**Executive Summary**

This document provides an overview of the functionality and system interactions within the card game Hanabi. The planned functionality of the game system is broken down into three categories: must-have, should-have, and might-have. These levels of functionality correspond to definitive game requirements—must-haves—and features that may be implemented—should-haves and might-haves. The game systems requirements are also discussed, including both software and hardware requirements. Elements of planned functionality are described in detail through the use of system actors and actions, the external entities that interact with the system and the ways they perform those interactions. Detailed images and descriptions for all planned user-interface elements are also included within the document’s appendices. This document does not delve into detailed implementation of system interactions, but provides the information required to describe the fully realized Hanabi card game.

**Introduction**

Hanabi is a card game played with two to five players. Hanabi is a cooperative game, i.e. a game where the players do not play against each other but work together towards a common goal. The deck of cards contains five suits (white, yellow, green, blue, and red): three 1’s, two each of 2’s, 3’s, and 4’s and one 5. The goal is to build all five fireworks by building sets of each colour in correct numerical order. The game begins with 8 available information tokens and 3 fuse tokens. The players are dealt a hand containing five cards (four for 4 or 5 players). The players are unable to see their own cards but can see every other players hand. Each turn a player can give information, discard a card, or play a card. Players lose immediately if all fuse tokens are gone, and win immediately if all 5’s have been played successfully. Otherwise the game continues until the deck is becomes empty, and for one full round after that. At the end of the game, the values of the highest cards in each suit are summed, resulting in a total score out of a possible 25 points.

**Glossary**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Give Information | The player points out the cards of either a given number or a given suit in the hand of another player |
| Discard a card | The player chooses to discard a card from their hand and adds it to the discard pile, then draws a card to replace it. The discarded card is out of the game and can no longer be played. Discarding a card replenishes one information token |
| Play a card | The player chooses a card from his hand and attempts to add it to the cards already played. This is successful if the card is a 1 in a suit that has not yet been played, or if it is the next number sequentially in a suit that has been played. Otherwise a fuse token is consumed and the misplayed card is discarded. Successfully playing a 5 of any suit replenishes one information token. Player draws a card regardless of outcome. |
| Fuse Token | 3 fuse tokens, players lose once they are all gone |
| Information Token | 8 information tokens, can be replenished throughout the game |

**References**

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

**Overview of Document**

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.