Software Requirements Specification

Project: Self-Teaching Tool for Concepts in Number Theory

Introduction

1. Purpose

The purpose of this document is to define the software requirements of a self-teaching visualisation tool for concepts in number theory.

2. Intended Audience

This product is intended for students or teachers in higher education for fields directly involving or adjacent to pure mathematics.

3. Intended Use

The visualisation tool can be used by individuals to develop a more profound intuition for how concepts in number theory interact and fit together. Educators can also use it in a group setting to aid class interactivity.

4. Scope

The scope of this document is to outline the functional and non-functional requirements of the tool mentioned above.

The tool's scope is to enhance the teaching of concepts in number theory - particularly those relevant to the mathematics underlying the RSA cryptosystem. The tool is not intended to be a sufficient teaching resource in isolation. Still, it will not explicitly refer to/require access to other resources.

5. Definitions and Acronyms

RSA: Refers to the Rivest-Shamir-Adleman algorithm for public-key encryption that is widely used in digital communications.

Screen: This document will take a screen to mean the illustration or teaching of one ‘lesson’ or set of interlocking concepts of a similar difficulty level that can be visualised in one scene.

System Features

1. Functional Requirements

User Management

The tool will **not** require users to log in/register and thus should not rely on any external information to determine its state at any point.

Content Management

1. Users should be able to alter the system's state to illustrate the highlighted concepts under variable parameters. For example, the user should be able to change the number of nodes in a ring or vary the speed at which the auto-play feature executes.
2. Each screen should clearly and visibly show how its content builds upon that illustrated in the previous screen(s).
3. The user must be able to access any other screen from every state of the system to refresh earlier concepts, skip ahead to previous states, and so on.

Visual Aids

1. Interactivity: As mentioned above, the user should be able to alter the state of the tool as desired to make the learning process more active than passive.
2. Animation: The tool should support the animation of various concepts to demonstrate state changes with the passage of ‘time’ - i.e. the repeated application of some operation.

Liveness

The tool must always render an initial graphic for relevant screens upon loading to be ready and responsive to any user input. This implies that the tool does not rely on the running environment, existing download/installation status, etc.

2. User Interfaces

The tool should have a simple and intuitive interface that allows the user to navigate easily between screens, with a responsive layout that is accessible on different devices and environments.

3. Hardware Interfaces

The users and tool itself do not require interfacing with hardware; hardware-related decisions and issues are handled by the technologies used to develop the system.

Non-Functional Requirements

1. Performance

The tool should load quickly and efficiently, with real-time re-rendering of graphics in response to user input.

2. Safety

The tool must not be able to access the user's running environment. Cookies are irrelevant to the tool's operation as session management is a clear non-requirement.

3. Security

There are no security requirements for this system, such as access control, encryption, etcetera, as no sensitive or user information is taken, stored, or processed.

4. User Experience

The user interface should be intuitive and easy to use, with minimal time/effort required for familiarisation with layouts.

5. Compatibility, Accessibility, and Availability

The tool must be widely accessible, with availability on a range of devices and platforms. Design choices must also consider accessibility needs, for example, visual impairments yielding varying colour schemes, font sizes, and animation speeds. These judgements should be in line with accessibility standards and guidelines the University of Edinburgh follows.

Constraints

1. Budget

The monetary budget for the project is very limited - presumed zero, with any expenditure requested needing tedious approval.

2. Timeline

The project must be completed within the well-defined timeline of The University of Edinburgh's School of Informatics Honours project. This involves completing all related work and user studies analysed by 6th April 2023.

Assumptions and Dependencies

1. Internet and Connectivity

This tool depends on the user(s) having a stable internet connection and an up-to-date web browsing application.

2. User Knowledge

Users are assumed to have a basic understanding of technology and the internet and an awareness of introductory concepts and areas in pure mathematics, such as those introduced in pre-honours university courses.

Acceptance Criteria

The product must meet all functional requirements outlined in this document, with evidence of validation and verification from testing and user acceptance. The non-functional requirements must all be met with evidence from testing and performance evaluation, or decisions to acknowledge failure to meet requirements must be justified with sufficient evidence and reasoning.