Documentation Flutter app

I. INTRODUCTION

A) Concept du projet

L'idée de base de mon projet était de créer un mini jeu vidéo 2D comme Mario Bross mais dans l'univers de One piece, du nom de « LAST END ». Le résultat de mon application est donc un mix des deux.

Une partie du code du jeu vidéo est un tuto que j'ai suivi sur youtube du youtuber : Mitch Koko

https://www.youtube.com/watch?v=Q0RTaOkFxWM

Mon application est constituée de quatre pages :

- main -> appelle la page Home qui sert au lancement de l'application
- Home -> contient toute la page principale, le jeu vidéo entre autre
- Boutik -> contient d'autres personnages qu'on peut achter
- Luffy -> gestion de l'image du personnage joueur

Mon application est constituée de deux pages principales : la page Home et la page Boutique.

La page Home commence par le nom du jeu en police pixélisé, ensuite une fenêtre de jeu suivit de trois bouton (à gauche, sauter, à droite) pour permettre à l'utilisateur de faire bouger le personnage du jeu vidéo. Puis en bas de page un bouton de navigation pour aller dans la boutique.

La page Boutique est faite de la même façon que la page Home. Avec le titre boutique en haut, suivit de deux avis de recherches des certains membres de l'équipage du chapeau de paille. Et en-dessous de chaque avis un bouton recruter pour acheter le personnage.

Et comme pour la Home page un bouton de navigation pour retourner au jeu.

II. EXPLICATION DU CODE

A) Main

```dart
//Fonction qui exécute le code
Run | Debug | Profile
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'LAST END',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ), // ThemeData
      //Appelle de la class HomePage du fichier home.dart
      home: const HomePage(),
    ); // MaterialApp
```

B) Home

La class home est en statefulwidget car elle aura des changement d'état comme avec les bouton par exemple.

Déclaration des fonctions pour que le personnage saute et avance.

```dart
//réinitialise les variables qui font sauter le personnage
void prejump() {
  setState(() {
    direction = "ok";//image perso
    time = 0;
    initialHeight = luffyY;
  });
}

void jump() {
  prejump();
  Timer.periodic(const Duration(milliseconds: 40), (timer) {
    time += 0.05;
    //permet de faire un saut plus naturel
    height = -4.9 * time * time + 5 * time;

    if (initialHeight - height > 1) {
      setState(() {
        luffyY = 1;
        //annule le timer sinon saut de + en + vite
        timer.cancel();
      });
    } else {
      setState(() {
        luffyY = initialHeight - height;
```

```dart
//move perso vers la droite
void moveRight() {
  direction = "right";//image perso
  Timer.periodic(const Duration(milliseconds: 50), (timer) {
    //permet de faire avancé le perso en maintenant le button
    //et de ne pas le faire dépassé du cadre de jeu
    if (userIsHoldingButton() == true && (luffyX + 0.08) < 1) {
      setState(() {
        luffyX += 0.08;//perso avance vers la droite
        midrun = !midrun;//hilusion que le perso marche
      });
    } else {
      timer.cancel();
    }
```

De même pour aller à gauche.

Body app :

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      //container principal
      width: double.infinity, //taille container
      height: double.infinity, //taille container
      decoration: const BoxDecoration(
        image: DecorationImage(
            //fond de l'app
            image: AssetImage("assets/images/bgd.png"),
            fit: BoxFit.cover), // DecorationImage
      ), // BoxDecoration
      child: SingleChildScrollView(
        child: Column(
          //colone qui va aider pour l'affichage
          //column va center auto les container suivant
          children: [
```

```
Container(
  //container du titre du jeu
  //pour décoler le titre du top
  margin: const EdgeInsets.fromLTRB(0, 20, 0, 0),
  width: 240, //taille container
  height: 70, //taille container
  decoration: BoxDecoration(
    color: const Color(0xff000000), //color fond
    //angle arrondi
    borderRadius: BorderRadius.circular(10),
    boxShadow: [
      BoxShadow(
        color: Colors.white.withOpacity(0.5),
        spreadRadius: 5, //taille/épaiseur boxshadow
        blurRadius: 7, //possition boxshadow
        // changes position of shadow
        offset: const Offset(0, 3),
      ), // BoxShadow
    ],
  ), // BoxDecoration
  child: Center(
    //centre le text dans le container hauteur
    child: Text(
      "LAST END",
      //permet d'avoir des polices spécial
      style: GoogleFonts.pressStart2p(
          //style text
          textStyle: Theme.of(context).textTheme.headline4,
          fontSize: 22, //taille text
          color: const Color(0xfff1f1f1)), //color text
      //centre le text dans le container longueur
      textAlign: TextAlign.center,
```

Container pour le mini jeu

```
Expanded(
  //container ciel
  //prend 4/5 place du container
  flex: 4,
  child: Container(
    //container bleu
    decoration: const BoxDecoration(
      //affichage container
      borderRadius: BorderRadius.only(
        topLeft: Radius.circular(10), //arrondi les bords
        topRight: Radius.circular(10),
      ), // BorderRadius.only
      color: Colors.blue, //color fond
    ), // BoxDecoration
    child: AnimatedContainer(
      //widjet d'animation perso
      alignment:
          Alignment(luffyX, luffyY), //possition perso
      duration: const Duration(milliseconds: 0),
      child: MyLuffy(
        //appelle class MyLuffy
        direction: direction, //image direction clas myluffy
        midrun: midrun, //image direction clas myluffy
      ), // MyLuffy
```

```
Expanded(
  //container terre
  //prend 1/5 place du container
  flex: 1,
  child: Container(
    decoration: const BoxDecoration(
      borderRadius: BorderRadius.only(
        bottomLeft: Radius.circular(10),
        bottomRight: Radius.circular(10),
      ), // BorderRadius.only
      gradient: LinearGradient(
          //dégradé de color
          begin:
              Alignment.topCenter, //dégradé de haut en bas
          end: Alignment.bottomCenter,
          colors: [
            Color(0xff103c1a), //color verte
            Color(0xff4c392d), //color marron
          ]), // LinearGradient
```

```
Container(
  //container des button pour faire bouger le perso
  width: 350,
  margin: const EdgeInsets.fromLTRB(0, 35, 0, 120),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: [
      GestureDetector(
        //button left
        onTapDown: (details) {
          //appuyer sur le button
          holdingButton = true; //maintient button on
          moveLeft(); //fonction pour faire bouge à gauche
        },
        onTapUp: (details) {
          // relacher le button
          holdingButton = false; //maintient button off
        },
```

Le code des autres bouton jump et right sont similaires.

Pour le bouton de redirection je n'ai rien ajouté de nouveau si ce n'est le widjet MaterialButton et le onPressed qui exécute un bout de code lors de l'appui du bouton.

```
MaterialButton(
  //button redirection boutique
  child: SizedBox(
    width: 160,
    height: 50,
    child: Padding(
      padding: const EdgeInsets.all(10),
      child: Text(
        "BOUTIQUE",
        style: GoogleFonts.medievalSharp(
            textStyle: Theme.of(context).textTheme.headline4,
            fontSize: 30,
            fontWeight: FontWeight.bold,
            color: const Color(0xff000000)),
        textAlign: TextAlign.center,
      ), // Text
```

```
// ),
onPressed: () {
  //quand on appuie sur le button
  Navigator.push(
    context,
    //redirige vers la page deux "boutique"
    MaterialPageRoute(builder: (context) => const SecondPage()),
  );
```

C) Luffy

```
//pas de changement dynamique
class MyLuffy extends StatelessWidget {
  final direction; //déclaration variable    Prefer typing uninitialize
  final midrun; //déclaration variable    Prefer typing uninitialized v

  //affectation des variables à la class Luffy
  MyLuffy({this.direction, this.midrun});    Use key in widget construc

  @override
  Widget build(BuildContext context) {
    //condition if pour modifier l'imag du perso selon son orientation
    if (direction == "right") {
      return Container(    SizedBox for whitespace.
          width: 70,
          height: 70,
          child: midrun //enchaine les deux image à la suite
          //illusion de marcher
              ? Image.asset('assets/images/luffy2.png')
              : Image.asset('assets/images/luffy1.png')); // Container
    }
    if (direction == "left") {
      return Container(    SizedBox for whitespace.
          width: 70,
          height: 70,
          child: midrun
              ? Image.asset('assets/images/luffy4.png')
              : Image.asset('assets/images/luffy5.png')); // Container
    } else {
      //possition/image de base perso
      return Transform(
        alignment: Alignment.center,
        transform: Matrix4.rotationY(pi),
        child: Container(    SizedBox for whitespace.
          width: 70,
          height: 70,
          child: Image.asset('assets/images/luffy5.png'),
```

D) Boutik

Comme dit dans l'introduction, la page boutique s'inspire grandement de la page home. C'est pourquoi je vais juste détailler ce qui change, c'est-à-dire les containers de ventes de personnages.

```
Row(//alligne les deux affiches de perso
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    Container(//container premier perso
    //design container
      margin: const EdgeInsets.fromLTRB(0, 40, 0, 20),
      width: 200,
      decoration: BoxDecoration(
        color: const Color(0xffc6b69f),
        borderRadius: BorderRadius.circular(3),
      ), // BoxDecoration
      padding: const EdgeInsets.all(3),
      child: Column(//contenude l'affiche
        children: [
          Container(//titre affiche
            margin:const EdgeInsets.fromLTRB(0, 2, 0, 6),
            height: 30,
            child: Center(
              child: Text(
                "WANTED",
                style: GoogleFonts.rye(
                    textStyle:
                        Theme.of(context).textTheme.headline4,
                    fontSize: 30,
                    color: const Color(0xff000000)),
                textAlign: TextAlign.center,
```

```
CarouselSlider(//slider d'image des perso
  options: CarouselOptions(
      //image slider centrer
      enlargeCenterPage: true,
      //activer le scroll infini
      enableInfiniteScroll: false,
      //slide automatique
      autoPlay: true), // CarouselOptions
  items: [//image à dérouler
    Image.asset("assets/images/zoro1.png"),
    Image.asset("assets/images/zoro2.png"),
    Image.asset("assets/images/zoro3.png"),
  ],
),//la suite n'est que du text dans des containers
Container(
  margin: EdgeInsets.fromLTRB(0, 2, 0, 2),    Prefer
  height: 30,
  child: Center(
    child: Text(
      "DEAD OR ALIVE",
      style: GoogleFonts.rye(
          textStyle:
```

III. CONCLUSION

A) Problèmes rencontrés

J'ai eu quelque problème pour ajouter un fond d'écran, aussi pour créer un bouton il y avait tellement de manière différente que j'ai été perdu un moment donner.

Le dernier problème auquel j'ai eu a faire c'était pour le jeu vidéo car je n'avait vraiment aucune idée de comment procéder ni par où commencer. Alors j'ai suivi un tuto qui m'a bien appris les bases.

B) Rendu final du projet