

## PRUEBA TÉCNICA UNITY DEVELOPER

En este documento explicaré el proceso de desarrollo de la prueba y las decisiones que he ido tomando durante la prueba.

Primero de todo, he decidido estructurar la **UI** a partir de diversas *Vertical* y *Horizontal Layouts* para hacer la aplicación adaptable a las diversas propuestas de diseño que se puedan llegar a implementar.

- Evitando solapamiento de textos en la sección de "Aciertos" y "Fallos"
- Posibilitando la implementación de más de tres opciones a escoger.

En cuanto a la estructura de **código** he decidido implementar un controlador principal, nombrado "MainController", y dos sub controladores:

"ShowNumberController" y "ShowOptionsController". Estos se encargan de gestionar toda la funcionalidad del ejercicio.

- **MainController:** contiene la base de datos de los números que se irán mostrando. Esta base de datos es un documento Json que contiene un *array* con estructuras de números con un valor numérico y un valor textual. De esta manera se puede ampliar y modificar como se desee los números que aparezcan por pantalla. Este controlador también se encarga de iniciar el ejercicio e irse guardando la opción correcta de cada ronda, así como el total de aciertos y fallos.
- **ShowNumberController:** controlará la aparición y desaparición del número (texto) inicial.
- **ShowOptionsController:** se encargará de generar de manera aleatoria las opciones a escoger, siempre guardando una opción para el número correcto. Cada opción a escoger es controlada individualmente por un script nombrado "NumberObject". Este controla la funcionalidad relacionada con la interacción con

el elemento (habilitar/deshabilitar interacción, cambios de color, animaciones de entrada y salida). Este notificará mediante un evento a los sistemas pertinentes de qué opción ha sido escogida.

Podemos observar un patrón *Singleton* implementado en este sistema, ya que los sub controladores pueden acceder a la instancia del controlador principal para realizar funcionalidades y comprobaciones concretas.

También podemos ver un patrón *Observer* en el hecho de notificar mediante evento que se ha seleccionado una opción concreta. De esta manera favorecemos la ampliación y modularidad del proyecto si en un futuro se implementan más funcionalidades, permitiendo que gestionen independientemente la información recibida.

En cuanto a las **animaciones**, he decidido implementarlas vía código, con un script llamado "AnimHelper" que básicamente lanza una corutina que gestiona durante X tiempo una acción específica. Es una funcionalidad que usamos habitualmente en la empresa y permite crear cualquier tipo de animación, modificando el *alpha* de un *CanvasGroup*, la posición de un *RectTransform*, el tamaño, etc.

Las animaciones también controlarán el flujo del juego, habilitando y deshabilitando la interacción en los puntos específicos y así evitar la aparición de errores que corrompan la acción del juego.