

PROGRAMACIÓN DE SISTEMAS 22/23 Q2



**FloatAR**

**Autores:**

Antonio Vila Leis - antonio.vila@udc.es

Hugo Sanjiao Varela - h.sanjiao@udc.es

Breogán Fernández Moreira - breogan.fernandez@udc.es

Óscar Alejandro Manteiga Seoane - oscar.manteiga@udc.es

**Fecha:** *A Coruña, 16 Marzo 2023*

# Índice

Capítulos	Página
<b>1. Introducción</b>	<b>2</b>
1.1. Objetivos . . . . .	2
1.2. Motivación . . . . .	2
1.3. Trabajo relacionado . . . . .	2
<b>2. Análisis de requisitos</b>	<b>3</b>
2.1. Funcionalidades . . . . .	3
2.2. Prioridades . . . . .	4
<b>3. Planificación inicial</b>	<b>4</b>
3.1. Iteraciones . . . . .	4
3.2. Responsabilidades . . . . .	5
3.3. Hitos . . . . .	5
3.4. Incidencias . . . . .	6
<b>4. Diseño</b>	<b>6</b>
4.1. Arquitectura . . . . .	6
4.2. Persistencia . . . . .	7
4.3. Vista . . . . .	7
4.4. Comunicaciones . . . . .	8
4.5. Sensores . . . . .	8
4.6. Trabajo en background . . . . .	8
<b>Bibliografía</b>	<b>8</b>

Cuadro 1: Tabla de versiones.

Versión	Fecha	Autor
1	16 de marzo de 2023	Antonio — Hugo — Breogán — Óscar
2	18 de abril de 2023	Antonio — Hugo — Breogán — Óscar

# 1. Introducción

La aplicación FloatAR es una aplicación de realidad aumentada del juego "Hundir la flota" para dispositivos Android. Esta aplicación utiliza la API ARCore y se desarrollará para un trabajo de clase de la asignatura Programación de Sistemas en un grupo de 4 personas. El juego podrá ser utilizado de forma estándar o con el uso del modo de realidad aumentada.

## 1.1. Objetivos

El objetivo principal de FloatAR es proporcionar una experiencia de juego más inmersiva para los usuarios utilizando la tecnología de realidad aumentada. Esta tecnología no está muy expandida en las aplicaciones de Android, pero aun menos en los videojuegos, por lo que otro objetivo implícito podría ser la investigación de esta tecnología. Los objetivos secundarios incluyen:

- Desarrollar de forma convencional el juego de "Hundir la flota".<sup>en</sup> modo local.
- Implementar el modo para 2 jugadores.
- Integrar un sistema de puntuación y registro de puntuación para los usuarios.
- Añadir el espacio de juego al mundo real con la realidad aumentada utilizando la API ARCore [1].
- Implementar una detección precisa de la ubicación de los objetos en donde poder utilizar el juego utilizando la API ARCore [1].

## 1.2. Motivación

La tecnología de realidad aumentada ha ganado popularidad en los últimos años y se está utilizando cada vez más en juegos y aplicaciones. El uso de la realidad aumentada en juegos como hundir la flota puede proporcionar una experiencia de juego más inmersiva y emocionante para los usuarios. A mayores, la investigación e implementación de esta tecnología en juegos es un apartado que nos produce gran interés, ya que no está muy extendida y creemos que será muy usada en el futuro. La aplicación FloatAR también puede ser aplicable en otros juegos que requieren que ambos jugadores (o más de 2) no puedan ver lo que ven los demás. Esto hace que en un futuro próximo podamos expandir el juego si es necesario.

## 1.3. Trabajo relacionado

Hay otras aplicaciones de realidad aumentada para juegos, como Pokémon Go (Niantic) [2], la aplicación más exitosa que implementa realidad aumentada (y una de las más exitosas a nivel de aplicación general) que utilizan esta tecnología para crear una experiencia de juego más inmersiva. Otros ejemplos son Angry Birds AR:

Isle of Pigs (Rovio Entertainment) [3], Jurassic World Alive (Ludia Inc.) [4], Harry Potter: Wizards Unite (Niantic) [5], Minecraft Earth (Mojang) [6]... Todas ellas implementan la RA, pero de formas muy distintas:

- Pokémon GO, Jurassic World Alive y Harry Potter: Wizards Unite la usan para mostrar sus Pokémons, dinosaurios o entidades fantásticas.
- Minecraft Earth la usa para mostrar su mundo en nuestra habitaciones o espacios de la vida real a través de la cámara.
- Angry Birds AR: Isle of Pigs y nuestra aplicación FloatAR la usan para mostrar el espacio de juego en la vida real para ver como se desarrolla el evento.

## **2. Análisis de requisitos**

Antes de comenzar el desarrollo de la aplicación FloatAR, se debe realizar un análisis preliminar de los requisitos. Esto ayudará a definir las funcionalidades clave de la aplicación y establecer prioridades.

- Jugar de forma local contra la IA al juego sin realidad aumentada.
- Jugar contra otro jugador sin realidad aumentada.
- Interfaz de usuario intuitiva y simple con sistema de puntuación.
- Modo de realidad aumentada local.
- Modo de realidad aumentada en multijugador.

### **2.1. Funcionalidades**

1. Tablero de juego sin ser en realidad aumentada.
2. Colocación de barcos en el tablero con sistema de colisiones.
3. Menú de ajustes, de información y de ayuda.
4. Juego local contra la IA, con turnos, detección de victoria/derrota...
5. Juego online contra otro jugador con las mismas características.
6. Posibilidad de jugar a los 2 anteriores modos en realidad aumentada.

## 2.2. Prioridades

1. Crear un tablero sin ser en realidad aumentada (Core).
  - a) Permitir al usuario colocar sus barcos en el tablero virtual (Core).
  - b) Implementar un sistema de detección de colisiones para asegurar que los barcos no se superpongan (Core).
2. Crear pantalla de ajustes, ayuda e información sobre la aplicación (Core).
3. Implementar sistema de juego local (Core).
  - a) Implementar un sistema de juego por turnos contra una IA (Core).
  - b) Implementar un sistema de detección de victoria y derrotas (Core).
4. Implementar sistema de juego online (Core).
  - a) Implementar el sistema de comunicación entre jugadores para sincronizar el juego y las puntuaciones/eventos (Core).
  - b) Implementar un sistema de juego por turnos contra una otro jugador (Core).
5. Desarrollar la realidad aumentada (Core).
  - a) Desarrollar la parte de realidad aumentada para jugar contra la IA (Core).
  - b) Desarrollar la parte de realidad aumentada multijugador (Core).
6. Mostrar la puntuación y el estado del juego en tiempo real (Accesorio).
7. Implementar un sistema de animación para ciertas acciones (Accesorio).

## 3. Planificación inicial

En esta sección se establecerá una planificación muy simple y básica del proyecto, definiendo las iteraciones, responsabilidades, hitos y entregables.

### 3.1. Iteraciones

Consideramos 12 semanas para la implementación de las funcionalidades del juego:

- Inicio del proyecto, organización, reparto de trabajo y funcionalidades 1, 2 y 3 (base de aplicación y juego). [4 semanas].
- Funcionalidades 4 y 5 (sistema de juego offline/online) [4 semanas].
- Funcionalidad 6 (realidad aumentada) [4 semanas].

### 3.2. Responsabilidades

- Óscar Alejandro Manteiga Seoane:
  1. Desarrollo de la base de la aplicación (pantallas, ajustes...).
  2. Tablero básico de juego.
  3. Prueba y estudio de ARCore para implementarla una vez llegada su implementación.
- Antonio Vila Leis:
  1. Desarrollo del sistema de colocación de barcos en el tablero.
  2. Desarrollo del sistema de juego contra la IA.
  3. Prueba y estudio de ARCore para implementarla una vez llegada su implementación.
- Hugo Sanjjiao Varela:
  1. Desarrollo de la base de la aplicación (pantallas, ajustes...).
  2. Desarrollo del sistema de juego multijugador.
  3. Redacción de la documentación necesaria.
- Breogán Fernández Moreira:
  1. Desarrollo del sistema de juego contra la IA.
  2. Desarrollo del sistema de juego multijugador.
  3. Desarrollo de los test.

Es importante mencionar que la implementación de la realidad aumentada la intentaremos entre todos ya que ninguno tiene experiencia desarrollándola. Serán Óscar y Antonio los encargados de su estudio a lo largo del desarrollo para poder partir de una base de conocimiento sobre la API ARCore. Las responsabilidades de cada uno pueden variar a lo largo del proyecto.

### 3.3. Hitos

1. Entrega de la primera versión de la memoria, con la planificación inicial y requisitos/funcionalidades.
2. Entrega de la base de la aplicación y del juego.
3. Entrega del juego completo sin realidad aumentada.
4. Entrega del juego con la realidad aumentada.

### 3.4. Incidencias

Posibles incidencias y planes de contingencia: En caso de problemas genéricos durante el desarrollo, se realizarán pruebas (tests) durante la implementación para intentar corregirlos antes de cada entrega. En caso de encontrarlos, se buscarán alternativas y soluciones en conjunto con todo el equipo. El desarrollo del juego básico lo sabemos hacer, por lo que los principales inconvenientes se pueden generar con la implementación de la realidad aumentada, la sincronización online de jugadores y la sincronización de la realidad aumentada.

En el caso de la implementación de la realidad aumentada en el proyecto, si no conseguimos un correcto funcionamiento de esta tecnología, se pasará a desarrollar una aplicación de juegos de mesa online. Los juegos que incorporemos serán varios y distintos entre si para incrementar la complejidad del proyecto.

En caso de poder implementar la realidad aumentada pero sin que sea de forma sincronizada por internet, se creará el juego de hundir la flota de forma local para visualizar tu parte del tablero en esta tecnología mientras que de fondo y para sincronizar el online se juega con el modo básico sin RA.

## 4. Diseño

El diseño del proyecto es la propuesta inicial y aún está sujeto a cambios y mejoras en función de los conocimientos adquiridos a lo largo del desarrollo. Para el diseño de la aplicación se han utilizado varios bloques funcionales básicos, y se ha tratado de hacer uso de elementos no explicados en el temario para demostrar la capacidad de investigación y ampliación de conocimientos, siendo la realidad aumentada el buque insignia del desarrollo.

### 4.1. Arquitectura

La arquitectura de la aplicación se basará en la utilización de diferentes componentes que trabajarán en conjunto para ofrecer una experiencia completa al usuario. Estos componentes serán los siguientes:

- Actividades: la aplicación contará con varias actividades que permitirán al usuario acceder a diferentes secciones de la aplicación. Por ahora tenemos las siguientes:
  1. Activity de inicio: se ejecutará cuando se inicie la aplicación y mostrará la pantalla de inicio con opciones para iniciar un nuevo juego, cargar un juego guardado (sin confirmar esta funcionalidad) o ver las opciones de configuración.
  2. Activity de creación del tablero: mostrará una pantalla donde el usuario podrá colocar sus barcos sobre su tablero, en esta actividad también se crea el tablero del rival para el modo de un jugador.

3. Activity de juego "singleplayer": mostrará el tablero del jugador y permitirá jugar contra la computadora.
  4. Activity de juego "multiplayer": mostrará una lista de las puntuaciones más altas de los jugadores.
  5. Activity de configuración: permitirá a los usuarios personalizar la configuración de la aplicación, como cambiar el idioma, el nivel de dificultad (sin confirmar esta funcionalidad), la configuración de sonido, etc.
  6. Activity de ayuda: mostrará información de ayuda y tutorial para los nuevos usuarios.
  7. Activity de información: proporcionará información de la aplicación y de los autores de la misma.
  8. Activity de inicio de sesión (sin confirmar esta funcionalidad).
  9. Activity donde se guardan todas las preferencias de la aplicación
- Servicios: se utilizarán servicios para realizar tareas en segundo plano, como la sincronización online.
  - Broadcast receivers: se utilizarán broadcast receivers para gestionar las notificaciones push que se envíen al usuario.
  - Threads: se utilizarán threads para evitar que la aplicación se bloquee durante tareas que requieran mucho tiempo de procesamiento.

## 4.2. Persistencia

La aplicación necesita almacenar y gestionar distintos tipos de datos. En concreto, se van a almacenar los datos de usuario (nombre, apellidos, correo electrónico y contraseña), así como información de las actividades realizadas por el usuario. Para ello, se va a utilizar una base de datos gestionada con Firebase y si es posible el servicio de Google Play Juegos.

## 4.3. Vista

La aplicación contará con varias actividades y fragmentos que permitirán al usuario acceder a diferentes secciones de la aplicación. Todas ellas están sujetas a cambio importante durante el desarrollo. Además, se incluirán notificaciones para mantener al usuario actualizado sobre las últimas novedades. A continuación, se describen los elementos visuales que se incluirán en la aplicación:

- Actividades: la aplicación contará con varias actividades, como la creación del tablero, la ejecución de las partidas, el menú principal y las pantallas de ayuda o configuración.
- Diálogos: Se utilizaran diálogos para informar al usuarios de diferentes eventos, por ejemplo del fin de partida.



## 4.4. Comunicaciones

Para la comunicación con el servidor, se utilizarán servicios web RESTful para acceder a los datos almacenados en la base de datos local y en la nube. Además, se explorará la posibilidad de utilizar un sistema de autenticación para garantizar la seguridad de los datos del usuario. Es importante que este apartado también es posible que sufra muchos cambios durante la implementación.

## 4.5. Sensores

Los sensores utilizados serán gestionados con la API ARCore [1], aunque no sabemos por ahora cuales son.

## 4.6. Trabajo en background

La aplicación utilizará servicios para realizar tareas en segundo plano, como la sincronización con el servidor o la descarga de modelos 3D usados en la realidad aumentada (si es necesario, aun por confirmar). Además, se utilizarán threads para evitar que la aplicación se bloquee durante tareas que requieran mucho.

# Referencias

- [1] “Arcore sdk for android,” 2021.
- [2] “Pokemon go,” 2016.
- [3] “Angry birds ar: Isle of pigs,” 2018.
- [4] “Jurassic world alive,” 2018.
- [5] “Harry potter: Wizards unite,” 2018.
- [6] “Minecraft earth,” 2018.