Prac 06 Design

B, Bukanga

221005009

**Problem Description**

Initialisation:

• The size of the array is always 20 items.

• Each array item has a 50% chance of starting with either a zero or one value.

• The pointer is placed in a random location at the start of the game.

Movement:

• The player may move the pointer one step either left or right. The pointer may not move outside the array area.

• When the pointer moves, the value that the pointer points to changes its value. A value of one change to zero. A value of zero changes to one.

• The player may choose to quite the game at any time.

End game:

• The game ends in success if all the array elements have been changed to either ones or zeros. For this program, you must use user-defined libraries with functions defined and declared in the BattleSpace namespace. Activities and functions in the program should use a fixed size one dimensional array.

**Input and Output**

Case 1

|  |  |
| --- | --- |
| Input |  |
| (a) To move left | Standard input stream |
|  |  |
|  |  |
| Output |  |
| Pointer moves Left or right |  |

Case 5

|  |  |
| --- | --- |
| Input |  |
| (d) To move right | Standard input stream |
| Output |  |
| Pointer Moves Right | Standard output stream |

Option 6

|  |  |
| --- | --- |
| Input |  |
| Null |  |
| Output | |
| Exit Application |  |

**Data Format**

|  |  |  |
| --- | --- | --- |
| Identifier | Data type | Description |
| chInput | Char | Select a or d |
| arrNum | Integer | Array of ones and zeros |
| Void Movement |  | Handles the pointer movements |
| Void Output |  | Displays the user interface |
| Void InitGame |  | Initializes the Game |
| GetRand | Integer | Creates a random value in each position in the arrNum between 1 and 0 |

**Pseudo Code (for initializer)**

For each position in arrNums

Int intSlot 🡪 GetRandom()

arrNum[intSlot] 🡪 0

Int intSlot 🡪 GetRandom()

arrNum[intSlot] 🡪 1

Pointer 🡪 arrNum[0]

UML

