2021

TEAM ANTI-XENON

A picture containing text

Description automatically generated

Content

[**1. Authors** 2](#_Toc57670242)

[**2. Summary** 2](#_Toc57670243)

[2.1. Goals of the project 2](#_Toc57670244)

[2.2. Stages of realization 2](#_Toc57670245)

[2.3. Difficulty level and obstacles 2](#_Toc57670246)

[**3. Technologies we used** 2](#_Toc57670248)

[**4. Description of the functions** 2](#_Toc57670249)

# **1. Team members**

* Georgi Milev – 10G class

- Scrum Trainer

[-GNMilev19@codingburgas.bg](mailto:-VPPeev19@codingburgas.bg)

* Vasil Mladenov – 10V class
* Back-end Developer

[-VLMladenov19@codingburgas.bg](mailto:-VLMladenov19@codingburgas.bg)

* Stanimir Dimitrov – 10B class
* QA Developer

[-SMDimitrov19@codingburgas.bg](mailto:-SMDimitrov19@codingburgas.bg)

* Zhivko Nanev – 10A class

- Frond-end Developer

[-ZDNan](mailto:-AGAndreev19@codingburgas.bg)ev19@codingburgas.bg

# **2. Summary**

## 2.1. Goals of the project.

Our goal, as set in the regulation of the project activity, was to make a maze game. We had ideas, we gathered each time, we gave each other who, what to do, and so almost seamlessly created the game in its current form.

## 2.2. Stages of realization.

In general, we tried not to do things chaotically, and everyone knows what to do, and if someone needs help, the others helped, and so in the end we managed, dare I say, to realize the project, after a lot of discussions, code changes, etc.

## 2.4. Difficulty level and obstacles.

We dare say that the project went almost seamlessly, saying "almost", I mean that the problems we had were not an obstacle to the realization of the code, but were inferior problems that immediately after their discovery we solved them. At first we were a little less active, but at one point we told ourselves that we have to start doing it, and after a period of time, the code has been implemented.

# **3. Technologies we used**

## 3.1. Discord- for communication

## 3.2. GitHub- for organization of the project

## 

## 3.3. Visual Studio- for code writing



## 3.4. Word- for documentation.



# **4. Description of the functions we used**

|  |  |
| --- | --- |
| **Functions** | **Description** |
| menu(string arrow, int arrowPos) | *Displays the Main menu.* |
| options(string arrow, int arrowPos) | *Prints the Options menu.* |
| colorMenu(string arrow, int arrowPos) | *Prints the Color Menu.* |
| chooseColor(string arrow, int arrowPos) | *Used to changes game’s color.* |
| characterMenu(string arrow, int arrowPos) | *Prints the Character Menu.* |
| chooseCharacter(string arrow, int arrowPos, char\* player) | *Used to change player’s symbol.* |
| chooseOptions(string arrow, int arrowPos, char\* player) | *Used to choose options to edit.* |
| difficulty(string arrow, int apropos) | *Prints the Difficulty menu.* |
| chooseDifficulty(string arrow, int arrowPos, char\* size) | *Used to choose game’s difficulty.* |
| chooseMenu(string arrow, int arrowPos, char\* size, char\* player) | *Used to choose whether to start playing or change the options.* |
| printMaze(CELL\*\* maze, int size, char free, char player) | *Print the maze.* |
| createWalls(CELL\*\* maze, int size, int\* cellCount) | *Makes all cells to walls so that we can generate the path later.* |
| toVisited(CELL\*\* maze, int\* cordY, int\* cordX, int dir, int\* unvisitedCells) | *Checks if the walls are unvisited and generates the path from the start point to the end point.* |
| playerMovement(CELL\*\* maze) | *Moves our player character, while in the same time checking if the movement is possible.* |
| freeCheck(int dir, int size, int cordY, int cordX) | *Check if the cells near the given coordinates are maze borders.* |
| winningText() | *It displays a winning text when the player wins the game.* |