

Prueba Técnica Innovamat

Planteamiento

Estados y Contexto

Para poder llevar a cabo un control detallado de la actividad y asegurar su modularidad se ha optado por dividir la misma en varios estados (“ActivityState”). A partir del enunciado se han extraído los siguientes pasos a grandes rasgos y se dividió en 4 bloques:

Generar enunciado (GenerateStatementState)	Animaciones introducción (PreselectionState)
<ul style="list-style-type: none">- Configurar enunciado- Configurar botones	<ul style="list-style-type: none">- Animación de enunciado- Animación de botones
Espera input de usuario (SelectionState)	Corrección y animaciones de salida (PostselectionState)
<ul style="list-style-type: none">- Escuchar input del usuario	<ul style="list-style-type: none">- Comprobar input recibido- Decidir próximo paso- Reiniciar flujo

Cada uno de esos bloques conforma un “ActivityState” y poseen funcionalidad es independiente. Para manejar el flujo entre estos 4 estados se ha creado un “ActivityContext”, el cual nos permite cambiar entre estados y almacenar aquellos elementos que puedan ser de interés común para el resto de estados.

Todos estos estados heredan de la clase abstracta “ActivityState” e implementan los métodos “Enter()”, “Exit()” y “SetContext(ActivityContext)”, este último permitiendo acceso al contexto que controla los cambios de estado.

Aparte, se ha creado “ActivityStatesDictionary” donde se almacenan cada uno de los estados y se enlazan a un tipo del enumerador “StateType” de forma que tengamos fácil acceso a cualquier estado sin necesidad de tener una referencia directa a la instancia del mismo. De este modo, si en un futuro se quisiera, se podrían generar nuevos estados de manera muy sencilla e incluirlos dentro del flujo de la aplicación a partir del contexto.

Enunciados como ScriptableObject

Teniendo en cuenta que es posible que los enunciados se quisieran, hasta cierto punto, configurar a mano por parte del equipo didáctico, se ha optado por usar ScriptableObject, ya que nos permite modificar los datos del enunciado sin necesidad de acceder a ninguna escena ni componente. La clase creada para tal tarea es “Statement”.

Para facilitar el control de estos enunciados se han agrupado todos en un “StatementsDictionary”, una clase que hereda de “ScriptableObject” y que posee una lista de “Statements” y métodos para obtener enunciados aleatorios.

Posibles mejoras

Gestión de Tasks

No me he parado a gestionar las tareas inacabadas, por lo tanto, al ejecutarse de forma asíncrona, a veces al finalizar la ejecución saltan “NullReference”.

Optimización de instancias de botones

Ahora mismo, los botones de respuesta se instancian cada vez que se genera un nuevo enunciado y se destruyen al finalizarlo. En lugar de eso, se podrían reutilizar los botones existentes, pero como cada “Statement” puede poseer un número variable de posibles respuestas y el poco tiempo que me quedaba, no se ha aplicado la optimización.