M256 Project 3 Review Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. On a scale of 1(not much completed) to 5(all requirements and enhancements completed), how would you rate your project? Explain.

1 2 3 4 5

I completed all the requirements and enhancements, but didn’t really add anything original

1. What is the best feature of your project (What are you most proud of)?

The page layout is maybe the most impressive, with the different suits in the corners and the cards remaining in the deck off to the sides of the page

1. List all enhancements (or any “extras”) that you completed.

Pacifist button, displaying the cards remaining in the deck, not shuffling the deck between rounds, adding css

1. List all enhancements (or any “extras”) that you worked on but did not complete.

I tried giving the pacifist button the properties of the other button, but wasn’t sure how to not have an infinite loop at the end when each deck only has a few non-identical cards

1. On a scale of 1 (I didn’t work very hard in class or out) to 5 (I worked hard every day both inside of class and out), how would you rate your effort on this project? Explain.

1 2 3 4 5

I focused on this every day during class and thought about it outside of class, but I didn’t progress much outside of class since I couldn’t access the picture files

1. On a scale of 1(not at all) to 5(it was the highest priority), how much did you consider efficiency in developing your code? Explain.

1 2 3 4 5

I attempted to be as efficient as possible, but I was unable to figure out a way to display the correct cards which remained in the deck without using 8 separate arrays

1. Give a specific example of a way in which you moved forward in your knowledge of HTML/CSS/JS (or something new you learned how to do) in the process of creating this program.

I was able to specifically position elements using CSS, such as the deck displays and suit pictures

1. What was the most challenging part of creating this project? Explain.

Enhancement 2, I had thought of using loops to only generate valid cards, but didn’t succeed in any attempts until the hint was posted

1. Describe the incremental development process of your program, focusing on the midpoint of the development process (think of Friday of last week)
   1. Discuss the specifics of how your program was functioning. (What did you have working on that day?)

I was able to complete a game while adjusting the score and displaying the cards and winner, and the pacifist button was functional

* 1. Describe any difficulties encountered with the intended functionality (What were you struggling with at that point?)

All of the loops that I tried using with the delete function eventually either did not display the card, or resulted in an infinite loop. When I tried using the splice function, the cards no longer corresponded with the number that was generated

* 1. Explain how those difficulties were resolved.

After the hint was posted, I realized that I didn’t need to perform some mathematical magic, and edited my code to just continually generate numbers until a valid one was found

Grade your project, using the rubric below:

* HTML Elements:

*Element with innerHTML property for:*

*Button to play a round*

*<img> element to display player’s card*

*<img> element to display computer’s card*

*Element to display the outcome of the current round*

*Element to display the current score*

*Button to restart the entire game*

|  |  |
| --- | --- |
| *6* | */ 6* |

* Generate a random number between 0 and 51 to represent the player’s card, and

another random number between 0 and 51 to represent the computer’s card.

|  |  |
| --- | --- |
| *3* | */ 3* |

* Correctly decode the meaning of each card number 0-51, and use that information

to determine who the winner is (or if it’s a tie).

|  |  |
| --- | --- |
| *6* | */ 6* |

* Correctly use a loop to generate a 52-element array to store “0.jpg”, “1.jpg, … , “51.jpg”, and use it

to display a picture of each player’s card on the page.

|  |  |
| --- | --- |
| *6* | */ 6* |

* After each round, adjust the score accordingly.

|  |  |
| --- | --- |
| *2* | */ 2* |

* Do all processing in functions with appropriate programming structures

(output strings, local & global variables, decision statements, etc.)

|  |  |
| --- | --- |
| *2* | */ 2* |

* Overall quality of your code, including appropriate comments, formatting, indenting,

descriptive variable & function names, etc.

|  |  |
| --- | --- |
| *2* | */ 3* |

* Pacifist Button

|  |  |
| --- | --- |
| *2* | */ 2* |

* Buttons to display the player’s remaining deck and the computer’s remaining deck

I didn’t use individual buttons, instead I just incorporated this into the main button

|  |  |
| --- | --- |
| *0?2?* | */ 2* |

* Enhancement #2: Deck does not get shuffled in between rounds. When we run out of cards, the game is over and declare a winner.

|  |  |
| --- | --- |
| *8* | */ 8* |

* Design enhancements:

|  |  |
| --- | --- |
| *2* | */ 3* |