

SIMON GAME



Héctor Company Ortega

1. ÍNDICE

A. Introducción

B. Reglas generales del juego

C. Diagrama de flujo o ordinograma

D. Descripción del programa que se ha desarrollado

E. Conclusión y propuestas de mejora

A. INTRODUCCIÓN

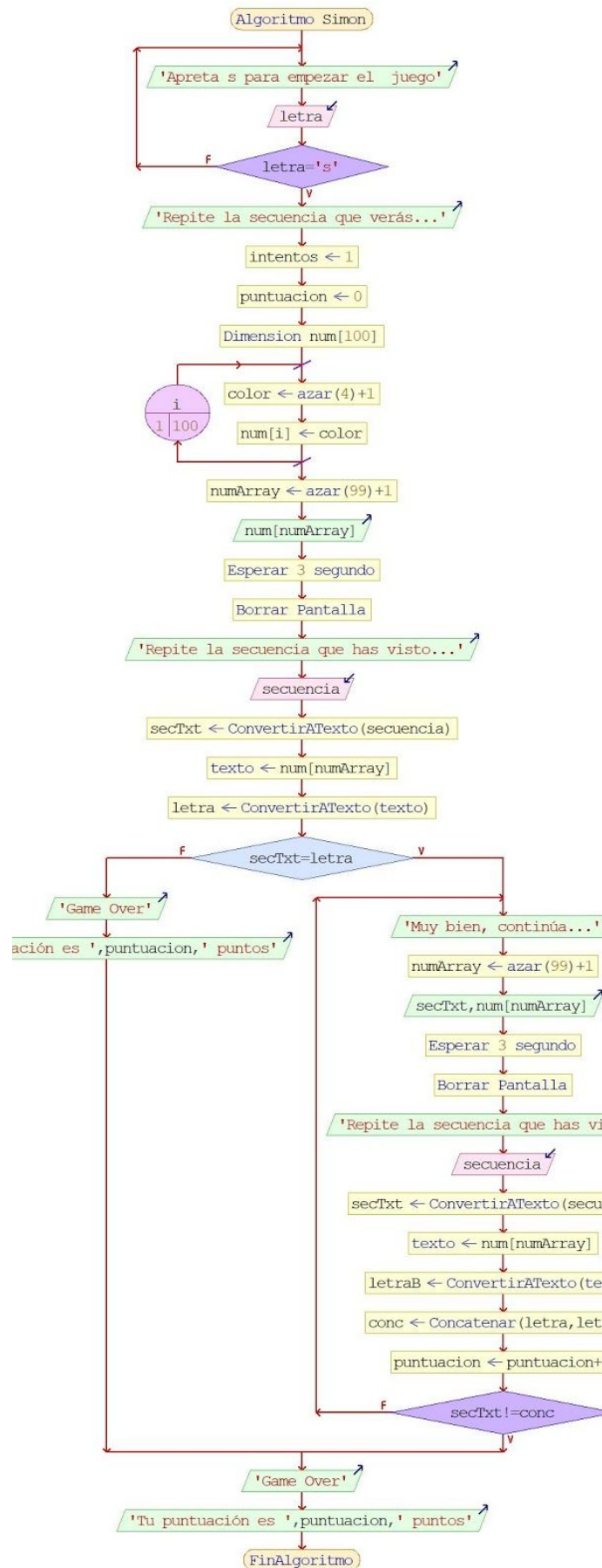
Una empresa dedicada al entretenimiento nos ha encargado el diseño del algoritmo y un prototipo del famoso juego de mesa, "SIMON".

El juego se presenta como un disco dividido en 4 zonas, cada una de ellas pintada de un color distinto. El juego consiste en que, aleatoriamente, se van iluminando cada una de las zonas y a la vez que se iluminan también se emite un sonido. Tras escuchar la secuencia el usuario debe repetirla de forma exacta, si lo consigue el juego le enviará una nueva secuencia más rápida y larga que deberá repetir el usuario y así sucesivamente hasta que falle. Si falla deberá volver a empezar.

B. REGLAS GENERALES DEL JUEGO

1. Iniciamos el juego con un control de inicio
2. La máquina realiza un sonido y se ilumina el color del primer elemento de la secuencia
3. El jugador tiene un tiempo determinado para repetir la secuencia
4. El proceso se repite hasta que el jugador falla
5. La puntuación corresponde con el número de veces que el jugador ha acertado la secuencia
6. El juego debe disponer de un control de reinicio que establezca el contador y el resultado al estado inicial.

C. DIAGRAMA DE FLUJO O ORDINOGRAMA



F. DESCRIPCIÓN DEL PROGRAMA QUE SE HA DESARROLLADO

Al comenzar a desarrollar el algoritmo en Javascript me di cuenta que generar previamente un array con 100 objetos antes de comenzar el juego no era una buena solución así que opté por adaptar el algoritmo inicial. También había que tener en cuenta que ese mismo algoritmo inicial estaba basado en estética (jugaría con números y no colores) que no hacía factible seguirlo al pie de la letra. Así pues fui adaptándolo a medida de las necesidades de la siguiente manera :

Primero creamos las variables globales necesarias para el juego :

```
var numeros = [];  
var puntuacion = 0;  
var pos = 0;  
var ciclo = "maquina";
```

- 1) Un array (**numeros**) donde introduciremos la secuencia que el usuario deberá reproducir
- 2) La puntuación (**puntuacion**) del usuario
- 3) El valor (**pos**) que nos marcará la posición dentro del array (numeros)
- 4) El modo de juego (**ciclo**) "usuario" o "**maquina**".

El siguiente paso es crear una función que nos sirva tanto para iniciar el juego como para resetearlo :

```
document.getElementById("start").onclick = function () {  
  numeros.splice(0, numeros.length);  
  generarObjeto();  
  pos = 0;  
  let puntuacion = 0;  
  document.getElementById('contador').innerHTML = "0" + puntuacion;  
  document.getElementById('start').innerHTML = "RESET";  
  ciclo = "maquina";  
  play();  
};
```

Esta función está a la espera de que el usuario presione el botón **start** y en ese momento desencadena una serie de acciones.

- 1) Borra el array (**numeros**)
- 2) Genera un nuevo número aleatorio entre 1 y 4 que sube al array(**numeros**) a través de la función **generarObjeto** que hemos creado previamente y que describiremos después :

```
function generarObjeto() {  
  let numero = Math.floor(Math.random() * 4) + 1;  
  numeros.push(numero);  
}
```

- 3) Se coloca en la primera posición del array (**numeros**) dándole a (**pos**) el valor de 0.
- 4) Le asigna el valor 0 a la puntuación inicial.
- 5) Seleccionamos el elemento contador y también lo reseteamos a 0 puntos
- 6) Seleccionamos el botón "**start**" y cambiamos su texto por "**RESET**".
- 7) Cambiamos el ciclo al valor "maquina" para que pueda iniciarse un nuevo juego.
- 8) Y finalmente reproduce la función **play** que después también describiremos.

Posteriormente realizo cuatro funciones iguales para cada uno de los botones del juego "**Simon**" que solo se diferencian en en la asignación necesaria para las correctas interacciones del juego, es decir, que cuando se pulse el botón rojo se desencadene la función **rojoF**, cuando se pulse el botón azul la **verdeF**... y así con el resto de botones.

```
document.getElementById("rojo").onclick = function rojoF() {
  document.getElementById("rojo").style.opacity = "1";
  document.getElementById("audioBotonRojo").play();
  setTimeout(function () {
    document.getElementById("rojo").style.opacity = "0.5";
  }, 500);

  if (ciclo == "usuario") {

    if (numeros[pos] == 1) {
      pos++;

      if (pos == numeros.length) {
        ciclo = "maquina";
        puntuacion = puntuacion + 1;
        document.getElementById('contador').innerHTML = "0" + puntuacion;
        generarObjeto();
        pos = 0;
        setTimeout(play, 1500);
      } else {}
    } else {
      document.getElementById('contador').innerHTML = "0" + puntuacion;
      document.getElementById('start').innerHTML = "START";

      alert("GAME OVER");
    }
  } else {}
};
```

Esta función, estéticamente, hace que cuando se pulse un botón en particular este se ilumine cambiando su opacidad, suene el audio correspondiente y que pasado medio segundo regrese a su iluminación original creando así el efecto de iluminación.

Por otro lado, también está indicando al programa que si se desencadena esta función y está en modo “usuario” y si la posición dentro del array (**numeros**) es igual a uno (valor que corresponde al botón rojo) corra una posición más dentro del array para continuar haciendo sonar la secuencia del juego Simon. En caso de que no se cumpla esta condición el programa emite un mensaje de finalización de juego y devuelve la puntuación final obtenida.

Esta función también está indicando que si las posiciones marcadas por el usuario son iguales a las que tiene el array (**numeros**) cambie el ciclo a “**maquina**”, sume uno a la puntuación del usuario, pinte la puntuación en el HTML, genere un nuevo valor aleatorio y vuelva al inicio del array (**numeros**) para hacer sonar la siguiente secuencia al cabo de 1 segundo y medio.

A continuación desarrollo la función **generarObjeto** que he comentado anteriormente :

```
function generarObjeto() {  
  let numero = Math.floor(Math.random() * 4) + 1;  
  numeros.push(numero);  
}
```

Esta función es muy simple y nos genera un número aleatorio entre 1 y 4 que posteriormente sube a nuestro array(**numeros**).

Finalmente llegamos a la función **play** :

```
function play() {  
    if (pos < numeros.length) {  
        switch (numeros[pos]) {  
            case 1:  
                // Se utiliza el metodo click para llamar al evento onclick  
                document.getElementById("rojo").click();  
                break;  
            case 2:  
                // Se utiliza el metodo click para llamar al evento onclick  
                document.getElementById("verde").click();  
                break;  
            case 3:  
                // Se utiliza el metodo click para llamar al evento onclick  
                document.getElementById("azul").click();  
                break;  
            case 4:  
                // Se utiliza el metodo click para llamar al evento onclick  
                document.getElementById("amarillo").click();  
                break;  
        }  
        pos++;  
        setTimeout(play, 1000);  
    } else if (pos == numeros.length) {  
        pos = 0;  
        ciclo = "usuario";  
    }  
}
```

Esta función está indicando al programa que si el valor de nuestra variable **pos** es inferior a la longitud total del array (**numeros**) evalúe los valores existentes dentro de nuestra array y ejecute la función correspondiente a dicho valor. Cuando termina va a la siguiente posición de nuestro array (**pos++**) y ejecuta de nuevo la función **play** al cabo de un segundo. De esta manera podremos escuchar y ver la secuencia de Simon en la pantalla.

En caso de que el valor de **pos** no fuera inferior a la longitud total de nuestro array(**numeros**) sería igual por lo que se ejecutaría la siguiente sentencia que le indica al programa que debe volver a la posición inicial del array(**numeros**) y cambiar al ciclo "usuario" pues de esta manera podremos evaluar si el usuario está introduciendo correctamente la secuencia gracias a las funciones antes descritas y que corresponden a cada uno de los botones.

F. CONCLUSIONES Y PROPUESTAS DE MEJORA

No me he encontrado con dificultades relevantes más allá de la de saber enfocar inicialmente el proyecto pues comencé basándome en el algoritmo creado en Pselnt pero al no estar este pensado para

Las principales propuestas de mejora serían las siguientes:

Añadir tiempo para dar la secuencia correcta.

Pintar “**GAME OVER**” dentro del HTML cuando finalice la partida y no por **alert**.

Añadir posibilidad de música durante el juego

Añadir niveles de dificultad