

SE 3XA3: Software Requirements Specification DNA Says

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

October 11, 2016

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	Solution Constraints	2
1.3.2	Partner or Collaborative Applications	2
1.3.3	Budget Constraints	2
1.3.4	Scheduling Constraints	2
1.3.5	Enterprise Constraints	2
1.4	Naming Conventions and Terminology	2
1.5	Relevant Facts and Assumptions	3
1.5.1	Relevant Facts	3
1.5.2	Assumptions	3
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	4
2.1.3	Individual Product Use Cases	5
2.2	Functional Requirements	6
3	Non-functional Requirements	10
3.1	Look and Feel Requirements	10
3.1.1	Appearance Requirements	10
3.1.2	Style Requirements	11
3.2	Usability and Humanity Requirements	11
3.2.1	Ease of Use Requirements	11
3.2.2	Personalization and Internationalization Requirements	12
3.2.3	Learning Requirements	13
3.2.4	Understandability and Politeness Requirements	13
3.2.5	Accessibility Requirements	14
3.3	Performance Requirements	14
3.3.1	Speed and Latency Requirements	14

3.3.2	Safety Critical Requirements	15
3.3.3	Precision of Accuracy Requirements	15
3.3.4	Reliability and Availability Requirements	15
3.3.5	Capacity Requirements	16
3.3.6	Scalability Requirements	16
3.3.7	Longevity Requirements	16
3.4	Operational and Environmental Requirements	16
3.4.1	Expected Physical Environment	16
3.4.2	Release Requirements	17
3.5	Maintainability and Support Requirements	17
3.5.1	Maintenance Requirement	17
3.5.2	Supportability Requirements	18
3.5.3	Adaptability Requirements	18
3.6	Security Requirements	18
3.6.1	Privacy Requirements	18
3.7	Cultural Requirements	18
3.8	Legal Requirements	19
4	Project Issues	19
4.1	Open Issues	19
4.2	Off-the-Shelf Solutions	19
4.3	New Problems	19
4.4	Tasks	20
4.5	Migration to the New Product	20
4.6	Risks	20
4.7	Costs	20
4.8	User Documentation and Training	20
4.9	Waiting Room	21
4.10	Ideas for Solutions	21
5	Appendix	22
5.1	Symbolic Parameters	22

List of Tables

1	Revision History	iii
2	List of Terminology	3

3	List of Events	4
---	--------------------------	---

List of Figures

1	Context of Work Diagram	4
---	-----------------------------------	---

Table 1: **Revision History**

Date	Version	Notes
2016/10/10	1.0	Completion of sub-section 1 & 2
2016/10/10	2.0	Completion of sub-section 4
2016/10/11	3.0	Completion of sub-section 3

This document describes the requirements for DNA Says. (Robertson and Robertson, 2012).

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

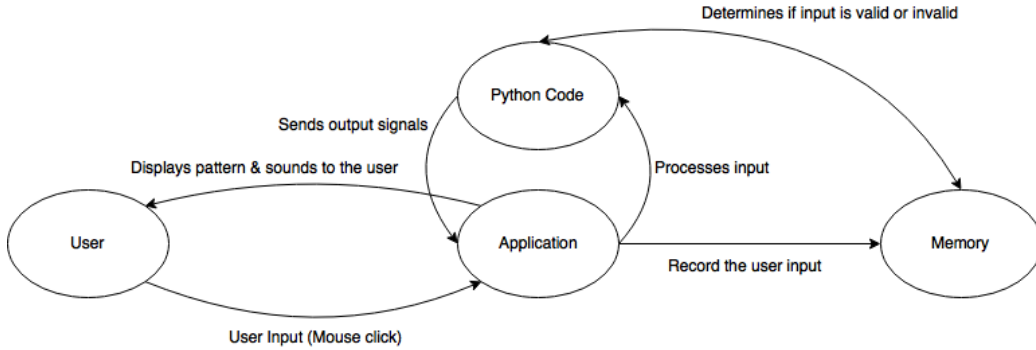


Figure 1: Context of Work Diagram

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 applicaion 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

3.1.1 Appearance Requirements

- Requirement #1
 - Description: The product shall have an appealing colorful appearance.
 - Rationale: The display should always be engaging so as to keep user interested in game. The product must be aesthetically pleasing and easy to use to benefit the end-users
 - Originator: Shady Nessim
 - Fit Criterion: Stakeholder satisfaction regarding the appearance, user attraction to game.
 - Priority: High
 - History: Created October 5, 2016
- Requirement #2
 - Description: The buttons must be well designed and colored.
 - Rationale: The game revolves around pressing buttons in a pattern. It is the main entity of the game and must thus be aesthetically pleasing to attract user interest
 - Originator: Shady Nessim
 - Fit Criterion: User reaches high levels as a result of uniqueness and beauty of buttons.
 - Priority: High
 - History: Created October 5, 2016
- Requirement #3
 - Description: The product shall have attractive sound patterns. The associated sounds with buttons must be well constructed and notes must follow harmonically.

- Rationale: The user follows a pattern based on colors and sounds, the sounds must thus be well designed to be easy to follow. When user hears an attractive pattern, naturally they are inclined to repeat it.
- Originator: Shady Nessim
- Fit Criterion: User shall be invested in game and spend a lot of time playing the game.
- Priority: High
- History: Created October 5, 2016

3.1.2 Style Requirements

- Requirement #4
 - Description: The product shall have enough buttons to keep game engaging but not too many as to make the screen feel cluttered. DNA Says will appear to be a bright upbeat game.
 - Rationale: The game must induce a style and feel to the user that will be a driving factor to use the game if the user likes the style of the game
 - Originator: Shady Nessim
 - Fit Criterion: Stakeholder satisfaction regarding the style, user attraction to game.
 - Priority: Medium
 - History: Created October 5, 2016

3.2 Usability and Humanity Requirements

3.2.1 Ease of Use Requirements

- Requirement #5
 - Description: The product shall be easy to use for people of all ages, including children.

- Rationale: The game involves no reading or writing, it does not involve intelligence either. The game involves short term memory. As such it should be easy to use for all people to improve their short term memory.
 - Originator: Shady Nessim
 - Fit Criterion: User figures out how to play the game within the first couple of minutes of use.
 - Priority: High
 - History: Created October 5, 2016
- Requirement #6
 - Description: The product shall be easy to install for all users.
 - Rationale: This product is simply a game so the user will probably not go through the trouble of downloading and installing the game if it is not an easy process
 - Originator: Shady Nessim
 - Fit Criterion: User easily downloads and installs the game in a timely manner.
 - Priority: High
 - History: Created October 5, 2016

3.2.2 Personalization and Internationalization Requirements

- Requirement #7
 - Description: The product shall operate with the English language.
 - Rationale: The application is intended for use by English and non-English speakers, however with minimal required text use, this game can easily be figured out and used by non-English speakers
 - Originator: Shady Nessim
 - Fit Criterion: User easily understands objective of game and how to play.
 - Priority: Medium
 - History: Created October 5, 2016

3.2.3 Learning Requirements

- Requirement #8
 - Description: The application shall not require a tutorial and shall be clear and simple enough in early levels to communicate to the user how the game is played.
 - Rationale: The application is intended for use by people of all ages. Must thus be easy to understand.
 - Originator: Shady Nessim
 - Fit Criterion: User easily understands objective of game and how to play.
 - Priority: Medium
 - History: Created October 5, 2016

3.2.4 Understandability and Politeness Requirements

- Requirement #9
 - Description: The application shall not produce ugly sound patterns or offensive visual patterns to respect all users
 - Rationale: The application is intended for entertainment and as a cure for boredom, if user feels uncomfortable or offended they will not use the game.
 - Originator: Shady Nessim
 - Fit Criterion: User feels good about game and patterns are appealing and attractive.
 - Priority: Medium
 - History: Created October 5, 2016
- Requirement #10
 - Description: The product shall produce friendly messages when user loses or wins a level.
 - Rationale: The application is intended for entertainment and as a cure for boredom, if user feels uncomfortable or offended they will not use the game.

- Originator: Shady Nessim
- Fit Criterion: User feels good about level progression and is encouraged to play again.
- Priority: Medium
- History: Created October 5, 2016

3.2.5 Accessibility Requirements

- Requirement #11
 - Description: The product shall produce patterns both visually and auditory so as to accommodate for users with visual or auditory problems that they can use an alternative pattern means.
 - Rationale: The application is intended all users, should be easy to use for someone by just following visual patterns or just following auditory patterns.
 - Originator: Shady Nessim
 - Fit Criterion: User with auditory or visual problems feel comfortable playing the game.
 - Priority: Medium
 - History: Created October 5, 2016

3.3 Performance Requirements

3.3.1 Speed and Latency Requirements

- Requirement #12
 - Description: The application should be able to recognize whether the user has entered the right pattern as soon as they finish pressing the last button.
 - Rationale: The user should not have to wait for the application to calculate whether their input was correct or not.
 - Originator: Shady Nessim

- Fit Criterion: Application should respond immediately to user input and the upcoming pattern should start soon after user input ends.
- Priority: High
- History: Created October 5, 2016

3.3.2 Safety Critical Requirements

There are none applicable to this project.

3.3.3 Precision of Accuracy Requirements

- Requirement #13
 - Description: The application must be specific to each button press. Button press confusion or mistake must not be tolerated
 - Rationale: The purpose of the game is to produce exact same pattern shown by application. If program does not detect a mistake even if it is just one wrong button, then that defeats the fairness and purpose of the game
 - Originator: Shady Nessim
 - Fit Criterion: Application should perceive and evaluate user pattern input impeccably.
 - Priority: High
 - History: Created October 5, 2016

3.3.4 Reliability and Availability Requirements

- Requirement #14
 - Description: The application must be available at all times
 - Rationale: The purpose of the game is to defeat boredom which may come at any time and thus the game must be available at all times
 - Originator: Shady Nessim

- Fit Criterion: User should be able to play the game whenever they are bored.
- Priority: High
- History: Created October 5, 2016

3.3.5 Capacity Requirements

- Requirement #15
 - Description: The application must be able to produce and receive patterns as long as 25 buttons
 - Rationale: In order to make the game challenging enough, patterns up to 25 in length should be produced and received by program.
 - Originator: Shady Nessim
 - Fit Criterion: User should be able to reach level 25 in each mode.
 - Priority: Medium
 - History: Created October 5, 2016

3.3.6 Scalability Requirements

There are none applicable to the project

3.3.7 Longevity Requirements

There are none applicable to the project

3.4 Operational and Environmental Requirements

3.4.1 Expected Physical Environment

- Requirement #16
 - Description: The product should be able to be used on laptops and desktops.
 - Rationale: The clients will use the product from these devices.
 - Originator: Shady Nessim

- Fit Criterion: User should be able to run the game on any laptop or desktop.
- Priority: High
- History: Created October 8, 2016

3.4.2 Release Requirements

- Requirement #17
 - Description: The product will be revised yearly and updated according to changing demands and needs of the client. The product will undergo maintenance upon realization of any errors in game-play behavior.
 - Rationale: The game has to stay updated and problems have to be handled in order to maintain user interest and usage.
 - Originator: Shady Nessim
 - Fit Criterion: App should be updated at least annually.
 - Priority: Medium
 - History: Created October 8, 2016

3.5 Maintainability and Support Requirements

3.5.1 Maintenance Requirement

- Requirement #18
 - Description: The source code for the application shall be visible to the public.
 - Rationale: This enhances the ability to monitor and maintain the system.
 - Originator: Shady Nessim
 - Fit Criterion: Source code is available in a public repository.
 - Priority: Low
 - History: Created October 10, 2016

3.5.2 Supportability Requirements

None applicable for this project

3.5.3 Adaptability Requirements

- Requirement #19
 - Description: The product shall run in Windows, Linux and Mac OS environments.
 - Rationale: The users may be using any of these operating systems.
 - Originator: Shady Nessim
 - Fit Criterion: The product works on listed platforms in the test groups.
 - Priority: Medium
 - History: Created October 10, 2016

3.6 Security Requirements

3.6.1 Privacy Requirements

- Requirement #20
 - Description: The application shall not store, transmit, or upload any user data.
 - Rationale: This is required in order to protect the privacy of users.
 - Originator: Shady Nessim
 - Fit Criterion: No functionality to perform these tasks is implemented in the application.
 - Priority: Low
 - History: Created October 10, 2016

3.7 Cultural Requirements

- Requirement #21

- Description: The application shall not contain any imagery or text that can be reasonably foreseen as potentially offensive to users of all cultures, backgrounds and ethnicities.
- Rationale: User satisfaction will be greatly reduced if they notice any offensive patterns in the game.
- Originator: Shady Nessim
- Fit Criterion: Application does not contain offensive patterns or references.
- Priority: Medium
- History: Created October 10, 2016

3.8 Legal Requirements

There are none applicable to this project

4 Project Issues

4.1 Open Issues

There are no major issues at this point of the course of the project. The only challenge that is being faced at the moment is analyzing the undocumented code, and determining the purpose of every line that is written.

4.2 Off-the-Shelf Solutions

In general, there are many games that share the similar nature and purpose. However, with respect to Simon Says, there is the original oral game in which one designated person speaks out an action for the others to do, having them only do the action if they say "Simon says" before it. As for digital versions, there exists multiple ones online.

4.3 New Problems

The only problem that could arise from this project is addiction. As there could be individuals that instead of enjoying this game during their free time or as a short break from their schedule, they would consume their other

priorities? time to play. This game could potentially be the reason behind a missed deadline, or anything in that manner.

4.4 Tasks

All tasks that need to be accomplished are covered within this group?s Gantt Chart, including their start and end dates.

- Gantt Chart

4.5 Migration to the New Product

There will not be issues transferring from another version of this game to this current version. Only the user?s preferences matter for this, and that will be discussed as a risk in the section below.

4.6 Risks

The user?s preferences are what will determine most of the success of this game. As it is designed based on the team?s preferences, it might conflict with what other individuals consider adequate. It could be that the colors used in the game are not appealing to some users. Also, the sounds that are used might not be preferred by some individuals.

4.7 Costs

There are no monetary costs included in this project.

4.8 User Documentation and Training

Instructions explaining the different modes and how the game is played will be available in the main menu which the user can access easily whenever desired.

4.9 Waiting Room

At this point, the analysis of the code will be conducted along with implementing the back-end of this project. A running prototype should be available for the Proof of Concept Demonstration.

4.10 Ideas for Solutions

To make sure that most of the users will enjoy this game, a survey should be conducted to collect different thoughts and preferences as to what the users would like to see in this game, what they are looking forward to, and what they expect. That survey should include the desired colors, sounds, interface and functionality. Later, the implementation of this project should be modified to suit those preferences.

References

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.