

UNIVERSIDAD CATÓLICA BOLIVIANA “SAN PABLO”

INGENIERÍA DE SISTEMAS

SIS-241 TECNOLOGÍAS WEB



JUEGOS

Integrantes:

Apaza Aguilar Chris Alan
Alvarado Callisaya Diego Cesar
Avila Flores Juan Carlos
Flores Ayala Alejandro Paul

Docente:

Ing. Marcelo Ticona

Fecha de entrega:

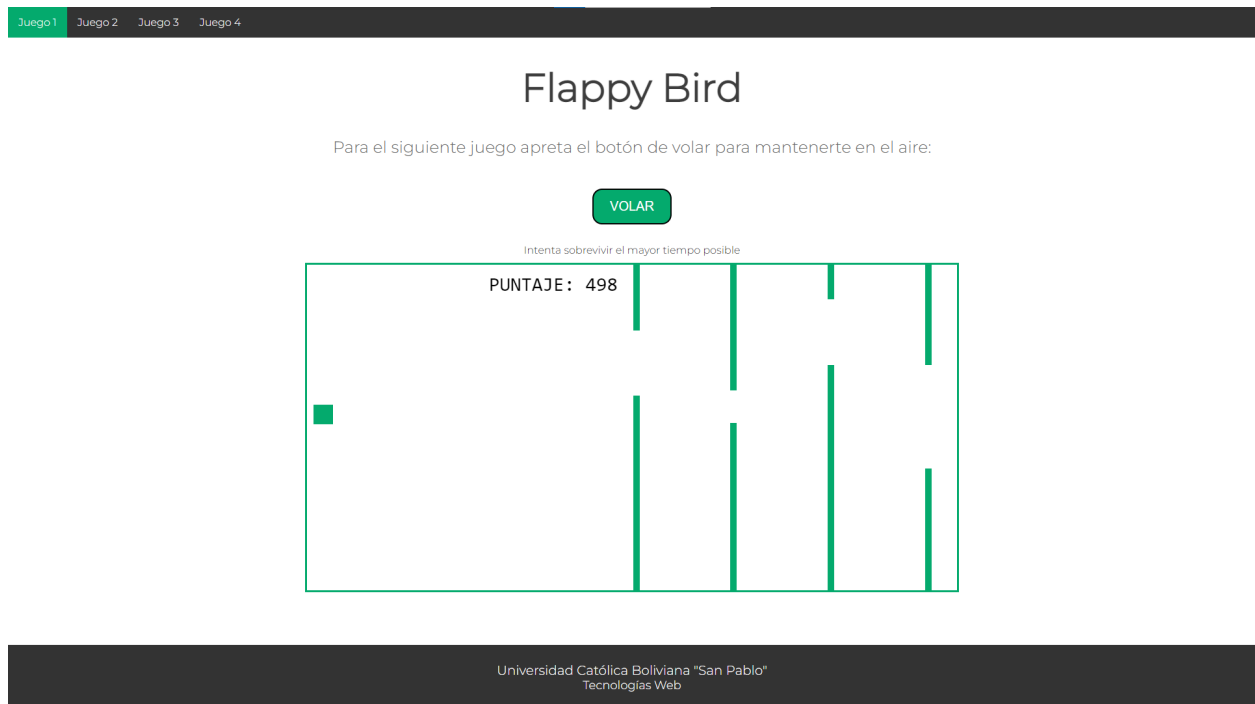
29 / 09 / 2022

La Paz – Bolivia

Tecnologías Web (Juegos)

Juego 1 (Juan Carlos Avila Flores)

Este juego se encuentra inspirado en el famoso juego de Flappy Bird, en este el jugador debe apretar un botón para mantenerse en el aire, para poder conseguir el mayor puntaje posible debe evitar todos los obstáculos que aparecen en la pantalla y si este choca con alguno de ellos el juego termina.



Para este juego primero se deben declarar dos componentes principales, el cuadrado verde que controla el jugador y el puntaje que va a aumentando conforme el juego avanza:

```
var myGamePiece;  
var myObstacles = [];  
var myScore;  
  
function startGame() {  
  myGamePiece = new component(  
    30,  
    30,  
    "#04aa6d",  
    10,  
    120  
  );  
};
```

```

myGamePiece.gravity = 0.05;
myScore = new component(
    "30px",
    "Consolas",
    "black",
    280,
    40,
    "text"
);
myGameArea.start();
}

```

Luego se debe declarar el elemento como tal donde se colocaran estos componentes, en este caso este elemento se llama "canvas":

```

var myGameArea = {
    canvas: document.createElement("canvas"),
    start: function () {
        this.canvas.width = 1000;
        this.canvas.height = 500;
        this.context = this.canvas.getContext("2d");
        document.body.insertBefore(
            this.canvas,
            document.body.childNodes[9]
        );
        this.frameNo = 0;
        this.interval = setInterval(updateGameArea, 20);
    },
    clear: function () {
        this.context.clearRect(
            0,
            0,
            this.canvas.width,
            this.canvas.height
        );
    },
};

```

En este se especifica el tamaño, que en este caso es de 1000 x 500 píxeles, se especifica la ubicación del mismo que es el childNode[9], y también se especifica que es un contexto "2d".

Para los diferentes obstáculos se declaran como funciones para que aparezcan los mismo de manera random y con un tamaño diferente:

```

minGap = 50;
maxGap = 200;
gap = Math.floor(
    Math.random() * (maxGap - minGap + 1) + minGap
);
myObstacles.push(
    new component(10, height, "#04aa6d", x, 0)
);
myObstacles.push(
    new component(
        10,
        x - height - gap,
        "#04aa6d",
        x,
        height + gap
    )
);

```

Finalmente se coloca en puntaje para que este aumente conforme el tiempo avanza:

```

myScore.text = "PUNTAJE: " + myGameArea.frameNo;
myScore.update();
myGamePiece.newPos();
myGamePiece.update();

```

Juego 2 (Alejandro Paul Flores Ayala)

El presente juego consta del clásico juego snake. Como cambios notables a primera vista al código original, es que se cambiaron tanto las texturas como los colores, ahora la serpiente tiene un color verde, el tablero tiene color y se le dio a la manzana una forma de cuadrado rojo



Las principales funciones utilizadas en el juego son las siguientes:

En la función que se puede ver en la imagen de abajo, su funcionalidad consiste en darle la forma a la serpiente, la función contiene puro JavaScript

```
function tick() {
  const head = state.snake[0];
  const dx = state.direction.x;
  const dy = state.direction.y;
  const highestIndex = state.snake.length - 1;
  let tail = {};
  let interval = TICK;

  Object.assign(tail,
    state.snake[state.snake.length - 1]);

  let didScore = (
    head.x === state.prey.x
    && head.y === state.prey.y
  );

  if (state.runState === STATE_RUNNING) {
    for (let idx = highestIndex; idx > -1; idx--) {
      const sq = state.snake[idx];

      if (idx === 0) {
        sq.x += dx;
        sq.y += dy;
      } else {
        sq.x = state.snake[idx - 1].x;
        sq.y = state.snake[idx - 1].y;
      }
    }
  } else if (state.runState === STATE_LOSING) {
    interval = 10;

    if (state.snake.length > 0) {
      state.snake.splice(0, 1);
    }
  }
}
```

La siguiente función es la más importante y en ella se puede observar el uso de DOM al solicitar el canvas que básicamente es el contenedor con “getElementById(‘canvas’)”

```
window.onload = function() {
  state.canvas = document.getElementById('canvas');
  state.context = state.canvas.getContext('2d');

  window.onkeydown = function(e) {
    const direction = DIRECTIONS_MAP[e.key];

    if (direction) {
      const [x, y] = direction;
      if (-x !== state.direction.x
        && -y !== state.direction.y)
      {
        state.direction.x = x;
        state.direction.y = y;
      }
    }
  }

  tick();
};
```

Gracias a las funciones que se pueden visualizar abajo es posible que los colores “Siguen a la serpiente” y no se deforme la imagen.

```
function drawPixel(color, x, y) {
  state.context.fillStyle = color;
  state.context.fillRect(
    x * SQUARE_SIZE,
    y * SQUARE_SIZE,
    SQUARE_SIZE,
    SQUARE_SIZE
  );
}

function draw() {
  state.context.clearRect(0, 0, 500, 500);

  for (var idx = 0; idx < state.snake.length; idx++) {
    const {x, y} = state.snake[idx];
    drawPixel('#22dd22', x, y);
  }

  const {x, y} = state.prey;
  drawPixel('#ef233c', x, y);
}
```

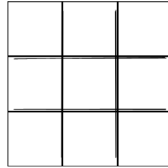
Juego 3 (Chris Alan Apaza Aguilar)



Juego 3

El juego de tres en raya, también conocido como gato o tres en línea, es un juego de mesa para dos jugadores, que consiste en marcar tres casillas en línea, ya sea horizontal, vertical o diagonal, con una ficha de cada jugador. El jugador que consiga primero tres en línea gana la partida.

Reiniciar



Universidad Católica Boliviana "San Pablo"
Tecnologías Web

Primero se tiene la función que verifica un tablero ganado, esto implica que hayan 3 fichas iguales de manera contigua.

```
function win(i, j, k) {  
  if (movesArray[i] == movesArray[j] &&  
      movesArray[j] == movesArray[k] &&  
      movesArray[i] != null)  
    return true  
  return false  
}
```

A continuación se muestran los posibles tableros y se retorna el tablero ganador para terminar de dibujar la línea que une las 3 fichas.

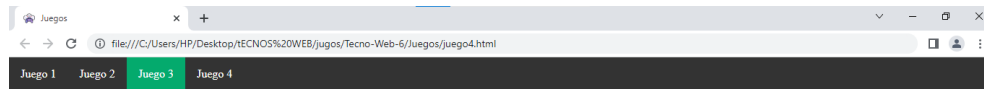
```
function numberWin() {  
  if (win(0, 1, 2)) {  
    return 1  
  }  
  if (win(3, 4, 5)) {  
    return 2  
  }  
  if (win(6, 7, 8)) {  
    return 3  
  }  
  if (win(0, 3, 6)) {  
    return 4  
  }  
}  
  
if (win(1, 4, 7)) {  
  return 5  
}  
if (win(2, 5, 8)) {  
  return 6  
}  
if (win(0, 4, 8)) {  
  return 7  
}  
if (win(2, 4, 6)) {  
  return 8  
}  
return null  
}
```

Se inicializa el turno en el tablero, la parte del DOM es en base a la línea de código `document.querySelectorALL`. con la que hacemos conexión al html donde inicialmente estará dibujado el tablero. Se agrega un evento para cada uno de los nueve botones ante la acción click del ratón: en caso de que turn sea falso se coloca la ficha X y caso contrario se coloca la ficha O.

Ante cada movimiento se evalúa con la función `numberWin()` para ver si el tablero es ganador, en caso de que se gane se dibujan las líneas de acuerdo a lo devuelto por la función anterior.

```
// Control de turnos de X y O  
let turn = true  
// Selecciona las cajas del tablero  
const boxList = document.querySelectorAll('.table .box')  
const movesArray = new Array(9).fill(null)  
  
boxList.forEach(function (box, index) {  
  box.addEventListener('click', function () {  
    if (turn) {  
      box.classList.add('mark-x')  
    } else {  
      box.classList.add('mark-o')  
    }  
    movesArray[index] = turn  
    const mayWinner = numberWin()  
    if (mayWinner) {  
      const lineWinner = document.createElement('div')  
      lineWinner.classList.add('line')  
      lineWinner.classList.add('line-winner-' + mayWinner)  
      document.querySelector('.table').append(lineWinner)  
    }  
    turn = !turn  
  }, { once: true })  
})
```


Juego 4 (Diego Cesar Alvarado Callisaya)



Universidad Católica Boliviana "San Pablo"
Tecnologías Web

Esta Función de juego es sobre el movimiento de un objeto utilizando las teclas de dirección del teclado

```
juego4.html M X JS juego4.js # cuadrado.css
juego4.html >
31 <!-- !Icons -->
32 <link
33   rel="stylesheet"
34   href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"
35 />
36
37 </head>
38
39 <!--* BODY -->
40 <body class="bodyWhite">
41   <!-- !Navbar -->
42
43   <ul class="topnav">
44     <li><a href="main.html">Juego 1</a></li>
45     <li><a href="juego2.html">Juego 2</a></li>
46     <li><a class="active" href="juego3.html">Juego 3</a></li>
47     <li><a href="juego4.html">Juego 4</a></li>
48   </ul>
49   <center>
50   </center>
51   <!-- Juego -->
52   <div id="cuadrado"></div>
53   <div id="misil"></div>
54   <script src="js/juego4.js"></script>
55
56   <!-- !Footer -->
57   <div style="height: 150px"></div>
58   <footer class="sticky-footer">
59     <p>
60       <a href="https://www.ucb.edu.bo/">
61         >Universidad Católica Boliviana "San Pablo"</a>
62     >
63     <br />Tecnologías Web
64   </p>
65   </footer>
66 </body>
67 </html>
```

Primeramente declaramos los objetos en el html en este caso serían los div cuadrado y misil, a continuación conectamos el html al javascript del código

```
juego4.html M  JS juego4.js  # cuadrado.css

js > JS juego4.js > ...
1  let x = 0;
2  let y = 0;
3  let cuadrado = document.getElementById('cuadrado');
4  let misil = document.getElementById('misil');
5  function movimiento(event){
6      if(event.keyCode == '39'){//derecha
7          x= x + 50;
8          cuadrado.style.left = x + 'px';
9      }
10
11     if(event.keyCode == '37'){//Izquierda
12         x= x - 50;
13         cuadrado.style.left = x + 'px';
14     }
15
16     if(event.keyCode == '38'){//arriba
17         y = y + 50;
18         cuadrado.style.top = (-y) + 'px';
19     }
20
21     if(event.keyCode == '40'){//abajo
22         y = y - 50;
23         cuadrado.style.top = (-y) + 'px';
24     }
25
26     if(event.keyCode == '32'){//espacio
27         misil.style.visibility='visible';
28         misil.style.left = (x+100) + 'px';
29         misil.style.top = (-y) + 'px';
30         misil.classList.add('moviendo');
31     }
32 }
33
34 window.onkeyup = movimiento;
```

En el mismo usamos la metodología DOM para obtener los id de los diferentes elementos, siendo estos cuadrado y misil.

si la ventana detecta que se presiona alguna tecla compara si esta es una de dirección el objeto se moverá en la direccion de estas usando el style.top y el style.left respectivamente, si se detecta la tecla espacio el objeto misil se hará visible y se dejará en el lugar.