

BALL game

(Ball Game)

MANUAL TÉCNICO

El propósito de este manual es de explicar el código del juego.

Clases de objetos

Nota: Todas las variables de posición en los objetos son para poder alinearlos en la pantalla, el fondo del juego está controlado por el widget de flutter, Expanded, que se adapta al tamaño disponible de la pantalla. Al alinear un objeto en un Expanded, los valores mínimos y máximos son -1 y 1 correspondientemente, por lo que determinar la posición de los objetos con **doubles** permite ponerlos en posiciones exactas y el moverlos, más suave.

Bola

Atributos:

double posicionX: Posición de la bola en el eje X

double posicionY: Posición de la bola en el eje Y

Plataforma

Atributos:

double posicionX: Posición de la plataforma en el eje X

Botón

Atributos:

IconData icon: Icono mostrado en el botón

function: Lineas de código que se ejecutarán al pulsar el botón

double height: Altura del botón

double width: Anchura del botón

Color color: Color del botón

Clases de pantallas

Al ejecutar el programa la primera pantalla en mostrarse tendrá un Drawer arriba a la derecha para navegar por las pantallas, main se encarga del Drawer y como primera opción se mostrará la pantalla HomePage.

main

Variables:

int _selectedIndex: Cada número está asignado a una pantalla del programa, el número almacenado en esta variable será la pantalla que se muestre al seleccionarla en el Drawer (ej: HomePage es 0).

Funciones:

void _onItemTapped(**int** index): Se ejecuta al seleccionar una de las opciones del Drawer, obtiene el número de la pantalla pasado por index y se almacena en _selectedIndex.

Juego

Variables:

double plataformaPosX: Posición inicial de la plataforma en el eje X.

double bolaPosX: Posición inicial de la bola en el eje X.

double bolaPosY: Posición inicial de la bola en el eje Y.

double bolaVelocidadX: Velocidad horizontal de la bola.

final double bolaVelocidadY: Velocidad vertical de la bola.

var direccionBola: Define la dirección en la que se mueve la bola (valores solo pueden ser “izquierda” o “derecha”).

int puntos: La cantidad de veces que la bola ha tocado la plataforma en la partida actual.

int segundos: Tiempo en segundos descendente hasta que termine el juego.

int mejorPuntuacion: Mayor puntuación recibida entre partidas.

Funciones:

void moverPlataforma[1-5]: Son 5 funciones que cambian el valor de plataformaPosX para mover la plataforma a esa coordenada cuando se pulsa esa zona de la pantalla.

void jugar **async**: Empieza el juego, aquí dentro es donde se obtiene y guarda el valor de la puntuación máxima de otras partidas y la crea si no existe, también inicia unas nuevas variables que definirán el movimiento y bote de la bola, como gravedad, tiempo y altura además de cambiar el valor de direccionBola y si bolaPosX aumenta o disminuye cada vez que toque el borde de la pantalla. En la función también se controla que cuando segundos llegue a cero el juego se detenga y muestre una pantalla de Game Over proporcionando botones de Reintentar y Volver al Menú.

double alturaAPosicion(**double** altura): Una ecuación matemática que transforma un double que representaría la altura de la bola a coordenadas verticales para actualizar la altura de la bola.

Justificación de la existencia del juego

El motivo por el que quise hacer un juego con persistencia de datos respecto a una app usando una API es sencillo, con el juego me enseñaría a mi mismo flutter en vez de usar una guía de una API sobre como usarla, podría aprender los conceptos esenciales y además supuse que el código sería más parecido a hacer un programa en Java (donde soy fuerte) al tener que programar orientado a objetos, haciéndomelo más fácil (no lo fue).

Bibliografía

Documentación de Flutter (lenguaje usado): <https://docs.flutter.dev/>

Shared Preferences (para la persistencia de datos): https://pub.dev/packages/shared_preferences

Bubble Trouble (inspiración): https://play.google.com/store/apps/details?id=com.kresogames.bubbletrouble&hl=en_US&pli=1