2015

Team "Manhattan"

Telerik

7/31/2015

Manhattan Project



* Aleksandra92 - Александра Стойчева

**Team Members:**



* Logic for cards
* ShuffleCards method
* Put Pot on the correct place
* Deal three pots(group)
* Pot Dealer
* Get three pots
* Cards picture animation
* IvanR90 - Иван Русев
* vvn050 - Владислав Николов
* Validation Input
* Check if null
* Check if a number
* Check if a whole number
* Check if NaN
* Check whether the input is within the right range
* Custom JavaScript Alert
* Design of the message
* Logic implementation included – Event Listener etc.
* When active, disables the main page
* niya.omerska - Ния Омерска
* Draw cards backs
* Last card animation
* setTimeout for cards drawing
* setIntervals for cards drawing
* Three Pots(group)
* mivancheva - Маргарита Иванчева
* Page on load logic
* Following up the already created styles
* Border animation styles
* SVG and CSS used
* Employment of the game design and layout
* SideBar options menu
* jQuery used
* Presentation and Documentation Files
* divided.zero - Николай Илиев
* draw cards on canvas
* card images with SVG
* play sound
* get random card
* fill deck with cards
* Three pots (group)
* nlpcsh - Николай Петров
* animation text with Raphael

Goal of the game:

**Feel the magic of the cards:**



**Behind the Curtains:**



Have you heard the saying that New York never sleeps? Now just think about what can be said about Manhattan. A place where everything is moving in such a high speed that we can barely remember the good old fashioned card tricks from the not so distant past.

No wonder each one of us has played with cards. We have shuffled them, passing them from hand to hand, trying to impress our audience (occasionally imaginary one). Sometimes we get lucky enough and in the end everyone watching is satisfied with the trick and looking forward to see it again.

Now imagine all this atmosphere transferred to your computer screen. All the mystery hidden behind carefully designed logic. All the tricks happening right in front of you. So let us tell you the secret…..

The game is played with 27 cards. At first you have to think about a number from 1 to 27, and consequently write it down. After that you are prompt to choose your favorite card from the deck. Don’t forget it, because you have to search for it three times within the presented pots. In the end your card magically appears in the deck on the position you entered. How cool is that?

OnLoad Page:

A canvas element is used, where with javaScript a short list of instructions and a button are included. Once the button is clicked the player is redirected to the page where a number should be entered. The old canvas is set to be not displayed and on the screen appears a new container.

The animation with the card is made using SVG element, following a concrete previously created path. Once the user is redirected the animation display is set to none.

For the animated border SVG element is used too. The effect of moving line is created with the help of some CSS.

## Enter a number page:

**Behind the Curtains (continue):**



Here the container consists of an input text box and a button. Once clicked the button first triggers a validation input algorithm. The value of input is check for certain conditions. If the value is incorrect the deployment of the program stops and a custom windows alert is displayed (javaScript is used, because windows.alert cannot be customized). The message makes the page inactive until “OK” button is clicked. If the entered value is the correct format, the execution of the program continues.

## Deal the Deck page:

For the images of the cards different SVG files are used. After that all the possible cards are stored in an array. With a getRandomCard function a card is pulled from the deck and displayed on the screen. Once displayed, the card is removed from the array and the next getRandomCard is invoked.

Each card is displayed using a custom setInterval function, which once the end of the deck is reached, is cleared. In this way a smooth transition between each of the cards is achieved, which significantly improves the user experience.

Here is the moment, when the user must choose his card and click the button in order to continue. Once the button is clicked the container is cleared and ready for the presentation of the pots.

## Three pots page:

All the cards are dealt in three pots and then the user chooses three times the pot where his card is. To the number that the player had previously entered is set a code. For each of the dealing to the pot the code is used to put the pot with the chosen card on the right position for guessing. Finally the pots are put together and the process of distribution begins again.

Once the user clicks for the third time, the container is cleared and ready for the answer execution of the code.

## The Answer Page:

**Behind the Curtains (continue):**



Here the cards are displayed with their backs, and the only one displayed with its face is the chosen card. It is placed on the exact location, which is taken from the user input. The cards are placed as they have last appeared in the three pots page are pushed, making an array with the whole deck.

As soon as all the cards are displayed, they are turned, using the turnCards function. Each card is displayed with its SVG image face at a constant setInterval function.

## Rotation of the chosen card:

Finally, the chosen card is zoomed and rotated, using rotateMagicCard function.

## SideBar Options Menu:

A jQuery library is used for the creation of the sidebar options menu. There are two buttons included. The first one is the “Start Over” button. It reloads the page, and redirects the player to the begging of his journey. The second button is “How to play”. The button triggers the custom javaScript message, like the one used for the validation of the user input. It deactivates the page, and the player should click on the “OK” button in order to continue.

## Game Instruction functionality:

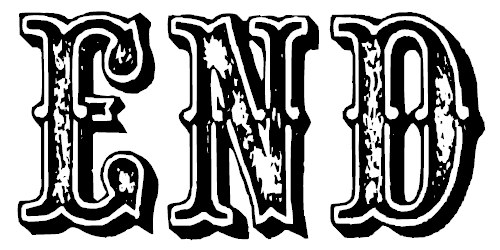
Each instruction uses a Raphael library to be position and displayed.

## GitHub repository:

<https://github.com/aleksandra992/Team-Work-Manhattan>

## Presentation Link:

<https://prezi.com/gl_8wzeig0hh/manhattan/>



**Resources:**

