Video Game

Software Requirements Specification

1.0

2/4/2022

Joseph Traglia
Programmer
Brian Kollgaard
Lead Designer
Emanuel Luna
Lead Programmer

Revision History

Date	Description	Author	Comments
2/4/22	Writing Introduction (Section 1)	All members	We completed the Purpose, Scope, Definitions, and Overview of the Introduction section.

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
Brian Kollgaard	Brian Kollgaard	Mr. Kollgaard	2/4/22
Emanuel Luna	Emanuel Luna	Mr. Luna	2/4/22
Joseph Traglia	Joseph Traglia	Mr. Traglia	2/4/22

Table of Contents

1. Introduction **1**

1.1 Purpose 1
1.2 Scope 1
1.3 Definitions, Acronyms, and Abbreviations 1
1.4 Overview 1

2. General Description **1**

2.1 Product Perspective 1
2.2 Product Functions 2
2.3 Users and Characteristics 2
2.4 General Constraints 2
2.5 Assumptions and Dependencies 2
2.6 Operating Environment 2

3. Specific Requirements **2**

3.2 Functional Requirements 3
3.2.1 <Functional Requirement or Feature #1> 3
3.2.2 <Functional Requirement or Feature #2> 3
3.3 Use Cases 4
3.3.1 Use Case #1 4
3.3.2 Use Case #2 4
3.4 Non-Functional Requirements 4
3.5.1 Performance 5
3.5.2 Reliability 5
3.5.3 Availability 5
3.5.4 Security 5
3.5.5 Maintainability 5
3.5.6 Portability 5
3.5 Design Constraints 5
3.6 Logical Database Requirements 5
3.7 Other Requirements 5

4. Analysis Models **5**

4.1 Sequence Diagrams 6

5. Change Management Process **6**

A. Appendices **6**

A.1 Appendix 1 7
A.2 Appendix 2 7

List of Figures

Figure 1: Data Flow Diagram Example 1 6
Figure 2 Data Flow Diagram Example 2 6

List of Tables

Table 1: Use case 1 4

1. Introduction

This section gives information about the [PROJECT CHART] game version 1.0 and all functional and non-functional aspects of the game. This document is intended for members of the project team that will check all functions that are implemented in the system. Unless noted by the project team, all requirements specified in this section are of high priority and committed for the release of version 1.0.

1.1 Purpose

The purpose of this document is to inform the software engineer of the requirements for the "video game" software. It will show the different functions of all the systems being used in the game. It will explain how each system works with each other in tandem such as how the user interface coincides with the back-end systems. The customer/intended audience will use this document to see if they approve of its contents and as an allusion to the first version of the game.

1.2 Scope

The scope of this project will be to create a short-term video game, which is unnamed at the moment, without utilizing a great amount of system memory. This video game will aim to deliver the end-user with an enjoyable computer video game experience by including five to fifteen minutes of pre-made levels where the player's end goal is to reach the end. This game will not utilize a formalized game engine, include three-dimensional graphics/gameplay, or include an immense amount of levels. The goals of this project are to deliver a gratifying gaming experience to the end-users, finish the project before its planned release date, and operate within the budget allotted for the development team. The development team is striving to outsell its competitors on the market and generate an overall profit on the video game.

1.3 Definitions, Acronyms, and Abbreviations

3D Modeling/Gameplay refers to the creation of a three-dimensional object inside of simulated software. This will be used in the project to provide an interface that the end-user is able to interact with when playing the game.

Gameplay is the end-user interacting with the game, clicking specific buttons that cause an action in the game to occur.

Graphical User Interface (GUI) is a visual way of interacting with a computer using items such as windows, icons, and menus, used by most modern operating systems. The project will utilize multiple GUIs, which will allow the end-user to navigate menus, modify options, and play the game itself.

Level is the total space available to the end-user (Consumer interacting with the software) during the course of completing a discrete objective. The project will consist of pre-made levels as well as levels produced by the development team.

Menu is an option that the end-user is given the ability to change, only referring to the volume of music, sound effects, or brightness of the GUI. The project will consist of multiple menus to assist the user when interacting with the GUI.

Web-based refers to applications or systems that run on web browsers. The project will not consist of a web-based application, instead, the project will feature a game that could run directly on the end user's desktop.

System Memory refers to the random access memory (RAM) a computer possesses. The project will utilize a minimal amount of memory in order to maximize other aspects of the game.

1.4 Overview

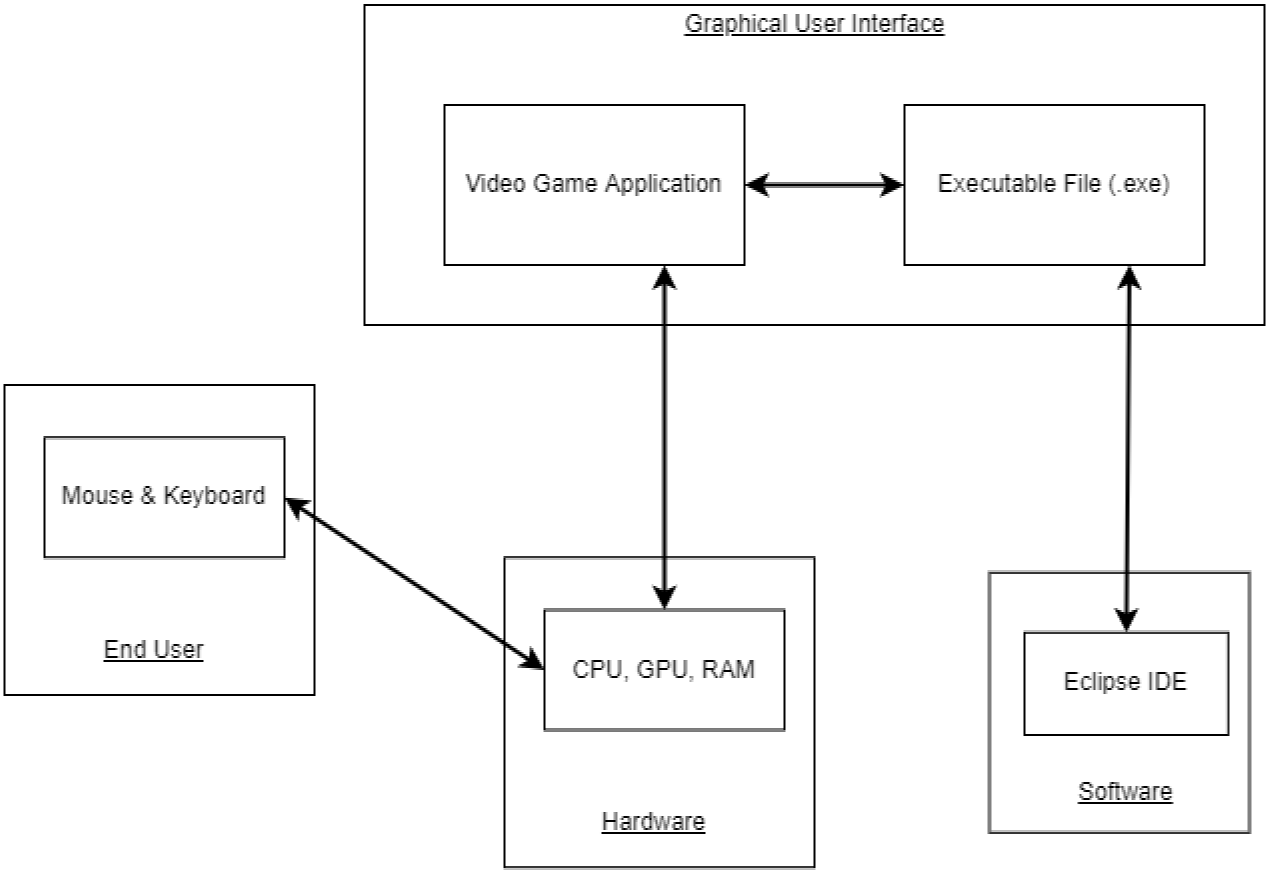
- The rest of this document contains five other sections. The next section contains a General Description of the product and factors that can or will affect it in the growing development.
- The third section is about specific requirements in great detail about the different system interfaces. Different languages will be used in order to make it understandable for different readers of varying literacy.
- The fourth section will go over and comprehensively explain different sequence diagrams and display visual examples.
- The fifth section will go over the procedure to renovate the SRS in case of needed changes as well as how and who can submit changes and how they will be contented.
- The final section is a simple appendix.

2. General Description

This section will talk about the entire system. The system will be described in detail including how it works with other systems. The stakeholders will be shown who these products are and how they contribute and interact with the system. Finally, the constraints and assumptions for the systems will be presented.

2.1 Product Perspective

The [VIDEO GAME TITLE] game consists of a GUI and any personal computer with a keyboard and mouse compatible with the GUI (Laptop and trackpad for laptop users). The GUI will provide the game included while the keyboard will provide the controls to interact with the game. In order to play this game, an installation of the software is required through the Windows or macOS operating systems.



2.2 Product Functions

The overall purpose of the video game will be to produce a computer-based video game that integrates multiple different functions. The major functions that the video game will contain are an interactive user experience containing a graphical user interface, movement using a keyboard that will allow the end-user to control their player, and player animations that allow walking, running, and jumping. Other important functions of the game include collision detection on walls, floors, and other objects, parallax scrolling to ensure the user is progressing through the level, a main menu containing a start and quit option, a pause menu that allows the end-user to return to the main menu, background music that enhanced the user experience, a title screen, and mouse support in the title screen and menus.

2.3 Users and Characteristics

Age groups 6 - 12 This age group is considered young to older children who are still not yet in high school. This age group will find fun gameplay and visuals to keep their attention span while playing. This group may not fully understand the story or symbolism used that deeply compared to other groups but that should not impact their full enjoyment and usage. The gameplay will be enough to carry their interest throughout their playtime.

Age groups 13 - 17 This age group is considered young adults who want more out of their media then when they were a child. The story and symbolism will keep their group entertained throughout their playthrough. They should not need a lot of time to fully understand the systems used in order to engage in the experience.

Age groups 18+ This age group is the adult group who are considered to be highly critical with the media they consume. They will enjoy breaking out the symbols on a more technical level than the younger audiences. They should require no major tutorial/training to fully understand and should be able to learn diegetically as they play. This group will be the hardest to satisfy as again they are the most critical with the entertainment they consume.

2.4 General Constraints

- The lack of using a major game engine may limit the efficiency and complexity of the system used.
- The use of Java may not be the best for coding video games and could slow down development.
- The lack of high-quality professional experience and assets may hinder the front-end systems, which could degrade the end-user's experience.

2.5 Assumptions and Dependencies

One assumption about the video game will be that the end-user is running the game on a desktop computer or laptop with adequate resources on their system. This application will not be available on mobile devices. If the end-user is unable to meet this requirement, it is probable that the game will not be able to run or the user will have an unenjoyable experience. Another assumption about the product is that the user possesses basic knowledge of PC gaming including how to properly utilize a keyboard and mouse and navigate menus. Failure to meet this assumption will render the game unplayable and an unenthusiastic experience for the end-user.

2.6 Operating Environment

The operating environment will work on Windows 10 and macOS Monterey computers only.

3. Specific Requirements

This segment contains every requirement and function needed for the system. Furthermore, each requirement will be described in detail.

3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.2.1 Resolution Change Feature

3.2.1.1 Description and Priority

The user should be able to change the resolution of the visuals in game in case of performance issues. This feature is relatively medium to low priority. The game should run fine on low end systems so the feature would really benefit people with extra power who want to play at a higher resolution.

3.2.1.2 Stimulus/Response Sequences

The user will click on the settings icon or press the keybind for the settings menu. Once the settings menu is opened they will have to navigate to the setting labeled "resolution." They can navigate via mouse movement or with keyboard movement. Once selected the game will open a menu with different resolution options including 480p, 720p, 1080p each with full screen and windowed options for each. The user will select one option then select the “apply” button where the menu will automatically close and the game will resume with the settings the user selected.

3.2.1.3 Functional Requirements

3.2.1.3.1 REQ-1: For the user to select the resolution the menu system has to be working and recognize user input so that the user can select the resolution.

3.2.1.3.2 REQ-2: In order for the resolution to be changed there needs to be a separate system where the game will automatically scale the size of in-game objects.

3.2.2 User Tutorial Replay Feature

3.2.2.1 Description and Priority

If the user forgets a function or feature of the game they should be able to go into the settings menu to review the tutorial. It is a high priority feature which would highly benefit (7/9) the player from getting frustrated and be relevely lost cost as the tutorial is just being reused so nothing new needs to be created (cost 2/9).

3.2.2.2 Stimulus/Response Sequences

The user will click on the settings icon or press the keybind for the settings menu. Once the settings menu is opened they will have to navigate to the setting labeled “tutorial.” They can navigate via mouse movement or with keyboard movement. Once selected the game will replay the tutorial for the user.

3.2.2.3 Functional Requirements

3.2.2.3.1 REQ-1: In order for the user to even get to the tutorial, a tutorial must be created first that can be drawn from.

3.2.2.3.2 REQ-2: For the user to select the tutorial the menu system has to be working and recognize user input so that the user can select the tutorial.

3.2.2.3.3 REQ-3: In case the system fails to bring something up to the user, namely the tutorial, the system should be able to output that the tutorial failed to load to the user.

3.2.3 - Using a Keyboard and Mouse to Play the Game

3.2.3.1 Description and Priority

The user will only be able to use a keyboard and mouse in order to play the game. It is a high priority feature that is necessary for the game to function (9/9) because with it the player will not even be able to start the game. The cost will be relatively high as adding keyboard, mouse, and controller support will take a lot of time (cost 5/9).

3.2.3.2 Stimulus/Response Sequences

Once the game opens, the user will be automatically told that the game only supports keyboard and mouse. In a later update, the user will be able to click the settings icon. Once the settings menu is opened, the user can then click “Control Configurations” and see a new button named “Connect a controller”. The user then can connect their controller through Bluetooth and should appear a screen that says “Controller connected”.

3.2.3.3 Functional Requirements

3.2.3.3.1 REQ-1: In order for the game to use a keyboard and mouse, the program must automatically recognize the keyboard and mouse the user is using to play the game.

3.2.3.3.2 REQ-2: In order to connect a controller to the game, the settings menu must work correctly and connections to a controller that is supported must be implemented efficiently and correctly.

3.2.4 Saving the Score While Playing the Game

3.2.4.1 Description and Priority

When playing the game, the user will see a counter displaying the number of points they have gained. When the game is saved, the score will be saved and the user can view the score the next time they enter the game.

3.2.4.2 Stimulus/Response Sequences

Once the game opens, the user will see the menu screen, and press "New Game" or "Continue" depending on the progress that was made. The player can choose either and while playing the level, will they receive points. The user can then save the game by opening the menu and clicking "Save Game". When the game is saved, all points will be saved before and right at the point of clicking "Save Game".

3.2.4.3 Functional Requirements

3.2.4.3.1 REQ-1: The game must have a saving feature that can save the progress made by the user.

3.2.4.3.2 REQ-2: The menu must work to correctly save the game's progress.

3.2.5 Adjusting Volume and Sound Effects

3.2.5.1 Description and Priority

The development team will strive to implement functions that will improve the user's experience playing the video game. Therefore, implementing how to change the volume and sound effects independently in the game is a feature of medium priority. After this function has been implemented, the end-user should be able to edit the sound output coming out of their computer, however, this change in sound should not affect the actual game volume (music) or sound effects.

3.2.5.2 Stimulus/Response Sequences

Stimulus: End-user pauses the game.
Response: System displays a menu giving the end-user the option to continue, change settings, or quit.
Stimulus: End-user navigates to the change settings option and selects "Enter."
Response: System displays volume settings, giving the end-user the option to change either "Game Volume" or "Sound Effects."
Stimulus: End-user selects the specific volume they would like to change.
Response: System allows the end-user to either increase or decrease the volume of their choice.
Stimulus: End-user modifies the volume until they are content with the audio output level.
Response: System reflects the changes initiated by the end-user.
Stimulus: End-user continues playing game after adjusting volume.

3.2.5.3 Functional Requirements

These are the requirements the user needs to meet in order to change the volume and sound effects within the game.

3.2.5.3.1 REQ-1

TITLE: Download computer application

DESC: The end-user should be able to download the computer application either through a website or the development team will provide them with the necessary files. The game is free to download.

RAT: In order for a user to download the computer application.

DEP: None.

3.2.5.3.2 REQ-2

TITLE: Functioning keyboard and mouse

DESC: The end-user will be able to navigate the menus of the game as well as play the game if they have a functioning keyboard and mouse. If not, no inputs can be taken from the user.

RAT: In order for a user to play the game and navigate menus.

DEP: None.

3.2.5.3.3 REQ-3

TITLE: Functioning audio drivers

DESC: The end-user should have functioning audio drivers on their computer's motherboard or a separate sound card in order to hear the audio output from the game.

RAT: In order for a user to hear the music and sound effects of the game.

DEP: None.

3.2.5.3.4 REQ-4

TITLE: End-user interaction - playing the game

DESC: Since the user has downloaded the game and is able to execute the application file, they should be at the title screen, which will present them with the option to start the game.

RAT: In order for a user to start playing the game.

DEP: REQ-1, REQ-2

3.2.5.3.5 REQ-5

TITLE: End-user interaction - adjusting the volume

DESC: Once the end-user has started the game, they will decide for themselves if the audio is too loud or too quiet. After making this determination, the end-user can pause the game and navigate to the audio settings.

RAT: In order for the user to obtain an enjoyable experience with their preferred sound volume.

DEP: REQ-1, REQ-2, REQ-3, REQ-4

3.2.6 Saving In-Game Progress

3.2.6.1 Description and Priority

The development team will strive to implement functions that will improve the user's experience playing the video game. Therefore, implementing a function to allow the end-user to save their in-game progress is a low priority. After this function has been implemented, the end-user should be able to save their progress at any major checkpoint within the game. This function is necessary in case the user does not finish the game completely in one attempt, they will not have to start from the beginning.

3.2.6.2 Stimulus/Response Sequences

Stimulus: End-user starts the game.
Response: System displays the game to the user.
Stimulus: End-user surpasses a checkpoint within the level.
Response: System saves the end-users progress at the certain checkpoint within the level. A banner will be displayed to the user to notify them that this action has occurred.
Stimulus: End-user reads the message and continues to play the game.

3.2.6.3 Functional Requirements

These are the requirements the user needs to meet in order to save their progress within the game.

3.2.6.3.3 REQ-1

TITLE: Reaching a checkpoint within the level

DESC: The end-user should reach a point within the level where they come across a checkpoint using their directional keys on the keyboard. At this point in the level, the game should save the user's progress into memory. This will ensure the user does not have to start from the beginning of the game if they do not complete a level in its entirety.

RAT: In order for a user to not repeatedly start from the beginning of a level.

DEP: None.

3.2.6.3.4 REQ-2

TITLE: Improperly saving end-user progress to memory (error condition)

DESC: Once the user has reached a checkpoint within the game, the game will save the end-users progress to memory. If this function is performed improperly, the game will display a banner to the user that an error has occurred when trying to save progress.

RAT: None.

DEP: None.

3.2.6.3.5 REQ-3

TITLE: Saving user progress to memory

DESC: Once the user reaches a checkpoint within the level the game will save their progress. The save file will be stored in memory, therefore, if the user exits the game before the next checkpoint is reached. They will continue the game from the last checkpoint they reached in the level.

RAT: None.

DEP: REQ-1

3.3 Use Cases

3.3.1 Use Case #1

Use Case Name	Working Menu
Reference	Section 3.2.2, Section 3.2.1
Trigger	The user clicks the menu or presses the menu button on their controller.
Actor's Goal	To open the menu.
Precondition	A settings icon is visible or they have a settings button on their controller.
Basic Path	1. (--->)The user moves the mouse to the top right of the screen where the settings icon is located. 2. (--->)The user clicks on the settings icon 3. (<---)The menu opens showing various selectable options each selectable with the mouse. 4. (--->)The user can close the menu by clicking on the red X on the top right of the settings menu to close it. 5. (<---)Menu closes.
Alternative Paths	1. (--->)The user can press whatever keyboard or controller bind they have for the settings menu. The default is ESC for the keyboard and the + button for controllers. 2. (<---)The menu opens 3. (--->)The user can navigate it via the keyboard and select an option with the action button. If using a controller they can navigate with the d-pad and select an option with the action button. 4. (--->)The user can close the menu by pressing ESC or + again. 5. (<---)The menu closes.
Postcondition	The menu is open.
Exception Paths	The user can close the menu at any time.
Other	The menu once opened can lead to a difference in systems by the user's choice

Table 1: Use case 1

3.3.2 Use Case #2

Use Case Name	Adjusting Volume
Reference	Section 3.2.5
Trigger	The user pauses that game using "ESC" on their keyboard.
Actor's Goal	To adjust the sound of the game successfully.

Precondition	The pause menu consists of a “settings” option that allows the user to adjust sound output.
Basic Path	1. (--->) The user pauses the game. 2. (<---) The pause menu is displayed. 3. (--->) The user navigates to the “Settings” option on the pause menu. 4. (<---) Two options for adjusting sound are displayed. 5. (--->) The user selects to either change the sound output of the music or the sound output of the game’s sound effects. 6. (<---) The system accurately reflects these changes created by the user. 7. (--->) The user continues playing the game with the sound output adjusted to the specified levels.
Alternative Paths	1. (--->) The user pauses the game. 2. (<---) The pause menu is displayed. 3. (--->) The user navigates to the “Settings” option on the pause menu. 4. (<---) Two options for adjusting sound are displayed. In addition, there is a third option that allows the user to reset the audio settings to default. 5. (--->) The user selects to reset their sound settings to the default audio output. 6. (<---) The system accurately reflects these changes created by the user. 7. (--->) The user continues playing the game with the default audio settings.
Postcondition	The user is able to play the game at an enjoyable volume level.
Exception Paths	The user adjusts the sound output on their computer or headphones using their operating system instead of the in-game menu.
Other	None.

Table 2: Use case 2

3.3.3 Use Case #3

Use Case Name	Recognition of Keyboard and Mouse or Controller
Reference	Section 3.2.3, Section 3.2.3.1, Section 3.2.3.2
Trigger	The program will automatically connect the keyboard and mouse (trackpad for laptops) when the game is opened. A controller can be connected through the menu.
Actor's Goal	To connect a keyboard and mouse or a controller to the game.
Precondition	The keyboard and mouse are connected to the computer.
Basic Path	1. The user must have a keyboard and mouse connected to their computer. 2. The application to open the game must be opened. 3. The keyboard and mouse will be recognized by the computer.
Alternative Paths	N/A
Postcondition	The controller has been connected.
Exception Paths	The controller can get disconnected or run out of battery.
Other	N/A

Table 3: Use case 3

3.3.4 Use Case #4

Use Case Name	Working Resolution Change System
Reference	**3.2.1**
Trigger	The user selects the "change resolution" button on the menu.

Actor’s goal	To change the resolution settings.
Precondition	The menu is opened.
Basic Path	1. (--->)The user moves the mouse to the top right of the screen where the settings icon is located. 2. (--->)The user clicks on the settings icon. 3. (<---)The menu opens showing various selectable options each selectable with the mouse. 4. (--->)The user moves the mouse to the “Change resolution” button and clicks on it. 5. (--->)The user clicks on the option “480p”, “720p”, or “1080p” if they wish to change the resolution and then click on apply. 6. (--->)The user can also select “windowed” or “full screen” options and then click on apply. 7. (<---)The menu will automatically close with the newly applied resolution being selected.
Alternative Paths	After step 4 the user can instead choose different options. 5. (--->)The user can instead choose the option for “Recommended” which will evaluate the user's PC and automatically choose a resolution that will give them the most optimal performance. 6.(<---)The user does not need to click apply as the menu will automatically close when the “Recommended” option is selected as well as the windowed or full screen being automatically selected and applied.
Postcondition	The menu is closed and the selected resolution or screen option is being used.
Exception Paths	The user can close the menu without clicking on apply and the choice will not be put into place.
Other	If the user can not select multiple resolutions at once, the system does not allow that, once an option is selected the previously chosen option will be unselected.

Table 4: Use case 4

3.3.5 Use Case #5

Use Case Name	Reaching/Saving Level Checkpoints
Reference	Section 3.2.6
Trigger	The user reaches a checkpoint (savepoint) in the level. This could be represented by a flag, platform, etc.
Actor's Goal	To reach a certain point in the level where they will not have to start from the beginning of the level upon restarting the game.
Precondition	There is an object in the level that will notify the game to save the user's progress at a specific point.
Basic Path	1. (--->) The user progresses through the game's level. 2. (<---) A checkpoint is visible on the level via an object. 3. (--->) The user reaches the point on the level where the checkpoint is located. 4. (<---) The game notifies the user they have reached a checkpoint by displaying a banner at the top of the screen. 5. (--->) The user will not have to start from the beginning of the level upon resetting the game.

Alternative Paths	1. (--->) The user progresses through the game's level and pauses the game. 2. (<---) The pause menu displays an option to save the game after the user has reached a certain checkpoint within the level. 3. (--->) The user selects to save the game. 4. (<---) The system displays a banner at the top of the screen notifying the user that the game has been saved successfully. 5. (--->) Upon restarting the game, the user will continue the level from the last reached checkpoint.
Postcondition	The user resumes the level from the last reached checkpoint, not the beginning of the level.
Exception Paths	The user never reaches a checkpoint on the level.
Other	None.

Table 5: Use case 5

3.3.6 Use Case #6

Use Case Name	Search Article
Reference	Section 2.2.1, Section 7.1
Trigger	The Reader assesses the Online Journal Website
Precondition	The Web is displayed with grids for searching

Basic Path	1. The Reader chooses how to search the Web site. The choices are by Author, by Category, and by Keyword. 2. If the search is by Author, the system creates and presents an alphabetical list of all authors in the database. In the case of an article with multiple authors, each is contained in the list. 3. The Reader selects an author. 4. The system creates and presents a list of all articles by that author in the database. 5. The Reader selects an article. 6. The system displays the Abstract for the article. 7. The Reader selects to download the article or to return to the article list or to the previous list.
Alternative Paths	In step 2, if the Reader selects to search by category, the system creates and presents a list of all categories in the database. 3. The Reader selects a category. 4. The system creates and presents a list of all articles in that category in the database. Return to step 5. In step 2, if the Reader selects to search by keyword, the system presents a dialog box to enter the keyword or phrase. 3. The Reader enters a keyword or phrase. 4. The system searches the Abstracts for all articles with that keyword or phrase and creates and presents a list of all such articles in the database. Return to step 5.
Postcondition	The selected article is downloaded to the client machine.
Exception Paths	The Reader may abandon the search at any time.

Other	The categories list is generated from the information provided when article are published and not predefined in the Online Journal database.

Table 6: Use case 6

3.4 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transactions shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc.).

3.5.1 Performance

3.5.2 Reliability

3.5.3 Availability

3.5.4 Security

<Example: The server on which the Online Journal resides will have its own security to prevent unauthorized *write/delete* access. There is no restriction on *read* access. The use of email by an Author or Reviewer is on the client systems and thus is external to the system. The PC on which the Article Manager resides will have its own security. Only the Editor will have physical access to the machine and the program on it. There is no special protection built into this system other than to provide the editor with *write* access to the Online Journal to publish an article.>

3.5.5 Maintainability

3.5.6 Portability

3.5 Design Constraints

<Specify design constrains imposed by other standards, company policies, hardware limitation, etc. that will impact this software project. Example, the software is required to have a login screen based on company policies.>

3.6 Logical Database Requirements

<Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc?>

3.7 Other Requirements

<Catchall section for any additional requirements that did not belong to the previous sections. If there are none, exclude this section>

4. Analysis Models

<List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS's requirements.>

4.1 Sequence Diagrams

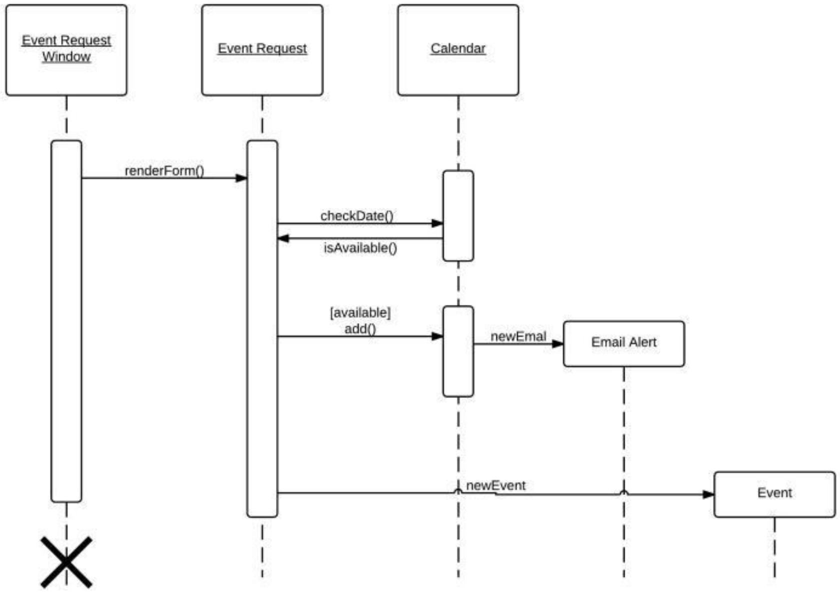


Figure 1: Data Flow Diagram Example 1

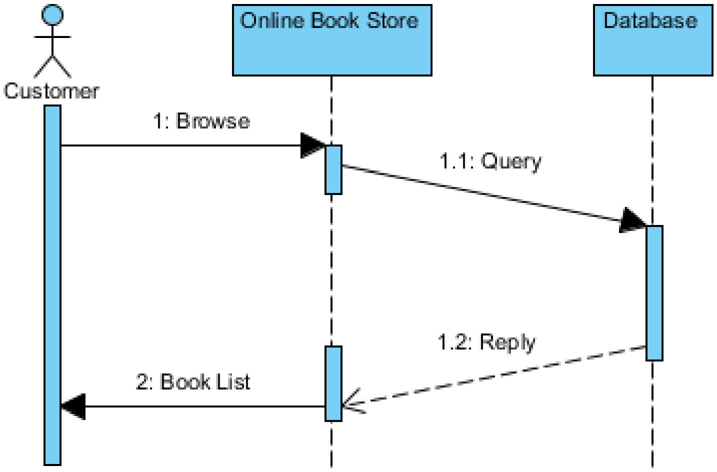


Figure 2 Data Flow Diagram Example 2

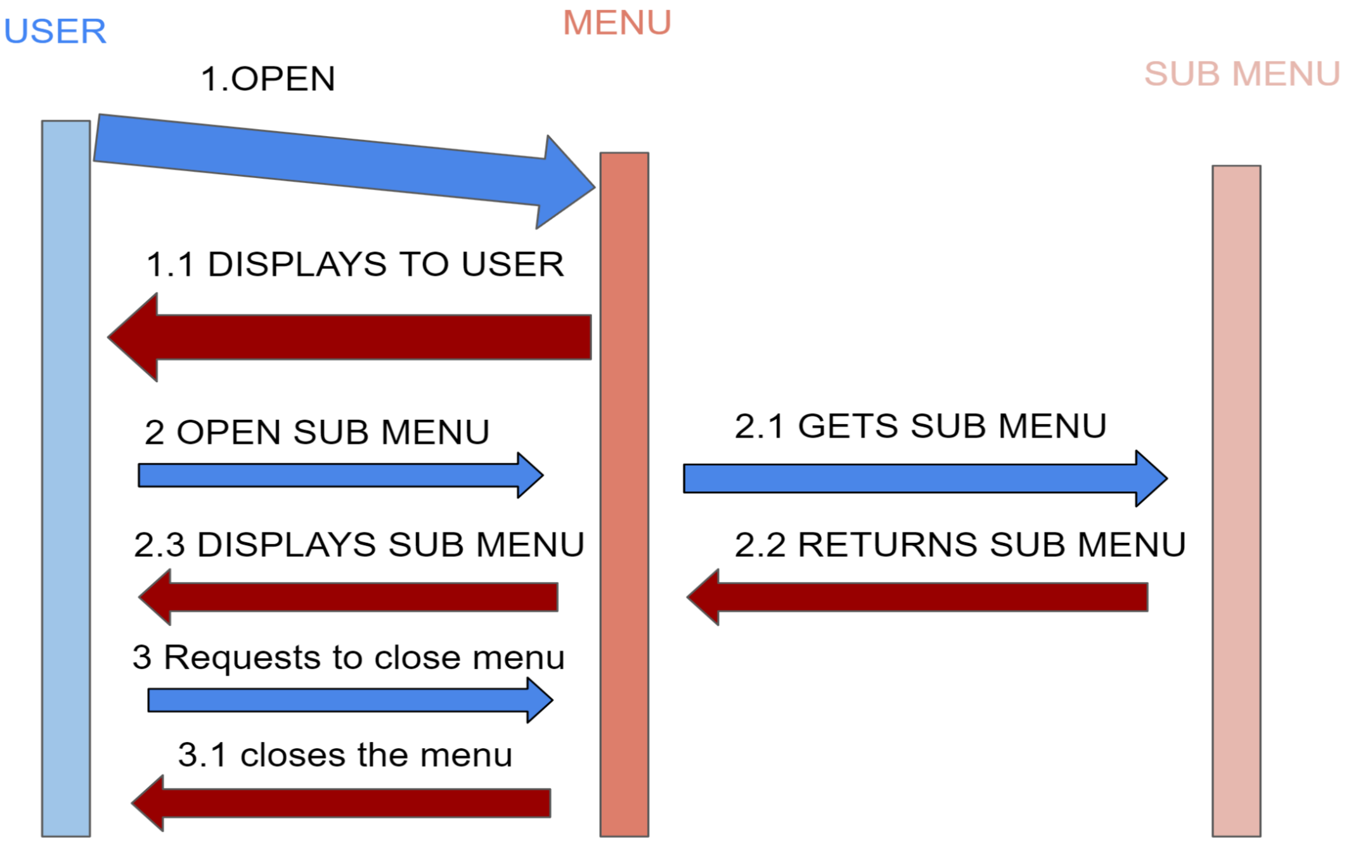


Figure 1: Menu System Flow Diagram

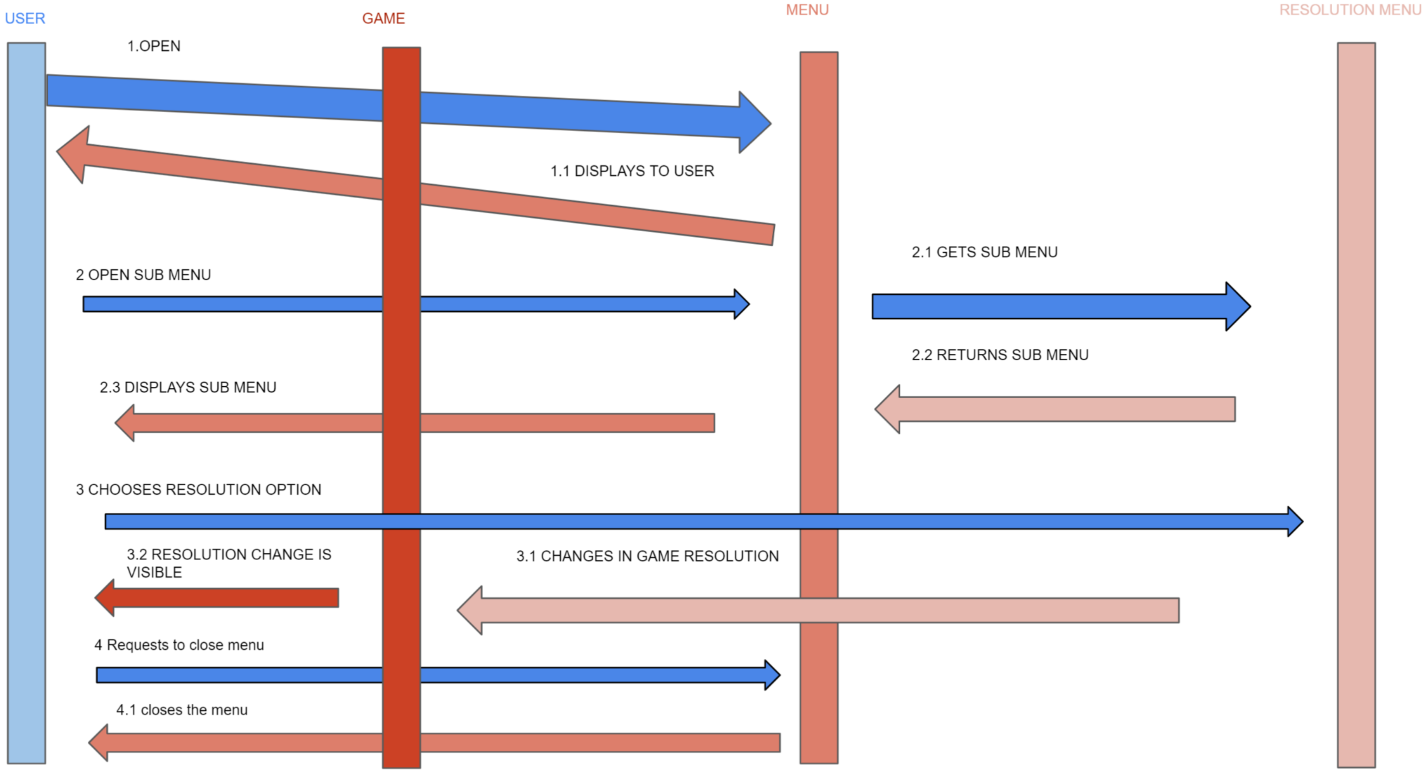


Figure 2: Game Resolution Change System

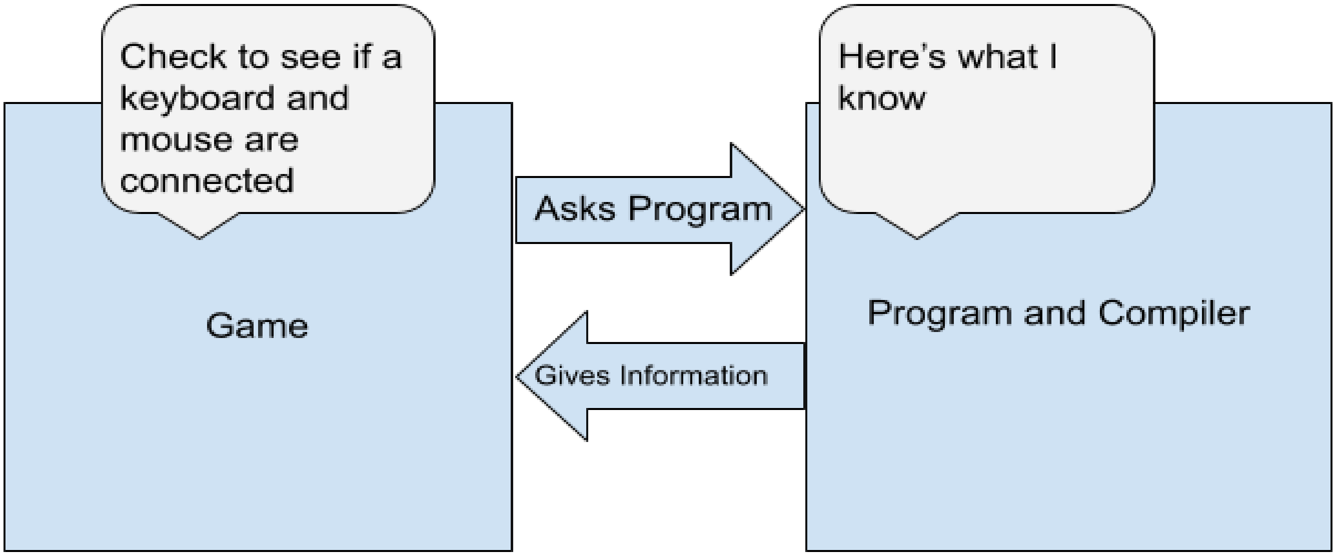


Figure 3: Keyboard and Mouse Connection Diagram

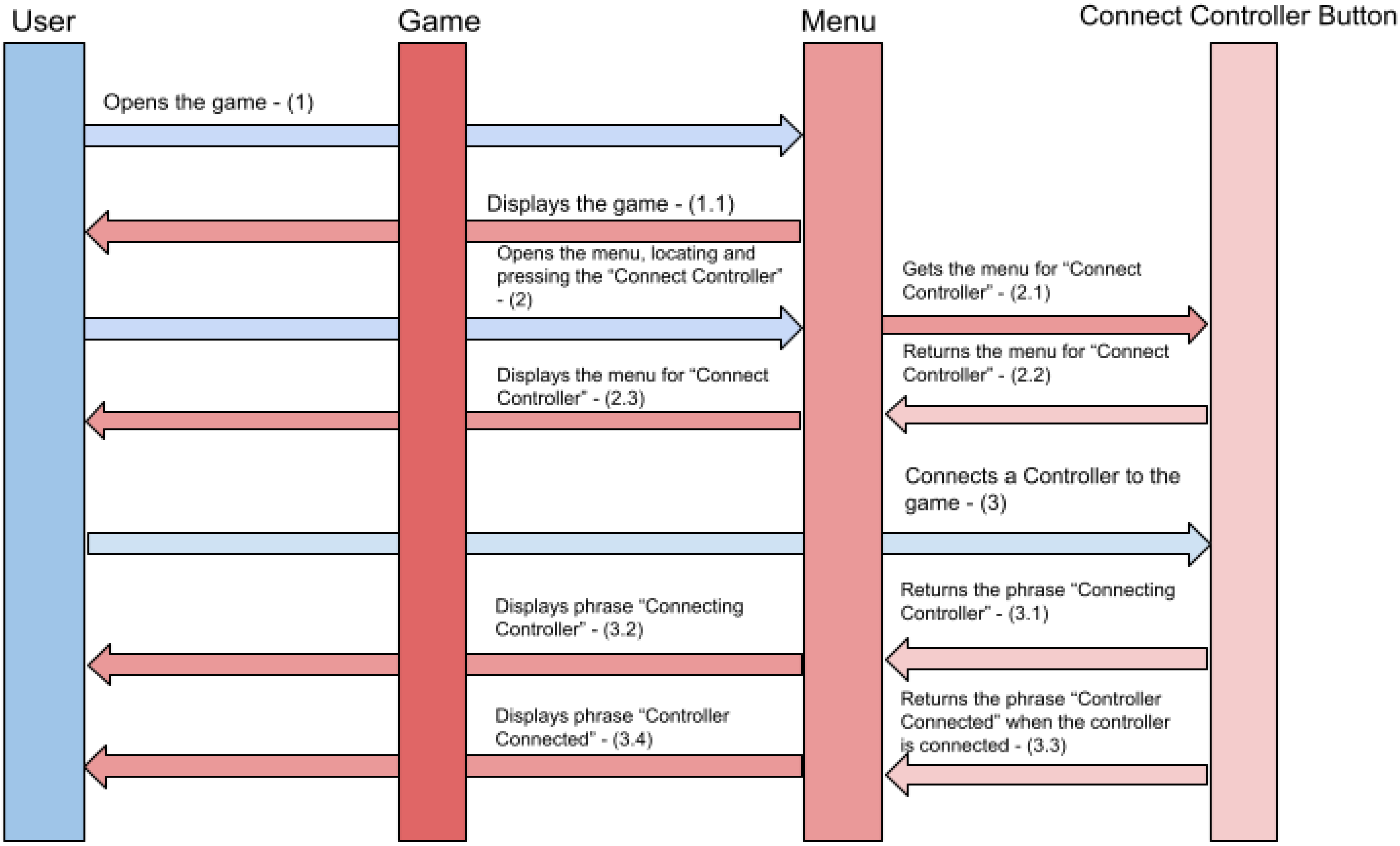

User
Game
Menu
Connect Controller Button
Opens the game - (1)
Displays the game - (1.1)
Opens the menu, locating and pressing the "Connect Controller" - (2)
Gets the menu for "Connect Controller" - (2.1)
Returns the menu for "Connect Controller" - (2.2)
Displays the menu for "Connect Controller" - (2.3)
Connects a Controller to the game - (3)
Returns the phrase "Connecting Controller" - (3.1)
Displays phrase "Connecting Controller" - (3.2)
Returns the phrase "Controller Connected" when the controller is connected - (3.3)
Displays phrase "Controller Connected" - (3.4)

Figure 4: Controller Connection Diagram

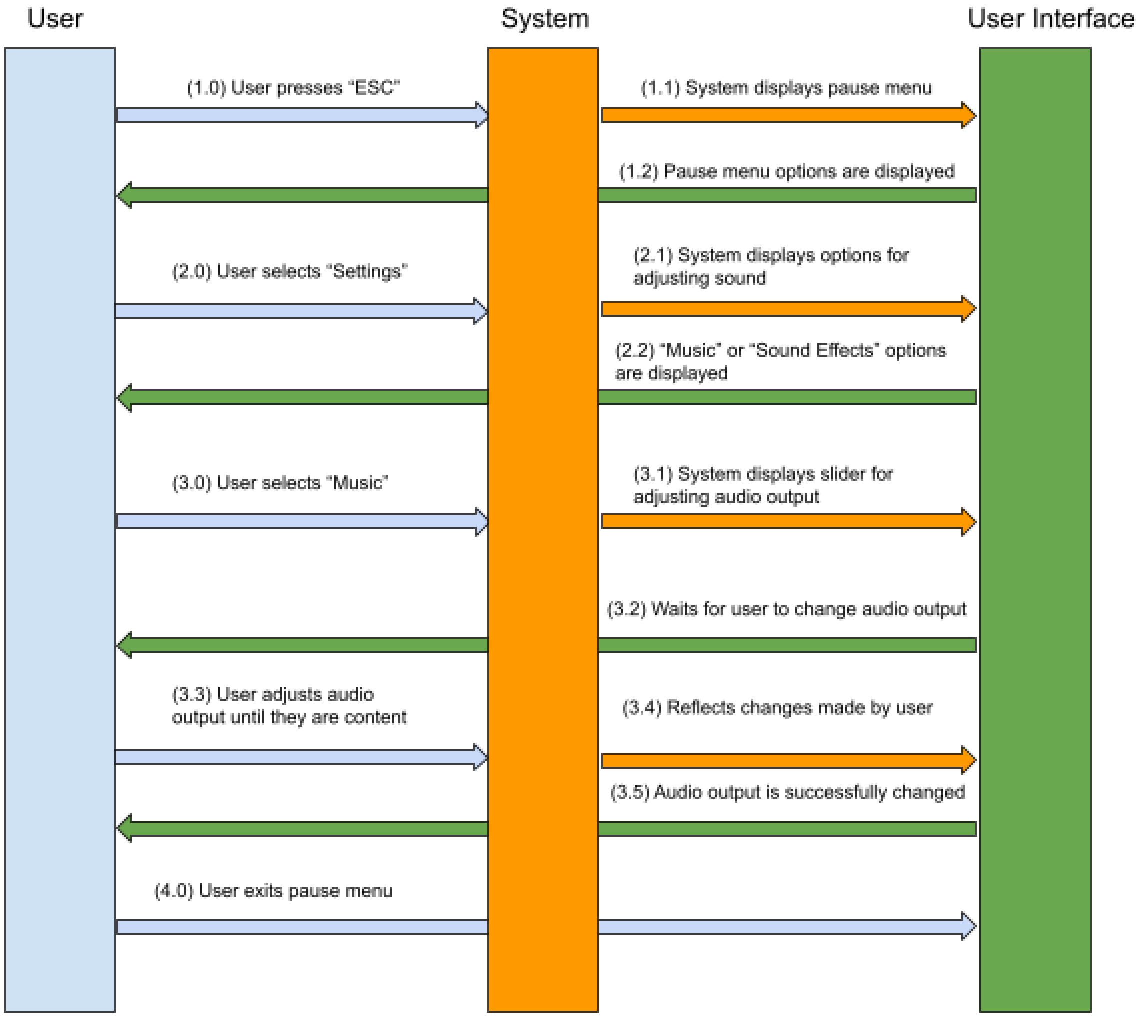

User
System
User Interface
(1.0) User presses "ESC"
(1.1) System displays pause menu
(1.2) Pause menu options are displayed
(2.0) User selects "Settings"
(2.1) System displays options for adjusting sound
(2.2) "Music" or "Sound Effects" options are displayed
(3.0) User selects "Music"
(3.1) System displays slider for adjusting audio output
(3.2) Waits for user to change audio output
(3.3) User adjusts audio output until they are content
(3.4) Reflects changes made by user
(3.5) Audio output is successfully changed
(4.0) User exits pause menu

Figure 5: Adjusting Volume Diagram

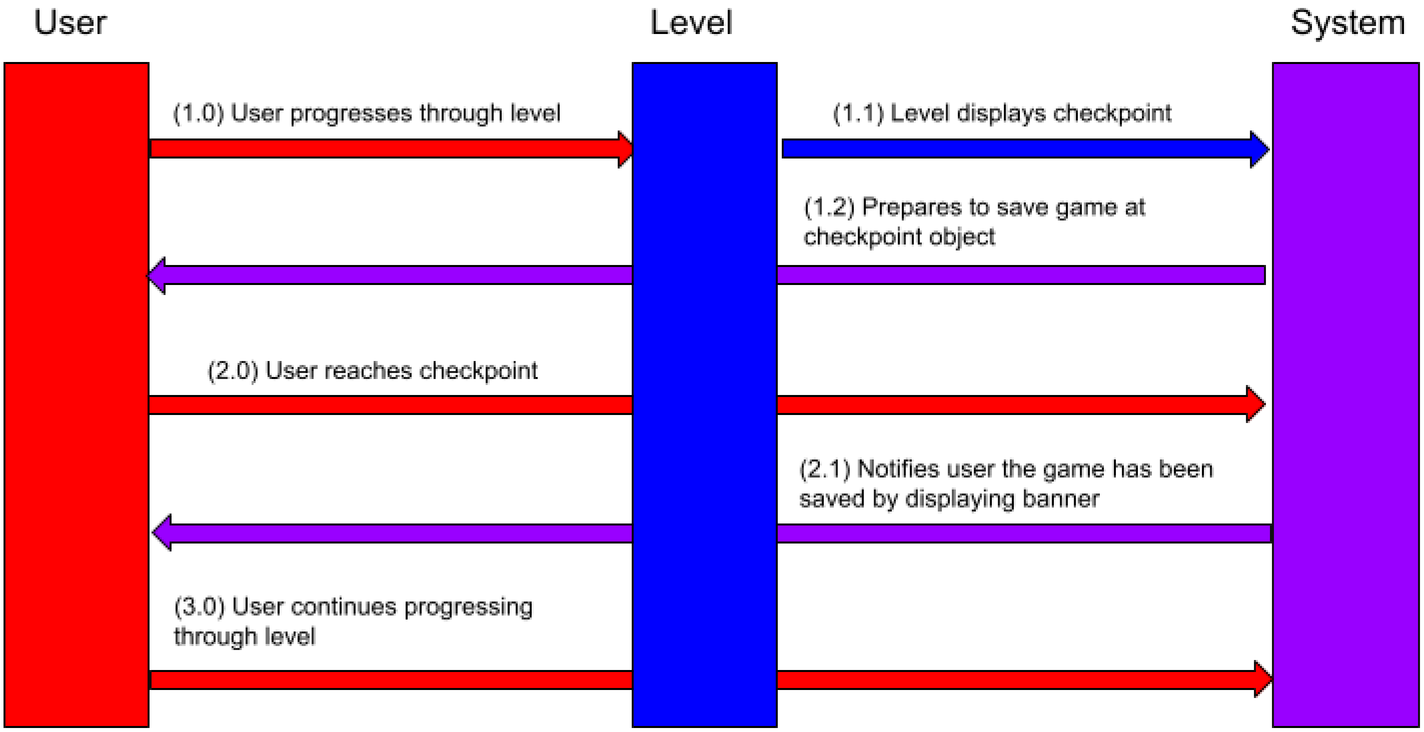

User
Level
System
(1.0) User progresses through level
(1.1) Level displays checkpoint
(1.2) Prepares to save game at checkpoint object
(2.0) User reaches checkpoint
(2.1) Notifies user the game has been saved by displaying banner
(3.0) User continues progressing through level

Figure 6: Saving Checkpoints Diagram

5. Change Management Process

<Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.>

References

A. Appendices

<Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS's overall set of requirements. Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.>

A.1 Appendix 1

A.2 Appendix 2