Project Mentality

Carson Perry

CST-451 Capstone Project Final Architecture & Design

Grand Canyon University

Instructor: Professor Michael Landreth

Revision: 5

Date: 05/26/21

**ABSTRACT**

This project is now headed to the development phase. Currently, this game has two levels planned, both with a solid idea of how to get into players heads to help them think and empathize with those of the level’s desired mental illness or disorder. This is done through typical game mechanics that the majority of people have seen many times but using it in a different way.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Editor** | **Revision Notes** |
| 04/04/21 | Carson Perry | Initial draft for review/discussion |
| 05/22/21 | Carson Perry | Adding more diagrams and explanations |
| 05/23/21 | Carson Perry | Added more OCD diagrams and description, Schizophrenia description and Model state diagram |
| 05/29/21 | Carson Perry | Finished up with more comments and the abstract |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**TABLE OF CONTENTS**

Design Overview 4

Detailed High-Level Solution Design 5

Detailed Technical Design 6

Appendix A – Technical Issue and Risk Log 7

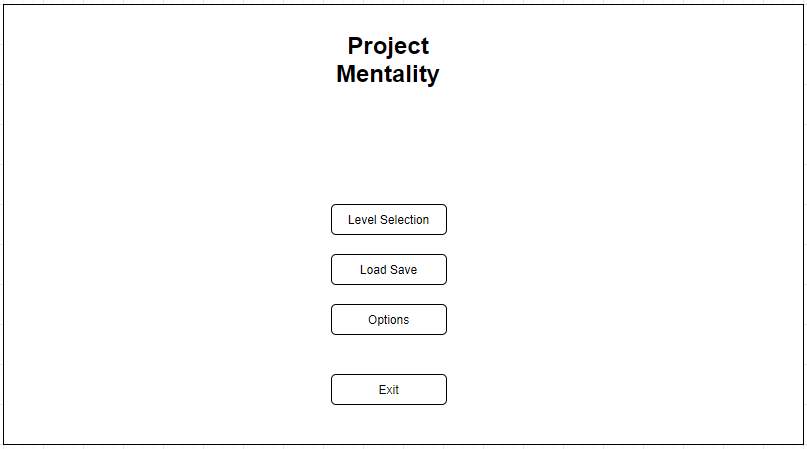
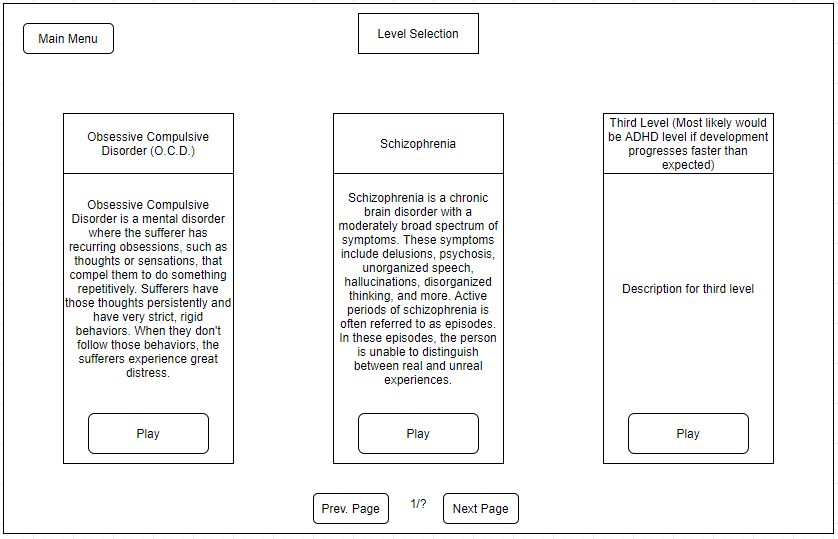
Appendix B – References 8

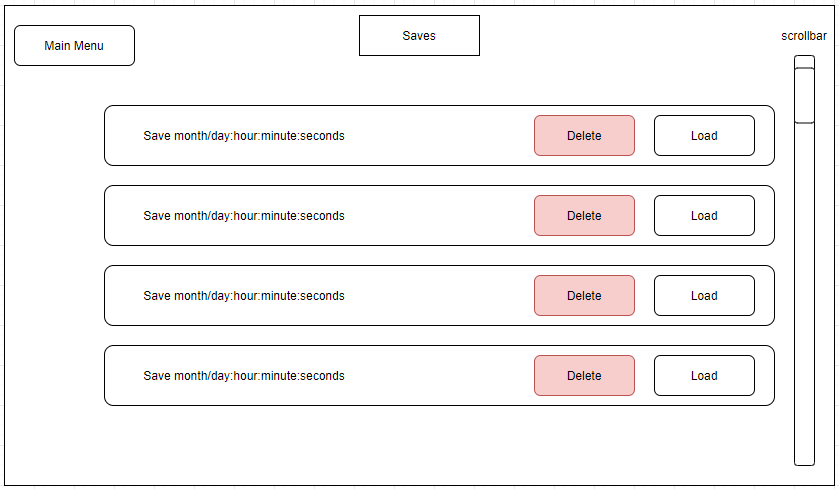
Appendix C – External Resources 9

Design Introduction

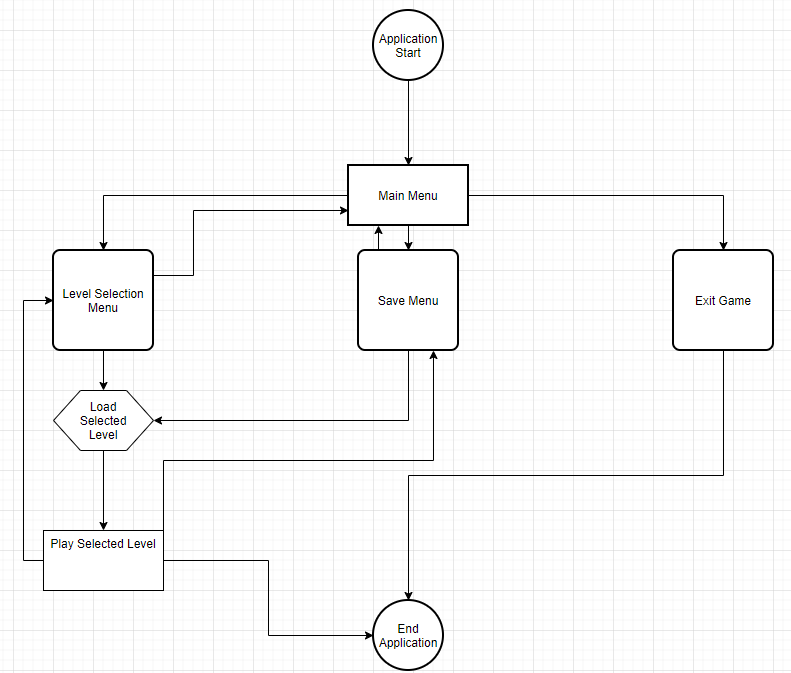
1. The User will select a level in the main menu, then be loaded into the level. Not all levels will have a UI. In fact, the OCD level is the only level with planned UI outside of text appearing at the bottom of the screen for dialogue. Minimal UI is to help with immersion, however with the OCD design, the UI serves the purpose of placing a sense of importance of the objectives of a typical morning, though there’s no real reason to do the tasks in that order, and it’s purely in the players head that they need to do those tasks in that order, just as someone with OCD thinks. The UI for that level serves the purpose of furthering perspective in the player’s head.

UI Screenshots

* 1. Main Menu:
  2. Level Selection Menu
  3. Load Save Menu



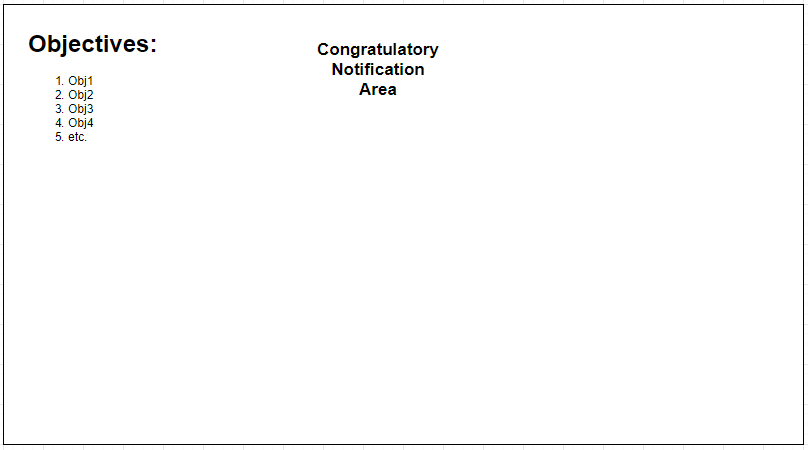
* 1. User Experience Flowchart

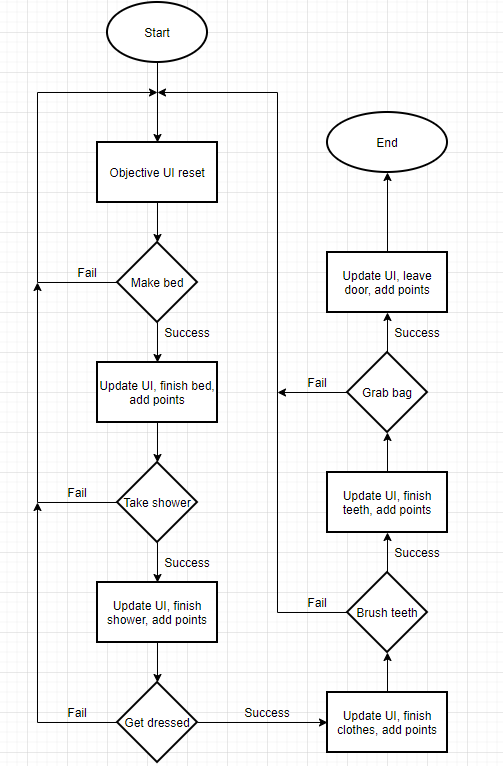
:

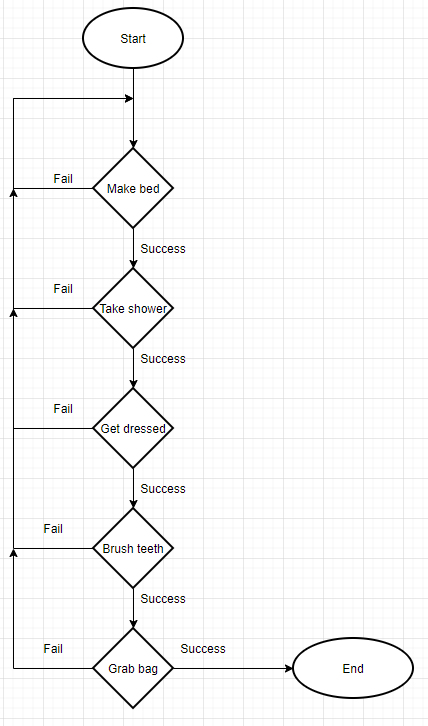
OCD Level UI and Objectives Control Flow Diagram

The UI and the actual hidden objects are on two separate systems. The UI’s purpose is to get in the head of the player by use of the on-screen objectives and point system. There’s no real reason to complete the objectives in this specific order, however the player will feel as though they have to because all their progress is reset and they have to redo everything if they get something wrong or complete something out of order. However, as more objectives are added on and become increasingly ridiculous and have no point, the player will eventually learn that they could have completed all of the objectives regardless of order and left without any repercussions. But it feels bad to lose all your points and progress. This helps to emphasize the distress that sufferers of OCD have when breaking their behaviors. By showing the players that the only reason to do any of those objectives in order, or redo them after failing, was all in their head, it helps empathize the player with the symptoms of OCD.

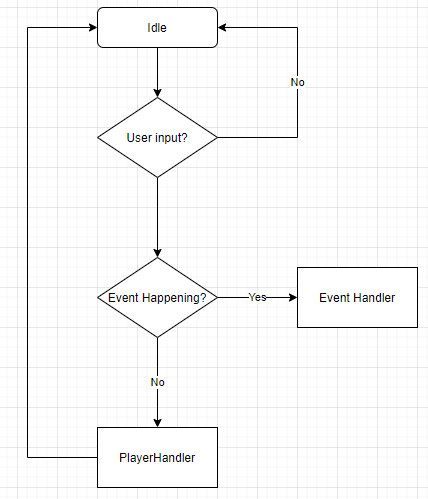
UI:







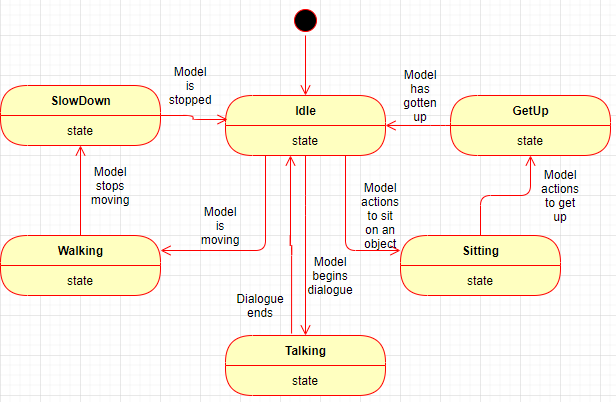
User Input Flowchart



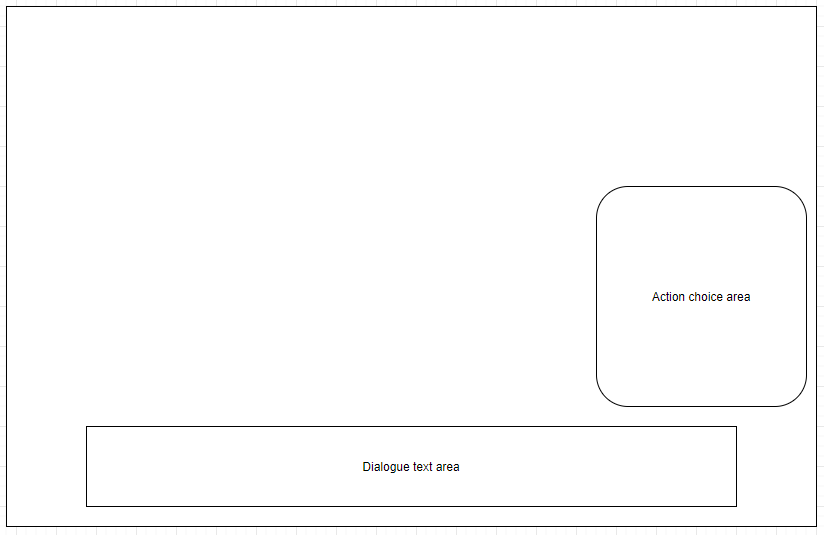
Schizophrenia Level:

The schizophrenia level won’t have much of a UI except for a text box at the bottom, and options on the right side of the screen. This will be the first level to include other characters outside of the player. In the previous level about OCD, the only character is the player, and there’s no need to render an entire player object into the scene, so there will only be bits and pieces that the player will see as animations that trigger from actions. The animation that plays for the model depends on the object’s current state. The state diagram shows how those states interact with each other, with each state having its own animation. In this level, the player will go through events in a fantasy world. The purpose of the setting being a fantasy world is to disconnect the player’s knowledge of reality and the world, from the game’s world. This way, it’s much easier to confuse the player on what is and isn’t real in the world. If you had a realistic level and had a train fall out of the sky, that would immediately break immersion and the player would immediately understand that the character’s schizophrenia is active. But, if you’re in a world where trains fly and can break down and fall from the sky, that is a much more reasonable thing to happen. The player will go through multiple events that contain real and unreal elements. Choosing bad options during an episode would bring bad consequences. Such as fighting off a perceived burglar, who is actually your friend, or breaking a mirror because it’s presented as a portal to allow an invading army into your room. Even an event as simple as believing you are having a conversation with a friend, but when you chat with them later, find out that conversation never actually took place.

Character Model State Diagram:

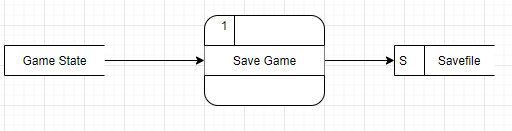


Schizophrenia Level UI:



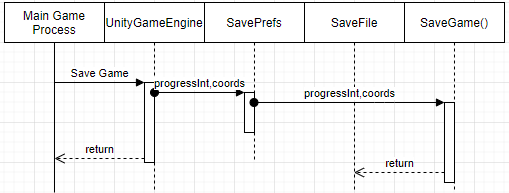
Saving:

The Dataflow for saving is very simple, the game state provides information for the save game process, which then creates a file with the save information



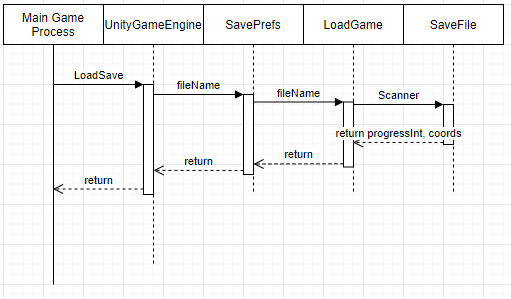
As the player progresses through each level, regardless of whether the player is aware, they are passing checkpoints. These checkpoints will increment a progress integer that allows the program to know where they player has progressed to. This int is what will be saved, along with coordinates of where the player is on the map. This information will be serialized and verified within the SavePrefs C# script to ensure that the information actually works with the game, and isn’t a custom file to break the application, or corrupted data.

Save Game Sequence Diagram:



For loading a game, when clicking on the available save files, it will send the name of the chosen file. This file is what will supply the game with the information to render the game to what it was during when the game was saved.

Load Game Sequence Diagram:



1. Use the template to list the project deliverables that are to be included external to this Design Specification (Data Dictionary, API Design, etc.).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Deliverable Acceptance Log | | | | | |
| ID | Deliverable Description | Comments | Evaluator (internal or external as applicable) | Status | Date of Decision |
| 1 | UI Wireframes | Wireframes for the planned UI of the Main Menu and the OCD level | Carson Perry | Complete | 02/18 |
| 2 | User Input Flowchart | Flowchart for how user input is interpreted and sent | Carson Perry | Complete | 04/04 |
| 3 | User Experience Flowchart | Flowchart to show what the user experience will look like as a flowchart | Carson Perry | Complete | 05/22 |
| 4 | Objective Control Flow Diagrams | CFD to show how the objective system for the OCD level is structured | Carson Perry | Complete | 05/22 |
| 5 | Schizophrenia Models State Diagram | State Diagram to show the various states that the models in the schizophrenia level have. Their model animation should change based on the state they are currently in. | Carson Perry | Complete | 05/23 |
| 6 | Save Game Data Flow Diagram | DFD to show the data flow of saving the game. | Carson Perry | Complete | 05/23 |
| 7 | Save and Load Game Sequence Diagrams | Sequence Diagram to show the way the software orders and structures saving and loading the game. | Carson Perry | Complete | 05/24 |

Detailed High-Level Solution Design

1. Provide a detailed overview of how the proposed design fits into the overall solution.
   1. This solution will abuse the mentality of playing a game and immersing yourself in a game, to get players into the mindset of various

|  |  |  |
| --- | --- | --- |
| Proof of Concepts | |  |
| **Description** | **Rationale** | **Results** |
| 1.Not Applicable |  |  |
| 2 - |  |  |
| 3 - |  |  |
| 4 - |  |  |
| 5 - |  |  |

|  |
| --- |
| Hardware and Software Technologies |
| 1 - Windows 7 SP1+, macOS 10.12+, or Ubuntu 16.04+ |
| 2 – CPU that can handle SSE2 instruction sets |
| 3 - GPU capable of using DX10 |
| 4 – Unity Game Engine |
| 5 – FL Studio |
| 6 - Blender |

**Logical Solution Design:**

Not Applicable

**Physical Solution Design:**

Not Applicable

Detailed Technical Design

**General Technical Approach:**

This game’s purpose is not to immerse the users into a game, nor for it to be massively entertaining. It’s purpose is to get into the heads of the players and get them to think in as similar a fashion as those with the desired mental illness. A game that you just wake up and do a morning routine isn’t very fun, but the purpose is to get the player to think they have to do specific objectives in a specific order, resembling those rigid behaviors that OCD sufferers show. Or to place the player in a fantasy world, not for aesthetic reasons, but to break their knowledge of the world, where anything is possible since it isn’t based on our own world. This will make it much harder to discern real and fake to better challenge them in discerning what’s real and what’s fake, just like a schizophrenic.

**Key Technical Design Decisions:**

I am choosing Unity Game Engine, since it’s an amazing free game development suite and that I’ve used it before. I have experience making 2-dimensional games on this before, but I’ve never made a 3-dimensional game.

**Database ER Diagram:**

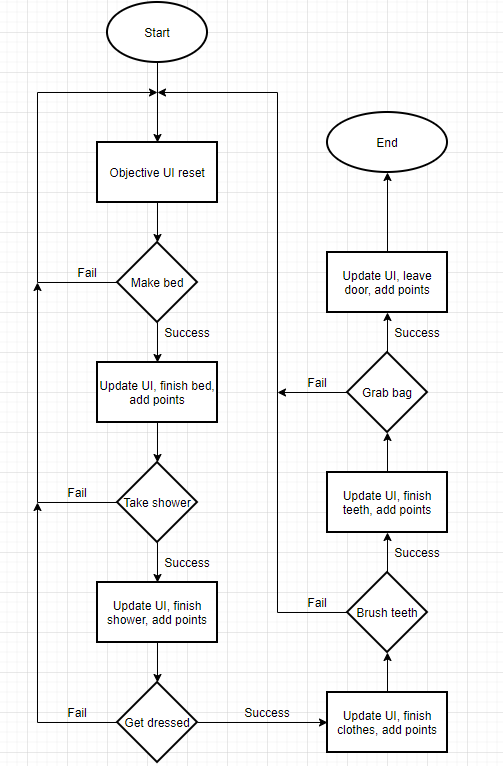
Not applicable

**Database DDL Scripts:**

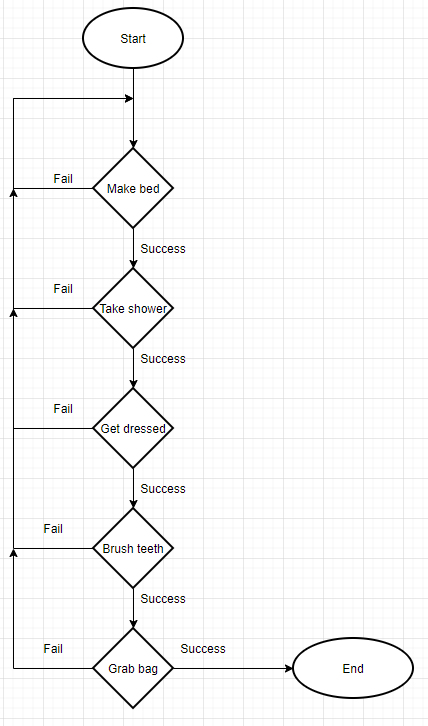
Not applicable

**Flow Charts/Process Flows:**

Fake Objectives Control Flow Diagram:



Real Objectives CFD (Note: These are not to be done in order, this is a check that occurs when the player tries to leave through the door)

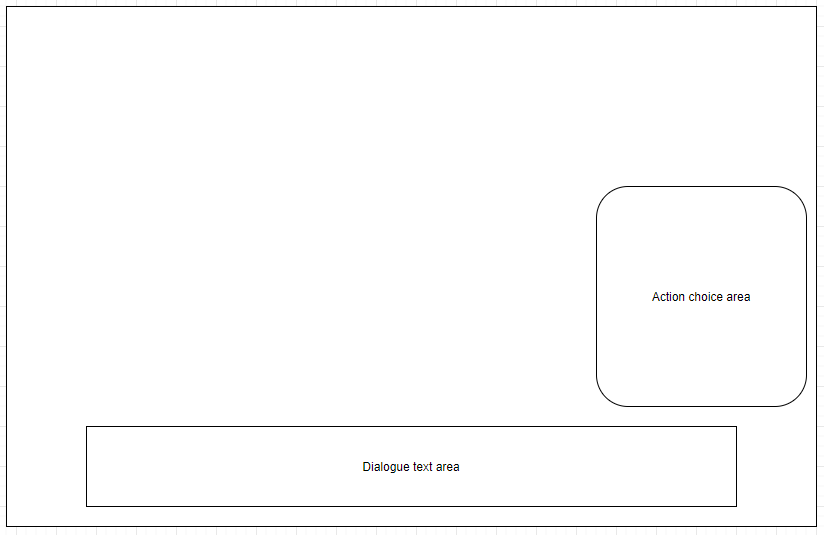


**Sitemap Diagram:**

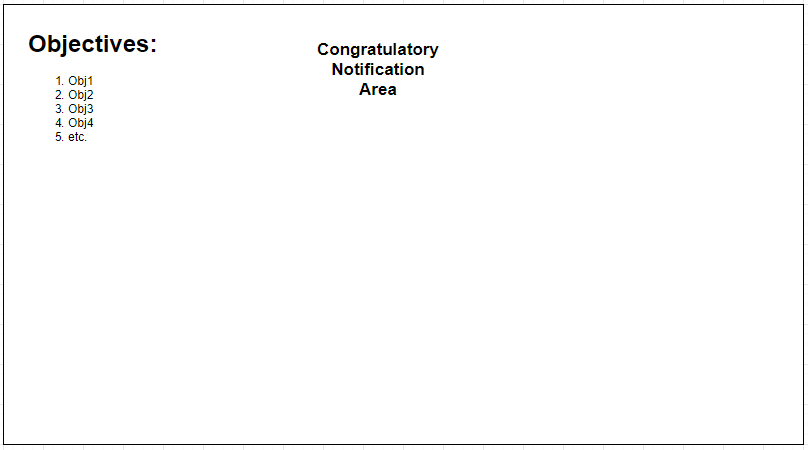
Not applicable

**User Interface Diagrams:**

Schizophrenia UI:

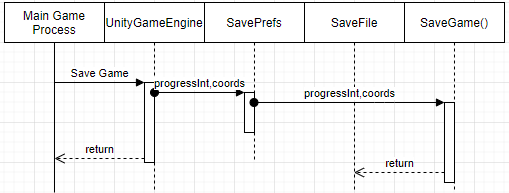


OCD UI:

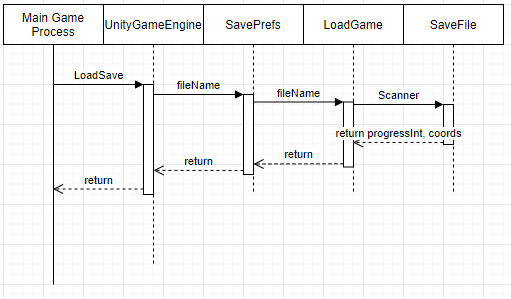


**UML Diagrams:**

Save Game Sequence Diagram:



Load Game Sequence Diagram:



**Service API Design:**

Not applicable

**NFR’s (Security Design, etc.):**

Not applicable

**Operational Support Design:**

Not applicable

**Other Documentation:**

Not applicable

Appendix A – Technical Issue and Risk Log

1. Use the template to identify and monitor project issues and risks.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues and Risk Log | | | | | | | | |
| **Issue or Risk** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| I/R | What is the issue or risk? | How will this impact scope, schedule, and cost? | How do you intend to deal with this issue? | Who manages this issue? |  |  |  |  |
| R | I’ve never made a 3-dimensional game | Can’t have too big of a scope, as I need to learn a lot of tools in Unity and external tools. | Going through and reading the online materials I’ve bought for creating games in Unity. | Carson | High | 05/29/21 |  |  |

Appendix B – References

*No References in this document*

Appendix C – External Resources

|  |  |
| --- | --- |
| **GIT URL:** | *The GIT URL (if applicable).* |
| **Hosting URL:** | *The Hosting URL (if applicable).* |