**Software Requirements Specification**

**for**

Super Snake

**Version 1.0 approved**

**Prepared by <authors>**

**<organization>**

**3/4/2024**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

## **Purpose**

*<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>*

## **Document Conventions**

*<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>*

## **Intended Audience and Reading Suggestions**

*<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>*

## **Product Scope**

*<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>*

## **References**

*<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>*

# **Overall Description**

## **Product Perspective**

*<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

## **Product Functions**

*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>*

## **User Classes and Characteristics**

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

## **Operating Environment**

*<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>*

## **Design and Implementation Constraints**

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>*

## **User Documentation**

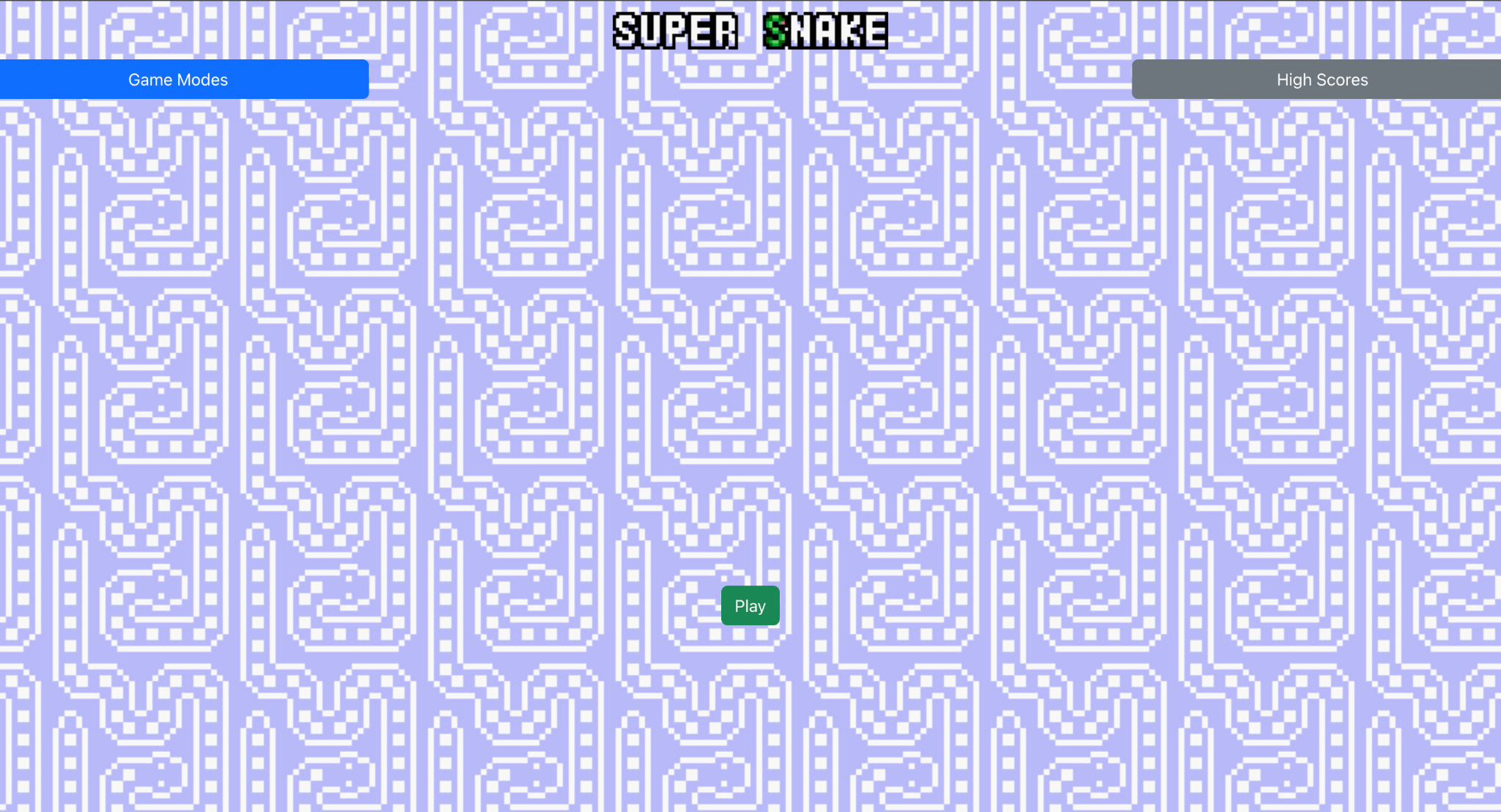
*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

## **Assumptions and Dependencies**

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

# **External Interface Requirements**

## **User Interfaces**

**

## **Hardware Interfaces**

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## **Software Interfaces**

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## **Communications Interfaces**

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# **System Features**

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

## Difficulty Setting

* + 1. Description:
    2. Stimulus:
    3. Functional Requirements:

REQ-1: The game shall have 3 difficulty settings: easy, normal, difficult.

REQ-2: The difficulty setting shall be adjustable before playing the game.

REQ-3: The difficulty settings shall have a brief description of how they alter the gameplay.

REQ-4: The difficulty setting shall be adjusted within the user interface.

## Player Sprite Setting

* + 1. Description:
    2. Stimulus:
    3. Functional Requirements:

REQ-1: The game shall have 3 different snake sprites to choose from: Regular, Inverted, and RBG.

REQ-2: The sprite setting shall be adjustable before playing the game.

REQ-3: The sprite settings shall have a brief description of lore for that skin.

REQ-4: The sprite setting shall be adjusted within the user interface.

REQ-5: The sprite setting shall be adjustable while playing the game.

REQ-6: New sprite settings shall be added as the player reaches higher scores.

## Score

* + 1. Description:
    2. Stimulus:
    3. Functional Requirements:

REQ-1: The game shall allow you to send your score in an sms message.

REQ-2: The game shall record the score of each run through.

REQ-3: The game shall allow you to view 10 other users' highest scores.

REQ-4: The game shall allow you to view your own previous 10 highest scores.

REQ-5: The game shall have a direct accumulation of one point per Snake-Bite consumed.

## Game Board

* + 1. Description:
    2. Stimulus:
    3. Functional Requirements:  
        REQ-1: The board size for the default game mode shall be 48 units wide by 27 units high.  
        REQ-2: The outer wall that kills the player will have a thickness of 1 unit.

# **Other Nonfunctional Requirements**

## **Performance Requirements**

NON-REQ-1: The game shall run at atleast 60 frames per second.

NON-REQ-2: The game shall not crash more than 2% of the time.

NON-REQ-3: Sending the score in an sms reliably will make it to the receiver 95% of the time.

NON-REQ-4: The game shall react to user input within 500ms 95% of the time.

NON-REQ-5: The game shall load within 500 ms from when the user clicks start at least 95% of the time.

## **Safety Requirements**

NON-REQ-1: The game shall offer a graphical mode that includes specific colors like blue and orange for color sensitive users to differentiate objects easier.

## **Security Requirements**

NON-REQ-1: The game shall stop 95% javascript script injection.

NON-REQ-2: The game shall not pose a risk of downloading viruses to a user’s computer at least 98% of the time.

NON-REQ-3: The game shall protect credential information given 95% of the time by storing all passwords as hashed.

NON-REQ-4: The game shall stop inappropriate usernames according to section 5.3.1 80% of the time

## Username Requirements

NON-REQ-1: Usernames including any profanity in English, Spanish, and French will be blocked 80% of the time.

NON-REQ-2: Usernames including any insults to other players will be blocked 80% of the time.

NON-REQ-3: Usernames including any offensive terms will be blocked 80% of the time.

## **Software Quality Attributes**

NON-REQ-1: The game will run at 30 fps on systems 99.9% of systems.

NON-REQ-2: The game will display at the proper resolution on systems that run with most of the common display settings, at least 95% of the time.

## **Business Rules**

*<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>*

# **Other Requirements**

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**Appendix A: Glossary**

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*