



PROJECT

Memory Game

A part of the Front-End Web Developer Nanodegree Program

PROJECT REVIEW

CODE REVIEW 5

NOTES

SHARE YOUR ACCOMPLISHMENT!  

Requires Changes

2 SPECIFICATIONS REQUIRE CHANGES

Dear Student,

This was a brilliant submission and the timer works as required and I really enjoyed reviewing your work though some points failed to meet specifications. Please check my feedback amend or add what is required. I am sure the next submission will be far better. Keep the momentum going and we will soon complete this project. Good luck for the next submission.

Game Behavior

The game randomly shuffles the cards. A user wins once all cards have successfully been matched.

Awesome job here. Each time the game is started, the cards are shuffled randomly, no shuffling pattern is observed. I was able to match all the cards and won the game indicated by a congratulation message. Way to go!

When a user wins the game, a modal appears to congratulate the player and ask if they want to play again. It should also tell the user how much time it took to win the game, and what the star rating was.

Great work! This submission displays a winning message with statistics about the game. Nice work implementing the modal.

A restart button allows the player to reset the game board, the timer, and the star rating.

The restart button was well implemented as it rightly resets the timer, the game board, and the star rating.

The game displays a star rating (from 1-3) that reflects the player's performance. At the beginning of a game, it should display 3 stars. After some number of moves, it should change to a 2 star rating. After a few more moves, it should change to a 1 star rating.

The number of moves needed to change the rating is up to you, but it should happen at *some* point.

Well done implementing a star rating that reduces as the player makes more moves. However, this rubric fails because the star rating varies between 3 and 0 meanwhile the rubric requires the rating to vary between 3 and 1.

Required

- The star rating should range from `3 to 1` and not from `3 to 0`.

Tips

- [Implementing a star rating in CSS.](#)

When the player starts a game, a displayed timer should also start. Once the player wins the game, the timer stops.

Excellent work providing a timer. The implementation adheres to this rubric item.

Game displays the current number of moves a user has made.

Nice job! Moves are correctly implemented and displayed to the player.

Interface Design

Application uses CSS to style components for the game.

Awesome! The game interface can speak for itself, it's very stylish and responsive with some great animations portraying matched and mismatched cards. It's actually attractive to play. Good work writing the amazing app.css file and using [bootstrap](#).

All application components are usable across modern desktop, tablet, and phone browsers.

Great work here. The application works across desktop, tablet and phone browsers.

Documentation

A `README` file is included detailing the game and all dependencies.

Nice job submitting a README file. However, this rubric fails to meet specification as the README file doesn't include the dependencies and or resources used in developing the game.

Required.

- Edit the README file and ensure it describes the following
 - How The game was created.
 - How the game is played.
 - What the player is to expect from the game.
 - Game dependencies if any.
 - Resources used to develop the game if any.

Tips

- One can write better README files by following [this Udacity course on README files](#).

Comments are present and effectively explain longer code procedure when necessary.

Well done commenting the code as it makes it easy for everyone to follow and understand the code.

Tips

- [Why and how to comment code?](#)
- You can follow the [Udacity javascript code style](#) style guide for better code style concerning. The `comment` section of this guide describes how one can effectively comment functions.

Code is formatted with consistent, logical, and easy-to-read formatting as described in the [Udacity JavaScript Style Guide](#).

Nice job here. Your code is readable and logical.

Suggestion

- Please have a look at the code review section.

 RESUBMIT DOWNLOAD PROJECT5 [CODE REVIEW COMMENTS](#)

Learn the [best practices for revising and resubmitting your project](#).

[RETURN TO PATH](#)[Student FAQ](#)