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	Added content to introduction, scope, purpose and hazard defination sections
Date2	Name(s)	Description of changes
...	...	...

Contents

[You are free to modify this template. —SS]

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.

[You can include your definition of what a hazard is here. Hemraj note: added vague intro and moved defination to its own section —SS]

2 Scope and Purpose of Hazard Analysis

[You should say what **loss** could be incurred because of the hazards. —SS]

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

Latex	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

[Dividing the system into components will help you brainstorm the hazards. You shouldn't do a full design of the components, just get a feel for the major ones. For projects that involve hardware, the components will typically include each individual piece of hardware. If your software will have a database, or an important library, these are also potential components. —SS]

5 Critical Assumptions

[These assumptions that are made about the software or system. You should minimize the number of assumptions that remove potential hazards. For instance, you could assume a part will never fail, but it is generally better to include this potential failure mode. —SS]

6 Failure Mode and Effect Analysis

[Include your FMEA table here. This is the most important part of this document. —SS] [The safety requirements in the table do not have to have the prefix SR. The most important thing is to show traceability to your SRS. You might trace to requirements you have already written, or you might need to add new requirements. —SS] [If no safety requirement can be devised, other mitigation strategies can be entered in the table, including strategies involving providing additional documentation, and/or test cases. —SS]

7 Safety and Security Requirements

[Newly discovered requirements. These should also be added to the SRS. (A rationale design process how and why to fake it.) —SS]

8 Roadmap

[Which safety requirements will be implemented as part of the capstone timeline? Which requirements will be implemented in the future? —SS]

Appendix — Reflection

[Not required for CAS 741 —SS]

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?
2. What pain points did you experience during this deliverable, and how did you resolve them?
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?
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?