

Documento de Decisiones de Diseño y Desafíos Encontrados

Decisiones de Diseño:

1. **Validación del nombre:** Se implementó una validación simple para el campo de nombre del jugador, utilizando una expresión regular (regex) que asegura que el nombre tenga al menos 4 caracteres y no contenga números. Esta validación se realiza antes de permitir al jugador iniciar el juego. La decisión de realizar esta validación en el cliente evita la necesidad de un servidor para esta funcionalidad básica y mejora la experiencia del usuario al proporcionar retroalimentación instantánea.
2. **Interactividad del tablero:** La creación de una tabla de 10x10 para representar el tablero del juego fue una decisión clave. Al principio de la partida, el héroe comienza en la celda inicial (1,1) y el objetivo es alcanzar el tesoro en la celda final (10,10). Esta tabla está generada dinámicamente mediante código JavaScript y permite a los jugadores interactuar con las celdas para mover al héroe según el resultado de un dado, proporcionando una experiencia visual y divertida.
3. **Uso de eventos para interacciones:** El uso de eventos de clic para las acciones principales (como iniciar el juego, lanzar el dado y mover el héroe) fue una elección natural dada la necesidad de interacción continua por parte del usuario. Se emplearon escuchadores de eventos (event listeners) para gestionar las acciones del jugador sin necesidad de recargar la página, permitiendo un flujo de juego fluido y sin interrupciones.
4. **Almacenamiento local para los récords:** El sistema guarda el récord de tiradas del jugador en el almacenamiento local del navegador (localStorage). Esto permite que el récord se mantenga entre sesiones, brindando un sentido de progreso y desafío adicional para el jugador. La decisión de utilizar localStorage fue tomada debido a su simplicidad y a la necesidad de persistencia de datos sin la complejidad de un servidor.

Desafíos Encontrados:

1. **Gestión de la lógica del movimiento del héroe:** Un desafío importante fue implementar la lógica para mover al héroe dentro del tablero. Debido a que las celdas de la tabla están identificadas de manera dinámica, fue necesario realizar cálculos para determinar la posición actual del héroe y las celdas adyacentes que podían ser resaltadas. Además, se necesitó un sistema para permitir que el héroe se moviera solo a las celdas válidas basadas en el resultado del dado. Esto implicó un manejo preciso de las celdas y un control del flujo de datos entre ellas.
2. **Interfaz dinámica y reactividad:** Crear una interfaz que fuera capaz de actualizarse dinámicamente (como ocultar/mostrar elementos y cambiar el contenido de las celdas) resultó ser un desafío de diseño. Se necesitó coordinar correctamente los estados de los botones y las celdas del tablero a lo largo del juego para asegurar que el jugador pudiera ver solo lo necesario en cada momento (por ejemplo, mostrar los botones en el momento adecuado y ocultar elementos innecesarios).
3. **Validación de entrada del usuario:** La validación del nombre del jugador, aunque simple, presentó el desafío de proporcionar retroalimentación clara y útil

al usuario sin ser demasiado restrictivo. Asegurarse de que el sistema sea lo suficientemente flexible para manejar entradas diversas pero al mismo tiempo estricto en cuanto a los requisitos del nombre (sin números y con al menos 4 caracteres) implicó un manejo cuidadoso de los errores y mensajes.

4. **Propagación del dado en el tablero:** Otro desafío técnico fue propagar la clase de resalte en el tablero según el valor del dado. A medida que el jugador tira el dado, las celdas adyacentes deben ser resaltadas, lo que implica una lógica compleja para determinar qué celdas deben cambiar de estado. El manejo de los límites de la tabla y el control de la visibilidad de las celdas activas fue un aspecto que requirió pruebas y ajustes.
5. **Persistencia del récord:** Aunque el uso de `localStorage` para guardar el récord del jugador era la opción más simple, también implicó asegurarse de que la información estuviera gestionada adecuadamente, como la verificación de si el jugador ha superado el récord previamente guardado. Además, la implementación de la lógica para comparar las tiradas y guardar un nuevo récord fue un aspecto que requirió un enfoque cuidadoso para evitar la sobrescritura de datos no deseados.