**CIS319 - Educational Game**

**Software Requirements Specification***Adapted from Long Software*

**Version 1.0**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 6/19/2017 |  | Project outline and specifications  Images research | Full Team  Chris.Korting, K.Williamson |
| 6/19/2017 |  | Storyboard | Chris Qashat |
| 6/19/2017 | 1.0 | Screen configurations  MazeMatics.java | Jared Wyatt |
| 6/20/2017 |  | Software Requirements Template | Keyri Williamson |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Executive Summary

This document is the Requirements Specification document for the CIS319 Case Study for an educational game. It provides detailed descriptions of the software, user, and hardware interfaces of the system, and includes a detailed description of the user interface for the system.

The intention of the game is to allow the users to practice math skills for SOL Math testing requirements in Virginia through interactive learning. Team is currently focusing on 6th grade students with intentions to scale to different grade levels in the future.

The objective of the Software Requirements Specification is to provide a summation of the findings thus far in the development stage of the project. It will be treated as a working document and provides a detailed outline of the system from the client's perspective.

# Table of Contents

1. Process Description
   1. Purpose p.4
   2. Scope p.4
2. Overall Game Description
   1. Product Perspective p.4
   2. User Characteristics p.5
   3. Constraints p.5
   4. Assumptions and Dependencies p.6
3. Specific Requirements
   1. Usability p.6
   2. Reliability p.7
   3. Online User Documentation and Help System Requirements p.8
   4. Interfaces p.8
4. Member Roles p.9
5. Card Questions

5.1 The Lens of Inner Contradiction p.10

5.2 The Lens of the Team p.10

## 1. Process Description

Team7: Creative Four custom software development services. We can handle all the developmental aspects of this case study including specification, architecture & design, implementation, testing, maintenance and training. We always involve our clients in each and every aspect of the software development process by providing prototypes and encouraging feedback. This helps ensure that our clients are highly satisfied with the end product.

### 1.1 Purpose

This document is a proposal for requirement specifications for the CIS319 Case Study and Request for Proposal posted on June 19, 2017. The Team7: Creative Four is very excited about the idea of the game and prospects of working on its development.

### 1.2 Scope

The requirements specified in this document will be used for designing all the aspects and components of the game and is a working document. The document will be updated and maintained by the project leader as the requirements grow and change over the design and development process.

## 2. Overall Game Description

### 2.1 Product Perspective

#### 2.1.2 User Interfaces

The interface for the students will be entertaining and engaging. The function of the buttons will be easy to understand and simple to use. Menus will be interactive and easily accessible throughout the game. Once the game is in playing mode, everything a player/student needs will be clearly visible on the screen and easily accessible.

#### 2.1.3 Hardware Interfaces

The product is required to operate on Windows systems.

The keyboard is required for student interaction with the game.

No answers require typing, and the character movement will be accomplished using the keyboard. Answers are multiple choice and fill in blanks with available answers. Students will engage in a maze like atmosphere by navigating the character towards the proximity of their selected answer.

The graphical content will be at most 256 colors at a resolution of 640x480. This will allow for the game to be played on older computers commonly found in Virginia Beach, VA schools. The amount of graphical content will also be limited to ensure that the total size of the game remains under the 80mb limit.

#### 2.1.4 Operations

The game will provide the following minimal operations:

* Practice SOL Math to 6th graders in an entertaining and engaging manner.
* Provide user interface and controls for the targeted audience.
* Provide difficulty levels to cater the skill level of the users.
* Utilize maze puzzles in combination to math exercise for a play role experience.

### 2.2 User Characteristics

This game is targeted directly towards children attending 6th Grade, ages ranging from 11 to 13 years. With the growth of technology, children in this age range have access to electronics, and are actively engaged with gaming and social media. Utilizing MazeMatics as a learning tool, users will enjoy the different learning environment from traditional schooling while merging technology tools they are comfortable using.

Users who lack access to personal computers at home, MazeMatics is a great way to involve them in technology use, teach them motor skills, and keyboard controls, all while learning mathematics.

Because the game is intended for in school use, teachers will be administrating the game, and must also be considered. Teachers represent a much more varied age range and background than children, but it can be assumed that they will have basic computer skills, such as word and spreadsheet processing.

### 2.3 Constraints

The following constraints are specified in the RFP:

* Platform independence is necessary since each school may have a different OS.
* Because a lot of schools are using older systems this game should run on a system with these requirements:
  + **Windows 7 and newer**: Pentium 60, 16 MB ram, 80 MB HDD space, Mouse, SVGA video card, 2x, or better, speed CD-ROM, DirectX compatible sound card.
* It will be necessary to test the game on children to ensure that it is entertaining and easy to use.
* The designers will coordinate with teachers, parents, and children. This is to ensure that the subject matter and educational material is appropriate for the students.

### 2.4 Assumptions and Dependencies

We assume the following responsibilities from the client during the game development process:

* Provide testing methods and help in developing the test cases with our test engineers.
* Upon the completion of the development process, organize meetings and workshops with the target user groups to test the software.

In order to finish the project on time, the documentation must be reviewed and signed off within one week of the delivery of our deliverables. The following are the deliverables deadlines:

|  |  |
| --- | --- |
| June 19 | Meeting to discuss RFP |
| June 21 | Provide Specification Requirements |
| June 21 | Initial Research |
| June 26 | Drawn Prototype |
| June 26 | Requirements Negotiations |
| June 28 | Divide Tasks |
| June 28 - 10 | Create MazeMatics 1.0 |
| Jul 10 | Test MazeMatics - Upgrade as needed |
| Jul 13 | Final Demo submission |

## 3. Specific Requirements

In this section, we will specify detailed requirements for the game. Our designers and programmers will design and build the game based on these requirements. Throughout section 3, requirements are indicated as functional or non-functional by the symbol (F) or (NF) respectively in the requirement heading.

### 3.1 Usability

This section addresses the software usability requirements for the students and teachers.

*3.1.1 (F) The system will test basic computer abilities prior to beginning the game, and*

*provide the necessary tutorials.*

Basic computer abilities include mouse and keyboard input, and understanding computer displays and feedback (alerts, menus etc). The system will provide a short test to determine the computer skills that the student is lacking, and will then provide tutorials in required areas.

*3.1.2 (NF) The system will provide written help instructions.*

The system is intended for use in classes, so the game will need to provide support for students that need help. A single teacher is not be able to help all students all the time. For this reason, the game is very simple to use maintaining focus in the math activities.

*3.1.3 (F) Maximum time from launching the game until it is playable will be 5 minutes.*

After the application is launched, it will take fewer than 5 minutes for the player to load their data and begin playing the game from where they last left off. This will help to ensure that children do not lose interest.

*3.1.4 (NF) Familiar user interface provided for children.*

The user interfaces for the game itself (including load screens, and in-game menus) will be similar to those found in other children’s games. “Similar” here refers to how the menu is accessed, its appearance and how it reacts to input. Menus will also use animations and colors to attract and maintain attention to important items. The game interface will be tested alongside other children’s games to ensure that the interface is usable.

#### 3.1.5 (NF) Familiar user interface provided for teachers

The interface for queries will be similar to a web-browser, with regards to navigation and file access. Navigation will use links to follow as well as forward and back buttons. Queries will be made using html-style forms and downloadable files. Responses to system querieswill be exported to Microsoft Office (i.e. Word and Excel) formatted documents, providing a familiar interface for data manipulation.

*3.1.6 (NF) Long Stretch Software will work with SME’s to ensure that the system will be*

#### usable by all parties

We will collaborate with teachers, child psychologists and other educators, as well as children to ensure that the system will be useable for people of all experience levels.

### 3.2 Reliability

*3.2.1 (NF) The software will be able to run 99% of the time when launched.*

There is a potential for errors relating to the state of the operating system that could prevent the game from launching (for example not enough resources available, etc.). The chance of such an occurrence is at most 1%.

#### 3.2.2 (NF) Expected system uptime will be 95%

The game itself will be able to run for at least 3 consecutive hours, 95% of the time. The back-end database should be able to support up to 100 connections of the game clients, 95% of the time.

#### 3.2.3 (NF) Online backup will be provided by the system

The software will employ mechanisms to ensure that there is no data loss if the game crashes.

#### 3.2.4 (F) The system will not be prone to errors caused by unexpected input

The software will be able to handle all sorts of input and be immune to side effects cause by undesirable inputs (such as buffer overflow), which could potentially create security holes in the system.

#### 3.2.5 (NF) The system will maintain network security

The software will employ appropriate network security protocols to ensure that it doesn’t create network security problems.

### 3.3 Online User Documentation and Help System Requirements

Online documentation will be available for the game upon completion and delivery. The game will also come with built in help files and tutorials for game play.

### 3.4 Interfaces (*replace example content below)*

#### 3.7.1 (F) User Interfaces

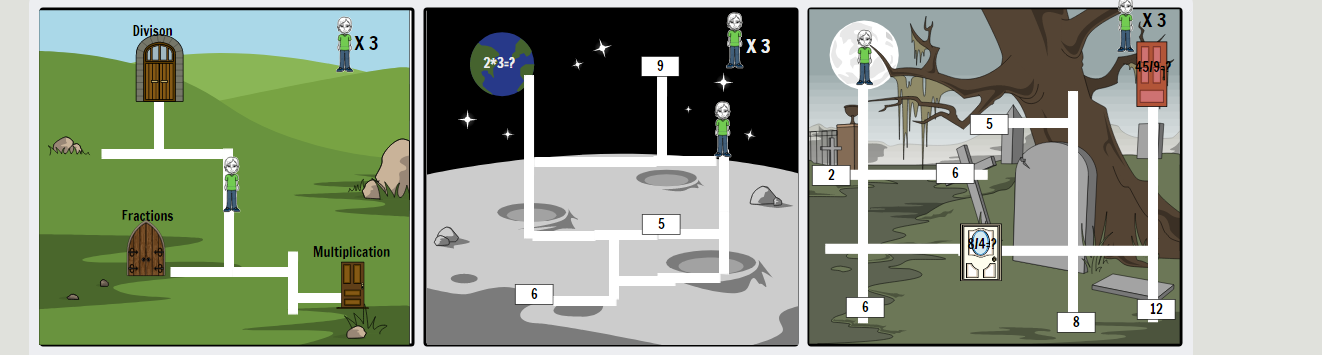
The first screen is the start screen. No login is necessary. The game is intended for in class practice. Child will type a username only to keep point scores.

#### 3.7.2 (NF) Hardware Interfaces

The gaming software does not require any additional or specialized hardware in order to operate. Existing hardware such a keyboard and mouse will be the only hardware required for input to the game.

#### 3.7.3 (NF) Software Interfaces

The gaming, administration and database components of the software will communicate securely via a local network. The school is responsible for the location and path of the file repository.



### 4 Member Roles

Keyri: Questions, Sprites, Documentation

Jared: Character Controller, Room Class, Level Layout, input manager, Window and rendering

Chris Q: Storyboard, Collision, boss fight

Chris K: Question delivery, Level floor tiling, boss fight

### 5 Card Questions

**5.1 The Lens of Inner Contradiction**

*1. What is the purpose of my game?*

The purpose of our game is expose children to math problems to keep them sharp and practiced while also playing a game that requires you to navigate through a maze to find the answers the questions.

*2. What are the purposes of each subsystem in my game?*

* Mazematics class holds the main loop.
* Game class holds the structure and layout of game objects in the world as well as the Game loop.
* CharacterController class holds properties related to the play and controls the movement of the player.
* Input class provides an easy interface for detecting key presses, key holds, and key releases, as well as the key bindings.
* Room class holds the properties and methods for creating rooms used in the maze
* LinkSide class holds the constants used for which side you want to link a room to another room
* Window class displays the game window and handles the rendering in the window
* QuestionDeliveryclass is responsible for displaying questions, generating wrong and right answers in the game world, and validating the correct answer
* Questions class holds the constants for the possible questions, wrong answers, and right answers

1. *Is there anything at all in my game that contradicts these purposes?*

Currently we have not identified any contradictions with our subsystems. We will keep this question in mind to ensure we eliminate any found contradictions.

*4. If so how can I change that?*

If we had any contradictions or when we find any we will make sure each system or class has only one role and related properties and methods. Also make sure it does not contradict other systems or do something another system or class should be doing.

**5.2 The Lens of the Team**

*1. Is this the right team for this project? Why?*

The team is right if you make it right. The biggest thing with our team is we did not know each other at all before starting this project. But we feel are team is coming together and we will make it through

*2. Is the team communicating objectively?*

When we are communicating, we are quite objective with ideas and code, Pretty much everyone is always thinking of the downsides of a certain thing or suggesting alternatives.

*3. Is the team communicating clearly?*

Our communication as whole does need work. Most of our communication has been in the class room. We have started working on ensuring more and clear communication to make sure everyone is on the same page.

*4. Is the team comfortable with each other?*

At first we were not the most comfortable as we were just meeting at that time but we were never uncomfortable. Now that we know each other much better, our comfort level has increased quite a bit.

*5. Is there an air of trust and respect amongst the team?*

Our members do trust each other and have a great working respect for each other.

*6. Is the team ultimately able to unify around decisions?*

Our team has been excellent at unifying around decisions. We are not just agreeing to agree either. Almost every suggestion made has been discussed and better alternatives suggested. The team has come together pretty well as far as decision making goes.