SE 3XA3: Software Requirements Specification
DNA Says

Team 10, Team Name: DNA
Kareem Abdel Mesih (abdelk2)
John-Paul Dakran (dakranj)
Shady Nessim (nessimss)

October 10, 2016

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	**Version**	**Notes**
2016/10/10	1.0	Completion of sub-section 1 & 2
Date 2	1.1	Notes

1 Project Drivers

1.1 The Purpose of the Project

Video games have always been one of the top choices with regards to entertainment. They are also named as one of the great ways to help with boredom. This project is a redevelop of the famous game Simon Says, with a slight modification that makes DNA Says unique while keeping the integrity of the game consistent with the original version. This interactive game is going to serve the purpose of allowing people of all ages, whether bored or simply having a break, to enjoy a fun and interactive game. The main basis of Simon Says is to remember a given pattern, and iterate it back. In addition, this project will aid in the enhancement of ones visual and auditory memory.

1.2 The Stakeholders

1.2.1 The Client

The client for this project is Dr. Spencer Smith - Professor of Software Engineer 3XA3 - Software Project Management at McMaster University.

1.2.2 The Customers

The customers for this project are the general public who will operate the game DNA Says. A typical customer will be any person ranging from 5 years of age and older - who can operate and access a computer.

1.2.3 Other Stakeholders

- The Development Team - Karim, John-Paul, Shady
- General Public
- Previous and future developers

1.3 Mandated Constraints

1.3.1 Solution Constraints

Description: The game will be compatible with Windows, Mac OSX, and Linux operating systems.

Rationale: The client will be using all of the operating systems listed above.

Fit Criterion: During the user testing phase, all operating systems will be tested.

1.3.2 Partner or Collaborative Applications

This project will be a redevelopment of the game Simon Says - Python open source code available. The new game DNA Says will need to support the current games user platform.

1.3.3 Budget Constraints

The operating budget of the project is $0. All resources needed to develop this game are currently owned by the developers.

1.3.4 Scheduling Constraints

The project must be fully completed - game completed, testing finished, and documentation complete - by December 8, 2016.

1.3.5 Enterprise Constraints

This game will be free and accessible to all users who have exposure to a computer.

1.4 Naming Conventions and Terminology

Table 2: List of Terminology

Term	Definiton
Windows	An operating system.
Mac OSX	An operating system.
Linux	An operating system.
Python	A programming language.
IDLE	Integrated Development Environment
LaTeX	A document preparation system.
Mode	Different subsections of the game.
GUI	Graphical user interface.
Gantt Chart	Chart outlining the timeline of the project.

1.5 Relevant Facts and Assumptions

1.5.1 Relevant Facts

Python will be used to develop this project. It will run in its basic IDLE Version 3.5. Framework testing will be automated to test the different cases and outcomes of the game. Family and friends will test the overall functionality and performance of the game. LaTeX will be used to generate required documents.

The previous implementation of this game has approximately 250 lines of code. This implementation only has one mode, however the version we will be implementing has 3 different modes. Therefore the number of lines of code will be greater than the original version. The original implementation has no licenses that need to be acquired by the team or McMaster University.

1.5.2 Assumptions

It is assumed that the user has basic understanding of operating a computer. The user must be able to open an application and follow simple instructions to interact with the GUI.

It is also assumed that the users computer has enough processing speed and storage to effectively run and host the application.

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

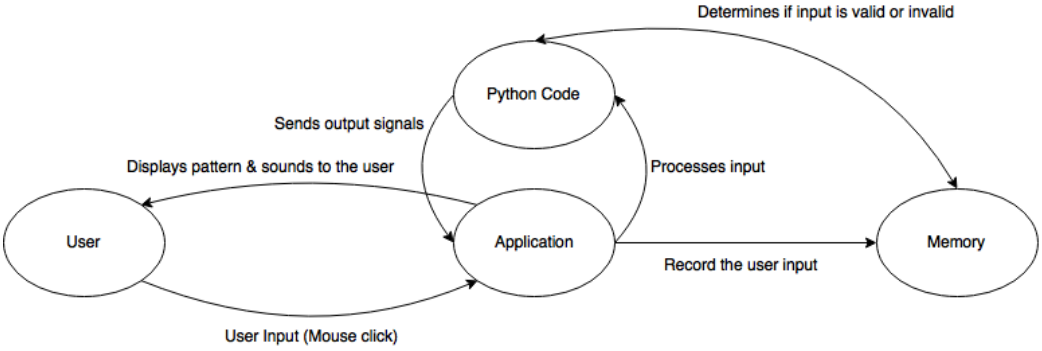

Figure 1: Context of Work Diagram

2.1.2 Work Partitioning

Table 3: List of Events

#	Event	Input	Output
1.	DNA Says Creation	Developer code	Executable file
2.	DNA Says Audio	Microphone	Audio output device
3.	DNA Says GUI	Developer code	Monitor
4.	Open the file	User input	New window
5.	Select a mode	User input	Disks appear
6.	Click a correct disk	User input	Light & sound
7.	Click an incorrect disk	User input	Sound
8.	Repeat pattern successfully	User input	New pattern
9.	Exit to main menu	User input	Main menu appears
10.	Exit game	User input	Window termination

2.1.3 Individual Product Use Cases

- Use Case #1

  - Name: Open the executable file.
  - Trigger: The user selects to open the file.
  - Precondition: The DNA Says icon must be available on the desktop.
  - Postcondition: The main menu will open.

- Use Case #2
  - Name: Select a mode.
  - Trigger: The user selects to choose 1/3 modes.
  - Precondition: The user must be in the main menu.
  - Postcondition: The user will be able to view the 4 disks and begin the game.

- Use Case #3
  - Name: Click a correct disk.
  - Trigger: The user selects a disk that was part of the pattern displayed.
  - Precondition: The user must be in a mode and the computer has displayed the pattern.
  - Postcondition: The disk will light up and make a sound.

- Use Case #4
  - Name: Click an incorrect disk.
  - Trigger: The user selects a disk that was not part of the pattern displayed.
  - Precondition: The user must be in a mode and the computer has displayed the pattern.
  - Postcondition: The disk will make a specific sound indicating an incorrect move and the correct disk will light up.

- Use Case #5
  - Name: Successfully repeat the pattern.

  - Trigger: The user selects the series of disks that composed the pattern displayed.
  - Precondition: The user must be in a mode and the computer has displayed the pattern.
  - Postcondition: The next pattern will be displayed to the user.

- Use Case #6
  - Name: Exit to the main menu.
  - Trigger: The user selects main menu icon.
  - Precondition: The user must be in a mode.
  - Postcondition: The user will leave a mode and the main menu will open.

- Use Case #7
  - Name: Exit game.
  - Trigger: The user selects the exit game icon
  - Precondition: The user must be in the main menu.
  - Postcondition: The appllicaion will be terminated

2.2 Functional Requirements

- Requirement #1
  - Description: The user will be able to open the executable file.
  - Rationale: The user must be able to open the game.
  - Fit Criterion: A new window will open on the users computer screen.

- Requirement #2
  - Description: The game interface will open in a new window.
  - Rationale: The game will be operated in a separate window.
  - Fit Criterion: A new window will appear on the users computer screen

- Requirement #3
  - Description: The game will have 3 separate modes - Kareem Says, JP Says, and Shady Says.
  - Rationale: The game is designed to have 3 distinct modes.
  - Fit Criterion: The 3 different modes will be displayed in the main menu of the game.
- Requirement #4
  - Description: The user will be able to select 1 of 3 modes to play.
  - Rationale: The user must be able to play 1 mode at a time.
  - Fit Criterion: The user will be able to select 1 of the 3 modes displayed in the main menu of the game.
- Requirement #5
  - Description: The main menu will display the 3 different modes.
  - Rationale: The user must be able to view which mode they wish to select.
  - Fit Criterion: Three distinct icons will be displayed in the main menu.
- Requirement #6
  - Description: Four different coloured disks will be shown to the user during each mode.
  - Rationale: The game is designed to have four distinct disks.
  - Fit Criterion: When a user selects a mode - the 4 disks will be displayed.
- Requirement #7
  - Description: Each disk will light up and produce a different sound when clicked.
  - Rationale: This gives the user the ability to detect the pattern that will be displayed.

  - Fit Criterion: When the user clicks a button - the disk will light up and produce a sound.

- Requirement #8
  - Description: The user will be able to exit the game at any time and go back to the main menu.
  - Rationale: The user must have a means of exiting an ongoing game and return to the main menu.
  - Fit Criterion: When the user clicks the main menu button, they will find their screen in the main menu window.

- Requirement #9
  - Description: Every time a user passes a level, the score goes up by 1 point.
  - Rationale: A record of a users score must be kept.
  - Fit Criterion: At level N, the score = N.

- Requirement #10
  - Description: Every time a user fails a level, the score is reset to 0.
  - Rationale: When a user fails a level - the game must restart from level 1.
  - Fit Criterion: Whenever the user makes a mistake, the score icon will reset to 0.

- Requirement #11
  - Description: There will be a score icon in the bottom right corner.
  - Rationale: The user must be able to view their score.
  - Fit Criterion: When the user selects a mode, the score icon will be set to 0.

- Requirement #12
  - Description: At level N, a random pattern of N disks will light up and be displayed to the user.

  - Rationale: The levels difficulty will increase as the levels progress.
  - Fit Criterion: During level 1, one random disk will light up and sound.

- Requirement #13
  - Description: The user cannot click the disks while the pattern is being displayed.
  - Rationale: The pattern must be displayed to the user in full effect.
  - Fit Criterion: The program will not record clicks the user inputs during this time.

- Requirement #14
  - Description: The user will be able to click the disks once the pattern has been displayed.
  - Rationale: The user must repeat the pattern correctly to pass the level
  - Fit Criterion: The program will monitor the users input clicks to determine if the entry is correct or not.

- Requirement #15
  - Description: A level is passed if the user repeats the pattern correctly.
  - Rationale: The user will be able to progress through the game.
  - Fit Criterion: The score will be increased by 1 when the user is successful.

- Requirement #16
  - Description: If the user fails, the game will restart - I.e. N = 1.
  - Rationale: The user must restart from the beginning of the game when a mistake is made.
  - Fit Criterion: Whenever a mistake is made, the user will be directed to level 1.

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

4.2 Off-the-Shelf Solutions

4.3 New Problems

4.4 Tasks

4.5 Migration to the New Product

4.6 Risks

4.7 Costs

4.8 User Documentation and Training

4.9 Waiting Room

4.10 Ideas for Solutions

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.