画板 APP 说明文档

Company：Robotrak
Author：舒志豪
Date：2020/08/03

目录

使用背景与介绍

1. 该 app 的作用是提供给要进行激光打击操作的医生的一个练习软件，本 app 并不能直接与激光器相连并进行手术。后期如有需求可根据需求完善该 app 并增加其激光打击，标定等功能。(注意：目前眼底图片或其他图片在该 app 上显示时会作为背景并进行缩放，所以标定坐标可能不准确)。
2. 该 app 可与 Robotrak 官网连接，后期可在官网选图进行标定操作（目前官网并未发布，选图页面并未完善)。
3. 该 app 并不是完善版本，还需继续改进。
4. 使用机型：**华为荣耀 Play4T HONOR**

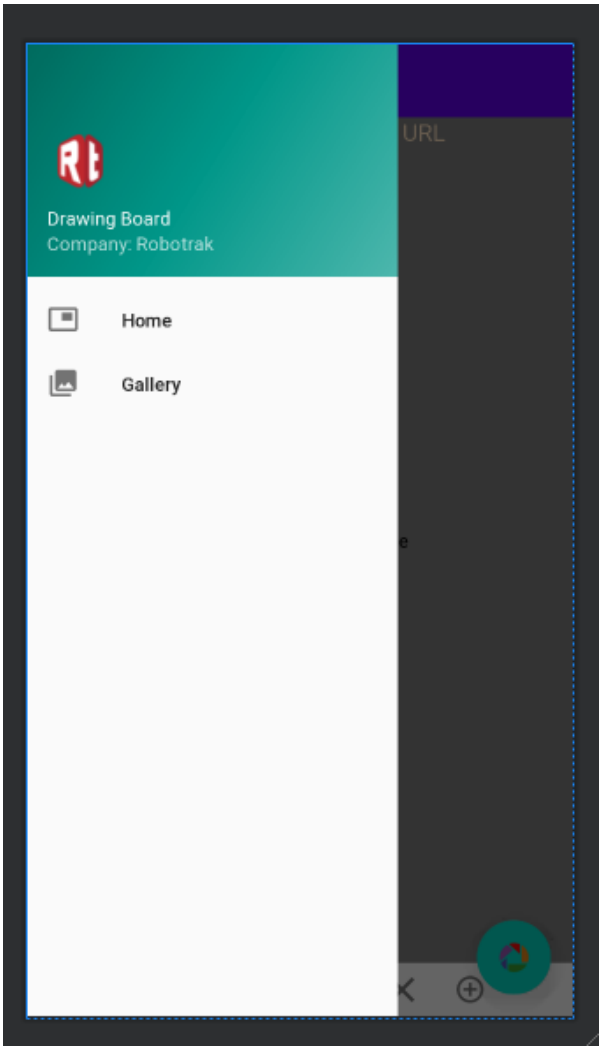
   App 使用 ide： Android studio 4.0

   SDK： 30

   理论上只要是 Android 系统都能用该 app。

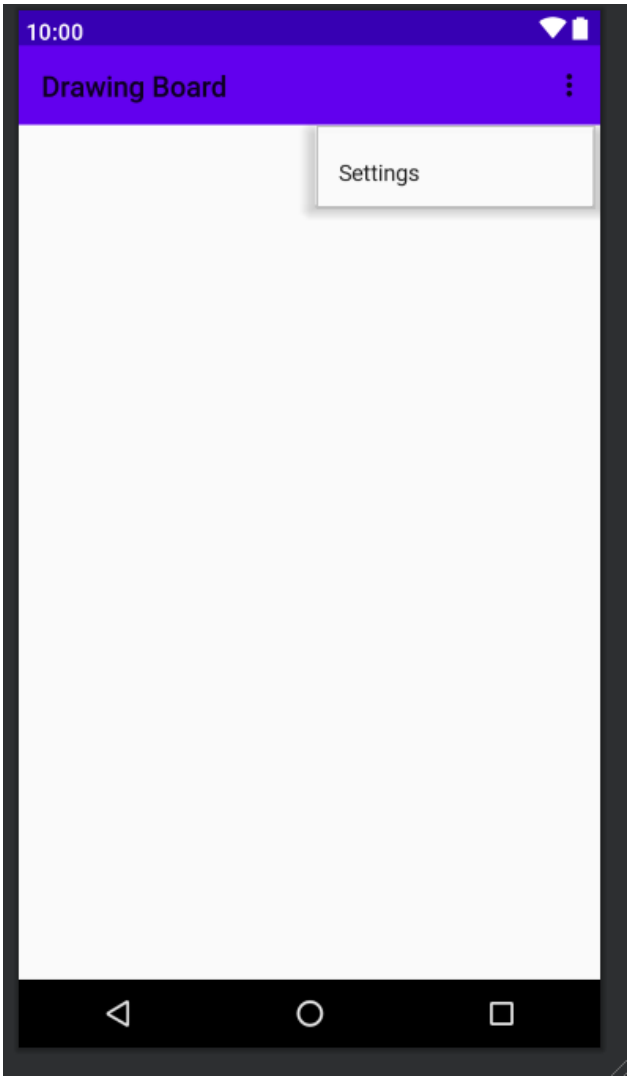
App 代码详解

- app 结构

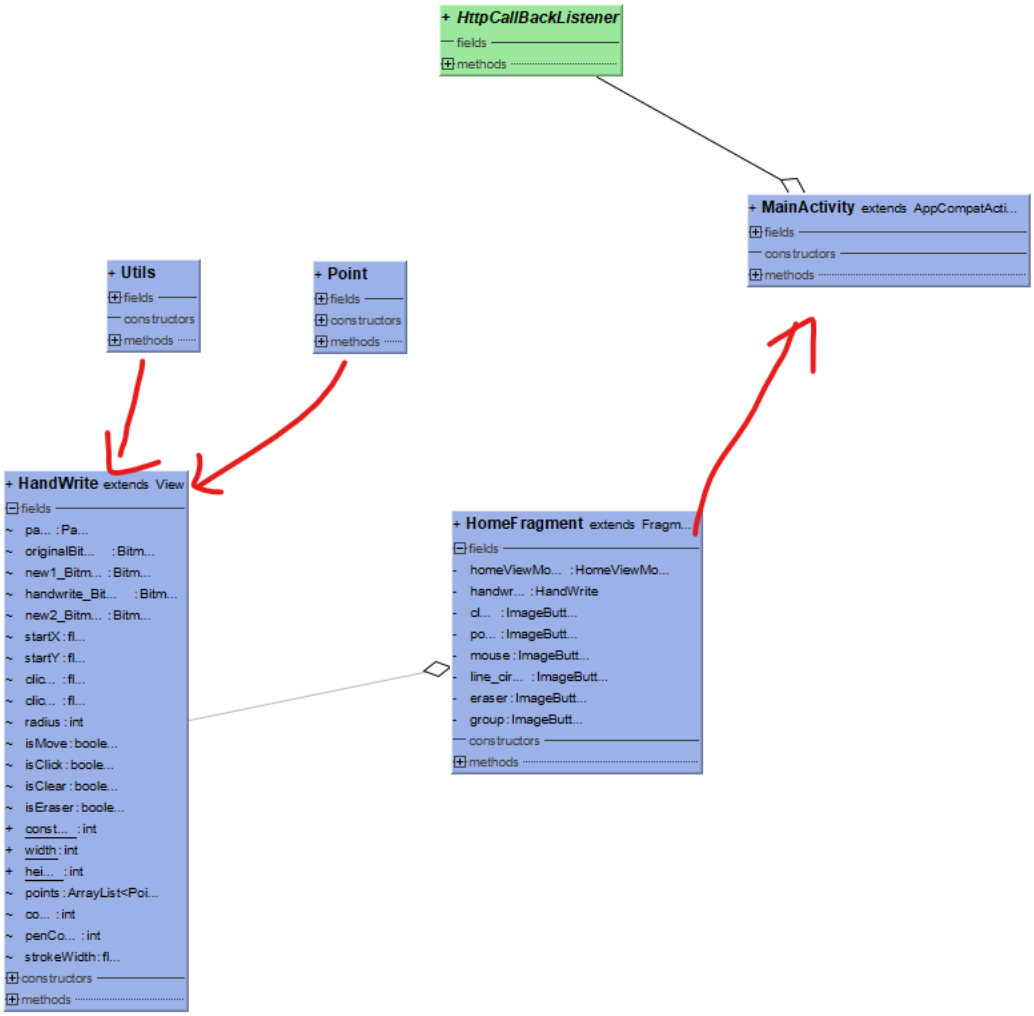


图表 1

图表 2

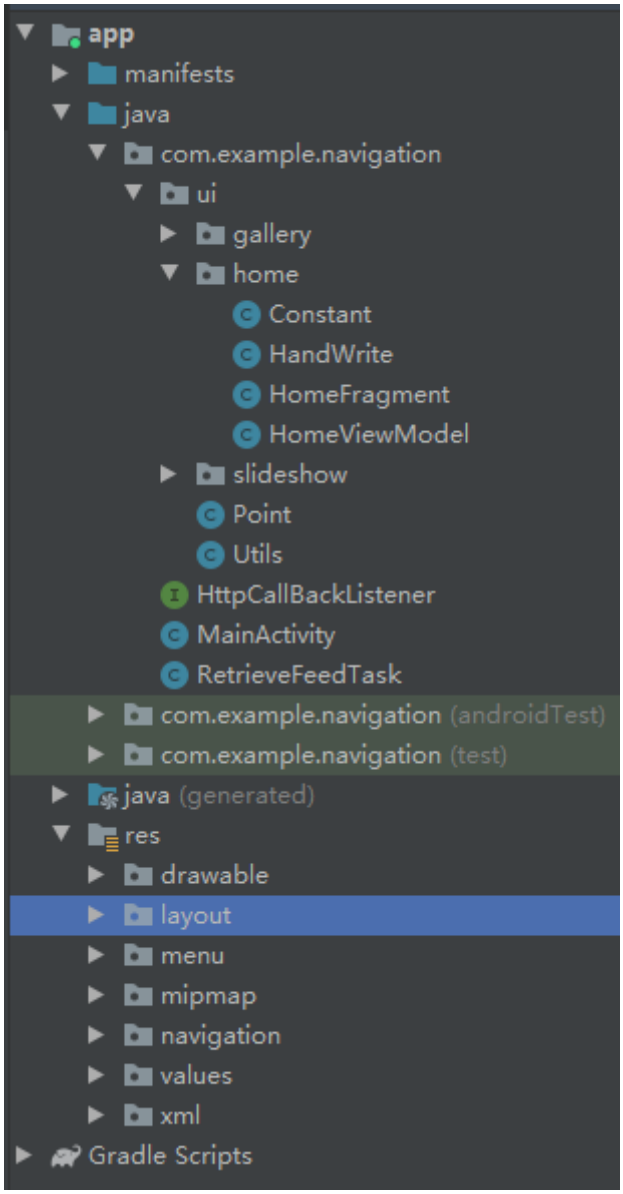


图表 3



图表 4

- 本 app 利用了 Google 的 navigation 模板。图表 1 为 app 的导航模块，可将 Home Fragment 或 Gallery Fragment 导入 Main Activity 主界面（Slide Show Fragment 目前废弃）



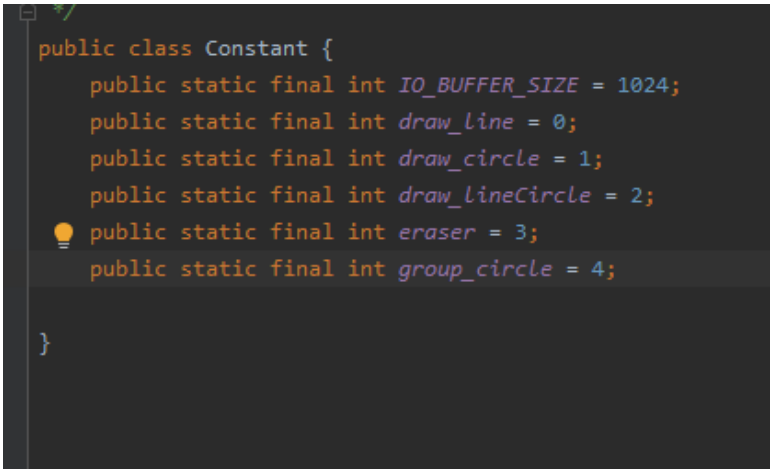
1. Manifests：在其中规定了该 app 的 app 图标，权限，屏幕方向等。 目前我将 app 设为了横屏，且不能更改。
2. UI：

   Gallery：目前没用。后期想法是将网站上的图直接导入这个模块中，做成 listView，到时候就不需要跳转到网页去选照片。关于 listview，推荐使用 Picasso，该 app 也已经导入了 Picasso 工具包。Picasso 使用：

https://www.jianshu.com/p/c68a3b9ca07a

home：**Constant**

a)



```
public class Constant {
    public static final int IO_BUFFER_SIZE = 1024;
    public static final int draw_line = 0;
    public static final int draw_circle = 1;
    public static final int draw_lineCircle = 2;
    public static final int eraser = 3;
    public static final int group_circle = 4;

}
```

规定了各种点击事件后所对应的值，用他们来表明接下来是要画单点，多点，线条，还是写字等等。

Handwrite：自定义的画布 view。

1. 其原理是将 mipmap 中的一张 default 眼底图当作背景存在 originalBitmap 中

```
public HandWrite(Context context, AttributeSet attrs)
{
    super(context, attrs);

   // originalBitmap = GetLocalOrNetBitmap("https://blog.foreverlove.us/girl2.png");

    points = new ArrayList<Point>();
    if(MainActivity.newBitmap != null){
        originalBitmap = null;
        originalBitmap = MainActivity.newBitmap;
        new1_Bitmap = Bitmap.createBitmap(originalBitmap);
        handwrite_Bitmap = Bitmap.createBitmap(originalBitmap);
    }else{
        originalBitmap = null;
        originalBitmap = BitmapFactory
                .decodeResource(getResources(), R.mipmap.test)
                .copy(Bitmap.Config.ARGB_8888, isMutable: true);
        new1_Bitmap = Bitmap.createBitmap(originalBitmap);
        handwrite_Bitmap = Bitmap.createBitmap(originalBitmap);
    }

}
```

再根据屏幕的长宽进行缩放，这一步在 onDraw 中完成。

如果从网上导入了新的照片，或从相册导入了新照片，将刷新 originalBitmap

```
@Override
protected void onDraw(Canvas canvas)
{
    super.onDraw(canvas);
    new1_Bitmap = scaleBmp(new1_Bitmap,width,height);

    switch (constant){
        case Constant.draw_line:{
            canvas.drawBitmap(HandWriting(new1_Bitmap), left: 0, top: 0, paint: null);
            break;
        }
        case Constant.draw_circle:{
            canvas.drawBitmap(DrawCircle(new1_Bitmap), left: 0, top: 0, paint: null);
            break;
        }
        case Constant.draw_lineCircle:{
            canvas.drawBitmap(LineCircle(new1_Bitmap), left: 0, top: 0, paint: null);
            break;
        }
        case Constant.eraser:{
            canvas.drawBitmap(Eraser(new1_Bitmap), left: 0, top: 0, paint: null);
            break;
        }
        case Constant.group_circle:{
            canvas.drawBitmap(groupCircle(new1_Bitmap), left: 0, top: 0, paint: null);
        }
    }

}
```

涂鸦方法：在 original bitmap 的基础上新建一个 new1_bitmap，再利用自定义方法在 new1_bitmap 上经行修改。以 HandWriting()为例：

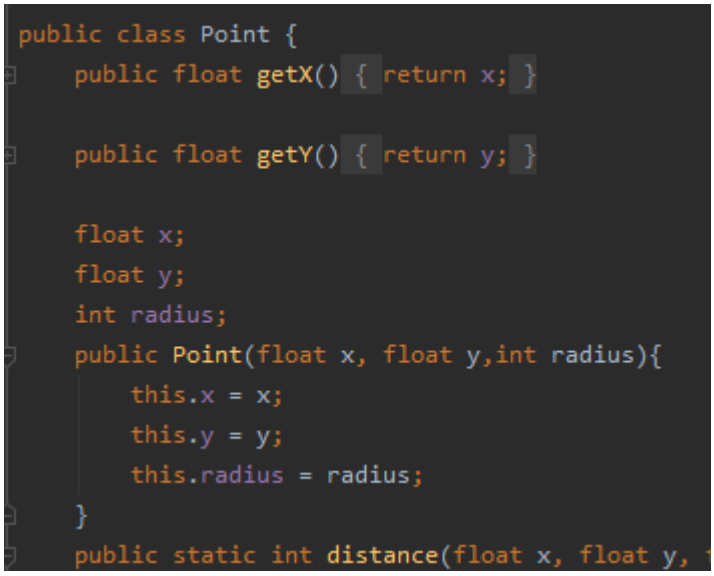
```
public Bitmap HandWriting(Bitmap o_Bitmap)
{
    Canvas canvas = null;
    if(isClear) {
        canvas = new Canvas(new2_Bitmap);
    }
    else{
        canvas = new Canvas(o_Bitmap);
    }
    paint = new Paint();
    paint.setStyle(Paint.Style.STROKE);
    paint.setAntiAlias(true);
    paint.setColor(penColor);
    paint.setStrokeWidth(strokeWidth);
    if(isMove)
    {
        canvas.drawLine(startX, startY, clickX, clickY, paint);
    }
    startX = clickX;
    startY = clickY;
    if(isClear)
    {
        return new2_Bitmap;
    }
    return o_Bitmap;
}
```

解释：当用户点击全部清理时，新建一个以 original bitmap 为基础的 new2_bitmap 并直接返回。

否则是在 new1_bitmap 上进行画图并返回更改过的图。

HomeFragment：确定了各种 onclick 方法，并相应地改变 Constant 值使之进行相应的涂鸦操作。

- Point：



```
public class Point {
    public float getX() { return x; }

    public float getY() { return y; }

    float x;
    float y;
    int radius;
    public Point(float x, float y,int radius){
        this.x = x;
        this.y = y;
        this.radius = radius;
    }
    public static int distance(float x, float y,
```

该类确定了每个打点的坐标，及其圆圈的半径。

原理：每当进行打点操作都会新建一个 point 对象并存入 HandWrite 的 Points（ArrayList）中。每当用橡皮擦或是全部清理，就会 remove 相应的 Point。

- Utils：在里面有各种关于 bitmap 的基本操作方法，详情请去看 code
- HttpCallBackListner：

  因为关于网络的请求必须是独立的，所以用了一个 callbacklistener 来进行多线程操作。当 onFinish 时返回从网络上取得图片路径
- MainActivity

  该 class 为程序的主入口，在其中有关于 fragment 跳转的逻辑，得到屏幕的长宽，跳转到网站的逻辑，跳转到相册并选择照片的逻辑，URL textView

的逻辑（目前从网络上只能复制图片链接并粘贴在 URL 中实现背景更新），等等。

Resources：

- Drawable：程序的一些小图标
- Layout：各个 fragment 和 activity 的 layout（可修改）
- Menu：界面上方的选项
- Mipmap：外部导入的图片
- Navigation：存各种 fragment（目前只有 home，gallery）
- Values：一些常数的设置
- Xml：关于 internet permission 的设置

其他

该 app 还有许多可增加的功能，如

1. 存储修改过的图
2. 完成 gallery 中的功能
3. 与网站相连接，只需点击图片就能更换背景
4. 丰富界面和画图功能

注意：

1. 该软件还没有进行 bug 测试，后期需加上。
2. 该软件还没有试过在其他机型上的测试。