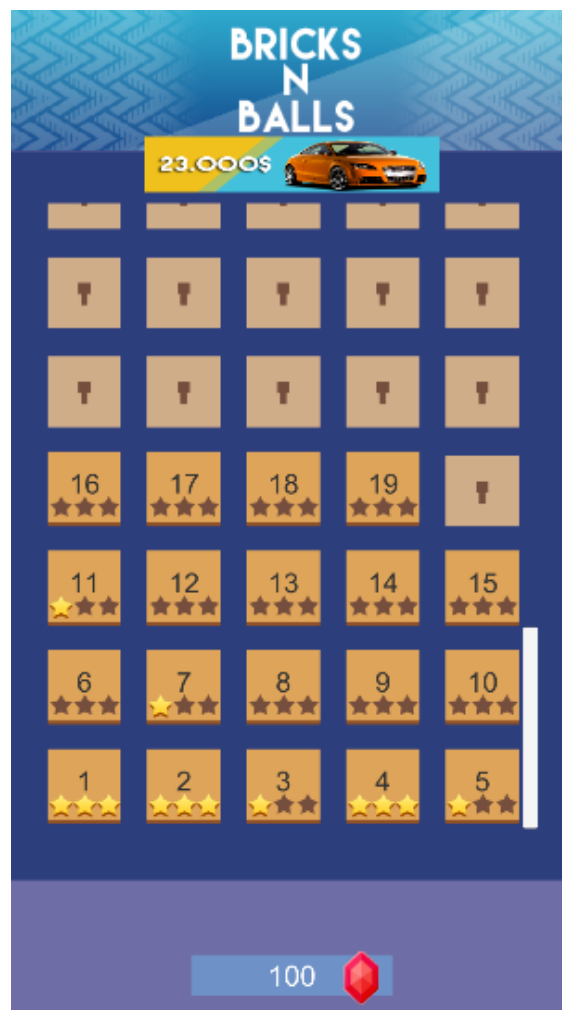


Bricks n balls – Práctica 2

Unity

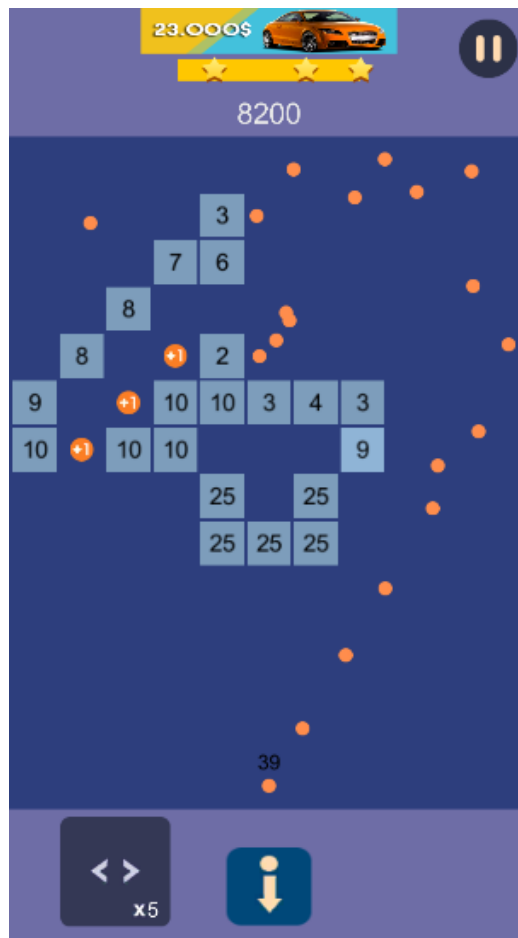
Menú de selección de niveles

Hemos utilizado un Scroll view, un elemento UI de Unity que permite realizar scroll vertical, para colocar los botones que permiten seleccionar los niveles en función de si están disponibles o no. También mostramos las estrellas obtenidas en cada nivel, el menú cuenta con publicidad y se muestran las gemas del jugador.



Escena de juego

La versión del juego que hemos implementado cuenta con los txts originales, los cubos básicos (1-6) y el cubo 21, que otorga una bola extra.



En los niveles generamos 11 filas de bloques y solamente se crean los bloques implementados, los que no, se quedan como un hueco vacío. Los bloques son almacenados en una lista del LevelManager.

En la parte inferior de la pantalla tenemos dos botones: (De izquierda a derecha)

- PowerUp: Todos los bloques de la escena pierden 4 puntos y tienen un efecto de “shake” o zarandeo. Sobre el botón aparece el número de veces que el usuario puede usarlo, cuando llegan a 0, aparece un texto pidiendo 100 gemas a cambio de otras 5 tiradas.
- Hacer retroceder todas las bolas que hay en pantalla (si las hay) y finalizar turno. Este botón solo aparece cuando hemos realizado un disparo.

Arriba hay un anuncio que otorga 100 estrellas si se termina de ver por completo, también hay un botón de pausa que nos lleva al menú de pausa, accesible también mediante el botón retroceso de Android:

Menú de pausa

Nos recibe con una animación de diamantes disparados en todas direcciones desde el anuncio (botón de x100), funciona igual que el resto de anuncios del juego.

El botón superior izquierdo (logo de casa) sirve para cerrar la aplicación, mientras que el de la derecha cierra el menú de pausa.

También se muestran las gemas poseídas en la sección superior del menú.

El botón inferior izquierdo reinicia el nivel actual, el botón central continua la partida pausada, y el último nos lleva al menú de selección de nivel.



Clases principales

- Ball: incluye métodos para:
 - o Acceder al rigidbody.
 - o Ser disparada (Punto origen, vector dirección + velocidad).
 - o MoveTo, corrutina que traslada la pelota a una posición en X tiempo.
- Bloque: el prefab contiene un gameObject hijo vacío que tiene un textMesh desde el que accedemos a través de este script, además:
 - o SetTipo: determinamos el tipo de bloque según el código numérico.
 - o SetPosition
 - o Corrutina de descenso de una posición, para que quede un efecto suave. Además de descender, comprueba si la posición destino es crítica (para avisar al jugador) o incluso si ha perdido la partida.
 - o OnCollisionEnter para gestionar el golpe de las pelotas y disminuir el contador de golpes.
 - o OnTriggerEnter para los bloques especiales que no colisionan con la bola.
- DeadZone:
 - o OnCollisionEnter comprueba que es objeto de tipo Ball y lo envía al LevelManager para aumentar el contador de bolas llegadas, destruirla, etc.
- Disparador (Spawner): el único con FixedUpdate para el Input
 - o Contiene el prefab de Ball para instanciarlas, variables como la fuerza del disparo, el contador de bolas, un line renderer para dibujar la estela del cursor al spawner.
 - o Métodos para establecer la posición "auxiliar" cuando termine el turno efectuado.
 - o Corrutina de disparo
 - o Método para interrumpir disparo activo.
- LevelManager: encargado de los elementos en escena, con listas públicas para las bolas, los bloques, variables para el control del disparo (saber cuándo ha terminado), los objetos de la escena para colocarlos (gamefield, bordes de pantalla...), variables para la puntuación y el guardado...
 - o Alertar: activa la pantalla roja, este método es llamado por los bloques cuando están en una coordenada Y concreta.
 - o IniciarObjetos: instancia prefabs
 - o colocarObjetos: en función del gameField se colocan los bordes de pantalla, spawner, deadzone...
 - o resizeCamera: obtenemos el aspect ratio deseado.
 - o iniciarContador: cuando se produce el primer disparo, se llama a esta corrutina que se llama así misma cada diez segundos para comprobar si quedan bolas en partida y de ser así, acelerarlas.
 - o LlegadaPelota: aumenta el contador de pelotas, las manda ir al spawner, las destruye... llamado desde la deadzone.
 - o Boton para hacer volver al spawner la lista de bolas y terminar el turno.
 - o Métodos para recargar nivel o avanzar al siguiente.

- GameManager: encargado de la lectura de txt y de Instanciar los bloques en la escena.
- UnityAdsButton: funcionalidad básica para mostrar un anuncio por defecto.
- Tuitear: colocado sobre un botón, este método abre en el dispositivo el navegador o la aplicación de Twitter para publicar un Tweet con el score actual del jugador.
- SaveLoad: Es la clase que se encarga de gestionar el archivo que serializamos. Tiene una variable que es la que se utiliza en el resto de clases que controlan las variables de los niveles. Además, contiene los dos métodos necesarios para controlar el archivo serializado, Save y Load. Estos métodos, crean o abren el archivo y lo serializan y deserializan, respectivamente.
- Game: contiene un diccionario con la info necesaria de cada nivel. Su índice y su puntuación.
- Level: Esta clase contiene la puntuación y el número del nivel que se está jugando.