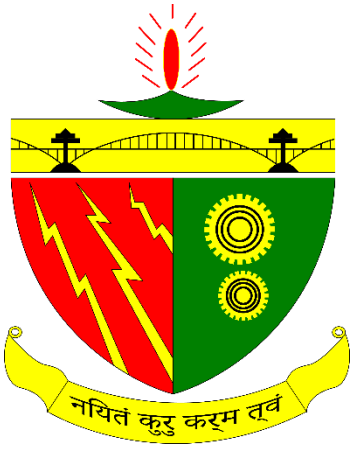


Name of Candidates:

Amritanshu Satyam (1906003)

Anand Kumar (1906004)

Registration Numbers:

19030460003 & 19030460004 (respectively)

Branch: ECE

Semester: 4th

Topic: A memory game based on Array Data Structure

INDEX

CONCEPTS USED

1. Array Data Structure:

An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together. This makes it easier to calculate the position of each element by simply adding an offset to a base value, i.e., the memory location of the first element of the array (generally denoted by the name of the array).

2. Modular Approach :

In modular programming approach, we follow two things when make the structure of the application:

- Separate different functionalities into smaller part which forms the module.
- Make the relation between the modules.

It is obvious that the modules will have dependency. This dependency is maintained through function call (with passing variables) among the modules. And it works only when the modules are loaded.

But during making the dependency structure, we must ensure that circular dependency is not formed. In case of circular dependency modules are not loaded properly (if module X and Y depends on each other, which one should load first?). Each module will wait for the other module to get loaded. So in this type of situation the dependency is not straight forward and the application fails. The developer should always take special care during design phase of the modules.

Modular Approach also helps us in making the data used, private to the function. This helps us to get rid of unwanted bugs.

3. Random Function:

Here, we are using JavaScript's inbuilt **Math.random()** function.
The **Math.random()** function returns a floating-point, pseudo-random number in the range 0 to less than 1 (inclusive of 0, but not 1) with approximately uniform distribution over that range — which you can then scale to your desired range. The implementation selects the initial seed to the random number generation algorithm; it cannot be chosen or reset by the user.

This function is used to generate random numbers which we have scaled enough so that we can use it effectively.

4. Floor Function:

Here, we are using JavaScript's inbuilt **Math.floor()** function.
The **Math.floor()** function returns the largest integer less than or equal to a given number.
It is used to get us to the required range of buttons we are giving to the users.

RULES OF THE GAME

1. User have to either press some keyboard button or the on-screen start button to start the game.

2. After start of game, the start button disappears and one colour blinks up, the user have to remember the colour that is blinked up.

3. Then, the user is given the chance to press the colour that is blinked up previously.

4. If the user presses the correct colour, the game proceeds to the next level.

5. If the user presses a wrong button, the game ends and the start button re-appear.

6. If the user chooses the correct colour, then the game proceeds to the next level, and now, again one random colour blinks up.

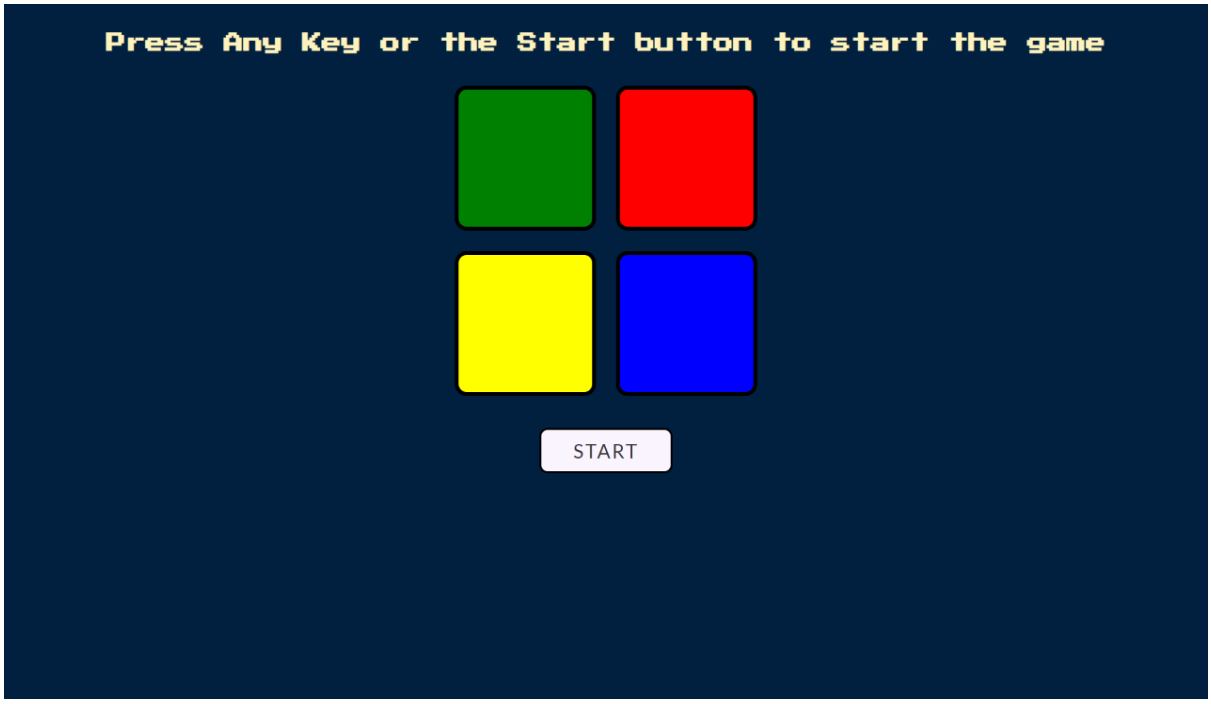
7. The user is again given the chance to press the button, but the user have to press the buttons in the same sequence as is represented before him/her since the start of game.

8. If the user chooses the correct sequence, then the game proceeds to the next level and again one colour blinks up, then the user have to press 3 buttons in the same sequence as blinked automatically by the computer since the start of game.

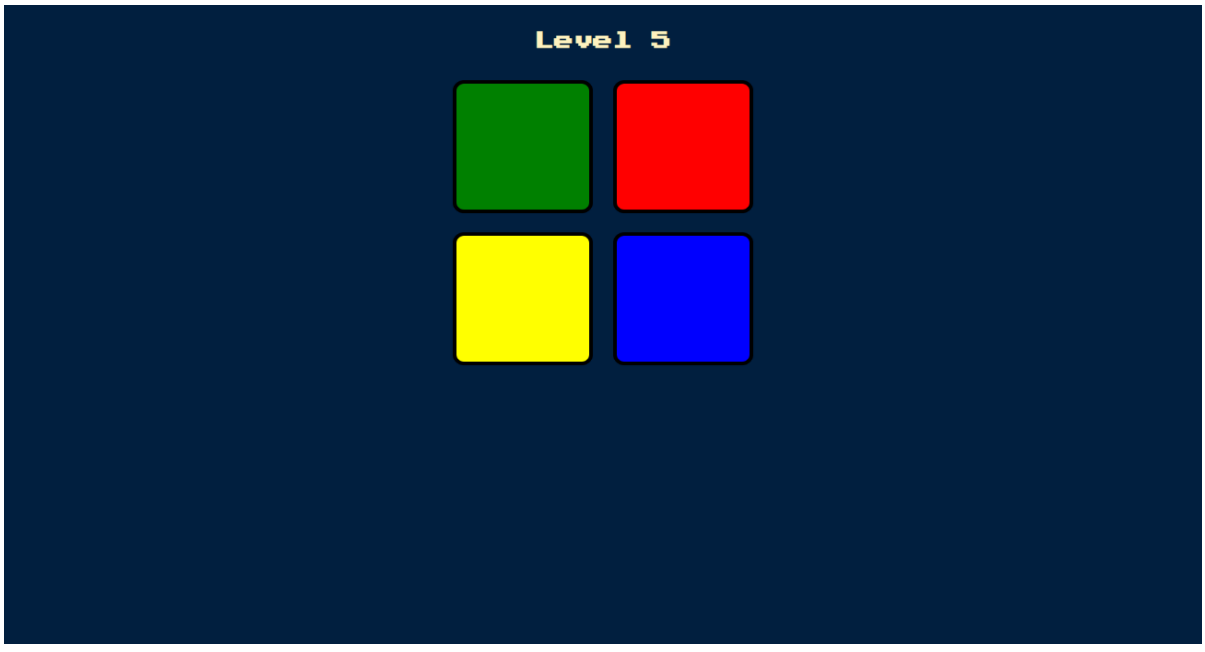
9. At any time, if the user wants to restart the game, he/she have to press a wrong key and then the restart button appears through which he/she can restart the game.

WORKING

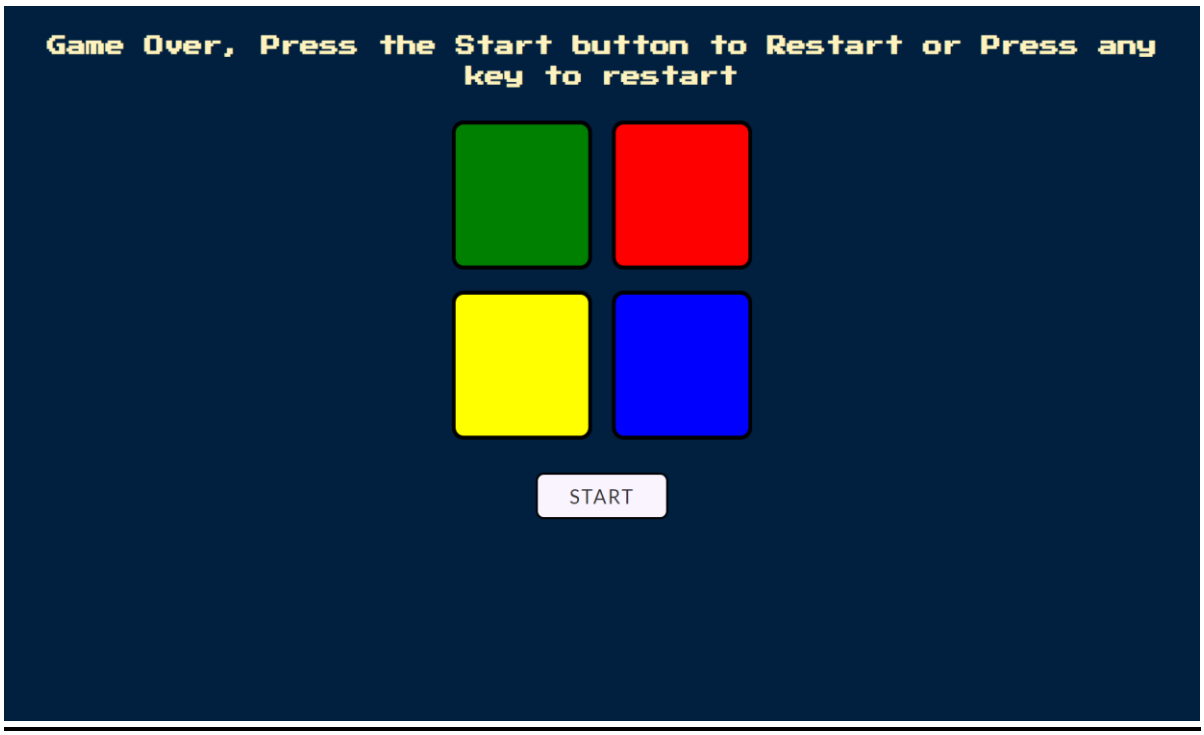
1. Start Screen



2. In-between the Game



3. Game Over


Game Over, Press the Start button to Restart or Press any key to restart
START