**Software Engineering**

**Software Requirements Specification**

**(SRS) Document**

**< >**

**<9/21/2024>**

**<1.0>**

**<RainCity Studios>**

|  |
| --- |
| **Revisions** |

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1 | Finley Meadows and Michael Anderson |  | tbd |

|  |
| --- |
| **Review & Approval** |

Requirements Document Approval History

| Approving Party | Version Approved | Signature | Date |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Requirements Document Review History

| Reviewer | Version Reviewed | Signature | Date |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| **Table of Contents** |

[1. Introduction 3](#_Toc244519333)

1.1 Purpose 3

1.2 Document Conventions…………………………………………………………………….3

1.3 Intended Audience…………………………………………………………………………3

1.4 Scope……………………………………………………………………………………….3

1.5 References………………………………………………………………………………….3

[2. General Description 3](#_Toc244519334)

2.1 Product Perspective………………………………………………………………………...3

2.2 Product Features 3

2.3 User Class Characteristics 3

2.4 Operating Environment 3

2.5 Constraints 3

2.6 Assumptions and Dependencies 3

[3. System Requirements 3](#_Toc244519335)

[4. External Interface Requirements 4](#_Toc244519336)

[4.1 User Interfaces 4](#_Toc244519337)

[4.2 Hardware Interfaces 4](#_Toc244519338)

[4.3 Communications Interfaces 4](#_Toc244519339)

[4.4 Software Interfaces 4](#_Toc244519340)

[5. Non Functional Requirements 4](#_Toc244519341)

1. Introduction

* 1. **Purpose:** The goal of your project and the objectives it aims to accomplish

The goal of our project is to create an interactive world with which players can create their own civilizations on many different planets. The game hopes to simulate the hardships of building a cohesive society.

* 1. **Document conventions:** The typographical methodologies followed within the document. For e.g. any abbreviations, typographical stylization of content or change of fonts and its significance.

There are no changes that are currently needed for typographical methodologies.

* 1. **Intended audience:** Describe which part of the SRS document is intended for which reader. Include a list of all stakeholders of the project, developers, project managers, and testers for better clarity.

**Stakeholders: We are not important enough to have any of these.**

**Developers: Developers should read through the entire document that way they understand all the requirements.**

**Project Managers: Project Managers also should make sure to understand the project throughly, so that they can divide work accordingly.**

**Testers: Testers should read sections 2 and 4 to understand the functions the game should perform as they are doing playthroughs.**

* 1. **Scope:** Specify how the software goals align with the overall business goals and outline the benefits of the project to business.

**This project benefits us because it allows us to test our programing and game design skills and create a fun project we can share with others.**

* 1. **References:** A list of other documents that the SRS document refers to including sources such as websites or written literature.

**We don’t have anything to refer to.**

## 2. General Description

**2.1 Product perspective:** Describe the context and origin of the product

We asked ChatGPT to give us ideas for games that can be created using java and jframe, and it gave us the idea of a city builder. Taking this generic idea, we organized different ideas and came up with some unique twists that our city builder game could have.

**2.2 Product features:** A high level summary of the functions the software would perform and the features to be included.

The game will be run through an IDE than runs java. When ran, it will display a window for the user to interact with and play the game. This window and the components within it will be generated using the javax.swing library and java.awt library for smaller details. It will begin with displaying a title screen, giving the user the option to either create a new save file, or load a preexisting save to continue a run. After the user has selected one of the options, the game will begin. The game window will display a peaceful meadow with tiles to be built upon in the bottom half of the screen. When the user clicks on one of the tiles, a shop window will appear, displaying all the build options and their prices. This will be the core mechanic of the game. Along with the complexities that come with building a city, the player will have to deal with spontaneous events such as natural disasters, plagues, etc. This will add another layer of difficulty and create a niche for the game. After the player reaches maximum capacity for the city after upgrading and optimizing all of their structures, they will be given the option to create a new civilization on a different planet. This would come with new spontaneous events, and new more valuable resources.

* 1. **User class and characteristics:** A categorization and profiling of the users the software is intended for and their classification into different user classes

The user needs to own a computer, have access to the internet, and have an IDE to run java.

* 1. **Operating environment:** Specification of the environment the software is being designed to operate in.

This game is designed to operate on laptops, PC’s, and desktop computers. It does not need to have the ability to be played on any other consoles.

* 1. **Constraints:** Any limiting factors that would pose challenge to the development of the software. These include both design as well as implementation constraints.

We are using JFrame, which can be very difficult to use and may not have easy ways to perform many of the tasks we may require it to accomplish.

* 1. **Assumptions and dependencies:** A list of all assumptions that you have made regarding the software product and the environment along with any external dependencies which may affect the project.

We are assuming that the player has access to an IDE that can run java and run the downloadable game files.

## 3. System Requirements

**3.1 Functional requirements**

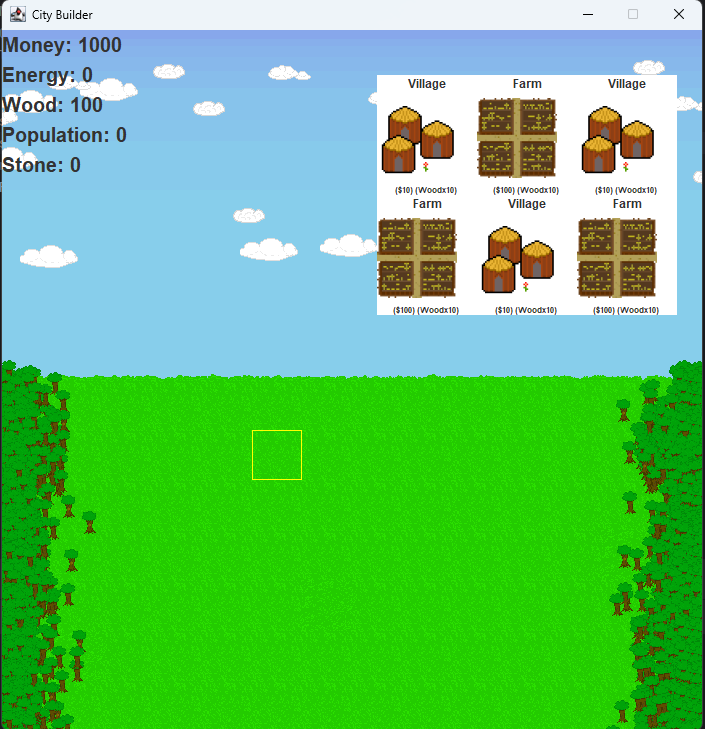
All the requirements within the system or sub-system in order to determine the output that the software is expected to give in relation to the given input. These consist of the design requirements, graphics requirements, operating system requirements and constraints if any.

The user needs to have a computer with enough RAM to generate and steadily maintain 500+ components at a time.

## 4.External Interface Requirements

4.1 User Interfaces

The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.



4.2 Hardware Interfaces

All the hardware-software interactions with the list of supported devices on which the software is intended to run on, the network requirements along with the list of communication protocols to be used.

The project requires an IDE that runs java.

4.3 Communications Interfaces

Determination of all the communication standards to be utilized by the software as a part of the project

There are no communications interfaces required.

4.4 Software Interfaces

The interaction of the software to be developed with other software components such as frontend and the backend framework to the used, the database management system and libraries describing the need and the purpose behind each of them.

Javax.swing: used to run all of the visual components

Java.awt: used to add detail to the components and provide mouse listeners

Java.util: provides the Collections class for data management

## 5. Non-Functional Requirements

**5.1 Performance requirements**

The performance requirements need to be specified for every functional requirement. The rationale behind it also needs to be elaborated upon.

A computer is required to run our game.

**5.2 Safety requirements**

List out any safeguards that need to be incorporated as a measure against any possible harm the use of the software application may cause.

We are creating a game for fun, so I do not believe that there is any way anyone could cause harm using our software application.

**5.3 Security requirements**

Privacy and data protection regulations that need to be adhered to while designing of the product

None, because we are making a f2p game designed for fun. The game does not require any data of any kind.

**5.4 Software quality attributes**

Detailing on the additional qualities that need to be incorporated within the software like maintainability, adaptability, flexibility, usability, reliability, portability etc.

There are no quality attributes required, besides a computer that won’t crash while running the code as we are not sure how to create save data yet, so you may have to complete the game all in one run without stopping or interrupting the run in any way.

**5.5 Other requirements**

These may include the legal requirements, resource utilizations, future updates etc.

There are no other requirements.