

BLOCKS, OUT!

Memoria – Blocks, out!

Multimedia

Daniel Loro Durán

Índice

1. Introducción, descripción y motivación	2
2. Consideraciones de diseño	2
3. Tecnologías usadas.....	3
4. Dificultades encontradas.....	4
5. Posible trabajo futuro.....	5
6. Conclusión	5

1. Introducción, descripción y motivación

Durante el curso 2019-2020 se nos ha pedido el desarrollo de un proyecto libre para la asignatura de Multimedia, siempre que se utilicen elementos de la misma.

En este caso, y en vista de que otros años también se han realizado, se ha optado por el desarrollo de un videojuego en 2D para PC llamado **Blocks, out!**.

Blocks, out! es un juego tipo Arkanoid, en el que utilizando el teclado, la idea es desplazar una barra por la parte inferior de la pantalla para hacer rebotar una pelota en la misma e ir destruyendo los bloques de la pantalla para ir sumando puntos. Para ayudar y/o dificultar al jugador, de forma aleatoria aparecerán potenciadores (vidas extra, sumas y restas de puntos etc.), cada uno con una probabilidad propia, que el jugador debe recoger o evitar con la barra para aplicar o no sus efectos.

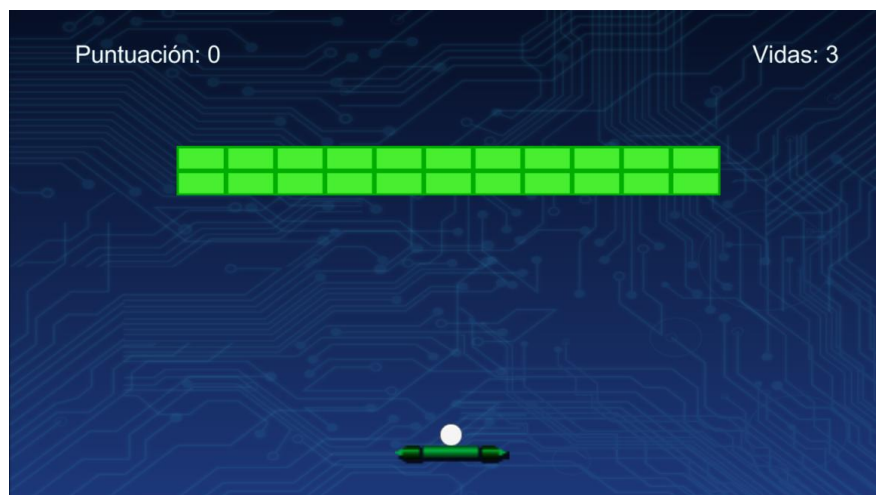
La dificultad del juego está diseñada de tal manera que en el primer tercio de niveles, hay más probabilidades de que aparezcan potenciadores de ayuda, en el segundo tercio, existe un equilibrio entre potenciadores de ayuda y aquellos que dificultan la partida, y en el último tercio, existen más probabilidades de que aparezcan potenciadores que dificulten la partida.

Además, para no saturar la pantalla de potenciadores, de cada bloque solo puede aparecer uno al mismo tiempo.

Como motivación, la idea de desarrollar un juego siempre me ha parecido bastante y llamativa, pero nunca me había animado a ello y creo que realizando un juego “sencillo” (dentro de lo que cabe), es buena idea entrar en este mundo.

2. Consideraciones de diseño

Como se menciona en la introducción de esta memoria, Blocks, out! parte de la idea del mítico Arkanoid, por lo que el diseño está basado en él.







Respecto a lo que no se ve en la interfaz, destacar que los bordes de colisión de los laterales están ambos inclinados ligeramente hacia dentro para evitar que la bola vaya de un lado a otro de la pantalla de forma infinita y no poder terminar el nivel o partida.

Los diferentes sprites del juego (bloques, potenciadores y el logo) han sido diseñados por mí utilizando herramientas que se explicarán más adelante.



Nótese que los bloques carecen de color de relleno, debido a que estos se colorean dentro del propio Unity. De algunos de estos potenciadores destacar que:

- Los potenciadores que restan puntuación (10^- , por ejemplo) lo hacen siempre y cuando tengas como mínimo un punto y nunca se va a tener una puntuación negativa.
-   y   hacen agrandar y encoger la barra, respectivamente, pero lo harán siempre hasta cierto punto.
- Casi todos los bloques son destructibles, aunque hay unos que necesitan más golpes que otros (también dan más puntos). Los únicos indestructibles son los grisáceos



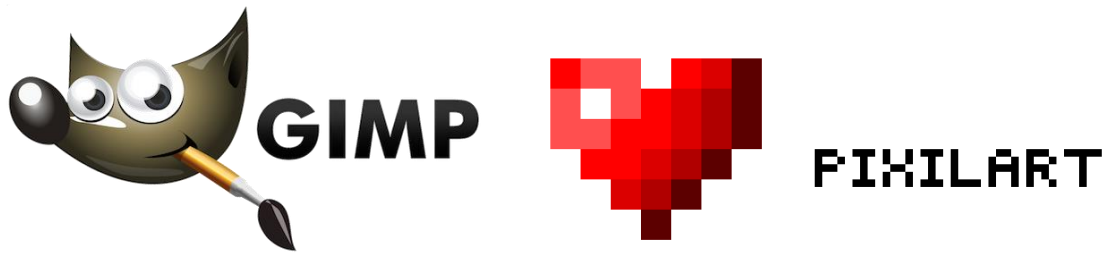
, y no se tienen en cuenta para la finalización de un nivel.

Este juego también cuenta con música y sonido para la carga de niveles, adquisición de potenciadores, pérdida de vidas etc. Un detalle a tener en cuenta es que, en ciertos potenciadores, el audio de aquel que va a perjudicar al jugador si lo adquiere es el inverso del que ayuda al jugador (por ejemplo, entre el que da una vida y el que te la quita).

3. Tecnologías usadas



Blocks, out! ha sido desarrollado con el motor gráfico **Unity** utilizando su licencia gratuita, una plataforma de desarrollo de videojuegos para diferentes plataformas (Windows, Mac, Android, iOS etc.), tanto para juegos en 2D como para juegos en 3D. En este caso, los scripts se han escrito en C#, lenguaje similar a otros de los cuales tengo conocimiento de uso debido a diferentes asignaturas de la carrera, utilizando **Visual Studio**.



En cuanto a los sprites mencionados en la sección anterior, se han realizado con la herramienta **GIMP**, un software gratuito de edición de imagen, y con la herramienta web **Pixilart**, diseñada para hacer imágenes en pixel art.



Estos audios han sido editados y recortados con la herramienta **Audacity**, un software gratuito para la edición de audio.

4. Dificultades encontradas

Principalmente, me he encontrado con tres problemas o dificultades.

El primero de ellos, el aprendizaje de las colisiones entre objetos, los rebotes y la instanciación y destrucción de los mismos de los mismos. Esto, evidentemente, ha sido solucionado para el correcto funcionamiento del juego.

El segundo, el emulador y el editor gráfico en tiempo real de Unity. Por alguna razón que no entiendo, la resolución tanto del emulador y del editor gráfico no son las mismas que la que se usará luego en la pantalla donde se va a probar, por lo que las colisiones de la bola entre los bordes laterales deben ser adaptadas de tal forma que, mientras en el propio Unity parece que la bola rebota fuera de la pantalla, en la versión ya compilada y funcional lo hace de forma correcta.

Por último, existe un bug con la velocidad de la bola. A partir del nivel 2 y en adelante, en la primera partida de cada nivel, la bola sale con una velocidad demasiado alta, algo que se soluciona a partir de la segunda partida de cada nivel.

5. Posible trabajo futuro

La idea de introducir niveles que se generen de forma procedural me pareció muy interesante, pero debido a ser muy novato con Unity y a no encontrar algún tipo de tutorial que se adapte a la forma en la que se ha desarrollado el juego, no ha sido posible su inclusión.

Dentro del código de los sprites del juego, existe un potenciador de una bomba (se muestra a continuación) que, finalmente, no ha sido utilizado donde la idea era que, al tocar dicho ítem, se eliminasen de forma aleatoria un número de bloques (evidentemente, este ítem tendría muy baja probabilidad de aparecer para no facilitar en exceso la partida al jugador).



Por último, y puestos a probar el desarrollo entre diferentes plataformas, me gustaría llevar este juego a Android (y posiblemente a iOS, aun no teniendo un dispositivo de Apple para hacer pruebas) cambiando el control con teclado por el uso de la pantalla táctil, para comprobar qué tan fácil o difícil es desarrollar en dichas plataformas con Unity.

6. Conclusión

La posibilidad de elegir un proyecto totalmente libre (siempre que se usen elementos de la asignatura), me ha resultado bastante útil de cara a aprender cosas básicas con el desarrollo de Unity, el cual siempre me había llamado la atención pero nunca me había animado a probar, por lo que este me parece un buen punto de partida de cara a seguir aprendiendo cosas nuevas y experimentando con este juego, o empezar de 0 otros juegos.