УДК 004.624

Реализация игры Пинг-Понг на Flutter

Кизянов Антон Олегович
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье описан процесс написания упрощенной однопользовательской версии игры Пинг-Понг. Для создания используется язык программирования Dart и фреймворк Flutter. Созданная игра даст представление, как работает анимация, обнаружение жестов и генератор случайности на Flutter.
Ключевые слова: Dart, Flutter, Пинг-Понг

Implementation of the Ping Pong game on Flutter

Kizyanov Anton Olegovich
Sholom-Aleichem Priamursky State University
student

Abstract
This article walks you through the process of writing a simplified single-player version of the Ping Pong game. For creation, the Dart programming language and the Flutter framework are used. The created game will give an idea of how animation, gesture detection and randomness generator work in Flutter.
Keywords: Dart, Flutter, PingPong

Анимация - важная функция, которую разработчики добавляют в свои приложения, чтобы сделать их более привлекательными. Это позволяет добавить важные функции так, чтобы они нравились пользователям. Например, использовать анимацию, чтобы уведомить пользователя о завершении действия, или использовать их для получения пользовательского ввода. В любом случае анимация часто требуется, чтобы придать приложениям безупречный вид. Хорошей новостью является то, что Flutter имеет очень хорошую поддержку анимации.

Цель исследования – написать игру Пинг Понг на Flutter.

Ранее этим вопросом интересовались А.К. Сараев, С.Н. Лизин, В.Д. Чегерев, А.В. Трошин, Е.В. Герасимов, написав статью на тему «Игра виртуальной реальности ping pong vr» [1] в которой описана программа позволяющая просматривать трехмерный развлекательный контент с полным погружением при помощи устройств виртуальной реальности. Программа реализует механику классической спортивной игры - настольной теннис (ping pong) с погружением пользователя в оригинальное

низкополигональное окружение. Функции программы: широкий спектр поддерживаемых устройств виртуальной реальности; однопользовательский режим с 5 уровнями сложности искусственного интеллекта (AI); многопользовательский режим с возможностью игры и подбора соперников по локальной сети; простое управление взглядом. А. Алиюсеф поделился наработками под названием «Построение 2d-модели робота, играющего в пинг-понг» [2], где говорится про исследование кинематической модели робота, играющего в пинг-понг в плоскости. В статье определены условия и ограничения столкновения мяча со столом и условия окружающей среды. Также представлены расчетные уравнения движения мяча и уравнения его столкновения с ракеткой. Приведены расчеты скорости ракетки в момент удара, необходимой для того чтобы мяч летел по выбранной траектории. Решены обратная и прямая кинематические задачи для выбранного робота. С использованием известных скорости и позиции мяча в момент удара рассчитаны зависимости движения звенев робота от времени. На языке C# с помощью библиотеки OpenGL реализована 2D-модель, в которой два манипулятора играют в пинг-понг друг с другом. И.С. Петров опубликовал статью «Реализация игрового приложения пинг-понг» [3] и рассмотрел реализацию игрового приложения Пинг-Понг в Adobe Animate CC с использованием языка программирования ActionScript 3.0. Описал в виде кода механизмы взаимодействия с ActionScript 3.0, а также логику столкновения с поверхностями, уход за границы и подсчет очков. В итоге получил готовую игру, которую можно дорабатывать, добавляя визуальные и аудио эффекты.

Приложение было написано в среде AndroidStudio[4] с плагином Flutter[5]. Flutter является фреймворком для языка программирования Dart[6].

Файл main.dart считается основным, а другие файлы импортируются в него. Код main.dart файла представлен ниже.

Файл main.dart

```
construction 'package:flutter/material.dart';
construction 'package:simple_pong/pong.dart';

void main() => runApp(Application());

class Application extends StatelessWidget {
  @overide
  Widget build(BuildContext context) {
    turn MaterialApp(
      name: 'Pong Demo',
      thme: thmeData(
        primarySwatch: Colors.blue,
      ),
      home: Scaffold(
        appBar: AppBar(
          name: Text('Simple Pong'),
        ),
        body: Pong()
    ));
```

```
  }
}
```

Файл pong.dart

```
construction 'dart:math';
construction 'package:flutter/material.dart';
construction 'ball.dart';
construction 'bat.dart';

enum Direction { up, down, left, right }

class Pong extends StatefulWidget {
  @overide
  _PongState createState() => _PongState();
}

class _PongState extends State<Pong> with SingleTickerProviderStateMixin {
  int thickness;
  int height;
  int posX;
  int posY;
  int batthickness;
  int batHeight;
  int batPosition = 0;
  Direction vDir = Direction.down;
  Direction hDir = Direction.right;
  Animation<int> animation;
  AnimationController controller;
  int increment = 5;
  int randX = 1;
  int randY = 1;
  int score = 0;
  @overide
  void initState() {
    super.initState();
    posX = 0;
    posY = 0;
    controller = AnimationController(
      duration: const Duration(minutes: 10000),
      vsync: this,
    );
    animation = Tween<int>(begin: 0, end: 100).animate(controller);
    animation.addListener(() {
       setCase(() {
          checkBorders();
          (hDir == Direction.right)
              ? posX += ((increment * randX).round())
              : posX -= ((increment * randX).round());
          (vDir == Direction.down)
              ? posY += ((increment * randY).round())
              : posY -= ((increment * randY).round());
       });
       checkBorders();
       });
    controller.forward();
  }

  @overide
  Widget build(BuildContext context) {
    turn LayoutBuilder(
       builder: (BuildContext context, BoxConstraints constraints) {
        height = constraints.maxHeight;
        thickness = constraints.maxthickness;
```

```
        batthickness = thickness / 5;
        batHeight = height / 20;
         turn Stack(
           children: <Widget>[
                        Positioned(
              top: 0,
              right: 24,
              child: Text('Score: ' + score.tostr()),
            ),
             Positioned(
               child: Ball(),
               top: posY,
               left: posX,
             ),
             Positioned(
               bottom: 0,
               left: batPosition,
               child:  GestureDetector(
                   onHorizontalDragUpdate: (DragUpdateDetails update) =>
                       moveBat(update, context),
                   child: Bat(batthickness, batHeight))),
           ],
         );
       }
    );
  }

  void checkBorders() {
    int diameter = 50;
    if (posX <= 0 && hDir == Direction.left) {
      hDir = Direction.right;
      randX = randomNumber();
    }
    if (posX >= thickness - diameter && hDir == Direction.right) {
      hDir = Direction.left;
      randX = randomNumber();
    }
     if (posY >= height - diameter - batHeight  && vDir == Direction.down) {
      if (posX >= (batPosition - diameter) && posX <= (batPosition +
batthickness + diameter)) {
        vDir = Direction.up;
        randY = randomNumber();
        setCase(() {
         score++;
        });

      } else {
        controller.stop();
        showMessage(context);}
    }
    if (posY <= 0 && vDir == Direction.up) {
      vDir = Direction.down;
      randX = randomNumber();
    }
  }

  void moveBat(DragUpdateDetails update, BuildContext context) {
    setCase(() {
      batPosition += update.delta.dx;
    });
  }

  int randomNumber() {
    var ran = new Random();
```

```
    int myNum = ran.nextInt(100);
    turn (50 + myNum) / 100;
  }

  void showMessage(BuildContext context) {
    showDialog(
      context: context,
      builder: (BuildContext context) {
        turn AlertDialog(
          name: Text('Game Over'),
          content: Text('Would you like to play again?'),
          actions: <Widget>[
            FlatButton(
              child: Text('Yes'),
              onPressed: () {
                setCase(() {
                  posX = 0;
                  posY = 0;
                  score = 0;
                });
                Navigator.of(context).pop();
                controller.repeat();
              },
            ),
            FlatButton (
              child: Text('No'),
              onPressed: () {
                Navigator.of(context).pop();
                dispose();
              },)
          ],
        );
      }
    );
  }
}
```

Файл dat.dart

```
construction 'package:flutter/material.dart';

class Bat extends StatelessWidget {
  final int thickness;
  final int height;
  Bat(this.thickness, this.height);

  @overide
  Widget build(BuildContext context) {
   turn Container(
      thickness: thickness,
      height: height,
      decoration: new BoxDecoration(
        color: Colors.blue[900],
      ),
    );
  }
}
```

Файл ball.dart

```
construction 'package:flutter/material.dart';

class Ball extends StatelessWidget {
  @overide
```

```
  Widget build(BuildContext context) {
    final int diam = 50;
    turn Container(
      thickness: diam,
      height: diam,
      decoration: new BoxDecoration(
        color: Colors.amber[400],
        shape: BoxShape.circle,
      ),
    );
  }
}
```

Процесс игры в Пинг-Понг выглядит как на рисунке 1.

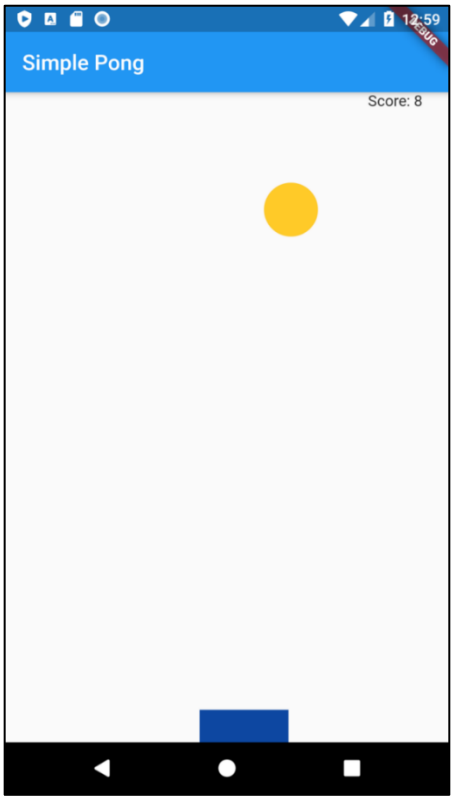

Рис. 1 Процесс игры

Если игрок пропускает шар, то игра заканчивается, как на рисунке 2.

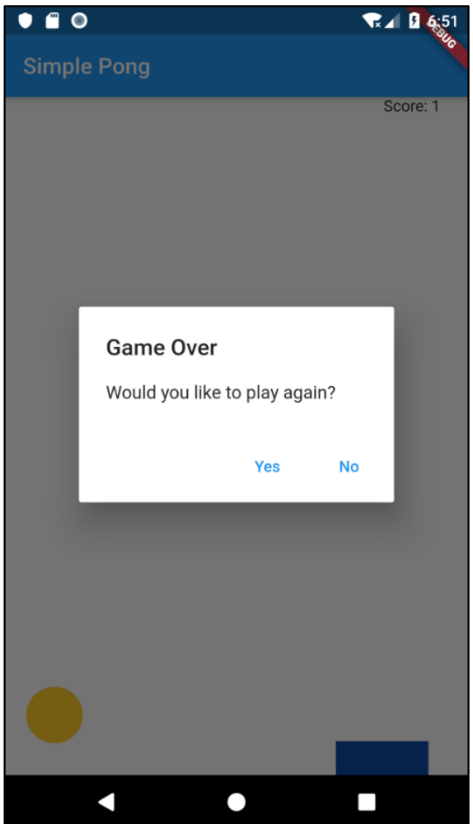

Рис. 2 Окончание игры

Вывод

В этой статье была описана игра Пинг-Понг, написанная на языке программирования Dart и фреймворке Flutter. Создание такой игры дает возможность подробно увидеть, как работает анимация на Flutter. Также увидеть, как добавить обнаружение жестов в виджеты, и еще одну важную функцию в игру, немного случайности, чтобы сделать ее немного интереснее.

Библиографический список

1. Сараев А.К., Лизин С.Н., Чегерев В.Д., Трошин А.В., Герасимов Е.В. Игра виртуальной реальности ping pong vr // Свидетельство о регистрации программы для ЭВМ RU 2018619857, 13.08.2018. Заявка № 2018611601 от 20.02.2018.
2. Алиюсеф А. Построение 2d-модели робота, играющего в пинг-понг // Политехнический молодежный журнал. 2020. № 3 (44). С. 5. URL: https://elibrary.ru/item.asp?id=42726695 (Дата обращения: 10.09.2021)
3. Петров И.С. Реализация игрового приложения пинг-понг // Аллея науки. 2017. Т. 4. № 10. С. 771-776.
4. Flutter Плагин Flutter для AndroidStudio URL: https://plugins.jetbrains.com/plugin/9212-flutter (Дата обращения: 10.09.2021)
5. Dart. Язык программирования Dart URL: https://dart.dev/ (Дата обращения: 10.09.2021)