**PBL**

**PROJECT REPORT**

|  |  |  |
| --- | --- | --- |
| **PROJECT TEAM** | | |
| **Student No** | **First Name** | **Surname** |
| **x19222114** | **Alexander** | **Selby** |
| **x19220448** | **Guilherme** | **Pereira** |
| **x19152086** | **Gercicleitom** | **Souza Miranda** |

**Group F**

**Word Count: 982**

**1. Project Description**

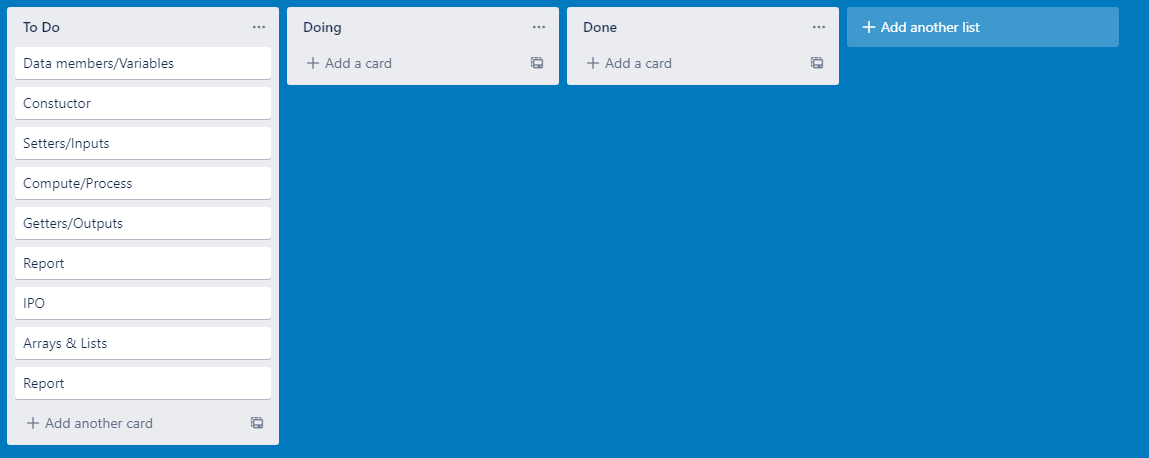
This project was developed by 3 students from the Software Development discipline.

Students were asked to develop a game called Morra Odds and Evens Variation that should allow a person to play against a computer. The person must choose between odd or even and also a number between 1 and 10. The computer would have the option that remained and the random number would be generated by the application that would count as the computer's decision. In the end, the system displayed the winner and distributed the scores among the players, with 3 points going to what hit between odd and even and 2 more points would be given to whoever took the number closest to the final result. Some tasks were requested and the solutions found by us can be found in item 3 of this report

**2. Project Scope**

The team started the work with a meeting where the ideas and steps were defined, we gathered information about the knowledge obtained during the course, points we should research from external sources, the phases we would follow and the final goal to complete the application after all the tests.

The application was developed with codes learned during classes and external research via channels like Stack Overflow. Google Drive and Github, both allowed us to exchange information simultaneously. Trello was also used to divide our tasks effectively and record all of the tasks left to do, in progress, or already done.



|  |  |  |
| --- | --- | --- |
| **Inputs** | **Process** | **Outputs** |
| **String side** – ask the user to choose either odd or even  **int userNumber** – ask the user to pick a number between 1 and 10  **String replay** – ask the user if they want to play again (Yes or No) | **Random rand –** function to generate random number  **int randNumber** – variable to store the random number  **int cRMarks** – stores how many points computer won in a round  **int pRMarks** – stores how many points player won in a round  **int sumRound -** stores total value of fingers per round  **int sumCMarks –** stores total amount of points computer won per round  **int sumPMarks –** stores total amount of points player won per round  **final int EXTPOINTS –** constant integer as given in question  **int won –** storesrounds won by player and displayed in array list  **int lost –** storesrounds lost by player and displayed in array list  **int evenP** – stores amount of times player chose an even number, displayed in array list  **int oddP –** stores amount of times player chose an odd number, displayed in array list  **int evenC** – stores amount of times computer chose an even number  **int oddC –** stores amount of times computer chose an odd number, displayed in array list  **int totalPExtra ­**– calculates total amount of extra points won by the player, displayed in array list  **int totalCExtra** – calculates total amount of extra points won by the computer, displayed in array list  **int gameCounter –** counts the games played  **int con** - keeps track of game  **int c0,c1,c2,c3 –** columns of array list  **List<Integer> arrRndWList** – array list calculating history of rounds won by player  **List<Integer> arrRndLList** – array list calculating history of rounds lost by player  **List<Integer> arrOddPList** - array list calculating history of odd numbers chosen by player  **List<Integer> arrEvenPList** – array list calculating history of even numbers chosen by player  **List<Integer> arrOddCList –**array list calculating history of odd numbers chosen by computer  **List<Integer> arrEvenCList –** array list calculating history of even numbers chosen by computer | **String message** – tell the user the points they win  **String winnerMessage** – tells the user who wins the game & how many points  **int arrHist –** array showing history of fingers shown by user  **int[][] arrHistResult –** displays rounds won and lost by player  **int[][] arrHistOE –** displays the even and odd numbers chosen by both the player and computer  **int[][] arrHistEXTPOINTS –** displays the amount of extra points won by both the player and computer |

**3. Code**

- **In each round, the game displays the computer’s choice:** Using the math.random function, we generated a random number each round.

-  **After each round the game displays the number of points each player has, and whether the user or the computer won the round:** By comparing each chosen number and whether the player chose even or odds, we used if statements to calculate who won the game.

**- A game ﬁnishes when one of the players accumulates 12 points:** An if statement allowed us to finish the game when either player’s score accumulates to 12 points

- **At the end of a game, the game displays who the winner is:** Storing and accumulating the points won by the player and computer allows us to sum up each player's points and declare a winner once either has reached 12 points.

- **A history of the numbers of ﬁngers shown by both the user and the computer per round:** Our initial array alongside a for loop stored the number chosen by the player, and the random number generated for each round and displayed them after a game had finished.

-  **Once a game has ﬁnished the application asks the player if he/she would like to play another game:** A simple input variable asking the user if they would like to play again. However, our reset method sets all variables to default, without removing all the data.

**At the end of all games, display a history of games played (using arrays):**

**- The history shows, for each game, the number of rounds won and lost by the human player**

**- How many even and odd numbers have been chosen by each player**

**- The extra points received by each player per game**

In order to be able to keep the game's history we made use of ArraysList, this allowed us to work with Arrays even without having their size previously defined. for each variable used during this stage an array was designated and at the end with the help of for loop we were able to group these values to be presented correctly

Gercicleitom started our PBL Project developing two classes that would be our instantiable and app class. Also in this first step was implemented all the process to interact and collect input from the player using JOptionPane, Do/While and IF Statement.

The next challenge was solved by Alexander. It consisted of the solution to define who in the game would win the extra points and the addition of the random and replay functions into our initial Project

Guilherme started from this point and designed the arrays to keep track of all history of the game. At this point it was detected that the replay function was conflicting with our array responsible to keep track of all games played. The path found to resolve it was creating another class. We transferred from the App to this new class all code where the input is collected.

After the members have reached the final stage of this project we invested time, reviewing the entire code taking into consideration the practice that would prepare us for the final test of this subject. This included all members of the project amending and cleaning code where applicable on each other’s contributions.Each participant had the opportunity to develop an explanation about their collaboration and at the end the project was reviewed by everyone and submitted within the requested date.

**4. Challenges**

As our team are all beginners in Java, we felt that it was a difficult task at hand trying to keep our code as concise and minimal as possible. One of the biggest challenges was to allow the application to save information in arrays and then display the information in various ways. One being at the end of each round and the other at the end of the game when the player decides that they would no longer like to participate.

**5. Conclusion**

In conclusion, we believe the PBL Project was a brilliant learning curve in our introduction to Java. Our in-class learnings, paired with external research gave us the tools to successfully develop a complex application such as this. The flexibility of the PBL Project allowed us to further develop our skills by finding new functions, and methods outside of what we learned in class. It was a great foundation that will be of great benefit to us in our studies.