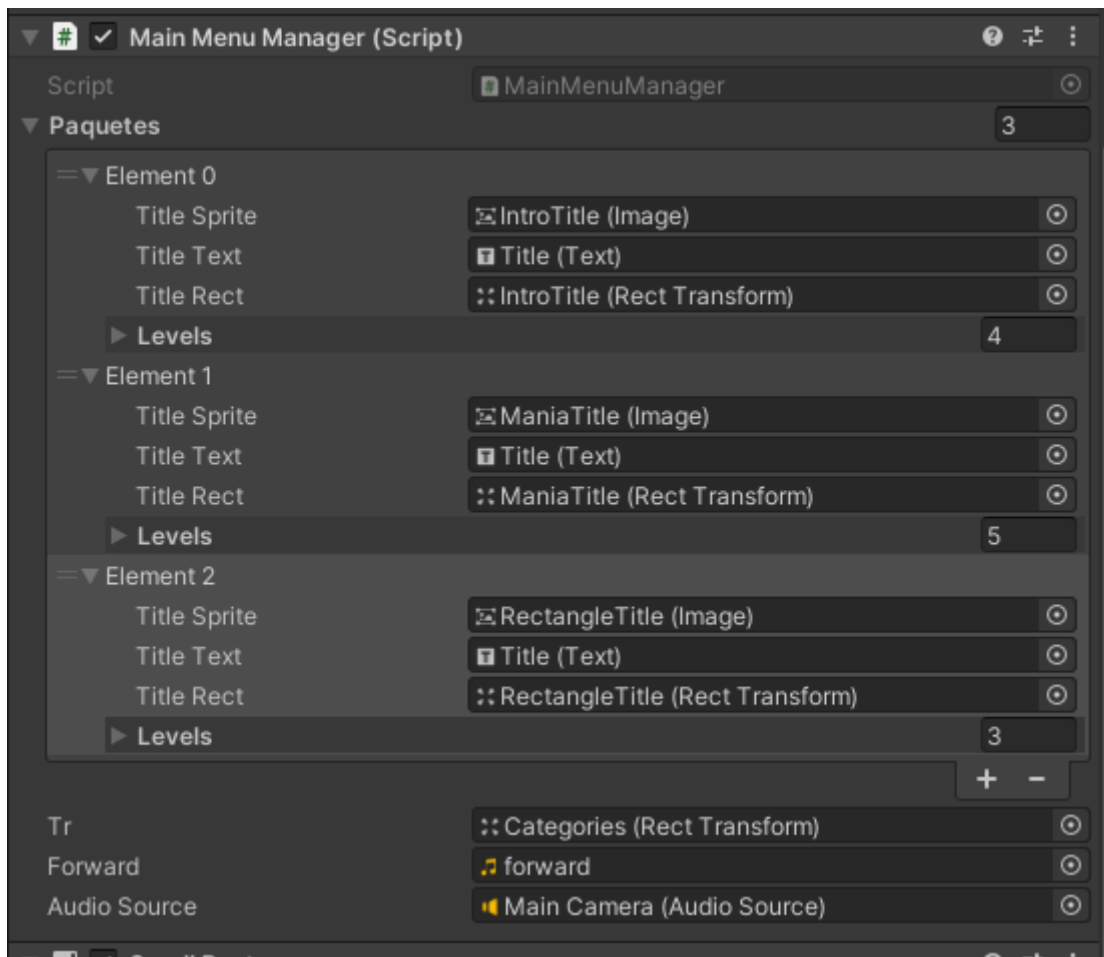


Resumen Practica 2 - FlowFree

MainMenuScene

Al arrancar la aplicación lo primero que se hace es cargar los datos mediante **DataManager**, script que se encarga de leer los datos guardados. Si no existen esos datos, entonces se crean unos por defecto. Además, para evitar posibles *hacks* se está utilizando un hash dentro del *.json* generado que comprueba que los datos sean los correctos, es decir, si se mantienen en el mismo estado que antes. En caso de que no sea así, los datos guardados se borrarán y se crearán unos nuevos por defecto.

Tras haber inicializado los datos (número de pistas, si el usuario es premium o no, los temas desbloqueados y cada uno de los niveles del juego y su estado correspondiente), **GameManager** se encargará de almacenarlos para que puedan ser consultados y modificados a lo largo del juego. En primer lugar, nos encontramos con el **MainMenuScene** donde **MainMenuManager** será el script que se encarga de inicializar todo el menú, de manera que dicho script posee atributos públicos para que cada uno de los *Objetos* de la escena se inicialicen con los datos correspondientes, los cuales se obtienen del **GameManager**

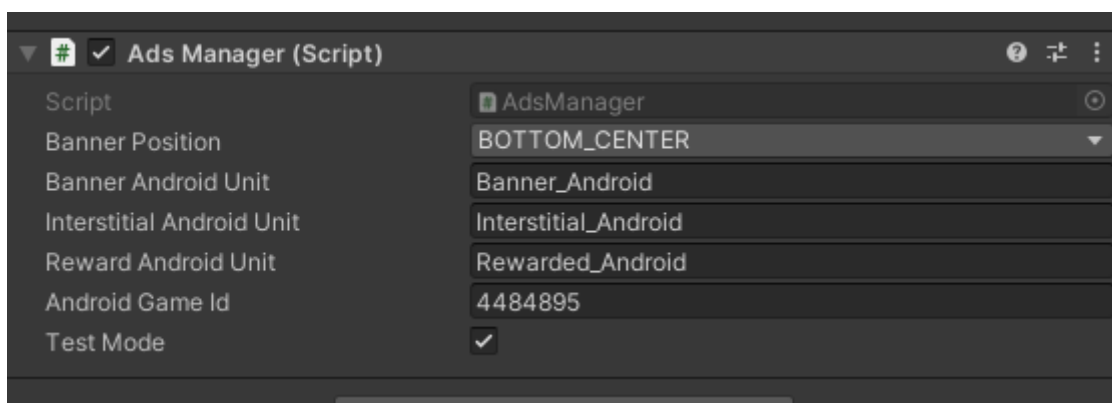


Desde aquí, se podrá cerrar la aplicación con la cruz que aparece arriba a la derecha de la pantalla, acceder a la tienda con el botón **Quitar Anuncios** y acceder a cada uno de los paquetes del juego.



Tanto para este menú como para los que vienen a continuación se estarán usando los componentes **Vertical Layout Group** u **Horizontal Layout Group** en diversos GO's para realizar un escalado automático de cada uno de los elementos del Canvas. Además, se hace uso del componente **Scroll Rect** que nos permite desplazar la pantalla arrastrando el dedo por la pantalla.

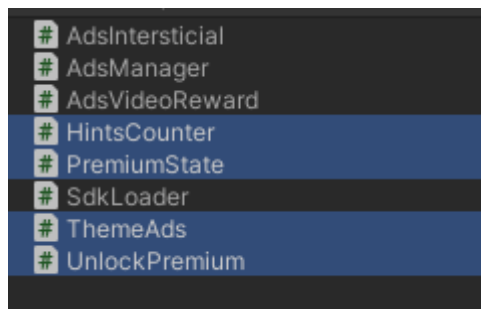
Dentro de esta escena nos encontramos también con un banner de anuncios, controlado mediante **AdsManager**



AdsScene

Si le damos al botón de los anuncios podremos acceder a la tienda, de manera que, dentro de éste nos encontramos con los siguientes elementos: primero está el anuncio sobre obtener la licencia premium del juego (simulando que cuesta 4.99 €, aunque en realidad está funcionando siempre ese botón), de manera que al pulsar el botón, dicho frame se elimina, pues se supone que ya has pagado por ello y siempre serás premium; por otro lado, se pueden comprar más pistas, simulando los botones de la misma manera que en premium y, por último, está la opción de adquirir nuevos temas para personalizar el juego.

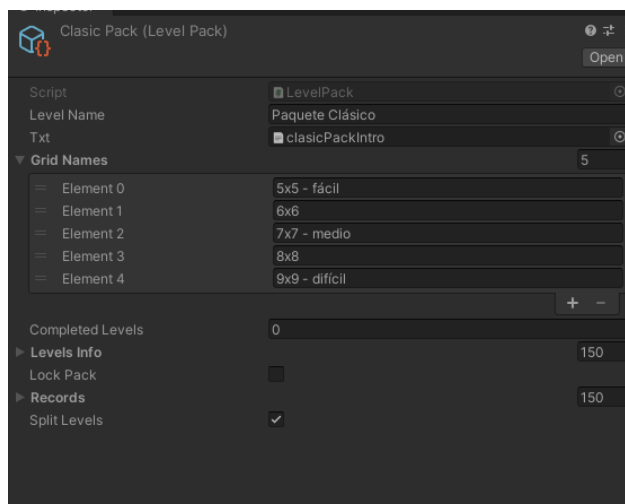
Dentro de esta escena, hacemos uso de los siguientes scripts para gestionar la compra de los distintos recursos: **HinstCounter**, **PremiumState**, **ThemeAds**, **UnlockPremium**.



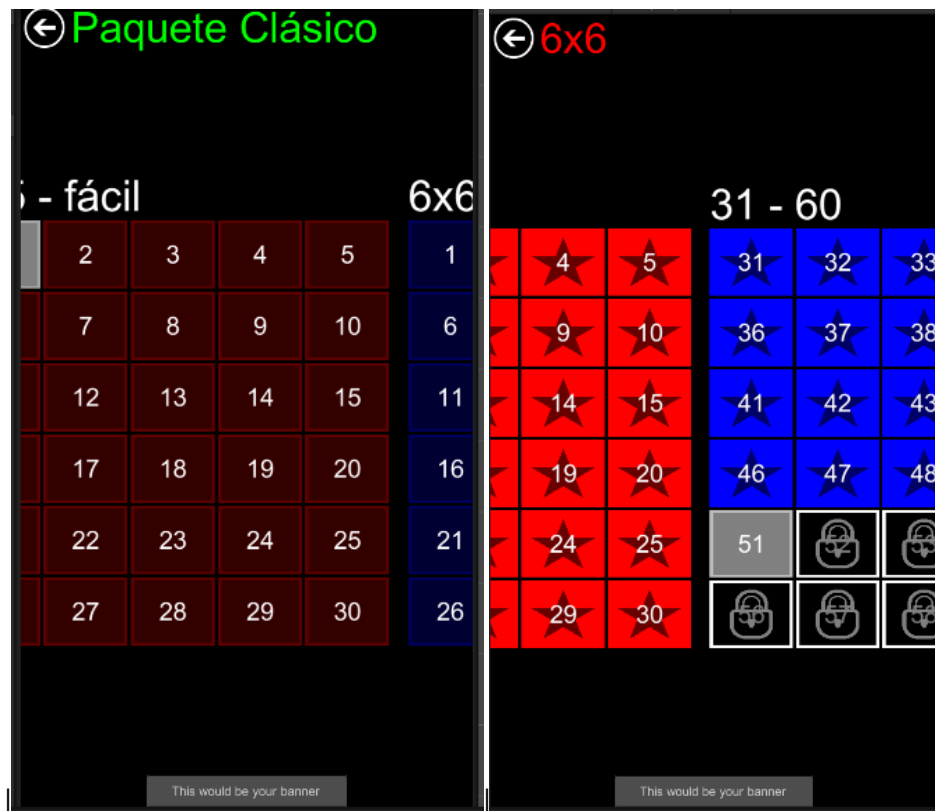
- **HintsCounter**: gestiona la cuenta de las pistas, informando correctamente al **GameManager** para guardar el estado del juego mediante **DataManager**.
- **PremiumState**: determina si hay que activar o no la oferta para ser premium, de manera que si eres premium, no aparecerá ese banner.
- **UnlockPremium**: desbloquea el premium y guarda el estado del juego al pulsar el botón del pago.
- **ThemeAds**: se encarga de gestionar la sección de los temas, dando feedback sobre si el tema está comprado o no y guardando el estado del juego al aplicar un nuevo tema o al comprar uno.

GameSelectionScene

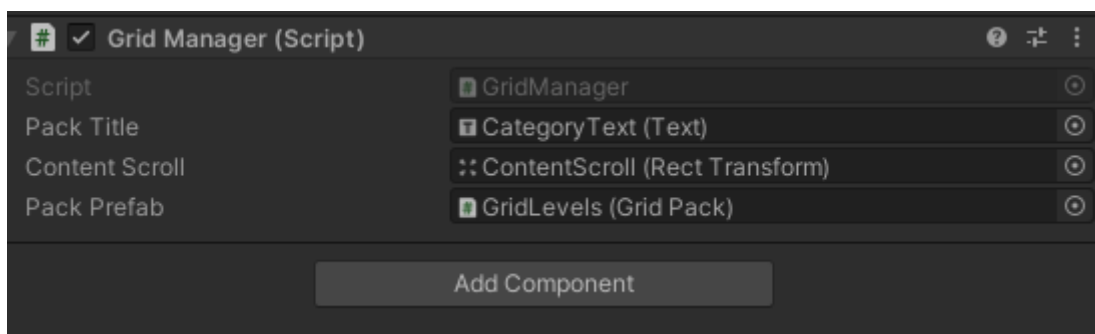
Para acceder al menú de selección de nivel hay que acceder desde el menú principal. Cada vez que se abre se leen los temas del nivel, los cuales están empaquetados mediante **ScriptableObject** conteniendo la información necesaria para cada paquete.



Dentro de estos paquetes se puede hacer lo siguiente: asignarle un nombre, asignar el archivo de texto que contenga la información del nivel, asignar nombre al título de los grid donde se selecciona el nivel, saber el número de niveles completados, información para nivel (si está completado y si es perfecto o no), si el paquete está bloqueado (como en el caso de las manías, donde solo está disponible el primer nivel hasta que sea superado y se desbloquee el siguiente), información sobre el récord de cada nivel y un *booleano* que determina la enumeración de los niveles dentro del menú de selección de nivel.



Para la creación de los diversos **Grids** se utiliza el script **GridManager**, funcionando de la misma manera que el **MainMenu** es decir, a partir de los datos leídos del **GameManager** se inicializan cada uno de los objetos de la escena, a partir de un prefab **GridLevels** el cual a su vez instancia, mediante el script **GridPack**, cada uno de los prefabs **Box** que será cada tile de cada grid, el cual poseerá la funcionalidad necesaria para cargar el nivel correctamente en **GameScene**.

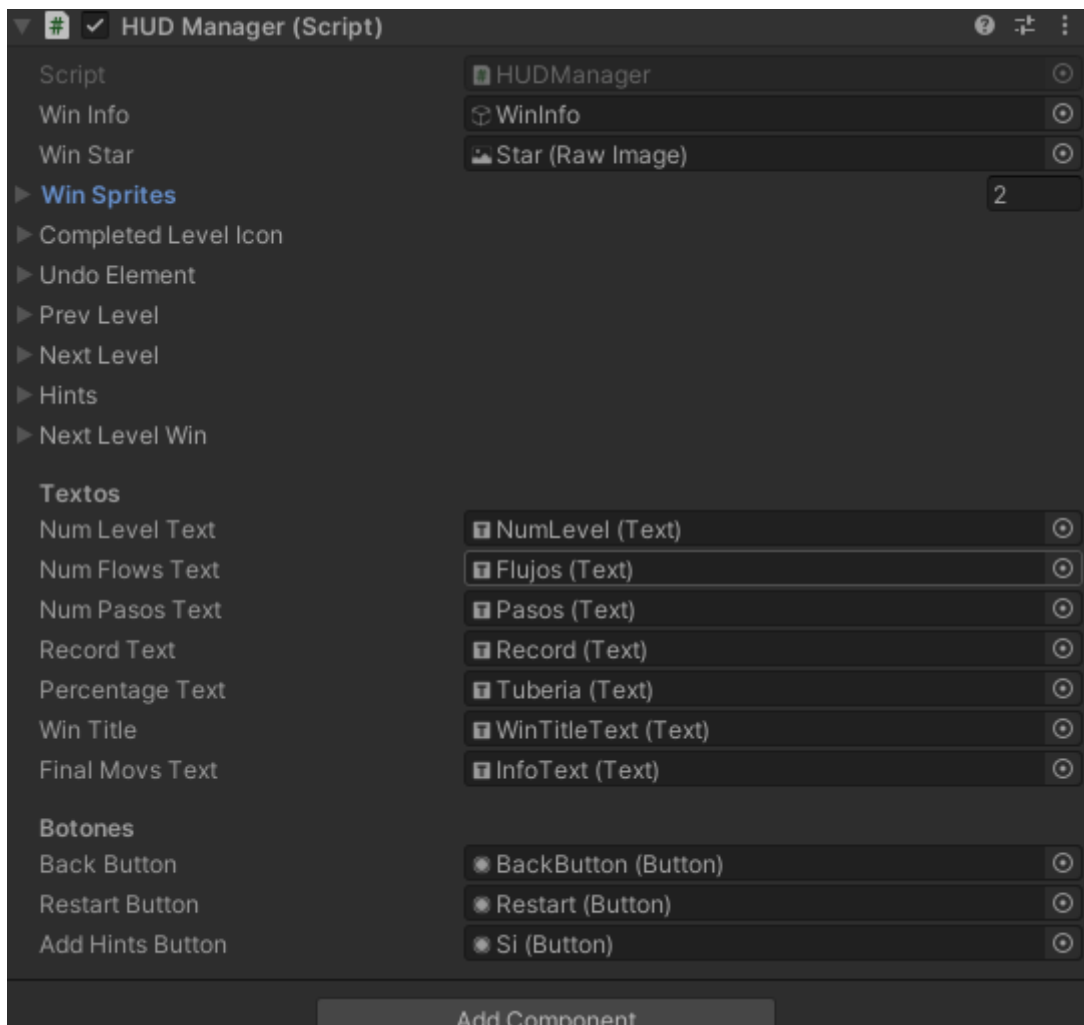


En esta escena, como en la del **MainMenu**, aparece un *banner* para mostrar anuncios en la parte inferior de la pantalla.

GameScene

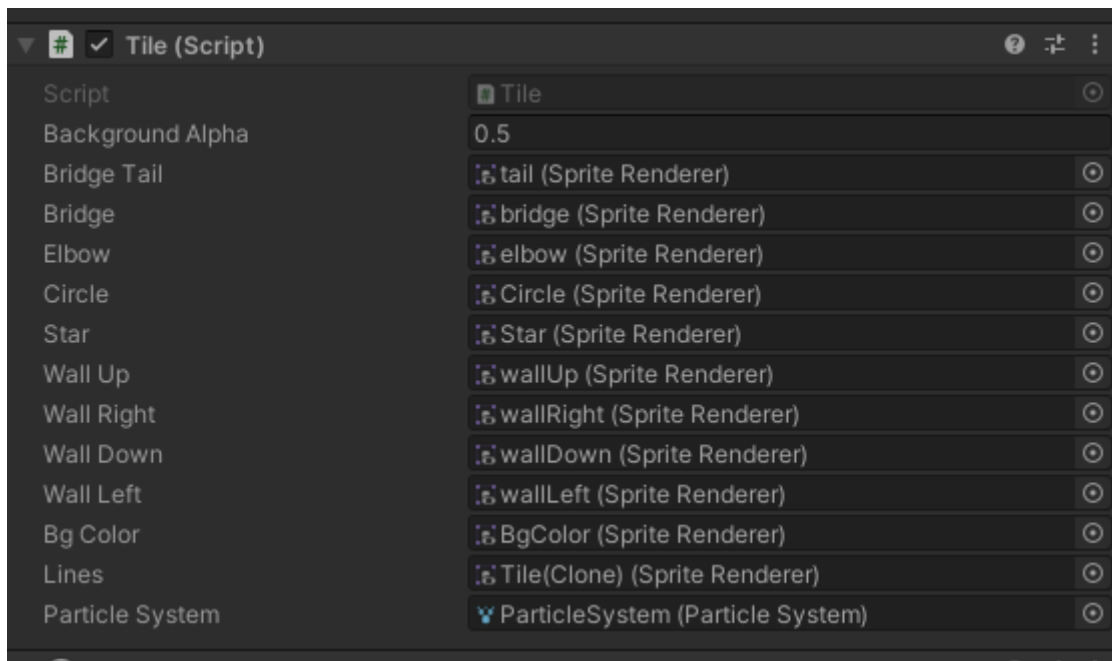
Tras haber seleccionado un nivel a través de la escena de selección de nivel, se inicializa, por un lado, el **tablero** mediante el script **BoardManager** y, por otro, el **HUD** mediante **HUDManager**.

Para ambos elementos, lo primero es inicializar cada uno de los objetos necesarios a través de los datos obtenidos mediante el **GameManager**. En el caso del hud, cada uno de los elementos del canvas está referenciado en el script mencionado para que cargue correctamente los datos.

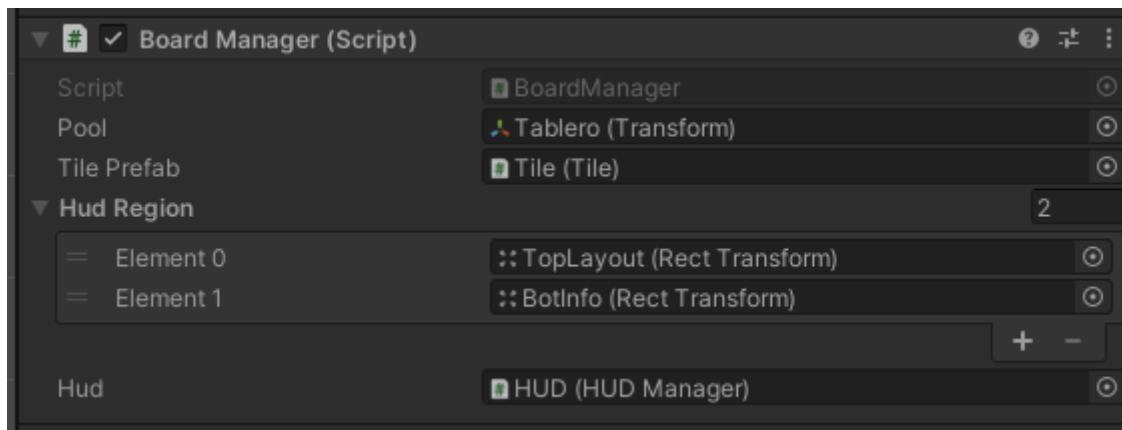


Además, posee métodos públicos para actualizar de forma dinámica los elementos necesarios para dar feedback al usuario como son el número de pistas, flujos conectados, porcentaje, movimientos, etc.

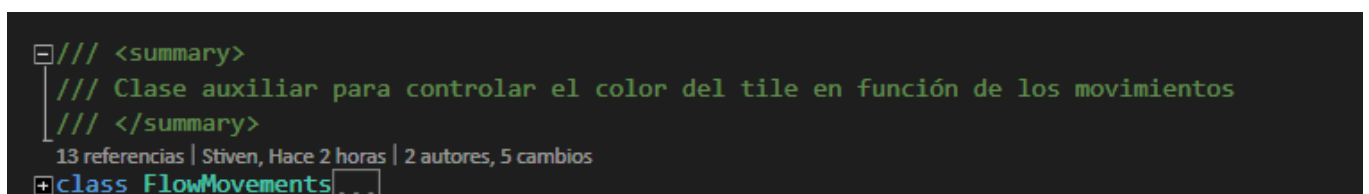
Por otro lado, para inicializar se instancian los distintos tiles en su posición correspondiente, los cuales poseen el script **Tile**, el cual se encarga de almacenar y gestionar la información necesaria de sí mismo, para que pueda ser modificado fácilmente a través del **BoardManager**,



quien se encarga de procesar la lógica del input para aplicar correctamente los cálculos lógicos al tablero, además de avisar al HUD sobre el cambio en la información del tablero.



BoardManager necesita también la información de lo que ocupa el hud, para obtener las dimensiones que ocupa éste y así poder escalar correctamente el tablero cuando se inicialice. Dentro de este script, existe una clase auxiliar **FlowMovements**, la cual contiene la funcionalidad necesaria para gestionar cada una de las tuberías del tablero, determinando de qué color hay que pintar cada tile, puesto que si una tubería ocupa un camino dentro del tablero, dicho camino ha de pintarse con las formas y los colores correspondientes a la tubería.



Así pues, será **BoardManager** quién se encargue de aplicar las distintas funcionalidades como son el deshacer un movimiento o aplicar una pista, también determina si el tablero está completo, cuántos flujos están conectados, qué porcentaje del tablero está completo y cuantos movimientos se están realizando durante la resolución del puzzle, entre otras funcionalidades lógicas necesarias para crear una jugabilidad fluida y estable.