

**Computer Science and Engineering**

### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Archie-ology, A Video Game**

**System Requirements and Analysis Specification**

**(RAS)**

### Version 1.3

Document Number: RAS-004

Project Team Number Team A25

Project Team Members:

Jeffrey Qiu yq544

Juliet Ramdass jsr483

Sabrina Supapkooha sps450

**REVIEW AND APPROVALS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Printed Name and**  **Title** | **Function**  **(Author,**  **Reviewer,**  **Approval)** | **Date** | **Signature** |
| Jeffrey Qiu | Recorder/Moderator | February 13, 2019 | Jeffrey Qiu |
| Juliet Ramdass | Reader/Author | February 13, 2019 | Juliet Ramdass |
| Sabrina Supapkooha | Inspector | February 13, 2019 | Sabrina Supapkooha |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**REVISION LEVEL**

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision Number** | **Purpose** |
| October 4, 2018 | Version 1.0 | Initial Release |
| October 19, 2018 | Version 1.1 | Fix previous defects, add section 7, functional requirements |
| November 20, 2018 | Version 1.2 | Fix previous defects, add non-functional requirements, analysis |
| February 13, 2019 | Version 1.3 | Fix previous Defects |
|  |  |  |
|  |  |  |
|  |  |  |

### TABLE OF CONTENTS

[1. INTRODUCTION 7](#_Toc921140)

[1.1 Purpose 7](#_Toc921141)

[2. SCOPE 7](#_Toc921142)

[2.1 Identification 7](#_Toc921143)

[2.2 Bounds 7](#_Toc921144)

[2.3 Objectives 8](#_Toc921145)

[3. OVERALL SYSTEM OVERVIEW 8](#_Toc921146)

[3.1 Context Diagram 8](#_Toc921147)

[3.2 Additional Descriptive Items 9](#_Toc921148)

[4. DOCUMENT OVERVIEW 9](#_Toc921149)

[5. REFERENCE DOCUMENTS 9](#_Toc921150)

[6. BUSINESS REQUIREMENTS 10](#_Toc921151)

[6.1 Technology 10](#_Toc921152)

[6.2 Economics 10](#_Toc921153)

[6.3 Regulatory and Legal 10](#_Toc921154)

[6.4 Market Considerations 10](#_Toc921155)

[6.6 Human Resources and Training 12](#_Toc921156)

[7. SPECIFIC REQUIREMENTS (DESCRIPTIVE FUNCTIONAL REQUIREMENTS) 12](#_Toc921157)

[7.1 Functional Descriptive Detailed Requirements 12](#_Toc921158)

[7.2 Requirement Use Cases 13](#_Toc921159)

[*7.3 Use Case Diagrams* 15](#_Toc921160)

[8. NON-FUNCTIONAL DESCRIPTIVE DETAILED REQUIREMENTSNONFUNCTIONAL REQUIREMENTS DEFINITIONS 20](#_Toc921161)

[9. ANALYSIS 21](#_Toc921162)

[9.1 Component (Component/Package/Subsystem) Architecture 21](#_Toc921163)

[9.2 Component Architecture Diagram 23](#_Toc921164) [9.3 Component Descriptions 23](#_Toc921165)

[9.4 Class Diagrams 24](#_Toc921166)

[9.4.1 Individual Class Diagrams 24](#_Toc921167)

[9.5 Events 25](#_Toc921168)

[9.6 Activity/State (Scenario) Section 26](#_Toc921169)

[9.7 State logic 31](#_Toc921170)

[9.8 Behavior 32](#_Toc921171)

[9.8.1 Sequence Diagram 32](#_Toc921172)

[10 SYSTEM TEST PLAN REQUIREMENTS 35](#_Toc921173)

[11 QUALIFICATION PROVISIONS 36](#_Toc921174)

[12 REQUIREMENTS TRACEABILITY 36](#_Toc921175)

[13 EVOLUTION OF THE RAS 37](#_Toc921176)

[14 RATIONALE 37](#_Toc921177)

[15 NOTES 37](#_Toc921178)

[16 APPENDICES 37](#_Toc921179)

[Schedule Tracking 38](#_Toc921180)

[Cumulative 39](#_Toc921181)

[Defect Tracking 40](#_Toc921182)

[Cumulative 41](#_Toc921183)

# 1. INTRODUCTION

## 1.1 Purpose

The purpose of the System Requirements and Analysis Specification document is to define the business domain which includes the business domain, the requirements, as well as the system test plan requirements and the qualification provisions.

This document is intended to be read by the software quality group, requirements group, development team, business management team, as well as the client.

# 2. SCOPE

The purpose of this project is to develop a game focused on item collection, exploration, and puzzle solving featuring a feline as the main protagonist. Many platformer games made in recent years have had a dark atmospheres or an underlying dark theme. In order to being new life to the genre, the solution is to create a game that will have a more pleasant atmosphere that is full of life and wonder. By doing this, the game will be able to attract a wider variety of consumers to the genre.

The player will play as Archie, a cat who wishes to fill his owner’s museum with rare artifacts. In order to do this, Archie must venture into the grout outdoors. The outdoors will consist of a large map with many subsections. Upon discovering a new artifact, Archie will develop either a new skill that will allow him to traverse the map in a new way or a new aesthetic item that can be used to customize the character. The player will be able to change his clothing and other aesthetic aspects such as fur pattern. There will be an achievement system as a way to give the player goals to strive for. Finally, the game will have an user interface that will be easy to navigate.

## 2.1 Identification

The Requirement and Analysis Document - Business Specification

Version 1.1

Project Team: A25

Project Name: Archie-ology

October 4, 2019

## 2.2 Bounds

Items that will be included in the project but not built by the development team includes the game music as well as the in-game art. Graphic content and visual displays, such as level backgrounds, concept art, and cosmetic overlays, will be provided by outside human resources (mixed-media artists). Likewise, audio APIs and royalty-free audio sharing sites will be utilized to either create original sounds or pull audio.

## 2.3 Objectives

Project proposal 09/20/2018

Software Business Specification (RAS) 10/04/2018

Software Requirement and Analysis Specification (RAS) 10/18/2018

Software Project Management Plan (SPMP) 11/01/2018

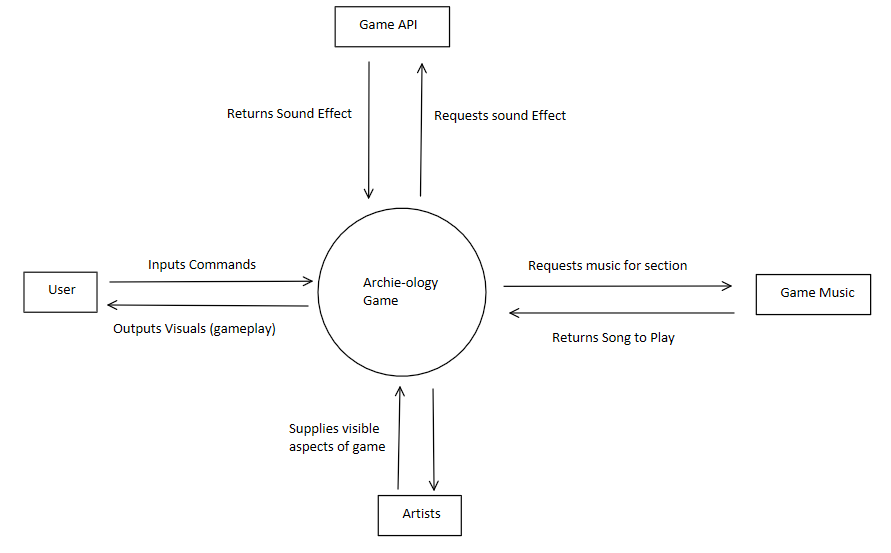
Software Analysis Specification – Final (RAS) 11/20/2018

Software Design Document (SDD) 11/29/2018

# 3. OVERALL SYSTEM OVERVIEW

The point of the system is to develop a game that anyone can find enjoyable. This is done by appealing to the player through game features. To interest new players, the game will have immersive music as well as well drawn sprites. The game also features a beautiful map that tons of mysteries to discover. To keep the player engaged, the game will feature a smooth movement system. This will allow the player to use any of Archie’s abilities at any time to give total player freedom. The game will introduce new challenges with each new ability learned. New areas will allow players to put these new abilities to the test. This is able to be done in several different orders to encourage replaying the game. The player will also be able to develop a personal connection with Archie as he is able to be customized however the player feels fit. In order to give the player positive feedback for playing the game, there is an achievement system that rewards the player to accomplishing certain goals. This gives the player more motivation as he or she players the game.

## 3.1 Context Diagram



## 3.2 Additional Descriptive Items

Players will interact the game which is controlled by the game engine. The game engine is responsible for deciding what to render onto each frame, what music to play, how to deal with collisions between different object types, and the effects, items, and achievements unlocked by the player.

The players for this game can be anyone that is able to use a controller. It is open to all ages.

The main constraints for game will be the amount of time given for this project. Full games often take many years to develop and this project is limited to one semester. Resources to create the assets for this project will also be limited.

The assumptions and dependencies for this project will be listed during another iteration of this document because this document will focus on the business domain.

# 4. DOCUMENT OVERVIEW

The rest of this document analyzes the plan for the system, specifically its requirements. Section 6 contains the business requirements: technology; economics; regulatory and legal; market considerations; risks and alternatives; and human resources and training. System test plan requirements are in Section 10. Qualification provisions are in Section 11. Requirements traceability is in Section 12. Evolution of the RAS is in Section 13.

The specific requirements, non-functional descriptive detailed requirements definitions, and analysis will not be covered in this version of the document but will be specified at a later time.

# 5. REFERENCE DOCUMENTS

Project Proposal

Project Name: Archie-ology

Project Team: A25

Version 1.0

September 20, 2018

RAS

Project Name: Archie-ology

Project Team: A25

Version 1.0

October 4, 2018

# 6. BUSINESS REQUIREMENTS

## 6.1 Technology

  The technology drivers for this product include online game distribution sites and game making tool sets to control the client side rendering and GUI of the game. Technology supports our business objectives by providing our team platforms to not only flesh out our vision for the game, but also garner the attention of both game makers and game players alike. The game engine, Unity, will play a large part since it will be the foundation that will build and run our game; likewise, Unity is user-friendly, to creators and players alike, therefore will help us bring our game to reach a broad amount of people. Additionally, social technology, such as Twitter, Instagram, Facebook, and YouTube, will be the main platforms to market our game since these platforms are used regularly.

## 6.2 Economics

If this project is as successful as we hope it will be, the game will be published online as an indie game. Our team hopes to add to the attention given to indie game development, specifically small team game companies. The hope is to open this genre to a larger consumer market and therefore increase financial support. Indie game developers are abundant; however, a few are able to make their name and game(s) well-known within the gaming community; indie games are often overshadowed by “name brand” games (i.e. Nintendo’s Super Mario franchise). One can argue that these games garner a larger consumer market and finances due the amount of work and effort needed to produce a larger and higher-quality game. However, these days, thanks to various game making tools, games creating is much less exclusive. The amount of small team creators producing deep and complex games is increasing and every successful game that is part of this genre adds to the reputation that their financial success depends on.

If our game is well received the reputation it provides us as designers will add to the potential success of any other games we may produce after. Having a reputation as a designer helps increase the hype and publicity of future projects.

## 6.3 Regulatory and Legal

Copyright strikes over sound/music usages will be the primary legal concern. Our team will aim to utilize royalty-free audio along with audio created by our team and additional specialized professionals willing to work with us. There will be a minor concern of a plagiarism accusation due to the possibility of our game being seen too similar to another game on the market though we don’t know any game that could make a claim like this at this time.

## 6.4 Market Considerations

Our game will be marketed on digital distribution platforms specific to video games. Platforms include: Steam, IndieGala, Humble Bundle, and G2G Gaming Marketplace. All aforementioned platforms offer a variety of games, ranging from high-production best-sellers to independent projects. Our game falls into the “independent projects” category – an “indie” game – since our system is being developed by a small team and with limited financial support. Since our game does not have a high production cost and since it lacks a reputation of quality due to this being the first game we as a team have produced, the price of the system will estimate five dollars.

In order to promote our system, we plan to reach out to YouTubers in order to ask if they are willing to create a video of them playing, and/or reviewing, our game. We will provide them with a free copy of the game and if possible offer some other promotional item as compensation. In order to qualify for this, the YouTuber(s) must have: at least 1k subscribers, consistent uploads (daily, every-other-day, weekly) and views, and at least 1-year experience as a content creator on YouTube. Only YouTubers whose current content is relevant to the intended audience of this project will be considered. This mainly includes gaming channels who make videos on a variety of games. Likewise, the YouTuber(s) are expected to provide an unbiased review and commentary without crude or vulgar language.

**6.5 Risks and Alternatives**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Business Risk | Operational Risk | Technology Risk | Economic Risk |
| Description: | Game does not launch by deadline | Personal or environmental factors affect a team member’s ability to work | Game does not run on certain systems (PC vs. Mac) | Legality fees |
| Probability: | 0.5% | 25% | 35% | 5% |
| How discovered: | Near launch and we realize that substantial work still needs to be done | Reported by the team member affected or noticed by other members | During later stages of play testing | Either approached by outside or discovered material ourselves |
| Responsible Party: | Sabrina Supapkooha & Jeffrey Qiu | Jeffrey Qiu &  Jungho Kook | Jungho Kook &  Juliet Ramdass | Juliet Ramdas & Sabrina Supapkooha |
| Status: | Not yet encountered | Not yet encountered | Not yet encountered | Not yet encountered |
| Mitigation Plan: | Set goals and flags for tasks; SCRUM | Having multiple team members involved in each role | Continuous play testing; baseline safety | Copyright our work and respect the copyright of others |

## 6.6 Human Resources and Training

Primarily human resources is our team of four: Sabrina Supapkooha, Juliet Ramdass, Jeffrey Qiu, and Jungho Kook. Additional human resources are associates who will provide talents that our team either does not possess or are not well-experienced in (i.e. artists and audio content producers). These additional human resources will be given ample time to accomplish tasks (i.e. character and world sketches, sound/music production) and assist the team (i.e. recommending open sources).

As for training, our team will be trained and advised by more experienced game developers who have already designed, created, produced, and distributed games. Training will primarily be mini-lessons in Unity and other game engines so that our team can smoothly efficiently implement the game. Likewise, some training in audio and digital character creation will be necessary. Feedback about the experience of playing the game will also serve as a way to learn about the ideal difficulty for a game of this type.

# 7. SPECIFIC REQUIREMENTS (DESCRIPTIVE FUNCTIONAL REQUIREMENTS)

This section is not covered in this version of the document.

## 7.1 Functional Descriptive Detailed Requirements

1. The game must provide means of representing and accessing the player character’s profile.
   1. The player should be provided with facilities to customize the appearance of the player character.
   2. Each costume will have an associated colored icon to indicate what color will be applied to the player character’s appearance.
   3. A preview of what the character player will look like should be on-screen as the player switches between costumes.
2. The game should have a gallery to showcase the collectable artifacts.
   1. Artifacts will be displayed in order of acquirement.
   2. Each artifact will be represented by a unique icon.
   3. Undiscovered artifacts will be represented by an undetailed, blacked-out silhouette of the icon; upon discovery, the icons of artifacts will be detailed and colored.
   4. If selected by the player, a short description of the selected artifact will appear.
3. The game must promote continuous play by means of an achievement system.
   1. The achievement system should be accessible through the main menu.
   2. Similar to artifacts, locked achievements will remain blacked-out while unlocked achievements will be colored.
   3. Locked achievements will have a short description of the requirements to unlock.
   4. Unlocked achievements will have congratulatory message.
4. The game must open to a menu main that has the options to: load game, start a new game, settings, achievements, gallery of artifacts, and quit game.
   1. Each option must have its own interface and features when activated by the player.
5. The game must provide the player the option to save and/or load a game file.
   1. Saving a game stores the current progress that player has made in an accessible file.
   2. Loading accesses the stored game data and allows the player to continue play without starting over.
   3. Both allows the player to close the game and return to the game at a later time without losing progress.
6. The game must have a pause screen to allow players access other features during play.
   1. The player should be able to click a button associated with the pause screen to display it.
   2. Upon display, the pause menu will appear; menu includes access other settings such as main menu, save game, load game, dressing room, and settings.

## Requirement Use Cases

1. Player must be able to enter the menu from the title page.

* 1. Player hits any button.
  2. System displays the main menu.

2. Player must be able to start a new game.

* 1. User hits new game.
  2. System accesses the game at the starting point.

3. Player must be able to load game

* 1. User hits load game.
  2. System accesses game from save file.

4. User must be able to view and change his settings.

* 1. User hits settings option
  2. System displays settings menu
  3. User has access to changeable game features (such as sound volume, resolution)

5. User will be able to view achievements.

* 1. User hits achievement.
  2. System displays achievements.
  3. User browses achievements.

6. User will be able to view gallery.

* 1. User hits gallery.
  2. System displays gallery.
  3. User browses artifacts.

7. User will be able to quit the game.

* 1. User hits quit game.
  2. System closes.

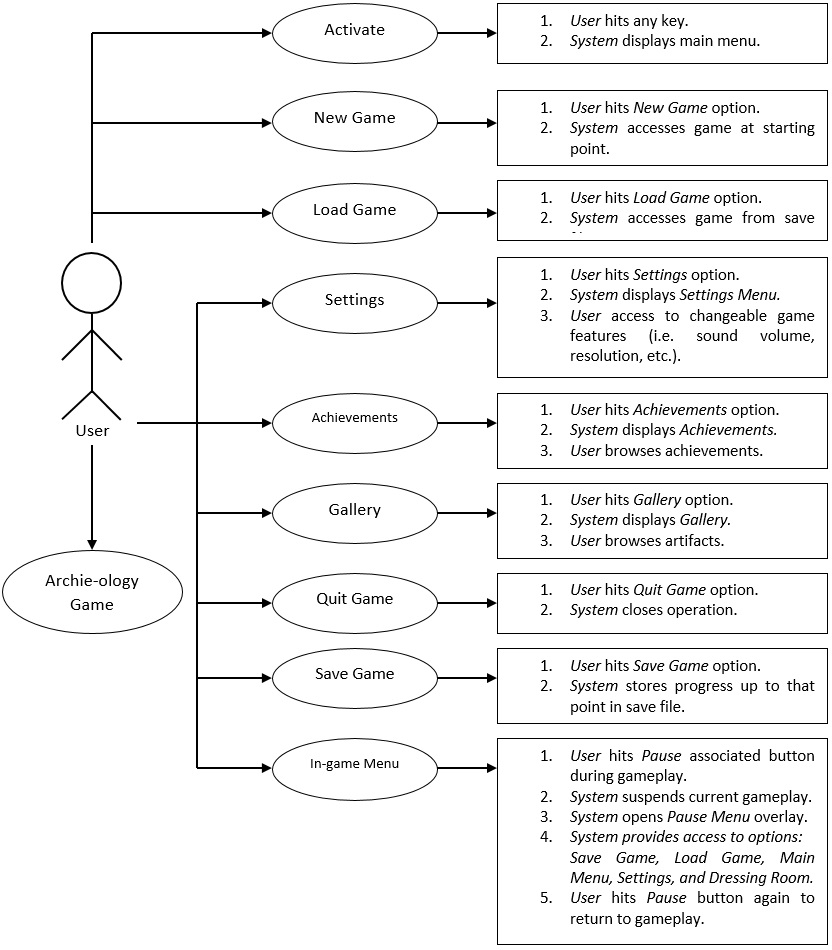
8. User will be able to save game.

* 1. User hits save game.
  2. System stores user progress.

9. User should be able the in-game menu.

* 1. User hits pause button during gameplay.
  2. System suspends current gameplay.
  3. System opens the pause menu overlay.
  4. System provides options to save game, load game, main menu, settings and dressing room.
  5. User hits pause button again to resume gameplay.

##### *7.3 Use Case Diagrams*



|  |  |  |
| --- | --- | --- |
|  |  | Activate |
| **Description** |  | Displays the main meu |
| **Pre- Conditions** |  | Game is on Title Page |
| **Flows** | **Basic or Normal Flows** | 1. User hits any key  2. System displays main menu |
|  | **Alternative Flows** | 1. None |
| **Post**  **Conditions** |  | Enter Main Menu |
| **Special**  **Requirements** |  |  |
| **Extension Points** |  |  |

|  |  |  |
| --- | --- | --- |
|  |  | New Game |
| **Description** |  | Starts a new game |
| **Pre- Conditions** |  | Game is on main menu |
| **Flows** | **Basic or Normal Flows** | 1. User hits New Game  2. System access game from starting point |
|  | **Alternative Flows** | 1. None |
| **Post**  **Conditions** |  | Game Running |
| **Special**  **Requirements** |  | None |
| **Extension Points** |  | None |

|  |  |  |
| --- | --- | --- |
|  |  | Load Game |
| **Description** |  | Resumes game from a saved state |
| **Pre- Conditions** |  | Game is on main menu |
| **Flows** | **Basic or Normal Flows** | 1. User hits Load Game  2. System access game from saved point |
|  | **Alternative Flows** | 1. None |
| **Post**  **Conditions** |  | Game Running |
| **Special**  **Requirements** |  |  |
| **Extension Points** |  |  |

|  |  |  |
| --- | --- | --- |
|  |  | Settings |
| **Description** |  | Access changeable game features |
| **Pre- Conditions** |  | Game is on main menu |
| **Flows** | **Basic or Normal Flows** | 1. User hits Settings  2. System displays Settings Menu |
|  | **Alternative Flows** | 1. None |
| **Post**  **Conditions** |  | Settings Menu |
| **Special**  **Requirements** |  |  |
| **Extension Points** |  | User has ability to change some game features |

|  |  |  |
| --- | --- | --- |
|  |  | Achievements |
| **Description** |  | Displays all achievements for user to see |
| **Pre- Conditions** |  | Game is on main menu |
| **Flows** | **Basic or Normal Flows** | 1. User hits Achievements  2. System displays Achievements  3. User browses Achievements |
|  | **Alternative Flows** | 1. Exit out of Achievements |
| **Post**  **Conditions** |  | Achievement Gallery |
| **Special**  **Requirements** |  |  |
| **Extension Points** |  |  |
|  |  | Gallery |
| **Description** |  | Opens a gallery of all the game images |
| **Pre- Conditions** |  | Game is on main menu |
| **Flows** | **Basic or Normal Flows** | 1. User hits Gallery  2. System displays Gallery  3. User browses Gallery |
|  | **Alternative Flows** | 1. None |
| **Post**  **Conditions** |  | Gallery |
| **Special**  **Requirements** |  |  |
| **Extension Points** |  |  |

|  |  |  |
| --- | --- | --- |
|  |  | Quit Game |
| **Description** |  | Closes out of game |
| **Pre- Conditions** |  | Game is running |
| **Flows** | **Basic or Normal Flows** | 1. User hits Quit Game  2. System closes all operation |
|  | **Alternative Flows** | 1. None |
| **Post**  **Conditions** |  | Achievement Gallery |
| **Special**  **Requirements** |  |  |
| **Extension Points** |  |  |

|  |  |  |
| --- | --- | --- |
|  |  | In-game Menu |
| **Description** |  | Opens an in-game menu |
| **Pre- Conditions** |  | Game Running |
| **Flows** | **Basic or Normal Flows** | 1. User hits Pause menu  2. System suspends current gameplay  3. System opens Pause Menu Overlay  4. System provides access to options: Save Game, Load Game, Main Menu, Settings, and Dressing Room  5. User can edit settings |
|  | **Alternative Flows** | 1. None |
| **Post**  **Conditions** |  | In-Game Menu |
| **Special**  **Requirements** |  |  |
| **Extension Points** |  |  |

|  |  |  |
| --- | --- | --- |
|  |  | Move Player |
| **Description** |  | Moves Archie |
| **Pre- Conditions** |  | Game is running |
| **Flows** | **Basic or Normal Flows** | 1. User hits an input direction  2. System determines outcome of input  3. Display outcome |
|  | **Alternative Flows** | 1. None |
| **Post**  **Conditions** |  | Game is Running |
| **Special**  **Requirements** |  |  |
| **Extension Points** |  |  |

|  |  |  |
| --- | --- | --- |
|  |  | Ability |
| **Description** |  | Uses an ability |
| **Pre- Conditions** |  | Game is running |
| **Flows** | **Basic or Normal Flows** | 1. User hits ability button  2. System determines outcome of ability  2. System displays outcome |
|  | **Alternative Flows** | 1. None |
| **Post**  **Conditions** |  | Game is Running |
| **Special**  **Requirements** |  |  |
| **Extension Points** |  |  |

# 8. NON-FUNCTIONAL DESCRIPTIVE DETAILED REQUIREMENTSNONFUNCTIONAL REQUIREMENTS DEFINITIONS

1. Computer Hardware Requirements
   1. OS: Windows 7 or higher (32-bit or 64-bit)
   2. Processor: Intel Core i5-650 3.2GHz
   3. Memory: 8 GB RAM
   4. Storage: 27 GB available space
   5. No network connection required (aside from downloading the application)
2. System Performance characteristics
   1. Game should load the title screen within three seconds.
   2. Gameplay should begin within three seconds of hitting new or load game.
   3. Any transition between menu pages should occur within one second
   4. Character movement using controller should feel instantaneous
3. System Human Interfaces
   1. System should be able to recognize when a Xbox controller is connected.
   2. System gives the user an option to choose controller or keyboard.
4. System Maintainability
   1. System should update periodically to fix bugs, prepare for new features, and add content.
   2. When an update is ready, system should prompt user to install update(s).
5. System Quality Factors
   1. System should not lag and graphics should clip into each other.
   2. System should power off unexpectedly or without user authorization.
   3. Each dangerous environmental hazard should be instantly recognizable
   4. Each artifact and checkpoint should be instantly recognizable.
6. Personnel-Related Requirements
   1. Development team should have experience in game design and narrative and Unity programming.
   2. Artist team should have portfolio showcasing their artistic abilities, strengths, and limits.
7. Training-Related Requirements
   1. Training for all aspects in areas that the development team does not have apt knowledge in. Training will be done by researching the target topic and applying information to project. Training will be done as a group, with team members supporting and helping each other.

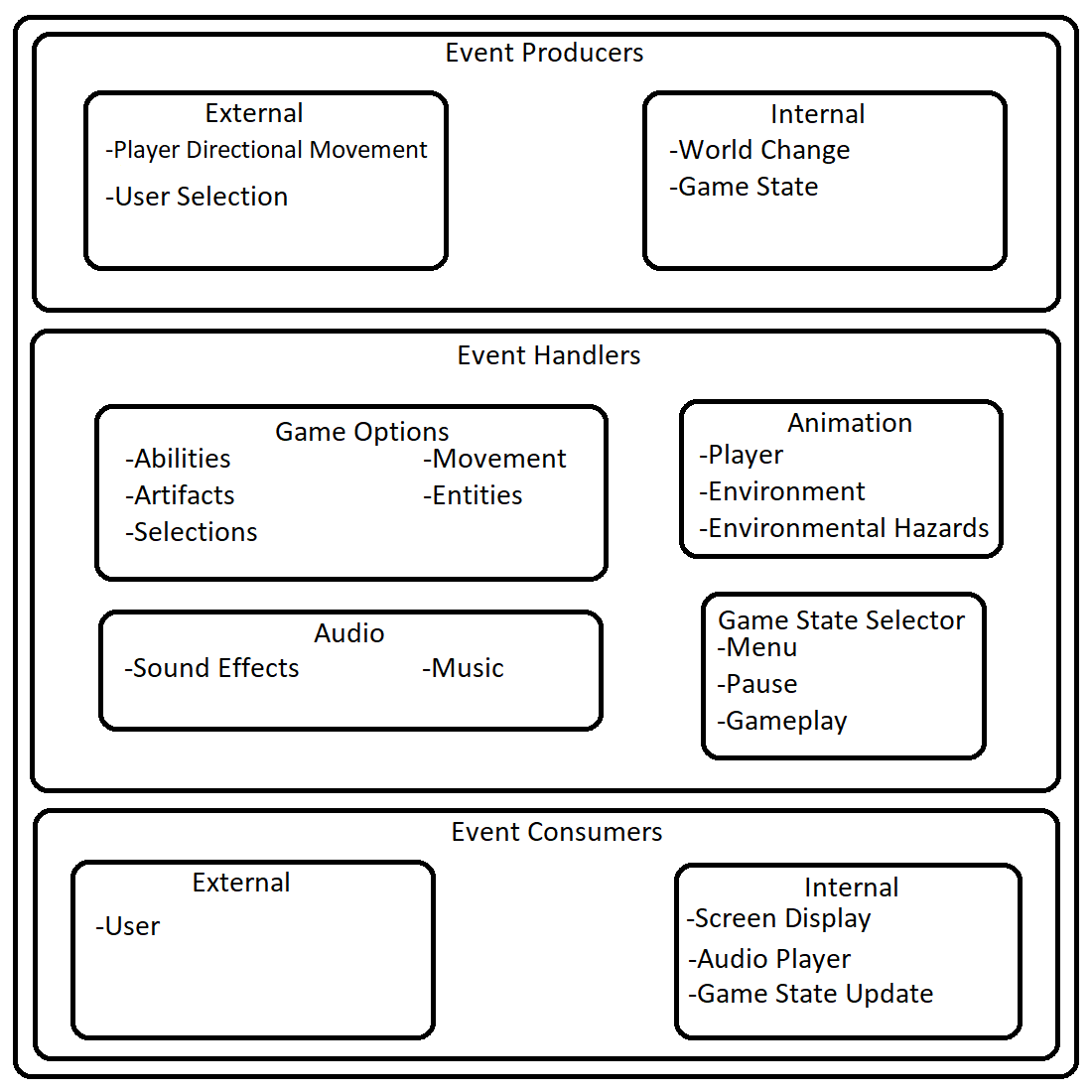
# 9. ANALYSIS

## 9.1 Component (Component/Package/Subsystem) Architecture

For the game, we will be utilizing an Event Systems Architecture since the system is based off a series of events and scenarios. When the system is in play, such that the user is engaging with the play environment, events will be trigger when the user encounters a trigger event. The event handler will determine the current state of the game and all that it involves such as the display, audio, and next potential events. As an example, once the user has acquired a certain artifact for one section of the game, a new section of the game will be unlocked. This is handled in the architecture as the current GameState encountering a WorldChange event and updating the possible actions by the player that are new events the game will handle. A more common example event that the system will handle are movement events that will be handled by updating the GameState with the new location of the character and calculating collisions in this new space. Essentially, by having an event-based system, when certain events occur, the system will update the game state to deal with the changes associated with them.

Components internal to the system itself are: movement, sound effects, music, ability, artifacts, menu, pause, and environmental hazard. Components external to the system are: the user, user input, keyboard and/or controller. Components will be packaged by relations. For example, movement, user, user input, and keyboard (or controller) will be packaged under “Player Directional Movement”. Likewise, each component can be a class with subclasses/methods. For example, Class Movement will have methods such as: up, down, left, right, and jump. Likewise, class Sound Effects will have subclasses such as: achievement sounds, failure sounds, and favor-text sounds; each of these subclasses will have methods that will play sound effects in events specific to the subclass.

Components fit into the architecture as either event producers, event handlers, or event consumers. Event producers and event consumers can be broken up into external and internal components of these types. Event handlers can be divided into subsystems based on topic. Not all event handlers are necessary or have an effect for each event. The following diagram displays an event-based system architecture that shows the division of subsystems and components.



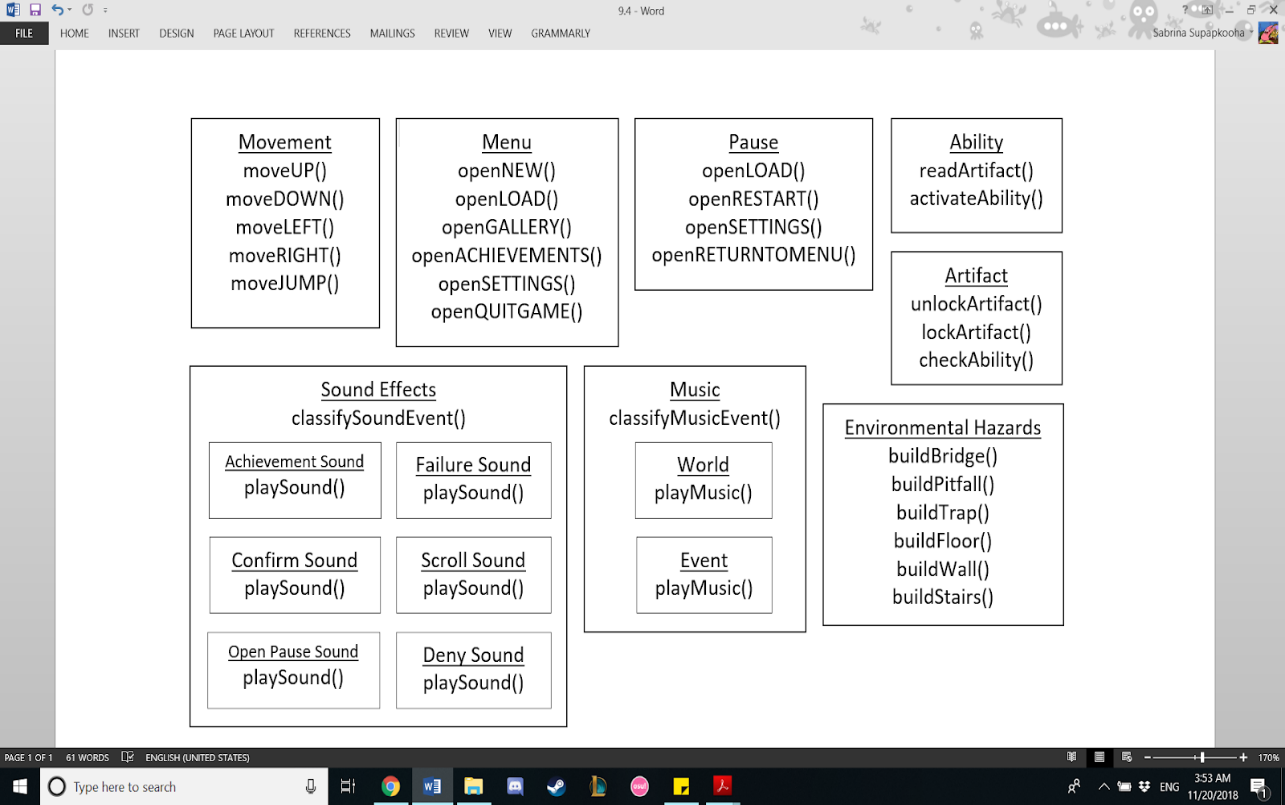
## 9.2 Component Architecture Diagram

## 9.3 Component Descriptions

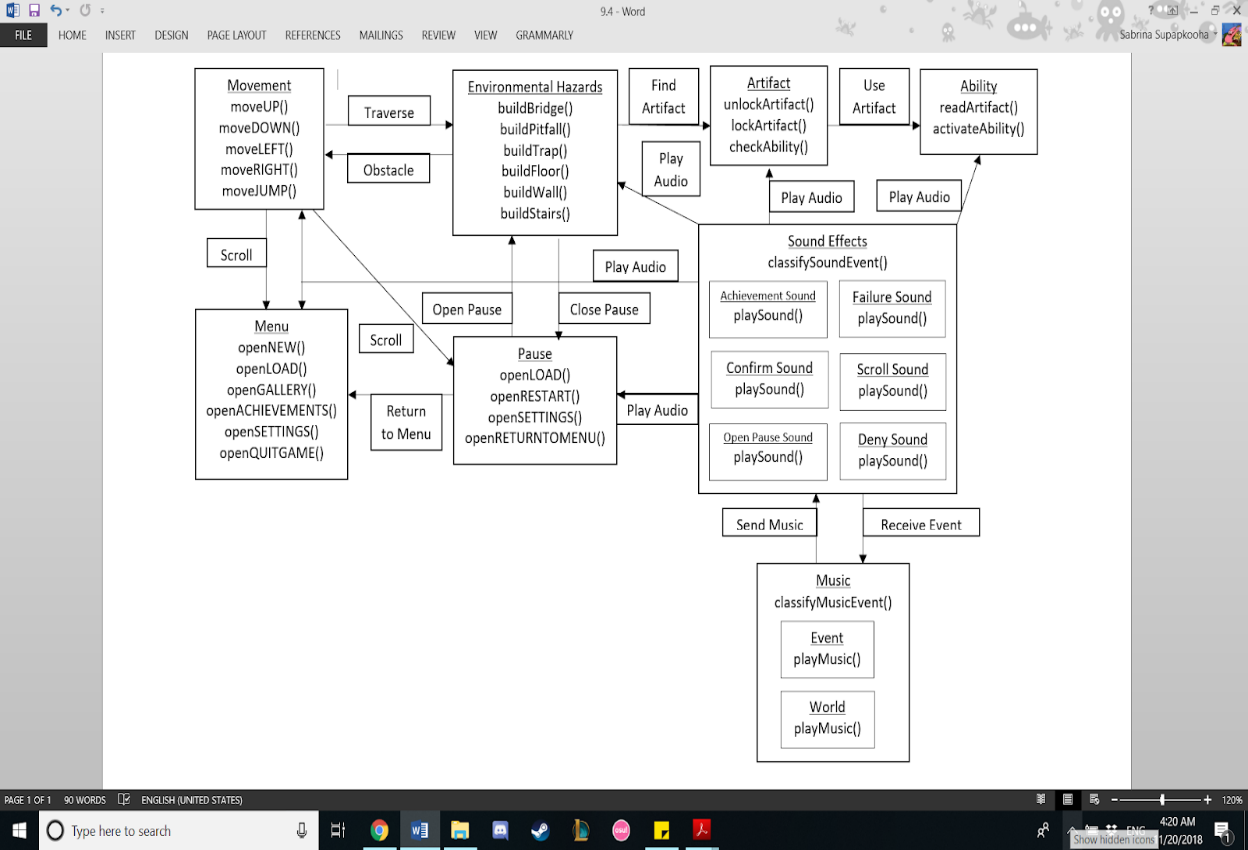
1. Movement
   1. User’s player character will move up, down, left, right, or jump according to user input using either the keyboard/mouse or controller; movement can also be read to scroll through on-screen choices. Input will be read as directional signals. For example, utilizing the WASD keys on the keyboard, W will be upward movement, S will be downward movement, A will be leftward movement, and D will be rightward movement; the SPACE key will be jump.
2. Sound Effects:
   1. Unique sounds will be triggered depending on the event. For example, a congratulatory sound will play when the user acquires an artifact. Likewise, when the player opens the pause screen, a sound will play signify that gameplay is momentarily suspended.
3. Music:
   1. Similar to sound, music will play in the background while the system is in play. Certain music will emphasize certain events/worlds. For example, during an exciting scene, upbeat and cheery music will play. Likewise, if the world is full of lush plants and bugs, the music will reflect the jungle-like atmosphere.
4. Ability
   1. Each ability will correspond to an artifact. The user must have acquired the artifact first before they can activate the ability. Each ability is unique to the artifact it corresponds to.
   2. Abilities can range from additional health to movement boosts.
5. Artifacts
   1. Every newly acquired artifact will unlock an ability that will not only help the user progress the game in terms of gameplay, but also in term of story.
   2. Artifacts, both locked and unlocked, can be view from the Gallery accessible through the menu screen.
6. Menu
   1. The menu will on the first interactive screen upon system start-up. After a short animation of the system start-up, the menu will display options such as: NEW, LOAD, GALLERY, ACHIEVEMENTS, SETTINGS, and QUIT GAME.
7. Pause
   1. Essentially has the same properties of the menu, but will exclude the NEW, GALLERY, ACHIEVEMENTS, or QUIT GAME options. In addition to included options, pause screen will have RESTART and RETURN TO MAIN MENU. During pause, gameplay is momentarily suspended.
8. Environmental Hazards
   1. Regards the levels and obstacles that the user will need to guide the player character to overcome.

## 9.4 Class Diagrams

##### 9.4.1 Individual Class Diagrams



**9.4.2 Class Relationship/Interaction Diagrams**



## 9.5 Events

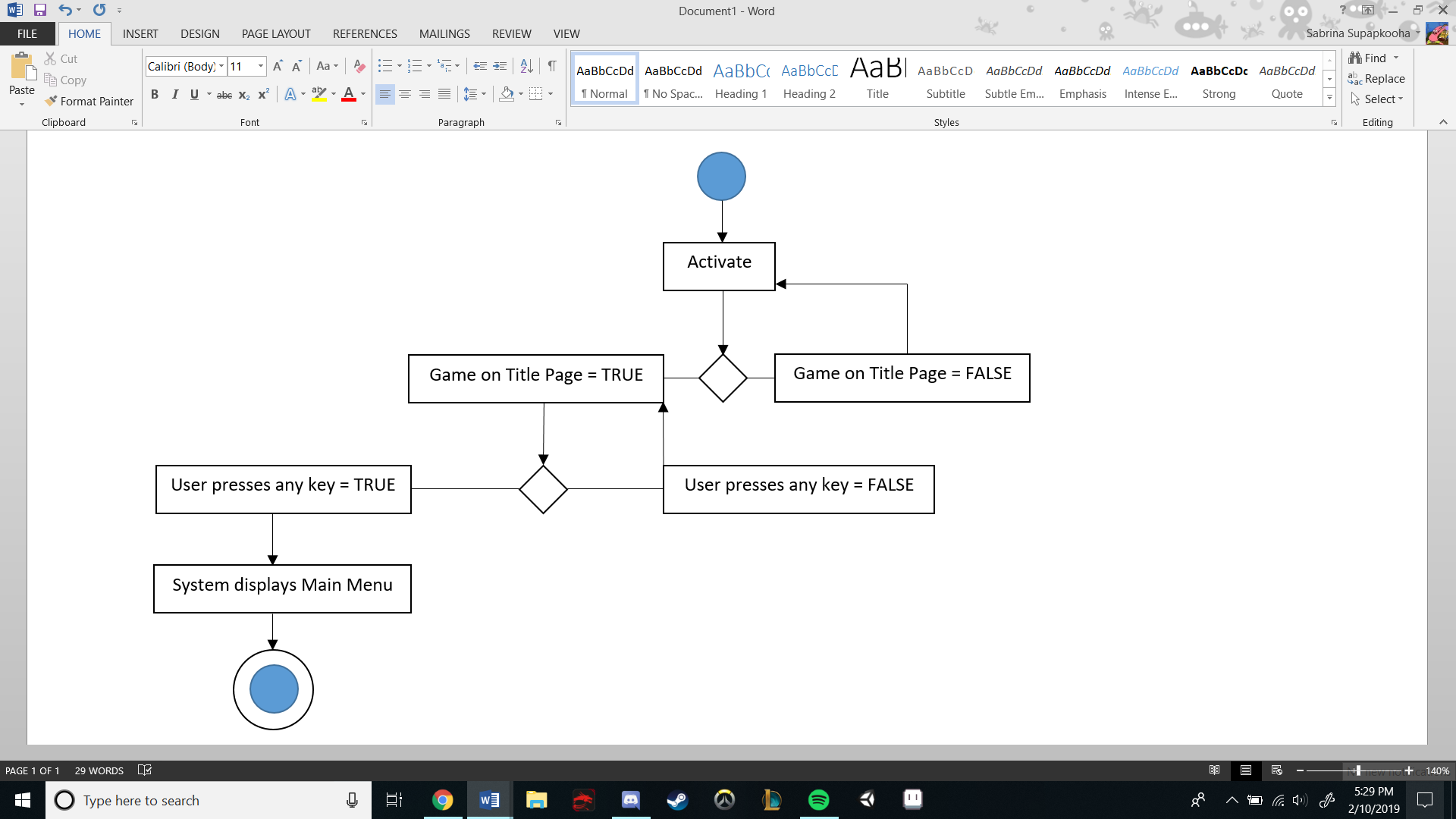
In order to keep the game as responsive as possible, there are many events that need to be tracked. These events include, when each different menu is opened (main menu, in-game menu, etc), when the user provides input, when an ability is used, when an artifact is collected, when a new world is entered, when a specific audio effect needs to be played, and when a soundtrack needs to be played.

**9.5.1 Motives**

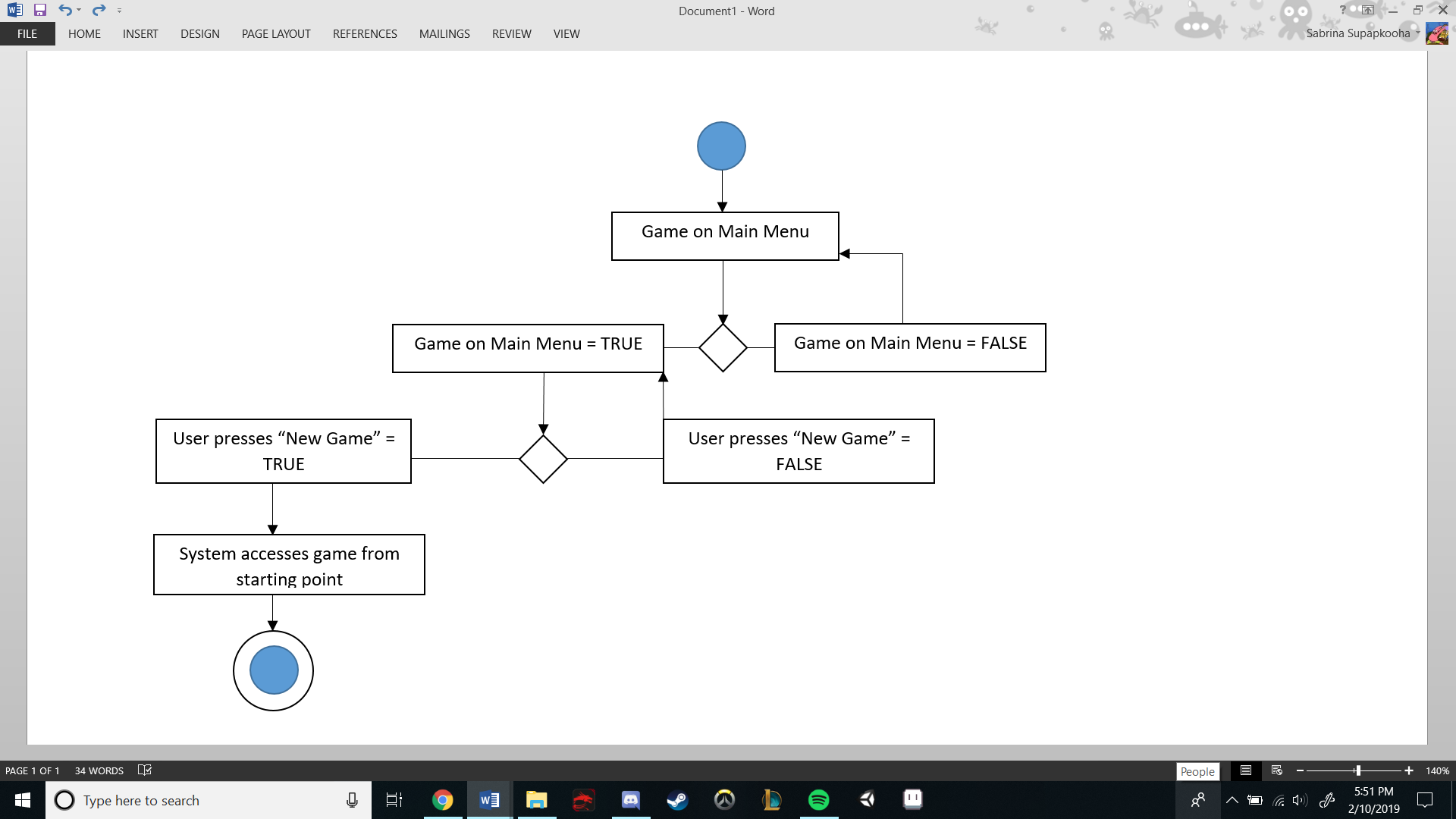
|  |  |
| --- | --- |
| Menu Opening | Opens the menu so the player is able to access more options for the game. |
| User Input | When the player inputs something we read the input and convert it to something in the program |
| Ability Usage | When the player activates an ability, the character needs to follow the instruction |
| Artifact Collected | Every time an artifact is collected, it is added to the gallery and a new ability is gained |
| New World | Every time the player wants to enter a new world; the map needs to reload |
| Audio Sound Effect | Certain sound effects are played when an event occurs |
| Music Soundtrack | Certain areas has their own music that needs to be played |

## 9.6 Activity/State (Scenario) Section

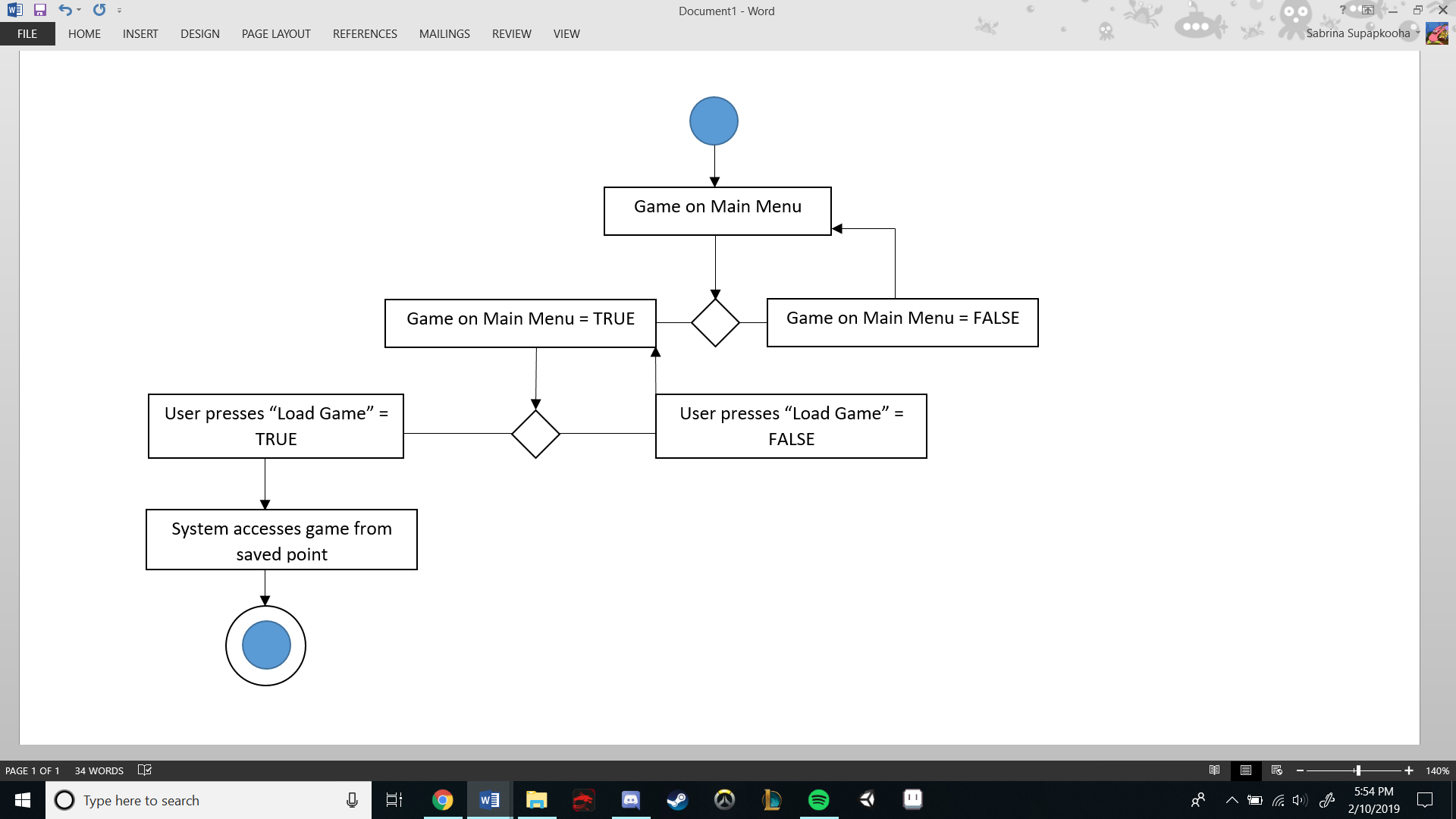
Scenario for Assessing Main Menu



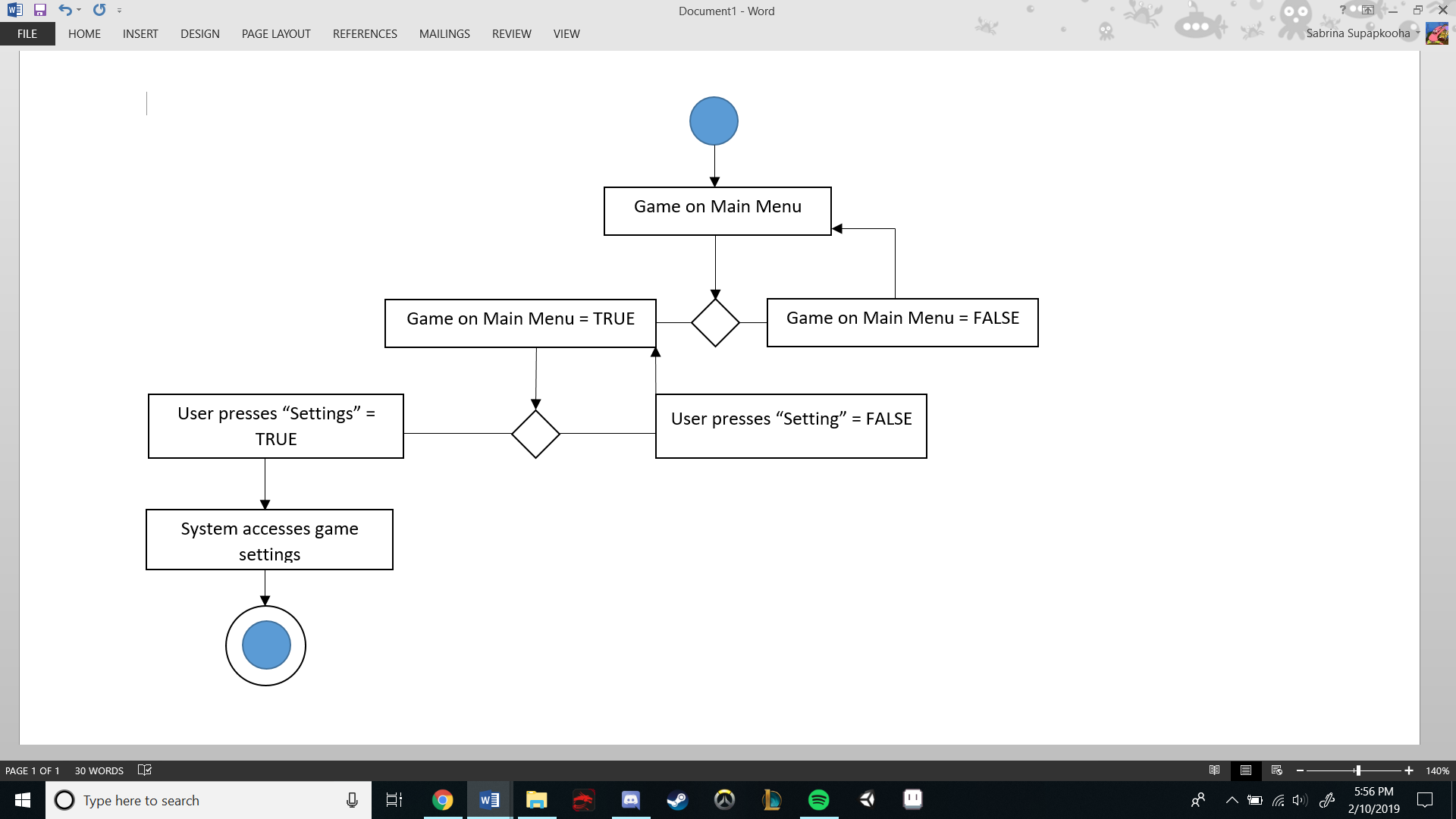
Scenario for Starting a New Game



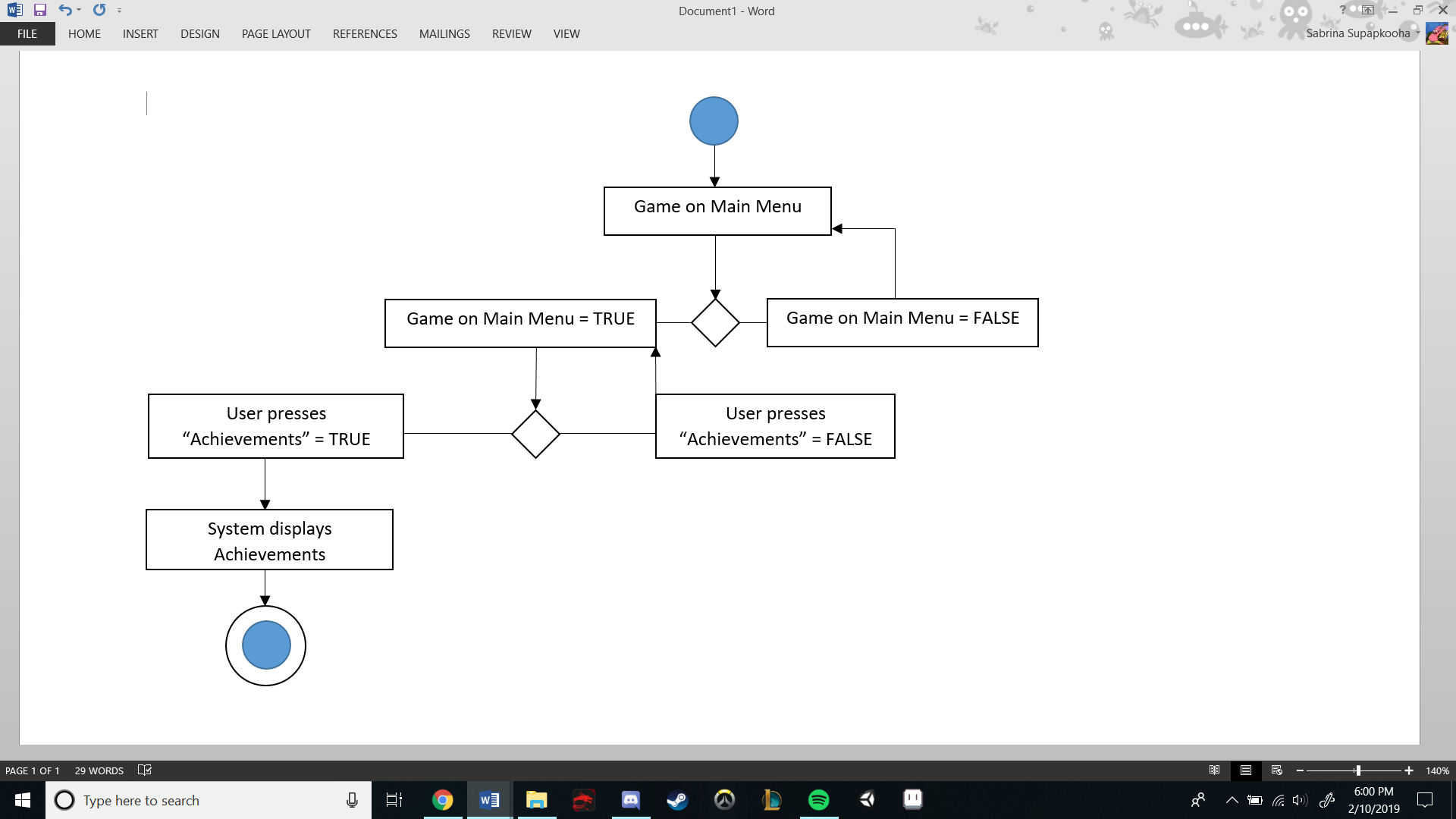
Scenario for Loading a Game



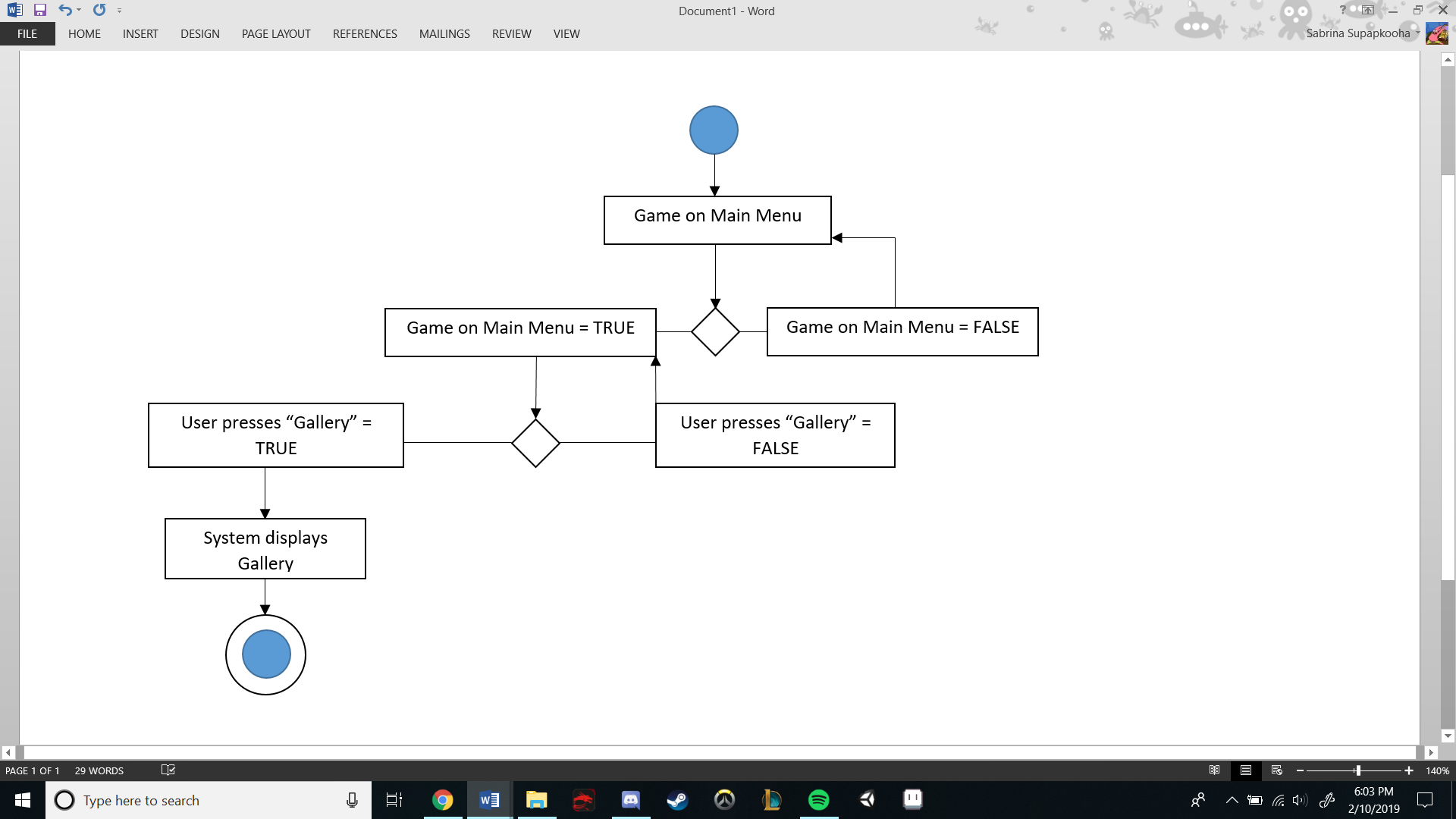
Scenario for Accessing Game Settings



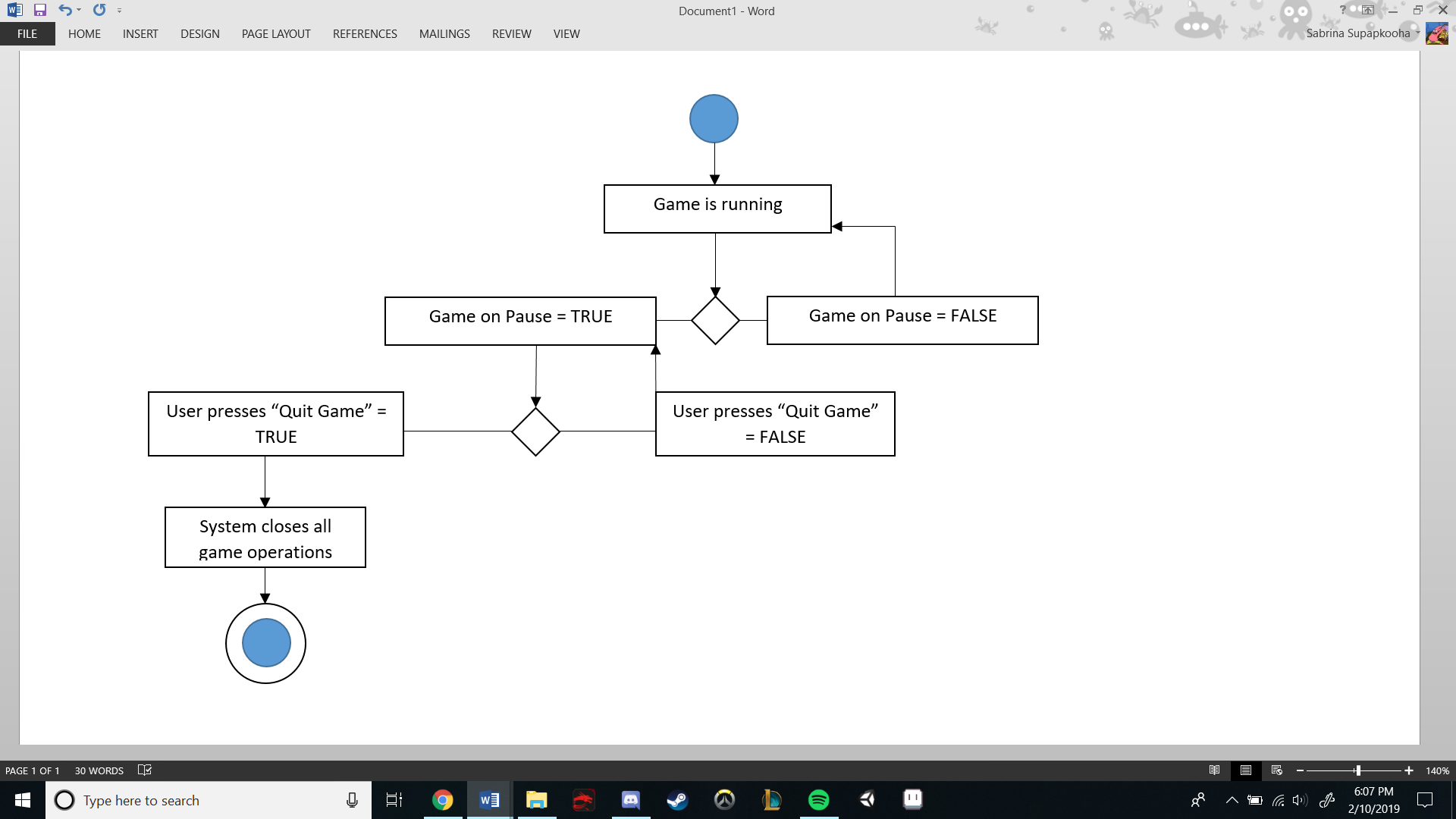
Scenario for Accessing Achievements



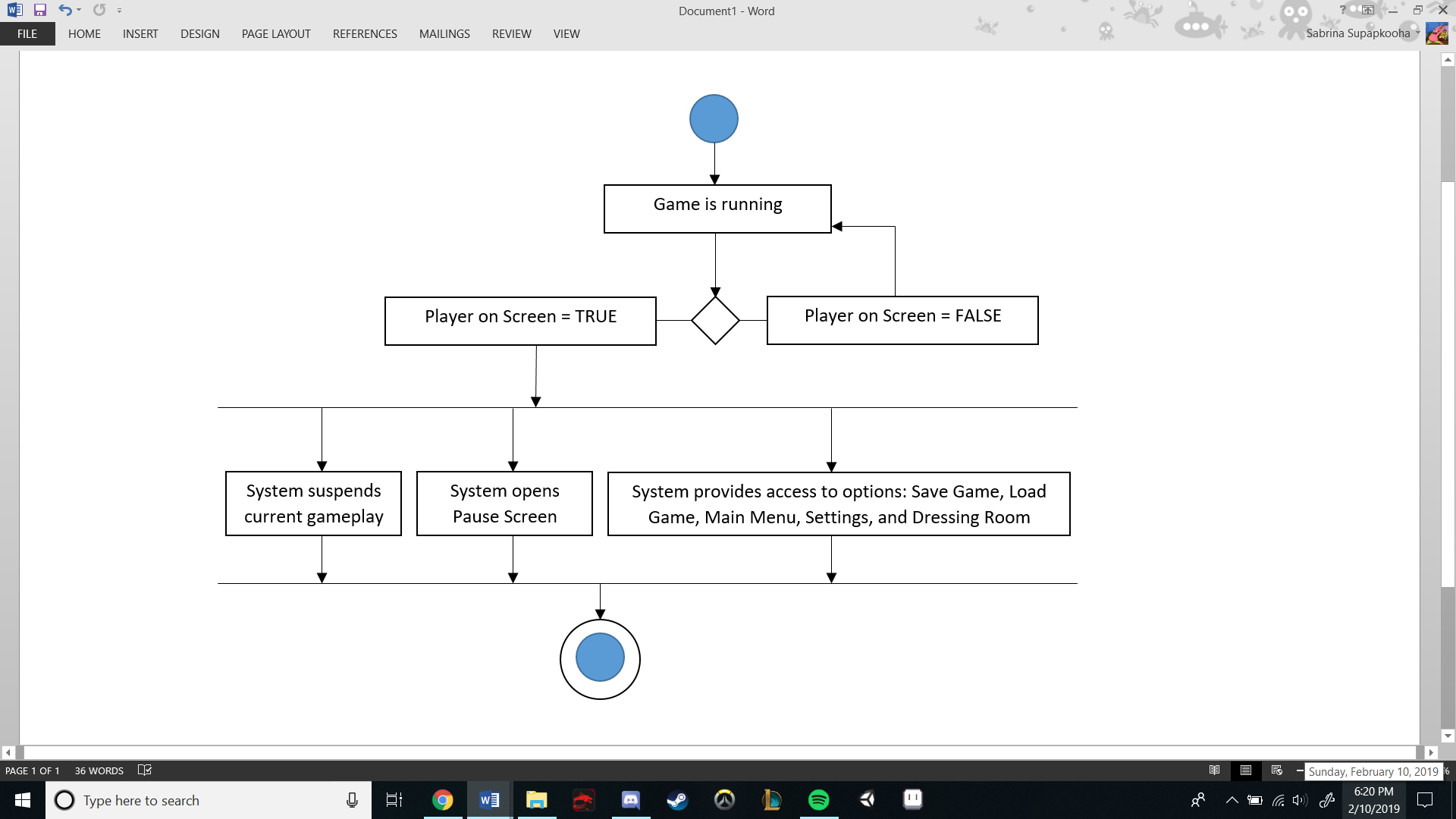
Scenario for Accessing Gallery



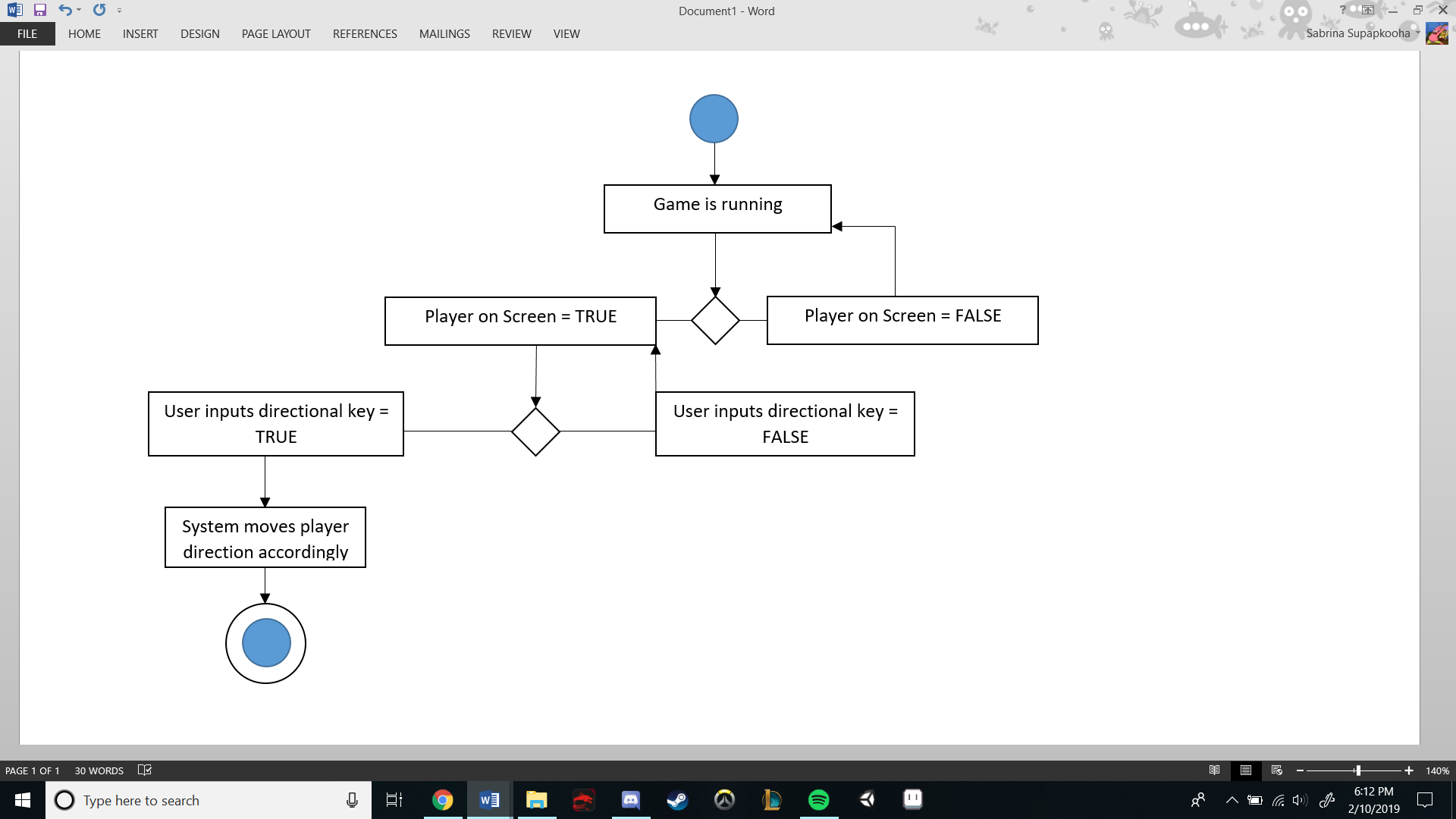
Scenario for Quitting Game



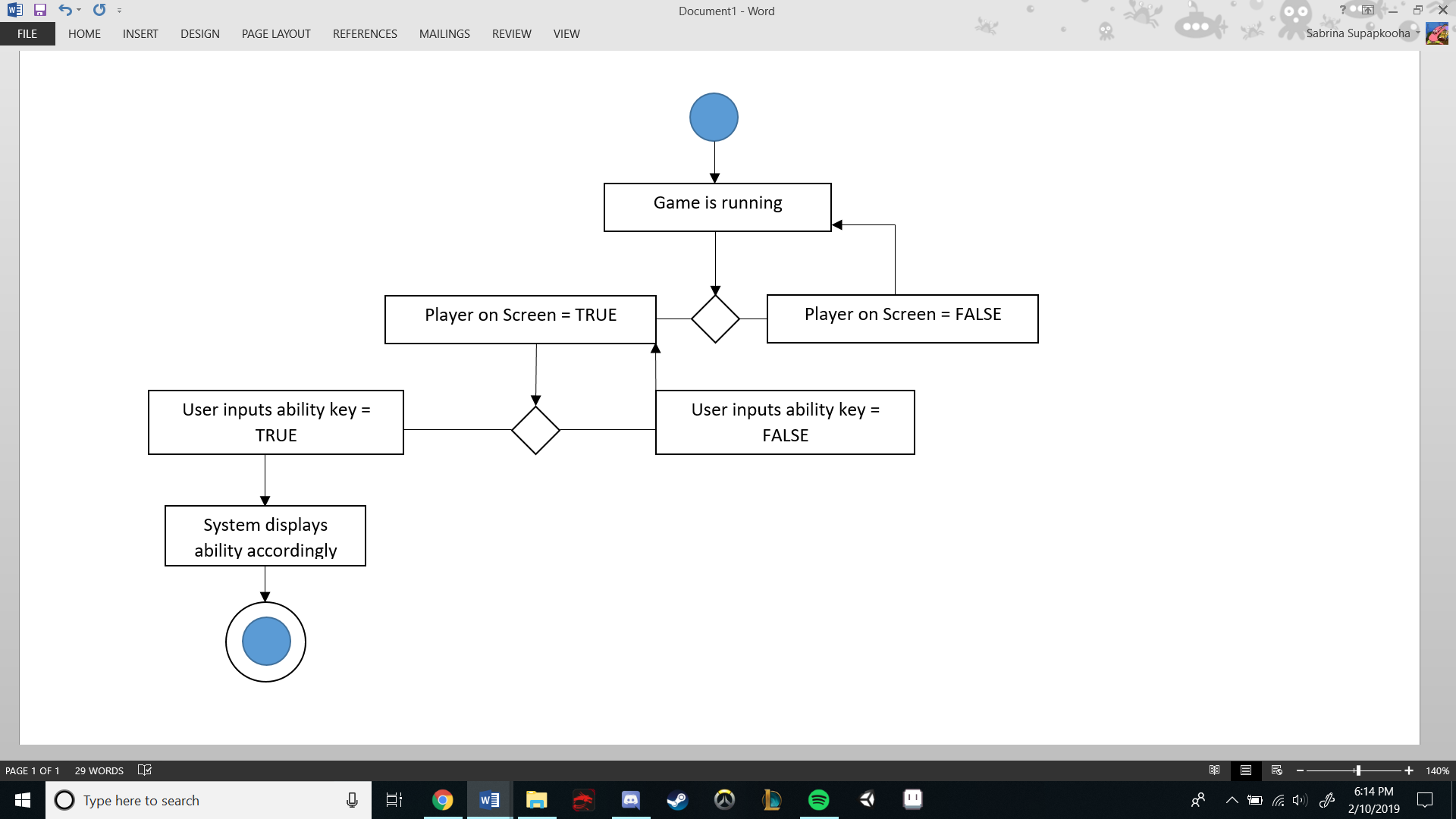
Scenario for Accessing In-Game Menu

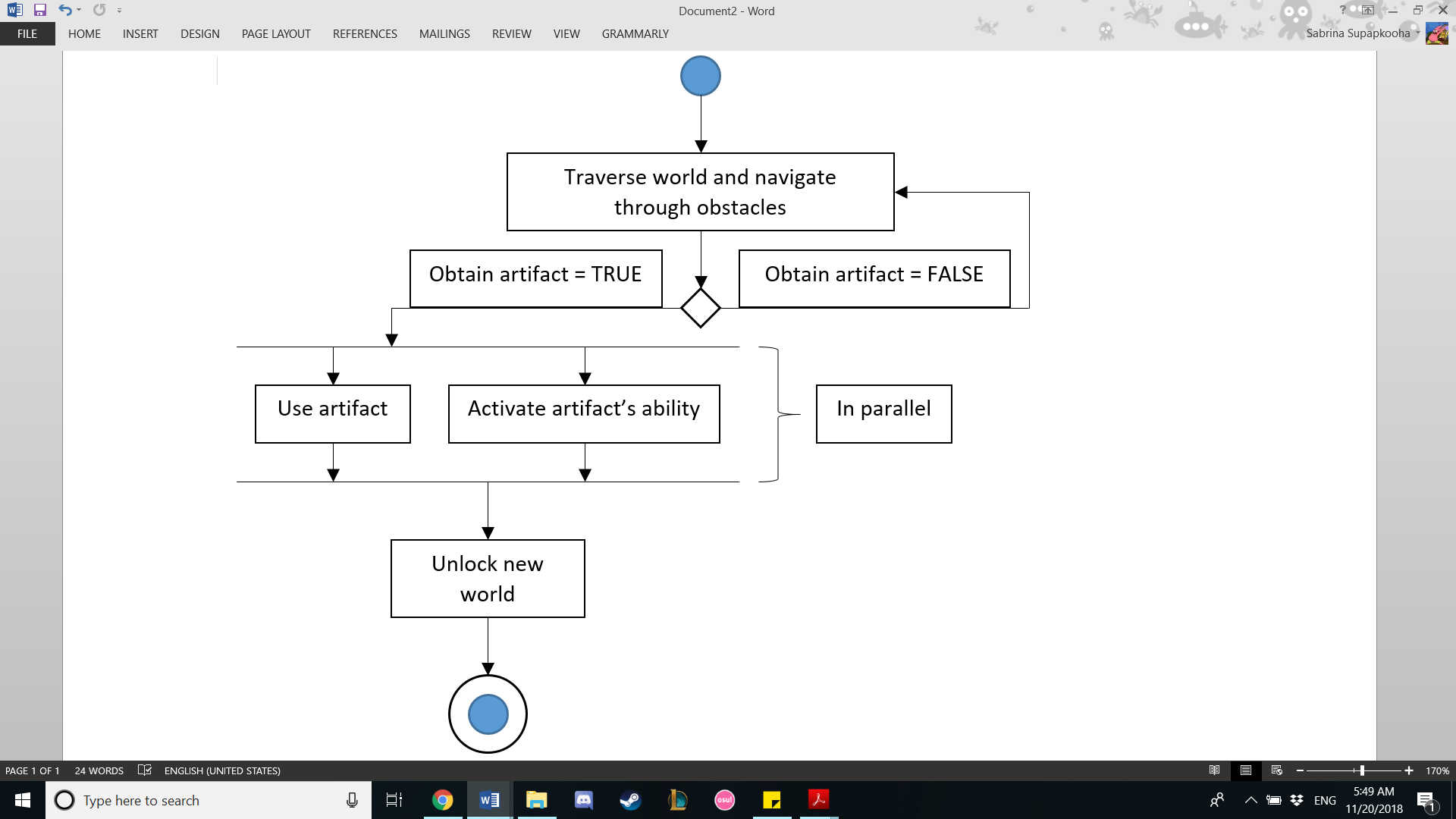


Scenario for Moving Player



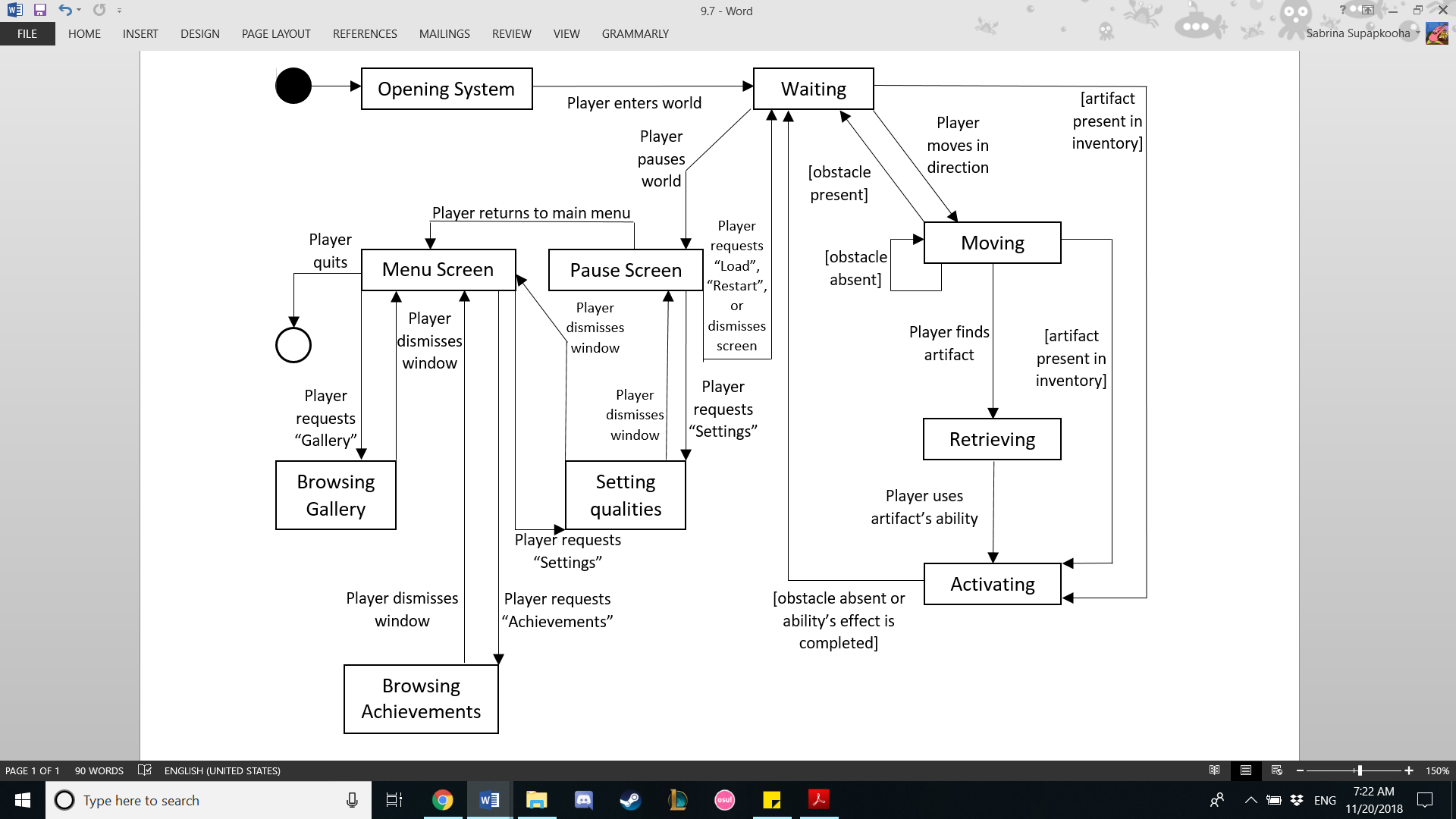
Scenario for Abilities



Scenario for General Gameplay and Game Progression

## 9.7 State logic

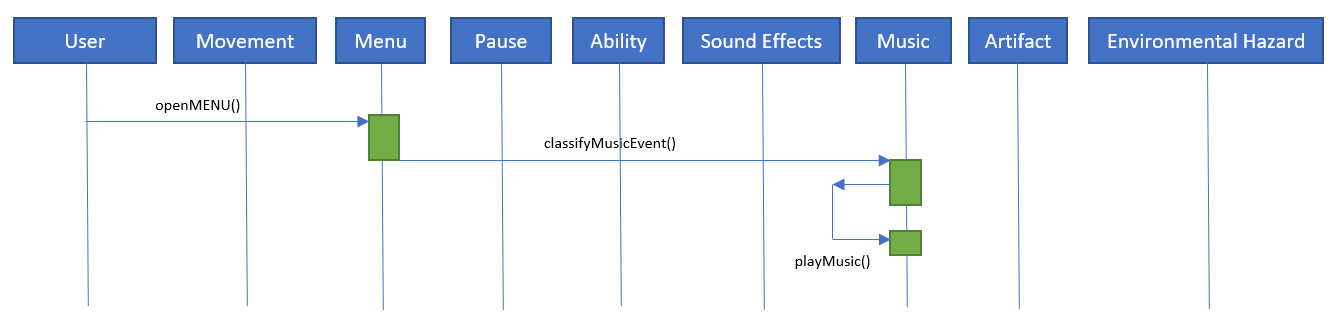
State-transition diagram for Archie-ology video game architecture



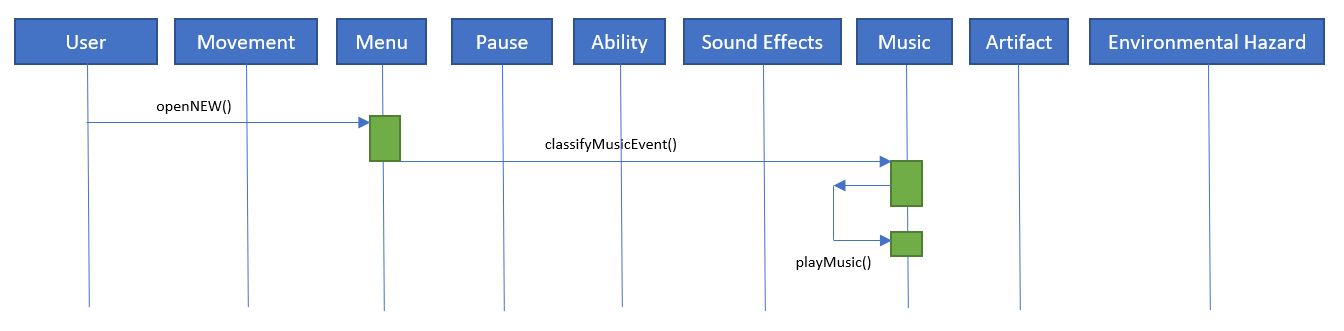
## 9.8 Behavior

# 9.8.1 Sequence Diagram

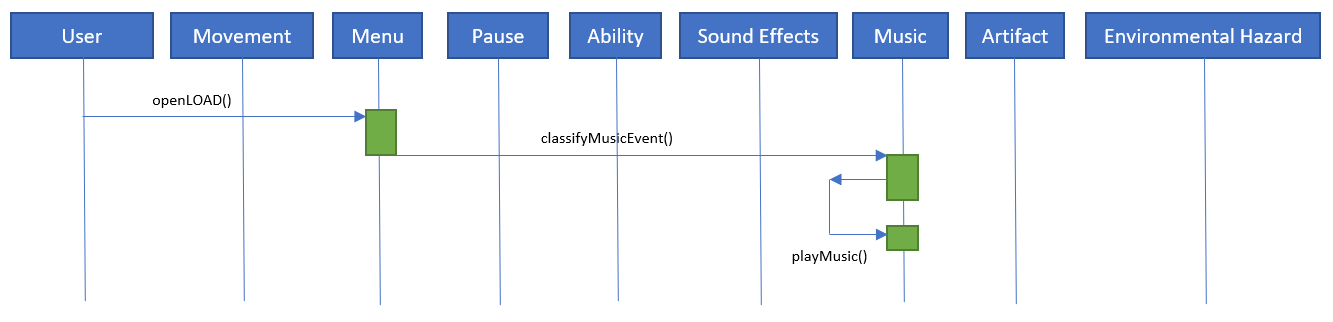
Activate



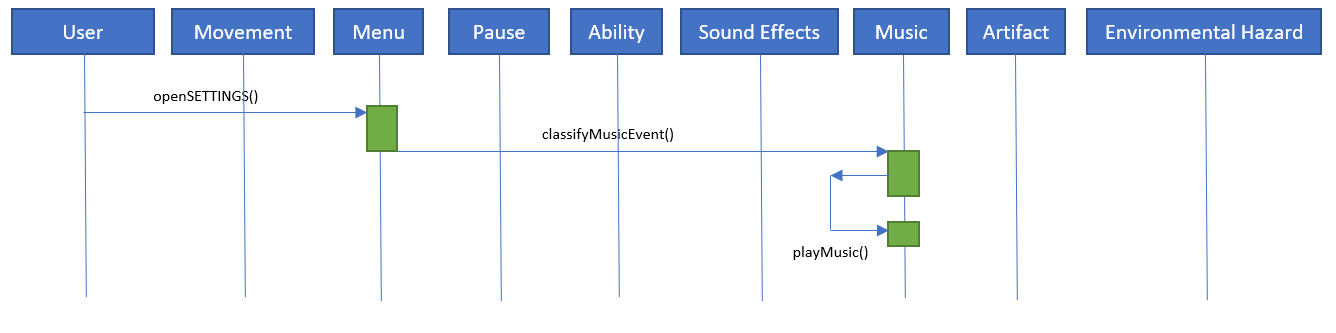
New Game



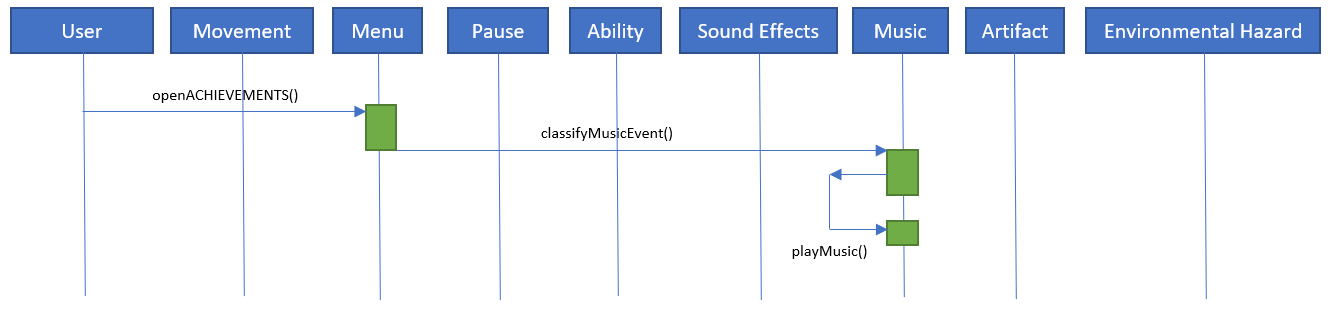
Load Game



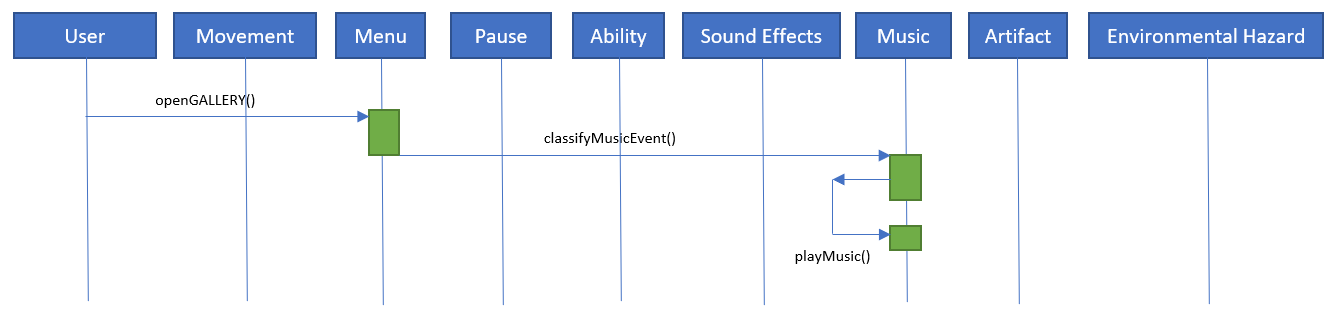
Settings



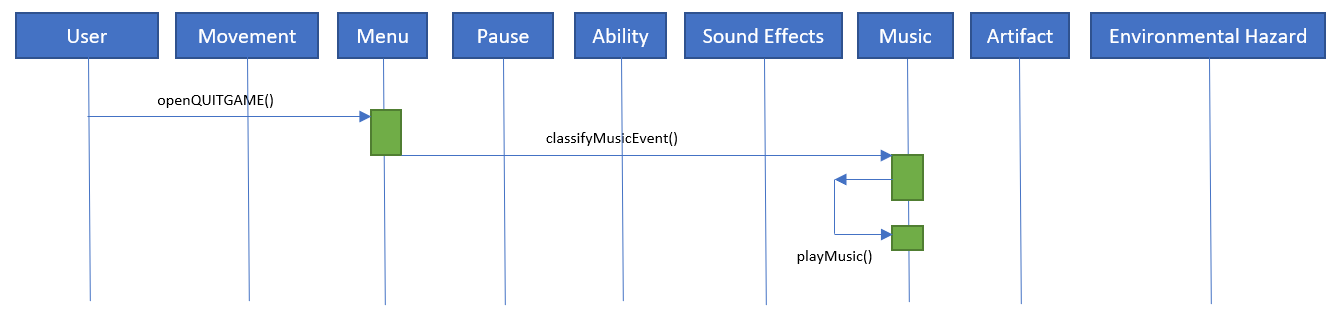
Achievements



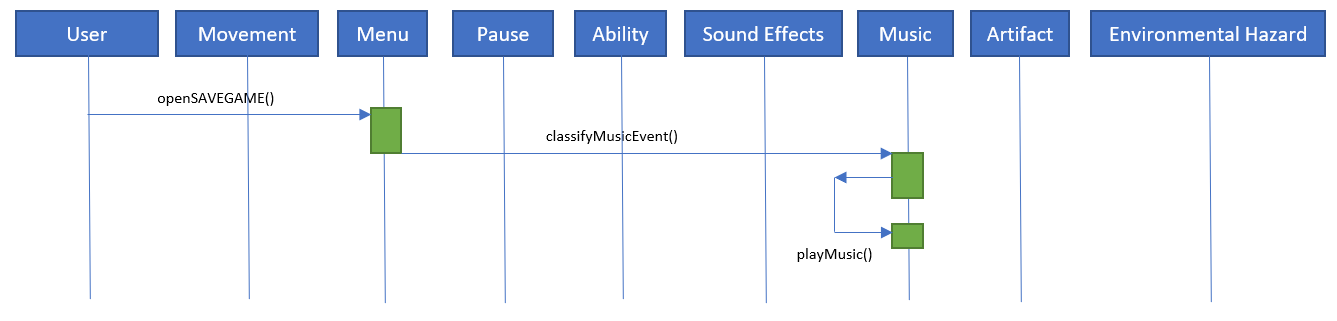
Gallery



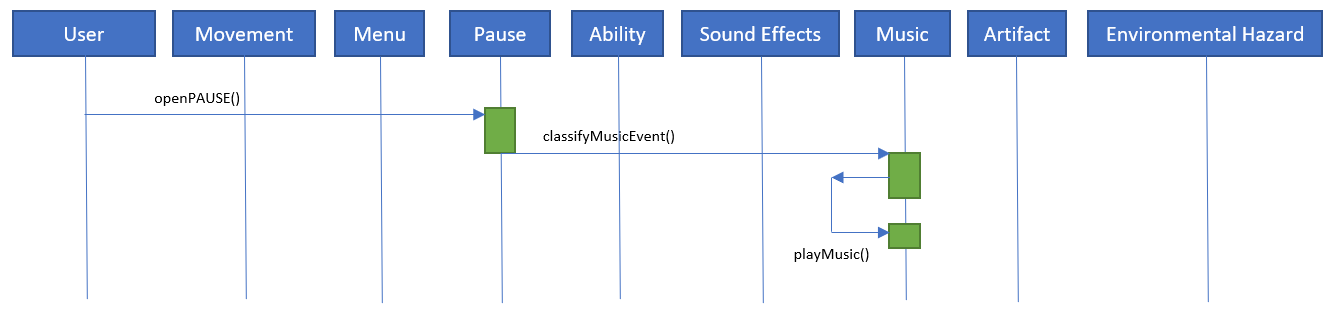
Quit Game



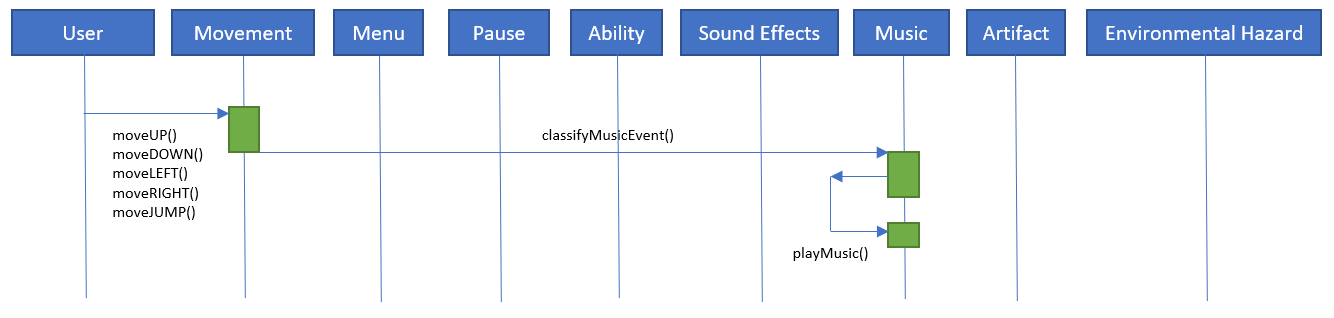
Same Game



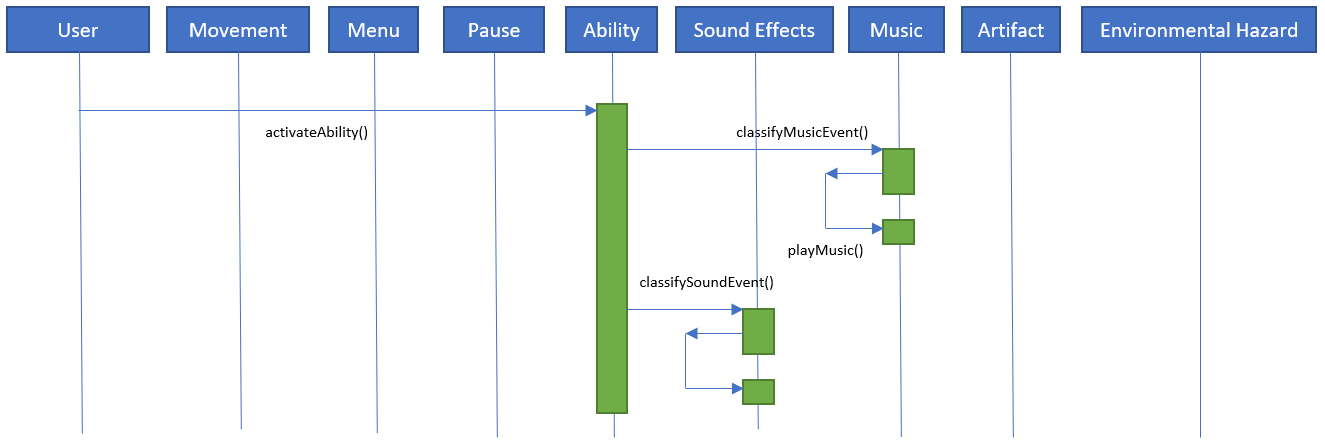
Ingame Menu



Move Player



Ability



**9.8.1 Collaboration Diagram**

|  |  |  |
| --- | --- | --- |
| Object of class | Sends the message | To an object of this class |
| Player | useAbility() | Ability |
| Player | changeMap() | MapChange |
| Player | getSound() | SoundEffect |
| Player | getArtifact() | Artifact |

# 10 SYSTEM TEST PLAN REQUIREMENTS

Our test plan consists of rigorous play testing done by our team and a willing outside personnel. Goals will be set for each play test, and each play test must meet these goals in order to be considered a success. Play tests done by our team will focus on code and implementation; these play tests will be prior to allowing outside play tests. Each new feature coded will first be tested on a baseline project. Then it will be tested on the current iteration with the other features that previously passed this baseline test. Play tests done by outside personnel will focus on learnability of gameplay, impact of aesthetics, and entertainment value.

Team play tests will act as a baseline and will occur after every development (i.e. character movement => environmental movements => character and environment interaction). Once our team agrees on that the game is adequate enough for outside play test, we will begin those play tests. In order to qualify for outside play tests, the game must have: at least 3 rooms completed and smooth character movement. Initially these qualifications will begin at a minimal level and scale upwards based on critique (i.e. initially, rooms must have a background and level design, but sound is not needed yet and level design will not be complex yet). With these play tests, our team aims to create and foster communication between us and the gamers who will find our game entertaining.

Each task will be handled by two team members. A member of the team not part of the pair responsible will check the others’ work for quality assurance and inspections. All members will maintain steady communications to ensure that everyone is on the same page and is aware of current progress. All members will have multiple responsibilities, such as development, resource gathering, and marketing.

SQA processes will include: weekly SCRUM meetings, scheduled inspections, defect/risk tracking and detection, and documentation.

SCRUM meetings will be weekly, either in person or virtual, depending on schedules and time conflicts. SCRUM meetings will follow the traditional methodology without any alterations.

Inspection roles will switch depending on the category of the object under inspection, but one pair of team members will be the moderator, author, and recorder while the other two will act as readers and inspectors. Objects under inspection include any documentation and the system itself.

Inspections will be the primarily method to detect risks and defects, but team members will be on alert for risks and defects while working. Risk management: if a risk is found, team members will group and discuss how to repair the risk as soon as possible.

Documentation, for coding and general information, will be available in the RAS and periodically updated. Any changes in the system, whether intentional or not, must be alerted to the group immediately for debilitation and documentation. Documentation, such as the RAS, will undergo peer review (by our team and other people) and inspections to ensure that documents are comprehensible and polished.

# 11 QUALIFICATION PROVISIONS

This document will be reviewed for quality through rigorous testing. The primary method of quality review will include desk checks peer review walkthrough, and inspection. Weekly scrum meetings will ensure that the document will be completed on time. During each meeting, each member will state what they have completed and what will be completed next. If there are any problems that is blocking progress, they will be brought up during the meeting.

Along with scrum meetings, there will be scheduled inspection sessions. The purpose of these inspections is to identify defects in the document. Defects include anything that makes the document incorrect, inconsistent, or unstable. Defects will not be fixed during these inspections. During inspections, each member of the team will assume a different role so that each member has a category to focus on. Role assignments will be given based on what is under inspection.

In order to manage the defects, each defect will be tracked. Some of the information tracked includes who discovered the defect and on what date. When the defect is fixed, we will also tracked who fixed the defect and when that defect was fixed.

Documentation will be used to ensure that all documents are correct, modifiable, and traceable. By keeping good documentation, we can ensure that everything being produced fulfills the requirements set at the beginning of the process.

# 12 REQUIREMENTS TRACEABILITY

In order to track each requirement from the original source to each instance in succeeding artifacts, there must be documentation stating the changes made to each requirement as well as approval from the rest of the team members before making that change. That piece of documentation also needs to say what has been changed to that requirement. This ensure that each requirement can be tracked throughout each stage of the process.

Github repositories will be used to store various iterations and tests of the game to mitigate the potential loss of work due to technology errors. It will be organized with a unique name that gives the sequence of each copy, the specific team members who edited it, and the main edit made. Documentation will be required that state each code change has been tested first with the basic line copy of the category the specific features fails under, and with a copy of the current iteration. This will have to be signed off by at least two team members and stored online as well as a hardcopy. There will also be a file of documentation of play test feedback that includes the tester, the team member administering the test, and the questions asked along with their answers.

# 13 EVOLUTION OF THE RAS

This document will be updated to include the explicit requirements needed to produce this project. In the future it will dictate the process of requesting and documenting recommended changes. These changes will likely be determined by feedback received during various play testing phases that will track the quality of the document. The specific questions that will be used as metrics for this product.

This document will have an analysis of the potential ways the different that each part can be executed. This document will be updated to reflect the most current vision of this project as the process goes on. The changes made will be recorded as they happen to track the iterations produced and the content of this document will be used to determine if additional changes should be made. Reasons for significant revisions will be listed. New risks that arise will also be recorder when their potential is noticed.

This document will also act as a record of the status of risks already listed in it. Future iterations will also be used to track the schedule, status, and defects of this product. It will be reviewed each time for accuracy in its reflection and the quality of the document itself.

# 14 RATIONALE

# 15 NOTES

# 16 APPENDICES

## Schedule Tracking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated (Hours) | Actual (Hours) | Difference (Hours) |
| RAS 1.0 | For each team member |  |  |  |
|  | Juliet Ramdass | 2.5 | 3 | 0.5 more |
|  | Jeffrey Qiu | 1 | 2 | 1 more |
|  | Sabrina Supapkooha | 2 | 2.5 | 0.5 more |
|  | Jungho Kook | 2 | 2 | 0 |
|  | Summary for entire team | 7.5 | 9.5 | 2 more |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated (Hours) | Actual (Hours) | Difference (Hours) |
| RAS 1.1 | For each team member |  |  |  |
|  | Juliet Ramdass | 0.5 | 1 | 0.5 more |
|  | Jeffrey Qiu | 1 | 1 | 0 |
|  | Sabrina Supapkooha | 1 | 1.5 | 0.5 less |
|  | Jungho Kook | 0.5 | 1 | 0.5 more |
|  | Summary for entire team | 3 | 4.5 | 1.5 more |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated (Hours) | Actual (Hours) | Difference (Hours) |
| RAS 1.2 | For each team member |  |  |  |
|  | Juliet Ramdass | 1 | 2 | 1 more |
|  | Jeffrey Qiu | 2 | 3 | 1 more |
|  | Sabrina Supapkooha | 2 | 2 | 0 |
|  | Jungho Kook | 4 | 3 | 1 less |
|  | Summary for entire team | 9 | 10 | 1 more |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated (Hours) | Actual (Hours) | Difference (Hours) |
| SPMP | Juliet Ramdass | 3 | 4 | 1 more |
|  | Jeffrey Qiu | 4 | 5 | 1 more |
|  | Sabrina Supapkooha | 4 | 5 | 1 more |
|  | Jungho Kook | 3.5 | 3 | .5 less |
|  | Summary for entire team | 14.5 | 17 | 2.5 more |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated (Hours) | Actual (Hours) | Difference (Hours) |
| RAS 1.3 | Juliet Ramdass | 2 | 2 | 0 |
|  | Jeffrey Qiu | 2 | 2.5 | 0.5 more |
|  | Sabrina Supapkooha | 2 | 2.5 | 0.5 more |
|  | Summary for entire team | 6 | 7 | 1 more |

##### Cumulative

|  |  |  |  |
| --- | --- | --- | --- |
| Who (individual or Team) | Estimated (Hours) | Actual (Hours) | Difference (Hours) |
| Team for RAS 1.0 | 7.5 | 9.5 | 2 more |
| Team for RAS 1.1 | 3 | 4.5 | 1.5 more |
| Team for SPMP | 14.5 | 17 | 2.5 |
| Team for RAS 1.2 | 9 | 10 | 1 more |
| Team for RAS 1.3 | 6 | 7 | 1 more |

# Defect Tracking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated | Actual | Difference |
| RAS 1.0 | Juliet Ramdass | 15 | 21 | 6 more |
|  | Jeffrey Qui | 12 | 14 | 2 more |
|  | Sabrina Supapkooha | 17 | 15 | 3 less |
|  | Jungho Kook | 10 | 17 | 7 more |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated | Actual | Difference |
| RAS 1.1 | Juliet Ramdass | 5 | 8 | 3 more |
|  | Jeffrey Qiu | 4 | 8 | 4 more |
|  | Sabrina Supapkooha | 7 | 5 | 2 less |
|  | Jungho Kook | 2 | 5 | 3 more |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated | Actual | Difference |
| RAS 1.2 | Juliet Ramdass | 7 | 4 | 3 less |
|  | Jeffrey Qiu | 6 | 9 | 3 more |
|  | Sabrina Supapkooha | 7 | 10 | 3 less |
|  | Jungho Kook | 4 | 10 | 6 more |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated | Actual | Difference |
| SPMP | Juliet Ramdass | 14 | 13 | -1 |
|  | Jeffrey Qiu | 18 | 17 | -1 |
|  | Sabrina Supapkooha | 14 | 17 | 3 |
|  | Jungho Kook | 16 | 20 | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or Team) | Estimated | Actual | Difference |
| RAS 1.3 | Juliet Ramdass | 5 | 6 | 1 more |
|  | Jeffrey Qiu | 6 | 10 | 4 more |
|  | Sabrina Supapkooha | 6 | 9 | 3 more |

##### Cumulative

|  |  |  |  |
| --- | --- | --- | --- |
| Who (individual or Team) | Estimated | Actual | Difference |
| Team for RAS 1.0 | 54 | 67 | 13 more |
| Team for RAS 1.1 | 20 | 26 | 6 more |
| Team for SPMP | 62 | 67 | 5 more |
| Team for RAS 1.2 | 24 | 33 | 3 more |
| Team for RAS 1.3 | 17 | 25 | 8 more |