



UNIVERSIDADE DA CORUÑA

PROGRAMACIÓN DE SISTEMAS 24/25 Q1  
Icono de la aplicación

## Juego de air hockey con pantalla dividida

**Autores:**

José Manuel Fernández Montáns (j.m.fmontans@udc.es)

Mateo Rivela Santos (mateo.rivela@udc.es)

Hugo Mato Cancela (hugo.matoc@udc.es)

**Persona de contacto:** X

**Fecha:** *A Coruña, 10 Octubre 2024*

**Versión:** *0.1*

**Nombre de la aplicación:** *X*

# Índice

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

Cuadro 1: Tabla de versiones.

Versión	Fecha	Autor
0.1	10/10/24	

# **1. Introducción**

## **1.1. Objetivos**

El objetivo principal será trabajar con una pantalla dividida con una baja latencia y gestionar los rebotes. Como objetivos secundarios, que existan barreras, una pelota que viaja entre dispositivos y añadir un menú.

## **1.2. Motivación**

Intención de experimentar con traspaso de datos entre dispositivos e interacción de elementos de la aplicación con la pantalla del dispositivo.

## **1.3. Trabajo relacionado**

¿? rev

# **2. Análisis de requisitos**

## **2.1. Funcionalidades**

Conectar dos teléfonos, cada uno mostraría una mitad del campo completo, de forma que los elementos en los que cada usuario se tiene que fijar pueden ser de mayor tamaño.

Los mazos con los que cada uno tiene que golpear el disco se controlarían con un solo dedo, apareciendo en la parte de la pantalla donde se mantiene este.

El disco, al ser golpeado por uno de los mazos, se mueve en una trayectoria determinada por el ángulo (y, preferiblemente, fuerza) del impacto, pudiendo ser alterada al chocar con los límites del campo, que corresponderían a los bordes del dispositivo.

## **2.2. Prioridades**

Lo principal sería lograr la conexión entre dispositivos, pasar los datos de forma correcta y apropiada, que el disco pase y rebote para implementar el juego de air hockey. Primero, se implementará el traspaso de objetos entre pantallas y dispositivos con conexión mediante Bluetooth. Luego, trabajaremos con rebotes y bordes de la pantalla como paredes. Una vez realizado lo más básico, añadiremos una pantalla de menú para acceder a los ajustes y a prototipos anteriores. Por último, uniremos

las funcionalidades básicas, añadiremos bloques destructibles que funcionarán como paredes.

## **3. Planificación inicial**

### **3.1. Iteraciones**

Aquí se definen los prototipos de nuestro proyecto y las funcionalidades principales que trabajamos en cada prototipo. Se implementarán en el orden establecido:

- P1:** Pasar un objeto entre dos dispositivos conectados por Bluetooth.
- P2:** Implementar los rebotes y los bordes de la pantalla del dispositivo como paredes.
- P3:** Añadimos una maza que el usuario puede manejar y que rebota con la pelota.
- P4:** Añadir un menú con el que acceder a la pantalla de ajustes y a los prototipos anteriores.
- P5:** Unión de funcionalidades de P1 y P2: un disco que rebota y se transporta entre dos pantallas al atravesar el borde superior.
- P6:** Añadir bloques destructibles que funcionen como paredes.
- P7:** Producto final

### **3.2. Responsabilidades**

**Físicas:** Mateo

**Conexión:** José Manuel

**Visualización:** Hugo

### **3.3. Hitos**

- Establecer una conexión de baja latencia entre ambos dispositivos.
- Implementar rebotes.
- Hacer que los bordes de la pantalla funcionen como paredes.
- Añadir bloques destructibles que funcionen como paredes.
- Crear un menú con varios ajustes.

Entregables: Cada protipo será una entrega.

### **3.4. Incidencias**

No tenemos conocimientos sobre gráficos o físicas por lo que tendremos que aprender y probar todo lo relacionado a eso. Además, para paliar esas carencias, tenemos pensado usar LibGDX [1] y KryoNet [2] para la codificación general de juego. Y usaremos las librerías Box2d [3], Tween Engine [4] y BluetoothLib [5] para las físicas, las animaciones y la conexión Bluetooth respectivamente.

En la primera iteración, probaremos la conexión entre dispositivos y que el objeto se pasa correctamente entre ellos. En la segunda, en un prototipo distinto, el objeto debe rebotar contra los bordes de la pantalla de un dispositivo para que funcionen como paredes. En la tercera, implementamos la maza, que debe responder correctamente al input del usuario, y nos aseguramos que la pelota rebote correctamente contra esta. En la cuarta, el menú nos debe permitir cambiar entre varios prototipos. En la quinta, uniremos las funcionalidades de las dos primeras iteraciones y nos aseguraremos de que funcionan correctamente ambas al mismo tiempo. En la sexta, añadiremos bloques que funcionen como paredes. La pelota debe rebotar contra estos como lo hace con los bordes de la pantalla. En la última, comprobaremos que, tras haber implementado todo, funciona correctamente.

En caso de que alguien enferme repartiremos el trabajo asignado a esa persona entre los demás integrantes del grupo.

## **4. Diseño**

### **4.1. Arquitectura**

### **4.2. Persistencia**

Respecto a qué información almacenaremos, ambos jugadores siempre tendrá los datos de la posición, ángulo y fuerza de la pelota. Además, cada uno guardará donde está situada la su propia maza en caso de que esté siendo usada.

### **4.3. Vista**

En lo que respecta el diseño de la aplicación, tendremos una actividad que presentará la pantalla del usuario, un fragment para el menú y otro para el campo de juego. La pelota será un elemento del fragment que aparecerá en la pantalla únicamente cuando las coordenadas estén en nuestro campo.

## 4.4. Comunicaciones

La comunicación en este juego es una parte fundamental. Para ello buscamos un sistema de baja latencia, pero que no tendrá que transmitir grandes cantidades de datos, por lo que hemos optado por Bluetooth.

## 4.5. Sensores

## 4.6. Trabajo en background

Se enviarán los datos de la posición y ángulo de la pelota, cada X tiempo. También se comprobará que, cuando rebote, en caso de que sea contra un bloque destructible, este desaparezca.

## Rev:

- Estilo portada modificable??
- Nombre de la aplicacion
- Persona de contacto
- Autor tabla de versiones
- Trabajo relacionado
- Incidencias contenido correcto?

## 5. Bibliografía

### Referencias

- [1] libGDX Team, “libgdx: Cross-platform game development framework. libgdx official website..” Available on: <https://libgdx.com/>, 2024.
- [2] Software, E., “Kryonet: A java network library based on kryo. github repository..” Available on: <https://github.com/EsotericSoftware/kryonet>, 2014.
- [3] libGDX Team, “Box2d: A 2d physics engine for games integrated with libgdx. libgdx documentation..” Available on: <https://libgdx.com/wiki/graphics/2d/box2d>, 2024.

- [4] Ribon, A., “Universal tween engine: An open-source java tween engine. github repository..” Available on: <https://github.com/AurelienRibon/universal-tween-engine>, 2015.
- [5] Omaflak, “Bluetoolib: Bluetooth library for android. github repository.” Available on: <https://github.com/omaflak/BluetooLib>, 2021.