

Diving Dolphin



Descripción del proyecto y ámbito de implantación

He desarrollado una aplicación móvil, la cual es un videojuego arcade en dos dimensiones, que consiste en controlar a un personaje y tener atrapar o esquivar unos elementos que se mueven por la pantalla.

Esta aplicación ha sido desarrollada para todo tipo de públicos, cualquier persona es apta para usar un videojuego, ya sea para pasar el rato o amenizar el tiempo.

Estamos en una época en la que casi todo el mundo, por no decir todo el mundo tiene al menos un videojuego en su dispositivo móvil, ya sean videojuegos más casuales como el Candy Crush, el Clash Royale, Flappy Bird, o videojuegos para públicos más específicos.

Para el desarrollo de la aplicación se ha utilizado el IDE Android Studio, y el lenguaje de programación Java.

Para el control de versiones se ha utilizado GitHub y la aplicación sourcetree.

De momento no se prevén cambios en el futuro, ya que en este tipo de aplicaciones se suele tener en cuenta el feedback de la comunidad de usuarios, en un principio sin ses feedback se podría actualizar y añadir más niveles, personajes, dificultades o modos de juego, por ejemplo.

Recursos de hardware & software

Para utilizar esta aplicación, es necesario un dispositivo Android, real o emulado, y una versión de Android 12.0 o superior, actualmente la aplicación tiene un peso de 48MB, por lo que sería ideal tener más de 50MB de espacio libre en nuestro dispositivo.

Para que la aplicación funcione fluidamente lo recomendado es tener 1GB de RAM en el móvil y a ser posible jugar en pantallas de entre 480 o 1080 px de ancho.

El entorno de desarrollo ha sido Android Studio, y para el entorno de producción se ha utilizado un dispositivo Xiaomi PocoPhone F3, y los emuladores de un Pixel 4, de u QVGA de 2.7 pulgadas, de un Nexus 4 y de un Fold-out de 8 pulgadas.

Temporalización del proyecto y fases de desarrollo

Lo primero fue definir el proyecto, de entre varias ideas me decante por hacer un videojuego para móviles por que me pareció lo mas entretenido y con muchos elementos estudiados durante el curso con los cuales podría demostrar el conocimiento adquirido, acto seguido empezó el planteamiento de como sería el videojuego, me decante por un juego en 2D, por que siendo un equipo de 1 persona, sin experiencia en diseño gráfico ni en animación en 3D, era la opción mas adecuada, una vez terminado el planteamiento me via con **3 layouts**, uno para el inicio del juego, uno para la pantalla final del Game Over y con otro que seria el gameplay (el juego en si), después utilice una **IA**, para generar los diseños e imágenes de fondo de pantalla, ya que no tengo conocimientos en diseño gráfico y no quería usar imágenes de internet posiblemente ya usadas en muchos otros proyectos, esta IA, me genero todos las imagenes de los enemigos, el delfín, los peces, los fondos de pantalla y los iconos de la app. Todo esto después lo tuve optimizar y re dimensionar para que cupiesen en la pantalla del dispositivo móvil.

Una vez obtenidas todas las imágenes del juego, las implemente y empecé con el **backend**, de la aplicación, ahí es donde empecé a crear la clase Delfín con todas la funcionalidades que le darían la vida al juego, las cuales serán explicadas en el siguiente punto del documento.

Una vez terminado me dedique a mejorar el aspecto visible de la aplicación, (colores y tamaños de los TextView, posiciones, etc.)

El gran problema vino, al darme de cuenta de que según el tamaño del dispositivo todo colapsaba y se solapaba, una imagen encima de otra, la puntuación entrecortada, los puntos de vida no aparecían en la pantalla, todo esto supuso un ultimo gran esfuerzo ya que no era tan sencillo como cambiar la medida de dp a sp en un TextView, casi todo lo utilizado en la app estaba hecho con **Bitmaps**, y **Paints**, lo cual supuso implementar métodos para que cambiases de tamaño según el tamaño de la pantalla del dispositivo, hecho eso, también utilizando de referencia el tamaño de la pantalla del dispositivo se ajustaron las velocidades de los elementos móviles por pantalla, ya que no es lo mismo atrapar un pez en la pantalla de una tablet, que en la pantalla de un móvil pequeño.

Las cosas que aprendí y mas me gustaron fue a añadir audio o video con MediaPlayer y la utilización de los Bitmap, tambien como cambiando los valores de ciertas variables como las velocidades, puedes literalmente subirle la dificultad al juego o adaptarlo a todo tipo de pantallas, lo aprendi mediante:

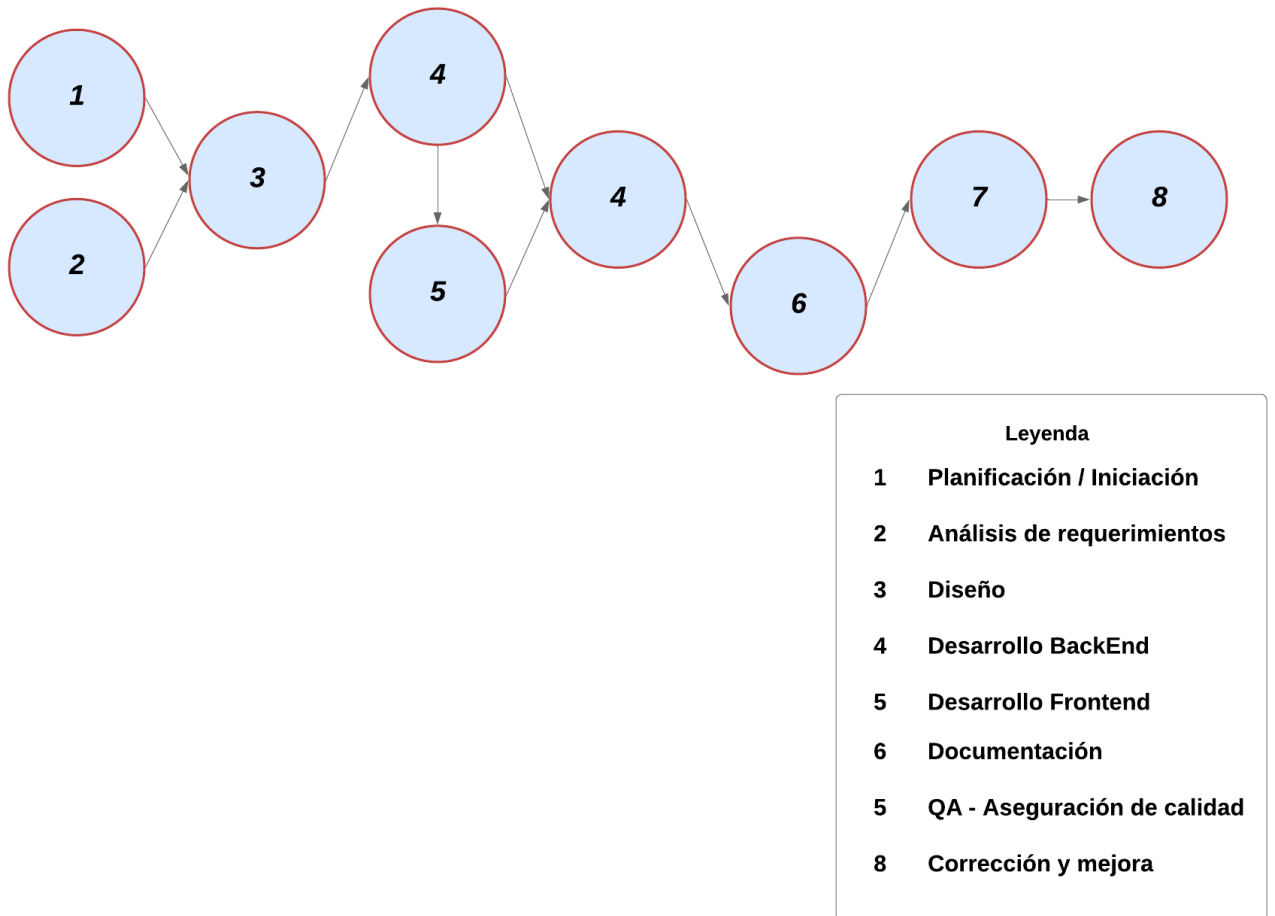
getResources().getDisplayMetrics().widthPixels; y me parecio my interesante para proyectos futuros.

DIAGRAMA DE GANTT

Diving Dolphin

TAREAS	20 - 28 Octubre	1 - 4 Noviembre	5 - 8 Noviembre	9 - 12 Noviembre	13- 20 Noviembre	1 - 7 Diciembre	16 -18 Diciembre
Planificación Iniciación							
Análisis de requerimientos							
Diseño							
Desarrollo Backend							
Desarrollo Frontend							
Documentación							
QA - Aseguración de calidad							
Corrección y mejora							

Diagrama de PERT



1 y 2 del 20 al 28 de Octubre y del 1 al 4 de Noviembre

3 del 1 al 4 de Noviembre

4 del 5 al 8 de Noviembre, del 9 al 12 de Noviembre y del 13 al 20 de Noviembre

5 del 5 al 8 de Noviembre y del 12 al 20 de Noviembre

6 del 1 al 7 de Diciembre

7 del 12 al 20 de Diciembre y del 1 al 7 de Diciembre

8 del 16 al 18 de Diciembre

Descripción de datos

La aplicación consta de las siguientes partes:

- Menú de inicio:
 - . XML → Es lo primero que observamos al entrar en la aplicación, es un menú con un **ImageView** del delfín protagonista, un **Background** ambiental, y un **TextView** con el nombre de la app.
 - . JAVA → Aquí podemos encontrarnos con ciertos procesos clave de la aplicación, como por ejemplo el **MediaPlayer**, que es una API de android que nos permite reproducir la música en el menú, mediante diferentes metodos (**onDestroy()**, **onPause()**, **onCreate()**) controlamos la música.
También podemos encontrarnos con un **Thread**, que a los 5 segundos nos manda directamente a la MainActivity, donde comenzara la partida.
- Clase Delfín:
 - Es la clase mas importante de la aplicación, todo lo que sucede aquí es el funcionamiento del videojuego y esto lo veremos plasmado en la MainActivity, nos encontramos con datos como:
 - Bitmaps**, para el delfín, los peces, los enemigos y el background, mediante un Bitmap es mas cómodo poder cambiar rápidamente la imagen por otra, ya sea por otro fondo o otro enemigo al pasar de nivel.
 - Paint**, se utiliza para el marcador de la puntuación, es un contador que cambia continuamente.
 - Int**, se utilizan en muchos procesos, los mas importantes son, todas las cordenadas y velocidades de los elementos móviles, la altura y la anchura de los niveles o la puntuación.
 - Boolean**, solo hay uno, y nos permite conocer si la pantalla esta teniendo un click o no.
 - Los métodos mas importantes que encontramos en esta clase son **"public boolean comerPez(int x, int y)"**, que nos permite saber cuando hemos interactuado con algun enemigo o pez, comparando las cordenadas del delfín y las del otro objeto, si coinciden la X y la Y, significa que están chocándose y por eso el delfín se estaría comiendo un pez.

O el evento “***boolean onTouchEvent(MotionEvent event)***”, el cual hace que el delfín suba por cada click que reciba la pantalla, recibiendo un evento cada vez que click = true.

- MainActivity:

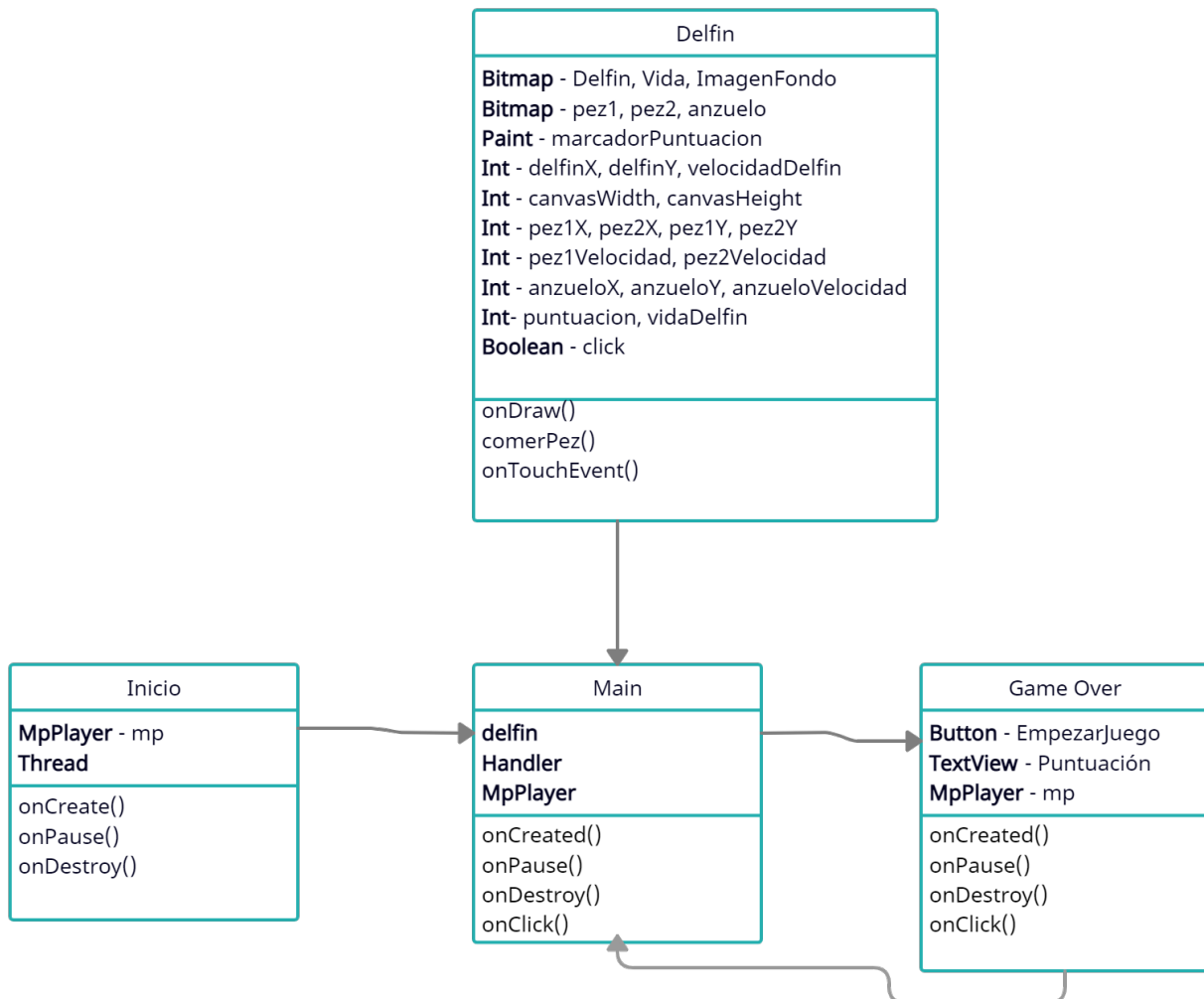
Aquí se instanciará todo lo anterior visto en la clase Delfín, añadimos la gravedad al Bitmap del delfín, mediante un **Timer** y un **Handler**, que nos permite poner en cola acciones que se ejecutaran en un thread de fondo, tiene también un **MediaPlayer**.

- Menú Game Over:

- . XML → Este menú se mostrara el perder y terminar la partida, físicamente consta de 3 **TextView**, uno de ellos nos mostrara la puntuación final, tiene también un **Background**, un **ImageView**, y un **Boton**, que nos permitirá jugar de nuevo otra partida, llevándonos otra vez al MainActivity.
- . JAVA → Lo mas importante que podemos ver aquí es el método **OnClick**, que asociado junto el botón en el .xml, nos permite volver a jugar de nuevo. También nos encontramos con un MediaPlayer, y un **TextView**, que mediante un **getIntent**, recibe la puntuación obtenida en el MainActivity y la muestra por pantalla en el Game Over Activity.

Arquitectura de software

Para esta aplicación de utilizo el patrón de diseño: Modelo, Vista, Controlador.



La aplicación comienza siempre en el Activity Inicio, que a los 5 segundos da paso automáticamente al Main Activity, en el cual tenemos una instancia de Delfin, en Delfin es donde está construido todo el videojuego, simplemente es un personaje (delfin) en una posición X estática, y solo se puede mover en el eje Y, y los otros personajes (peces y anzuelos), al contrario, solo se puede mover en un eje X, apareciendo aleatoriamente en un punto del eje Y. Lo que tenemos que hacer es que los 2 ejes, X e Y, del delfín coincidan con los 2 ejes, X e Y de los peces, y así puntuar, así mismo tenemos que evitar que eso pase con los anzuelos, si no se nos descontará una vida del contador, en total 3. Al perder las 3 vidas, la partida termina, y la Main Activity da paso a la Game Over Activity.