NEA Euan Worth

Card Game

08479

2021

Contents

[Analysis 2](#_Toc60998711)

[Stakeholders 2](#_Toc60998712)

[Success Criteria 2](#_Toc60998713)

[Practices 2](#_Toc60998714)

[Research 2](#_Toc60998715)

[Hardware & Software Requirement 2](#_Toc60998716)

[Design 2](#_Toc60998717)

[Development 2](#_Toc60998718)

[Iteration 1 2](#_Toc60998719)

[HTML (Non Functional) 2](#_Toc60998720)

[JS 3](#_Toc60998721)

[Evaluation 4](#_Toc60998722)

[Bibliography 4](#_Toc60998723)

# Analysis

## Stakeholders

Firstly, A stakeholder would be Louise, who commissioned the game, she wants the game to functional to the rules she has laid out. This means that she wants to game to function easily

I would qualify as a stakeholder, given that I will make the game, and want it made to the order of the client/ commissioner.

## Success Criteria

Users should be able to log on with correct details

Users should be able to use a shuffled deck.

Users should be able to create accounts

Users should be able to draw cards

Users should be able to win and find out that they have one

Users should be able to view a leaderboard

Administrators can manage accounts

Administrators can reset passwords

## Practices

This will be made in HTML/CSS/JS in webstorm

## Research

## Hardware & Software Requirement

# Design

# Development

## Iteration 1

### HTML (Non Functional)

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Play! - Card Quick Draw</title>  
</head>  
<body>  
  
<script src=".\game.js"></script>  
</body>  
</html>

### JS

const ***Colours*** = ["red", "yellow", "black"]  
class card{  
 constructor(colour, number) { //creates a new card  
 this.colour = colour  
 this.number = number  
 }  
  
 get colourNumber() { //returns a number of the card for maths purposes  
 return ***Colours***.indexOf(this.colour)  
 }  
  
 beats(card = new card("red", 1)) { //checks if one card beats another  
 if(this.colour === card.colour) {//if the colours are the same, check the numbers else  
 return (this.number > card.number)  
 } else {//else checks the colours (checks if the other colour is one less than its own colour)  
 return ((this.colourNumber - 1) % 3 === card.colourNumber)  
 }  
 }  
}  
  
class player{ //class for player  
 constructor() { //creates a deck  
 this.deck = []  
 }  
 get score() { //gives him a player  
 return this.deck.length  
 }  
}  
  
  
/\* Randomize array in-place using Durstenfeld shuffle algorithm \*/  
function shuffleArray(array) { //StackOverflow user: ashleedawg at <https://stackoverflow.com/questions/2450954/how-to-randomize-shuffle-a-javascript-array> (ashleedawg, 2020)  
 for (let i = array.length - 1; i > 0; --i) {  
 const j = ***Math***.floor(***Math***.random() \* (i + 1));  
 [array[i], array[j]] = [array[j], array[i]];  
 }  
}  
  
let ***deck*** = [] //deck array  
for(let i = 0; i < 3; i++) { //add cards to deck  
 const Colours = ["red", "yellow", "black"]  
 for(let j = 0; j < 10; j++) {  
 ***deck***.push(new card(Colours[i],j + 1))  
 }  
}  
shuffleArray(***deck***) //shuffles the deck  
  
  
let ***players*** = [new player, new player] //adds the players for scores  
  
  
for(let i = 0; i < 30; i+=2) { //checks who wins and adds the cards to their deck  
 if (***deck***[i].beats(***deck***[i+1])) {  
 ***players***[0].deck.push(***deck***[i],***deck***[i+1])  
 } else {  
 ***players***[1].deck.push(***deck***[i],***deck***[i+1])  
 }  
}  
  
if(***players***[0].score > ***players***[1].score) {//outputs the winner and their deck  
 ***console***.log("Players 1 Wins with:")  
 ***console***.log(***players***[0].deck)  
} else {  
 ***console***.log("Players 2 Wins with:")  
 ***console***.log(***players***[1].deck)  
}

# Evaluation

# Bibliography

ashleedawg. (2020, March 9). *How to randomise (shuffle) an JavaScript Array?* Retrieved from Stack Overflow: https://stackoverflow.com/questions/2450954/how-to-randomize-shuffle-a-javascript-array