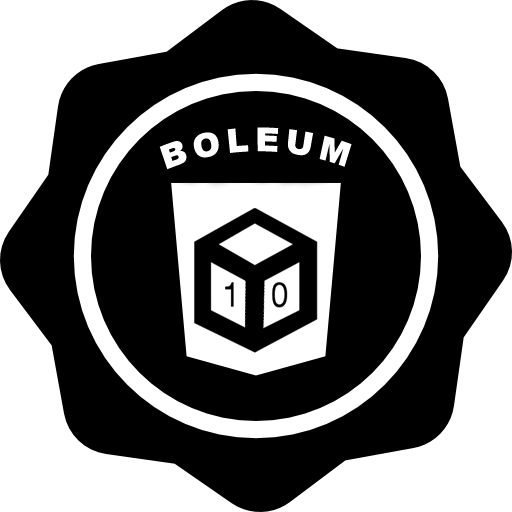
# TEAM BOLEUM

LOGIC GAME PROJECT



# MADE BY

BOLEUM

FIND US ON   
GITHUB

REPOSITORY

Logic-Game-Project

TABLE OF CONTENT

[IDEA OF THE PROJECT 3](#_Toc103550732)

[TEAM MEMBERS 3](#_Toc103550733)

[PROJECT DESCRIPTION 3](#_Toc103550734)

[PERFORMED TASKS 4](#_Toc103550739)

[FUNCTIONS 5](#_Toc103550749)

[BLOCK DIAGRAM 8](#_Toc103550755)

# IDEA OF THE PROJECT

The idea of the project is to create a game connected with the binary system called Booleo using c++.

# TEAM MEMBERS

|  |  |
| --- | --- |
| **№** | **Roles in the team** |
| **1** | Valeria Yaneva – Scrum trainer |
| **2** | Zhanet Petkova – Back-end developer |
| **3** | Hakan Chandar - Back-end developer |
| **4** | Victoria Bolashikova – QA Engineer |

# PROJECT DESCRIPTION

|  |  |
| --- | --- |
| **№** | **Description** |
|  | The idea in general.  The idea is to create a computer-based program connected with the binary system using C++ |
| **2** | How can you access the project?  You can find our project on GitHub. You can access the files by installing the repository or pasting this to your console - https://github.com/VVYaneva20/Logic-Game-Projects.git |
| **3** | Teamwork.  Our main communicating platform was Microsoft Teams. During the project, we met almost every day, so that everyone can catch up with the work. |
| **4** | What technologies are used?  The technologies we used are **Visual Studio** as our code editor, **GitHub** for collaborative work, **Microsoft Teams** for connection and communication, **PowerPoint** for creating the presentation, **Word** for creating documentation, **Excel** for the QA tests, and **Photoshop** for our logo. |

# PERFORMED TASKS

|  |  |
| --- | --- |
| **№** | **Completed tasks** |
| **1** | Create a menu  Using the menu you can navigate through the whole application. |
| **2** | Create “Person vs. Person” mode  This part of the application provides the option two individuals to play the game. |
| **3** | Create “Person vs. Computer” mode  This part of the application allows a user to play against a computer. |
| **4** | Create “Person vs. Person with not-cards” mode  At this level, the users can change the initial cards |
| 5 | Create “Person vs. Computer with not-cards” mode  At this level, the users and the computer can change the initial cards |
| **6** | Create QA documentation  In the QA documentation, you can find a description of steps and actions that have been taken to test the functionality of our program. |
| **7** | Make the README file  In the readme file, you can get a quick overview of the project. We have added the programming languages we have used and a block diagram. |
| **8** | Make the documentation  The documentation provides very useful information in terms of the technical aspect. You can learn pretty much everything by reading it. |
| **9** | Make the presentation  We have created a short presentation to quickly show what we have done. |

# FUNCTIONS

functions.cpp

|  |  |
| --- | --- |
| **Function** | **Description** |
| void gotoXY(int x, int y) | This function takes coordinates from the console |
| void emptyVariables(int numberOfcards) | This function resets variables’ value |
| void shuffleInitialCards() | This function reorganizes the initial cards |
| void shuffleCards(int numberOfCards) | This function reorganizes all cards |
| void removeCard(int card, int\* player) | This function discards a card |
| void takeCards(int cardsNeeded, int\* player, int numberOfCards) | By using this function, the player gets card from the deck of cards |
| void printCards(int\* player) | This function displays players cards |
| void printCardsWithNotCard(int\* player) | This function displays players cards |
| void printPositionsWithNotCard(int player, bool\* isOccupied, int\* values) | This function displays card positions in modes with not card |
| void printInitialPositions() | This function display the positions of the initial cards |

twoPlayersMode.cpp

|  |  |
| --- | --- |
| **Function** | **Description** |
| void printPositions(int player, bool\* isOccupied) | This function displays empty positions |
| void placeCard(bool\* occupiedPositions, int index, int pointerIndex, int x, int y, bool\* cardValues, bool value, int\* playersCards) | This function puts down a card in a chosen position |
| void checkCard(int\* playersCards, bool\* boolCardValues, int player, bool\* occupiedPositions, bool\* cardValues) | This function checks if a card is valid for the chosen position |
| void executeTurn(int\* playerCards, bool\* boolCardValues, int player, bool\* isOccupied, bool\* cardValues) | This function implements the players' turns |

twoPlayersWithNotCardMode.cpp

|  |  |
| --- | --- |
| **Function** | **Description** |
| void printPositionsNotCardMode(int player, bool\* isOccupied, int\* values) | This function displays card's positions |
| void notCardCheck(int\* values, bool\* isOccupied, bool\* initialCardValues, int playerNum) | This function checks if card is valid after Not card is used |
| void shuffleInitialCards() | This function reorganizes the initial cards |
| void printInitialCards(int position, bool reverse, int player) | This function displays initial cards after Not card is used |
| void checkCardNotCard(int\* playerscardsNotCard, bool\* initialvalues, int player, bool\* occupiedPositions, bool\* values, int\* cardValues) | This function puts down a card in a chosen position and checks if it's valid |

computerMode.cpp

|  |  |
| --- | --- |
| **Function** | **Description** |
| void chooseCardComputer(int\* player) | By calling this function the computer selects a card |
| void printPositionsComputerMode() | This function displays the empty positions |
| void placeCardPlayer(int\* playersCards, bool\* boolCardValues, int player, bool\* occupiedPositions, bool\* cardValues) | This function puts down a card in a chosen position |
| void placeCardComputer(int\* computersCard, bool\* boolCardValues, int player, bool\* occupiedPositions, bool\* cardValues) | This function puts down a card in a chosen position |

computerWithNotCardMode.cpp

|  |  |
| --- | --- |
| **Function** | **Description** |
| void printPositionsNotCardWithComputer(int player, bool\* isOccupied, int\* values) | This function displays the empty positions |
| void notCardWithComputerCheck(int\* values, bool\* isOccupied, bool\* initialCardValues, int playerNum) | This function checks if the placed cards are valid after Not card has been used |
| void chooseCardNotCardComputer(int\* player) | By calling this function the computer chooses a card |
| void printInitialCardsWithComputer(int position, bool reverse, int player) | This function displays Initial cards after Not card has been used |
| void placeCardNotCardWithComputer(bool\* occupiedPositions, int index, int pointerIndex, int x, int y, bool\* boolValues, bool value, int\* cardValues, int cardValue, int\* playersCards) | This function puts down a card in a chosen position |
| void checkCardPlayer(int\* playerscardsNotCard, bool\* initialvalues, int player, bool\* occupiedPositions, bool\* values, int\* cardValues) | This function puts down a card in a chosen position |
| void checkCardComputer(int\* playersallCards, bool\* initialvalues, int player, bool\* occupiedPositions, bool\* values, int\* cardValues) | This function puts down a card in a chosen position |

# BLOCK DIAGRAM

