Software Requirements Specification SNAKE

Alex Guerrero, Keyur Patel and Shafeeq Rabbani

December 8, 2015

Revisions

Table 1: Revisions

Name	Date	Description
Keyur Patel	06/12/2015	Created Revision Table
Keyur Patel	06/12/2015	Fixed up Requirements and numbering
Alex Guerrero	06/12/2015	Fixed up Requirements and numbering

Contents

Project Drivers

0.1 The Purpose of the Project

The purpose of Snake is to help users have fun. Snake is designed to engage users in an enjoyable activity which encourages them to use strategic planning and time management to achieve higher and higher scores.

0.2 The Stakeholders and Users

The client for this project is the administrative staff that teach SFWR ENG 3XA3. This includes Dr. Smith, and the teaching assistants for the course. They are the recipients of the course work and are responsible for the supervision and evaluation of the product. Other stakeholders include the team members working on the project, and classmates who would want to critique and enjoy playing the game. This product will be (potentially) available as an open source software to the online community. This group of users can vary from beginner programmers to professionals in the computing industry. Beginner programmers are the lowest level of users because this product is delivered in a programming language and uses libraries that does not allow it to be packaged into an executable.

Project Constraints

0.3 Mandated Constraints

There is a time constraint for this product which says that it must be completed by the first week of December 2015. Another constraint states that the product must not be implemented on a mobile device, the rationale being that the professor and TAs who will be evaluating this product may not have access to the devices used. Because the team members designing the product are university students, there is no funding and the budget must be limited to a total of $0.00 CAD. The current system is implemented on any personal computer or laptop which has 32-bit Python 2.7, and PyGame installed. No other hardware restrictions are known in order to implement the current system.

0.4 Naming Conventions and Terminology

- Client: Administrative staff of SFWR ENG 3XA3 (Doctor Smith and TAs)

- Constraint: A global requirement that affects decisions about the scope of the project.

- Stakeholder: A person, group or organization that has interest or concern in the project.

- Player: A person who plays the video game.

- Snake: The classic video game conceptualized in the late 1970s, where a player manoeuvres a line which grows in length upon touching obstacles that randomly appear throughout the game with the line itself being a primary obstacle

0.5 Relevant Facts and Assumptions

The existing application only has 91 lines of Python code, which provides the barebones of the ideal snake game. There is no documentation for this code and it has not been modularized.

Functional Requirements

0.6 The Scope of the Work

The scope of the work is limited to a simple redevelopment and implementation of the Snake game.

0.7 Business Data Model and Data Dictionary

Not applicable for this project.

0.8 The Scope of the Product

Refer to The Scope of the Work.

0.9 Functional Requirements

- R1: The player must be able to start the game by a single click.

- R1: The game must start with a main menu screen with a play game button, quit game button, and three difficulty buttons from 1-3

- R2: When the play game button is pressed, and instruction will appear and wait for user input.

- R3: The snake must be controlled by the keyboard. w, a, s, d or directional keys.

- R3: When the snake hits an obstacle, the game must freeze indicating that the game is over.

- R4: If the snake goes over the same location of a food object, a new food object will be generated and the snake will grow.

- R5: Preceeding the instructions screen, the gameboard will appear with a single snake and food object.

- R6: Pressing the esc key during the game brings up a pause screen that displays resume, main menu and quit game buttons.

- R7: As the player advances in the game, the snake moves faster.

- R8: When the snake hits the border or itself (after power up is used), the game is over.

- R9: The game calculates a score that is based on the length of the snake. ~~and the amount of time the player managed to play for without hitting any obstacle.~~

- R10: The game over screen displays the score and a retry buttons and quit game button. ~~outputs the score after the game is finished.~~

- R11: The snake will start with a power up that allows the player to collide with the snake body once without consequence.

Non-functional Requirements

0.10 Look and Feel Requirements

Appearance Requirements

- 1: The product must be able to display a main menu with buttons and lists.

- 2: The product must be able to display an error window if anything occurs.

- 3: The product should have a soundtrack playing throughout main menu selection and game play.

Style Requirements

- 1: The product shall appear minimalistic in terms of graphics and visuals.

0.11 Usability and Humanity Requirements

Ease of Use Requirements

- 1: The product must be easy to use by anyone from children to seniors (after installation of required software).

- 2: The casual user is expected to remember at most 7 keys and their functions.

Personalization and Internationalization Requirements

Not Applicable for this project.

Learning Requirements

- 1: The product shall be easy for a child to learn once the software requirements are met.

Understandability and Politeness Requirements

- 1: The product shall use common directional movement keys.

- 2: The product shall hide its implementation and construction from the user.

Accessibility Requirements

Not Applicable for this project

0.12   Performance Requirements

Speed and Latency Requirements

- 1: The response to any keys shall be fast enough to avoid delayed actions (lag).

Safety Critical Requirements

Not Applicable for this project.

Precision or Accuracy Requirements

Not Applicable for this project.

Reliability and Availability Requirements

- 1: The product shall be usable for 24 hours of the day, as it does not require network connectivity.

Robustness or Fault-Tolerance Requirements

Not Applicable for this project.

Capacity Requirements

Not Applicable for this project.

Scalability and Extensibility Requirements

Not Applicable for this project.

Longevity Requirements

Not Applicable for this project.

0.13   Operational and Environmental Requirements

Expected Physical Environment

- 1: The product shall be usable in any environment where the computer it is being run on can be used.

Requirements for Interfacing with Adjacent Systems

Not Applicable to this project.

Productization Requirements

- 1: The product shall be distributed as a ZIP file. It will be available for download from the team members' Gitlab.

- 2: The product shall be able to be installed by a user with the aid of instructions.

Release Requirements

Not Applicable for this project.

0.14   Maintainability and Support Requirements

Maintenance Requirements

- 1: The product shall be able to be maintained by developers who are not the original developers.

- 2: The structure of the code and the documentation provided should allow for maintainability of the code.

Supportability Requirements

Not Applicable for this project.

Adaptability Requirements

- 1: The product is expected to run on any OS which supports Python 2.7.

0.15 Security Requirements

Access Requirements

Not Applicable for this project.

Integrity Requirements

Not Applicable for this project.

Privacy Requirements

Not Applicable for this project.

Audit Requirements

Not Applicable for this project

Immunity Requirements

Not Applicable for this project.

0.16 Cultural Requirements

Cultural Requirements

- 1: The product shall not be offensive to religious or ethnic groups.

Compliance Requirements

Not Applicable for this project.

Standards Requirements

Not Applicable for this Project.

Project Issues

0.17 Open Issues

Currently there are no open issues to be considered.

0.18 Off-the-Shelf Solutions

There are a vast amount of off-the-shelf solutions for this Product. As mentioned before, snake has been around since the late 1970s and because of its simplicity as a game many people have made their own implementation of it of varying quality. One example of a well made snake game version on http://playsnake.org which has clean graphics, different playmodes, and leaderboards.

0.19 New Problems

Playing for prolonged periods of time can adversely affect people's health and their day to day lives.

0.20 Tasks

During the redevelopment process, the team members will frequently jump back and forth between low level implementation and high level system requirements, modifying each respectively as we progress through the project. The deliverable due dates are outlined on the 3XA3 Avenue to Learn website and must be completed by December.

0.21 Migration of the New Product

Not Applicable to this project.

0.22 Risks

Engineering students have busy schedules because they are required to take a heavier course load. With numerous other assignments, projects, and test going on concurrently with this project, managing time will be difficult.

0.23 Costs

Not Applicable to this project.

0.24 User Documentation and Training

Upon completion in December, the program will be accompanied by full documentation including: the problem statement, requirements document, design document, test plan, test report, user's guide, and the source code.

0.25 Waiting Room

These are ideas which are going to be considered in the future but are not being developed currently.

When the player collects a certain power-up, the snake gets the 'Intangibility' feature allowing it to pass through all obstacles including its own body. This power-up will make it more convenient for the player to collect food and to do so more quickly.

The 'Sacrifice' feature would allow the player to continue playing the game even though the snake is about to take the hit. It will do this by allowing the player to cut the length of the snake at the expense of some points.

The Freeze feature would slow the snake. This is a feature that is exceptionally handy when the snake is moving really fast at much later stages of the game.

0.26 Ideas for Solutions

Not Applicable to this project.