

SE 3XA3: Software Requirements
Specification
Title of Project

Team 30, VUA
Andy Hameed and hameea1
Usman Irfan and mac Id
Vaibhav Chadah and macid

October 5, 2018

Contents

1	Project Drivers	1
1.1	The Purpose of the Project	1
1.2	The Stakeholders	2
1.2.1	The Client	2
1.2.2	The Customers	2
1.2.3	Other Stakeholders	2
1.3	Mandated Constraints	3
1.4	Naming Conventions and Terminology	3
1.5	Relevant Facts and Assumptions	4
2	Functional Requirements	5
2.1	The Scope of the Work and the Product	5
2.1.1	The Context of the Work	5
2.1.2	Work Partitioning	5
2.1.3	Individual Product Use Cases	5
2.2	Functional Requirements	5
3	Non-functional Requirements	5
3.1	Look and Feel Requirements	5
3.2	Usability and Humanity Requirements	5
3.3	Performance Requirements	5
3.4	Operational and Environmental Requirements	5
3.5	Maintainability and Support Requirements	5
3.6	Security Requirements	5
3.7	Cultural Requirements	5
3.8	Legal Requirements	5
3.9	Health and Safety Requirements	5
4	Project Issues	5
4.1	Open Issues	5
4.2	Off-the-Shelf Solutions	6
4.3	New Problems	6
4.3.1	Effects on the Current Environment	6
4.3.2	Effects on the Installed Systems	6
4.4	Tasks	7
4.5	Migration to the New Product	7

4.6	Risks	8
4.7	Costs	8
4.8	User Documentation and Training	8
4.9	Waiting Room	9
4.10	Ideas for Solutions	9
5	Appendix	11
5.1	Symbolic Parameters	11

List of Tables

1	Revision History	ii
---	----------------------------	----

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

This document describes the requirements for The template for the Software Requirements Specification (SRS) is a subset of the Volere template (Robertson and Robertson, 2012). If you make further modifications to the template, you should explicitly state what modifications were made.

1 Project Drivers

1.1 The Purpose of the Project

Almost everyone nowadays relies on a computer as a multipurpose tool for research, video streaming, gaming and many other tasks. With the emergence of fast computing, gaming has become a popular pastime activity and a source of entertainment for many. However, not everyone has a device powerful enough to support extensive game applications. A simple installation and memory-efficient application of the Snake game allows it to be accessible for gamers without the need for extensive hardware or a high-performance computer. Our team, VUA30, will be creating a desktop application for the well-known “Snake” game with new enhancements and features. This competitive and addictive game can allow the user to play at their own pace and challenge their own high score.

Buying a computing device with high storage and faster performance can be out of budget. Complicated software covers up all the storage and the user is bound to use these applications as opposed to downloading other software. The importance of the redevelopment of “The Snake” is to save computing device’s personal storage and allow the user to play a game 24/7 with strong performance, even offline. Creating a desktop version of the snake game can fit into the category of downloadable classical games such as the solitaire suite. The recreation of this game will allow the user to enjoy the classical game anytime and anywhere as long as they have installed the application. Improving aspects such as graphics and custom speed will also make the game more interesting. We would like to add more features to the game to make it more customizable and help people enjoy the classical game in an exciting and new way.

1.2 The Stakeholders

Stakeholders involved will be contained within the gaming community, more specifically the desktop gaming community and casual PC owners who are looking for a fun reliever for boredom or quick game to play.

1.2.1 The Client

Since this game is a separate entity, the clients are the designers in this project team. In further developments and upon increase in game popularity, the clients could be a desktop gaming distribution service such as steam, google play or apple store. Otherwise, the main client would be Dr. Bokhari who has assigned the project.

1.2.2 The Customers

The main users or customers are desktop gamers, older generation of game enthusiasts, and youth, teens. However, the client can be anyone with a PC and an interest in classical gaming or a sudden craving for playing the classical Snake game. Often times, these games are a quick fix to boredom for those who are casually browsing their PC's, so the game will be designed to provide enough stimulus and excitement for regular computer users, similar to the solitaire suite.

1.2.3 Other Stakeholders

Aside from the clients and customers, other stakeholder include 3rd party Desktop game distribution stores and open source project banks which may make use of this project for development purposes:

- 3rd party Desktop game distribution stores
- Game Testers
- Technology Experts [Part of Project Team]
- Usability experts
- Dr. Bokhari
- Project Development Experts: This can include teaching assistants, the professor, experienced peers and so on

1.3 Mandated Constraints

Some constraints that apply to the project include the following:

- No project budget provided; Project cannot use costly API memberships or resources
- Application should take less than 400MB of storage space to meet requirements
- The project must be completed within a 4-month period
- Limited resources in terms of domain experts, specifically in graphic design
- Application will be developed for one OS due to time constraint
- open source project must be translated to Python due to development language and scope

1.4 Naming Conventions and Terminology

The naming conventions listed below will be used to clearly define words and terminology that will come up in the project development process. Below is a list of naming conventions, terms, and special vocabulary and their meaning. Since the desktop application is straightforward, there is not much terminology being used as of now:

- DDS: Digital Distribution Service such as play store, microsoft play, etc.
- OS: Operating System
- Python: The programming language used for application development
- Pygame: Computer graphics Python library
- Snake 2.0: The desktop application being developed in Python
- The interface: The graphics developed using Pygame
- The source game: The open source original Python snake game being used for this project

1.5 Relevant Facts and Assumptions

Some factors that might affect the outcome of the product are listed as follows:

- Digital Distribution services contribution will be necessary for the public release of the game
- Contribution of the development team will affect the outcome of the product
- Feedback from game testers
- Availability of resources from pygame library to replicate front-end design in HTML,CSS and JS
- Time remaining once initial objectives and goals are met. This could affect which additional functionality is added

Some Assumptions that are made pertain to the intended operational environment and anything effecting the product:

- Pygame library offers enough functionality to recreate the web app graphics in Python
- The user is using Windows for game execution otherwise they must compile the source code to run the application
- The application will not be an exact replica of the source game. Added functionality and a change of graphics is expected
- The game application will prioritize the completion of the snake game as the central attraction

Some user characteristics will affect the final deisgn and written requirements:

- Users expect the game to be responsive and timely due to the nature of wanting quick stimulus
- The game should have an attractive user inteface due to the nature of the users expectations. It is mainly used for entertainment and should have a smooth user-interface

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

2.1.2 Work Partitioning

2.1.3 Individual Product Use Cases

2.2 Functional Requirements

3 Non-functional Requirements

3.1 Look and Feel Requirements

3.2 Usability and Humanity Requirements

3.3 Performance Requirements

3.4 Operational and Environmental Requirements

3.5 Maintainability and Support Requirements

3.6 Security Requirements

3.7 Cultural Requirements

3.8 Legal Requirements

3.9 Health and Safety Requirements

This section is not in the original Volere template, but health and safety are issues that should be considered for every engineering project.

4 Project Issues

4.1 Open Issues

Below is a list of open issues pertaining to the project scope:

- Investigating and understanding the capabilities of the Pygame library is yet to be completed
- Integrating additional features is not decided on as of yet. It is dependant on time constraints
- snake-game multiplayer mode is an open issue on the open source project which we may or may not choose to implement as time permits

4.2 Off-the-Shelf Solutions

Although there are available solutions on developing such a game, the project team is aiming to enhance the game by producing a desktop version with added functionality.

Ready-made simple implementations of the projects are available and can be used as reference but otherwise, enhanced features will have to be created from scratch (light/ dark theme, custom player settings, high scores and so on)

4.3 New Problems

4.3.1 Effects on the Current Environment

The Microsoft Store contains the "250k snake" app for windows, an implementation of the old-school snake game. Aside from this application, other applications that appear when searching "snake" or "snake game" do not reflect the classical snake game. By developing the snake game as a desktop app, we will be able to provide game shoppers with more options to pick from.

4.3.2 Effects on the Installed Systems

The existence of the 250k snake will make it difficult to push the project team's implementation of the game, Snake 2.0, into the microsoft store market successfully. However, the new snake game will fill a niche for customizability by allowing users to pick from many different settings.

4.4 Tasks

An article on linkedIn by Sumit Jain summarizes the steps involved in the game development process [Jain (May 2017)]. In his article, he outline 6 main steps to the game development cycle: Idea & Story, Conceptualize & Design, Technical Analysis, Development, Testing, Deployment. Considering the project scope and the redevelopment of the snake game, the main three steps involved in the development cycle are the following:

- Technical Analysis: Use reverse engineering to understand how the game was originally built and analyze the main modules/ framework used to develop the game.
- Development: Using Python and Pygame to develop the source code for the game; Analysis from the previous step will be necessary to break down the development process.
- Testing: Test using unittest in python and principles of white box and black box testing. In further developments, this would also include integration testing with the user interface and the collection of modules created for the application.

Project members should expect the development cycle to resemble the previously mentioned framework. Once the cycle has been iterated until completion of Snake 2.0, the team will move on to the deployment stage, considering options for making the game available on a DDS such as the Microsoft Store.

4.5 Migration to the New Product

Snake 2.0 will require the following conversion tasks:

- Converting JS,HTML and CSS graphics and animations to Pygame graphics
- Converting the source project into modularized step-based tasks
- Converting from JS,HTML and CSS source code to Python source code

The source project will be run with Snake 2.0 for performance comparison and visual feedback on the accuracy of the redevelopment as well as the enhance features that were added to snake 2.0.

4.6 Risks

Snake 2.0 will be a classical desktop application and therefore does not present many risks to the user or any stakeholders involved. In terms of taking risk to advance the project, there is risk in striving for the completion of a multiplayer mode for the game since it may take substantial time and effort. However, this risk is low since the project requirements have already been met and other features of the game have been enhanced, aside from the addition of a multiplayer mode.

In the case that more risks are perceived in the future, the project team will take the following course of action to come up with early warnings:

- If the development is taking place 1 week prior to the project deadline, an early warning will be issued and the group must decide to continue or dismiss the development.
- If the development is currently taking place with 2 weeks left until the project deadline and less than 50% of the development is in place, it will be dismissed.
- If the main project is missing any component (testing, code modularization, documentation, commenting, etc.) no development will proceed until the main requirements (minimum requirements) are met.
- If any of the main project components are deemed to have lower quality, a warning is issued and the team members must discuss whether to continue with further development or improving the existing product.

4.7 Costs

As mentioned in the development plan document, team members will be dedicating 2 hours outside of lab time for team meetings and discussions along with 5 hours of individual work on the project itself. Since the project is open source and uses open libraries such as Pygame, the monetary cost is \$0. However, there may be additional costs to publishing Snake 2.0 with a DDS.

4.8 User Documentation and Training

The user will be provided with the following documentation and training:

- Snake 2.o User Manual: The document will explain the basic permisses of the game, user settings, graphic themes, menu headings, and any other information necessary for the user to understand the features of the game.
- Snake 2.o Installation Manual: Provided that the user will not be using Windows or the native OS that is decided on, this document will provide simple installation instructions for compiling the code on different OS's.

4.9 Waiting Room

In future releases of the project, the following requirements might be included in the revised requirements document:

- Snake 2.o User Manual - Multiplayer Mode: A section explaining how to connect and play the snake game with friends
- additional 'multiplayer mode' module: A separate module to encapsulate the multiplayer mode
- additional 'themes' module - a module encapsulating the different graphic themes available for the game

4.10 Ideas for Solutions

Some rudimentary ideas for project modules and solutions have been mentioned down below:

- Classes/modules for individual objects like the snake, food block, the frame, the menu bar, the settings bar and so on.
- import graphics developed in adobe illustrator into the game as characters, props and so on.
- the snake class can have method that correspond to the snakes functionality such as moveLeft, moveRight, moveUp, moveDown, and Lengthen.
- the food item can have a randomPlacement method for when being placed at random around the window.

- UI: a custom header section can contain the entry fields for custom speed and other important parameters.

References

Sumit Jain. Game development life cycle. May 2017.

James Robertson and Suzanne Robertson. *Volere Requirements Specification Template*. Atlantic Systems Guild Limited, 16 edition, 2012.

5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

5.1 Symbolic Parameters

The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.