THE RUNNERS 2021

PRESENTED BY

The Runners

WORK WITH

GitHub

REPOSITORY

Maze – 2021

Content

The Runners Maze Game 2
Information and Development of a plan 2
Roles in the team 2
Introduction 2
Method and Manner of implementation 3
Code plan 3
Plan testing 4
All tasks to perform 4
Rules of the Game 4
Table for Functions and Variables 5
Rules of the Game 5
The Runners Maze Game - DIagrams 6
Description of files 6
Description of the block diagram 6
Reaching and playing the game 7
Description of the block diagram 7
Moving in functions in the Application 8
Description of the block diagram 8
Maze Algorithm 9
Description of the block diagram 9

THE RUNNERS MAZE GAME

INFORMATION AND DEVELOPMENT OF A PLAN

PRESENTATION OF THE PROJECT

Our project is a C++ application for Maze Game, which you can download in our Website.

Link - https://therunnersmazegamesite.z6.web.core.windows.net/

ROLES IN THE TEAM

№	Roles in the team
1	Nikolay Brankov – Scrum Trainer
2	Veselin Stoyanov – Front End Developer
3	Kostadin Taligadzhiev – QA Engineer
4	Mario Berberov – Back End Developer

INTRODUCTION

№	Introduction
1	What is the product? The product is a C++ application. We have Login and Registration systems, where you can input your data. You can choose between three different levels of difficulty Easy, Medium and Hard. If you want you can download the app form our Website. **Link** - https://therunnersmazegamesite.z6.web.core.windows.net/
2	Where is it available? Our collaborative work took place in **GitHub** and in order for the files to be accessible to everyone they were uploaded in the GitHub Repository of our project. **Link** - https://github.com/KNTaligadzhiev19/Maze-Project-2021.git
3	Communication? Communication is realized through **Teams**. Thanks to all the features and the provided visualization - on-screen communication and feedback are sufficiently complete.
4	What technologies are used? The technologies used are **Visual Studio Code** as Code Editor, **C++** as a programming language and **HTML**, **CSS** & **JavaScript** only for the Website, **SFML** for graphic representation in the console, Visual Studio as IDE for creating C++ application, **Git** and **GitHub** for collaborative work. We use **Azure** to deploy and monitor our website. **Teams** - connection and communication, **PowerPoint** - preparing a Presentation, **Word** - preparing Documentation.

METHOD AND MANNER OF IMPLEMENTATION

№	Methods and manner of Implementation
1	Productive work The tasks are defined in a way that everyone is aware of the tasks performed so far in order to present and answer quickly, clearly and accurately, and teamwork is more efficient and productive.
2	Distribution of tasks For each task, a person is selected who is more familiar with the field and will be able to perform the task in the most competent way possible.
3	Terms Observance of dates was reminded by the Scrum Trainer. A meeting of the team is held every week to discuss the amount of time needed to complete the assigned task.

CODE PLAN

№	Plan for realization
1	Systems You have to be registered in our system to play. The registration is free and you have to enter only your email, your username will be the letters before '@' and your password. Your password is secret because we have hashing system.
2	Algorithm We had to create a random generated algorithm for players, only one maze wasn't good idea, so created new maze every time you want to play
3	Levels in Game The levels in game are three – Easy, Medium and Hard. You don't have to finish easy level to enter Medium, you just click on Level you want to play and the Maze will show
4	SFML Graphics introduction is one of the most important part of the project. We created buttons, Textboxes, Tables, Blocks illustration of heroes you play and goal, also when you arrive the end, "Congratulation" sign pop up in your console.
5	Website The website is for people who want to download the application and enjoy our game. General colors are used in site are teal, light teal and dark teal, these colors are also used in our application, documentation and other forms.

PLAN TESTING

ALL TASKS TO PERFORM

№	Completed tasks
1	Systems The system look and design are made by Front End, hashing and textboxes are made by Back End
2	Algorithm The plan for the Algorithm was invented by everyone on the team, the implementation and execution was carried out by Back End
3	Levels in Game Levels illustration and functionality on console are made by Front End & Back End using the Maze Algorithm, the size of the blocks is in accordance with the size of the console
4	SFML All SFML images, representation, placement were job to our Front End
5	Website Website, Documentation, Presentation, test cases in Word and Excel were made by QA Engineer and Scrum Trainer

RULES OF THE GAME

№	Regulation
1	Position and Goal Your Position is always an animal, your goal is food. Provided animals are rabbit, hedgehog and bee with their goals carrots, apple and honey
2	Buttons for playing You can play with buttons for left, down, right and up, or use the letters ‘A’, ‘S’, ‘D’, ‘W’ – regardless if it is Uppercase or Lowercase
3	Moving in the Maze The only way you can move your hero is a teal color way, there are blocks you can’t move there
4	Finishing the game The only way when you are in the game to go back is to win. When you win Congratulation message pops up in your console, there you can choose to go back if you want to play again.

TABLE FOR FUNCTIONS AND VARIABLES

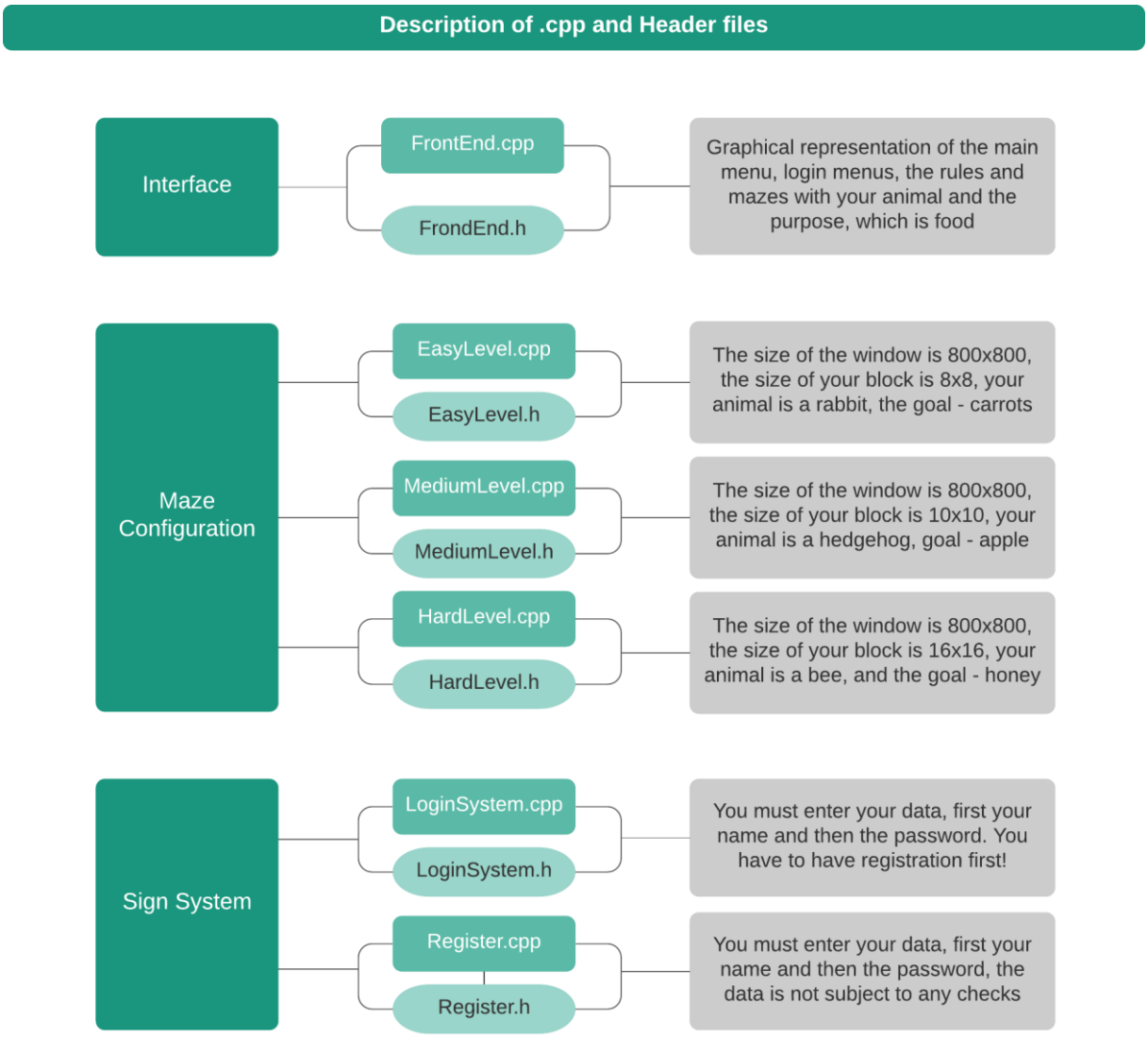
NAME	TYPE	DESCRIPTION
setDataValue()	Int Function	Giving the option to the user to choose
checkValues	namespace	Page from which you download the app
loadFromFile()	method	Setting the default pictures to print
chooseSize()	Int Function	Set difficulty function
movePixelSize()	Function	Set maze pixels
size	Const variable	Setting the size for the maze
randomNumber	Int variable	Random wall to be removed
xTrack, yTrack	Int vector	Used for having the reversed path
checkLoginData()	Function	Checking for correct data
Cell	Structure	Structure for each cell
Positions	enum	Symbols, and coordinates
level	Variable	Integer as a counter for setting what difficulty of level player will play
checkIfUsernameOrPasswordAreEmpty()	Bool Function	Checks for correct data
initializeDataIntoIndividualFile()	Void function	Pushing user data into a file
t1	Texture	A texture is an image. being mapped to a 2D entity.
GoBack	Sprite	A sprite is nothing more than a textured rectangle.

RULES OF THE GAME

All these functions are described in the code by the hacker code. The table is indicative, but you can extract enough information. In the first column you can see the name of the object itself then in the middle to which column they belong - functions, methods or variables, arrays, structures and classes as functions and variables you can see what type they are, the following types we use in our application are - int, char, float, also sprite texture and text box, which are part of the library of SFML. In the right column we have a brief description of what action it performs, you can possibly follow in our code how.

THE RUNNERS MAZE GAME- DIAGRAMS

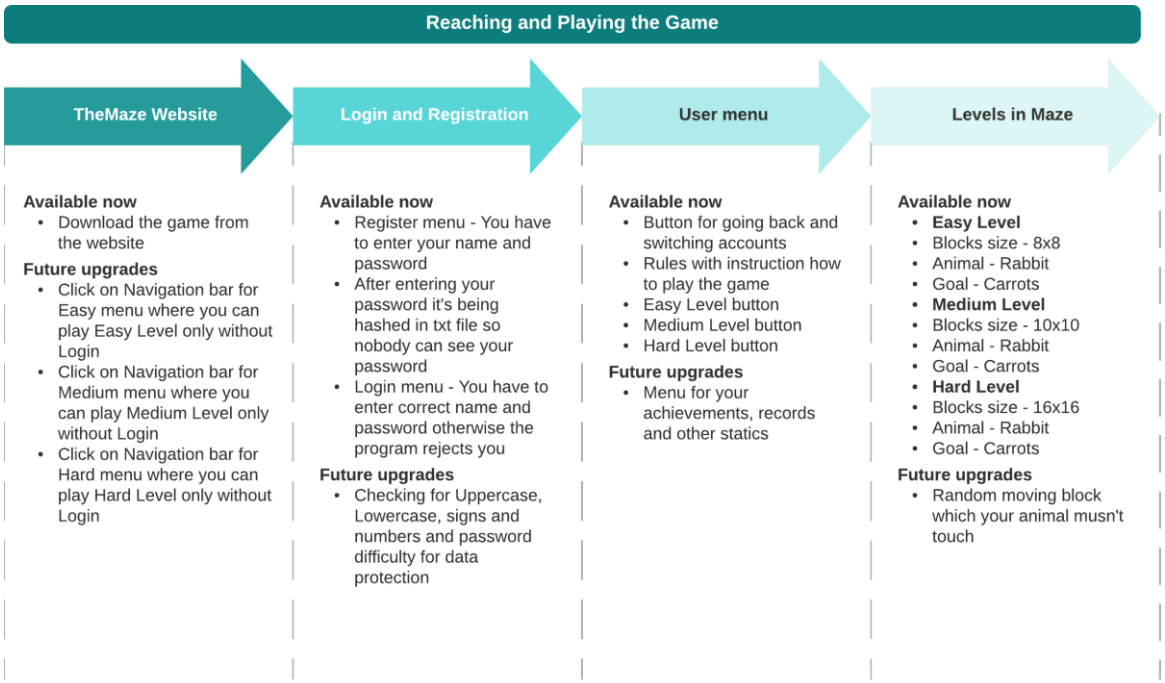
DESCRIPTION OF FILES



DESCRIPTION OF THE BLOCK DIAGRAM

№	Color	Meaning
2	Dark Teal	Folder
4	Teal	.cpp file
5	Light Teal	Header file

REACHING AND PLAYING THE GAME

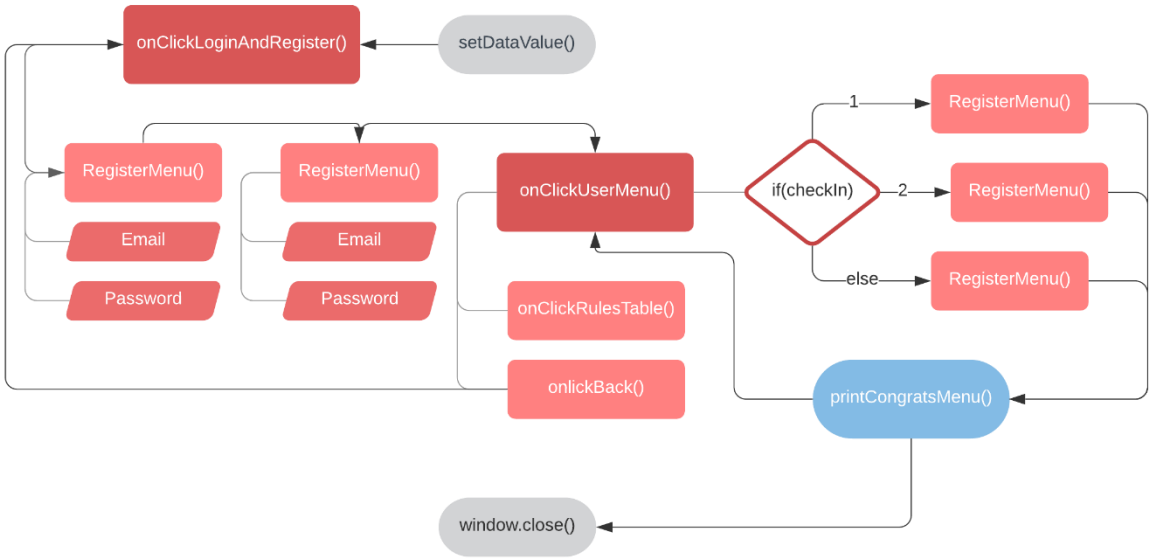


DESCRIPTION OF THE BLOCK DIAGRAM

№	Color	Meaning
2	Teal	Website
4	Light Teal	Login & Register
5	Light Teal	User menu
	Pale Teal	Maze Game

MOVING IN FUNCTIONS IN THE APPLICATION

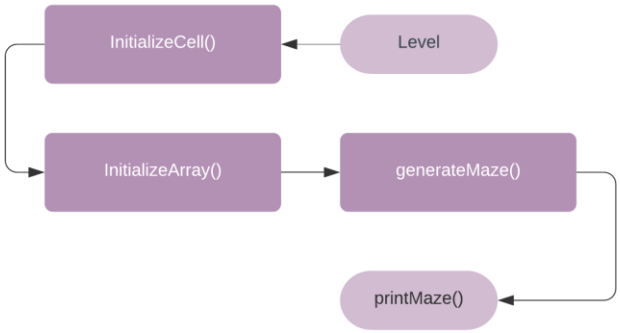
How to move in Functions in the Application



DESCRIPTION OF THE BLOCK DIAGRAM

№	Color	Meaning
2	Grey	Function for End or Start
5	Red	Main Function
	Light Red	Content
	Pale Red	Function
	Light Blue	Finishing Function
	White, Red Border	Check Result

How does the Maze Algorithm work



DESCRIPTION OF THE BLOCK DIAGRAM

№	Color	Meaning
2	Teal Purple	Function for End or Start
5	Light Purple	Function