

## GUI N DE ACTIVIDAD

# Desarrollo Web en Entorno Cliente

### ACTIVIDAD RECUPERACI N:

### OBJETIVOS

- Recuperar la nota de actitud de la primera evaluaci n.
- Profundizar en los conocimientos aprendidos de JavaScript.

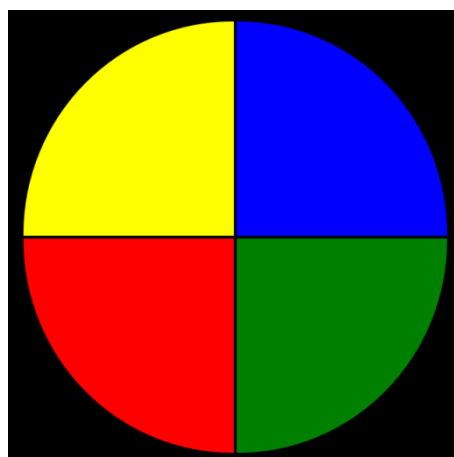
### TEMPORALIZACI N

*Aproximadamente 4-5 horas*

### PROCESO DE DESARROLLO

Los archivos index.html y styles.css proporcionados contienen los estilos necesarios para implementar una versi n evolucionada del cl sico juego Simon ([link](#)). En la presente actividad se precisa implementar un script simon.js que lleve a cabo las acciones necesarias para que pueda desarrollarse el citado juego. La versi n a desarrollar consta de cuatro niveles distintos de funcionamiento equivalente, tal y c mo se describen a continuaci n:

**Nivel 1:** compuesto por una interfaz gr fica que simula la apariencia del Sim n original, como se aprecia en la imagen siguiente:



En este nivel se llevarán a cabo las siguientes acciones:

- a. Al hacer click sobre el título de primer nivel con la inscripción “Let’s Play!”, el contenido de este desaparecerá y se dará inicio al juego. Para ello se calculará una secuencia aleatoria de cuatro números, representando cada uno de ellos una de las cuadrículas del tablero de juego -es decir, cada número representa cada uno de los cuatro cuartos de círculo que forman la interfaz gráfica del juego Simón-. A su vez, cada una de las cuatro cuadrículas tendrá asociada un sonido de sintetizador, obtenido mediante la librería Tone.js -cuyo CDN está ya incorporado en el archivo index.html- y cuya implementación se realiza mediante los cuatro pares de instrucciones siguientes:

```
const synth = new Tone.Synth().toDestination()  
synth.triggerAttackRelease('C4', '8n')
```

```
const synth = new Tone.Synth().toDestination()  
synth.triggerAttackRelease('C5', '8n')
```

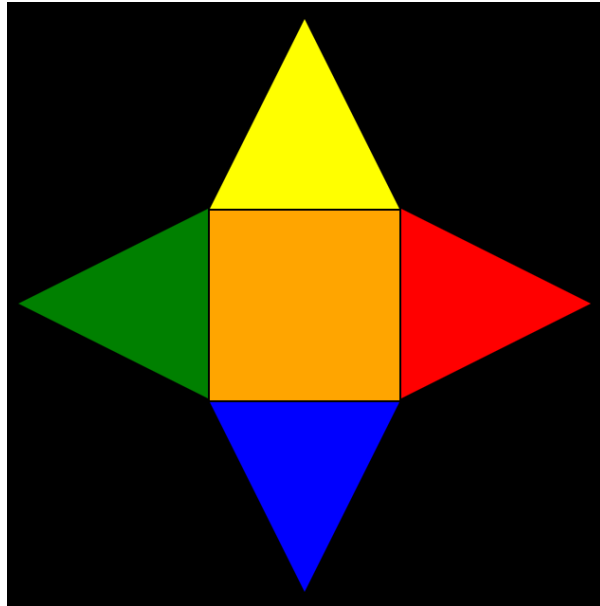
```
const synth = new Tone.Synth().toDestination()  
synth.triggerAttackRelease('C6', '8n')
```

```
const synth = new Tone.Synth().toDestination()  
synth.triggerAttackRelease('C7', '8n')
```

- b. Una vez calculada la secuencia, empezará el juego y con ello se mostrará primeramente y durante un intervalo de tiempo de 500 ms -a implementar mediante la función predefinida de JavaScript setInterval- la siguiente acción: el cuarto de círculo correspondiente al primer número de la secuencia aleatoria calculada aparecerá de color blanco y al unísono sonará el sonido correspondiente a dicho cuarto del tablero.
- c. Pasado el intervalo temporal, el tablero volverá a mostrar sus colores originales, esta vez con un evento ‘click’ implementado para cada una de las cuatro partes del tablero de juego.
- d. Cuando el usuario haga ‘click’ en una de las cuadrículas, se comprobará si se ha pulsado en la cuadrícula que se había cambiado el color a blanco.
- e. En caso de que la cuadrícula señalada por el usuario y la cuadrícula que había cambiado de color no coincida, se habrá terminado el juego, eliminándose el tablero y mostrándose únicamente por pantalla un título de primer nivel con la inscripción “Game Over”. En caso de que la cuadrícula señalada por el usuario y la cuadrícula que había cambiado de color sí coincidan, se volverá al punto b, pero esta vez se cambiará primero de color la cuadrícula correspondiente al primer número de la secuencia y después se hará lo propio con el segundo número de la misma. Ambos cambios se realizarán durante un intervalo de 500 ms y se llevarán a cabo mientras se emiten sus sonidos asociados.
- f. Se realizarán otra vez los pasos c y d y luego la comprobación del paso e. En caso de que la comprobación sea afirmativa se realizará otra vez los pasos b, c, d y e, pero esta vez para los tres primeros números de la secuencia.

- g. Se repetirá el paso f con los cuatro primeros números de la secuencia. En caso de que el usuario acierte la secuencia, se cambiará al nivel siguiente.

**Nivel 2:** El segundo nivel comprenderá el mismo algoritmo descrito para el primer nivel, pero esta vez la interfaz tendrá cinco figuras geométricas y cada una de ellas tendrá un sonido asociado, tal y cómo se indica a continuación:



```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C3', '8n')
```

```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C4', '8n')
```

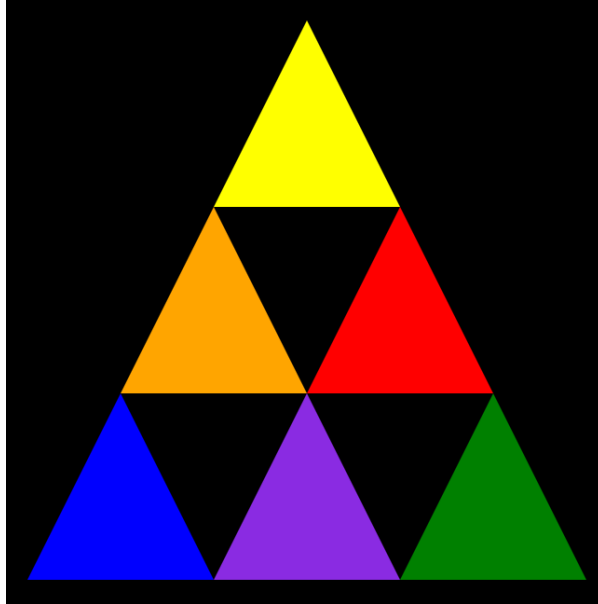
```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C5', '8n')
```

```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C6', '8n')
```

```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C7', '8n')
```

Para este nivel también se tendrá que calcular una secuencia aleatoria de cuatro números y realizar las mismas comprobaciones que en el nivel anterior. El nivel finalizará si el usuario llega a acertar una secuencia de cuatro cambios consecutivos de cuadrículas con sus sonidos asociados.

**Nivel 3:** El tercer nivel comprenderá el mismo algoritmo descrito para el primer y segundo nivel, pero esta vez la interfaz tendrá seis triángulos y cada uno de ellos tendrá un sonido asociado, tal y cómo se indica a continuación:



```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C2', '8n')
```

```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C3', '8n')
```

```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C4', '8n')
```

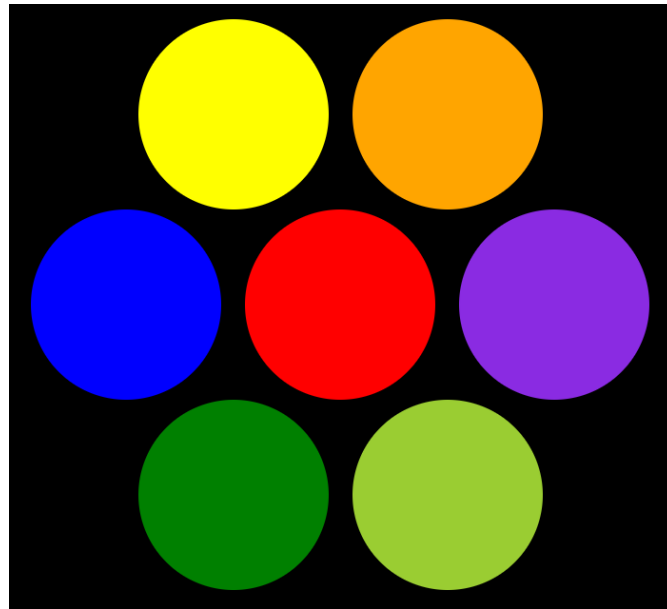
```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C5', '8n')
```

```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C6', '8n')
```

```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C7', '8n')
```

Para este nivel también se tendrá que calcular una secuencia aleatoria de cuatro números y realizar las mismas comprobaciones que en los dos niveles anteriores. El nivel finalizará si el usuario llega a acertar una secuencia de cuatro cambios consecutivos de cuadrículas con sus sonidos asociados.

**Nivel 4:** El cuarto y último nivel comprenderá el mismo algoritmo descrito para los tres primeros niveles, pero esta vez la interfaz tendrá siete círculos y cada uno de ellos tendrá un sonido asociado, tal y cómo se indica a continuación:



```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C2', '8n')
```

```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C3', '8n')
```

```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C4', '8n')
```

```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C5', '8n')
```

```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C6', '8n')
```

```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C7', '8n')
```

```
const synth = new Tone.Synth().toDestination()
synth.triggerAttackRelease('C8', '8n')
```

Para este último nivel se calculará una secuencia aleatoria de quince números y se realizarán las mismas comprobaciones que en los cuatro niveles anteriores. El nivel finalizará si el usuario llega a acertar una secuencia de quince cambios consecutivos de cuadrículas con sus sonidos asociados. Si eso sucede, el usuario se habrá pasado el juego.

## **EVALUACIÓN**

---

Esta actividad corresponde a la totalidad nota de la parte de actitud de la 1ª evaluación.

## **OBSERVACIONES**

---

Se entregará un único archivo comprimido con los tres archivos simon.html, simon.js y simon.css. La entrega se realizará por correo electrónico a la dirección [fsegura@florida-uni.es](mailto:fsegura@florida-uni.es).