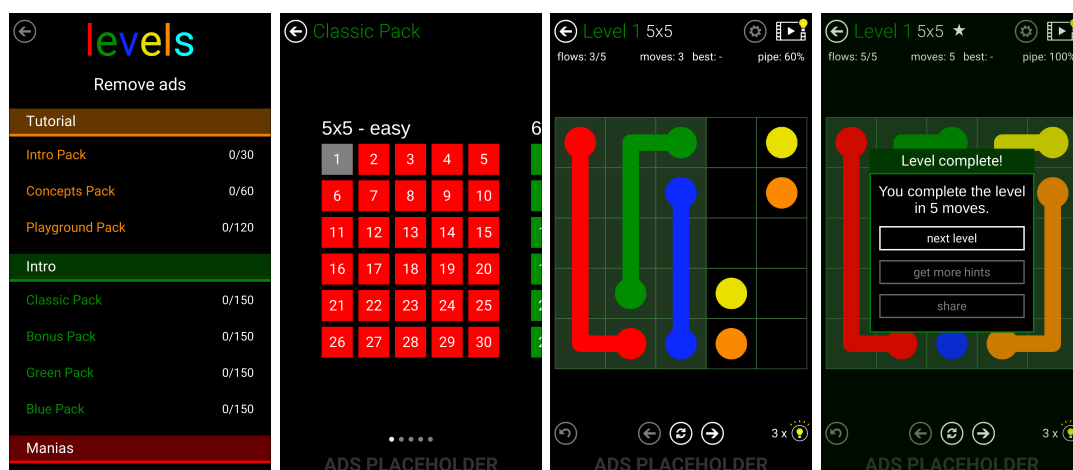

Práctica 2: Clon de *Flow Free*

Fecha de entrega: 17 de diciembre de 2021, 23:55

Objetivo: Desarrollo para móvil con Unity. Progreso, publicidad y bienes digitales



*Flow Free*¹ es un juego para móvil de la compañía *Big Duck Games* con más de 100.000.000 descargas. El objetivo del jugador es conectar los círculos del mismo color a través de tuberías para crear un “flujo” (de ahí el nombre). Al emparejar los colores se debe cubrir todo el tablero para resolver el puzzle. Pero, ¡cuidado! Las tuberías se romperán si se cruzan.

1. El juego

La práctica consiste en la implementación utilizando Unity de un *clon* del juego *Flow Free* que incorpore algunas de las características habituales en los juegos *freemium* para móvil.

Cada nivel del juego es una cuadrícula rectangular en la que aparecen distribuidas varias parejas de “sumideros” de colores. El objetivo del jugador es conectar los sumideros

¹<https://play.google.com/store/apps/details?id=com.bigduckgames.flow>

del mismo color con tuberías de modo que no se crucen y que ocupen todas las celdas del tablero. Los niveles tienen diferente nivel de dificultad, con distintos tamaños, número de flujos, huecos, paredes, puentes. . .

Para crear una nueva tubería basta con arrastrar un extremo en la pantalla, lo que va creando la tubería detrás. Si se pasa por encima de una tubería creada con antelación, ésta se cortará, aunque la rotura no se consolida en el tablero hasta que el jugador no deja de arrastrar la tubería en curso. Una tubería no puede unir sumideros de colores distintos. El nivel se supera cuando se crean todas las tuberías de tal manera que los sumideros del mismo color queden unidos y todas las celdas del tablero estén ocupadas.

Durante la resolución de un nivel, el juego cuenta el número de movimientos que se realizan. Se considera un movimiento la creación o manipulación de una tubería. Si se manipula la misma tubería que en el movimiento inmediatamente anterior, o si la tubería manipulada se deja en el mismo estado que estaba, no se cuenta como movimiento.

Los niveles están organizados en categorías (*Intro*, *Manías*, etcétera). Cada categoría tiene un color asociado, que se utiliza en diferentes puntos del interfaz (etiquetas de título, paredes del nivel). Además, cada categoría posee un conjunto de “lotes” de niveles (*Classic Pack*, *Bonus Pack*, etcétera). Al principio, el juego muestra un menú con la lista de categorías y sus lotes para que el jugador decida qué lote quiere jugar. Cuando se elige uno, se muestra la lista de sus niveles y el jugador debe elegir a cuál enfrentarse. Todos los lotes tienen una cantidad de niveles múltiplo de 30, por lo que éstos aparecen en la lista agrupados en “páginas” en el interfaz. Las páginas tienen un “título” que depende del lote.

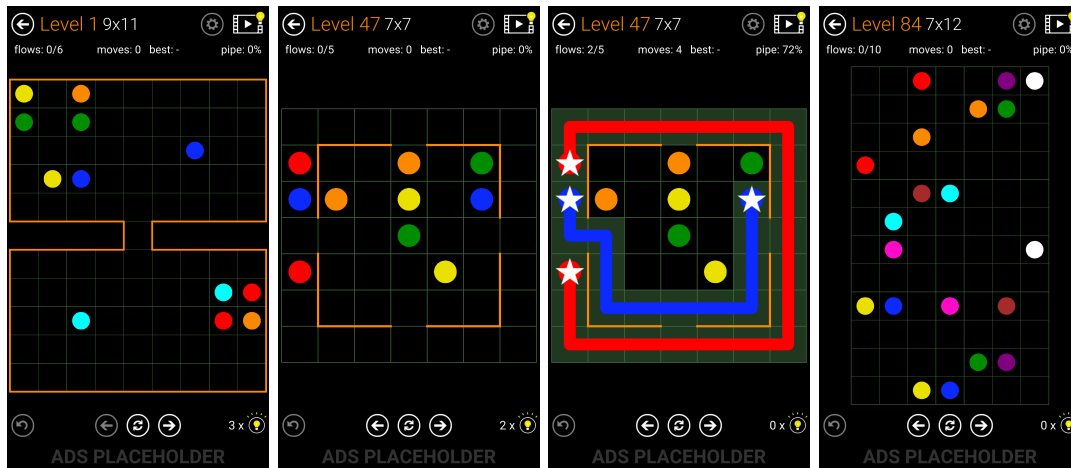
El juego *mantiene el progreso* entre sesiones, recordando qué niveles han sido ya superados y con cuantos movimientos, de modo que se sabe si han sido resueltos con el mínimo número de movimientos (que coincide con el número de flujos del nivel). Además, al instalar el juego algunos lotes tienen sus niveles bloqueados (salvo el primero) de modo que para jugar en los niveles superiores hay que superar antes los que le preceden.

El interfaz del juego muestra el número de flujos conectados hasta el momento (no necesariamente correctamente), el porcentaje de las tuberías creadas, y el número de movimientos realizados, junto con el mejor conseguido hasta el momento. También muestra una estrella o una marca si el nivel se ha jugado previamente, dependiendo de si se resolvió o no de forma óptima. Se permite reiniciar el nivel, y moverse al nivel anterior y siguiente (si está desbloqueado) sin tener que pasar por el menú de selección de nivel.

Además del progreso, el jugador acumula *ayudas disponibles* que le permiten *adquirir pistas*. El uso de una pista ayuda en la resolución del nivel colocándole uno de los flujos y marcando sus extremos con una estrella. Los flujos conseguidos por las pistas pueden seguir rompiéndose si se ponen otros nuevos que los cruzan, aunque la estrella indicando que es una pista resurgirá cuando se vuelva a colocar el flujo correcto. Cuando se consigue una pista, puede que se rompan flujos previos mal colocados.

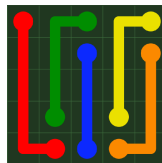
Las *ayudas* se consiguen a través de *vídeos publicitarios recompensados* (“*rewarded videos*”) que el usuario puede pedir ver para recibir una ayuda a cambio. En el juego original, también se pueden adquirir pistas en la tienda de la plataforma.

Los colores de las tuberías dependen del *tema de color* activado, que también se utiliza para colorear cada letra de algunas de las etiquetas de título del interfaz. El juego proporciona varios temas de serie, aunque también se pueden adquirir otros nuevos, al igual que niveles adicionales.



2. Detalles del juego

Los lotes de los niveles se guardan en ficheros de texto, en los que cada línea serializa un nivel. La línea está formada por varios “campos” separados por un punto y coma. A modo de ejemplo, la figura siguiente muestra el primer nivel de la categoría “Intro”, junto con su serialización. Por claridad, cada “campo” se ha separado en una línea, aunque en el fichero aparecen todas seguidas.



```
5, 0, 1, 5;
0, 5, 10, 15, 20, 21;
2, 1, 6, 11, 16;
7, 12, 17, 22;
4, 3, 8, 13, 18;
9, 14, 19, 24, 23;
```

El primer campo hace de cabecera, con información general sobre el nivel. Después van los flujos de la solución, con una secuencia de números separados por comas. Cada número hace referencia a una celda. Las celdas se numeran empezando por 0, de izquierda a derecha y de arriba a abajo.

La cabecera contiene varios campos separados por comas:

- Tamaño del tablero: si es cuadrado, tendrá un único número con las dimensiones de ancho y alto. Si no, aparecerán dos números separados por dos puntos (:). Por ejemplo 11:14 indica un tablero de 11 celdas de ancho y 14 de alto.
- Reservado (siempre es 0).
- Número del nivel dentro de su página. Se utiliza en los botones de la pantalla de selección de nivel.
- Número de *flujos* del nivel.
- Campo opcional con la lista de celdas que tienen *puentes*, separadas por dos puntos.
- Campo opcional con la lista de celdas huecas (inaccesibles), separadas por dos puntos. A veces tras cada número aparece información adicional que se puede descartar.
- Campo opcional con la lista de muros. Cada muro se separa de los siguientes con dos puntos. Un muro se especifica con el identificador de dos celdas consecutivas separadas por una barra vertical (por ejemplo 2|9).

Algunos niveles contienen indicaciones adicionales sobre características particulares, como celdas hexagonales o niveles “cíclicos” (en modo toroide), que no se describen aquí.

El juego original proporciona una gran cantidad de niveles, y es posible “adquirir” más (que vienen con el propio juego). La tabla siguiente muestra la información de tres de las categorías.

Lote	Fichero	Lote	Fichero	Lote	Fichero
<i>Intro</i> (#008D00)		<i>Manias</i> (#FF0000)		<i>Rectangles</i> (#FB8900)	
Classic Pack	1	6x6 Manias	17	Tower Pack	30
Bonus Pack	6	7x7 Manias	13	Rectangle Pack	24
Green Pack	3	8x8 Manias	7	Hourglass Pack	47
Blue Pack	4	9x9 Manias	9		
		10x10 Manias	10		

3. Requisitos de la implementación

El juego se implementará en Unity, y tendrá que funcionar en móviles y tablets en orientación vertical (retrato). Al menos deberá:

- Incluir la funcionalidad descrita para las categorías *Intro* y *Manias* y *Rectangles*, lo que incluye niveles con “huecos” y paredes. En caso de duda, se puede utilizar el juego original como referencia.
- Adaptar correctamente la visualización en pantalla de acuerdo a la relación de aspecto (ancho–alto) del dispositivo. El juego debe ser operativo en configuraciones en los que el alto sea mayor o igual que el ancho y con cualquier densidad de pantalla. El tablero de juego debe adaptarse para ocupar el máximo espacio disponible, ya sea a lo alto o a lo ancho sin superponerse al HUD. Éste deberá “anclarse” a los extremos de la pantalla.
- Leer los mapas de ficheros de texto.
- Gestionar los niveles y los temas *como datos*, de modo que añadir nuevos niveles o temas *no* suponga cambiar código (siempre que no incorporen características nuevas). En particular, se debe poder parametrizar, además de los propios mapas, el nombre de la categoría y los lotes, y el color asociado. Los lotes indicarán los títulos de las páginas y si están o no bloqueados al instalar la aplicación. Para los temas, se incluirán los colores de los hasta 16 flujos². Esto permitiría que el juego descargara niveles y pieles como DLC.
- Mantener el progreso del usuario (niveles desbloqueados y resolución óptima o no) y el número de ayudas disponibles, de tal forma que sea difícil romperlo y modificarlo externamente accediendo a los ficheros en el dispositivo.
- Mostrar anuncios al usuario (con UnityAds) para *monetizar* el juego. Se mostrará un *banner* en el pie de la ventana de selección de nivel y juego³, y otro intersticial al superar cada nivel. El usuario podrá lanzar vídeos recompensados durante la resolución de un nivel para conseguir una pista.

²El tema “*Classic*” predefinido del juego usa los colores #FF0000, #008D00, #0C29FE, #EAE000, #FB8900, #00FFFF, #FF0AC9, #A52A2A, #800080, #FFFFFF, #9F9FBD, #00FF00, #A18A51, #09199F, #008080 y #FE7CEC

³Las capturas muestran el texto “ADS PLACEHOLDER” en su lugar.

En general el juego deberá estar razonablemente *cerrado*, es decir proporcionar una experiencia de juego fluida y pulida. Se deberá tener cuidado, por ejemplo, con no permitir al usuario interactuar con el juego mientras se está proporcionando *feedback*, o cuidar los detalles poniendo, por ejemplo, icono a la aplicación. Desde el punto de vista de programación, el proyecto deberá estar ordenado, el código documentado y bien estructurado. En particular, por ejemplo, *no* se debe acceder a componentes de otros `GameObject's`, utilizar directorios `Resources`, buscar objetos en la escena por nombre o tipo, o hacer un sobreuso de `Update()` que ocasione un consumo de batería innecesario.

No es necesario incorporar la posibilidad de deshacer un movimiento (no descrito en este guión), marcar el fondo de las celdas ocupadas, “paginar” los lotes de niveles en el interfaz (basta un simple *scroll* normal) o pintar el fondo en movimiento del juego original.

4. Consejos de implementación

Empezad programando la funcionalidad básica con la lógica de un nivel, sin preocuparos de la carga desde ficheros, el progreso o la cantidad de ayudas disponibles. En la primera versión, incluso, podéis ignorar el ajuste del tablero a la pantalla.

Aunque tengáis que refactorizar código con frecuencia, tened siempre *un proyecto jugable* al que ir incorporando funcionalidades adicionales poco a poco. Probadlo con frecuencia en diferentes resoluciones de pantalla y relaciones de aspecto.

Pensad si necesitáis algún *manager*, para qué, y cuál va a ser el modelo general de comunicación y coordinación del juego. Decidid qué escenas vais a tener y cómo os vais a comunicar entre ellas. Plantead también el uso de *prefabs* que se instancien dinámicamente.

Para el desarrollo del HUD aprovechad las funcionalidades proporcionadas por Unity, con su sistema de *anclas*, y cálculos automáticos en los *canvas*. Explorad las posibilidades que proporcionan los *layout* y los manipuladores del tamaño, como los de los componentes `HorizontalLayoutGroup` y `ContentSizeFitter`.

5. Partes opcionales

Siempre que los requisitos básicos de la práctica funcionen correctamente y estén bien implementados, se valorará positivamente la incorporación de características adicionales:

- Facturación integrada de bienes digitales: que el usuario pueda eliminar los anuncios completamente tras un pago en la plataforma, o adquirir ayudas sin ver anuncios recompensados.
- Conexión con redes sociales: permitir al usuario compartir su avance en Twitter o Facebook a cambio de pistas.

6. Entrega de la práctica

La práctica debe entregarse utilizando el mecanismo de entregas del campus virtual, no más tarde de la fecha y hora indicada en la cabecera de la práctica.

Sólo un miembro del grupo deberá realizar la entrega, que consistirá en un archivo `.zip` con el proyecto completo de Unity eliminando los ficheros temporales. Se añadirá también un fichero `alumnos.txt` con el nombre completo de los alumnos.

Dentro del .zip incluid un documento con una breve descripción de la estructura general del proyecto, tanto desde el punto de vista de los ficheros como de las clases, componentes y su relación entre ellas. Indicad las características del juego que habéis implementado y cómo lo habéis hecho, incluyendo las posibles ampliaciones opcionales.

El .zip deberá tener como nombre los nombres de los integrantes del grupo con la forma Apellidos1_Nombre1-Apellidos2_Nombre2-Apellidos3_Nombre3.zip.

Por ejemplo para un supuesto grupo de solo dos personas formado por Miguel de Cervantes Saavedra y Santiago Ramón y Cajal, el fichero deberá tener como nombre DeCervantesSaavedra_Miguel-RamonYCajal_Santiago.zip.