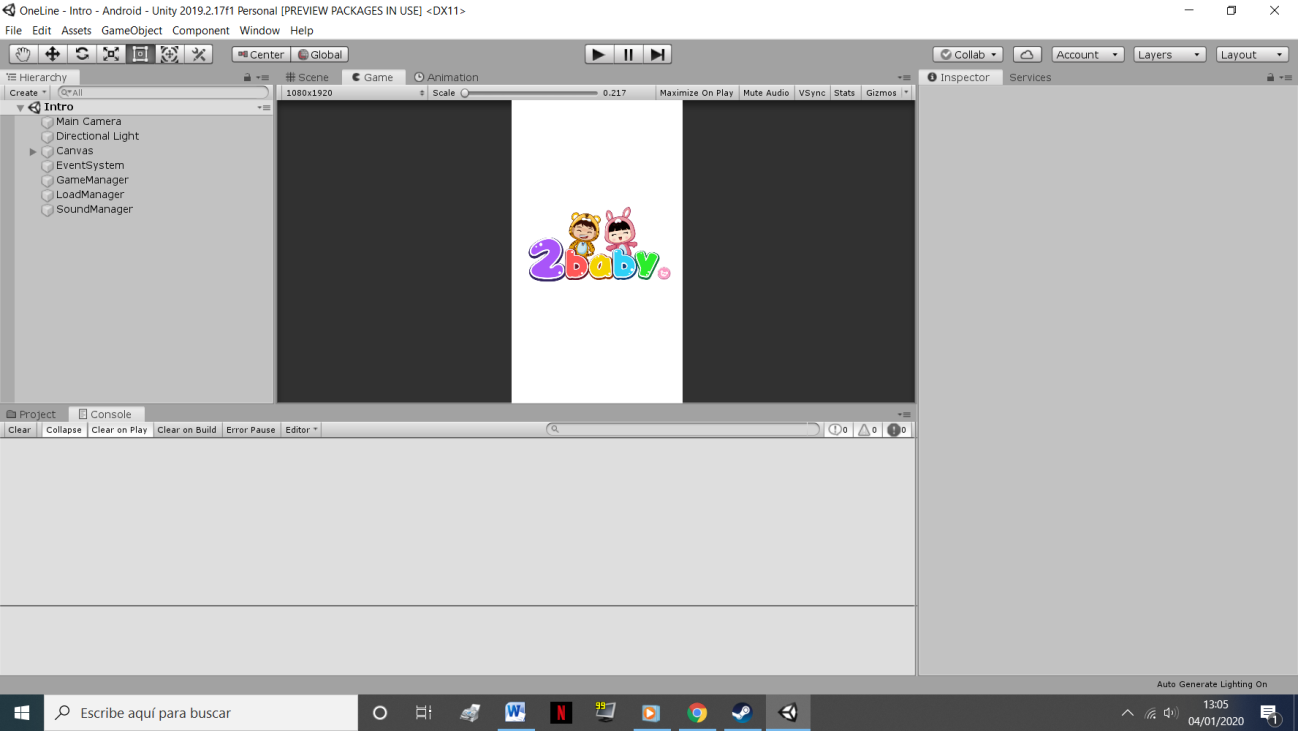
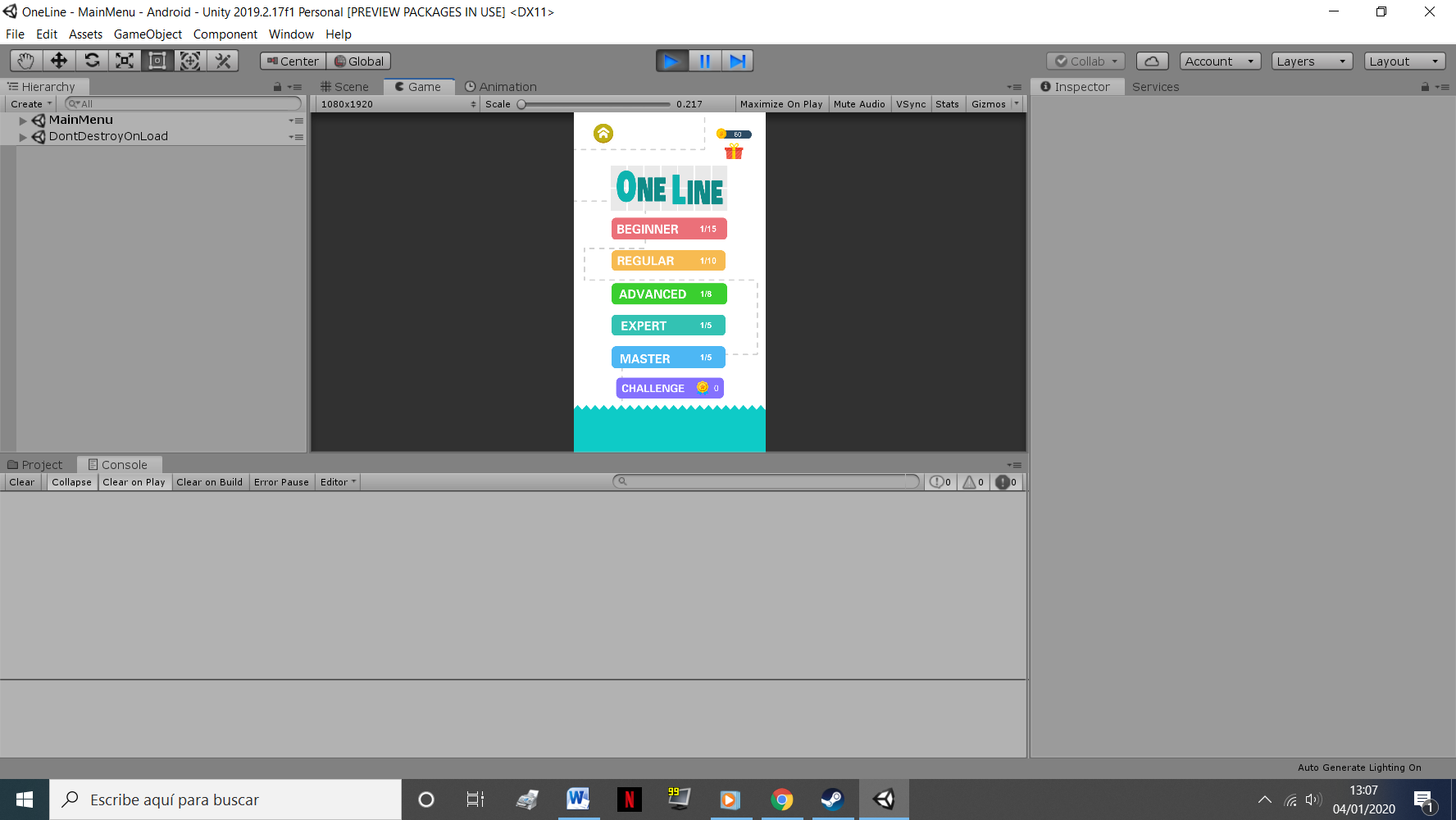
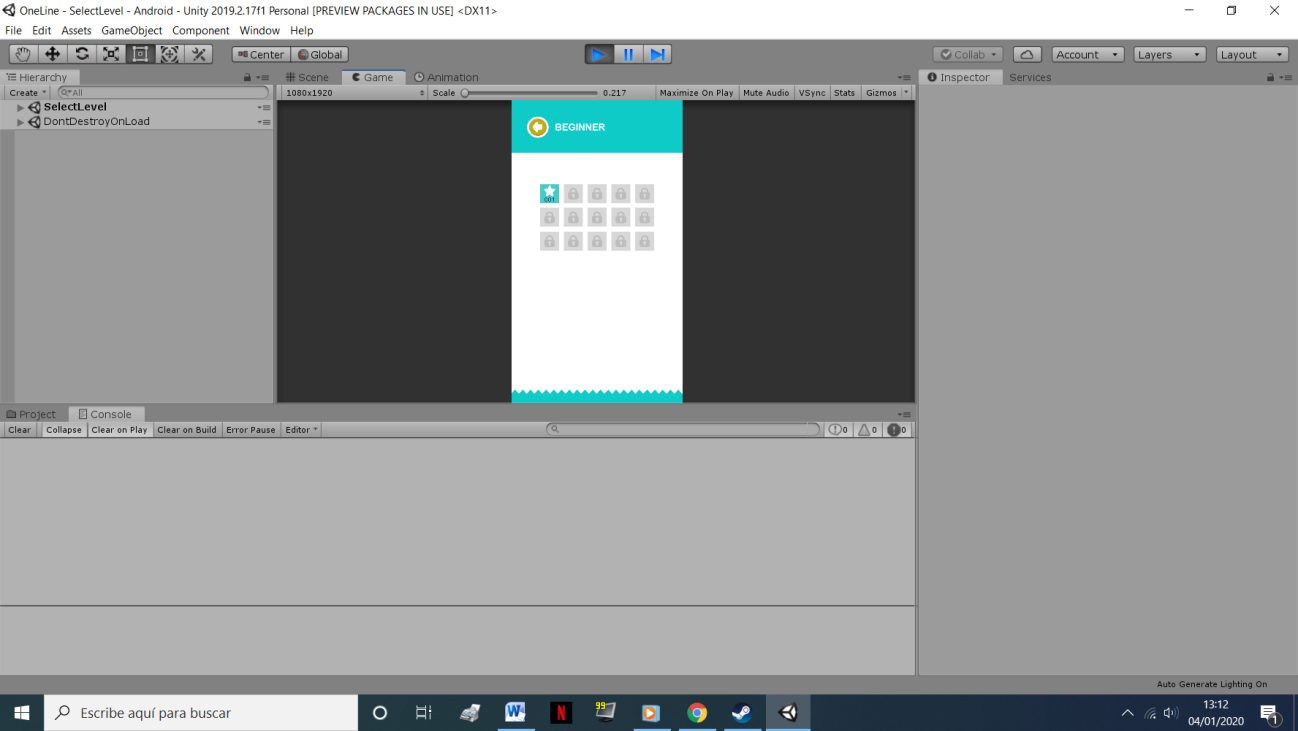
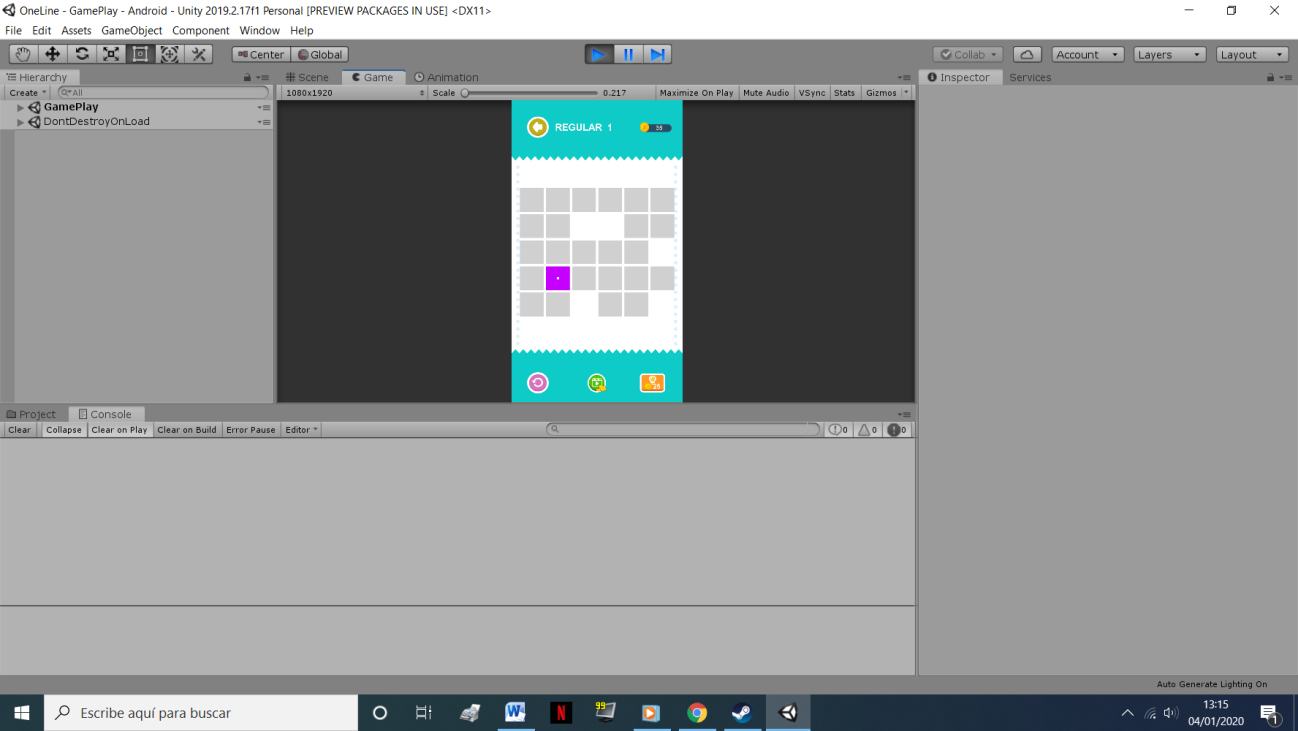
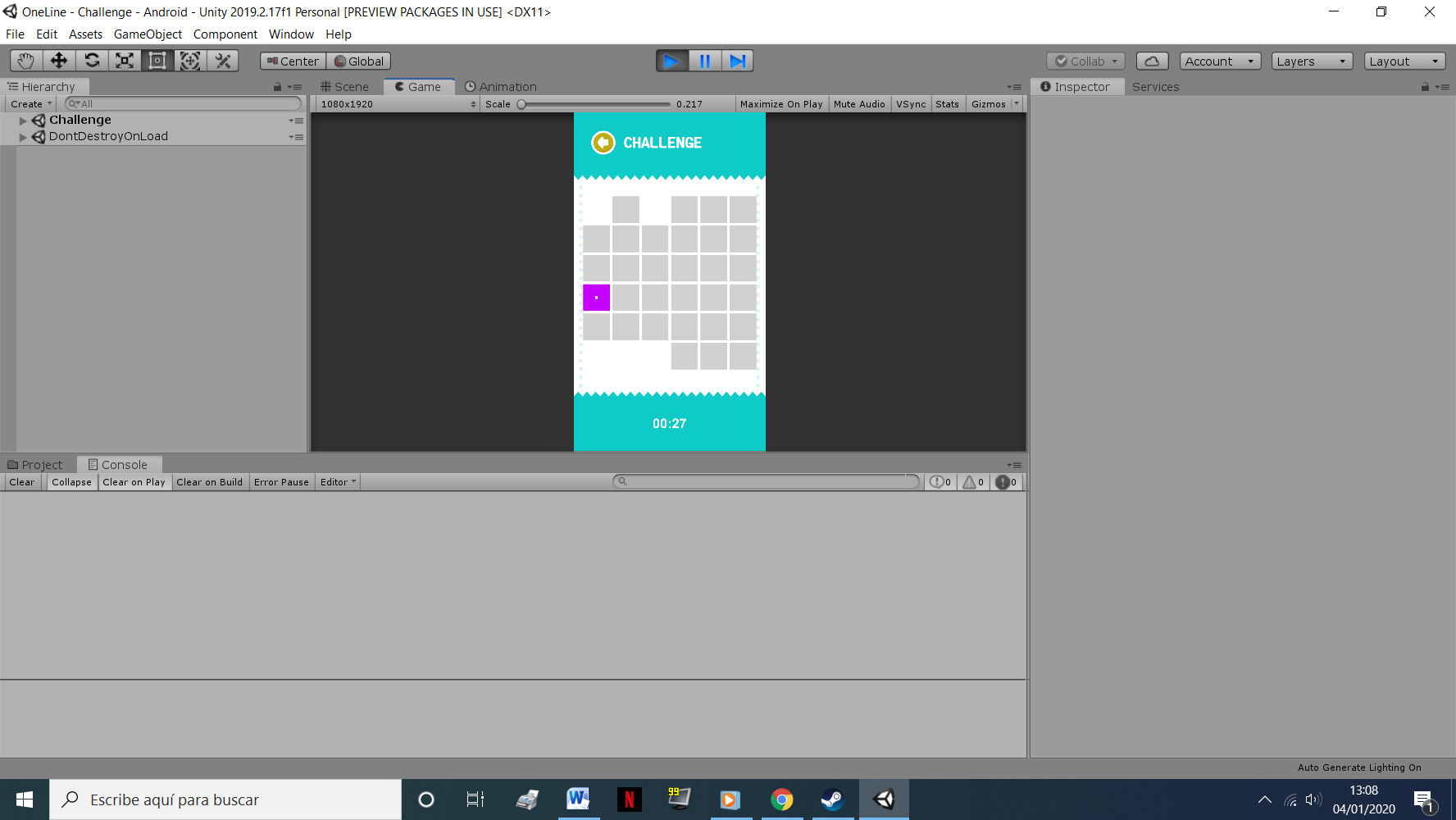
One Line

Alejandro Ortega Álvarez

Antonio Cardona Acosta

# Escenas

Primero explicamos un poco las 5 escenas que hemos decidido introducir.



3º SelectLevel

2º MainMenu

1º Intro

5º Challenge

4º Gameplay

1º Intro: Escena básica de no retorno para inicializar los managers globales que van a estar en todas las escenas sin destruirse, y para cargar el progreso antes de iniciar lo que es el juego.

2º MainMenu: Al igual que en el juego original, tenemos el selector de dificultad con los niveles de cada una que hayamos cargado. Tenemos también un botón para acceder al challenge, un marcador que muestra las monedas, un regalo que nos da monedas de vez en cuando y un botón para salir del juego.

3º SelectLevel: Sólo hay una escena de selector de niveles, que llamando al GameManager sabrá si ha de sacar los botones correspondientes a la dificultad Beginner, advanced, etc., ya que cada dificultad puede tener (y de hecho, tiene en nuestra práctica) un número distinto de niveles máximos a jugar.

4º Gameplay: Escena del juego que creará un grid de bloques correspondiente al nivel que el jugador haya hecho click en el selector de niveles. El comportamiento es igual que en el juego original, habrá un bloque de un color aleatorio pintado ya en el tablero y se puede comenzar a jugar a partir de él. Cuando se levante el dedo de la pantalla (o en la versión para ordenador, al hacer release del ratón) se comprobará si se ha completado o no el nivel. Hay botones para salir al menú anterior (perdiendo el posible progreso del nivel), para limpiar el tablero hasta el primero que hubiera pintado, otro botón para ver un vídeo que proporciona monedas, y otro para mostrar pistas si se tienen las monedas suficientes. Al mostrar pista, se limpia el tablero para ver con claridad el camino desde el primer bloque. No se restarán monedas si se hace click al botón de pista y ya se había completado el tablero entero con pistas anteriores.

5º Challenge: Nos pareció mejor idea el tener dos escenas diferenciadas para Challenge y Gameplay, eso nos ahorraba algo de código. El único inconveniente es que, a la hora de cambiar referencias públicas en el grid o en los managers, había que tener en cuenta que había que cambiarlas en las 2 escenas. Cuando se accede a un desafío desde el menú principal, comienza un nivel aleatorio entre las dificultades regular-advanced-expert-master que deberá ser completado en menos de 30 segundos para ganar monedas. Tanto si se ha conseguido como si no, no se podrá volver a jugar un desafío hasta pasados 30 minutos.

# Managers

A continuación explicaremos con detalle qué hace cada uno de los managers que hemos decidido añadir al juego.

**Managers globales** (no se destruyen entre escenas):

* GameManager: Gestor global de toda la información del progreso que va a ser guardada en un fichero, y que hace que cualquier escena pueda acceder o modificar dicha información (monedas, desafíos logrados, niveles completados, última vez que se jugó un desafío…). Toda la gestión y lo que haya que tener en cuenta para modificar todo lo referente a estos datos lo calculará el GameManager, desde fuera sólo se llamarán a sus funciones. Por ejemplo, a la hora de gastar monedas saber si tiene suficientes, o si al completar un nivel ya no quedan más por pasar de esa dificultad.
* LoadManager: Encargado de guardar los datos de progreso (serializados en un fichero) cuando se lo indique el GameManager (que puede ser en cualquier momento), y de cargarlos al inicio de la aplicación. Los datos se guardan en una carpeta Saves, que con borrarla es suficiente para comenzar a jugar desde 0.
* SoundManager: contiene un Audio Source con cada uno de los sonidos del juego, y es llamado desde otros managers para reproducir o parar los sonidos. Nos interesa que no se destruya entre escenas para que no se pare un sonido en seco si da la casualidad de que suena justo al cambiar de una escena a otra (por ejemplo, sonido del botón que comienza un nivel).

**Resto de managers:**

* AdsManager: Lo contienen las escenas en las que se puedan mostrar anuncios (MainMenu y Gameplay). Tiene dos funciones principales, una que muestra un vídeo si es posible (tal como se indica en la práctica, sólo era necesario mostrar rewarded videos) y otra función que se llama como callback sólo si el jugador ha visto el video entero, entonces conseguirá la recompensa que corresponda (indicada anteriormente).
* ButtonsManager: dado que puede haber dificultades con distintos niveles a pasar, era necesario un manager que creara botones en el selector de niveles dependiendo del nivel máximo pasable que existiera de esa dificultad.
* InputManager: recoge datos del input para ordenador si estamos en Unity Editor, o para Android en caso contrario. El InputManager sólo es necesario en el Gameplay y en el Challenge, ya que el resto de escenas era suficiente con añadir botones. Se comunica directamente con el Grid para ver si se ha pulsado, mantenido o hecho release encima de algún bloque.
* LevelManager: sólo en Gameplay Y Challenge, se encarga de mostrar los menús de win/lose cuando el grid le indique que está completo del todo o cuando se pase el tiempo en caso del challenge. También gestiona el ver si se puede comprar o no una pista.
* ScenesManager: script de ayuda para cambiar entre escenas. Todas las escenas lo contienen. Pero no interesaba que fuera global ya que nos facilitaba la referencia para ser llamado desde los callbacks de los botones.

# Resto de scripts

* ChallengeButton: control del challenge en el menú principal, el cual sabe si se puede jugar un desafío o, de lo contrario, tiene que mostrar el cronómetro de los 30 minutos dependiendo de lo que le diga el GameManager.
* Challenges: texto que indique el número de challenges que se hayan conseguido.
* Coins: texto que indique el número de monedas que se tienen en ese monmento.
* InfoLevelText: para el resto de textos que indican la dificultad y/o número de niveles.
* IntroLogo: efecto de aparición del logo de la intro.
* SelectLevelButton: Se le indica un nivel al ser creado desde el ButtonsManager para que cuando se le haga click indique al GameManager qué nivel se está jugando.
* Timer: Cronómetro. Se le proporciona un tiempo en segundos que empezará a descontar en cuanto se active el objeto. Cuando llegue a 0, llamará a quien tenga que llamar para decirle que ha pasado el tiempo.

En la carpeta Gameplay/

* BlockBehaviour: comportamiento de los bloques individuales. Se le indica desde fuera si tiene que ser tapeado/destapeado, o si tiene que mostrarse encima el camino o una pista.
* Grid: crea el tablero de bloques al comienzo del juego que le llega del parseador y los guarda en una matriz para poder tener la información de las posiciones. Tiene una pila con el camino recorrido de bloques para que sea fácil deshacerlo en caso de que se seleccione un bloque ya tapeado. Comprueba si puede tapearse un bloque porque está al lado del último tapeado, si tiene que quitar camino, si puede mostrar una pista, y si se ha completado el tablero del todo.
* Pointer: detalle visual que marca dónde se está pulsando con el dedo o con el ratón. Es siempre del mismo color que los bloques, sólo se ve en las escenas Gameplay y Challenge.
* SetRandomColor: contiene todas las “pieles”, pointers y hints en orden de color para que al comienzo de una partida se escoja un color aleatoriamente. Para introducir un nuevo color, simplemente se tienen que arrastrar los sprites a los arrays desde el editor de unity (pero si no cuadra el tamaño de los 3 arrays se cogerá siempre el primer color).

Por último, en la carpeta Parser está todo lo referido al parseado de niveles de .txt al juego.

Hay un txt para cada dificultad, conteniendo todos los niveles de dicha dificultad numerados desde 1 hasta máximo de niveles que haya en ese fichero. El mismo fichero contiene el tablero y la solución con un camino de chars, para las pistas. Para añadir un nuevo nivel, aparte de agregarlo al txt que corresponda, también habrá que indicárselo al GameManager desde un array público que contiene un número con el número máximo de niveles pasables por cada dificultad, y que puede ser modificado desde el editor.

Estos son los scripts:

* LevelInfo: información que contiene un nivel (filas, columnas, y matriz de bloques). La matriz de bloques contiene información del tipo de bloque (primer bloque, bloque normal o bloque vacío) y el orden para las pistas.
* LevelParser: es llamado desde el LevelConstructor, que le indica un fichero de dificultad y un nivel a parsear. El levelParser cogerá solo el tablero del nivel que le haya indicado y devolverá todo el LevelInfo de ese nivel.
* LevelConstructor: en el start de un nivel, coge el nivel actual del GameManager, llama al parseador para que le devuelva el LevelInfo, y se lo pasa al Grid para que construya el tablero. En el caso del Challenge, en lugar de coger el nivel del GameManager, elige un nivel aleatorio de entre todas las dificultades menos la Beginner (para que no sea demasiado fácil de pasar).

# Otros datos a tener en cuenta

* El menú de inicio no se puede pasar haciendo click, al igual que en el juego original.
* Se ha dado soporte tanto a una versión de ordenador como una para móvil. ( En verdad sólo cambia en el input en este juego).
* Como en la explicación de la práctica no vi nada del tiempo de aparición de los regalos en el menú principal, decidí dejarlo como recurso para obtener monedas infinitamente y así probar todo más rápidamente sin tener que estar tocando el GameManager.
* No estoy seguro de cómo funciona la aleatoriedad de niveles del Challenge, pero como no disponemos de muchos niveles decidimos que fuera un nivel aleatorio de entre regular y master, incluidos.
* Como ya te comenté en los correos, tuve que hacer cambios de versión para el tema de los anuncios. Primero la práctica la desarrollamos en la versión 2018.2 que no conseguía lanzar ningún anuncio. Luego pasamos a la 2018.3 que nos iba perfecto pero nos mosqueaba porque salía un error constante en el log de que no encontraba una dll y que podía crashear. Pero bueno, al cambiar a la versión de los laboratorios que nos dijiste (2019.2) para que va todo bien y que el error ese ya no lo da.
* Había dos sonidos en los recursos del juego que no conseguí ver a qué correspondían, así que los he quitado.
* Con respecto al reescalado de la pantalla, las imágenes del canvas y el tablero se reescalan correctamente pero los textos por lo visto no se reescalan de la misma forma, ya que se cambia el tamaño del objeto que contiene el texto y el tamaño de la fuente, y en ocasiones no se queda como debería. Preferimos poner texto a cargar las imágenes de las fuentes de los recursos del juego ya que apenas hay texto en el juego y la fuente además era blanca y sosa sin más.

# Partes opcionales

* Traté de hacer por mi cuenta lo de compartir en redes sociales pero me resultó complicado, así que cogí un script ya hecho de un youtuber que explicaba cómo hacerlo (en el script ShareImageCanvas.cs dejo puesto el enlace de dónde lo cogí). El script lo que hace es tomar una captura de pantalla del estado actual del juego y dar la opción de compartir en las redes sociales que el jugador tenga instaladas en su teléfono en ese momento. Decidí poner la opción de compartir cada vez que se completa un nivel.

No creo que te sirva dado que el código está tomado prestado pero ya que lo necesitaba probar para otro juego que estoy haciendo por mi cuenta, no me costaba nada meterlo aquí también.

El botón sólo funciona al hacer build para móvil, no hace nada en el editor de

Unity.