

The challengers



Table of Contents

[1.Information about our team 3](#_Toc95680139)

[2.Recap 3](#_Toc95680140)

[2.1 Main goal 3](#_Toc95680141)

[2.2 Tasks 3](#_Toc95680142)

[2.3 Realisation (apps and language that we used) 4](#_Toc95680143)

[1.Apps,that we used: 4](#_Toc95680144)

[2.Programming language that we used: 4](#_Toc95680145)

[3.Srtucture 4](#_Toc95680146)

[3.1 Structure of the files 4](#_Toc95680147)

[3.2 Structure of the functions 5](#_Toc95680148)

[3.3 Block scheme 7](#_Toc95680149)

[4. Summary 7](#_Toc95680150)

# 1.Information about our team

Kameliya Yaneva 10g – Scrum Trainer

Mirena Dzhebarova 10b – Front-end Developer

Iven Staev 10v – Back-end Developer

Dimitar Berdankov 10a – QA Engineer

# 2.Recap

# 2.1 Main goal

Our main goal is to make a c++ application that includes Gray code and historical events

# 2.2 Tasks

1. We assigned roles.
2. We collected information about the task.
3. We made the menu
4. We wrote the code.
5. We made historical application with Gray code.
6. We made presentation, documentation and README file.
7. We made QA documentation.

# 2.3 Realisation (apps and language that we used)

# 1.Apps,that we used:

* **Teams** and **GitHub** for communication and team synchronization.
* **Visual Studio** for writing the code.
* **Photoshop** for making the logo.
* **PowerPoint** and **Word** for the making of the presentation and documentation.
* **Excel** for the making of the QA documentation.

# 2.Programming language that we used:

* **C++**

3.Srtucture

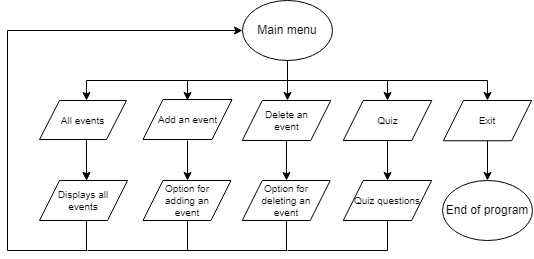
# 3.1 Structure of the files

|  |
| --- |
| Name of the file |
| main.cpp |
| Frontend.cpp |
| Backend.cpp |
| LinkedList.cpp |
| LinkedListFunctions.cpp |
| Quiz.cpp |
| Backend.h |
| Frontend.h |
| LinkedList.h |
| LinkedListFunctions.h |
| Quiz.h |

# 3.2 Structure of the functions

|  |  |
| --- | --- |
| № | Name of the function |
| 1. | void Quiz::setQuestion(string question) |
| 2. | void Quiz::setAnswer(string answer) |
| 3. | Void Quiz::setIncorrectAnswer(vector<string>incorrectAnswer) |
| 4. | String Quiz::getQuestion() |
| 5. | String Quiz::getAnswer() |
| 6. | Vector<string>Quiz::getIncorrectAnswer() |
| 7. | String swapStr(string v); |
| 8. | String decToBin(string dec); |
| 9. | String decToGray(string dec); |
| 10. | Node\* initializeDefaultUprisings(); |
| 11. | Bool checkIfValidYear(string year); |
| 12. | Void addEvent(Node\*\* head); |
| 13. | Void displayDeleteMenu(Node\* head,int option); |
| 14. | Bool checkIfValidUserImput(string userImput); |
| 15. | Void deleteNode(Node\* head,int index); |
| 16. | Void deleteFirstNode(Node\*\* head); |
| 17. | Bool deleteFirstNode(Node\* head,int index); |
| 18. | void deleteFirstNode(Node\*\* head); |
| 19. | bool deleteAnEvent(Node\*\* head); |
| 20. | vector<Quiz> initializeQuiz(); |
| 21 | void displayQuestion(Quiz quiz, int option, int n); |
| 22. | bool submitAnswer(int option, int r); |
| 23. | vector<Quiz> randomizeQuiz(vector<Quiz> quiz); |
| 24. | bool runQuiz(); |
| 25. | bool runProgram(); |
| 26. | void printMenuHeading(int option); |
| 27. | void printMenuOptions(int option = 1, string arrow = "=>"); |
| 28. | void allEventsHeading(); |
| 29. | void addAnEventHeading(); |
| 30. | void deleteAnEventHeading(); |
| 31. | void quizHeading(); |
| 32. | void printMenu(int option); |
| 33. | string Node::getYear() |
| 34. | string Node::getName() |
| 35. | string Node::getInfo() |
| 36. | Node\* Node::getNext() |
| 37. | void Node::setYear(string year) |
| 38. | void Node::setName(string name) |
| 39. | void Node::setInfo(string info) |
| 40. | void Node::setNext(Node\* nextNode) |
| 41. | void displayList(Node\* node); |
| 42. | void insertFirstNode(Node\*\* head, string y, string n, string d); |
| 43. | void insertLastNode(Node\*\* head, string y, string n, string d); |
| 44. | void insertAt(Node\* prev, string y, string n, string d); |
| 45. | int getNodeCount(Node\* node, int count = 0); |

# 3.3 Block scheme



# 4. Summary

The idea of the project is to make a program that is related with historical events. We chose to focus on the bulgarian uprisings. One of the main things that we focused on was the Gray code that we should've used in our program. We also had to use linked lists, so we did it by saving all uprisings as a linked list. We used C++ as main language.