

*Marco Iván Merino Hernández
Antonio Jesús Guerra Garduño
William Molina Cumba
4º U - Desarrollo de Videojuegos - UCM*

PRÁCTICA 2 : FLOW

Videojuegos para dispositivos móviles

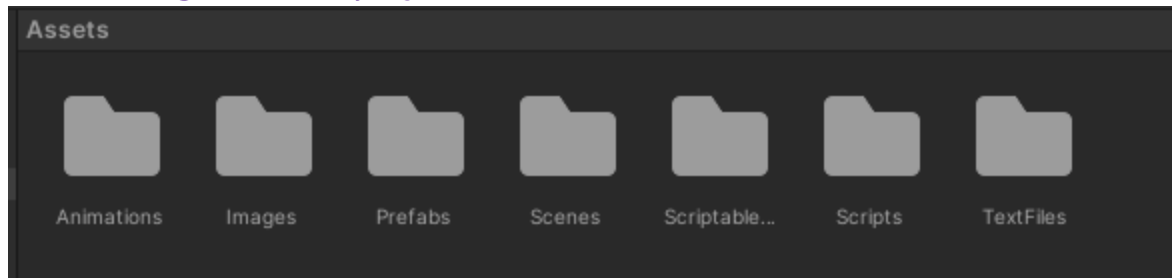
17 de Diciembre de 2021



Documentación técnica. Puntos a tratar:

- Estructura general del proyecto: Assets.
- Estructura general del proyecto: Ficheros.
- Estructura general del proyecto: Escenas y componentes.
- Estructura general del proyecto: Prefabs.
- Estructura general del proyecto: Relaciones.
- Características implementadas del juego
- ¿Cómo se hace?

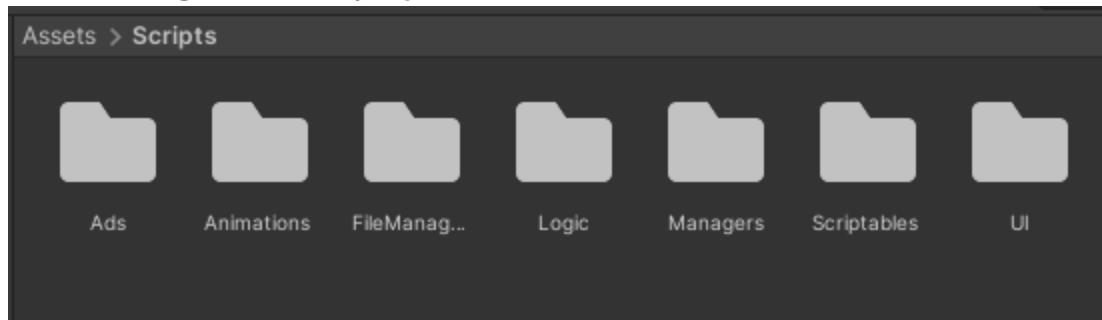
Estructura general del proyecto: Assets



Los Assets del proyecto están divididos en distintas categorías:

- **Animations:** Contiene las distintas animaciones que se han implementado.
- **Images:** Directorio para almacenar todas las imágenes del juego.
- **Prefabs:** Directorio para los distintos prefabs del juego.
- **Scenes:** Contiene las escenas del proyecto.
- **Scriptables:** Directorio para los distintos scriptables, estos se dividen en tres subcategorías:
 - **LevelLots:** Los relacionados con los lotes de las distintas secciones: Intro, Manía y Rectangles.
 - **Sections:** Los relacionados con las secciones implementadas: Intro, Manía y Rectangles.
 - **Skins:** Scriptables para las skins.
- **Scripts:** Directorio para el total de scripts del proyecto.
- **TextFiles:** Directorio que almacena los diferentes ficheros de texto para la carga del juego: Levels y maps.

Estructura general del proyecto: Ficheros

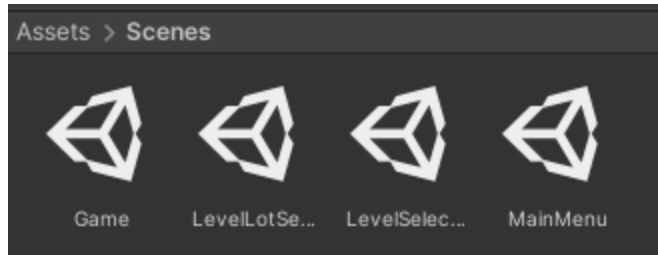


Los ficheros, al igual que los Assets están divididos en diferentes categorías. Cada una de las categorías engloba el total de scripts relacionados con el nombre de la carpeta raíz:

- **Ads: Scripts para el control de anuncios**
 - **AdManager:** Inicializa Advertisement y ofrece la creación de banners y videos.
 - **BannerAd:** Solicita y reproduce un banner.
 - **RewardedAdsButton:** Solicita y reproduce un vídeo, al terminar añade una pista.
- **Animations:** Script para el control de animaciones.
 - **GameBoxAnimController:** Métodos para controlar las distintas animaciones de los “GameBox” del juego.
- **FileManagement:** Scripts para el tratamiento de ficheros de texto
 - **MapParser:** Tratamiento del formato del string representativo de un nivel para la conversión de este en Map.
 - **SaveDataManager:** Contiene los atributos para la gestión del progreso, gestiona la serialización de datos y además carga/guarda/crea los datos de progreso. Responsable de la comprobación de Anti-Hacks.
- **Logic:** Scripts para el tratamiento de la lógica del juego
 - **Flow:** Lleva el control de los GameBox que constituyen una tubería, así como el control del corte de estas y los distintos estados.
 - **GameBox:** Clase para gestionar una “celda” del tablero.
 - **InputTransformer:** Transforma coordenadas en espacio de coordenadas globales a coordenadas locales.
 - **Map:** Clase que contiene la información de un nivel y ofrece funciones observadores para consultar datos.

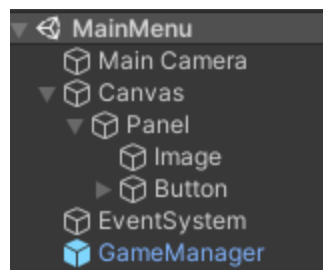
-
- **Managers: Diferentes managers utilizados en el proyecto**
 - **BoardManager:** Lleva el control del tablero y del input relacionado con este.
 - **GameManager:** Gestor del juego que sobrevive a nivel de escena. Ofrece funcionalidades y observadores al resto de Scripts para el control y la gestión del juego.
 - **LevelManager:** Manipula la escena de juego, ofrece control sobre los SubMenús de dicha escena, crea el BoardManager y gestiona el total de la UI de la escena de juego.
 - **Scriptables:** Scripts necesarios para los Assets “Scriptables” añadidos al proyecto.
 - **LevelLot:** Gestiona el comportamiento de colores, nombre del lote, textos de las páginas (5x5-fácil..), archivo de texto con todos los niveles y un bool para el control de desbloqueo de todos los niveles o no.
 - **Sections:** Gestiona colores y nombre de la sección y el conjunto de lotes de esta.
 - **Skin:** Colores de las tuberías y el nombre de la skin.
 - **UI:** Scripts para el tratamiento de UI de las diferentes escenas.
 - **SectionDisplayer:** Responsable de montar la segunda escena. Configurar scroll, crear headers, secciones, etc.
 - **HeaderVisuals:** Controla la parte visual de los encabezados para las secciones.
 - **LevelLotDisplayer:** Responsable de instanciar las páginas de los lotes a través de sus prefabs. Se entiende como página al espacio de escena (Grid) para un conjunto de 30 niveles.
 - **LevelDisplayer:** Responsable de instanciar los distintos niveles en la escena de selección de nivel y de personalizarlos visualmente en función al progreso y el tema correspondiente.
 - **LevelLotSelector:** Gestiona la selección de lotes en la segunda escena.
 - **LevelSelector:** Gestiona la selección de nivel en la tercera escena.
 - **LevelLotVisual:** Tratamiento de colores y progreso de forma visual en la segunda escena.
 - **LevelVisuals:** Tratamiento de colores y activación de las imágenes de un nivel y de forma visual en la tercera escena.
 - **SceneSwitchButton:** Tratamiento del cambio de escena.

Estructura general del proyecto: Escenas y componentes



El proyecto está compuesto por 4 escenas. Cada una de ellas representa una “Pantalla” diferente del juego. Menú principal, menú de selección de lote, menú de selección de nivel y pantalla de juego.

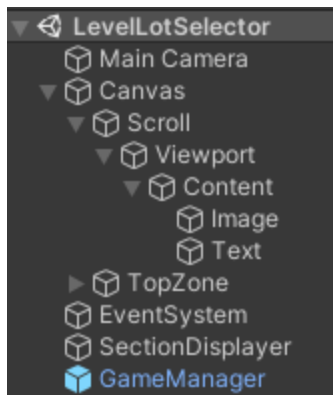
ESCENA 1: MAINMENU



Escena simple de inicio para el menú inicial.

Está compuesto de un Canvas con una Imagen y un botón para empezar el juego.

ESCENA 2: LEVELLOTSELECTOR

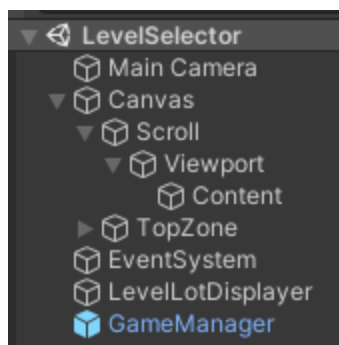


Escena para la selección de lote de juego.

Esta escena cuenta con un Canvas que tiene un Scroll Vertical y un ViewPort en el que se instanciarán por código las distintas secciones y lotes para elegir a cual jugar.

Además cuenta con una imagen de “Levels” y un botón para volver a la escena 1, así como componentes de Texto.

ESCENA 3: LEVELSELECTOR

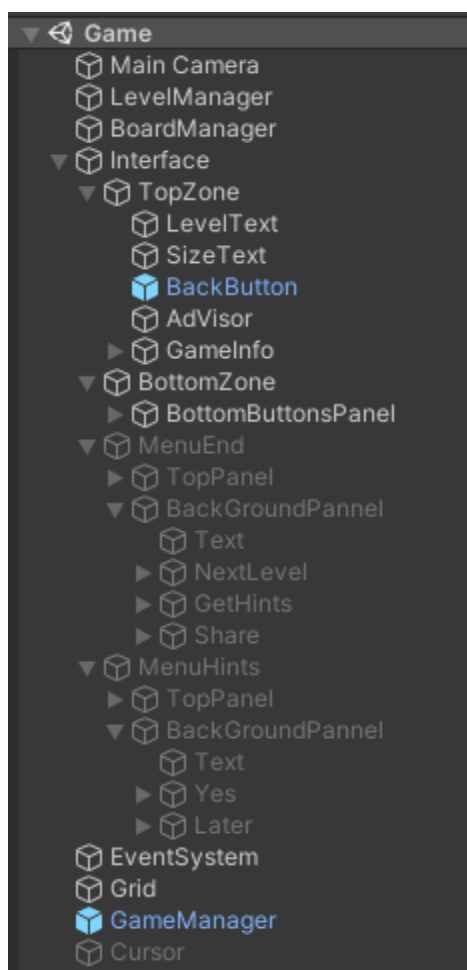


Escena para la selección de nivel de juego.

Cuenta con un Canvas que dispone de dos zonas principales: Scroll Horizontal para las distintas páginas completas de los lotes (se entiende por página completa la parte visual de un bloque de 30 niveles).

La TopZone cuenta con un botón para volver a la escena 2 y un componente de texto.

ESCENA 4: GAME



Escena para el desarrollo de una partida, escena de juego.

Cuenta con multitud de objetos vacíos que sirven para mantener en escena elementos necesarios como LevelManager, BoardManager y Grid.

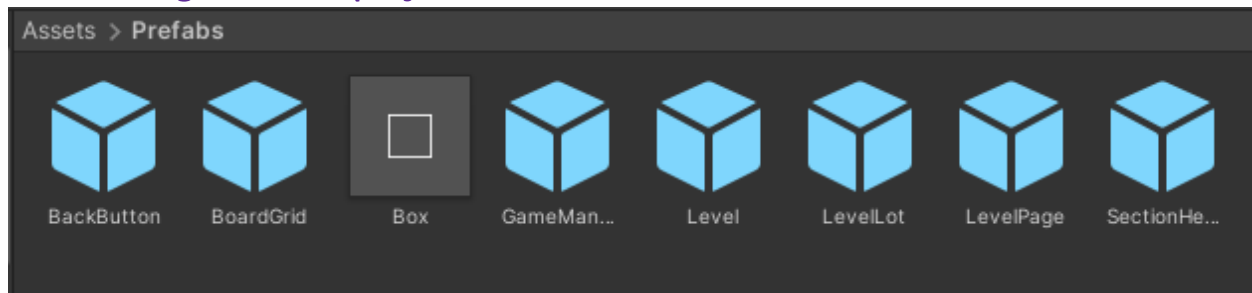
La zona de Interface tiene el contenido de los elementos visuales.

TopZone manipula la zona superior de la escena de juego, es decir, la relacionada con la información de la partida. Tiene componentes de UI como paneles, imágenes, textos y botones.

MenuEnd y MenuHints son los distintos SubMenús que aparecen en el juego. Tiene componentes de UI como paneles, imágenes, textos y botones.

En global, se han utilizado componentes tipo: Image, Sprit, Animator, Scripts, Renderers, Transforms, Buttons, Scrolls Layouts y Grid Layouts.

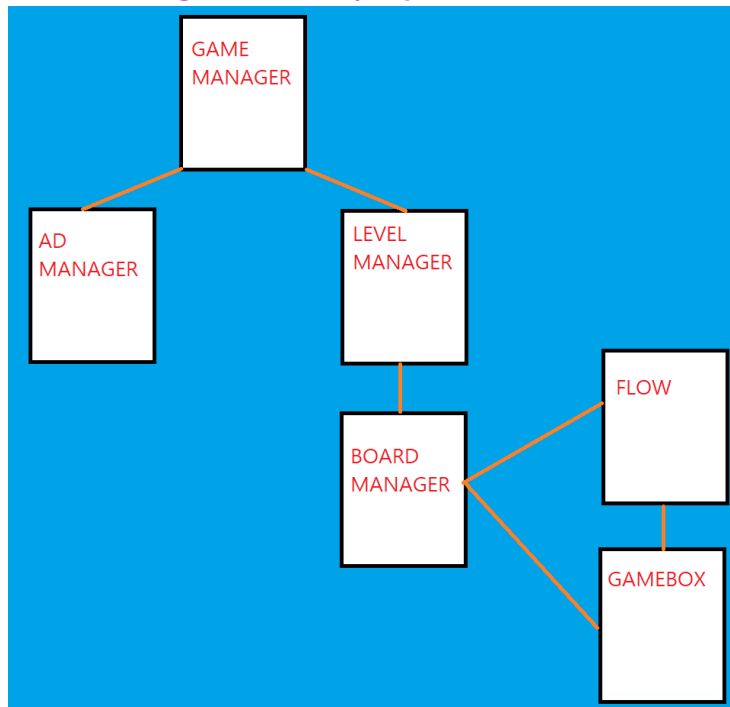
Estructura general del proyecto: Prefabs



Durante el desarrollo del proyecto se han ido creando una serie de prefabs para facilitar las tareas de gestión de instancias de objetos y creación de escenas:

- **BackButton:** Botón para volver a la escena anterior.
- **BoardGrid:** Prefab con un GridLayout y una imagen para instanciar el tablero.
- **Box:** Prefab de “celda” del tablero, está compuesto en esencia por imágenes.
- **GameManager:** Prefab del GameManager compuesto por un script y un gameobject.
- **Level:** Prefab compu
- **LevelLot:** Prefab para mostrar un lote en la escena 2 compuesto por una imagen y dos textos.
- **LevelPage:** Prefab para la representación de una página gestionado por script de la escena 3. Compuesto por un Grid Layout Group y un texto.
- **SectionHeader:** Prefab para el control de cabeceras gestionado por script de la escena 2. Compuesto por una imagen y un texto.

Estructura general del proyecto: Relaciones



Hemos creído relevante hablar, de forma general, de las relaciones que existen entre los ficheros de la lógica del juego, obviando la parte de UI por estar esta de forma más dispersa y ser menos relevante.

GameManager es el único que sobrevive entre los cambios de escena, por ello, su labor principal es mantener todas las escenas en concordancia y servir de observadora para el tratamiento de los distintos cambios que se producen en estas.

Aparte de ofrecer información sobre la sección, lote o nivel seleccionado y las características visuales se encarga de inicializar el AdManager cuando corresponde.

LevelManager consultará datos a GameManager y además responderá ante BoardManager para mantener la UI y el tablero Actualizados. Además solicitará a GameManager el guardado de datos.

BoardManager contiene un array con todos los Flows del nivel y el total de GameBox del tablero.

Cada GameBox tiene referencia al flow a que pertenece y a su vez, los Flows se comunica con BoardManager para la gestión de cambios en el tablero.

Características implementadas del juego

Al juego se han añadido las siguientes funcionalidades:

- Escenas de juego.
- Gestión dinámica de temas para la interfaz de usuario.
- Scrolls para visualización de elementos.
- Reconocimiento de niveles superados.
- Implementación del desarrollo de una partida.
- Implementación de pistas para el desarrollo de una partida.
- Feedback visual necesario en el transcurso de una partida.
- Botones para la facilidad de cambio y reseteo de nivel.
- SubMenús “Pistas” y “Fin de nivel” para la escena de juego.
- Animaciones.
- Visualización de anuncios.
- Obtención de recursos a través de visualización de anuncios.
- Carga y guardado de datos de progreso.
- Gestión visual de progreso.
- Protección anti hacking.

¿Cómo se hace?

A continuación se habla brevemente sobre cómo se han implementado las diferentes funcionalidades del juego.

- **Escenas de juego:** En su mayoría, están compuestas por un Canvas para el tratamiento de los elementos de UI y en aquellas escenas que son necesarios, los distintos GameObjects que han sido oportunos implementar.
- **Gestión dinámica de temas para la interfaz de usuario:** Cada sección tiene su color y una serie de información que en su conjunto, se almacenan en los ficheros de texto y en los Scriptables adjuntos al proyecto. El GameManager proporciona acceso a esta información al resto de elementos para que en aquellos sitios que sea necesario, se consulte y se pueda aplicar, como por ejemplo, en la segunda escena de selección de lote donde se muestran las distintas secciones con sus temas, en la tercera donde se muestran los distintos niveles o en la escena de juego para los colores de grid, tuberías, título del nivel y demás elementos.
- **Scrolls para visualización de elementos:** Existen dos scrolls principales. El primero es un Scroll Vertical que se utiliza en la escena de selección de lote para poder mostrar el total de secciones y lotes. En la escena de selección de nivel se utiliza un Scroll Horizontal para la gestión de los distintos bloques de niveles.
- **Reconocimiento de niveles superados:** Cuando un nivel está bloqueado, inicialmente se muestra con un candado y no se puede acceder a él hasta que no se supera el nivel anterior. No todas las secciones disponen todos los niveles bloqueados, para ser fiel al juego se ha tenido en cuenta aquellas secciones que aparecen bloqueadas y aquellas que dan acceso al total de niveles. Se mostrará información visual en las distintas escenas cuando el nivel se supere y se distinguirá los casos de haberlo superado con el menor número de movimientos o no.
- **Implementación del desarrollo de una partida:** La partida se puede jugar con normalidad de principio a fin y dispone de los elementos básicos como flows y muros interiores, además de las casillas de tipo hueco y los muros exteriores.
- **Implementación de pistas para el desarrollo de una partida:** Si el jugador dispone de pistas, puede hacer uso de ellas, si es posible ya que no siempre se puede aplicar una pista, como por ejemplo, si ya se ha pedido pista para el total de flows diferentes o si el/los único/s disponible/s tienen su camino ya formado de forma correcta. En caso de aplicar pista se creará el camino y se utilizará la estrella de forma visual para indicar a qué flow se ha aplicado. Las pistas contienen el total del comportamiento solicitado de la práctica.
- **Feedback visual necesario en el transcurso de una partida:** La escena de juego cuenta con elementos de UI para mostrar información como el número de flows conectados, el número de movimientos, etc. Además existe otro tipo de información visual, como por ejemplo el Cursor que aparece cuando se genera un input y se adapta al color de la tubería en tratamiento.

-
- **Botones para la facilidad de cambio y reseteo de nivel:** Se han añadido botones para poder cambiar al anterior/siguiente nivel sin necesidad del cambio de escena y otro para resetear el nivel actual.
 - **SubMenús “Pistas” y “Fin de nivel” para la escena de juego:** El juego cuenta con dos submenús en la escena de juego, uno para la gestión de vídeos que permite ver un vídeo para ganar una pista y otro para cuando se completa un nivel.
 - **Animaciones:** Existen tres animaciones principales:
 - Al hacer click en un extremo de una tubería, ambos extremos del mismo color harán un pequeño parpadeo.
 - Al conectar los dos extremos de una tubería, ambos crecerán.
 - Al completar el nivel todos los extremos de todas las tuberías crecerán.
 - **Visualización de anuncios:** Integración de videos y banners a través del sistema de AdVerisement de Unity. El banner inferior se carga con el juego y permanece siempre visible mientras que los vídeos se puede decidir cuándo verlos.
 - **Obtención de recursos a través de visualización de anuncios:** Al ver un anuncio completo de tipo Vídeo se ganará una pista.
 - **Carga y guardado de datos de progreso:** El progreso del juego se mantiene siempre actualizado mediante un fichero json y la gestión del GameManager. El progreso incluye: niveles desbloqueados y con cuántos movimientos, niveles superados con el menor número de movimientos, total de pistas acumuladas, total de niveles desbloqueados de cada lote.

Para minimizar el impacto sobre la pérdida de datos ante un posible crash de la aplicación, se ha tomado la decisión de guardar el progreso siempre que se genera o se gasta una pista y siempre que se completa un nivel.

- **Gestión visual de progreso:** El juego cuenta con la capacidad de adaptarse en todo momento a los diferentes eventos y cambios que se producen en relación al progreso. Por ejemplo, si se completa un nivel al volver a la escena de selección de nivel ya se reconoce esta información, al igual que si volvemos a la escena de selección de lote, donde podremos encontrar que el progreso ha aumentado.
- **Protección anti hacking:** Para el tratamiento de la manipulación de ficheros “offrecord”, cada vez que se guarda o se carga el progreso, se crea un hash sobre el json de datos de progreso con ayuda de una pimiento. Al cargar se comprueba que el hash generado en el último guardado coincide con el hash que se crea en tiempo de verificación para la carga.