

Software Requirements Specification for
SFWRENG 4G06:
Dice Duels: Duel of the Eights

Team 9, dice_devs

John Popovici

Nigel Moses

Naishan Guo

Hemraj Bhatt

Isaac Giles

January 29, 2025

Contents

| | | |
|----------|--|-----------|
| 1 | Reference Material | iv |
| 2 | Introduction | 1 |
| 2.1 | Purpose of Document | 1 |
| 2.2 | Scope of Requirements | 1 |
| 2.3 | Characteristics of Intended Reader | 2 |
| 2.4 | Organization of Document | 2 |
| 3 | General System Description | 4 |
| 3.1 | System Context | 4 |
| 3.2 | User Characteristics | 5 |
| 3.3 | System Constraints | 5 |
| 4 | Specific System Description | 6 |
| 4.1 | Problem Description | 6 |
| 4.1.1 | Terminology and Definitions | 6 |
| 4.1.2 | Goal Statements | 8 |
| 4.2 | Solution Characteristics Specification | 12 |
| 5 | Stakeholders | 13 |
| 5.1 | Traditional Yahtzee enthusiasts | 13 |
| 5.2 | Video Game enthusiasts | 13 |
| 5.3 | Personas | 14 |
| 5.4 | Priorities Assigned to Users | 15 |
| 5.5 | User Participation | 15 |
| 5.6 | Ongoing Support | 15 |
| 6 | Requirements | 16 |
| 6.1 | Functional Requirements | 16 |
| 6.2 | Non-Functional Requirements | 19 |
| 7 | Requirements Implementation Roadmap (Phase In Plan) | 22 |
| 7.1 | Critical Priority | 22 |
| 7.2 | High Priority | 22 |
| 7.3 | Non-Functional Requirements Priority | 23 |
| 7.4 | Stretch Goals | 23 |
| 8 | Standards, Codes, Legal, and Regulatory Factors | 24 |
| 8.1 | Data Privacy and Protection | 24 |
| 8.2 | Network and Online Standards | 24 |
| 8.3 | Game Fairness and Anti-Cheating Measures | 24 |
| 8.4 | Legal Considerations for Online Play | 24 |

| | | |
|-----------|---|-----------|
| 8.5 | Accessibility Standards | 25 |
| 8.6 | Intellectual Property and Copyright | 25 |
| 8.7 | Regulatory Requirements for Online Play | 25 |
| 9 | Standards Compliance Roadmap | 26 |
| 10 | Likely Changes | 27 |
| 11 | Unlikely Changes | 29 |
| 12 | Traceability Matrices and Graphs | 31 |
| 13 | Development Plan | 34 |
| 14 | Values of Auxiliary Constants | 34 |
| 15 | Commonalities | 35 |
| 15.1 | Dice Rolls | 35 |
| 15.2 | Yahtzee Hands | 35 |
| 15.3 | Score Sheet Structure | 35 |
| 15.4 | Simultaneous Play | 35 |
| 15.5 | 2-Player or Single-Player Only | 35 |
| 15.6 | Always Playing Against an Opponent | 35 |
| 16 | Variabilities | 36 |
| 16.1 | Number of Dice | 36 |
| 16.2 | Sides on Dice | 36 |
| 16.3 | Individual Sides | 36 |
| 16.4 | Scoring Calculation | 36 |
| 16.5 | Time Per Turn | 36 |
| 16.6 | Hand Restrictions | 36 |
| 17 | Parameters of Variations | 37 |
| 17.1 | Number of Dice | 37 |
| 17.2 | Sides on Dice | 37 |
| 17.3 | Individual Sides | 37 |
| 17.4 | Scoring Calculation | 37 |
| 17.5 | Time Per Turn | 37 |
| 17.6 | Hand Restrictions | 37 |

Revision History

| Date | Developer(s) | Change |
|------------|---------------|---|
| 2024-09-27 | John Popovici | Set up formatting |
| 2024-09-30 | Nigel Moses | Added three blank sections (Commonalities related sections) |
| 2024-10-04 | Nigel Moses | Added content to section 11, 12, 13 |
| 2024-10-05 | Hemraj Bhatt | Added content to section 2 and 3 |
| 2024-10-05 | John Popovici | Added content to section 4.1 and 4.1.1 |
| 2024-10-05 | John Popovici | Fixed compiling issues and formatting in section 3 |
| 2024-10-07 | John Popovici | Added goals and stretch goals to section 4.1.3 |
| 2024-10-07 | Isaac Giles | Added functional and non-functional requirements to section 5 |
| 2024-10-08 | John Popovici | Removed sections based on physical description |
| 2024-10-08 | John Popovici | Added stakeholders section template |
| 2024-10-09 | Hemraj Bhatt | Added content to section 9 |
| 2024-10-09 | Naishan Guo | Added Likely and Unlikely changes to section 6 |
| 2024-10-09 | John Popovici | Added content to stakeholder section |
| 2024-10-09 | John Popovici | Added content to reflection and removed comments |
| 2024-10-10 | Isaac Giles | Updated non-functional requirements and added to reflection |
| 2024-10-10 | Hemraj Bhatt | Updated content on section 2 and 3 |
| 2024-10-10 | Nigel Moses | Added content to Traceability section |
| 2024-10-11 | John Popovici | Fixed document layout and formatting |
| 2024-10-11 | John Popovici | Updated document introduction |
| 2024-10-11 | Isaac Giles | Updated a NFR for better accuracy |
| 2024-10-25 | Nigel Moses | Added Requirement from Hazard Analysis, and updated traceability matrices |
| 2024-11-22 | John Popovici | Clarified dice details |
| 2024-11-22 | John Popovici | Added requirements to changes section and updated traceability matrix |
| 2024-11-30 | Isaac Giles | Made various modifications throughout document to improve notations and conventions |
| 2025-01-24 | John Popovici | Refined mathematical formalization |
| 2025-01-29 | John Popovici | Added explicit requirements fit criterion |
| ... | ... | ... |

Table 1: Revision History

1 Reference Material

This section records information for easy reference.

This section has remained blank due to the simplistic nature of our problem space. No unit symbols are used in our document and anywhere a unit is used, it is stated. Additionally all abbreviations and acronyms are explained where they first appear in the document and are from there on out self-explanatory and contextually understandable. There are also no mathematical formulas or mathematical notation within our software requirements specification document. This document was compiled for the SFWRENG 4G06 course, the software engineering capstone course at McMaster University.

2 Introduction

Do you know the game Yahtzee? Played with dice, it's like Poker but more varied, requiring different skills, without betting. It uses 5 dice, with the usual dots representing the numbers 1 to 6. In rolling the dice, players try to create specific formations, thereby scoring points. At the end of a set number of rounds of dice rolls, the player with the higher score wins. But the cube isn't the only possibility for dice as the octahedron, with 8 sides could be used. Scoring could likewise be done each round. The number of dice could be changed. These suggest an expanded version of Yahtzee.

This project creates an online multiplayer game platform that allows for the creation of custom Yahtzee-like games and variants. This family of games will come with some presets such as classic Yahtzee, Dr. Paul's octahedron version, and more, but will allow for the users to set their own variables such as number of dice, what kind of dice, and some elements of scoring, and then play that game. Kinds of dice would include cubes and octahedrons, among other multi-sided dice, and scoring could be calculated at the end of the game, as in classic Yahtzee, or on a per-round basis, where hands go in a head-to-head matchup.

This section includes a general overview of the entire SRS document providing descriptions of all sections. It also outlines of the areas of knowledge needed to grasp the documentation accurately.

2.1 Purpose of Document

The Software Requirements Specification (SRS) document aims to clearly define the functional and non-functional needs of the project. So all parties involved have a common understanding of the objectives and requirements of the project. Throughout the whole software lifecycle, the SRS will serve as the development team's fundamental guide, aiding in the phases of design, implementation, and testing. In addition, it will facilitate effective communication between stakeholders and the development team while creating a framework that ensures that the final result satisfies user requirements and is in line with the project's pre-determined goals.

2.2 Scope of Requirements

The scope of the project involves developing a modified Yahtzee game that features adjustable numbers of playable dice and various game attribute variations. This game will be available for both offline and online play, offering single-player and multiplayer. The scope does not include the implementation of advanced graphics, advanced computer opponents and other board games.

2.3 Characteristics of Intended Reader

The intended readers of this SRS document are mainly software developers and game designers with an understanding of game development and game design.

- The reader should have a basic understanding of the classic Yahtzee game, and well as an understanding of different dice types and their uses.
- Familiarity with casual gaming and turn-based gameplay will be beneficial for the reader's understanding of the decision making process.
- Basic engineering requirements understanding is required to understand the breakdown of the SRS document and its purpose.

2.4 Organization of Document

This SRS document is structured to provide a clear roadmap for readers. The sections are as follows:

- **Introduction**
 - Outlines the purpose and scope of the SRS.
- **General System Description**
 - Outlines an overview of the project, along with user characteristics, system constraint and the interfaces between the system and its environment.
- **Specific System Description**
 - In-depth system description containing high-level problem and goal descriptions.
- **Stakeholders**
 - Outlines stakeholder groups and relevant system user information.
- **Requirements**
 - Outlines both functional and non-functional requirements of the system.
- **Likely Changes**
 - Outlines anticipated modifications to the system.
- **Unlikely Changes**
 - Outlines aspects of the system that will remain static.
- **Traceability Matrices and Graphs**

- Tracking of requirements throughout the development process.
- **Development Plan**
 - Outlines the general timeline and milestones for the project.
- **Values of Auxiliary Constants**
 - Outlines constant parameters used in report.
- **Commonalities**
 - Outlines commonalities between different aspects of the system.
- **Variabilities**
 - Outlines variables between different aspects of the system.
- **Parameters of Variations**
 - Outlines the different variations of aspects of the system.
- **Appendix — Reflection**
 - In-depth insights into the development process and lessons learned.

3 General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

3.1 System Context

System context includes an abstract view of the software following the design pattern of Inputs \rightarrow Calculations \rightarrow Outputs. The user provides inputs, such as game settings and preferences, the system performs the necessary calculations (e.g., score tracking, dice roll simulation), and then outputs the results to the user. The system will also interact with external entities, such as players in a multiplayer environment.

The following high-level requirements are relevant to this context:

- The game must support online multiplayer functionality, allowing two players to play against each other.
- The system must calculate scores based on standard Yahtzee rules.
- The game will feature realistic dice rolling physics and random outcomes.
- The game will provide a user interface that displays essential game information (scores, player names, dice states).
- Players will be able to customize game settings and access different game modes.

Inputs:

- Game settings (number of dice, player names, game modes).
- Player actions (dice selection, roll decisions).

Outputs:

- Game results (scores, dice outcomes).
- Feedback on game progress (turns, current scores, remaining rolls).

External Entities:

- Other players (for online multiplayer functionality).
- Peer-to-peer networking services for connecting players.

User Responsibilities:

- Provide valid game input data, such as names and settings.
- Make strategic decisions regarding dice rolls and game modes.

System Responsibilities:

- Ensure that all input data is properly validated (e.g., detect type mismatches).
- Perform calculations for scorekeeping and game logic.
- Provide feedback and results to the user in an intuitive interface.

Additionally, the system is intended for casual use, primarily focused on entertainment and social interaction. It is not a mission-critical or safety-critical system, which will influence the degree of formality in the system design.

3.2 User Characteristics

The user characteristics for the game can be considered to be the following and are further developed in the stakeholders section:

- The end user of *Dice Duels: Duel of the Eights* should have an understanding of the Yahtzee game, including its rules and strategies.
- Users should have a rudimentary understanding of probability theory, as it is important for comprehending the game's mechanics and making informed decisions during gameplay.
- Familiarity with casual gaming and turn-based gameplay will be beneficial for the user's overall enjoyment and understanding of the pacing of the game.
- Basic digital literacy is expected, particularly in understanding how to adjust game settings, access help menus, or troubleshoot simple connectivity issues.

3.3 System Constraints

The following constraints must be adhered to during the development of the system:

- The game must use the Godot engine exclusively for its development.
- The codebase must be written in GDScript and C# and use .NET framework exclusively.
- The system must support peer-to-peer and multiplayer capabilities.

4 Specific System Description

This section presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, definitions and finally the instance models.

4.1 Problem Description

Games are a staple of entertainment and are used to bring people together for both competition and fun. There can often be a desire to share the experience of playing a game with someone but meeting can be hard or impossible in-person and so having an online version of popular games allows for such opportunities. One such game is the game of Yahtzee, which while it does have online versions, are limited to the classic rule-set and do not allow for variants to be designed and played. *Dice Duels: Duel of the Eights* looks to solve the issue of not having online access to Yahtzee and the ability to create custom versions of the game and play them.

4.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- Dice rolls can be given in the form AdX where A and X are variables representing the number of dice and number of sides, respectively. When A is 1, its inclusion is optional. For example $4d6$ represents rolling 4 6-sided dice, and $d12$ represent 1 die with 12 sides. These terms can also be used in contexts such as "we are playing with d8s".
- Since a fundamental mechanic of *Dice Duels: Duel of the Eights* is that of re-rolling dice, an additional variable will be added to the notation to denote the number of rolls, $AdXrB$. For example, typical Yahtzee with 3 rolls would be denoted as $5d6r3$.
- Limits on dice parameters can be expressed as inequalities. For example:
 - $1 \geq A \geq A_{max}$ where A is the number of dice and A_{max} is the maximum allowable dice.
 - $1 \geq X \geq X_{max}$ where X is the number of dice sides and X_{max} is the maximum allowable dice sides, in values acceptable and modeled by the system.
 - $1 \geq B \geq B_{max}$ where B is the number of dice rolls and B_{max} is the maximum allowable dice rolls.

These limits can vary by game mode or player-defined settings.

- Dice typically take the forms of platonic solids, meaning where the dice faces are congruent regular polygons. The five such polyhedra are:
 - A tetrahedron has four faces ($d4$)
 - A cube has six faces ($d6$)
 - An octahedron has eight faces ($d8$)
 - A dodecahedron has twelve faces ($d12$)
 - An icosahedron has twenty faces ($d20$)

Other die shapes can be used such as a pentagonal trapezohedron with ten faces ($d10$) or even dice meant to be rolled lengthwise such as a triangular prism which despite having five total faces, is used as having three ($d3$), or has caps to prevent rolling an unintended face.

- Dice rolls indicate a value integer when rolled. These can be represented by the numeric value of the integer or by representing the integer value as dots, called pips.
- In typical Yahtzee some patterns for scoring have terms. These will have to be added to and abstracted for a game with a different number of dice, but for a $5d6r3$ these would be the scoring opportunities:
 - Rolling for aces, twos, threes, fours, fives, or sixes is rolling for as many of the number.
 - Chance is any combination of dice as a sum of all dice values.
 - A yahtzee is rolling all five of five dice with matching faces.
 - A three of a kind is having at least three of five dice match. For four of a kind, it is having at least four dice the same.
 - A straight is a set of sequential dice values. This can come in the small straight variant with four sequential values, or a large straight where all five dice are part of a sequence.
 - A full house is having two dice of a kind and three of another.
- A gameplay mechanic where players perform their turns concurrently with outcomes revealed simultaneously is referred to as a "simultaneous turn-based mechanism" in this document.
- This document makes reference to "game settings" and "game modes" where game settings are some specific settings that players will be able to customize, such as the number and type of dice, scoring methods, time limits, etc. and any permutation of these settings can define a "game mode".

4.1.2 Goal Statements

These primary goals should be achieved in the development of our system, providing criteria for completeness. We have additionally organized stretch goals for further development, but they are not to be a metric for system completeness.

The goal statements are:

GS1: Enjoyable game. The project is more than just a capstone, and we need the game to be an enjoyable experience.

- User feedback collection. Implement a simple feedback system such as rating or comments for use in interviews or surveys to gather player insight.
- Testing iteration. Conduct at least two rounds of user testing to identify and iteratively improve the system based on user experience.
- Quality assets. Ensure that graphics, animations, and sounds are of a quality to add positively to the overall enjoyment.

Measurement: Based on user feedback, a minimum of 75% consider the experience as enjoyable.

GS2: Online multiplayer functionality. We need to be able to connect two concurrent players to play the game together.

- Connection setup. Develop a server-client connection system where two players can connect.
- Game state synchronization. Each player's actions are to be reflected to both user outputs using real-time synchronization.
- Disconnection handling. Design a method to handle disconnections.

Measurement: Two players can connect such that both players can affect the game state and both players are notified of the updates.

GS3: Customizable game settings. Core game elements must be modifiable to create custom Yahtzee variants. As a goal we would need these options to be implemented:

- Dice quantity option. Create an interface element for users to select from at least three options for number of dice.
- Dice type option. Create an interface element for users to select from at least three options for type of dice. These different types of dice will have different number of faces.
- Scoring method option. Offer at least two scoring systems that can be used.
- Timer feature. Implement a timer that can be turned on or off, providing a countdown for turn time.

Measurement: The above options are implemented and are compatible with each other.

GS4: Preset game settings. By having some preset game configurations, it would allow players to more quickly learn the game or jump into an environment that has been tested.

- Create presets. Develop at least three preset configurations and test them for gameplay balance.
- Preset selection menu. Create a simple way to select a preset.
- Preset names and description. Give each available preset a name and a brief description to help players understand them.

Measurement: At least three preset game configurations would be available for players to load up and play.

GS5: 3D dice rolling. Rolling the dice will need to be or look to be three dimensional to recreate the tactile feel of the original game.

- Dice 3D models. Develop 3D dice models, or the appearance of such.
- Dice interaction. Allow for user interaction with rolling dice through clicks or drags.

Measurement: Dice will have the appearance of the preset die shape, and of being rolled, based on a minimum of 75% of user feedback considering it so.

Additional stretch goal statements are the following, and can additionally be considered for when looking to add to the system's complexity and to better fulfill the intended goal of being an enjoyable game.

The stretch goals are:

SG6: Local multiplayer. This would allow for players to play together on a single computer, but would require a different user interface and allow for different user interactions.

- Player input methods. Develop a system to handle inputs from multiple players on the same device.
- User interface adjustments. Design a UI (user interface) layout suitable for two players sharing a single screen.

Measurement: The ability for two players to play together using a single interface and game instance without an internet connection.

SG7: Singleplayer variants. A singleplayer game could be achieved either through a computer-run opponent in a game, or through a custom designed experience that could leverage the different environment.

- AI opponent development. Create an AI that can play against a human player.
- Difficulties. Design and allow players to adjust the AI opponent difficulty.
- Solo game modes. Design at least one unique solo challenge or mode that provides a self-contained experience.

Measurement: A single person can play at least one variant made specifically to be singleplayer without requiring a second human player to update the game state.

SG8: Online matchmaking. The game would provide users with the option to connect to another concurrent user based on a matchmaking score.

- Player rating system. Develop a rating system to categorize players by experience and skill.
- Matchmaking algorithm. Implement a system to match players.

Measurement: A player can connect to another unknown concurrent player who was selected as a compatible opponent.

SG9: Saving custom game setting. Having this ability would allow for a user who created a custom game variant to save them for the ability to replay it without the need to recreate those specific settings.

- Save and load system. Build a system for players to save custom setting to a file or local storage.
- Edit and delete options. Allow players to edit or delete saved custom settings for better management.

Measurement: A custom game variant, as per the "Customizable game settings" goal and "More game setting customization" stretch goals, can be saved locally and loaded up to be played.

SG10: More game setting customization. Besides the options in the goals section, some additional game customization options would include:

- Scorecard customization. Provide methods for players to adjust what options and hands appear on the scorecards.
- Scoring points options. Offer players the ability to modify the scores of scoring options.
- Additional scoring mechanisms. Include options for different methods of round or game scoring.

- Gambling mechanism. Add additional ways to act on probabilities such as wagering on specific rolls.
- Feedback-based features. Gather ideas through user feedback and testing to further expand available customization options.

Measurement: Additional game options outside the ones listed in the "Customizable game settings" goal would be available.

SG11: Dice customization. Dice could be made to appear differently, either as a means for personalization or for aiding with different impairments. An example could be a dice with pips versus a dice with a numbered faces.

- Dice colour options. Create at least three different dice appearances.
- Dice number representation. Create different ways to represent dice face values such as traditional pips, numbers, and symbols.
- Personalization menu. Implement a menu to allow for easy selection of dice appearance.

Measurement: At least five different dice appearance variants players can choose from, that would appear in the game.

SG12: Post game statistics. This could allow for players to analyze a game after completion in a more quantitative manner, aiding in better understanding statistical probabilities.

- Key statistic tracking. Track important game stats during play.
- Post-game summary screen. Present collected stats on a summary screen after a game.

Measurement: A post-game summary showing at least three key game stats, available after each game.

SG13: Multi-platform support. While most gaming experiences are for windows, this would allow for the game to be run on more than just the Windows operating system, allowing for a wider audience.

- Compile for systems. Compile the created system for other operating systems.
- Platform testing. Test the game on multiple operating systems.
- Cross-platform functionality. Verify the features such as online multiplayer work across different operating systems.

Measurement: The game can be run on operating systems other than Windows.

SG14: Dice highlighting. This would aid in determining what dice are used when scoring.

- Automatic dice highlighting. Implement a system to automatically highlight the dice that contribute to a player’s score.
- Optional setting. Allow players to enable or disable the feature.

Measurement: Dice used in scoring will be highlighted when appropriate.

4.2 Solution Characteristics Specification

This section, along with Physical System Description are not included within our document and have been removed from the template. The purpose of this section is to reduce the problem into one expressed in mathematical terms. Mathematical expertise is used to extract the essentials from the underlying physical description of the problem, and to collect and substantiate all physical data pertinent to the problem. Important elements that might otherwise have been in this section can be found in other sections where they may be more pertinent. Given the focus of this section on the physical description, and the fact there is no physical description as we have a software based project that is mostly hardware agnostic where the operating system would be more pertinent to the execution of the program.

5 Stakeholders

The stakeholders for this game project are those with vested interests in its development, release, and continued use. While there may be individuals with vested interest in *Dice Duels: Duel of the Eights*, they all fall within the broader categories below and generally act as a member of the whole rather than an individual stakeholder. Stakeholders for the capstone but not for the project itself are included in some sections below, but are not considered a stakeholder of the project due to not influencing the direction and development of the game outside of requiring a certain challenge level and documentation.

5.1 Traditional Yahtzee enthusiasts

- **Demographics:** Generally older players who have nostalgia for the classic Yahtzee game. They may have experience with the physical dice game playing with family and friends and seek a familiar experience to bring back memories and connect with people.
- **Technical Comfort Level:** May be lacking in experience with technology and be more comfortable with straightforward, user-friendly interfaces and are likely to appreciate clear instructions.
- **Motivations:**
 - *Nostalgia.* They seek to relive the classic Yahtzee experience in a more accessible format.
 - *Social connection.* They are interested in playing with friends they may not be able to meet in person.
 - *Relaxed gameplay.* They generally prefer low-stakes slower-paced gameplay that doesn't require complex strategies or fast decision making. They are more likely to appreciate a more relaxed experience.
- **Preferences:**
 - *Classic game mode.* They'll likely be drawn to a game mode that replicates the original Yahtzee experience.
 - *Simplicity over customization.* While they may be willing to try out some customization, they may prefer options that don't stray too far from the classic.
 - *Minimalistic design.* An interface that's easy on the eyes with intuitive navigation will help make the experience enjoyable for them.

5.2 Video Game enthusiasts

- **Demographics:** A diverse group spanning casual players to more experienced players focused on optimization, they are familiar with online multiplayer games and are comfortable with technology and video games.

- **Technical Comfort Level:** Most typical gamers are comfortable with technology, online multiplayer, and faster-paced games. They enjoy exploring game mechanics and may be open to more complex overlapping game mechanics. Are comfortable downloading a new video game with straightforward procedures.
- **Motivation:**
 - *Challenge and skill expression.* They're interested in games that allow for strategy and competitive play where skill and quick decision-making have an impact on the outcome.
 - *Engagement.* Video game enthusiasts seek a fast-paced experience with more action and reward cycles that provide instant feedback and keep them engaged.
 - *Replayability.* They generally enjoy experiences with unique challenges every time they play and can develop strategies over multiple runs.
- **Preferences:**
 - *Customization and variability.* They are likely to appreciate the ability to customize gameplay to explore possibilities and develop a game system that is challenging but enjoyable for them.
 - *Faster game modes.* They may enjoy options that speed up gameplay, such as round timers or different scoring methods that add intensity and variety.
 - *Reactive feedback.* Faster-paced action and instant feedback to keep them engaged would be preferred over slower gameplay.

5.3 Personas

- **Joan, 55 years old (The Nostalgic Yahtzee Player)** Joan enjoys simple board games that remind her of family gatherings. She values a straightforward interface, classic gameplay, and the option to play casually with friends over the internet when they cannot meet in person.
- **James, 24 years old (The Casual Gamer)** James plays games on the weekends with friends. He enjoys the flexibility of customizable game modes and fast-paced rounds. Alex values multiplayer gameplay with friends and occasional solo play.
- **Julian, 19 years old (The Competitive Gamer)** Julian enjoys strategic games that involve skill, competition, and learning over time. He prefers playing with friends that are similarly competitive, customization options, and post-game statistics to analyze his performance.

5.4 Priorities Assigned to Users

- **Primary Priority:** Casual and competitive video game enthusiasts would be the primary priority as they are the most plentiful and would be most likely to come across the game to play it. There is currently no game that provides customizable Yahtzee-like mechanics, and with feedback and testing, the game can be made to tailor to their preferences and be engaging.
- **Secondary Priority:** Traditional Yahtzee enthusiasts are a secondary priority in that using customization mechanics their preferred classic game can be recreated, but the game system overall would be more than just that. Their needs for a simple and intuitive user interface must be taken into consideration, as must options be available for their experience to likewise be enjoyable.
- **Tertiary Priority:** Outside of the context of stakeholders for the program, there are stakeholders for the capstone project. Their considerations do not shape the game itself as they are unlikely to play the game, but rather require a specific challenge level and documentation to be produced. As such, they can be considered tertiary despite not being stakeholders to the game unless also included in one of the above stakeholder groups.

5.5 User Participation

- **Development Phase:** Selected playtesters and the development team will regularly engage with the game throughout the development cycle to identify bugs and shape the user experience.
- **Testing Phase:** Both playtesters and target user groups will participate in beta testing to validate usability, customization options, and multiplayer functionality.
- **Feedback Phase:** In the feedback phase, when functionality is mostly validated, users will provide feedback through surveys and interviews to refine the game before a final release.

5.6 Ongoing Support

This project will be released with a final build that is complete and of quality. Since this game will also be released under a permissive license, the codebase will be publicly available on GitHub, allowing others to further develop it and add features they themselves may wish to have available, or fix any bugs that may be present. The current developers may also be available in their free time if this project or a spin-off is something that will continue past the capstone project timeframe.

6 Requirements

6.1 Functional Requirements

- R1 The game shall support an online player vs player mode, where 2 players can play against each other.

Rationale: Player vs player support is a central functionality for this game and one that enhances the overall enjoyment of the game by fostering a more social game dynamic.

Fit Criterion: 2 players can connect to each other over an internet connection.

- R2 The game shall handle score calculations using similar rules to standard Yahtzee under default game settings.

Rationale: Score calculations that follow from a well established, and well balanced game will help users jump into the game faster and foster a more fair game experience.

Fit Criterion: Scores in default Yahtzee must match those obtained from an official Yahtzee score calculator (depending on house rules).

- R3 The game shall simulate realistic physics for 3D dice rolls with pseudo-randomness on every roll that replicates accurately to all users. If rolling A dice, A numbers will have to be determined from the range $1 \leq A_i \leq X$.

Rationale: It is important that players get to visually see the dice roll and feel that the outcome of the roll is not clearly predetermined in order to enhance the user experience and maintain continuity between the game and real life.

Fit Criterion: The result of a roll must be pseudo-random, adhere to physics-based simulation, and be accurately displayed to all players.

- R4 The game shall use the real outcome of a roll to get the values from the dice.

Rationale: In order to get true randomness off a simulated dice roll, it makes the most sense to actually just simulate the roll and read the result rather than precalculating the results and forcing the roll to match the output.

Fit Criterion: Dice values displayed on the UI must match the final resting state of the 3D dice model after rolling.

- R5 The game shall support both regular six-sided dice and a set of dice with different number of sides, such as octahedral dice.

Rationale: By providing support for more dice types, we can increase the level of complexity of the game for users that are looking for a more interesting variation of the game.

Fit Criterion: Players can select dice with different numbers of sides, and the game correctly processes rolls for all supported dice types.

- R6 The game shall implement a simultaneous turn based mechanism such that each player (or computer) takes a turn at the same time and then results are revealed simultaneously to each other at each dice roll.

Rationale: This allows users to play together synchronously, while preserving the secrecy of each dice roll so that no player gets an advantage by knowing whether their opponent has rolled well or not.

Fit Criterion: Each player takes actions simultaneously, and results are revealed to both players once each has selected an action.

- R7 The game shall allow players to pick which dice they would like to use for each roll, and which dice they would like to omit. Selected dice will be a subset of the current dice in play belonging to that player $D' \subseteq D$.

Rationale: This is an important part of standard game rules, and also increases the range of strategic decision making that players get to have.

Fit Criterion: Players can select a subset of their dice before rolling, and the game correctly rolls only the selected dice.

- R8 The game shall display some sort of user interface to display scores, number of rolls, time limits, state of dice, and player names.

Rationale: Some form of user interface is necessary to convey important game information to the player.

Fit Criterion: The UI must display the score, roll count, time limits, dice states, and player names without omissions or inconsistencies.

- R9 The game shall provide controls for the player to modify the game settings to access unique variants of the game.

Rationale: A core part of our team's vision for this game is that players should be able to customize their playing experience by altering things like the number of dice, type of dice used, and time settings.

Fit Criterion: Players must be able to access and modify at least 3 game settings before a game is started.

- R10 The game shall provide presets for different game modes.

Rationale: This should help new players get used to the game before exploring more unique setting configurations.

Fit Criterion: Players must be able to select from at least one preset game mode in the settings menu.

***** The following requirements relate to our stretch goals *****

- R11 The game shall support a local player vs player mode, where 2 players can play against each other on the same computer.

Rationale: Given that this is a turn based game it is perfectly possible for this game to be setup with 2 player functionality on one device. This simply increases the number of ways in which the game can be played.

Fit Criterion: Two players can take turns using the same computer without errors or UI conflicts.

- R12 The game shall support a player vs computer mode, where 1 player can play against another entity without needing to find a human match.

Rationale: Player vs computer support is important for instances where a user may not be able to find a human match to play against.

Fit Criterion: A player can start a game against a computer opponent without needing a human opponent.

- R13 The game shall implement some sort of algorithmic computer opponent for player vs computer gameplay option.

Rationale: For player vs computer to function, some sort of algorithmic opponent will be necessary to implement the computer player or else there will be no challenge for the human player.

Fit Criterion: The computer opponent makes legal and strategic moves, ensuring a playable experience.

- R14 The game shall implement online matchmaking.

Rationale: For players that want to play the game against a real opponent but don't know anyone who is available to play against them, this is a great way for that player to find someone to play against.

Fit Criterion: Players can search for and be matched with online opponents.

- R15 The game shall provide the option to save specific game settings to be reused in future sessions.

Rationale: For frequent players of the game, they may have a preferred custom variation of the game settings that they may wish to save rather than having to input the game settings on every new game.

Fit Criterion: Players can save and load custom game settings across different play sessions.

- R16 The game shall display round statistics to each player at the end of every round.

Rationale: Players will likely want to see some statistics at the end of the round to quantify how well they played.

Fit Criterion: At the end of a round, the game displays statistics such as total score, average roll, and average score per round.

**** *The following requirements relate to our hazard analysis* ****

R17 The game shall always show the correct current state accurately.

Rationale: The player needs to be able to understand the current state of the game to understand their current standings, and strategic the next move. Similarly, the current state indicates what is the next action to be done by the user. This is extended to states outside of an active game such as game settings selection, match-up, and system settings selection.

Fit Criterion: The game state (e.g., current scores, dice states, turn order) is always correctly updated and visible to the player.

6.2 Non-Functional Requirements

NFR1 **Performance** The game shall maintain a frame rate of at least 30 FPS at all times.

Rationale: Players need the frame rate of the game to be high enough at all times to see what's going on.

Fit Criterion: The game consistently runs at 30 FPS or higher on supported hardware.

NFR2 **Usability** The game shall implement a clear and easy to use/understand user interface.

Rationale: Users should not struggle to figure out where controls are or how certain features work. This would create a barrier to entry for new players, and it would be best if players could pick up this game and its controls with little difficulty or time required.

Fit Criterion: Players can navigate the UI and access game functions without external instructions or confusion.

NFR3 **Portability** The game shall be supported on systems running Windows 10 or later.

Rationale: Most PC users use Windows 10 or 11, so ensuring that the game ports well to these operating systems is an important requirement if we want to reach a large demographic of users.

Fit Criterion: The game successfully installs and runs on Windows 10 and later without compatibility issues.

NFR4 **Reliability** Multiplayer games of Yahtzee should crash less than 1% of the time.

Rationale: Players need to feel that the game is reliable and should not need to be concerned about the game crashing in the middle of a round.

Fit Criterion: Multiplayer sessions have a crash rate of less than 1% across 100 test runs.

NFR5 **Responsiveness** The game shall respond to inputs from the user within 500 milliseconds in the worst case.

Rationale: This figure represents a minimum acceptable response. If a user needs to wait longer than this to see the result of their input, they may become frustrated and even try spamming the control.

Fit Criterion: User input results in a visible response within 500ms in at least 99% of interactions.

NFR6 **Modularity** The game's codebase shall be modular, such that it is easily extendable and reusable, and allows for quick fixes to bugs that may occur.

Rationale: Our team has minimum goals in mind for what this game must be upon release, but to be able to continue updating the game to meet stretch goals, our codebase should be designed with modularity in mind to streamline the development process.

Fit Criterion: Developers can add new game modes and dice types without altering existing unrelated code.

NFR7 **Efficiency** The game shall be optimized to run on lower end systems which might have minimal CPU and GPU resources.

Rationale: The game that our team is developing is not such a graphically or computationally intensive game that it should require high-end PCs to run. Additionally, being able to run the game on lower-end systems will allow our game to reach a broader audience.

Fit Criterion: The game runs smoothly on systems meeting the minimum system requirements, with no major performance drops.

NFR8 **Enjoyability** The game shall be found enjoyable by at least 75% of users.

Rationale: The game should appeal to the majority of its users, because if it doesn't then the game serves no particular purpose.

Fit Criterion: At least 75% of users, as measured by post-game surveys or analytics, should find the game enjoyable based on scoring and qualitative feedback.

NFR9 **Appearance** The game shall maintain a consistent UI style and 3D visual style throughout all in game views.

Rationale: A style that evokes some sense of continuity throughout the game is important to help users get used to the layout faster and to give the appearance of a more professionally developed gaming experience.

Fit Criterion: A usability test with at least 10 users should confirm no major inconsistencies with UI elements and 3D assets adhering to a predefined style guide.

***** The following requirements relate to our stretch goals *****

NFR10 **Portability** The game shall be supported on MacOS devices.

Rationale: There are many users who use MacOS devices and would be a good target audience for this game.

Fit Criterion: The game must run on MacOS devices without major performance issues or compatibility errors, as verified through testing on at least two different MacOS versions.

7 Requirements Implementation Roadmap (Phase In Plan)

The implementation roadmap outlines the priority and timeline for completing each functional and non-functional requirement for the game. Requirements are categorized into four main priority sections to ensure a clear implementation plan throughout development.

7.1 Critical Priority

These functional and non-functional requirements are necessary for every stable build generated and are required through all stages of development, hence have no specific Phase In plan date.

- **R2:** The game must allow players to roll up to five dice simultaneously.
- **R3:** Players must be able to re-roll selected dice up to two times per turn.
- **R4:** The game must implement scoring categories, such as "Three of a Kind," "Four of a Kind," and "Full House."
- **R5:** The game must support calculating and displaying the current score for each player.
- **R8:** The game must provide a user interface for players to select scoring categories and view scores.
- **R17:** The game must support networked multiplayer functionality for PvP matches.

7.2 High Priority

These requirements are required for all stable builds but are flexible based on the specific game variant implementation, hence have no specific Phase In plan date.

- **R1:** The game must provide a digital representation of a Yahtzee scorecard.
- **R6:** Players must be able to end their turn, and scores must be locked in accordingly.
- **R7:** The game must display the current round number and the total number of rounds.
- **R9:** The game must include audio cues for rolling dice and scoring actions.
- **R10:** The game must include visual effects to enhance the player experience, such as dice animations.

7.3 Non-Functional Requirements Priority

These non-functional requirements (NFRs) are Medium Priority and will be improved upon during the course of development, starting with lower standards of NFR satisfaction, with the goal of satisfying each NFR completely by Final Demo.

- **NFR1:** The game should have an average response time of less than 100 ms for player interactions.
- **NFR2:** The game should provide a consistent frame rate of at least 30 FPS.
- **NFR3:** The game should support a variety of screen resolutions and aspect ratios.
- **NFR4:** The game must handle network disconnections gracefully, allowing players to reconnect.
- **NFR5:** The game should provide accessibility features, such as colorblind-friendly UI options.
- **NFR6:** The game should support cross-platform play between different operating systems.
- **NFR7:** The game should ensure data privacy and protection, including encryption of sensitive player data.
- **NFR8:** The game should include an in-game tutorial to guide new players.
- **NFR9:** The game should have logging and diagnostics features for debugging purposes.

7.4 Stretch Goals

Lower Priority, and will be tackled in order of individual priority, in the given order, based on scope during or at the end of the project, or post capstone.

- **R15:** The game should include leaderboards to track high scores across all players.
- **R16:** The game should allow players to customize their dice and scorecard appearance.
- **R14:** The game should support in-game chat for players during multiplayer matches.
- **R11:** The game should include soundtracks that can be toggled by players.
- **NFR10:** The game should support integration with third-party social media for sharing scores.
- **R12:** The game should include a feature to save and resume matches.
- **R13:** The game should support various game modes, such as "Classic Yahtzee" and "Custom Rules."

8 Standards, Codes, Legal, and Regulatory Factors

Developing an online multiplayer game requires adherence to various standards, codes, legal, and regulatory factors to ensure a safe, fair, and compliant gaming experience. The following standards are identified for the multiplayer aspect of the game, especially in the context of player-versus-player (PvP) matches:

8.1 Data Privacy and Protection

- **GDPR (General Data Protection Regulation):** Given that the game may involve players from the European Union, compliance with GDPR is essential. This includes ensuring user consent for data collection, providing clear privacy policies, and safeguarding player data.
- **CCPA (California Consumer Privacy Act):** If players are based in California, the game must comply with CCPA to give users control over their personal information.
- **COPPA (Children’s Online Privacy Protection Act):** If the game targets children under 13, compliance with COPPA is mandatory to protect children’s privacy.

8.2 Network and Online Standards

- **IEEE 802.11 (Wi-Fi Standards):** Ensuring reliable local area network (LAN) connectivity follows IEEE 802.11 standards to provide consistent communication quality.
- **WebSockets (RFC 6455):** The use of WebSockets for real-time communication between players must comply with RFC 6455 to ensure reliable, secure data exchange.

8.3 Game Fairness and Anti-Cheating Measures

- **ISO/IEC 27001 (Information Security Management):** Implementing practices from ISO/IEC 27001 helps in managing game security, reducing the risk of cheating, and protecting the game from malicious activities.
- **Fair Play Guidelines:** To ensure fair play in PvP matches, the game should adopt fair matchmaking and anti-cheating measures, such as monitoring unusual behavior and implementing server-side verification of actions.

8.4 Legal Considerations for Online Play

- **Terms of Service and End-User License Agreement (EULA):** A clear EULA must be provided, detailing acceptable behavior, limitations of liability, and consequences for misuse or cheating.

- **Content Rating Standards:** The game must be rated appropriately, such as following the **ESRB (Entertainment Software Rating Board)** or **PEGI (Pan European Game Information)** standards, to ensure suitability for the intended audience.

8.5 Accessibility Standards

- **WCAG (Web Content Accessibility Guidelines) 2.1:** For an inclusive experience, the game's online features and user interface should follow WCAG 2.1 guidelines, ensuring accessibility for players with disabilities.

8.6 Intellectual Property and Copyright

- **Copyright Compliance:** The game must avoid unauthorized use of copyrighted material, including sound effects, graphics, and other assets. Original content must be used, or appropriate licenses must be obtained.
- **Trademark Considerations:** Any use of trademarks must be properly authorized to avoid infringement issues.

8.7 Regulatory Requirements for Online Play

- **Online Gambling Regulations:** If any aspect of the game involves virtual currency or random rewards, it may fall under online gambling regulations in certain jurisdictions. The game must be reviewed for compliance with applicable laws to avoid any unintended legal issues.
- **Consumer Protection Laws:** Compliance with consumer protection laws is crucial, ensuring transparency in in-game purchases and providing users with refund policies.

By adhering to these standards, codes, and legal factors, the development of the online multiplayer game will ensure a fair, compliant, and enjoyable experience for all players, while minimizing legal and regulatory risks.

9 Standards Compliance Roadmap

The following compliance roadmap outlines when each of the identified standards will be adhered to during the development of the online multiplayer game:

- **Data Privacy and Protection (GDPR, CCPA, COPPA):** Compliance will be met by the Final Demo.
- **Network and Online Standards (IEEE 802.11):** Compliance will be met by the Final Demo.
- **Network and Online Standards (WebSockets):** Compliance will be addressed post-capstone based on the game's scale.
- **Game Fairness and Anti-Cheating Measures (ISO/IEC 27001):** Compliance will be addressed post-capstone based on the game's scale.
- **Game Fairness and Anti-Cheating Measures (Fair Play Guidelines):** Compliance will be met by the Final Demo.
- **Legal Considerations for Online Play (Terms of Service, EULA, Content Rating Standards):** Compliance will be met by the Final Demo.
- **Accessibility Standards (WCAG 2.1):** Compliance will be addressed post-capstone based on the game's scale.
- **Intellectual Property and Copyright (Copyright Compliance, Trademark Considerations):** Compliance will be continuously verified, with final verification by the Final Demo.
- **Regulatory Requirements for Online Play (Online Gambling Regulations):** Compliance may be required post-capstone based on the direction of the game and will be evaluated accordingly.
- **Regulatory Requirements for Online Play (Consumer Protection Laws):** Compliance will be met by the Final Demo if the game direction leads to the introduction of in-game purchases.

10 Likely Changes

LC1: Adding more ways to find opponents to play with (single player, local multiplayer on same device, etc)

- After setting up our initial on-line multiplayer and making sure that the online multiplayer is stable, we can expand our projects scope by adding other ways that users can find opponents to play against, such as a singleplayer gamemode against an algorithmic opponent, or a local multiplayer mode that can be played on the same device.
- This relates to the following requirements:
 - * [GS2: Online multiplayer functionality](#)
 - * [SG6: Local multiplayer](#)
 - * [SG7: Singleplayer variants](#)
 - * [SG8: Online matchmaking](#)

LC2: the amount and variability of dice options may increase and decrease based on the scope of the project

- As the Scope of our project increase and decreases, we will consider experimenting with different amounts and types of dice.
- This relates to the following requirements:
 - * [GS3: Customizable game settings](#)

LC3: We may expand comparability to other operating systems depending of the scope of the project (Mac OS, etc).

- While we will initially design our system to work on Windows 10/11 devices, should the scope of our project expand, we are likely to adapt our system to work on other OS systems too such as Mac OS, or Linux, allowing our product to reach more players.
- This relates to the following requirements:
 - * [SG13: Multi-platform support](#)

LC4: We may change how the scores are calculated.

- As we change and add various options for the players, such as changing the number and types of the dice, we may need to account for the different probabilities that these new options may create, which in turn would require us to modify our scoring system to account for these radically different probabilities.

- This relates to the following requirements:
 - * [GS3: Customizable game settings](#)
 - * [SG10: More game setting customization](#)

LC5: We may expand the amount of players that can play the same game from just the initial 2 depending on scope.

- While initialized for two players, it is known that Yatzee can easily support more players, given that the fundamental mechanics of the game don't change when you add them. Thus, when we stabilize our two-player multiplayer, we could expand upon it to allow more than the initial two players to play the same game, allowing more users to enjoy our product.
- This relates to the following requirements:
 - * [SG10: More game setting customization](#)

11 Unlikely Changes

UC1: We will not remove the multi-player component of the game.

- The game is inherently a social game that involves chance and strategy in an attempt to get the highest score compared to other players, As such multiplayer is a core component of the game and it must be included to ensure that an important aspect of the game isn't lost.
- This relates to the following requirements:
 - * [GS1: Enjoyable game](#)
 - * [GS2: Online multiplayer functionality](#)
 - * [SG6: Local multiplayer](#)

UC2: We will not remove the dice and it's probabilities

- Dice are a critical component of Yahtzee and it's variations, and the probabilities that the dices rolls provide are a core component in the way that score is calculated in game. Thus, our game will always involve the usage of dice and the calculation of the probabilities involving them.
- This relates to the following requirements:
 - * [GS3: Customizable game settings](#)
 - * [GS5: 3D dice rolling](#)

UC3: We will not switch from our engine Godot for the duration of the project.

- After conducting research on other game engines, and comparing our options, we have concluded that the Godot engine is the game engine best suited for our project and that would help us best achieve all our requirements but especially related to the following requirements:
 - * [SG13: Multi-platform support](#)

UC4: We will always have customization between game variants, and not just presets.

- One of the selling points for this project is to have customised settings for our game, allowing the user to tailor their experience to the way they want it. Thus, it is unlikely we will alter our plans to include this feature.
- This relates to the following requirements:
 - * [GS4: Preset game settings](#)
 - * [GS3: Customizable game settings](#)
 - * [SG10: More game setting customization](#)

- * [SG9: Saving custom game setting](#)

UC5: We will not change the 3D format of our game to 2D

- When playing a physical game like Yahtzee, one of the most engaging aspects is the action of rolling the dice. We wish to recreate the feel of playing the physical game as closely as possible by allowing our player to be able to visually see the dice roll in a way that mirrors the physical experience.
- This relates to the following requirements:
 - * [GS5: 3D dice rolling](#)

12 Traceability Matrices and Graphs

This section has been adapted from the original "Traceability Matrices and Graphs" portion of this document, which was initially geared toward scientific computation and the dependencies between physical components. In this revised section, we focus on aligning the specific goals of the game with the corresponding requirements, design changes, and other relevant elements outlined in this document. By doing so, we aim to create a clear and structured mapping between our Goals and the necessary steps for implementation.

The traceability matrices in this section provide an overview of these connections. Where an 'X' appears in the table, it signifies that a requirement or other item was developed in response to a particular goal or is linked to that goal. Two separate tables are presented: the first traces the core Goals of the project, while the second maps the Stretch Goals, allowing for a more comprehensive understanding of how different project objectives are supported through specific requirements.

Summary of goals and stretch goals:

- [GS1: Enjoyable game](#)
- [GS2: Online multiplayer functionality](#)
- [GS3: Customizable game settings](#)
- [GS4: Preset game settings](#)
- [GS5: 3D dice rolling](#)
- [SG6: Local multiplayer](#)
- [SG7: Singleplayer variants](#)
- [SG8: Online matchmaking](#)
- [SG9: Saving custom game setting](#)
- [SG10: More game setting customization](#)
- [SG11: Visual Dice customization](#)
- [SG12: Post game statistics](#)
- [SG13: Multi-platform support](#)
- [SG14: Dice highlighting](#)

| Reference Item | GS1 | GS2 | GS3 | GS4 | GS5 |
|----------------|-----|-----|-----|-----|-----|
| R1 | | X | | | |
| R2 | | | | X | |
| R3 | | | | | X |
| R4 | | | | | X |
| R5 | | | X | | |
| R6 | | X | | | |
| R7 | X | | | | |
| R8 | X | | | | X |
| R9 | | | X | | |
| R10 | | | | X | |
| R17 | X | | | | X |
| NFR1 | X | | | | |
| NFR2 | X | | | | |
| NFR3 | | | | | |
| NFR4 | X | | | | |
| NFR5 | X | | | | |
| NFR6 | | | | | |
| NFR7 | X | | | | |
| NFR8 | X | | | | |
| NFR9 | X | | | | X |
| LC1 | | X | | | |
| LC2 | | | X | | |
| LC4 | | | X | | |
| UC1 | X | X | | | |
| UC2 | | | X | | X |
| UC4 | | | X | X | |
| UC5 | | | | | X |

Table 2: Traceability Matrix for Goals

Legend: Each 'X' in this table indicates a direct relationship where the requirement or change in the row supports the goal in the column.

| Reference Item | SG6 | SG7 | SG8 | SG9 | SG10 | SG11 | SG12 | SG13 | SG14 |
|----------------|-----|-----|-----|-----|------|------|------|------|------|
| R11 | X | | | | | | | | |
| R12 | | X | | | | | | | |
| R13 | | X | | | | | | | |
| R14 | | | X | | | | | | |
| R15 | | | | X | | | | | |
| R16 | | | | | | | X | | |
| R17 | | | | | | X | X | | X |
| NFR10 | | | | | | | | X | |
| LC1 | X | X | X | | | | | | |
| LC3 | | | | | | | | X | |
| LC4 | | | | | X | | | | |
| LC5 | | | | | X | | | | |
| UC1 | X | | | | | | | | |
| UC3 | | | | | | | | X | |
| UC4 | | | | X | X | | | | |

Table 3: Traceability Matrix for Stretch Goals

Legend: Each 'X' in this table indicates a direct relationship where the requirement or change in the row supports the stretch goal in the column.

13 Development Plan

Please refer to separate development plan located in docs: [link](#)

14 Values of Auxiliary Constants

This section, titled "Auxiliary Constants," has been intentionally omitted as it is more appropriate for documents that require the definition and organization of numerous constants and parameters. In the context of our project, which focuses on the game variants, the need for such a section is minimal.

The constants and parameters relevant to the game family are already comprehensively addressed in other parts of this document, particularly within the sections dealing with Commonalities, Variabilities, and Parameters of Variations. These sections provide all necessary explanations and definitions in a self-contained manner. Given that these parameters are specific to certain features of the game and are clearly outlined where applicable, adding an additional section on auxiliary constants would be redundant and not add any meaningful value to the document. Therefore, we have chosen to skip this section for the sake of clarity and to avoid unnecessary duplication.

15 Commonalities

This section outlines the shared elements across the different games in the product family, including core mechanics, UI components, and gameplay goals.

15.1 Dice Rolls

A fundamental aspect of the game is its reliance on dice rolls. All variants will incorporate dice rolling as a primary mechanic, with corresponding UI elements to visually represent the action, ensuring consistency across game versions.

15.2 Yahtzee Hands

The scoring system in each game variant will be based on Yahtzee-style hands. While the specific scoring mechanics and available hands may vary slightly, the core objective of rolling a hand that aligns with predefined scoring categories remains consistent throughout all versions.

15.3 Score Sheet Structure

Similar to the Yahtzee Hands mechanic, the structure of the score sheet will be uniform across all game variants. This provides players with a familiar interface and ensures consistency in tracking scores.

15.4 Simultaneous Play

The game is designed for real-time play rather than a turn-based system. Players will always be engaged in the same stage of rolling, preventing any strategic advantage in head-to-head game variants.

15.5 2-Player or Single-Player Only

None of the game variants will support more than two human players. The focus will remain on single-player modes or head-to-head matchups with only two participants.

15.6 Always Playing Against an Opponent

Players will never play in isolation to achieve the highest possible score. Instead, each game will involve competing either against another player or against a predetermined computer score criterion, reinforcing the competitive nature of the gameplay.

16 Variabilities

This section details the variabilities between the different games in the product family, including changes in UI elements, game goals, and core mechanics.

16.1 Number of Dice

The number of dice used in the game can vary between different versions. Some variants may increase or decrease the number of dice to adjust the complexity and strategy of the game.

16.2 Sides on Dice

The number of sides on each dice can differ across game variants. This allows for customization of gameplay, where different variants may feature dice with more or fewer sides, influencing the probability and outcomes of each roll.

16.3 Individual Sides

The specific values or symbols on the sides of each dice can vary between versions. This variability provides an opportunity to introduce different themes or scoring dynamics depending on the game's rules.

16.4 Scoring Calculation

How points are calculated for different hands and the influence of dice rolls on the final score can be customized in different game variants. This allows for changes in how specific hands are valued and how scoring impacts the overall gameplay strategy.

16.5 Time Per Turn

The amount of time a player has to roll the dice and make their selection can be adjusted between variants. Time limits can vary, creating different levels of pressure and pacing in the game.

16.6 Hand Restrictions

Some game variants may restrict certain hands from being scored or may allow hands to be scored multiple times. These variations provide flexibility in game strategy and scoring dynamics.

17 Parameters of Variations

This section details the specific values that the variabilities between the different games in the product family can take, corresponding to the variabilities outlined in the previous section.

17.1 Number of Dice

The number of dice used in the game can range from 1 to a maximum based on testing results. Having a high maximum would allow for more customization options, but a reasonable limit based on what is realistically usable and fun but also based on physics simulation performance needs to be set. This allows for variations in gameplay complexity, depending on the specific variant.

17.2 Sides on Dice

The usable dice will be a subset of dice modeled with the following number of sides: 4, 6, 8, 10, 12. This flexibility in dice sides introduces variability in the probability and potential outcomes for each roll.

17.3 Individual Sides

The values displayed on the individual sides of each die can range from 1 to the respective number of sides on that die and can be displayed in different representations. This variability allows for different numerical or symbolic configurations to match specific game variants.

17.4 Scoring Calculation

Each hand can be assigned any integer score value or modified by factors based on the dice rolls. This provides flexibility in defining how each hand is valued in the different game variants.

17.5 Time Per Turn

The time allotted for each turn can range from 5 seconds to 2 minutes, or be set to unlimited. This parameter affects the pacing of the game, allowing for both fast-paced and more thoughtful gameplay.

17.6 Hand Restrictions

Hands can be restricted by removing up to all but one from play. Additionally, hands can be allowed to be repeated between 1 to 10 times or an unlimited number of times, giving options for different scoring strategies.

References

- Author Author. System requirements specification. <https://github.com/...>, 2019.
- Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.
- Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995. URL <http://citeseer.ist.psu.edu/428727.html>.
- David L. Parnas. On the criteria to be used in decomposing systems into modules. *Comm. ACM*, 15(2):1053–1058, December 1972.
- David L. Parnas. Designing software for ease of extension and contraction. In *ICSE '78: Proceedings of the 3rd international conference on Software engineering*, pages 264–277, Piscataway, NJ, USA, 1978. IEEE Press. ISBN none.
- David L. Parnas and P.C. Clements. A rational design process: How and why to fake it. *IEEE Transactions on Software Engineering*, 12(2):251–257, February 1986.
- D.L. Parnas, P.C. Clement, and D. M. Weiss. The modular structure of complex systems. In *International Conference on Software Engineering*, pages 408–419, 1984.
- James Robertson and Suzanne Robertson. *Volere Requirements Specification Template*. Atlantic Systems Guild Limited, 16 edition, 2012.
- W. Spencer Smith. Systematic development of requirements documentation for general purpose scientific computing software. In *Proceedings of the 14th IEEE International Requirements Engineering Conference, RE 2006*, pages 209–218, Minneapolis / St. Paul, Minnesota, 2006. URL <http://www.ifi.unizh.ch/req/events/RE06/>.
- W. Spencer Smith and Nirmitha Koothoor. A document-driven method for certifying scientific computing software for use in nuclear safety analysis. *Nuclear Engineering and Technology*, 48(2):404–418, April 2016. ISSN 1738-5733. doi: <http://dx.doi.org/10.1016/j.net.2015.11.008>. URL <http://www.sciencedirect.com/science/article/pii/S1738573315002582>.
- W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Ågerfalk, and N. Kraiem, editors, *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*, pages 107–121, Paris, France, 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.
- W. Spencer Smith, Lei Lai, and Ridha Khedri. Requirements analysis for engineering computation: A systematic approach for improving software reliability. *Reliable Computing, Special Issue on Reliable Engineering Computation*, 13(1):83–107, February 2007.

- W. Spencer Smith, John McCutchan, and Jacques Carette. Commonality analysis of families of physical models for use in scientific computing. In *Proceedings of the First International Workshop on Software Engineering for Computational Science and Engineering (SECSE 2008)*, Leipzig, Germany, May 2008. In conjunction with the 30th International Conference on Software Engineering (ICSE). URL <http://www.cse.msstate.edu/~SECSE08/schedule.htm>. 8 pp.
- W. Spencer Smith, John McCutchan, and Jacques Carette. Commonality analysis for a family of material models. Technical Report CAS-17-01-SS, McMaster University, Department of Computing and Software, 2017.

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning.

1. What went well while writing this deliverable?
 - The SRS document had a lot of work to be done in a lot of sections. In an effort to mitigate merge conflicts and better divide the workload, a main SRS.tex file was created which would reference and include other .tex files which contain the information for the sections. - John P.
 - All team members were able to contribute to the document. Sections were divided up well so that each team member could focus on their own specific part. - Isaac G.
 - Team members communicated often and effectively. - Isaac G.
 - Code changes were handled well as described in our workflow plan such that we were able to keep track of changes and avoid merge conflicts. - Isaac G.
2. What pain points did you experience during this deliverable, and how did you resolve them?
 - Since our project is solely a software without any hardware components or dependencies, the template was a little difficult to navigate. While we used the default SRS template as guidance, we made use of inspiration from the Volere SRS template for sections that would be more relevant to our own and removed sections that were not relevant from the default SRS template. - John P.
 - Some sections for this SRS were not entirely clear and some sections were not very applicable to our specific project. We were however able to meet with our TA to iron out some of these issues and find a clear path forward. - Isaac G.
 - Our project was able to benefit from a commonalities section but this is not something that any of us knew how to do. But through some discussion with the prof and some research, Nigel was then able to look into this issue and ultimately did a great job building out the commonalities section for the team. - Isaac G.
 - With many busy schedules finding time to meet and work was a challenge. We were however able to compare schedules and find meeting times that we could all make, and everyone communicated well to let team members know when they would have time to get to their part(s) of the deliverable done. - Isaac G.
3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?

- Our primary stakeholders are those who would be interested in playing the game, so indirectly we are members of the stakeholder group as well as our supervisor. As such, through consensus and group discussion, game mechanics that would be interesting and enjoyable were drawn up which determine most of our functional requirements. - John P.
 - Our look and feel requirements are all inspired by speaking with potential users. - Isaac G.
 - Our functional requirement for being able to customize the dimensionality of our dice is inspired by our supervisor who is a stakeholder. - Isaac G.
4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.
- For the calculations of probabilities, SFWRENG 4E03 proved to be helpful and discrete time markov chains were used to determine probabilities in rolls and how many of a kind. - John P.
 - In thinking how we are to organize the program in terms of architecture styles, SFWRENG 3A04 is helpful in providing examples and information on different architectures and their applications. - John P.
 - Since our project is a game, there will be constant player input and output and interaction through an interface. As such, SFWRENG 4HC3 provided helpful information on the design of user interfaces and principles of good interface design. - John P.
 - All of our software design courses will be crucial for the initial planning phases of this project. And these courses were key to the development of our skillsets relating to gitflow and large project setups. - Isaac G.
 - Our software testing and requirements course is proving to be useful now and will likely continue to be useful as the project continues. - Isaac G.
 - Our object oriented programming course will be useful when coding the project. - Isaac G.
 - Our engineering project courses (1P13, 2PX3, 3PX3) are also helpful for many of the planning schemes that we use and for the familiarity that they have provided us with group work. - Isaac G.
5. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

- I will need to learn to develop in Godot, as that is the game engine we will be using, as well as learning C# as a language. I have also already learnt more GitHub that before with branched, merging, issues, etc. Finally, I hope to further explore probability throughout this project and more advanced calculations such as DTMC. - John P.
 - All team members will need to develop a strong working understanding of the Godot game engine and associated coding language. - Isaac G.
 - 2D design skills should be developed by at least 1 team member to be able to create visually appealing user interfaces and graphics. - Isaac G.
 - 3D design skills should be developed by at least 1 team member to be able to create visually appealing 3D assets for the game. - Isaac G.
6. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?
- In terms of learning Godot, there are many online tutorials that can be found and followed along, and all of us should be making use of this resource before we contribute to the project to become accustomed to the process. A second way to acquire experience in Godot is through making a small project on our own once we have followed the online tutorials. This would also be the case for C# and the scripting that is available in Godot. - John P.
 - To learn about the Godot game engine and its coding language, team members can read the documentation, watch tutorials, and follow along in making sample Godot projects. This will be the primary method of learning for all team members. - Isaac G.
 - I(Isaac) will focus on Godot because I have game development experience and feel very comfortable with coding. I also do not feel that artistically inclined which is most beneficial for 2D and 3D design. - Isaac G.
 - To learn about 2D design, team members can watch tutorials and practice using software like Photoshop, GIMP, or PAINT.NET. - Isaac G.
 - To learn about 3D design, team members can watch tutorials and practice using software like Blender. - Isaac G.