BLACKJACK

BY: SANTIAGO MARIN

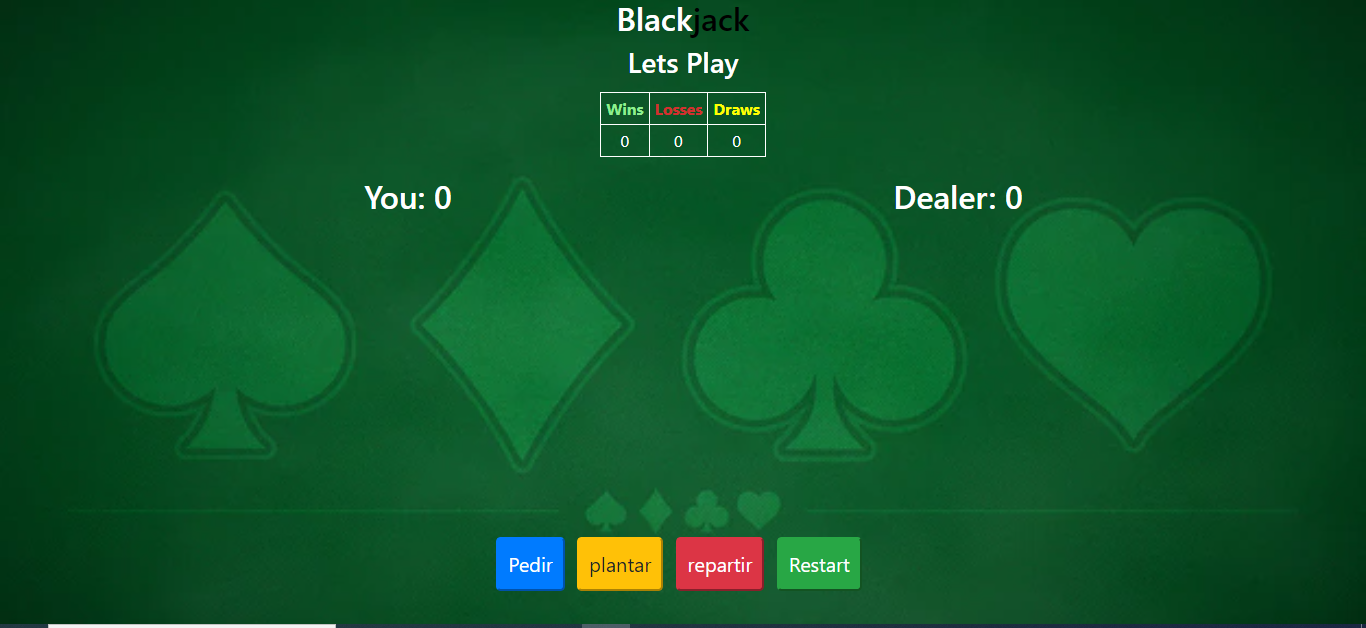
INTRODUCCION

El Blackjack es un juego de cartas popular en todo el mundo, en el que el objetivo es obtener una mano con un valor total de 21 puntos o lo más cerca posible a este valor, sin pasarse. Cada carta tiene un valor numérico y el jugador debe tomar decisiones estratégicas para determinar si debe pedir otra carta o mantener su mano actual. Nos gusta a todos desde ludópatas, adictos al dinero, a las apuestas o simplemente por diversión entre amigos.

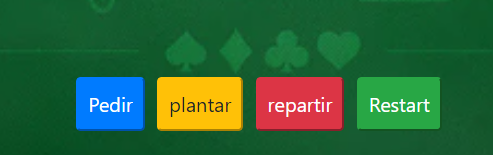
En este proyecto, utilicé JavaScript vanilla para crear una versión funcional del juego con una interfaz amigable para el usuario y una jugabilidad atractiva. Además, diseñé la estructura visual del juego utilizando HTML y CSS. Durante el proceso de desarrollo, utilicé mi conocimiento de programación en JavaScript para crear funciones que manipularan el DOM y actualizaran la interfaz del usuario en tiempo real. A través de este proyecto, espero demostrar mi capacidad para crear soluciones efectivas y creativas utilizando tecnologías de programación web.

Realización del proyecto

Comenzamos con una interfaz atractiva y sencilla para facilitar la experiencia del usuario en el momento de usar la aplicación.



Como podemos apreciar tiene una tabla de estadísticas la cual nos indica nuestras “wins” luego nuestros “loses” y nuestros “draws” o lo que sería nuestro empate en el sentido anglosajón de la palabra, no son nada nuevas estas palabras para amantes de blackjack y ni siquiera es necesario estar cercano al lenguaje anglosajón para saber significado de estas 3 palabras.

Decidí poner 4 botones básicos y fáciles de usar para que el usuario tenga una experiencia de juego más dinámica y divertida.

Tenemos un botón para pedir nuestra carta, para plantarnos, el repartir es para reiniciar el juego, el botón restart reinicia absolutamente todo incluso nuestras estadísticas.

INICIO DEL JUEGO

Iniciamos el juego procederé a pedir cartas a ver si tengo suerte de sacar ese valioso 21 para ganar mi partida contra la casa.



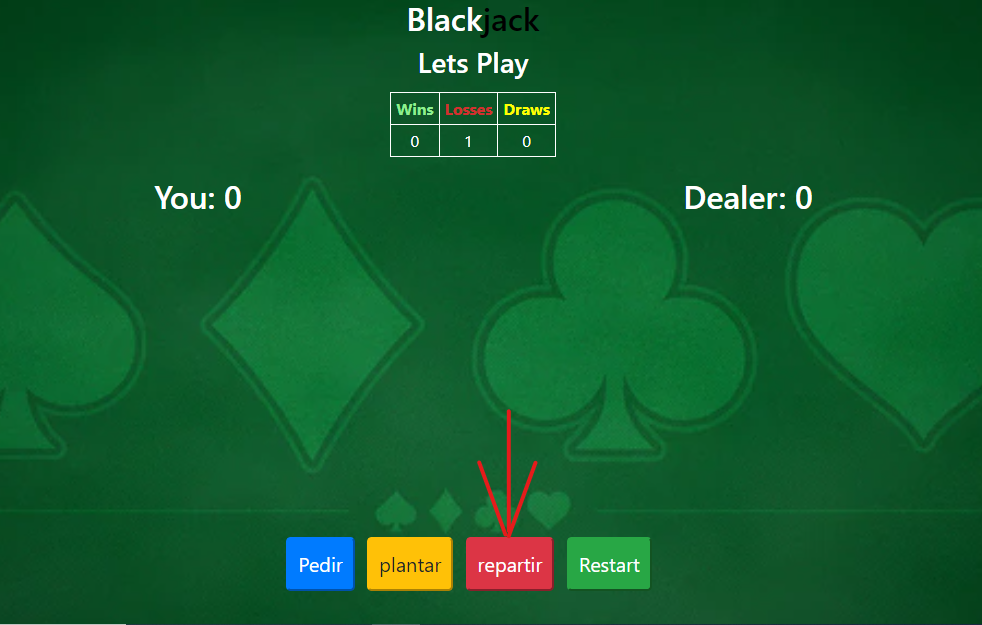
Me he pasado de 21 significa que me he pasado.

Se me ocurrió la elegante idea de señalizar en mi juego al jugador que se ha pasado o “BUST” como se le llama en el blackjack. Está señalizado en rojo para que sea un aviso importante y notable para el usuario.

Después de haberme pasado he decidido plantarme ahora la casa muestra sus cartas y efectivamente me ha ganado con un 20, ahora mi derrota es marcada con “loss” dejando el marcador a favor de la casa



Hasta este punto tengo una derrota y he decidido seguir jugando, si quiero un nuevo juego procedo a presionar el botón de repartir

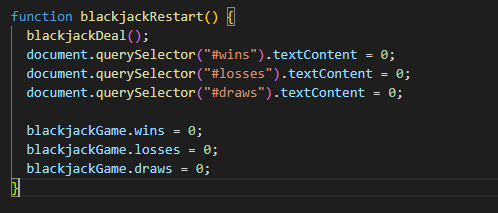


Ahora las cartas han desaparecido para así pedir de nuevo carta y probar nuestra suerte.

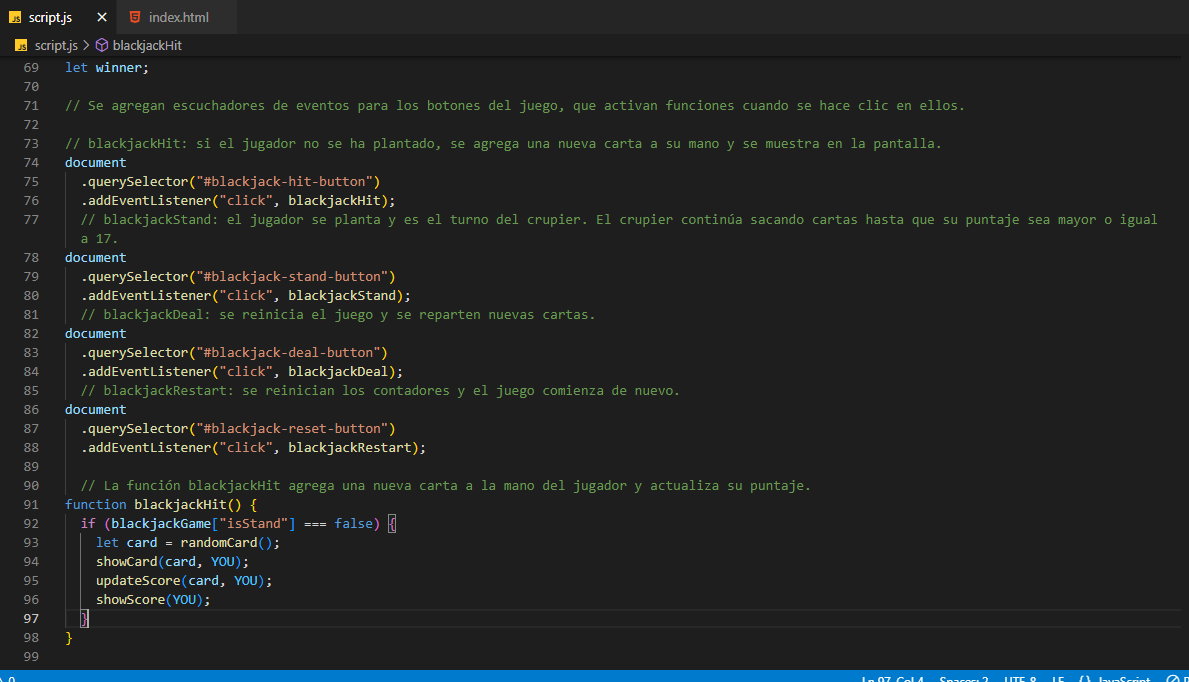


Ahora que le he ganado a la casa plantándome a tiempo en nuestras estadísticas se marcará una victoria a favor de nostros con el mensaje en verde de “You Won”.

Ahora el cuarto botón de restart reinicia nuestras estadísticas a 0 dejando el juego totalmente en 0 wins, 0 loses y 0 empates, muestro como funciona en la siguiente imagen.

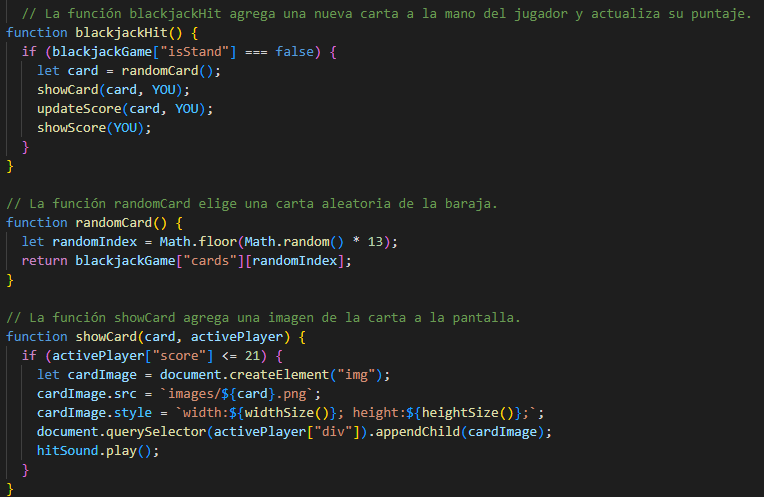


Muestro el funcionamiento de cada uno de los 4 botones que están anclados a las funciones gracias a los escuchadores de eventos.



Cada uno de nuestros botones tiene una función ya sea para pedir plantarse, reiniciar el juego como he mostrado con el “repartir” o simplemente reiniciar las estadísticas con el restart.

PARTE DEL CODIGO

Ya he mostrado mucho de la parte visual, es hora de ver la parte del código

La función blackjackHit es llamada cuando el jugador hace clic en el botón "Hit" para pedir otra carta. Primero, verifica si el jugador ya ha decidido "Stand" (mantenerse con sus cartas actuales) utilizando la propiedad isStand del objeto blackjackGame. Si el jugador no ha decidido "Stand", entonces la función utiliza la función randomCard para obtener una carta aleatoria de la baraja, la agrega a la mano del jugador usando la función showCard, actualiza el puntaje del jugador usando la función updateScore, y finalmente muestra el puntaje actual del jugador en la pantalla usando la función showScore.

La función randomCard elige una carta aleatoria de la baraja del juego. Utiliza la función Math.random() para obtener un número aleatorio entre 0 y 1, lo multiplica por 13 (ya que hay 13 cartas en cada palo), y luego redondea hacia abajo utilizando la función Math.floor() para obtener un índice de carta aleatorio entre 0 y 12. Luego, utiliza este índice para acceder a la carta correspondiente en la matriz cards del objeto blackjackGame

La función`showCard agrega una imagen de la carta a la pantalla. Primero verifica si el puntaje actual del jugador es menor o igual a 21 (si es mayor, el jugador ha perdido y no se deben mostrar más cartas). Si el puntaje es menor o igual a 21, la función crea un nuevo elemento de imagen (<img>) utilizando document.createElement("img"), establece su atributo src para que muestre la imagen de la carta correspondiente, y establece su estilo para que tenga un ancho y alto definido por las funciones widthSize y heightSize respectivamente. Luego, agrega este elemento de imagen al elemento HTML correspondiente a la mano del jugador (usando la propiedad div del objeto activePlayer) utilizando `document.querySelector(...).appendChild(cardImage). Finalmente, reproduce un sonido de "hit" utilizando la variable hitSound.

Me parece totalmente importante explicar el funcionamiento del del botón “repartir” ya que este es importante y resetea el juego sin afectar el contador de las estadísticas.



La función blackjackDeal se encarga de reiniciar el estado del juego y mostrar una nueva mano. Primero, comprueba si el juego ha terminado (`blackjackGame["isTurnsOver"] === true`) para poder comenzar una nueva ronda. Si se cumple esta condición, la función realiza las siguientes acciones:

- Selecciona todas las imágenes de las cartas del jugador y del crupier para poder eliminarlas de la pantalla y comenzar una nueva ronda.

- Cambia el color del texto que muestra el resultado del juego a blanco.

- Reinicia los puntajes del jugador y del crupier a 0.

- Actualiza los valores de los elementos HTML que muestran los puntajes a 0 y cambia el color del texto a blanco.

- Actualiza el texto que muestra el resultado del juego a "Let's Play".

- Elimina las imágenes de las cartas del jugador y del crupier de la pantalla.

- Restablece las propiedades isStand y pressOnce del objeto blackjackGame a false.

- Establece isTurnsOver a false, lo que significa que el juego ha comenzado de nuevo y no ha terminado todavía.

En resumen, la función blackjackDeal es responsable de limpiar la pantalla y reiniciar el estado del juego para que se pueda comenzar una nueva ronda.

He llegado al final de mi informe, me siento cómodo explicando cada parte del funcionamiento de mi blackjack y sus partes.

CONCLUSION

Este proyecto enseña a hacer paginas o aplicaciones disfrutables con mucha interacción con el usuario, debemos hacer paneles sencillos pero a la vez atractivos para que estos sigan en nuestra aplicación durante más tiempo, debemos mostrarles sus logros como fracasos mediante estadísticas y mostrar avisos importantes en pantalla cada vez que ocurra una acción o evento para que estos se den cuenta de lo que pasa en nuestro juego, así podremos enganchar más a nuestros usuarios.