****

**Contents**

[Project description 3](#_Toc130591850)

[Team information 3](#_Toc130591851)

[Project information 3](#_Toc130591852)

[Technologies used 4](#_Toc130591853)

[Ways of Realization 5](#_Toc130591854)

[Work Plan 6](#_Toc130591855)

# **Project description**

* A game about bitwise operations, where you push logic gates to complete puzzels
* The project was built by using “raylib” and C++

# **Team information**

|  |  |  |
| --- | --- | --- |
| № | Name | Role |
| 1 | Ivelin Bozhilov | Scrum trainer |
| 2 | Ivan Stoychev | Back-end Developer |
| 3 | Stoyan Ivanov | Front-end Developer |
| 4 | Stoyan Skuliev | QA engineer |

# **Project information**

|  |  |
| --- | --- |
| № | Information |
| 1 | **Description**  A game about bitwise operations, where you push logic gates to complete puzzels |
| 2 | **Installation**  To install the project, you can open our GitHub repository and follow the instruction in the README.md file. |
| 3 | **Communication**  For communicate we used Teams. Which made it easy to share file and text messages. |

# **Technologies used**

|  |  |  |
| --- | --- | --- |
| 1 | Technologies | Usage |
| 1 | Visual Studio 2022 | As out IDE |
| 2 | Visual Studio Code 2022 | As out IDE / Text editor |
| 3 | GitHub and Git | For collaboration |
| 4 | C++ | As programming language |
| 6 | Raylib | As a graphical library |
| 7 | Rayguy | As ingame UI |
| 8 | Premake-core | As a build system |
| 9 | Python | For startup script |
| 10 | Word | For the documentation |
| 11 | Doxygen | For the code documentation |
| 12 | PowerPoint | For the presentation |
| 13 | Excel | For the QA documentation |
| 14 | Teams | For communication |
| 15 | Figma | For the UI / UX design |
| 16 | Aseprite | For the sprite design |
| 17 | SFXR | For the sound design |
| 18 | Netlify | To host the doxygen docs |

# **Ways of Realization**

|  |  |  |
| --- | --- | --- |
| № | Issue | Solution |
| 1 | Task Distribution | When we distributed the tasks, we took in consideration the skills of each member and where he could be most productive.  For each task we made a GitHub issue which helped us to stay in track and made it easy for each member to see his tasks. |
| 2 | Task Completion | There were team meetings almost every day where we discussed problem and the overall state of the project.  Each member worked in a convenient for him time. When he was ready with his part of the project, he committed it to GitHub and closed the respective issue. This made it easy to track the progress of the project. |
| 3 | Deadlines | In our GitHub repository there were also milestones for each of the four weeks.  We split out issues based on how hard they are and how necessary they are for the project. |

# Work Plan

|  |  |
| --- | --- |
| № | Task Description |
| 1 | **Add player**  The player was added by our Backend developer and tested by our QA engineer. He moves around the grid while pushing the gates to form a solution to the puzzle |
| 2 | **Add logic gates** The logic gates were added by our Backend developer and tested by our QA engineer. The magic gates are basic building blocks of digital circuits that perform a logical operation on one or more binary inputs to produce a single binary output |
| 3 | **Add lamps** The lamps were added by our Backend developer and tested by our QA engineer, and they represent 1s and 0s |
| 4 | **Add levels** The levels were added by our Backend developer and tested by our QA engineer. |
| 5 | **Add shaders** The shaders were added by our Backend developer and tested by our QA engineer. The shaders add different post-processing effects through the game |
| 6 | **Add sprites** The sprites were added by our Frontend developer and tested by our QA engineer. The sprites are the art of the whole project |
| 7 | **Add sound effects** The sound effects were added by our Backend developer and tested by our QA engineer. Add sound effects for the UI, the player and the player’s interaction with the environment |
| 8 | **Add main menu** The main menu was added by our Frontend developer and tested by our QA engineer. The main menu offers the option to play the game or quit. It’s the first thing you see when you open the game |
| 9 | **Create Doxygen documentation** The Doxygen documentation was created by our QA engineer, and it shows in detail how the code works |
| 10 | **Create QA documentation** The documentation was created by our QA engineer |
| 11 | **Create documentation** The documentation was created by our Scrum Trainer to show of precess of work in our team |
| 12 | **Create presentation** The presentation was created by our Scrum Trainer |