SE 3XA3: Software Requirements Specification[1]

Team L03G09
Qiang Gao gaoq20
Zhiwei Li liz342
Longwei Ye yel16

February 11, 2022

Contents

1 Project Drivers **1**
- 1.1 The Purpose of the Project . . . 1
- 1.2 The Stakeholders . . . 1
  - 1.2.1 The Client . . . 1
  - 1.2.2 The Customers . . . 1
  - 1.2.3 Other Stakeholders . . . 1
- 1.3 Mandated Constraints . . . 2
  - 1.3.1 Schedule Constraints . . . 2
  - 1.3.2 Enterprise Constraints . . . 2
- 1.4 Naming Conventions and Terminology . . . 2
- 1.5 Relevant Facts and Assumptions . . . 4

2 Functional Requirements **4**
- 2.1 The Scope of the Work and the Product . . . 4
  - 2.1.1 The Context of the Work . . . 4
  - 2.1.2 Work Partitioning . . . 5
- 2.2 Functional Requirements . . . 5
  - 2.2.1 Start the game . . . 5
  - 2.2.2 Playing the game . . . 6
  - 2.2.3 End the game . . . 7
  - 2.2.4 During the game: snake eats a booster element . . . 8

3 Non-functional Requirements **9**
- 3.1 Look and Feel Requirements . . . 9
- 3.2 Usability and Humanity Requirements . . . 9
- 3.3 Performance Requirements . . . 9
  - 3.3.1 Capacity Requirements . . . 9
- 3.4 Operational and Environmental Requirements . . . 9
- 3.5 Maintainability and Support Requirements . . . 9
  - 3.5.1 Maintainability Requirements . . . 9
  - 3.5.2 Support Requirements . . . 10
- 3.6 Security Requirements . . . 10
  - 3.6.1 Privacy Requirements . . . 10
- 3.7 Cultural Requirements . . . 10
- 3.8 Legal Requirements . . . 10
- 3.9 Health and Safety Requirements . . . 10

4 Project Issues **10**
4.1 Open Issues . . . 10
4.2 Off-the-Shelf Solutions . . . 10
4.3 New Problems . . . 11
4.3.1 Effects on the Current Environment . . . 11
4.3.2 Effects on the Installed Systems . . . 11
4.4 Tasks . . . 11
4.5 Migration to the New Product . . . 11
4.6 Risks . . . 12
4.6.1 Technical Risks . . . 12
4.7 Costs . . . 12
4.8 User Documentation and Training . . . 12
4.8.1 User Documentation . . . 12
4.8.2 Training . . . 12
4.9 Waiting Room . . . 12
4.10 Ideas for Solutions . . . 12

5 Appendix **13**
5.1 Gantt Chart . . . 13

List of Tables

1 **Revision History** . . . 1
2 **Definitions** . . . 3
3 **Work Partitioning** . . . 5

List of Figures

1 Work Context Diagram . . . 4

Table 1: **Revision History**

Date	Version	Notes
2022/2/9	1.0	Complete Section 1
2022/2/10	1.1	Complete Section 3
2022/2/10	1.1.1	Complete Section 4
2022/2/11	1.2	Complete Section 2 and Revise

1 Project Drivers

1.1 The Purpose of the Project

Gaming on a computer is a convenient and effective way for people's daily entertainment, especially when people are trying to relax themselves. We, as the game developers, are trying to provide a satisfying gaming experience for the users by ensuring the quality of the in-game graphic interface as well as the playability of the game itself. The game we are trying to modify is already a classic game. However, we are modifying the project with a modernized and attractive user interface and an enhanced in-game mechanic so that the game can keep track of the latest trends.

1.2 The Stakeholders

1.2.1 The Client

Project managers who is playing a role as the commissioner and the consultant of the project.

1.2.2 The Customers

End users, especially the people who are willing to play games in their spare time.

1.2.3 Other Stakeholders

developer, tester, operator

1.3 Mandated Constraints

1.3.1 Schedule Constraints

We have a project deadline on April 12th, which should be a date we finish 1 every development, improvement modification, review, test and documentations.

1.3.2 Enterprise Constraints

Since we develop the open-source project on gitlab, there is no limit on players to play the game if they have a browser with JavaScript implemented on their device.

Budget Constraints

N/A

1.4 Naming Conventions and Terminology

Table 2: **Definitions**

Terms	**Definitions**
JavaScript	A programming language used to implement core script for World Wide Web.
CSS	Cascading Style Sheets, a style sheet language used to describe presentation of the document written in a markin.
HTML	The code that is used to structure a web page and its content.
GUI	Graphic user interface
MVC	Model-view-controller
History	Player score history
Single Player Mode	One player playing the game.
Double Player Mode	Two players playing against each other.
SRS	Software Requirements Specification.
Supplement	Food or boosters eaten by the snake to change its velocity or length.
End-game Prompt	A prompt with a message "Game Over" appears after the snake dies.

1.5 Relevant Facts and Assumptions

Since we develop our game on HTML with JavaScript, we assume that our player have a basic browser with JavaScript implemented in their system.

2 Functional Requirements

2.1 The Scope of the Work and the Product

Deliverable Deadlines:

- Requirements Document Revision 0 (February 11)
- Proof of Concept Demonstration (Week of February 28)
- Test Plan Revision 0 (March 11)
- Design & Document Revision 0 (March 18)
- Revision 0 Demonstration (Week of March 21)
- Lab Exercises (Throughout Term)
- Final Demonstration (Revision 1) (Week of Apr 4)
- Peer Eval of Other Teams Final Demo (Week of Apr 4)
- Final Documentation (April 12)

2.1.1 The Context of the Work

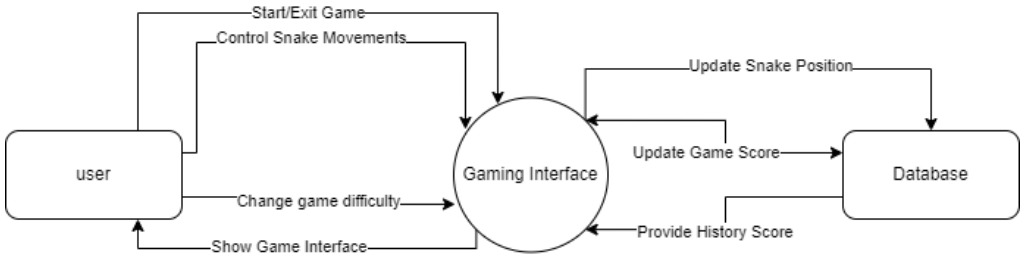

Figure 1: Work Context Diagram

2.1.2 Work Partitioning

Table 3: **Work Partitioning**

Events	**Input/Output**	**Description**
Name the player	A text name(In)	The user names the player for identifying before starting the game.
Choose a mode	Mode name(In)	The user chooses a mode (single-player mode or two-player mode) before starting a game.
View history	Click history(In) Playing history(Out)	The user views previous playing history including date, scores,name of the player
Choose difficulty	Difficulty name(In)	The user chooses the difficulty of the game(Easy, Normal,Difficult).
Change snake's direction	Up/down/left/right key(In)	The user changes the direction for the snake moving to.

2.2 Functional Requirements

2.2.1 Start the game

- Requirement number :FR1

  The user must be able to choose outlook for the snake by choosing different colour before starting the game.

  Rationale: Choosing different outlook for the snake makes the game more interesting and playable.

- Requirement number :FR2

  The user must be able to name the player by typing before starting the game.

  Rationale: Name can be used to identify the player and improve penalization, especially useful in two-player mode.

- Requirement number :FR3

  The user must be able to choose mode(single-player mode and two-player mode).

  Rationale: There are two modes in the game, one has to be chosen.

- Requirement number :FR4

  The user must be able to view previous playing history including date, scores, name of player.

  Rationale: Viewing history helps the user to review their scores and set a goal.

- Requirement number :FR5

  The user interface must be able to display the rules of the game beside the playing window, both in single-player mode and two-player mode.

  Rationale: Displaying rules of the game beside the game window efficiently helps the user know how to play the game.

- Requirement number :FR6

  The user interface must be able to display a short cartoon before starting the game.

  Rationale: A short cartoon before starting the game can catch the user's eyes and enhance fun.

- Requirement number :FR7

  The user must be able to choose the difficulty of the game(Easy, Normal, Difficult).

  Rationale: Choices for difficulty improve interestingness of the game.

2.2.2 Playing the game

- Requirement number :FR8

  The user interface must refresh the screen at the beginning of the game, updating initial snake position and a random eatable supplement position.

Rationale: This should be the starting point for a new turn of the game. We need to ensure that old gaming screen should be refreshed to a new one for this turn.

- Requirement number :FR9

  When user press direction keys on the keyboard(Up, Down, Left, Right), except for the direction for snake's body, the heading direction of the snake must be changed to the indicated direction.

  Rationale: This is the way user can control the snake's movement in the game.

- Requirement number :FR10

  The user interface must be able to refresh the gaming screen with certain frequency, updating the snake's length and position.

  Rationale: The current state and position of the snake of the game should be provided to user by refreshing the screen.

- Requirement number :FR11

  After every refreshing, it must be checked that whether the snake's head touches walls or its own body. If so, the game should be ended.

  Rationale: The basic criteria to end a game should be checked in every snake's move turn.

- Requirement number :FR12

  After every refreshing, it must be checked that whether the snake's head touches a supplement. If so, the snake's status should be updated as the rule specified.

  Rationale: This is the checking phase for the snake possible eaten supplement, as eating is an important main part of the game to raise user's goal.

2.2.3 End the game

- Requirement number :FR13

  The screen must be frozen if the game reach an end.

  Rationale: To inform a game turn's ending to player, we need to freeze the screen to prevent further in-game action.

- Requirement number :FR14

  A pop-out information prompt must be displayed to show "Game Over" information with the cumulative game score.

  Rationale: To explicitly inform player the ending of the game, a prompt is needed.

- Requirement number :FR15

  The user interface must be able to update the score in history, keep the screen frozen till the player click the new game button

  Rationale: We need to keep the record of the players score and wait for the new game command, otherwise we should not allow player to modify the game or the data.

2.2.4 During the game: snake eats a booster element

- Requirement number :FR1

  If the eaten supplement is blue, the snake is moving twice faster from now on without adding the length. 2 score is added to the cumulative score.

  Rationale: It is acted as an element in the game rule to raise games' amusement.

- Requirement number :FR2

  If the eaten supplement is red, the snake is gaining 3 length in the following 3 refresh turn. 5 score is added to the cumulative score.

  Rationale: It is also acted as an element in the game rule to raise games' amusement.

- Requirement number :FR3

  If the eaten supplement is white, the snake is gaining 1 length in the following 1 refresh turn. 1 score is added to the cumulative score.

  Rationale: It is the basic condition for the snake to get a supplement to raise the score.

3 Non-functional Requirements

3.1 Look and Feel Requirements

- The user interface should be basically clear to recognize every functionality.
- The GUI should look relaxing and amusing, some warm and light color should be mostly used.
- In the main gaming window, the distinct color should be used to inform different element in the game.

3.2 Usability and Humanity Requirements

- The instructions should be clearly declared in the user interface as guidance.
- The difficulty of the game should be reasonable for most people, like reaction speed demand and color distinguishing.

3.3 Performance Requirements

3.3.1 Capacity Requirements

The game shall not exceed the memory the machine that is running the game.

3.4 Operational and Environmental Requirements

The game should be run on different devices as long as the device has connected to the keyboard hardware and has the web browser application preinstalled.

3.5 Maintainability and Support Requirements

3.5.1 Maintainability Requirements

The code of the project should have a clear structure for the convenience of further maintains.

3.5.2 Support Requirements

Ensure most of the machines(e.g. laptops installed with different operation systems) can run the game.

3.6 Security Requirements

3.6.1 Privacy Requirements

The game shall not collect any means of user data from the machine that is running the game.

3.7 Cultural Requirements

- The game should be be offensive to ethnic or religious believes.
- The game should provide a English-based interface to the users.

3.8 Legal Requirements

- The game should not violate any laws.
- The game should follow the *Apache License 2.0 license.*

3.9 Health and Safety Requirements

The game should not contain any means of violent contents.

4 Project Issues

4.1 Open Issues

There is currently no open issues.

4.2 Off-the-Shelf Solutions

The modular components from the original SnakeGame project.

4.3 New Problems

4.3.1 Effects on the Current Environment

The change of the appearance of the GUI will provided the users a modernized look of the game. Besides, the addition of the in-game props will affect the game by increasing the play-ability.

4.3.2 Effects on the Installed Systems

The new interface is independent to the old system. In other words, they are not coexisting with each other.

4.4 Tasks

The SnakeGame project is delivering under the schedules from *SE 3XA3* outline. The planned final demonstration and documentation of the project will be delivered no later than April 12th, 2022. During the developing process, the deliverable stages include:

- Problem statement
- Development plan
- Requirements document Revision 0
- Proof of concept demonstration
- Test plan Revision 0
- Design & Document Revision 0
- Revision 0 demonstration
- Peer evaluation of other teams final demo
- Final demonstration & documentation

4.5 Migration to the New Product

The original MVC structure can be kept as basics to add new features on.

4.6 Risks

4.6.1 Technical Risks

- New programming language learning(JavaScript, Html, Css), familiar with the coding environment.
- Understanding existing GUI and making some adjustments suitable with the current GUI design.
- Testing environment for JavaScript should be chosen carefully since we lack JavaScript testing experience.

4.7 Costs

No. Because all the resources used are free.

4.8 User Documentation and Training

4.8.1 User Documentation

N/A

4.8.2 Training

A short game instructions is displayed on the user-interface, which includes snake control functions and in-game elements.

4.9 Waiting Room

Additional in-game props to enrich the game content, and user-customized features like optional background.

4.10 Ideas for Solutions

- Supporting documentations for JavaScript, HTML, and CSS.
- Current examples(open-source projects) that can be referred.

5 Appendix

5.1 Gantt Chart

Please refer *ProjectSchedule.pdf* at the same folder.

References

[1] Dr Asghar A Bokhari. *BlankProjectTemplate*. 2017.
URL: https://gitlab.cas.mcmaster.ca/bokhari/se3xa3/-/tree/fa885b8f73735515260d499c037a572bec804f29/BlankProjectTemplate.