**Project 8 Design Document**

**Program Requirements**

This program is going to mimic a memory game. There will be two rows of face down cards with each row having cards numbered from 1-10 at random. This will be done using a 2-D array. The cards will start out face down with both rows labeled from 1-10. The user will take two guesses (one guess for the first row and another guess for the second row). The objective of the game is to match the corresponding value of the cards in each row. If the user guesses correctly, the cards will be labeled with an X. The game is over once the user guesses all of the matches correctly.

**Program Inputs**

* Guess 1
  + guess1
  + int value
* Guess 2
  + guess2
  + int value

**Program Outputs**

* Card Display
* cardrows[][]
* 2-D integer type array

**Test Plan**

* Test for validity of program (not highlighted)
* Test for valid input from user (red)
* Test for correct output from printBoard function (yellow)
* Test for card match validity (blue)
* Test for correct insert of random values (green)
* Test for implementation of ‘X’ when card is selected (orange)
* Test for duplication in randomization (green)
* Row 1:: 1 2 3 4 5 6 7 8 9 10
* Row 2:: 1 2 3 4 5 6 7 8 9 10
* What number card would you like to check from the first row?
* 1
* 1 is a(n) 3
* What number card would you like to check from the second row?
* 1
* 1 is a(n) 10
* No match!
* Row 1:: 1 2 3 4 5 6 7 8 9 10
* Row 2:: 1 2 3 4 5 6 7 8 9 10
* What number card would you like to check from the first row?
* 1
* 1 is a(n) 3
* What number card would you like to check from the second row?
* 2
* 2 is a(n) 1
* No match!
* Row 1:: 1 2 3 4 5 6 7 8 9 10
* Row 2:: 1 2 3 4 5 6 7 8 9 10
* What number card would you like to check from the first row?
* 1
* 1 is a(n) 3
* What number card would you like to check from the second row?
* 3
* 3 is a(n) 3
* Match!
* Row 1:: X 2 3 4 5 6 7 8 9 10
* Row 2:: 1 2 X 4 5 6 7 8 9 10
* What number card would you like to check from the first row?
* 2
* 2 is a(n) 1
* What number card would you like to check from the second row?
* 4
* 4 is a(n) 2
* No match!
* Row 1:: X 2 3 4 5 6 7 8 9 10
* Row 2:: 1 2 X 4 5 6 7 8 9 10
* What number card would you like to check from the first row?
* 2
* 2 is a(n) 1
* What number card would you like to check from the second row?
* 5
* 5 is a(n) 4
* No match!
* Row 1:: X 2 3 4 5 6 7 8 9 10
* Row 2:: 1 2 X 4 5 6 7 8 9 10
* What number card would you like to check from the first row?
* 2
* 2 is a(n) 1
* What number card would you like to check from the second row?
* 6
* 6 is a(n) 5
* No match!
* Row 1:: X 2 3 4 5 6 7 8 9 10
* Row 2:: 1 2 X 4 5 6 7 8 9 10
* What number card would you like to check from the first row?
* 2
* 2 is a(n) 1
* What number card would you like to check from the second row?
* 7
* 7 is a(n) 8
* No match!
* Row 1:: X 2 3 4 5 6 7 8 9 10
* Row 2:: 1 2 X 4 5 6 7 8 9 10
* What number card would you like to check from the first row?
* 2
* 2 is a(n) 1
* What number card would you like to check from the second row?
* 8
* 8 is a(n) 7
* No match!
* Row 1:: X 2 3 4 5 6 7 8 9 10
* Row 2:: 1 2 X 4 5 6 7 8 9 10
* What number card would you like to check from the first row?
* 2
* 2 is a(n) 1
* What number card would you like to check from the second row?
* 9
* 9 is a(n) 6
* No match!
* Row 1:: X 2 3 4 5 6 7 8 9 10
* Row 2:: 1 2 X 4 5 6 7 8 9 10
* What number card would you like to check from the first row?
* 2
* 2 is a(n) 1
* What number card would you like to check from the second row?
* 10
* 10 is a(n) 9
* No match!
* Row 1:: X 2 3 4 5 6 7 8 9 10
* Row 2:: 1 2 X 4 5 6 7 8 9 10
* What number card would you like to check from the first row?
* 2
* 2 is a(n) 1
* What number card would you like to check from the second row?
* 2
* 2 is a(n) 1
* Match!
* Row 1:: X X 3 4 5 6 7 8 9 10
* Row 2:: 1 X X 4 5 6 7 8 9 10
* What number card would you like to check from the first row?
* 3
* 3 is a(n) 4
* What number card would you like to check from the second row?
* 5
* 5 is a(n) 4
* Match!
* Row 1:: X X X 4 5 6 7 8 9 10
* Row 2:: 1 X X 4 X 6 7 8 9 10
* What number card would you like to check from the first row?
* 4
* 4 is a(n) 2
* What number card would you like to check from the second row?
* 4
* 4 is a(n) 2
* Match!
* Row 1:: X X X X 5 6 7 8 9 10
* Row 2:: 1 X X X X 6 7 8 9 10
* What number card would you like to check from the first row?
* 5
* 5 is a(n) 6
* What number card would you like to check from the second row?
* 9
* 9 is a(n) 6
* Match!
* Row 1:: X X X X X 6 7 8 9 10
* Row 2:: 1 X X X X 6 7 8 X 10
* What number card would you like to check from the first row?
* 6
* 6 is a(n) 9
* What number card would you like to check from the second row?
* 10
* 10 is a(n) 9
* Match!
* Row 1:: X X X X X X 7 8 9 10
* Row 2:: 1 X X X X 6 7 8 X X
* What number card would you like to check from the first row?
* 7
* 7 is a(n) 7
* What number card would you like to check from the second row?
* 8
* 8 is a(n) 7
* Match!
* Row 1:: X X X X X X X 8 9 10
* Row 2:: 1 X X X X 6 7 X X X
* What number card would you like to check from the first row?
* 8
* 8 is a(n) 10
* What number card would you like to check from the second row?
* 1
* 1 is a(n) 10
* Match!
* Row 1:: X X X X X X X X 9 10
* Row 2:: X X X X X 6 7 X X X
* What number card would you like to check from the first row?
* 9
* 9 is a(n) 5
* What number card would you like to check from the second row?
* 6
* 6 is a(n) 5
* Match!
* Row 1:: X X X X X X X X X 10
* Row 2:: X X X X X X 7 X X X
* What number card would you like to check from the first row?
* 10
* 10 is a(n) 8
* What number card would you like to check from the second row?
* 7
* 7 is a(n) 8
* Match!
* Row 1:: X X X X X X X X X X
* Row 2:: X X X X X X X X X X
* Game Over!

**Solution Overview**

Start by including all necessities for the program (iostream, cstdlib, ctime, using standard namespace, main function).

Before the main function, you will declare all of the functions. For initializeArray, it will accept a 2-D integer type array and an integer. For insertRand, it will accept a 2-D integer type array and an integer. For printBoard, if will accept a 2-D integer type array and an integer. For getGuess, it will accept a 2-D integer type array, and 2 integers (one pass by reference). For print board, it will accept a 2-D integer type array and an integer.

Inside the main function, include a random unix time seed (srand(time(0))). Then, create two constant integers for the rows and columns of the 2-D array that will be created (2 rows and 10 columns). Then, create an integer to count for the total cards found and set it equal to 0. Then, create your 2-D array (cardrows) and set the rows and columns to the constants that were created.

Then, call for the initializeArray function that accepts the array and the rows constant. Then, call for the insertRand function that accepts the array and the rows constant. Then, call for the printBoard function that accepts the array and the rows constant. Then create a while loop that states if the total number of cards found is less than 0, call for the getGuess function (accepts they array, the rows constant, and the total number of cards found) and the printBoard function (accepts array and rows constant). After the while loop, print “Game Over!”

After the main function, you will define all the functions that were declared:

For printBoard, you will accept a 2-D array and an integer for the rows. (int a[][], int r). You will then use this function to print out all of the 2-D array using a nested ‘for’ loop. If the values detected within the 2-D array are -1, you will print out an X in its position.

For initializeArray, you will accept a 2-D array and an integer for the rows (int a[][], int r). With this, you will set all of the values inside the 2-D array equal to -1.

For insertRand, you will accept a 2-D array and an integer for the rows (int a[][], int r). This function will use a for loop to insert values into the 2-D array at random from 1-10 without duplication. You will do this using a linear search system in order to avoid duplication.

For getGuess, you will accept a 2-D array, an integer for the rows, and an integer passed by reference for the number of matches found (int a[][], int r, int& f). This function will prompt the user for 2 guess inputs for the cards (one for the first row and one for the second row). After the first guess, the program will output the value of the card, and same with the second guess. You will use an “if else” statement stating that if the values do not match, print out “No match!” If the values, however, do match, you will print out “Match!” The program will then detect this match, and update the game and place an “X” for the position of the array that was guessed. You will do this by setting the matches values equal to 0 so that print board can detect the updated guess.

**Algorithm Flowchart**

Diagram, engineering drawing

Description automatically generated