

PRACTICA T3

DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MOVIL

Alberto Blanco Barrios

Jesús Tellez Serrano

Tabla de contenido

Introducción	3
Requisitos completados	3
Requisitos mínimos	3
Desafíos plata	4
Desafíos Oro	4
Aspectos importantes de los desafíos	5
R.1.3 Nivel de juego.....	5
R.3 Enemigos y R.10. Múltiples enemigos.	5
R.4. Interfaz durante el juego.....	5
R.7. Disparo múltiple	5
Pantallas de la aplicación	6
Conclusión	7
Referencias	8

Introducción

En este documento se explicará el desarrollo de una aplicación para móviles en Android Studio API 28 y con compatibilidad hasta la API22. La aplicación es un juego de naves espaciales tipo “shoot'em up” en 2D. Esta aplicación se ha desarrollado sobre el motor de videojuegos realizado en clase.

En el videojuego controlas una nave que avanza en el espacio, esta nave dispara automáticamente balas verticales y si se presiona en el lado derecho de la pantalla disparará balas a 45º y 135º respecto al eje de abscisas. Al desplazarse la nave encontrará naves enemigas que avanzan en sentido contrario, estas naves varían en dificultad para el jugador, teniendo diferentes características y aportando diferente puntuación al derrotarlas.

El jugador puede controlar la nave con la parte izquierda de la pantalla, esta se moverá en la dirección en la que se haya desplazado el dedo que se encuentre sobre la pantalla.

Para finalizar el nivel satisfactoriamente el jugador deberá conseguir una puntuación determinada antes de recibir tres impactos. En el caso de que no lo lograra, pasará a la pantalla de score con su puntuación actual.

En el apartado de Configuración, el jugador puede elegir diferentes tipos de naves con los que jugar.

Requisitos completados

Requisitos mínimos

R.1. Arquitectura básica

1. **Menú principal:** punto de entrada al juego que permite iniciar una partida y pasar a la pantalla de selección de nave.
2. **Menú pausa:** se activa a través de un botón de pausa o cuando el juego pasa a segundo plano.
3. **Un nivel:** Nivel completo en el que se puede ganar consiguiendo una determinada puntuación, o “perder” si golpean la nave del jugador tres veces, ya sea con naves enemigas o con los disparos enemigos.

R.2. Controles

En el lado izquierdo de la pantalla se encuentra un joystick para mover la nave del jugador y presionando en el lado derecho de la pantalla, la nave realizará disparos especiales.

R.3. Enemigos

Se han incluido dos tipos de enemigos con los que se puede colisionar. Se pueden destruir disparándoles, y dependiendo del tipo de enemigo aportará una puntuación distinta al jugador.

Uno de los enemigos dispara, si este alcanza a la nave del jugador con una bala, la nave del jugador será destruida. Si la nave aún cuenta con vidas extra, reaparecerá en la pantalla, en caso contrario el juego habrá acabado y proseguirá hacia la pantalla de puntuación.

R.4. Información en pantalla durante el juego

Durante la partida se puede visualizar:

1. El número de vidas en forma de corazones en la parte superior izquierda de la pantalla.
2. La puntuación que ha obtenido el jugador hasta el momento, esta se encuentra en la parte superior de la pantalla.

R.5. Fin de partida

El juego consta de una pantalla de fin de partida, a la que se llegará obteniendo una puntuación determinada (ganando la partida) o perdiendo las tres vidas de las que dispone el jugador.

Esta pantalla se compone del título del juego, un texto que indica la puntuación obtenida en la partida y un botón para volver al menú principal.

Desafíos plata

R.6. Diseño

El diseño de la aplicación se ha realizado pensando en que sea homogéneo, con el aspecto acorde a un juego de naves. Para ello se han utilizado las siguientes características:

1. Se utiliza una paleta de colores homogénea, en la que incluimos los colores: azul de los fondos del espacio, el blanco y el negro.
2. Se ha añadido una imagen de fondo en la pantalla de menú y de fin de partida del espacio.
3. Se ha añadido una fuente que su tipografía se asocia al espacio.
4. Se han creado nuevos botones, para dotar a la aplicación de un estilo propio.
5. Se ha creado un “*AlertDialog*” modificable para darle estilo a la pantalla de pausa.
6. Se ha añadido un “*Parallax Background*” al juego
7. Se han incluido diferentes “*Sprites*” para las naves enemigas.
8. Se han incluido diferentes “*Sprites*” para la elección de nave del jugador.
9. Se han añadido “*Sprite*” de corazones para marcar las vidas.
10. Se han creado unas balas de otro color para diferenciar los disparos enemigos de los de la nave del jugador.
11. Se ha añadido un Texto para indicar que has ganado la partida.

R.7. Disparo múltiple

El juego consta de dos tipos de disparo:

1. Vertical, este se produce automáticamente desde la nave.
2. En 45 y 135 grados con respecto al eje de abscisas, este se produce al realizar la acción de disparo.

R.8. Varios tipos de naves

Se ha implementado varios tipos de naves. El jugador puede seleccionarlas desde el menú principal, pulsando el botón “*config*” para acceder al fragmento de selección de nave. En esta pantalla se encuentran cuatro botones con imágenes con cada tipo de nave. Una vez hagamos click en una de ellas se seleccionará automáticamente y se dispondrá de ella durante la partida.

Desafíos Oro

R.9. Paralaje

El juego dispone de un fondo que se mueve a una velocidad diferente a la de los objetos en primer plano.

R.10. Múltiples enemigos

Se ha implementado dos tipos de enemigos que pueden destruir la nave del jugador chocando con ella:

1. **Enemigo débil:** Este consta de un Sprite específico, 1 punto de vida (un golpe para ser destruido), otorga 100 puntos al jugador y no dispone de disparo.
2. **Enemigo fuerte:** Este consta de otro Sprite, 2 puntos de vida, otorga 300 puntos al jugador y dispone de un disparo vertical que puede destruir nuestra nave si una de las balas colisiona con esta.

Aspectos importantes de los desafíos

R.1.3 Nivel de juego

Se ha modificado la estructura inicial, para que todos los *"GameObjects"* que se utilizan en el juego se creen en la clase *"GameController"* en vez de inicializarlos desde el *"Gamefragment"*, ya que nos parece más lógico que éste pueda controlar todos los *"GameObjects"* que influyen en el juego y estos no solo se añadan a la lista de *"GameObjects"*, si no que tengamos referencias a ellos por si hay que realizar alguna acción sobre estos durante el juego.

R.3 Enemigos y R.10. Múltiples enemigos.

- Se ha realizado una clase abstracta *"Enemy"* que consta de los métodos coincidentes entre los enemigos y dos clases que heredan de esta para cada uno de los tipos de enemigos (*"EnemyPro"* y *"EnemyWeak"*). Con esto conseguimos polimorfismo y en la clase *"GameController"* solo tenemos que tener una lista de *"Enemy"*.
- Se pensó en hacer lo mismo con los disparos, pero los disparos enemigos y amigos difieren más de lo que coinciden y no conseguiríamos polimorfismo ya que se usan en clases distintas (*"SpaceShipPlayer"* y *"EnemyPro"*).

R.4. Interfaz durante el juego

- Vidas: Se ha creado una clase *"Lives"* que coloca un número de corazones en el juego, según vidas tengamos. Las vidas se controlan desde la clase *"GameController"*.
- Score: Se ha creado una clase *"Score"* que coloca un texto con la puntuación en el juego. La puntuación se controla desde la clase *"GameController"*.

R.7. Disparo múltiple

Inicialmente se crearon dos clases para cada disparo, pero dado que diferían muy poco los métodos de las balas de nuestra nave, se modificó para tener solo una clase *"Bullet"*.

Pantallas de la aplicación

Menú principal



Selección de nave



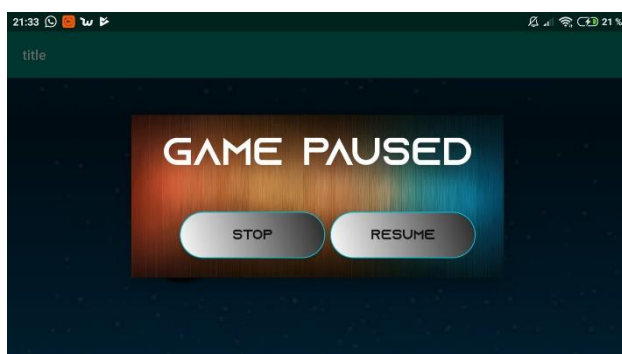
Múltiples disparos y enemigos



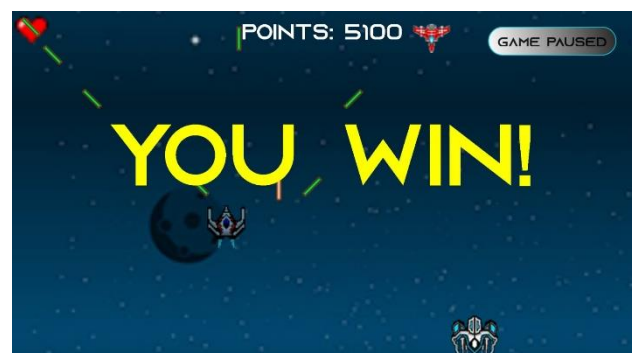
Reducción de vida, puntuación y otra nave elegida



Menú de pausa



Ganar partida



Pantalla de fin de juego



Conclusión

Con el desarrollo de esta práctica hemos aprendido como realizar un pequeño motor para implementar juegos en dos dimensiones en Android Studio. Seguimos siendo partidarios de utilizar motores ya establecidos en el mercado y con años de trayectoria, como serían Unity3D o Unreal Engine 4, ya que traen todas las funcionalidades básicas de un videojuego ya implementadas y ofrecen infinitas facilidades a la hora de desarrollar juegos para móviles multiplataforma.

Aun así, es bueno conocer más opciones a la hora de realizar este tipo de desarrollos, y como última alternativa poder optar por este motor de videojuegos que, aunque es limitado, nos ofrece la posibilidad de expandirlo y desarrollar nuevos módulos e, incluso, introducir elementos 3D si utilizamos OpenGL.

Referencias

Libro indicado en clase: *Mastering Android Game Development* de Raúl Portales

Parallax Background <https://www.youtube.com/watch?v=JPJ1doUobGY>

Secondary Background https://www.youtube.com/watch?v=AY3P_P0esFg

Font: <https://www.fontspace.com/mixofx/azonix>

Pause Background: https://pngtree.com/freebackground/metallic-color-gradient-background_814519.html

Corazón: <https://www.deviantart.com/jembvby>

Naves enemigas:

<http://pixelartmaker.com/art/30b3fcd44c7701a>

<http://pixelartmaker.com/art/7dcea30e81a6f56>

Naves Jugador:

<http://www.mattkeeter.com/projects/pixelsim/>

<http://pixelartmaker.com/art/d38ed748d6e559a>

<http://pixelartmaker.com/art/b855f115133f3bc>

<http://pixelartmaker.com/art/6a7a884cc5870b0>