Hazard Analysis
SFWRENG 4G06

Team 9, dice_devs
John Popovici
Nigel Moses
Naishan Guo
Hemraj Bhatt
Isaac Giles

Table 1: Revision History

Date	**Developer(s)**	**Change**
2024-10-23	Hemraj Bhatt	Added content to sections 1, 2, and 3
2024-10-23	Isaac Giles	Added content to sections 5 and 6
2024-10-25	Isaac Giles	Updated reflection section
2024-10-25	John Popovici	Fixed LaTeX compiling error and table format
2024-10-25	Hemraj Bhatt	Added content to section 4
2024-10-25	Hemraj Bhatt	Updated reflection section
2024-10-25	Nigel Moses	Added content to Safety Requirements and Roadmap
2024-10-25	John Popovici	Updated formatting; Added to reflection
2024-10-25	Naishan Guo	Added to reflection
2024-10-28	John Popovici	Fixed FMEA labels and added de-sync, as per peer review
2024-11-05	John Popovici	Added FMEA table value/rating explanations section 6.2
2024-11-22	John Popovici	Added List of Tables and fixed table 2
2024-11-22	John Popovici	Expanded upon section 5 and 6 based on feedback
2024-11-22	John Popovici	Improved FMEA Table traceability
...	...	...

Contents

List of Tables

1 Introduction

Game system design is often perceived as straightforward because users interact primarily with the front end, unaware of the complexities that lie behind the scenes in the back end. In reality, it consists of multiple different components working together in order to create a seamless experience for the user.

As digital gaming continues to evolve, understanding these various components and their interplay is essential. Thus it is crucial to examine the potential challenges and requirements that may emerge within these components to improve the overall system.

2 Scope and Purpose of Hazard Analysis

The purpose of this document is to assess the potential hazards associated with the system under development. The ultimate goal is to implement strategies that either eliminate these hazards or reduce them to an acceptable level. To achieve this, the Failure Modes and Effects Analysis (FMEA) method was employed, which aided in systematically identifying and prioritizing hazards. A thorough analysis was conducted on various aspects of the system, including requirements, design, and code implementation.

The scope of this document is to identify possible hazards within the software components of the game system, including the game mechanics, user interface, and multiplayer functionalities. It aims to analyze the effects and causes of potential failures such as performance degradation and outright system failure. Through this, mitigation strategies, safety and security requirements for users were established. Importantly, the scope does not include any hardware components as the system is purely software based and any hardware hazards are not within the control of the developers.

3 Hazard Definition

Term	**Definition**
System Hazard	A condition that could foreseeably cause or contribute to the system going down or loss of performance.
Risk	A measure that indicates the likelihood of a system hazard.

Table 2: Definitions of System Hazard and Risk

A hazard, in the context of this system, is defined as any property, software, or component that leads to reduced performance or complete system failure.

4 System Boundaries and Components

4.1 System Overview

The system referred to in this document consists of the following major software components within the system:

- Dice Roll Simulation
- Score Calculation
- Player vs. Computer AI
- Multiplayer Functionality
- Dice Rendering
- User Interface
- Camera Control
- Audio Feedback
- Scoreboard Display

These components collectively form the core functionality of the game, from simulating realistic 3D dice rolls to providing engaging interactions between players. Each component plays a unique role in delivering a comprehensive gameplay experience and is essential to the operation of the system.

4.2 Component Descriptions

4.2.1 Dice Roll Simulation

This component simulates the physics-based rolling of 3D dice, providing a realistic and interactive experience for the players. It uses randomized initial velocities and rotations to ensure that each roll outcome is unpredictable, closely mimicking physical dice rolls.

4.2.2 Score Calculation

This component handles the calculation of scores based on Yahtzee rules, modified for the game's unique mechanics. It evaluates the outcome of each roll, applies any special bonuses, and determines the impact on each player's health bar, reducing it accordingly when a player loses.

4.2.3 Player vs. Computer AI

This component simulates a computer opponent, applying strategic choices based on current dice values and score potential. The AI adapts its moves based on the game state, using offensive or defensive strategies to reduce the player's health bar while attempting to maximize its own score. This AI is assumed to be basic in capabilities.

4.2.4 Multiplayer Functionality

This component allows multiple players to compete either locally or online. It manages turn-taking, score calculations, and health updates for each player, ensuring a smooth and engaging multiplayer experience.

4.2.5 Dice Rendering

This component is responsible for the 3D visual representation of the dice. It includes realistic textures, shadows, and animations to enhance the gaming experience. The rendering component ensures each dice roll feels tangible, giving feedback based on movement and resulting in a visually engaging display of outcomes.

4.2.6 User Interface

This component provides an accessible and organized display for players, including game controls, player information, health bars, and scoring details. It guides the player through the gameplay with minimal effort, enhancing user engagement and intuitiveness.

4.2.7 Camera Control

This component manages the in-game camera, providing a dynamic view of the dice, players, and game area. The camera adjusts automatically to showcase dice rolls and other key moments, ensuring players have a clear and immersive view of the action.

4.2.8 Audio Feedback

This component provides audio cues for interactions such as dice rolls, score updates, and health deductions. Sounds are integrated to give players feedback on their actions, enhancing immersion and reinforcing gameplay events.

4.2.9 Scoreboard Display

This component displays scores and health bar statuses for all players. It updates in real-time, showing each player's remaining health and overall score, allowing players to track their performance and make strategic decisions based on current standings.

5 Critical Assumptions

- **Godot Stability**: The Godot game engine is assumed to be stable and function correctly.
- **External Libraries Correctness**: We assume all third party libraries and plugins to function correctly given they have been used and tested by others.

6 Failure Mode and Effect Analysis

6.1 Hazards Considered Out of Scope

1. **Hardware-Specific Failures**
   - Issues related to hardware malfunctions such as GPU or CPU overheating, RAM failures, or hard drive corruption.
   - *Rationale*: The FMEA table is focused on software development within the Godot game engine, and hardware reliability is typically managed by the user's computer environment.
2. **Operating System Crashes or Instability**
   - Operating system crashes, updates, or security vulnerabilities that interrupt game sessions.
   - *Rationale*: These are dependent on the player's system and not directly related to the game's software development or behavior.
3. **Network Infrastructure Failures**
   - Failures due to external network outages, router malfunctions, or ISP-level disruptions. These would be long-term outages that make play impossible and thus would be out-of-scope, while latency and data de-sync issues would be considered.
   - *Rationale*: These are beyond the game's control and depend on the player's internet setup or service provider.
4. **Third-Party Library Bugs or Vulnerabilities**
   - Bugs or security vulnerabilities in third-party libraries or plugins used within the Godot engine.
   - *Rationale*: While the game relies on third-party tools, they are assumed to be correct, as we focus on bugs and issues within the game code itself.
5. **Player Misuse or Exploits**
   - Players intentionally trying to exploit the game, cheat, or use unauthorized modifications.
   - *Rationale*: Handling intentional misuse or hacking is outside the game's core development, and managing these issues requires external anti-cheat measures or monitoring. The game is intended to be played by invite link, meaning the two players would be communicating beforehand and have a level of trust.
6. **Data Privacy and Security Breaches**
   - Unauthorized data access, or privacy breaches.
   - *Rationale*: The FMEA focuses on in-game functionalities like dice rolls, AI, and scoring. Data privacy concerns involve external security practices and infrastructure, which are beyond the game's core software behavior.

7. **Non-Game Software Interference**
    - Interference from other software running on the user's machine, like antivirus programs or system background tasks.
    - *Rationale*: These external software influences are outside the game's scope of control and would be handled by system administrators or users.

6.2 FMEA Table

Text FMEA table allows for us to quantify values related to possible failures and document each failure's severity, likelihood, and detectability. These values can help guide our countermeasures and priorities in risk mitigation.

- Severity of Failure (0-10): This column measures the impact of a failure on the system, users, or other critical elements. The scale is based on how severe the consequences are if this failure occurs, where:
    - 0-3: Minor impact, little to no disruption or harm.
    - 4-6: Moderate impact, leading to some disruption or user inconvenience.
    - 7-10: Major impact, potentially causing system failures, or major user dissatisfaction.
- Likelihood of Occurrence (0-10): This column assesses how frequently the failure is likely to occur, based on past experiences and expert judgment. The scale can be interpreted as:
    - 0-3: Unlikely to occur, rarely observed in similar systems.
    - 4-6: Occasional, may occur under certain conditions.
    - 7-10: Frequent, likely to happen under normal operating conditions.
- Likelihood of Failure Detection (0-10): This column measures the probability that a failure will be detected before it leads to an unacceptable event, with lower scores indicating high detectability and higher scores indicating low detectability:
    - 0-3: Very likely to be detected in early stages due to robust detection mechanisms.
    - 4-6: Moderately likely, may require specific testing or user reports to detect.
    - 7-10: Very unlikely to be detected, likely only observable after causing a significant issue.

Table 3: FMEA Table

Component	**Failures**	**Unacceptable Event**	**Severity of Failure (0-10)**	**Cause of Failure**	**Likelihood of Occurrence (0-10)**	**Recommended Action**	**Likelihood of Failure Detection (0-10)**
Dice Roll Simulation	Physics misbehaves	Unrealistic dice behavior	8	Physics engine glitch	6	Refine physics settings; improve collision detection	4
Score Calculation	Incorrect scoring	Inaccurate score computation	9	Logic error in scoring algorithm	4	Unit test scoring algorithms thoroughly	6
Player vs. Computer AI	Poor AI decisions	Computer opponent is too easy/unpredictable	3	Sub-optimal AI strategy	6	Refine AI strategy based on probability analysis	9
Multiplayer Functionality	Connection loss	Player disconnects mid-game	7	Network instability	5	Implement reconnect feature; improve connection stability	5
Multiplayer Functionality	Data de-sync	Player have different outputs displayed	7	Network instability	4	Perform checks between clients; improve connection stability	7
Dice Rendering	Dice not visible	Players cannot see the dice clearly	3	Rendering glitch	3	Reduce 3D model poly counts for best rendering reliability; ensure camera angles cover dice	3

Component	Failures	Unacceptable Event	Severity of Failure (0-10)	Cause of Failure	Likelihood of Occurrence (0-10)	Recommended Action	Likelihood of Failure Detection (0-10)
User Interface	Missing or confusing UI	Players are confused by the interface	7	Inadequate UI design	5	Conduct user testing; iterate on UI design	8
User Interface	Invalid input	Player inputs are not valid	4	Unclear or unsanitized inputs	3	Conduct user testing; iterate on UI design	3
Camera Control	Unclear view of the board	Players can't properly view game elements	6	Inadequate camera angle logic	5	Allow manual camera adjustment; improve auto camera control	6
Audio Feedback	Missing or incorrect sounds	No sound feedback for player actions	4	Sound trigger event missed	5	Ensure audio events are linked to game actions with low latency	6
General Game Stability	Unexpected crashes during gameplay	Game session terminates abruptly	9	Memory leaks, unhandled exceptions, or rendering overload	4	Conduct stress tests; improve error handling and resource management	9
Scoreboard Display	Incorrect scores or missing player data on the scoreboard	Confusion over game results	6	Display update not synchronized with scoring logic	3	Ensure that scoreboard updates are triggered accurately, add validation	4

7 Safety and Security Requirements

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.

8 Roadmap

8.1 POC Demonstration

By POC Demonstrations, we will work towards R8 and R17, and it's base implications of current game standings, dice values, and win conditions to be accurately reflected to the user. We will also work towards NFR4, but will not expect to reach the less than 1% crash condition until later in the project

8.2 Revision 0 Demonstration

Expanding on R16 and R17, we will show a detailed description of the game's intermediary states at the end of a game. By this stage we will also have R1 functional on a stable peer to peer connection.By this stage, we will also have achieved NFR4's requirement of less than 1% crash rate.

8.3 Final (Revision 1) Demonstration

By Final Demonstration, based on R13, we will have a sufficiently sophisticated AI opponent that the player can play against. It will not be expected to have the best strategy, but enough for the player to enjoy the game consistently, to match NFR8's 75% user enjoyability metric. R3, R4, and NFR5 will also be completed by this stage.

8.4 Outside of Capstone Time line

After Final Demonstrations, R13's requirement for a sophisticated AI opponent can be expanded, creating different levels of opponents and for the different game modes.

Appendix — Reflection

1. What went well while writing this deliverable?

   - Team members communicated often and effectively. - Isaac G.
   - Code changes were handled well without merge conflicts. - Isaac G.
   - All team members contributed to the document, with sections distributed equally - Hemraj B.
   - Having already written the SRS documentation, we had a better, unified understanding of the project we were undertaking. - John P.

2. What pain points did you experience during this deliverable, and how did you resolve them?

   - With many midterms for everyone finding time to meet and work was a challenge. We were still 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.
   - Since the game is entirely software-based, identifying hazards initially posed a challenge, as there were no typical hazards one might associate with the term, such as risks to human safety. Therefore, we had to delve a bit deeper to identify potential hazards.

3. Which of your listed risks had your team thought of before this deliverable, and which did you think of while doing this deliverable? For the latter ones (ones you thought of while doing the Hazard Analysis), how did they come about?

   - We had already considered the importance of accurately simulating the dice rolls prior to this deliverable. If the dice simulation were not executed correctly, it could lead to a negative gameplay experience and undermine the game's integrity. - Hemraj B
   - In addition, we had also considered the importance of ensuring the stability of the network connection between the players prior to this deliverable. As Online Multiplayer is one of our core features, we knew if the network connection between users was too unstable, then they could be pre-maturely disconnected from their game, leading to a more negative gameplay experience - Naishan Guo
   - During the hazard analysis, we recognized the potential risks associated with audio feedback that we hadn't initially considered. These concerns emerged as we examined the various components more closely, prompting us to think about how each element could affect the overall user experience. Audio plays a significant role in indicating certain scenarios, and without it, users could be left confused about gameplay events. - Hemraj B

4. Other than the risk of physical harm (some projects may not have any appreciable risks of this form), list at least 2 other types of risk in software products. Why are they important to consider?

- Performance Risks: These involve the software failing to meet performance expectations due to degrading factors such as slow loading times or network lag in the case of online functionalities. Poor performance can lead to user frustration and the game could lose player base. - Hemraj B.
- Usability Risks: These risks stem from a poorly designed user interface or experience, which makes it difficult for users to use or comprehend the software. Thus usability risks are crucial to examine since usability concerns can cause user irritation, decreased engagement, and lower adoption rates. In our case, if the game is hard to navigate due to poor design, users may opt to try other games instead not wanting to waste their time. - Hemraj B.