

proyecto_black_jack\assets\js\juego.js

```
1 // Inicialización del deck (mazo de cartas) como arreglo vacío
2 let deck = [];
3
4 // Tipos de cartas (corresponden a los palos: Clubs, Diamonds, Hearts, Spades)
5 const tipos = ["C", "D", "H", "S"];
6
7 // Cartas especiales (As, Jota, Reina, Rey)
8 const especiales = ["A", "J", "Q", "K"];
9
10 // Puntos del jugador y de la computadora inicializados en 0
11 let puntosJugador = 0;
12 let puntosComputadora = 0;
13
14 // Referencias a elementos del HTML para los botones
15 const btnPedir = document.querySelector("#btnPedir");
16 const btnDetener = document.querySelector("#btnDetener");
17 const btnNuevo = document.querySelector("#btnNuevo");
18
19 // Referencias a elementos del HTML para las cartas
20 const divCartasJugador = document.querySelector("#jugador-cartas");
21 const divCartasComputadora = document.querySelector("#computadora-cartas");
22
23 // Referencias a elementos del HTML para mostrar los puntos
24 const puntosHTML = document.querySelectorAll("small");
25
26 // Función para crear un nuevo deck
27 const crearDeck = () => {
28   // Crear cartas numéricas del 2 al 10 para cada tipo de palo
29   for (let i = 2; i <= 10; i++) {
30     for (let tipo of tipos) {
31       deck.push(i + tipo);
32     }
33   }
34
35   // Crear cartas especiales para cada tipo de palo
36   for (let tipo of tipos) {
37     for (let esp of especiales) {
38       deck.push(esp + tipo);
39     }
40   }
41
42   // Barajar el deck
43   deck = _.shuffle(deck);
44   return deck;
45 };
46
47 // Llamar a crearDeck para inicializar el juego
48 crearDeck();
49
50 // Función para pedir una nueva carta
51 const pedirCarta = () => {
52   if (deck.length === 0) {
53     throw "No hay cartas en el deck";
54   }
55   return deck.pop();
56 };
57
```

```

58 // Función para determinar el valor de una carta
59 const valorCarta = carta => {
60     const valor = carta.substring(0, carta.length - 1);
61     return isNaN(valor) ? (valor === "A" ? 11 : 10) : valor * 1;
62 };
63
64 // Turno de la computadora
65 const turnoComputadora = puntosMinimos => {
66     do {
67         const carta = pedirCarta();
68         puntosComputadora += valorCarta(carta);
69         puntosHTML[1].innerText = puntosComputadora;
70
71         // Crear imagen de la carta y añadirla al HTML
72         const imgCarta = document.createElement("img");
73         imgCarta.src = `assets/cartas/${carta}.png`; // Ejemplo: 3H, JD
74         imgCarta.classList.add("carta");
75         divCartasComputadora.append(imgCarta);
76
77         if (puntosMinimos > 21) {
78             break;
79         }
80     } while (puntosComputadora < puntosMinimos && puntosMinimos <= 21);
81
82     // Evaluar el resultado después de un breve retraso
83     setTimeout(() => {
84         if (puntosComputadora === puntosMinimos) {
85             alert("Nadie gana :(");
86         } else if (puntosMinimos > 21) {
87             alert("Computadora gana");
88         } else if (puntosComputadora > 21) {
89             alert("Jugador Gana");
90         } else {
91             alert("Computadora Gana");
92         }
93     }, 100);
94 };
95
96 // Eventos para los botones
97 btnPedir.addEventListener("click", () => {
98     const carta = pedirCarta();
99     puntosJugador += valorCarta(carta);
100     puntosHTML[0].innerText = puntosJugador;
101
102     // Crear imagen de la carta y añadirla al HTML
103     const imgCarta = document.createElement("img");
104     imgCarta.src = `assets/cartas/${carta}.png`; // Ejemplo: 3H, JD
105     imgCarta.classList.add("carta");
106     divCartasJugador.append(imgCarta);
107
108     // Condiciones para finalizar el juego
109     if (puntosJugador > 21) {
110         console.warn("Lo siento mucho, perdiste");
111         btnPedir.disabled = true;
112         btnDetener.disabled = true;
113         turnoComputadora(puntosJugador);
114     } else if (puntosJugador === 21) {
115         console.warn("21, genial!");
116         btnPedir.disabled = true;
117         btnDetener.disabled = true;

```

```
118     turnoComputadora(puntosJugador);
119 }
120 });
121
122 btnDetener.addEventListener("click", () => {
123     btnPedir.disabled = true;
124     btnDetener.disabled = true;
125     turnoComputadora(puntosJugador);
126 });
127
128 btnNuevo.addEventListener("click", () => {
129     // Reiniciar el juego
130     console.clear();
131     deck = [];
132     crearDeck();
133     puntosJugador = 0;
134     puntosComputadora = 0;
135     puntosHTML[0].innerText = 0;
136     puntosHTML[1].innerText = 0;
137     divCartasComputadora.innerHTML = "";
138     divCartasJugador.innerHTML = "";
139     btnPedir.disabled = false;
140     btnDetener.disabled = false;
141 });
```