**Tac Tac Toe**

**Software Design Document**

**Version 1.0**

**Kyle Shermer**

**7/21/2020**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 7/21/2020 | 1.0 | Initial version of the document | Kyle Shermer |
|  |  |  |  |

**Table of Contents**

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

1.5 Overview

2. Glossary

3. Use Cases

3.1 Actors

3.2 List of Use Cases

3.3 Use Case Diagrams

4. Design Overview

4.1 Introduction

4.2 System Architecture

4.3 System Interfaces

* 1. Constraints and Assumptions

5. System Object Model

5.1 Introduction

5.2 Subsystems

5.3 Subsystem Interfaces

6. Object Descriptions

6.1 Objects

7. Object Collaboration

7.1 Object Collaboration Diagram

8. Data Design

8.1 Entity Relationship Diagram

9. Dynamic Model

9.1 Sequence Diagrams

9.2 State Diagrams

10. Non-functional Requirements

10.1 Performance Requirements

10.2 Design Constraints

11. Supplementary Documentation

11.1. Tools Used to Create Diagrams

Software Design Document

1. **Introduction**

The purpose of this software design document is to provide documentation which will be used in the conjunction with the software development, by providing details of how the product will be built. Contained within this document are the narrative and graphical documentation of the design for the project including use case models, sequence diagrams, etc…

* 1. **Purpose**

The purpose of the software design document is to provide an insight into the design of the system to fully allow the software development to continue with an understanding of what is to be built.

* 1. **Scope**

This software design document represent a base level system which will be used as a proof of concept to build a product that provides base level functionality. This will also be used as an example to prove feasibility for large scale production use.

* 1. **Definitions, Acronyms, and Abbreviations**

XML – eXtensible Markup Language, http://en.wikipedia.org/wiki/XML

* 1. **References**

Tac Tac Toe Software Design Document Version 1.0, Last Updated on 7/26/20

* 1. **Overview**

This Software Design Document is divided into 11 sections with the subsections listed below

1. Introduction

2. Glossary

3. Use Cases

4. Design Overview

5. System Object Model

6. Object Descriptions

7. Object Collaborations

8. Data Design

9. Dynamic Model

10. Non-functional Requirements

11. Supplementary Documentation

**2 Glossary**

This section will not be used during this time.

**3 Use Cases**

**Use-Case Model Survey**

**3.1 Actors**

**3.1.1 Game Player**

3.1.1.1 Information: The game player is a user who would utilize the tic tac toe app. The game player would perform such actions as initiating the game, performing actions such as utilizing the tic tac toe board to win the game, and load the last saved previous version of the game if he/she desires.

**3.1.2 Computer Player**

3.1.2.1 Information: The computer player would utilize the tic tac toe board to perform moves against the player. The computer player would also predetermine the optimal moves to win or decide if it needs to block the player to prevent the player from winning.

**3.2 List of Use Cases**

**3.2.1 Tic Tac Toe use cases**

3.2.1.1 Starting the game

3.2.1.2 Loading/Saving the game

3.2.1.3 View the game board

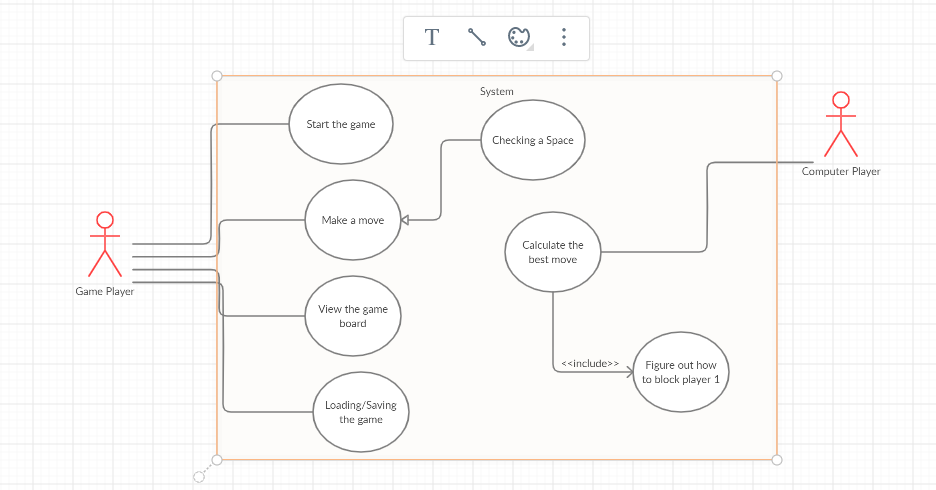
3.2.1.4 Making a move

3.2.1.4 Computer player calculates the best move

3.2.1.5 Computer player determines if it needs to block the player

**3.3 Use case diagrams**

**3.1 Tic Tac Toe use case**



**4 Design Overview**

**4.1 Introduction**

This Design Overview section is meant to give a brief overview of the design. The System Architecture represents a way to give an overall view of the system and how it fits in with external systems. This allows the user and reader to familiar themselves with the design before proceeding into the overall details.

**4.2 System Architecture**

None specified at this time

**4.3 System Interfaces**

**4.3.1 External User Interface Requirements**

4.3.1.1 User Interface

The user interface will allow the user to start the game and load their last previous saved game if they desire. The user interface will present the user with a tic tac toe board and when the user clicks within the panel and X will appear. The user interface will also display the computer player moves and perform calculated moves to block player 1 moves if necessary. The user interface will also keep track of the player one’s name and keep track of the wins/losses against the computer player.

4.3.1.2 Software Interfaces

None specified at this time

4.3.1.3 Communication Interfaces

None specified at this time

**4.4 Constraints and Assumptions**

4.4.1 List of Assumptions

None specified at this time

4.4.2 List of Dependencies

None specified at this time

**5 System Object Model**

**5.1 Introduction**

The System Object Model Section allows the description of the subsystems that are in use. This would allow to describe the system in the overall manner to demonstrate the different groups in their respective subsystems.

**5.2 Subsystems**

None specified at this time

**5.3 Subsystem Interfaces**

None specified at this time

**6 Object Descriptions**

**6.1 Objects**

None specified at this time

**7 Object Collaboration**

**7.1 Object Collaboration Diagram**

None specified at this time

**8 Data Design**

**8.1 Entity Relationship Diagram**

None specified at this time

**9 Dynamic Model**

**9.1 Sequence Diagrams**

None specified at this time

**9.2 State Diagrams**

None specified at this time

**10 Non-functional Requirements**

**10.1 Performance Requirements**

**11 Supplementary Documentation**

**11.1 Tools Used to Create Diagrams**

**11.1.1 UML Modeling Tools**

11.1.1.1 Creately