



# Desarrollo de videojuegos en Android

DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES

PRÁCTICA 3 – GRUPO 9

MARIO SIMÓN SOTO

BORJA MARTÍNEZ REQUENA

## Índice

1.- Introducción .....	2
2.- Implementación específica de requisitos .....	2
Arquitectura básica .....	2
Controles .....	3
Enemigos .....	3
Información en pantalla durante el juego.....	3
Fin de partida .....	4
Diseño.....	5
Disparo múltiple .....	6
Paralaje.....	6
3.- Conclusiones .....	7
4.- Referencias.....	7

## 1.- Introducción

El objetivo de esta práctica es utilizar el motor empleado en las prácticas guiadas para hacer un videojuego para Android de tipo *matamarcianos*. Para esto se ha utilizado de base la última versión del proyecto que se ha hecho en clase y luego se han ido introduciendo mejoras o nuevas funcionalidades.

## 2.- Implementación específica de requisitos

A continuación, se comentarán los requisitos del enunciado que se han implementado y los medios utilizados para ello.



*Ilustración 1: Menú principal del juego*

### Arquitectura básica

Se pedía que el juego tuviera al menos un menú principal, un menú de pausa, una pantalla de fin de partida y el propio nivel del juego.

La función del menú principal es servir de punto de entrada para la aplicación y permitir iniciar una partida del juego. La funcionalidad principal del menú ya venía implementada en la base utilizada, aunque se tuvo que tocar un poco para que el botón para cerrar la aplicación funcionara. Además de esto se realizaron cambios visuales para hacerlo más agradable a la vista. Estos cambios se explicarán en mayor detalle en la sección de diseño.

El menú de pausa del proyecto base consistía en una alerta que contenía botones para reanudar la partida o volver al menú principal. Esto ha sido sustituido por un fragmento con el mismo contenido. Al igual que la alerta, el fragmento no ocupa toda la pantalla, por lo que es posible seguir viendo el juego pausado al fondo.

La pantalla de puntuación tuvo que hacerse desde 0. Para ello se utilizó un fragmento que muestra la puntuación obtenida y dos botones; uno para volver a jugar y otro para regresar al menú principal. Para que la aplicación lleve al jugador a esta pantalla es necesario que el jugador pierda todas sus vidas o supere el nivel de juego y en el momento en el que sucede la partida termina, por lo que el botón para volver a jugar debe volver a inicializar el juego como hace el botón para jugar del menú principal. La pantalla de puntuación mostrará mensajes distintos en función de si se ha superado o no el nivel.

En cuanto al propio nivel, se ha cambiado la orientación del juego para que vaya en horizontal en lugar de en vertical como ocurre con el juego base. Destruir asteroides otorga ahora un número de puntos que dependerá de las vidas restantes del jugador (cuantas más vidas tenga, más puntos obtendrá) y destruir varios asteroides sin recibir un golpe otorgará un bonificador de puntuación que duplicará los puntos obtenidos hasta que el jugador pierda una vida. También se han introducido múltiples cambios visuales que se explicarán en la sección de diseño, así como información para el jugador que también se explicará en más detalle en una sección futura.

## Controles

El juego debía poder controlarse mediante un joystick para poder moverse y un botón que permitiera disparar. Esto venía implementado en el proyecto base, aunque la funcionalidad del botón de disparo se ha cambiado de acuerdo con un futuro requisito para poder alternar entre modos de disparo.

## Enemigos

El nivel habría de tener al menos un tipo de enemigo con el que el jugador pueda chocar, perdiendo vidas.

Los enemigos en cuestión son asteroides. La lógica detrás de estos enemigos ya venía implementada en el proyecto base, pero se ha modificado para que además de restar una vida, las colisiones activen un tiempo de invulnerabilidad para el jugador para darle tiempo a recuperar la concentración tras el impacto. Como se ha dicho antes, destruir estos asteroides antes de que choquen con el jugador o pasen de largo otorgará puntos en función de las vidas restantes del jugador y de si está o no en racha. También, se han modificado la cantidad de asteroides que pueden aparecer así como el punto de aparición y desaparición por los márgenes de la pantalla, adaptando así el número de asteroides para que estos sean acordes a la jugabilidad con el disparo constante.

## Información en pantalla durante el juego

Se requería que la aplicación mostrara información actualizada del estado del juego por pantalla de manera que el jugador sea consciente en todo momento de las vidas que le quedan o de la puntuación que tiene.

El jugador comienza con 4 vidas. En lugar de mostrar un número o varias imágenes que lo indiquen se ha optado por utilizar una imagen de la nave. A medida que el jugador pierde vidas algunas partes de la nave se vuelven rojas, indicando que han sido dañadas.

La puntuación se muestra a un lado de la pantalla y se actualiza de manera automática cada vez que se destruye un asteroide. También se muestra el indicador de bonificación por racha cuando está activo.

Por último, en el centro de la pantalla se muestra el tiempo restante que debe aguantar el jugador para completar el nivel.



*Ilustración 2: Pantalla de juego*

### Fin de partida

La partida debía poder terminar y al hacerlo tendría que mostrar un resumen de los logros obtenidos en la pantalla de puntuación.

La partida termina si el temporizador llega a 0, indicando que el jugador ha sobrevivido el tiempo suficiente o si el jugador se queda sin vidas, destruyendo su nave y perdiendo la partida. Al terminar la partida se muestra la puntuación obtenida, así como un mensaje cuyo contenido varía en función de si se ha superado o no el nivel y de si has conseguido una nueva puntuación record.





Ilustración 3: Pantalla de victoria



Ilustración 4: Pantalla de derrota

## Diseño

Se decidió seguir con la temática del proyecto base (temática espacial), aunque se ha intentado no utilizar ningún *asset* visual de dicho proyecto. Para ello se ha creado un nuevo Sprite para la nave que *controlará* el jugador, así como para los disparos de esta. El fondo del nivel y del menú principal se ha hecho con una herramienta externa diseñada para crear fondos de esta temática divididos por capas para poder crear un efecto de paralaje. Estas herramientas

pueden ser encontradas en el apartado de referencias. Para la creación del paralaje, estos fondos tuvieron que ser *tileados* por nosotros, así como la creación de las capas de velocidad. Se añadió una fuente personalizada, así como un estilo unificado a todo el juego. Nuestro objetivo fue personalizar y hacer nuestro todos los apartados visuales del mismo.



Ilustración 5: Menú de pausa dentro del juego

### Disparo múltiple

Se pedía hacer que la nave disparara de manera automática y cambiar la funcionalidad del botón de disparo para que en lugar de disparar al pulsarse cambiara el modo de disparo.

Con unos ajustes a la variable que indica si el jugador está disparando y a los métodos asociados al botón de disparo se hizo que la nave disparara de manera continua sin necesidad de interacción del jugador.

Como disparo alternativo se ha hecho que la nave dispare dos proyectiles en distintas direcciones. Es un método más eficiente de conseguir puntos, pero deja el frente de la nave desprotegido, por lo que conviene ir alternando según la situación y que así el jugador tenga la capacidad de elegir su forma de jugar.

### Paralaje

Para la creación del paralaje, se rompió en capas el fondo del juego y se *tileó* de forma que este pudiera ser *loopeable*. De esta forma, se crearon dos capas de tipo *ImageView* por cada nivel del paralaje incrustada como *background*. De esta forma, se creó una animación con las dos capas iguales, de forma que cuando la segunda capa (que, en definitiva, es la misma que la primera) llegaba a la posición máxima de pantalla, la animación se reiniciaba comenzando de nuevo y dando un efecto de continuidad perpetuo. De estas capas se ajustó la velocidad de la

animación para dar efecto de profundidad más una primera capa con un movimiento muy rápido para dar la sensación de velocidad.

### 3.- Conclusiones

Esta práctica nos ha resultado más interesante que las anteriores y lamentamos no haber tenido tiempo de realizar todos los desafíos. Durante el desarrollo hemos encontrado algún que otro problema (el paso de la puntuación desde el juego a la propia pantalla de puntuaciones, por ejemplo, o el tener que arreglar las *hitboxes* de los nuevos *assets* que añadíamos), pero hemos sabido salir adelante. Con todo, estamos satisfechos y creemos que hemos hecho un juego que demuestra cierto nivel de competencia por nuestra parte. El juego es lo bastante entretenido como para echar un par de partidas en lo que quieres matar el tiempo (con la duración de un nivel aproximada a lo que tarda una parada de metro) y tiene potencial para ir mejorándose en el futuro, fuera del contexto de la asignatura.

### 4.- Referencias

<https://deep-fold.itch.io/pixel-planet-generator>

<https://deep-fold.itch.io/space-background-generator>

<https://www.dafont.com/es/space-age.font>