# Hazard Analysis
# Software Engineering

Team 8, RLCatan
Matthew Cheung
Sunny Yao
Rebecca Di Filippo
Jake Read

Table 1: Revision History

| **Date** | **Developer(s)** | **Change** |
|---|---|---|
| Date1 | Name(s) | Description of changes |
| Date2 | Name(s) | Description of changes |
| ... | ... | ... |

# Contents

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

# 1 Introduction

A hazard is defined as a property or condition in the system together with a condition in the environment that has the potential to cause harm or damage. We define harm as the underperformance of our system in Catan that leads to the incorrect instruction of players. In addition, we consider disruption of play, loss of game data/state as harm.

# 2 Scope and Purpose of Hazard Analysis

The loss that can be incurred because of the hazards defined above include the forfeiture or inability to continue a game of Catan. This can affect the user experience of our software and lead to a loss of trust in the product.

# 3 System Boundaries and Components

Components:

- The board representation
- The game state digital twin
- The AI model
- User Interface
- Game State Database

# 4 Critical Assumptions

We assume that the users of our system will be familiar with the rules of Catan and know how to play the game. We also assume that the users will have a basic understanding of how to interact with a digital interface. We also assume that existing consumer hardware will be able to run full depth processing of our AI model.

# 5 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]

## 6 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]

## 7 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?