

Memoria Prueba Innovamat

La prueba ha consistido en desarrollar una aplicación 2D que implemente un dictado numérico. El usuario tendrá que elegir, de entre tres posibles números, cuál de ellos se corresponde con el texto mostrado.

En esta prueba hay varios elementos importantes a tener en cuenta. En primer lugar, el texto donde aparece escrito con letras el número generado. Seguidamente, los botones que muestran los dígitos que el usuario debe pulsar para avanzar en el nivel. También hay que tener en cuenta los contadores de aciertos y fallos que se irán actualizando a medida que el usuario progresa. Por último, necesitamos un Game Controller que contenga la lógica de la aplicación y un sistema que genere un número aleatorio y lo convierta a texto.

Para el texto, se ha generado un controlador Singleton para asegurar que solo el Gameobject encargado de gestionar la lógica pueda usar dicho controlador. Este controlador será el encargado de actualizar en cada bucle el contenido del texto. Además, se encargará de actualizar los estados de la animación de aparición y desaparición del texto. Por último, una vez el texto haya desaparecido, lanzará un evento para avisar a los botones que deben mostrarse.

Cada botón contiene un controlador que se encarga de gestionar los estados de la animación dependiendo de lo que haga el usuario. Además, gestiona el reseteo de los estados del botón para crear un nuevo nivel. En este caso no se ha usado el patrón Singleton ya que cada botón tiene su propio controlador que lo gestiona individualmente, en vez de tener un controlador de botones único.

En el caso del controlador de los contadores de aciertos y fallos, al igual que en el caso del texto, se ha decidido usar el patrón Singleton debido a que solo habrá un controlador para ambos contadores. Este controlador se encarga de actualizar constantemente los valores de aciertos y fallos y de proporcionar métodos para que otros elementos de la escena sean capaces de acceder a estos datos.

En el caso del generador de números, no se ha usado el patrón Singleton debido a que no controla ningún aspecto de la escena. Este generador aleatoriza un número entero positivo, entre 0 y el valor máximo de enteros. Estos límites se han hecho serializables para que se puedan modificar de manera cómoda. Además, se encarga de generar una cadena de texto que se corresponda con el número leído a partir de un entero.

El GameController se encarga de generar niveles. Para realizar este proceso se hace uso de todos los elementos descritos anteriormente: Se generan tres números aleatorios y de uno de ellos se genera el texto usando el generador de números. Se actualiza el contenido del texto y se pide a su controlador que inicie la animación. Se configuran los botones de manera que cada botón contenga la información necesaria para saber si la respuesta que incluye es correcta o incorrecta y que, una vez sea pulsado, el propio controlador del botón actúe en consecuencia. Ambos comportamientos están implementados en este controlador.

Con esta estructura, nos aseguramos de que todo el código esté encapsulado, se puedan modificar comportamientos sin afectar al resto de componentes y sea fácilmente ampliable, por ejemplo, añadiendo varios niveles de dificultad en el que cambie el número de respuestas disponibles, ampliando o disminuyendo el rango de números generados o realizando la media de aciertos y fallos de un alumno.