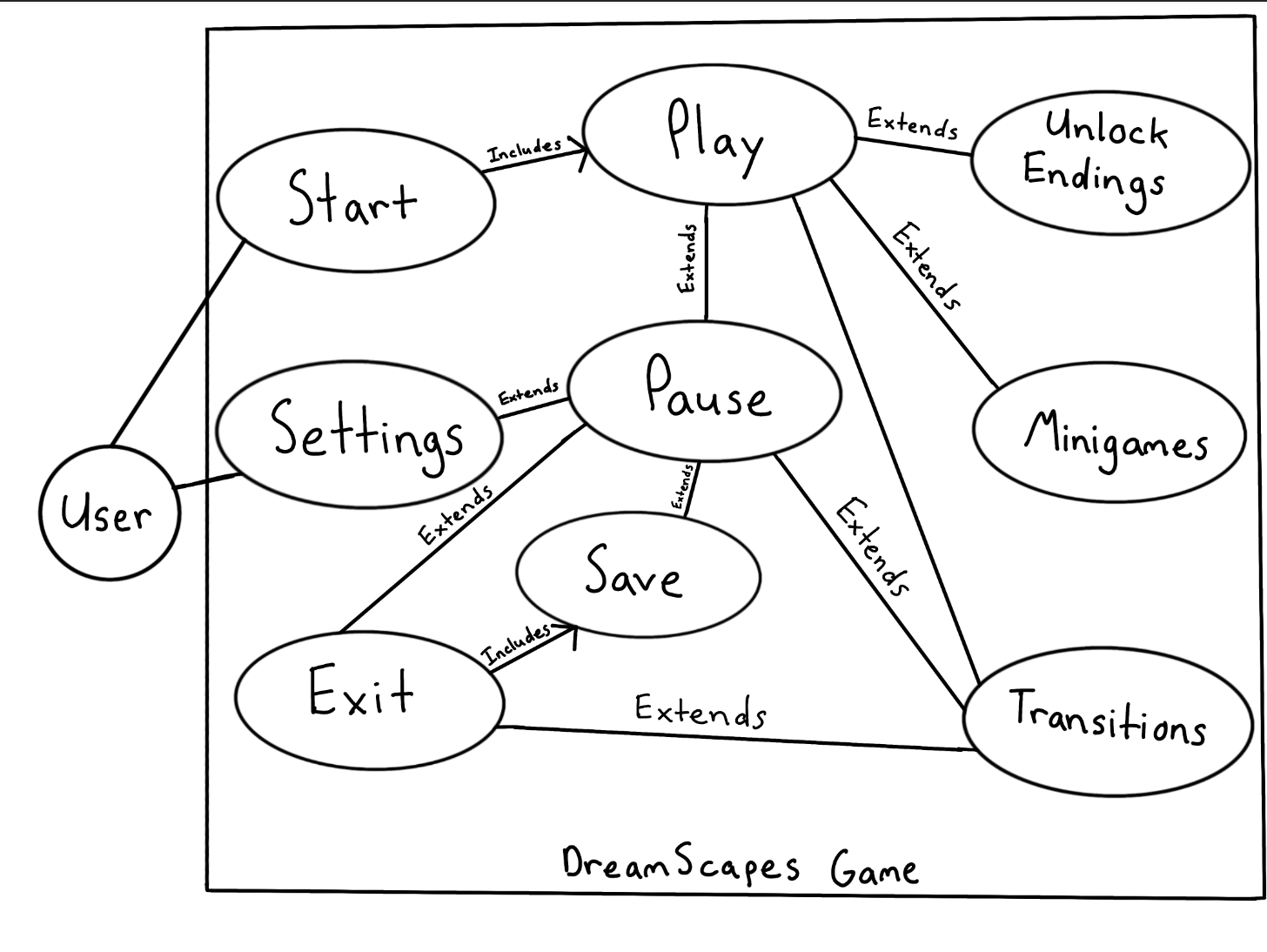
Main Menu Screen

1. The main menu screen contains options to start the game from the beginning, resume from the save point, access saved checkpoints, and settings. (Medium)
2. The main menu screen should clearly indicate the name of the game. (High)
3. Button options on the menu screen should be clearly visible and easy to access. (High)

# Non-functional Requirements (10 points)

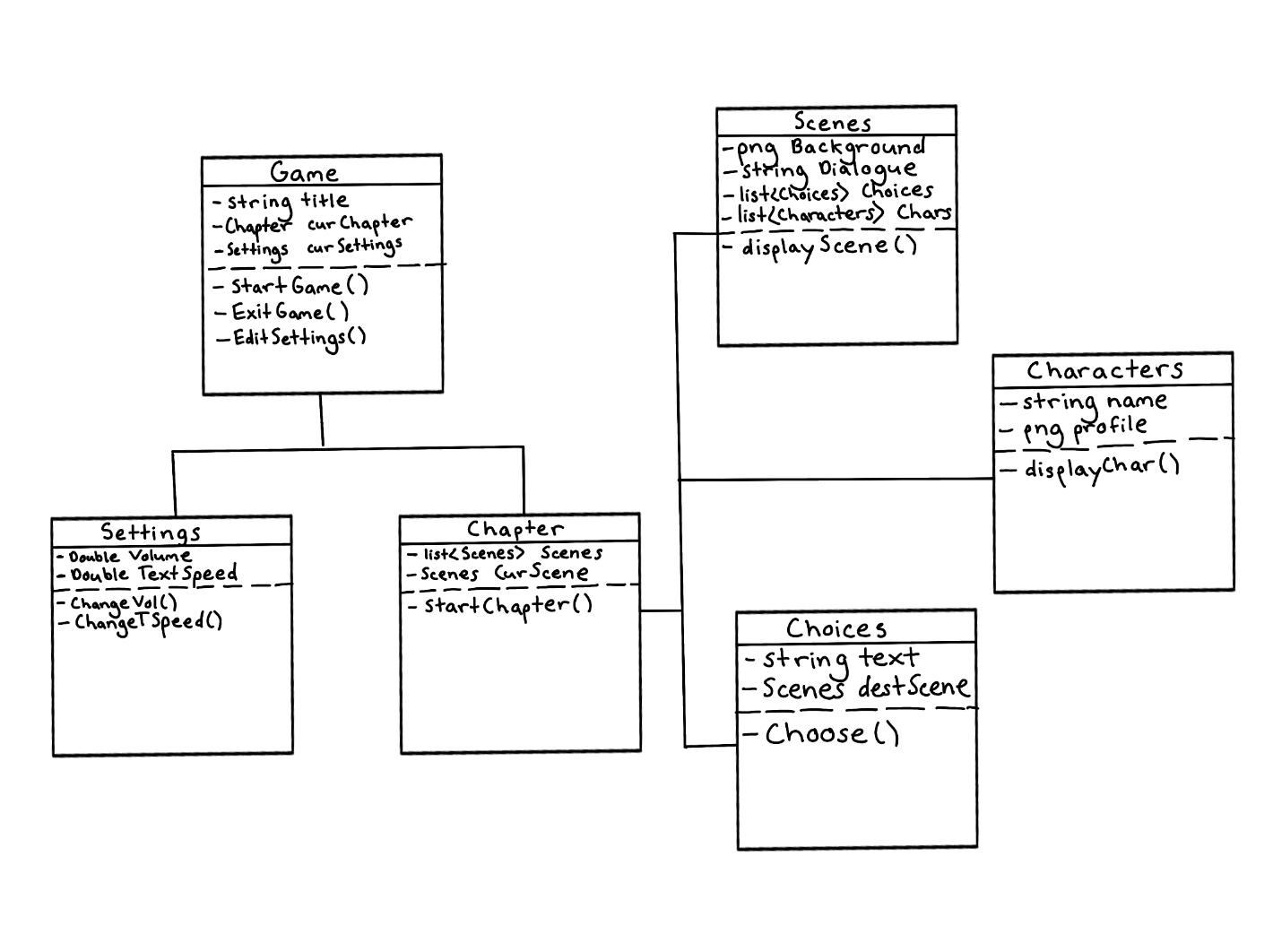
1. The system should be able to run on both Windows and Mac operating systems.
2. Game should run within 30 seconds.
3. Game should be accessible offline.
4. The game will have minimal crashes, errors, or bugs
5. Game should run smoothly with no delays or lag in loading visual assets.
6. User interface should be intuitive and easy to navigate regardless of user experience.
7. User data shall be kept separate and private from other users. (?)
8. Loading saves should take a maximum of 10 seconds under normal circumstances.

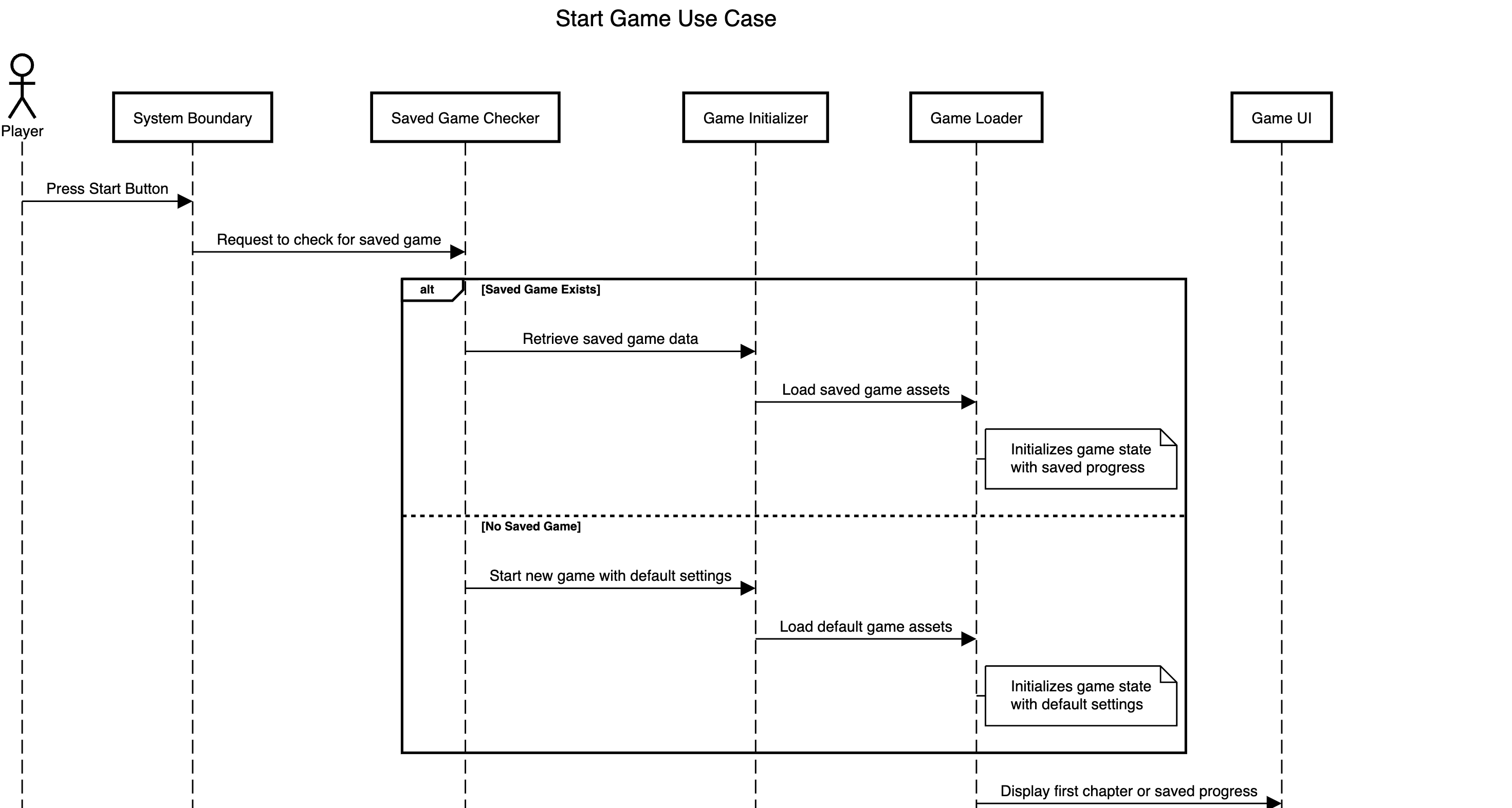
# Use Case Diagram (10 points)



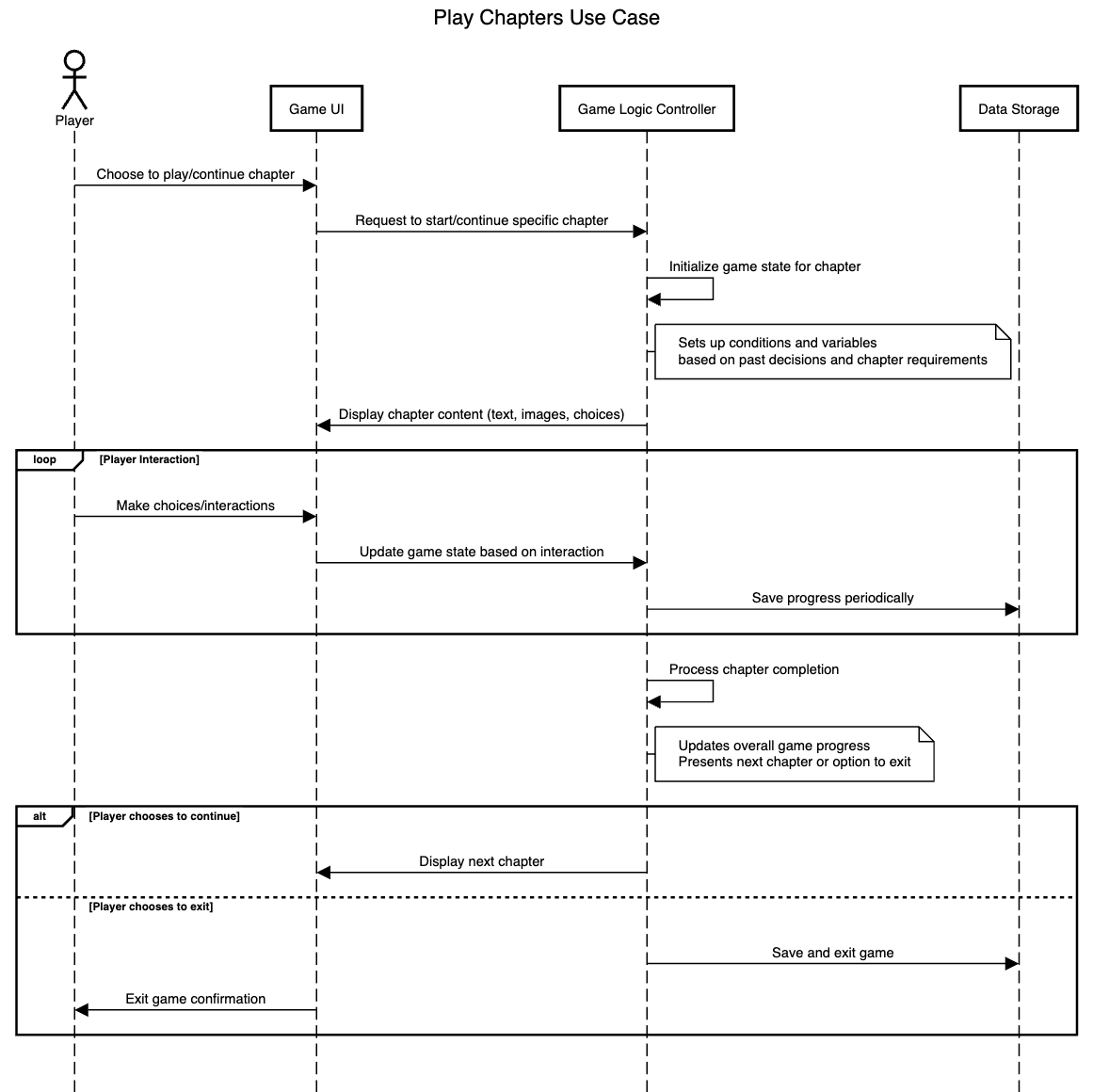
# Class Diagram and/or Sequence Diagrams (15 points)

Class Diagram:

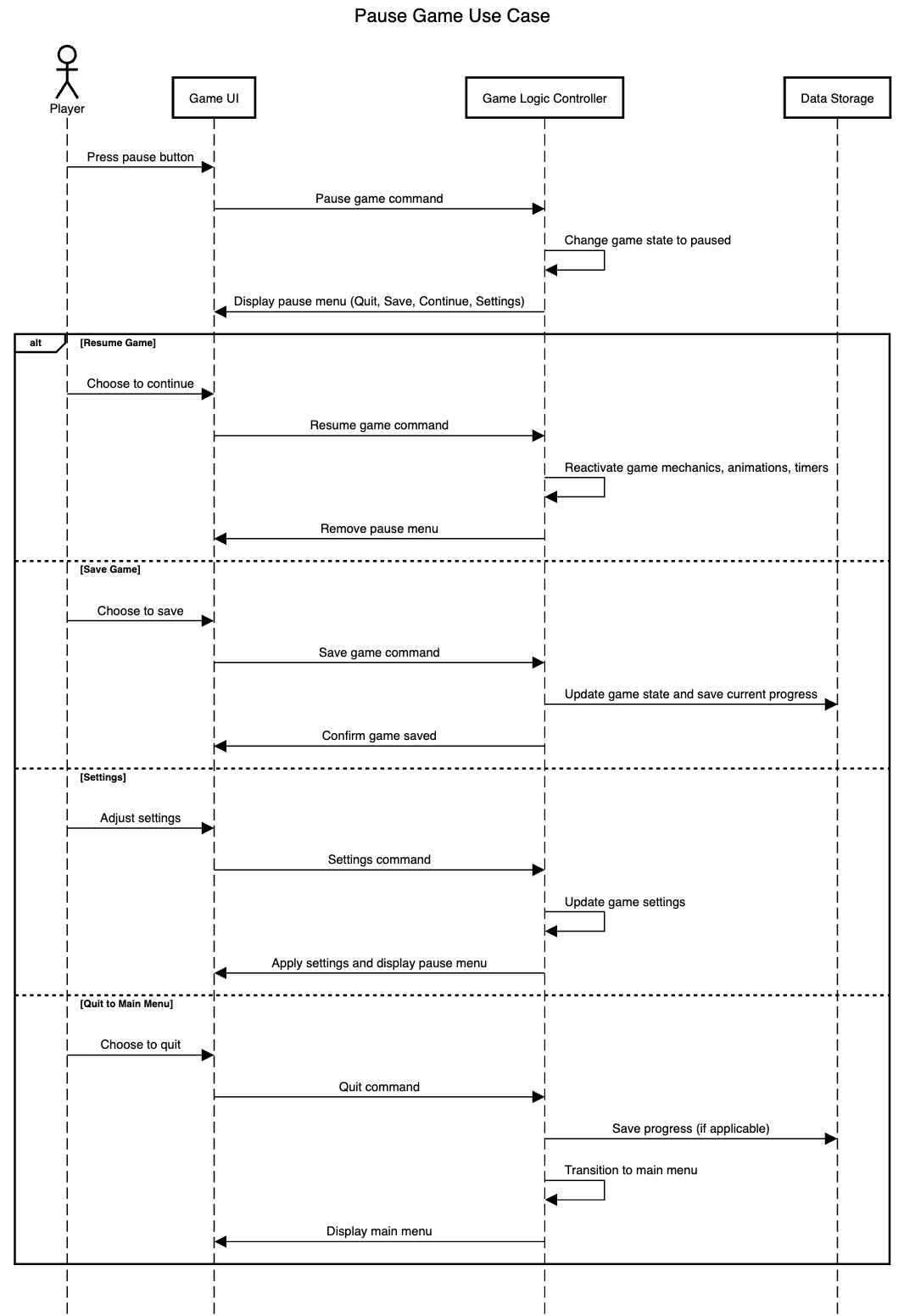
Start Game Use Case Sequence Diagram:



Play Chapters Use Case Sequence Diagram:



Pause Game Use Case Sequence Diagram:



# Operating Environment (5 points)

The software, developed on Ren’Py 8.2.0 (the latest official release), will operate on both Windows and MacOS.

On Windows, the software should be compatible with commonly used devices running Windows, such as desktops, laptops, and tablets. These devices may have different processors, such as Intel and AMD, and may have different storage and memory capacities. Ren’Py is able to run on Windows versions 7+ (Windows 7, Windows 8, Windows 8.1, Windows 10, Windows 11, etc.).

As for Mac OS, the software should be compatible across Apple devices such as MacBooks, Mac Pros, Mac Minis, etc. This should include Apple devices with different processors such as an Intel chip and the Apple silicon chip. The software is able to run on Mac OS X 10.10+ (MacOS Catalina, MacOS Mojave, MacOS High Sierra, etc.).

The software must be able to utilize visual assets and audio libraries necessary for rendering music. The software must also be compatible with input devices commonly used in both operating systems, such as keyboards and mice.

# Assumptions and Dependencies (5 points)

* Assuming that Ren’Py continues to release new updates, this may affect compatibility with past versions of Windows and MacOS operating systems.
* Ren’Py may be limited in its ability to create custom minigames that align with the game’s story given that it is a visual novel development platform.
* Libraries used to render music for the game depend on their continued compatibility with Ren’Py software.
* It is assumed that all creative assets, including art and music, are original or properly licensed for legal use.