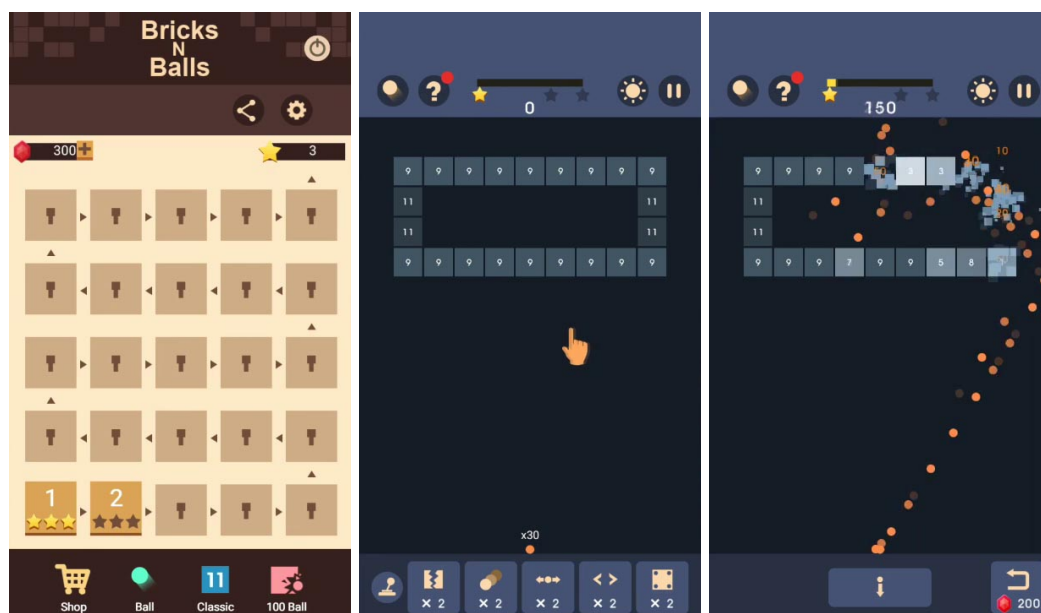

Práctica 2: Clon de Bricks'n'balls

Fecha de entrega: 7 de enero de 2019, 23:55

Objetivo: Desarrollo para móvil con Unity. Progreso, publicidad y moneda virtual



*Bricks'n'Balls*¹ es un juego para móvil de la compañía china *Cheetah Games*. Cada nivel contiene un conjunto de “ladrillos” que hay que destruir golpeándolos con bolas lanzadas desde abajo. En cada tirada, el jugador elige la dirección de su disparo, y una sucesión de bolas es enviada hacia arriba para que golpeen y reboten entre los bloques tanto como sea posible antes de volver a caer. Tras cada disparo, los bloques restantes descienden una posición, y hay que conseguir limpiar de ladrillos la pantalla antes de que el primero llegue abajo del todo.

¹<https://youtu.be/Tj2R8JPDgnc>

1. El juego

La práctica consiste en la implementación utilizando Unity de un *clon* parcial del juego *Bricks'n'balls* que incorpore algunas de las características habituales en los juegos *freemium* para móvil.

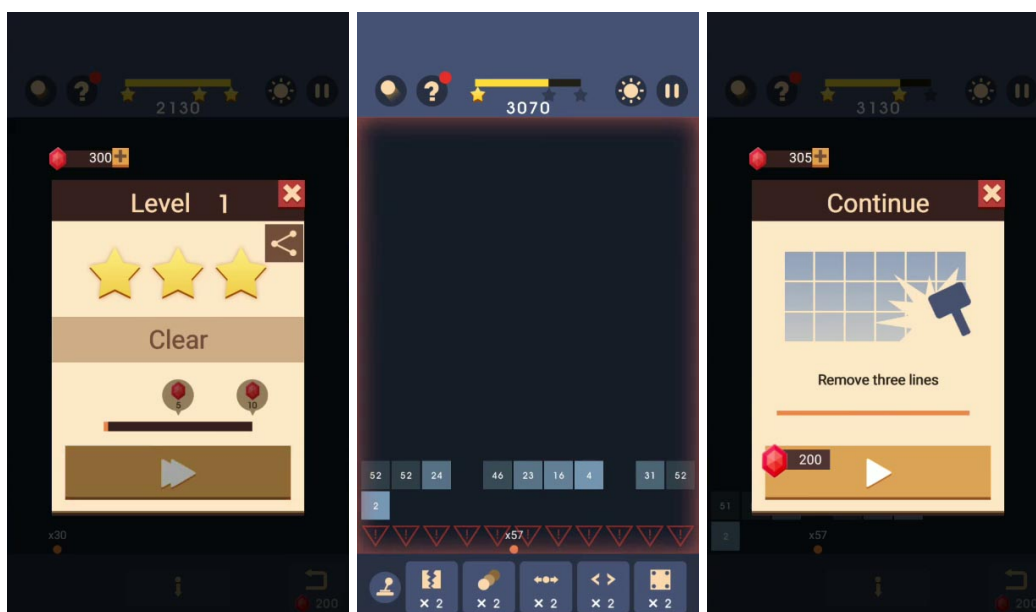
Cada nivel del juego consiste en un conjunto de bloques organizados en una cuadrícula que hace las veces de tablero. El tipo de bloque básico ocupa una celda completa, y debe ser golpeado un número determinado de veces para ser destruido. Existen diferentes variantes, como por ejemplo los que ocupan únicamente un *triángulo* de la celda completa.

Los niveles se superan realizando *disparos* desde abajo. Para cada uno, el jugador selecciona una dirección, y múltiples bolas serán enviadas desde una determinada posición en el borde inferior de la pantalla siguiendo la dirección especificada. Las bolas rebotarán y golpearán a los bloques hasta que terminen cayendo de vuelta abajo del todo. La primera bola que alcanza el límite inferior determina la posición exacta desde la que saldrán las bolas del siguiente disparo, y según van cayendo las demás se concentrarán en ese punto.

Tras cada disparo los bloques que no hayan sido destruidos descienden una posición. Dependiendo del nivel, la fila superior puede rellenarse con bloques adicionales, que permanecían ocultos fuera de la pantalla. Cuando se eliminan todos los bloques, se consigue un número de estrellas (entre 1 y 3) dependiendo de la puntuación y se pasa al nivel siguiente. Si, por el contrario, la fila inferior, justo encima de donde se disparan las bolas es ocupada por algún bloque, se avisa al usuario con un marco rojo. Si el descenso hace que los bloques lleguen a la posición de las bolas, el nivel no se supera y se debe volver a repetir.

El juego mantiene el progreso del jugador entre sesiones. Al lanzar la aplicación, se muestra la lista de niveles disponibles y cuáles, de todos ellos, están desbloqueados y con cuantas estrellas. El jugador puede elegir el que quiere comenzar a jugar.

Además del progreso, el jugador acumula “rubíes”, que hacen las veces de moneda virtual del juego con la que poder comprar potenciadores (*power-ups*) durante la partida.



Bloque	Id	Significado
	1, 2, 3, 4, 5, 6	Bloques sólidos
	7, 8	Emiten un rayo destructivo horizontal o vertical
	9	Bomba. Explota y elimina los bloques de su fila
	10	Como (2) pero no baja tras cada disparo
	12, 13, 16, 17	Como (1) pero abriéndose y cerrándose
	20	Bloques “de hierro” que no caen ni se destruyen
	21, 22, 23	Conceden una, dos o tres bolas durante el nivel
	24	Desvía ligeramente las bolas que lo tocan

Tabla 1: Algunos de los bloques

Power-up	Significado
	Terremoto: destruye un número diferente de cada bloque
	Rayo: coloca 4 rayos al azar
	Eliminar: elimina la fila inferior
	Añade algunas bolas solo en este disparo
	Añade 5 ladrillos de hierro en la parte inferior de la pantalla

Tabla 2: Power-ups disponibles antes de cada disparo

2. Detalles del juego

En el tablero entran 11 bloques de ancho, y son visibles 12 de alto, aunque siempre queda una fila vacía por encima, y otra por debajo (donde está el punto de disparo). Los mapas se guardan *en ficheros de texto*, con una primera parte indicando el tipo de cada bloque, y otra con el número de toques que deben recibir (si son bloques sólidos normales). La tabla 1 lista algunos de los tipos de bloques disponibles, junto con sus identificadores en el fichero. De los cuadrados, hay dos variantes, la segunda de las cuales *duplica* el número de toques necesario respecto al leído en el mapa. Los bloques de tipo rayo o desvío (7, 8, 24) actúan cada vez que son tocados por una bola, pero al acabar el disparo de su primer toque desaparecen. Los bloques de *bola extra* “caen” y quedan esperando en la parte inferior de la pantalla hasta que hay posición para el próximo disparo, momento en el que se concentran en él. Los bloques “semáforo” (12, 13, 16, 17) deben ser golpeados un número de veces como los bloques normales, pero alternan su estado entre abierto y cerrado tras cada disparo. De ellos, el 13 y 17 empiezan abiertos, y el 12 y 16 cerrados. Además, el 12 y 13 son bloques “clavados” que *no* descienden. Tampoco lo hacen los bloques 10 y 20, lo que puede impedir también que los que tienen por encima lo hagan. Los de tipo 20, indestructibles, quedarán al terminar el nivel. Tampoco es necesario destruir los especiales (bolas extra, rayos y desvío) para superar el nivel.

Antes de disparar, el jugador puede decidir invertir sus *rubíes* en alguno de los potenciadores enumerados en la tabla 2 a través de los botones que aparecen en la parte inferior de la pantalla.

El primer bloque sólido golpeado en un disparo proporciona 10 puntos. El segundo proporciona 20, el tercero 30 y así sucesivamente. Al ir consiguiendo puntos va creciendo una barra de desplazamiento y el número de estrellas conseguidas.

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:

- Adaptar correctamente la visualización en pantalla en función de la relación de aspecto (ancho-alto) del dispositivo.
- Leer los mapas de ficheros de texto.
- Incluir al menos un tipo de bloque especial, más allá de los sólidos (1...6) que se deben destruir.
- Mantener el progreso del usuario (niveles desbloqueados) de tal forma que sea difícil romperlo y modificarlo externamente accediendo a los ficheros en el dispositivo.
- Incorporar al menos un potenciador. Para ello será necesario incorporar *moneda virtual*, por lo que también deberá guardarse, de forma segura, la cantidad de rubíes acumulados. El modo de conseguir esos rubíes puede escogerse como se desee.
- Mostrar anuncios al usuario (con UnityAds) para *monetizar* el juego.

En general el juego deberá estar razonablemente *cerrado*, es decir proporcionar una experiencia de juego fluida y pulida. Desde el punto de vista de programación, el proyecto deberá estar ordenado, el código documentado y bien estructurado.

4. Consejos de implementación

Empezad programando la funcionalidad básica con la lógica de un nivel, sin preocuparos de la carga desde mapas, el progreso o los *power-ups*. 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 en si necesitáis algún *manager*, para qué, y cuál va a ser el modelo general de comunicación y coordinación general del juego. Decidid qué escenas vais a tener y cómo os vais a comunicar entre ellas.

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:

- Permitir al usuario solicitar la visualización de anuncios a cambio de rubíes.
- Facturación integrada de bienes digitales: que el usuario pueda eliminar los anuncios completamente tras un pago en la plataforma, o adquirir rubíes.

- Conexión con redes sociales: permitir al usuario compartir su puntuación en Twitter o Facebook a cambio de rubíes.
- Selección de la dirección del disparo con el acelerómetro (disparo en vertical)

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.zip`. Por ejemplo para el grupo formado por Miguel de Cervantes Saavedra y Santiago Ramón y Cajal, el fichero se llamará `DeCervantesSaavedra_Miguel-RamonYCajal_Santiago.zip`.