

**BAKLAVA**



Contents

[Project Description 2](#_Toc121762563)

[Team Information 2](#_Toc121762564)

[Project Information 2](#_Toc121762565)

[Technologies used 2](#_Toc121762566)

[Ways of Realization 3](#_Toc121762567)

[Work Plan 4](#_Toc121762568)

# Project Description

* *A tower defense where to protect your tower you need to solve mathematical equations.*
* *The project was build using “raylib” and C++.*

# Team Information

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

# Project Information

|  |  |
| --- | --- |
| № | Information |
| 1 | **Description**  The project is an unusual take on tower defense where to defend your tower you need to solve mathematical equations. |
| 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

|  |  |  |
| --- | --- | --- |
| № | Technologies | Usage |
| 1 | Visual Studio 2022 | As out IDE |
| 2 | GitHub and Git | For collaboration |
| 3 | C++ | As programming language |
| 4 | Raylib | As a graphical library |
| 5 | Word | For the documentation |
| 6 | PowerPoint | For the presentation |
| 7 | Excel | For the QA documentation |
| 8 | Teams | For communication |
| 9 | Doxygen | For the code documentation |
| 10 | Adobe XD | For the UI and UX |
| 11 | PhotoShop | For the creation of the logo |
| 12 | 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 | **Implementing the tower**  The tower was added by our Backend developer and tested by our QA engineer. The tower is the main objective which must be protected. |
| 2 | **Add enemies**  The enemies were added by our Backend developer and tested by our QA engineer. The enemies have for goal to destroy the tower. |
| 3 | **Add wave system**  The wave system was added by our Backend developer and tested by our QA engineer. The wave system adds delay between the different waves and calculates the health and damage base on the current wave. |
| 4 | **Add equations**  The equations were added by our Backend developer and tested by our QA engineer. The equation is generated which the player must solve to kill an enemy and respectively protect his tower. |
| 5 | **Add currency and score**  The currency and score were added by our Backend developer. The purpose of the gold is to be spend for upgrade which make the game easier. And the purpose of the score is to motive the player to solve the equation and not just skip them till he finds an easier to solve. |
| 6 | **Add upgrade system**  The upgrade system was added by our Backend developer and tested by our QA engineer. The upgrade system is used to buy upgrade for the tower and to make the game easier. |
| 7 | **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 and is the first menu you see when you open the game. |
| 8 | **Add win screen**  The win screen was added by our Frontend developer and tested by our QA engineer. When you complete all 10 waves you are presented by the win screen which shows your score and asked you if you want to close the game or play again. |
| 9 | **Add death screen**  The death screen was added by our Frontend developer and tested by our QA engineer. When the tower health reaches 0 you are presented by a death screen which show your score and ask you if you wan to play again or quit the game. |
| 10 | **Add upgrade menu UI**  The upgrade menu UI was added by our Frontend developer and tested by our QA engineer. The UI screen allows the player to choose what upgrades to buy. |
| 11 | **Create Doxygen documentation**  The Doxygen documentation was created by our QA engineer. And it shows in detail how the code works |
| 12 | **Create QA documentation**  The QA documentation was created by our QA engineer. |
| 13 | **Create Documentation**  The documentation was created by our Scrum trainer. To show the process of work in our team. |
| 14 | **Create Presentation**  The Presentation was created by our Scrum Trainer. |