Software Requirements Specification (SRS)

Project: ArithMagic

Team: 4

Authors: Michael Batbouta, Liliana Byan, Justin Dom, Michael Santana

Customer:

Instructor: Professor James Daly

# Introduction

This document is intended to provide detailed information about the proposed software product to be created. This document begins with basic information about the purpose, scope, and goals of the system to be created. High level descriptions of the system’s perspective, functionality, and constraints are given. Then, specific requirements of a successful solution are enumerated. Next, this document models the proposed system though a series of diagrams. A description of the prototype follows as well as sample scenarios of using the prototype. A list of references and points of contact can be found at the end of the document

##  Purpose

The purpose of this document is to formalize the requirements of the educational game to be created. It is intended as a formal agreement between the client and the system designers about all aspects of the program.

##  Scope

The purpose of this project is to create an educational game called ArithMagic for children in grades one through four. There will be content targeted for each grade based on the education standards for each grade. The benefit of this game is that it provides students with an opportunity to learn mathematics in a different style than they’re exposed to during school. Each grade will be given a separate option which gives the benefit of changing the content based on the user’s proficiency. The objective of this project is to create something grade school children find fun to use that also teaches mathematical operations. The game is intended to be used as a tool to supplement learning, and not to replace any existing teaching methods. The goal of this project is to create an entertaining and educational game.

##  Definitions, acronyms, and abbreviations

Game refers to the product to be created, ArithMagic.

The user is the person interacting with the game.

A problem is a single mathematical equation that contains one arithmetic operation.

A level is a sequence of multiple problems.

The timer is the display of how much time the user has to answer a particular problem.

Answer refers to the solution to a particular problem.

The score is the number of correct answers the user chose before the level ended

DoE will be used as an abbreviation for the Department of Education.

Grade will be used to refer to an elementary school grade. For this document, the only relevant grades are one through four.

The difficulty of a level determines the content to be included in the level by using content appropriate for the grade.

##  Organization

The content in the rest of this document is as follows. Section 2 of this document will give a detailed description of the product including all properties, functions, constraints, assumptions, dependencies and requirements. Section 3 lists the specific requirements for an acceptable solution. Section 4 provides models and diagrams of the proposed system. Section 5 describes the prototype. Section 6 lists references used in the creation of the product. Section 7 lists points of contact.

# Overall Description

This section contains information regarding the context in which ArithMagic is to be used as well as assumptions that will be made in regards to how the product will be used, what environment it will be used in, and who will use it. This section will also cover the minimum requirements needed in order to use the product.

#  Product Perspective

ArithMagic is an offline computer video game which is designed to help and encourage students to hone their basic arithmetic skills.

 ArithMagic is designed to be used in an educational setting either in the classroom or at home. The operating system of the computer being used should have Python support. It is expected that the product will be played on either a laptop or desktop computer and support for mobile devices such as smart phones or tablets will not be implemented. It is also expected that the computer that the product is being played on will have at least 50 megabytes of space available as well as at least 32 megabytes of RAM.

#  Product Functions

Arithmagic will provide the functionality listed below:

* Practice basic arithmetic

	+ Grade 1 will practice addition
	+ Grade 2 will practice subtraction
	+ Grade 3 will consist of multiplication
	+ Grade 4 will consist of division
	+ The numbers being used for practice will be in agreement with the standards set by the Department of Education
* Teach the students how to perform the basic operations

	+ If the student is not familiar with an operation, they will be presented with a child-friendly page giving a brief explanation of the operator
	+ The page presented to the child will include the following

		- The name of the operator
		- Brief explanation of the operator
		- A real life example of the operator being used
		- Why the operator is important
		- An illustration of a scenario where the operator is relevant
* Adjust to math proficiency of the player

	+ If questions are being answered very quickly, the difficulty of the level will raise in the following ways

		- Giving larger numbers to perform operations with
		- The introduction of a time limit for each problem

The following state diagram depicts the expected course of action to be taken with ArithMagic, starting at the main menu.

 

The two main functions of Arithmagic are depicted in the following Use Case Diagram which models the product from the perspective of the student.



#  User Characteristics

The user is expected to be a child aged six to ten years old. The typical user will be in elementary school grades one through four. The user should have a general knowledge of how to use a desktop computer or laptop with a keyboard and mouse. They are expected to have vision and motor skills that are compliant with using a keyboard to type numbers and a mouse to click and move the cursor on the screen. It is assumed that the attention span of the user is less than a typical adult. An additional adult user is needed to install the product onto the devices used by a typical child user. The adult user is expected to have a general knowledge of computers and how to install programs by following given instructions to do so.

#  Constraints

ArithMagic will need at least 50 megabytes of space to be stored on the user device. Additionally, if there is not at least 32 megabytes of RAM available, then the product may not behave as intended. Additionally, ArithMagic relies on the functionality provided by the Python library Tkinter. If the computer is unable to support this functionality then the product may not work properly.

#  Assumptions and Dependencies

 Although an internet connection is not necessary to play the game, one is required to initially download the source code. It is assumed that an internet connection will be available to do so, since the product will not be distributed through other means such as USB or CD.

 It is also assumed that the computer being used to run the game satisfies the previously mentioned minimum requirements. At least 50 megabytes of storage and 32 megabytes of RAM will be required to run the game. If a lower end computer is being used and there is little storage or several other programs running at the same time, then the product may not behave as expected.

#  Apportioning of Requirements

In this version of Arithmagic, not much has been put into consideration regarding the teaching of mathematics beyond the four basic operations. Arithmagic is designed to be played by children from grades 1 to 4. That being said, Arithmagic will not be responsible for:

* Teaching subjects other than mathematics
* Teaching higher level topics such as algebra or calculus
* Keeping an online profile of each individual student
* Saving the progress of a user
* Allowing the teacher to adjust default difficulties
* Allowing the teacher to add in their own questions
* Allowing the teacher to monitor the progress of the child

 In the future, certain features may be added such as the creation of a user profile to track progress across device. This will allow students who are either transferring or using the product both at home and school to keep their progress synchronized. Additionally, adding the ability for teachers to add their own questions may be useful in several scenarios, since curriculum may vary between schools.

# Specific Requirements

1. The content in the game must adhere to the Massachusetts DoE frameworks for mathematical practice
	1. All numbers must be whole numbers
		1. The content for grade 1 must only use numbers equal to or less than 20
		2. The content for grades 2 and 3 must only use numbers equal to or less than 100
	2. The content for grades 1 and 2 must only use addition and subtraction operations
	3. The content for grades 3 and 4 may contain addition, subtraction, multiplication and division operations
2. After the user submits their answer for a problem, the game must give the user feedback about the correctness of their answer.
3. No problems should be displayed after the timer has ended.
4. The product should meet the software specifications described below
	1. The product should not be more than 50 megabytes of space
	2. The product should not use more 32 megabytes of RAM
	3. The product should not require the use of the internet to operate

# Modeling Requirements

|  |  |
| --- | --- |
| Use Case Name:  | Play Game |
| Actors:  |  User (initiator) |
| Description:  | The user opens the game and wants to play. |
| Type:  | Primary |
| Includes:  | None |
| Extends: | None |
| Cross-refs:  | Requirement: 1 |
| Uses cases:  | None |

The diagram for this use case is below.



|  |  |
| --- | --- |
| Use Case Name:  | Play Level |
| Actors:  |  User |
| Description:  | The user has started a game and wants to complete a level. The system must display the question so that the user can enter their answer. Once the question is displayed, the timer should start. |
| Type:  | Primary |
| Includes:  | Display question includes starting the timer |
| Extends: | None |
| Cross-refs:  | Requirements: 1, 2, 3 |
| Uses cases:  | None |

This diagram for this use case is below





State Diagram



Class Diagram

# Prototype

* Select game level by grade
	+ different levels have problems and challenges to teach users
* Select about section detailing the developers and the goal of the project
* Scores change on successful levels
* Select exit game

#  How to Run Prototype

1. Download the game from the Game GitHub Repository
2. To run prototype Python must be installed. For windows 10 installation Python can be installed from the Microsoft store. For Linux installations use the appropriate package manager to install python (e.g. *sudo apt-get install python* or *sudo yum install python*)
3. **Windows**: You can open file explorer, navigate to the game directory and double click on Game.py. This will start the game. Alternatively, open the command prompt, navigate to the game directory and run *python Game.py*.

**Linux**: Open the command line, navigate to the game directory and run *python Game.py*.

#  Sample Scenarios

A grade school student needs to practice math problems. From the menu page, he/she will select 1st grade. From there a question will appear with multiple answers listed below. Once the user selects the answer if they are correct their score will raise by one and if they are incorrect they won’t change score. Once the time has run out the user will see his final score displayed and be returned to the main menu to either play again or quit the program.

Below will be the menu screen displayed:



Once the grade is selected they will be asked a question like the one below.

# References

1. Project website: <https://maickii.github.io/educational_game/>
2. Massachusetts Department of Education Frameworks: <http://www.doe.mass.edu/frameworks/current.html?section=math-qrg>
3. Game GitHub Repository: <https://github.com/Maickii/educational_game>

# Point of Contact

For further information regarding this document and project, please contact **Prof. Daly** at University of Massachusetts Lowell (james\_daly at uml.edu). All materials in this document have been sanitized for proprietary data. The students and the instructor gratefully acknowledge the participation of our industrial collaborators.