



programación visual

uno



Materia: programación visual

Profesor: Torres Servín Emmanuel

Alumno: Cosme Santes Carlos Gerardo

Matricula: 131124250

Grupo: 4322IS

Carrera: Ingeniería en Software

Contenido

Introducción.....	2
mockup	3
Componentes visuales y no visuales.....	4
Explicación del código	5

Introducción

En el siguiente texto se presentará el desarrollo del juego de mesa uno.

Explicación del juego:

El juego consiste en 108 cartas repartidas

- 19 cartas azules un 0 y pares del 1 al 9
- 19 cartas amarillas un 0 y pares del 1 al 9
- 19 cartas verdes un 0 y pares del 1 al 9
- 19 cartas rojas un 0 y pares del 1 al 9
- 8 cartas especiales roba dos 2 de cada color
- 8 cartas especiales cambio de sentido dos de cada color
- 8 cartas especiales pierden el turno 2 cartas de cada color
- 4 cartas especiales comodín cambio de color
- 4 cartas especiales comodín cambio de color cuatro

Se pasa a explicar las reglas del juego

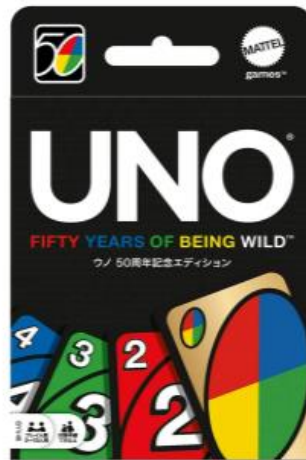
- Los jugadores no podrán poner una carta que sea diferente en color o número de la carta que este sobre la mesa
- Cuando un jugador tenga solo una carta deberá gritar uno, indicando que solo le falta una carta para ganar
- Si el jugador no tiene cartas del mismo color o número deberá agarrar continuamente cartas hasta que pueda bajar

Reglas especiales para las cartas

- El comodín cambio color cuatro tiene la singular regla, el jugador no debe tener una carta del mismo color o número que la carta que este en la mesa. Si el jugador aun teniendo cartas similares en su baraja juega a la carta y otro descubre su mentira la carta tendrá efecto en él, si no es descubierto no pasa nada.

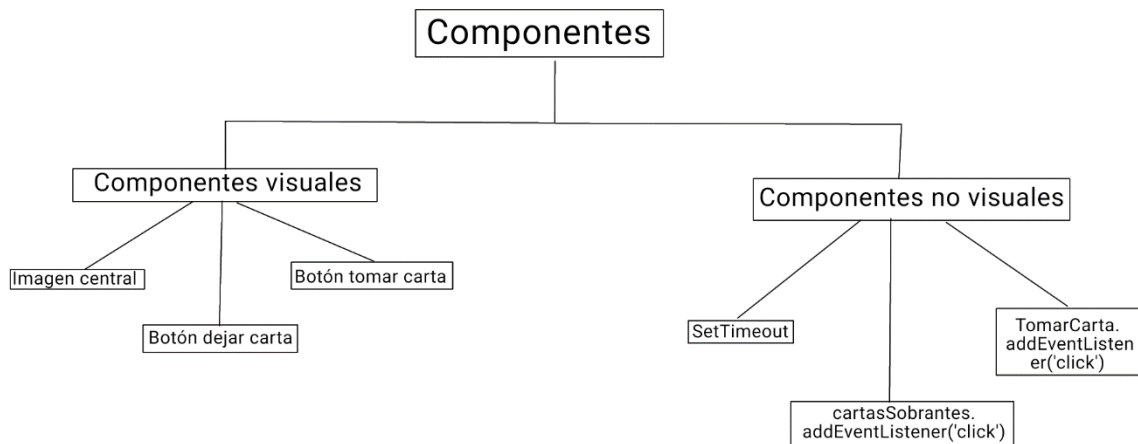
Finalizando la explicación se procede a explicar el código

mockup

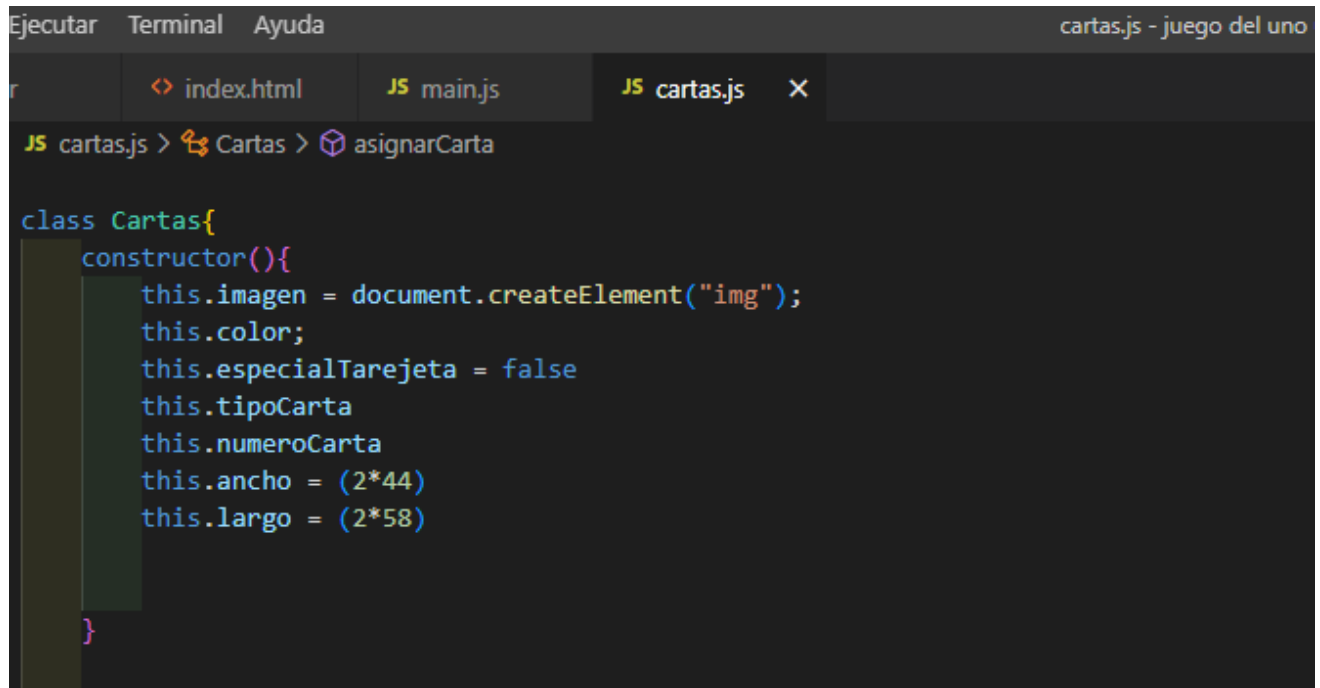


Componentes visuales y no visuales

Componentes visuales	Componentes no visuales
<ul style="list-style-type: none">• Imágenes de las cartas• Botones de tomar y dejar cartas• Imagen centrar del mazo	<ul style="list-style-type: none">• Settimeout para mostrar la carta cierto periodo después• Evento click en la carta central• Evento click en el botón tomar carta• Evento click en el botón de dejar carta



Explicación del código



```
Ejecutar Terminal Ayuda cartas.js - juego del uno
index.html main.js cartas.js X
JS cartas.js > Cartas > asignarCarta

class Cartas{
  constructor(){
    this.imagen = document.createElement("img");
    this.color;
    this.especialTarejeta = false
    this.tipoCarta
    this.numeroCarta
    this.ancho = (2*44)
    this.largo = (2*58)
  }
}
```

Se crea una clase con los atributos

- Imagen: la cual contiene la imagen que se mostrara en pantalla
- Color: guarda el color de la imagen en string
- especialTarjeta: booleano que indica si es una carta especial
- tipoCarta: guarda el color de la carta en forma número del 0 al 1
- numeroCarta: nos indica el numero de la carta
- ancho: la dimensión en pixeles de la carta imagen
- largo: la dimensiones en pixeles a lo largo de la carta imagen

la clase cuenta con un método en cual asigna la carta al azar

```

asignarCarta(random,numeroCarta,distanciaCarta,display = 'none'){
  if(numeroCarta >9){
    this.especialTarejeta = true
  }
  if(numeroCarta <=12){
    if(random == 0){
      this.imagen.src = "src/img/amarillo"+ numeroCarta + ".png";
      this.color = "Amarillo"
    }
    else if(random == 1){
      this.imagen.src = "src/img/azul"+ numeroCarta + ".png"
      this.color = "Azul"
    }
    else if(random == 2){
      this.imagen.src = "src/img/rojo"+ numeroCarta + ".png"
      this.color = "rojo"
    }
    else if(random == 3){
      this.imagen.src = "src/img/verde"+ numeroCarta + ".png"
      this.color = "verde"
    }
  }
  if(numeroCarta == 13){
    this.imagen.src = "src/img/especialtoma4.png"
    this.tipoCarta = 4
    this.color = "negro"
  }else if(numeroCarta == 14){
    this.imagen.src = "src/img/especialtoma41.png"
    this.tipoCarta = 5
    this.color = "negro"
  }

  this.imagen.style.top = (window.innerHeight-this.largo)+ "px"
  this.imagen.style.left = ((300+distanciaCarta)-this.ancho)+ "px"
  this.imagen.style.width = this.ancho+"px";
  this.imagen.style.height = this.largo + "px";
  this.imagen.style.position = "absolute"
  this.imagen.style.textAlign = "center"
  this.imagen.style.display = display
  document.body.appendChild(this.imagen)
}

```

el método asignar carta recibe como parámetros un random que es un numero ramdon, el numerocarta es igual un numero random del 0 al 15 y distaciaCarta es un numero que nos indica en que posición aparecerá la carta y por ultimo display que es un parámetro con un valor por defecto que nos indica si muestra la imagen o no.

Los if no ayudan a asignar las cartas comparando primero si es una carta especial y luego si es menor a 12, dependiendo su numero en numeroCarta asigna un color a la carta y una imagen.

```
Ejecutar Terminal Ayuda main.js - juego del uno uno - Visual Studio Code
index.html JS main.js JS cartas.js
s > JS main.js > dejarCartaMi
//declaracion de las variables del juego
let cartasSobrantes = document.getElementById("imagen");
let tomarCarta = document.getElementById("tomar carta");
let dejarCarta = document.getElementById("dejar carta");

let distanciaCarta = 1
let cartaCambiante = new Cartas()
var cartaAlAzar = new Array(2);
cartaAlAzar[0] = cartaCambiante
cartaAlAzar[0].tipoCarta = Math.floor(Math.random()*4)
cartaAlAzar[0].numeroCarta = Math.floor(Math.random()*15)
cartaAlAzar[0].asignarCarta(cartaAlAzar[0].tipoCarta,cartaAlAzar[0].numeroCarta,distanciaCarta,'none');

var numeroCartaG
var ColorCartaG
var cartaEspecialG
var direccionCartaG
let espacioRecorrido = 350

let contador = 0
let jugador1 = 7
let jugador2 = 7

let colorCarta
let numeroCartaLista

let mano1 = new Array(3)
let mano2 = new Array(3)

let temple = new Array(5)
```

cuantas cartas existen en la baraja actualmente.

verificarCartas es un método que recibe como parámetro tipo carta Carta y numeroCarta en la que atreves de if se nos verifica si la carta asignada por el método asignarCarta de la clase Cartas según sea el caso manda un true o un false

```
* @param {entero} tipoCarta
* @param {entero} numeroCarta
* @description elimina las malas prcticas
* @returns true or false
*/
function verificarCartas(tipoCarta,numeroCarta){
    if(tipoCarta <=3 && numeroCarta < 13){
        if(temple[tipoCarta][numeroCarta] != 0){
            temple[tipoCarta][numeroCarta]--
            return true
        }else{
            return false
        }
    }else{
        if(numeroCarta == 13){
            if(temple[4][0] != 0){
                temple[4][0]--
                return true
            }else{
                return false
            }
        }else if(numeroCarta == 14){
            if(temple[4][1] != 0){
                temple[4][1]--
                return true
            }else{
                return false
            }
        }
    }
}
```

se declaran las variables las tres [primeras son elementos del html dos botones y una imagen en la parte central de la pantalla

los array en los que se guarda la imagen

las variables globales nos ayudan a asignar números a la carta que se muestra en el centro de la pantalla

jugador1 y jugador2 nos indican cuantas cartas tiene cada jugador esta aumenta y disminuye según las circunstancias, los array mano1 y mano2 es donde se guardan los objetos(cartas de cada jugador) y temple indica


```

a su vez asigna valores
*/
* @return {array} objetos
*/
function repartirCartas(mano,contador,distanciaCarta,mostrarCartas,n = 7){
    do{
        let carta2 = new Cartas();

        carta2.tipoCarta = Math.floor(Math.random()*4)
        carta2.numeroCarta = Math.floor(Math.random()*15)
        let verificar = verificarCartas(carta2.tipoCarta, carta2.numeroCarta)
        if(verificar == true){
            carta2.asignarCarta(carta2.tipoCarta, carta2.numeroCarta, distanciaCarta, mostrarCartas);
            distanciaCarta+=50
            mano[contador] = carta2
        }else{
            contador--
        }
    }

    contador++
}while(contador != n)

```

repartirCartas método que nos ayuda a repartir las cartas entre los jugadores en este caso entre los dos jugadores

tomarCartaT el método crea objetos continuamente y los guarda en el array del jugador antes verifica si la carta existe en el maso si existe lo guarda y si no retrocede el contador y repite el procedimiento

```

* @returns
*/
function tomarCartaT(n,display = 'inline'){
    let verificar
    do{
        let carta2 = new Cartas();

        carta2.tipoCarta = Math.floor(Math.random()*4)
        carta2.numeroCarta = Math.floor(Math.random()*15)
        verificar = verificarCartas(carta2.tipoCarta, carta2.numeroCarta)
        if(verificar == true){
            carta2.asignarCarta(carta2.tipoCarta, carta2.numeroCarta, espacioRecorrido, display);

            espacioRecorrido+=50
            if(n == 1){
                return carta2;
            }else if(n==2){
                return carta2.imagen.src
            }
        }
    }

}while(verificar != true)

```

```
cutar Terminal Ayuda main.js - juego del
index.html JS main.js X JS cartas.js
JS main.js > tomarCartaT
*/
function dejarCartaMi(mano1,numeroCartaLista){
  if(cartasAlAzar[0].numeroCarta == 10){
    alert("alert turno bloqueado")
    cartasAlAzar[0] = tomarCartaT(1,'none')
    cartasSobrantes.src = cartasAlAzar[0].imagen.src
    return
  }else if(cartasAlAzar[0].numeroCarta == 12){
    alert("el castigo tomar dos cartas")
    mano1.push(tomarCartaT(1,'inline'))
    mano1.push(tomarCartaT(1,'inline'))
    cartasAlAzar[0] = tomarCartaT(1,'none')
    cartasSobrantes.src = cartasAlAzar[0].imagen.src
    return
  }else if(cartasAlAzar[0].numeroCarta == 14){
    alert("el castigo tomar cuatro cartas y pierdes turno")
    mano1.push(tomarCartaT(1,'inline'))
    mano1.push(tomarCartaT(1,'inline'))
    mano1.push(tomarCartaT(1,'inline'))
    mano1.push(tomarCartaT(1,'inline'))
    cartasAlAzar[0] = tomarCartaT(1,'none')
    cartasSobrantes.src = cartasAlAzar[0].imagen.src
    return
  }
  if((mano1[numeroCartaLista].tipoCarta == cartasAlAzar[0].tipoCarta) ||
    (mano1[numeroCartaLista].numeroCarta == cartasAlAzar[0].numeroCarta)
  ){
    cartasAlAzar[0] = mano1[numeroCartaLista]
    cartasSobrantes.src = mano1[numeroCartaLista].imagen.src
    mano1[numeroCartaLista].imagen.style.display = 'none'
    mano1.splice(numeroCartaLista,1)
    cartasSobrantes.src = cartasAlAzar[0].imagen.src
    jugador1--
  }else{
    if(mano1[numeroCartaLista].numeroCarta == 10){
      cartasAlAzar[0] = mano1[numeroCartaLista]
      cartasSobrantes.src = cartasAlAzar[0].imagen.src
      return
    }else if(mano1[numeroCartaLista].numeroCarta == 12){
      cartasAlAzar[0] = mano1[numeroCartaLista]
      cartasSobrantes.src = cartasAlAzar[0].imagen.src
      return
    }else if(mano1[numeroCartaLista].numeroCarta == 14){
      cartasAlAzar[0] = mano1[numeroCartaLista]
      cartasSobrantes.src = cartasAlAzar[0].imagen.src
      return
    }
  }
  alert("la carta no cuenta con las especificaciones dadas")
}
```

dejarCartaMi

método que recibe un array y una posición en el arreglo que es ingresada por el jugador

primero verifica si la carta que está actualmente es alguna carta especial y su efecto sobre el jugador de lo contrario se verifican las condiciones;

el numero de la carta es igual al numero de la carta que está en la mesa

el color de la carta es igual al numero de la carta que está en la mesa

si ninguna de estas condiciones se cumple se verifica si no se esta poniendo una carta especial, pero si no manda un mensaje de error indicando que la carta no cuenta con las condiciones para ser puesta en el maso

```

cartasSobrantes.addEventListener('click', function (e) {
    turnos(cartasSobrantes,jugador2,2,mano2)
    cartasSobrantes.src = cartaAlAzar[0].imagen.src
})
tomarCarta.addEventListener('click',function(){
    mano1.push(tomarCartaT(1))
    jugador1++
})

dejarCarta.addEventListener('click', function () {
    if(mano1.length == 0){
        alert("HAS GANADO")
    }

    alert("en que posicion se encuentra la carta que quiere dejar")
    numeroCartaLista = prompt("en que posicion se encuentra la carta que quiere dejar")
    dejarCartaMi(mano1,numeroCartaLista-1)
    console.log(mano1)
})

repartirCartas(mano1,contador,distanciaCarta,'inline')
repartirCartas(mano2,contador,distanciaCarta,'none')

```

se dejara

Al final se muestran los eventos en los que se podrá interactuar con el juego

cartasSobrantes tiene el evento click cuando se da un click en la superficie de la carta el jugador 2 inicia una secuencia donde pone una carta o toma una carta.

tomarCarta es un botón en el que se llama al método tomarCarta y se agrega con un push al arreglo de la mano1

dejarCarta evento que manda a llamar al método dejarCartaMi y pregunta la posición de la carta que