Functional Requirements

The Bletchley code breakers

Version	Description of Change	Author	Date
1.0	Initial document	Milena Georgieva	26.10.2020

CONTENTS

1 INTRODUCTION

The aim of this document is to define the requirements that need to be fulfilled during the implementation phase in order to deliver a fully functional application to our client.

Every part of the software should correspond strictly to this document and every functional change request should be added to this document.

1.1 Purpose

The purpose of this document is to present our vision of the software needed as detailed as possible.

Our mission is to deliver a software which help the codebreakers to decode the coordinates of the German battleships so the allies will be able to track them and destroy them.

1.2 Scope

The scope of the current assignment is to develop console-based application. The application should be covered by test cases where applicable.

Out of scope is anything different than application mentioned above. Things like environment, deployment and similar are considered out of scope at this time.

1.3 Assumptions and Constraints

The application that will be delivered should be console –based, no additional UI is required.

2 FUNCTIONAL REQUIREMENTS

2.1 Context

The game that should be developed is known as Bletchley. It is a popular board game available also online provided by different suppliers.

During the World War II a group of mathematicians was placed in Bletchley, UK. Their goal was to crack the German coding machine Enigma.

One of the greatest minds of our time was working there, the amazing Alan Turing. Thanks to his commitment the Germans were defeated, and the war was won by the alliance.

2.2 User Requirements

It’s our mission to implement a code cracking software that will help us crack codes in case we face similar conditions.

Our aim is to implement a game where we will guess the coordinates of a German battleship.

The gameplay:

The Germans (A player or the computer) place a combination of 4 random numbers in the range between 0 and 7, indicating the coordinates of their battleships (Figure 1, Battleship's coordinate).

The code breaker should guess the number within 13 tries using the board (Figure 1, player's guesses). The codebreaker places their guesses. The Germans (a player or a computer) should give a feedback whether just a number is guessed or a number and is on a correct place. If just a number is guessed, it should be given as in Figure 1, "Guessed numbers". If the code breaker guesses a number on the correct position, then the information should be displayed as in Figure 1, "Guessed numbers and positions".

The Germans are <u>NOT</u> allowed to give any further information except the correct number of the guessed numbers or the number of the guessed numbers on correct positions!

The game ends after 13 iterations or if the code breaker guessed the correct numbers and are placed on the correct positions.

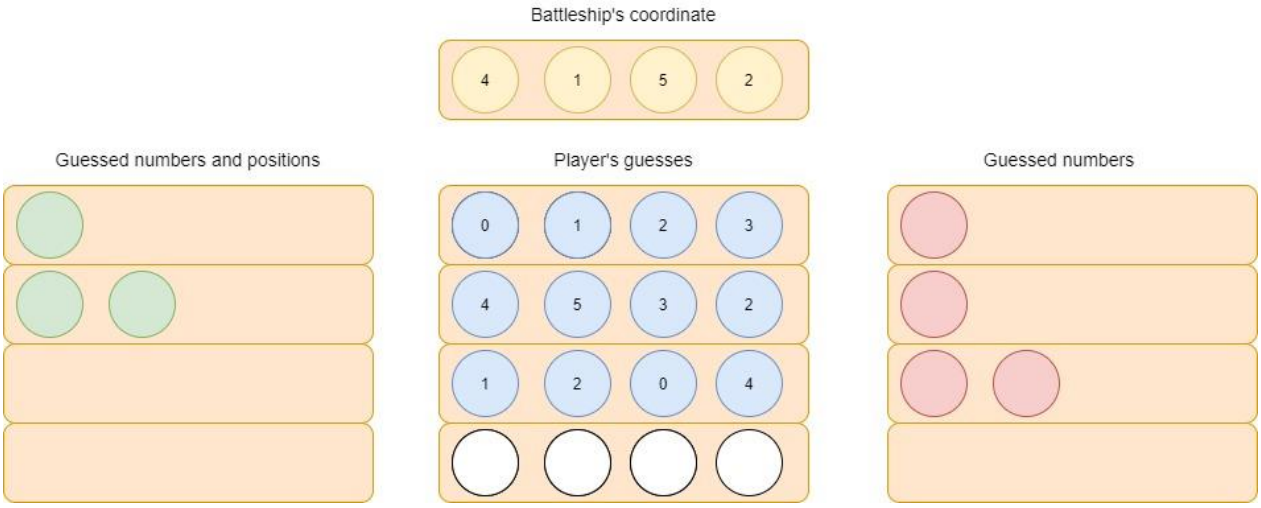

Figure 1. The gameplay

Your assignment is to implement the base rules of the game as follows:

Level 1: The game will allow two players to play against each other. One player will be in the role of the Germans and will set up the combination. The second player will be in the role of the code breaker and will try to guess the correct coordinates.

Task 1: The players have 8 numbers and will have to guess 4 of them. The number can't repeat in the initial set up.

Task 2: The players have 8 numbers and will have to guess 4 of them. The initial setup of the code CAN have repeated number.

Level 2: The game will provide the option a player to compete against the computer. The computer will generate coordinates and the codebreaker will try to guess them.

Task 1: The players have 8 numbers and will have to guess 4 of them. The number can't repeat in the initial set up.

Task 2: The players have 8 numbers and will have to guess 4 of them. The initial setup of the code CAN have repeated numbers.

Bear in mind the world counts on you! And you have no second chance to safe us all!

3 OTHER REQUIREMENTS

3.1 Interface Requirements

There is no requirement for UI currently.

3.2 Hardware Interfaces

The game should be optimized to use an average PC. It should not require more than 2 GB of RAM and no specific video card. The game should be installed on Windows machines. No Linux or Mac versions is needed.

3.3 Software Interfaces

The game should be a console application. The console should be cleared after every guess made by the code breaker with it. When new choice is added you should display the history of the game so far.

3.4 Error Handling

All errors and exceptions should be handled. User friendly messages should be provided.

3.5 Validation Rules

All data that will be entered by the users should be validated and error messages should be provided to the users when applicable.