université de BORDEAUX

Master of Science in BioInformatics – HTML/CSS/Javascript project

Raphael Marczak, based on Florent Grelard's work

This project can be implemented in pairs (two students), and should be sent to raphael.marczak@mmibordeaux.com by 24/12/2018. Please specify the names of the two students in the e-mail when sending the project.

The different images used in this project are available in the same archive containing this pdf.

Marathon Jack

The main objective of this project is to create a webpage allowing someone to play an adaptation of the "Black Jack" game ; termed "Marathon Jack". In this game, a player is competing against a dealer. The goal is to beat the dealer by getting a cards count as close to 42 as possible, without going over 42. If the player goes over 42, he loses. Cards values are defined as such :

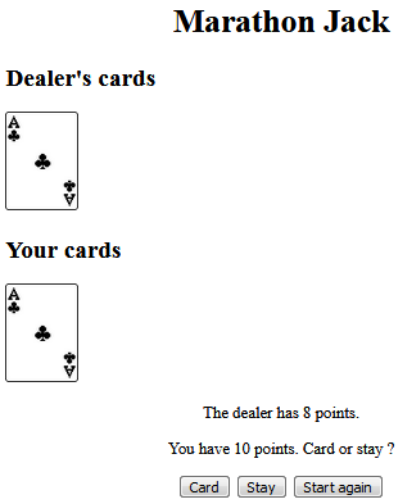
- ace : 1 point
- from 2 to 9 : the card number
- king, queen or jack : 10 points

When the game starts, the dealer gives a card to the player, and one for himself. He then asks the player to choose between two options :

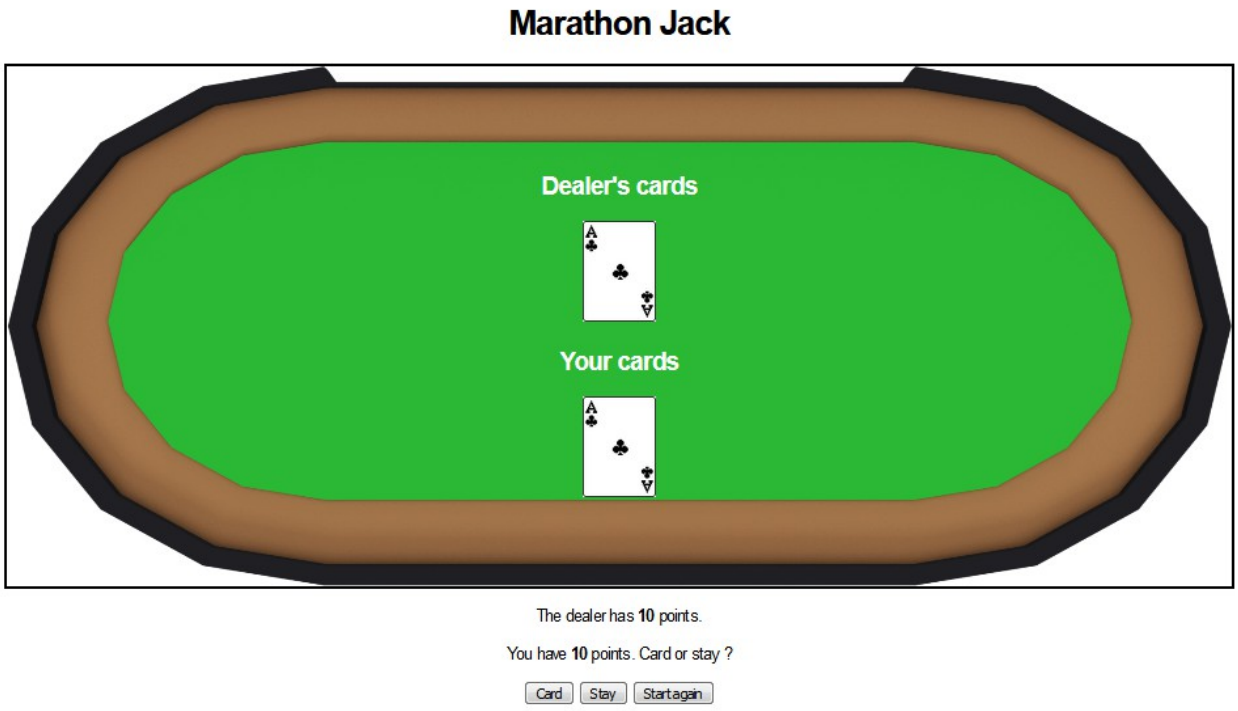
- "card", meaning that the player is asking for another card. The player can ask as many cards as he wants, but looses if the total value goes over 42.
- "stay", meaning that the player does not want more cards, and then the dealer will try to beat the player

HTML/CSS

1. Create a webpage (html) displaying the “marathon jack” environment, in a static way, and by using the image 1.png as a sample card. For the number of points, you can write a random one for now. You should obtain something like that :



2. Then, create a CSS file so that the page will look like the following figure. You can use the picture table.png as a background.



Javascript

1. When the page is loaded, give a random card to the dealer and to the player, by using the javascript function Math.random.
2. Each time the player is pressing the « card » button, a new card should be added to the existing ones. **Hint :** at the end of this pdf, you will find a source code showing how to add elements directly into the DOM (HTML tree)
3. Compute the player score, by adding the cards values. **Hint** : the value can be guessed through the filename (1.bmp, 14.bmp, 27.bmp and 40.bmp are all aces ; 2.bmp, 15.bmp, 28.bmp and 41.bmp are all twos, etc.)
4. Actualize the player's score each time he presses “card”.
5. If the score goes over 42, display a html text to the player, stating that he lost the game.
6. When the player presses “stay”, the dealer will add 6 random cards to himself (for a total of 7 cards).
7. Actualize the dealer's score when the seven cards are on the table.
8. If the dealer's score is over the player's score and bellow 42, then display a html text stating that the player lost the game ; else, the text should say “congratulation” to the player.
9. Make the dealer more clever. For instance, the dealer should only add cards to himself when his score is lower than the player's one.
10. Add a “start again” button.

```
function createImg(path) {
   var img = document.createElement('img');
   img.src = path;
   return img;
}

function addImgInDiv() {
   var path = "img/1.BMP"
   var newImg = createImg(path);
   var divJS = document.getElementById('myDiv');
   divJS.appendChild(newImg);
}
```