

THE NIM GAME

Software Requirements Specification



January 8, 2020

**Neumont college of computer science**

Prepared by Baydak, Byrd, Lopez, Holt, and Slocum

Version 1.0, pending approval

# Table of Contents

[Table of Contents 1](#_Toc29563378)

[Introduction 2](#_Toc29563379)

[1.1 Purpose 2](#_Toc29563380)

[1.2 Intended Audience and Reading Suggestions 2](#_Toc29563381)

[1.3 Product Scope 2](#_Toc29563382)

[1.4 References 2](#_Toc29563383)

[Overall Description 3](#_Toc29563384)

[2.1 Product Perspective 3](#_Toc29563385)

[2.2 Product Functions 3](#_Toc29563386)

[2.3 User Classes and Characteristics 3](#_Toc29563387)

[2.4 Operating Environment 3](#_Toc29563388)

[2.5 Design and Implementation Constraints 3](#_Toc29563389)

[2.6 User Documentation 4](#_Toc29563390)

[2.7 Assumptions and Dependencies 4](#_Toc29563391)

[External Interface Requirements 4](#_Toc29563392)

[3.1 User Interfaces 4](#_Toc29563393)

[3.2 Hardware Interfaces 4](#_Toc29563394)

[3.3 Software Interfaces 4](#_Toc29563395)

[System Features 5](#_Toc29563396)

[4.1 Description and Priority 5](#_Toc29563397)

[4.2 Stimulus/Response Sequences and Requirements 5](#_Toc29563398)

[4.3 Other Functional Requirements 5](#_Toc29563399)

[Appendix A: Glossary 6](#_Toc29563400)

[Appendix B: Analysis Models 6](#_Toc29563401)

# Introduction

## 1.1 Purpose

This document covers the software requirements for the desktop version of the Nim strategy game. It lays out how it is intended to be played, what classes there will be and how they interact, functionality and constraints, what environment the software is assumed to be running in, and any other information that might be of use to those maintaining or further developing the software.

## 1.2 Intended Audience and Reading Suggestions

This document is intended to be used by the maintainers and developers of the project, as well as the instructor of this course. It is not written for end-users as they will glean very little information that is pertinent to their operation of the software.

## 1.3 Product Scope

This document will mostly contain information on the structure of the program (i.e. what the developers of the software write), with little information on the technologies used (i.e. what others have written). The product will be a replication of the game of Nim [1]. It will also incorporate the features required as part of the CSC130 academic course [2].

## 1.4 References

* Wikipedia. (2019, November 30). *Nim.* Retrieved January 10, 2020, from https://en.wikipedia.org/wiki/Nim.
* Maple, R. (2020, January 8). *Waterfall Model* [In-class Document]. Neumont College of Computer Science.

# Overall Description

## 2.1 Product Perspective

This is a new, self-contained product. It has no planned follow-on members, isn't designed to replace any existing programs, and doesn’t integrate with another system.

## 2.2 Product Functions

* Heaps are displayed as rows.
* Pipes are used as the objects that compose heaps.
* A user can choose a row and a specific number of pipes to nim, up to all pipes in a heap.
* Players will take turns until all pipes have been nimmed.
* During the game setup, the player who chooses the last pipe or second to last pipe may be designated as the loser.
* A human player may choose to play against an A.I. or another human.
* Players can enter their names.
* The application will not close until the user chooses to exit.
* Instructions/help will be provided to the user.

## 2.3 User Classes and Characteristics

This software is designed for the enjoyment of college instructors and student peers.

## 2.4 Operating Environment

The program will run as a Discord bot and will be able to operate on any computer system provided that it meets the hardware requirements.

## 2.5 Design and Implementation Constraints

The only limitation should be the use of the JavaScript, Node.js, and Discord.js to write the program. No other limitation exists.

## 2.6 User Documentation

No documentation will be supplied with the software. The user should need no additional materials other than the built-in instructions included in the software.

## 2.7 Assumptions and Dependencies

This document should remain static provided that the document’s base understanding of the objectives are correct. This includes the understanding of Nim, the requirements of the assignment, and any requests from the instructor. The only dependencies this project will utilize are Discord.js, Node.js, and JavaScript

# External Interface Requirements

## 3.1 User Interfaces

The user interface will be provided by Discord.

## 3.2 Hardware Interfaces

Any computer system that supports Discord will be sufficient to run our software.

## 3.3 Software Interfaces

The software will be provided by Discord.js

# System Features

The Game of Nim

## 4.1 Description and Priority

We need the game to be able to work, as in every feature works correctly, which includes the difficulty and size selector, and “Artificial Intelligence” which in our case is the computer player. The AI is number 1 priority as the player wouldn’t be able to play the computer without it

## 4.2 Stimulus/Response Sequences and Requirements

The user will be able to start the app and choose whether they want to play with either the computer or another human person. Afterward, the program will ask them to choose the size/difficulty of the board. When the player selects an option, they will then play the game, if it’s with another person they take turns removing “matches” until they’re all gone, and the game will show who loses. The same thing happens when you select the play with computer option, except the computer is smart and will try to win 100% of the time.

## 4.3 Other Functional Requirements

Input validation and other improvements directed at preventing the program from crashing or the flow being disrupted.

# Appendix A: Glossary

* Heap: The row containing a set number of pipes.
* Pipe: The stick or line in the game

# Appendix B: Analysis Models