

RESEARCH & PROJECT SUBMISSIONS

**Program: CESS**

***Course Code: CSE334***

***Course Name: Internet Programming***

***Examination Committee***

**Dr. Hisham Farag**

**Ain Shams University**

**Faculty of Engineering**

**Spring Semester – 2020**

**Student Personal Information for Group Work**

|  |
| --- |
| **Student Names:** |
| **Ahmed Mamdouh Mohammed (CSS, restart button and game logic)** |
| **Nada Tarek (HTML, start button and allocating images to cards)** |

|  |
| --- |
| **Student Codes:** |
| **16P6020** |
| **16P6053** |

**Plagiarism Statement**

I certify that this assignment / report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they are books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment / report has not been previously been submitted for assessment for another course. I certify that I have not copied in part or whole or otherwise plagiarized the work of other students and / or persons.

**Signature/Student Name:**

**21-5-2020**

**Date:**

**Ahmed and Nada**

**Submission Contents**

**01: Abstract –** Page 3

**02: Introduction –** Page 5

**03: Methods –** Page 7

**04: Requirements Analysis -** Page 10

**05: Fifth Section Title**

**01**

***ABSTRACT***

# **ABSTRACT**

This report discusses in detail the development process of a matching cards web game. In the beginning, we introduced the game details, rules, the effect it has in improving the player’s memory over time and the background knowledge needed to develop such a game. Afterwards, we discussed the approaches associated with developing such a game, then we wrote down the functional and non-functional requirements that are associated with this game. Afterwards, our architectural, data and component designs of this gaming system are discussed in detail with diagrams that show the exact design we had in mind when we developed this game and we showed screenshots of the user interface, then we attached the actual code of our game which we wrote based on the design diagrams. The code files are basically one HTML file, one CSS file and one JavaScript file. In the end, we showed screenshots of the game while it’s being played to show the different states of the system, also we included all the sources and references we used.

**02**

***INTRODUCTION***

# **2.0 INTRODUCTION**

This memory matching cards game is a game in which there is a matrix of hidden images where each image has a copy of itself. Two players take turns to try and find as many matching images as possible before all cards run out. There are three levels for this game, they are:

1. Easy (matrix of 4x4 images)
2. Medium (matrix of 4x5 images)
3. Hard (matrix of 4x6 images)

The score of each player increases by two whenever he/she finds matching images. The player with a bigger score at the end of the game is the winner. The game is very important whether it is for improving your memorizing skills or for learning more about the development process of a web application using CSS, HTML and vanilla JavaScript. The idea we had to develop this game is to create a submission form which takes the players’ names and avatars as input and the level (easy, medium or hard) of their choice. Afterwards, the user clicks on start which shows the game board containing the matrix of hidden images, the names and avatars of the players, the scores of the players, the current player’s turn and a restart button which when clicked takes you back to the original submission form. When the game ends, either a celebration image appears to celebrate one of the two player’s victory or a sad emoji image appears when the game ends in a draw.

**03**

***METHODS***

# **METHODS**

This section discusses the various approaches and techniques that could be used to develop the matching cards memory game.

**The first approach** would be to use HTML and CSS to construct the user interface and vanilla JavaScript with the Document Object Model (referred to as DOM) to control and manipulate the HTML code based on the user’s actions. The DOM allows us to manage the game logic whenever the user presses any of the hidden images or the start/restart button. However, this approach does not allow us to store the players’ information as there is no backend server available, so players cannot create accounts and save their data, scores and results on the website. This approach is the one we decided to use in our project as creating user accounts is not needed in the requirements of this project.

**The second approach** would be to use JavaScript frameworks such as jQuery and React instead of vanilla JavaScript which would make the written JavaScript code a lot less than the first approach as these frameworks provide a lot of APIs to make it easier for us to code. Also, in an approach like this, Node.js framework could be used which is a JavaScript runtime environment which allows JavaScript to be used for backend development which could give us the option to allow the users to create accounts storing their data, scores

and results. We decided not to use this approach as it was too advanced and not really needed in a web application as simple as this game is.

**The third approach** would be to use a either the first approach or the second approach, but in addition, we would use a machine learning/deep learning JavaScript framework such as TensorFlow.js or Brain.js which would allow users to play against the AI in addition to playing against other players. The AI would have different difficulty levels. We decided not to use this approach as the required mode is to allow two players to play against each other.

**04**

***REQUIREMENTS ANALYSIS***

4.0 REQUIREMENTS ANALYSIS

4.1 Introduction to The Requirements

**The function** of this system is mainly to provide a smooth gaming experience following the previously mentioned rules of the matching cards memory game and it should be played by two players who take turns to play against each other.

**The scope** is as the following, the two players should be able to choose their names and display picture. There should be three levels to choose from. There should be a start button that submits the choices that the players have chosen and starts the gaming session. Once the gaming session starts, the hidden images appear, and players take turns to get a higher score. Once a matching image pair is found, users cannot choose the same images again. The game ends when the cards run out and we either have a draw or one player would be the winner. There should be a restart button that allows the users to go back to the submission form at any time during the gaming session.

**The success criteria** are that the game should run smoothly without any bugs or errors with a nice looking and friendly user interface that isn’t too complex for the user to use.

**Some technical functionalities:**

1. In the start and restart button, the DOM should be used to modify the HTML once any of the two buttons is clicked. If the start button is clicked, the HTML should be changed to show the matrix of hidden images. If the restart button is clicked, the HTML should be changed to make the image matrix disappear and show the submission form again, but it should also keep their previously chosen names and display pictures instead of letting the users type their names and choose their display pictures again.
2. When a player clicks on a card, the image hidden in this card should be revealed and kept visible until the player clicks on another card and reveals another image. This is done by using a counter variable which is equal to 0 when it’s the player’s first click and it should be equal to 1 when it’s his/her second click. The two images are then compared. If they match, then the player gets two points and the cards cannot be clicked on again however, if they don’t match, the two images remain visible for 1 second for both players to be able to memorize them. After the second has passed, the images become hidden again and the player’s turn is over.

4.2 Functional Requirements

The functional requirements of the matching cards game will be discussed in detail in this subsection. The functional requirements of our system are **a scoring system**, **multiple levels of difficulty**, **choosing a name and a display picture**, **a restart game functionality**, **revealing hidden images**, **comparing revealed images** and **announcing game results**.

4.2.1 Scoring System

This requirement is extremely essential for our game to work as expected. Whenever a player finds a matching pair of images, their score increases by two. If there is no match between the images, the player’s score shouldn’t change. By the end of the game session, the player with the higher score is the winner. If both players have the same score, it’s a draw. During the game session, the score of each player should be visible.

4.2.2 Multiple Levels of Difficulty

This requirement states that there must be various difficulty levels that the players can choose from. The easiest level should have a 4x4 image matrix which means that the game board should have 8 hidden image pairs (a total of 16 images). As the difficulty increases, the number of images in the game board should increase which would make the game harder for the players. We decided to have three levels: easy (16 total images), medium (20 total images) and hard (24 total images). The user should be allowed to choose the level before starting a game session.

4.2.3 Choosing the Name and the Display Picture

This requirement states that the players should be allowed to choose their names and display pictures. Each player should type his/her name in a submission form. As for the display picture, the players should be allowed to choose from a set of images available by the website. During the game session, the name and display picture of each player should be visible.

4.2.4 Restart Game Functionality

This requirement states that there must be an option available to the user anytime during the game session to restart the current game session. It should remove the hidden images and take the users back to the submission form where they can choose their names, display pictures and/or the game level. When a restart is provoked, the system should be able to store the previously chosen names and display pictures and include them in the submission form to make it easier for the players to restart the game as fast as possible. They should be still allowed to change them if they wanted to.