SHEMA



Table of Contents

**Adventurous geographical game** 3

[Project information and plan](#Project_Information) 3

[Team information](#Team_Information) 3

[Introduction](#Introduction) 3

[Realization](#Ways_of_Realization) 4

**Work plan** 5

[Tasks for completion](#Task_for_Completion) 5

Adventurous geographical game

**Project Information**

* We created adventurous geographical game using “raylib” and C++, over the span of a month.

**Team Information**

|  |  |
| --- | --- |
| № | Roles in the team |
| 1 | Maxim Marinov – Scrum Trainer |
| 2 | Stefan Hristov – Backend Developer |
| 3 | Emil Bogdanov – Backend Developer |
| 4 | Hakan Chandar – QA Engineer |

**Introduction**

|  |  |
| --- | --- |
| № | Introduction |
| 1 | **What is the product?**  The product is an educational traveling game. We used C++ and raylib. |
| 2 | **How can you access it?**  You can read about our collaborative work on GitHub and access our project’s repository files. |
| 3 | **What about communication?**  We communicated through Teams due to its helpful functions like screen sharing and text channels. The team was well connected and the work was efficient. |
| 4 | **What programs were used?**  We used GitHub for file management and collaborative work, Visual Studio 2022 for code editing, MS Teams for communication, MS PowerPoint for our presentation, MS Word for the documentation, MS Excel for our QA documentation, Doxygen for our code documentation and Figma/Photoshop for our design. |

**Ways of Realization**

|  |  |
| --- | --- |
| № | How did we do it? |
| 1 | **Task Distribution**  The Tasks were distributed based on the skillset of everyone. We also notified each other when a commit was made so everyone can stay up to date with the collaborative work. This way our team was as productive as possible. |
| 2 | **Task Completion**  Every two/three days we held a meeting to track the development of our application and help each other progress further. We also resolved issues/problems and discussed new ideas for improvement. |
| 3 | **Deadlines**  In these meetings we also discussed time management, how specific parts were coming along, what everyone had done in their specified time and what things should be completed in the near future. |

WORK PLAN

**Tasks for Completion**

|  |  |
| --- | --- |
| № | Task Breakdown |
| 1 | **Setting up raylib library**  The raylib library was set up by our Scrum Trainer. This is the main library that is being used in the project. |
| 2 | **Creating main menu and its options.**  Our main menu was created by our Backend developers. It is used to navigate through the application. |
| 3 | **Creating the “choose country” functionality.**  This functionality was created by our Backend developers. When a new game is started the player has to choose a starting country. |
| 4 | **Creating the player movement.**  This functionality was created by our Backend developers. With this functionality the player is allowed to travel to other countries. |
| 5 | **Creating the in-game currency.**  The in-game currency was created by our Backend developers. You can spend it on unlocking different countries. |
| 6 | **Creating country travelling.**  The card functionality was implemented by our Backend developers. A card can be placed if it corresponds to a valid Boolean operation. |
| 7 | **Creating player quests.**  Player quests were created by our Backend developers. By completing player quests the player can earn money. |
| 8 | **Creating the Doxygen documentation**  The Doxygen documentation was created by our QA Developer and Scrum trainer. You can see in great detail how the code works through this documentation. |
| 9 | **Creating the QA Documentation**  The QA Documentation was created by our QA Engineer. There are unit test reports about the application. |
| 10 | **Documentation**  Our QA Engineer and our Scrum trainer created the documentation using MS Word to summarise the application. |
| 11 | **Presentation**  The presentation was created by our Scrum trainer to explain the concept of the application. |