Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Комсомольский-на-Амуре государственный университет»

Факультет компьютерных технологий

Кафедра «МОП ЭВМ»

ЛАБОРАТОРНАЯ РАБОТА 7

по дисциплине «Программирование мобильных устройств»

2D-графика. Касания

Студент группы 6ИСб-1 О.Б. Хусейнов

Преподаватель В.А. Тихомиров

2020

**Задания**

По вариантам указано, что должна воспроизводить на экране программа, и приведен примерный вид экрана в момент работы программы (стрелки – показывают направление движение объекта, их программировать НЕ надо).

Слово HELLO при щелчке на нем разлетается на составляющие линии а потом собирается назад.

**Ход выполнения работы**

При разработке приложения были использованы 2 файла в Android Studio. Содержимое файла AndroidManifest.xml приведено в листинге 1.

Листинг 1 - MainActivity.xml

package com.knastu.lab7;

import android.app.Activity;

import android.os.Bundle;

public class MainActivity extends Activity {

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(new Board(this));

}

}

Содержимое файла Board.java приведено в листинге 2.

Листинг 2 - Board.java

package com.knastu.lab7;

import android.content.Context;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.Paint;

import android.view.MotionEvent;

import android.view.View;

import androidx.annotation.NonNull;

public class Board extends View {

//// USER SETTINGS

// Text property

int symbolHeight = 100, symbolWidth = 50, symbolSpace = 20;

int paintStartX = 360, paintStartY = 620;

// Animation property

int duration = 200;

//// SYSTEM SETTINGS (not edit)

int tempDuration = 0;

boolean animationProgress = false, animationReturn = false;

private Paint paint = new Paint();

public Board(Context context) {

super(context);

paint.setColor(Color.BLUE);

paint.setAntiAlias(true);

paint.setStrokeWidth(5);

}

@Override

protected void onDraw(Canvas c) {

int tempPaintStartX = paintStartX, tempPaintStartY = paintStartY;

// DRAW ANIMATION

if (animationProgress || animationReturn) {

// ANIMATION BACK

if (animationProgress)

if (++tempDuration > duration) {

animationProgress = false;

animationReturn = true;

}

// ANIMATION BACK

if (animationReturn)

if (--tempDuration < 0) {

animationProgress = false;

animationReturn = false;

}

// H

tempPaintStartY -= symbolHeight / 2 - tempDuration;

drawVerticalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY += symbolHeight / 2 - tempDuration;

drawHorizontalLine(c, tempPaintStartX - tempDuration, tempPaintStartY + tempDuration);

tempPaintStartX += symbolWidth;

tempPaintStartY -= symbolHeight / 2;

drawVerticalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY += symbolHeight / 2;

tempPaintStartX += symbolSpace - tempDuration;

// E

tempPaintStartY -= symbolHeight / 2;

drawVerticalLine(c, tempPaintStartX + tempDuration, tempPaintStartY + tempDuration);

drawHorizontalLine(c, tempPaintStartX - tempDuration, tempPaintStartY + tempDuration);

tempPaintStartY += symbolHeight / 2;

drawHorizontalLine(c, tempPaintStartX, tempPaintStartY + tempDuration);

tempPaintStartY += symbolHeight / 2;

drawHorizontalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY -= symbolHeight / 2;

tempPaintStartX += symbolSpace + symbolWidth;

// L

tempPaintStartY -= symbolHeight / 2;

drawVerticalLine(c, tempPaintStartX - tempDuration, tempPaintStartY - tempDuration);

tempPaintStartY += symbolHeight;

drawHorizontalLine(c, tempPaintStartX, tempPaintStartY + tempDuration);

tempPaintStartY -= symbolHeight / 2;

tempPaintStartX += symbolSpace + tempDuration + symbolWidth;

// L

tempPaintStartY -= symbolHeight / 2;

drawVerticalLine(c, tempPaintStartX + tempDuration, tempPaintStartY + tempDuration);

tempPaintStartY += symbolHeight - tempDuration;

drawHorizontalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY -= symbolHeight / 2;

tempPaintStartX += symbolSpace + symbolWidth;

// O

tempPaintStartY -= symbolHeight / 2;

drawVerticalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY += symbolHeight;

drawHorizontalLine(c, tempPaintStartX, tempPaintStartY - tempDuration);

tempPaintStartX += symbolWidth - tempDuration;

tempPaintStartY -= symbolHeight;

drawVerticalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartX -= symbolWidth;

drawHorizontalLine(c, tempPaintStartX + tempDuration, tempPaintStartY + tempDuration);

}

// DRAW HELLO

else {

// H

tempPaintStartY -= symbolHeight / 2;

drawVerticalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY += symbolHeight / 2;

drawHorizontalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartX += symbolWidth;

tempPaintStartY -= symbolHeight / 2;

drawVerticalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY += symbolHeight / 2;

tempPaintStartX += symbolSpace;

// E

tempPaintStartY -= symbolHeight / 2;

drawVerticalLine(c, tempPaintStartX, tempPaintStartY);

drawHorizontalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY += symbolHeight / 2;

drawHorizontalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY += symbolHeight / 2;

drawHorizontalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY -= symbolHeight / 2;

tempPaintStartX += symbolSpace + symbolWidth;

// L

tempPaintStartY -= symbolHeight / 2;

drawVerticalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY += symbolHeight;

drawHorizontalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY -= symbolHeight / 2;

tempPaintStartX += symbolSpace + symbolWidth;

// L

tempPaintStartY -= symbolHeight / 2;

drawVerticalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY += symbolHeight;

drawHorizontalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY -= symbolHeight / 2;

tempPaintStartX += symbolSpace + symbolWidth;

// O

tempPaintStartY -= symbolHeight / 2;

drawVerticalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartY += symbolHeight;

drawHorizontalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartX += symbolWidth;

tempPaintStartY -= symbolHeight;

drawVerticalLine(c, tempPaintStartX, tempPaintStartY);

tempPaintStartX -= symbolWidth;

drawHorizontalLine(c, tempPaintStartX, tempPaintStartY);

}

invalidate();

}

@Override

public boolean onTouchEvent(@NonNull MotionEvent event) {

if (event.getAction() == MotionEvent.ACTION\_DOWN && !animationProgress && !animationReturn) {

animationProgress = true;

}

return true;

}

private void drawHorizontalLine(Canvas canvas, int tempPaintStartX, int tempPaintStartY) {

canvas.drawLine(tempPaintStartX, tempPaintStartY, tempPaintStartX + symbolWidth, tempPaintStartY, paint);

}

private void drawVerticalLine(Canvas canvas, int tempPaintStartX, int tempPaintStartY) {

canvas.drawLine(tempPaintStartX, tempPaintStartY, tempPaintStartX, tempPaintStartY + symbolHeight, paint);

}

}

Результат работы приложения показан на рисунках 1-3.

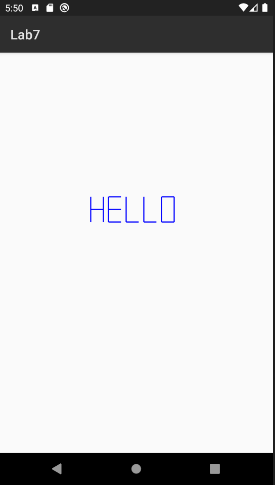


Рисунок 1 – Начальный экран приложения

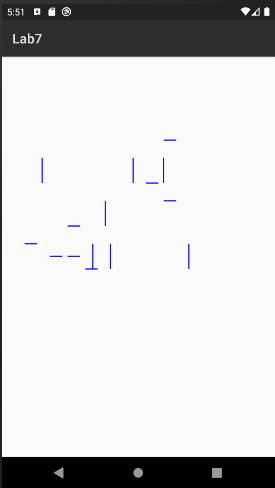


Рисунок 2 – Анимация разложения слова

**Список использованных источников**

1. Дейтел, П. Android для разработчиков. 3-е изд. / П. Дейтел, Х. Дейтел, А. Уолд. — СПб.: Питер, 2016. — 512 с.
2. Гриффитс. Д. Head First. Программирование для Android / Дон Гриффитс, Дэвид Гриффитс — СПб.: Питер, 2016. — 704 с.