面向对象的软件构造实践

实验三
（2学时）

2024春

HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ

实验内容

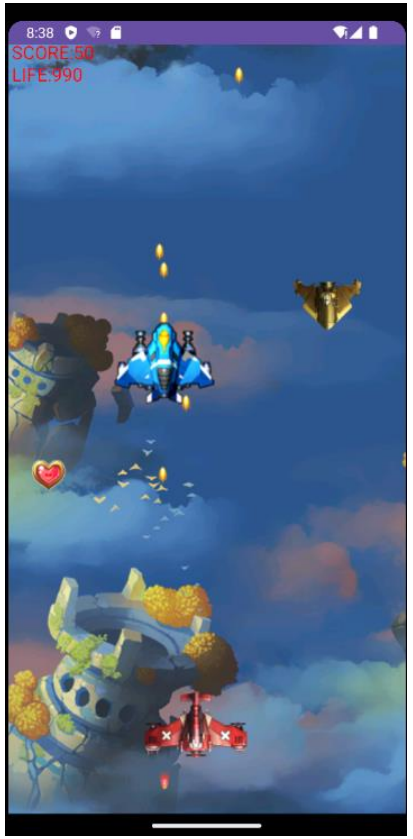
用户界面 | 事件处理

图形系统 | 数据存储与展示 | 音乐音效

网络编程

模块功能：使用图形处理API完成绘制

① 使用SurfaceView和图形处理API，完成不同难度下游戏主界面的开发及实时分数和生命值的显示



实验目的

- 掌握Android系统2D图形处理API的使用；
- 掌握开发Android自定义视图的方法;
- 掌握SurfaceView的使用，了解SurfaceView的生命周期。

3.1 Android图形处理API

3.2 Android SurfaceView

3.1 Android图形处理API

Android系统的图形处理能力非常强大，有一系列2D图形处理类

- Color类
- Paint类
- Canvas类

Color类

Color是颜色管理类。

Android中的颜色用4组数字表示，它们分别指定透明度、红色、绿色、蓝色（Alpha、Red、Green、Blue，即ARGB）所占的比重，每组数字取值在0~255之间（如果使用二进制数制表示，则占8位），为了编程方便，常常使用十六进制的数字来表示颜色，如#FFFFFF代表白色（透明度可省略，默认值为完全不透明）。

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="aubergine">#FF00FF</color>
    <color name="aubergine1">#7FFF00FF</color>
    <color name="pink">#FFC0CB</color>
</resources>
```

Color类

- 在资源文件中colors.xml定义颜色，在JAVA代码中使用getColor()方法根据资源ID查找指定的

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="aubergine">#FF00FF</color>
    <color name="aubergine1">#7FFF00FF</color>
    <color name="pink">#FFC0CB</color>
</resources>
```

```
int color=getResources().getColor(R.color.aubergine);
```

- 如果没有在资源文件中定义颜色，可以在JAVA代码中使用十进制数字来定义颜色。

```
int color=Color.argb(127,255,0,255);  //透明度为127的紫红色
int color=Color.rgb(255,0,255);        //完全不透明的紫红色
```

- 还可以在Java代码中使用系统预定义的标准颜色。

```
int color = Color.BLUE;
```

Paint类

Paint类是包含样式、颜色以及绘制图形所需的信息。

方法	说明
setARGB(int a, int r, int g, int b)	设置Paint对象颜色
setColor(int color)	设置颜色，可以使用系统预定义颜色
setTextAlign(Paint.Align align)	设置绘制文本时的文本对齐方式
setTextSize(float textSize)	设置绘制文本时的文本大小
setTypeface(Typeface typeface)	设置绘制文本时的文本字体

3.1 Android图形处理API

Canvas类

Canvas类代表可以在其上绘制图形的画布。通过重写View.onDraw方法，实现在指定的画布上绘制图形。

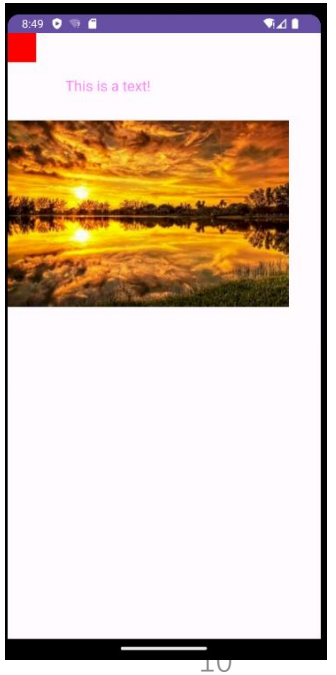
方法	说明
drawText(String text, float x, float y, Paint paint)	在屏幕上描绘文字，参数text是String类型的文本，参数x为水平轴坐标，参数y为垂直轴坐标，参数paint为画笔对象
drawPoint(float x, float y, Paint paint)	画点，参数x为水平轴坐标，参数y为垂直轴坐标
drawLine(float startX, float startY, float endX, float endY, Paint paint)	画线，参数startX为起始点的x坐标，参数startY为起始点的y坐标，参数endX为终点的x坐标，参数eyndY为终点的y坐标
drawCircle(float cx, float cy, float radius, Paint paint)	画圆，参数cx是圆心点的x坐标，参数cy是圆心点的y坐标，参数radius是半径
drawRect(RectF rect, Paint paint)	画矩形，参数rect为一个区域
drawBitmap(Bitmap bitmap, float left, float top, Paint paint)	绘制图片

开发自定义视图

```
public class MyView extends View {
    1 usage
    public MyView(Context context) {
        super(context);
    }
    @Override
    public void onDraw(Canvas canvas){
        super.onDraw(canvas);

        Paint paint = new Paint();
        paint.setColor(Color.RED);
        canvas.drawRect(new Rect( left: 0, top: 0, right: 100, bottom: 100),paint);

        Paint paint2 = new Paint();
        int color = Color.argb( alpha: 127, red: 255, green: 0, blue: 255);
        paint2.setColor(color);
        paint2.setTextSize(50);
        canvas.drawText( text: "This is a text!", x: 200, y: 200,paint2);

        Paint paint3 = new Paint();
        Bitmap pic = BitmapFactory.decodeResource(getResources(), R.drawable.pic);
        canvas.drawBitmap(pic, left: 0, top: 300,paint3);
    }
}
```

1. 创建自定义视图类；
2. 重写onDraw方法，在onDraw方法中使用Color、Paint和Canvas类绘制图形；
3. 在Activity中创建自定义视图类的实例，使用setContentView方法加载此视图实例。

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        MyView view = new MyView( context: this);
        setContentView(view);
    }
}
```



3.2 Android SurfaceView

Android游戏开发常用的三种视图：View、SurfaceView和GLSurfaceView。

View → **SurfaceView** → **GLSurfaceView**

- 显示视图，内置画布，图形绘制函数、触屏事件、按键事件函数等；
- 必须在UI主线程内更新画面，速度较慢。

- View类的子类；
- 使用双缓冲机制；
- 在新的子线程中更新画面，刷新界面速度比View快。
- 适用于2D游戏的开发。

- 基于SurfaceView视图再次进行拓展的视图类，专用于3D游戏开发的视图；
- 是SurfaceView的子类，OpenGL专用。

如何使用SurfaceView

1. 创建SurfaceView

创建自定义的SurfaceView继承自SurfaceView,并实现两个接口：SurfaceHolder.Callback和Runnable

```
public class MySurfaceView extends SurfaceView implements SurfaceHolder.Callback,Runnable
```

2. 初始化SurfaceView

在自定义的SurfaceView的构造函数中，定义并初始化如下的成员变量

```
1    private SurfaceHolder mHolder;
2    private Canvas mCanvas;//绘图的画布
3    private boolean mIsDrawing;//控制绘画线程的标志位
```

```
public MySurfaceView(Context context, AttributeSet attrs, int defStyle) {
    super(context, attrs, defStyle);
    initView();
}

private void initView() {
    mHolder = getHolder();
    mHolder.addCallback(this);
    setFocusable(true);
    setFocusableInTouchMode(true);
    this.setKeepScreenOn(true);
    //mHolder.setFormat(PixelFormat.OPAQUE);
}
```

如何使用SurfaceView

3. 实现SurfaceHolder.Callback接口

```
@Override
public void surfaceCreated(SurfaceHolder holder) {
    mIsDrawing = true;
    new Thread(this).start();
}

@Override
public void surfaceChanged(SurfaceHolder holder,
                           int format, int width, int height) {
}

@Override
public void surfaceDestroyed(SurfaceHolder holder) {
    mIsDrawing = false;
}
```

- 在surfaceCreated()方法中开启子线程进行绘制，子线程使用while(mIsDrawing)的循环来不停的进行绘制；
- 在surfaceDestroyed()方法中设置mIsDrawing为false结束绘制；

如何使用SurfaceView

4. 实现Runnable接口

```
@Override
public void run() {
    while (mIsDrawing) {
        draw();
    }
}
//绘图操作
private void draw() {
    try {
        mCanvas = mHolder.lockCanvas();
        // draw sth绘制过程
    } catch (Exception e) {
    } finally {
        if (mCanvas != null)
            mHolder.unlockCanvasAndPost(mCanvas);//保证每次都将绘图的内容提交
    }
}
```

- 在绘制的逻辑中通过lockCanvas()方法获取Canvas对象进行绘制，通过unlockCanvasAndPost(mCanvas)方法对画布内容进行提交。

实验任务

3-1 导论代码移植

- 移植导论代码到项目中，能够实现飞机大战的基本功能；
- 实现单机模式难度选择页面到游戏界面的跳转。

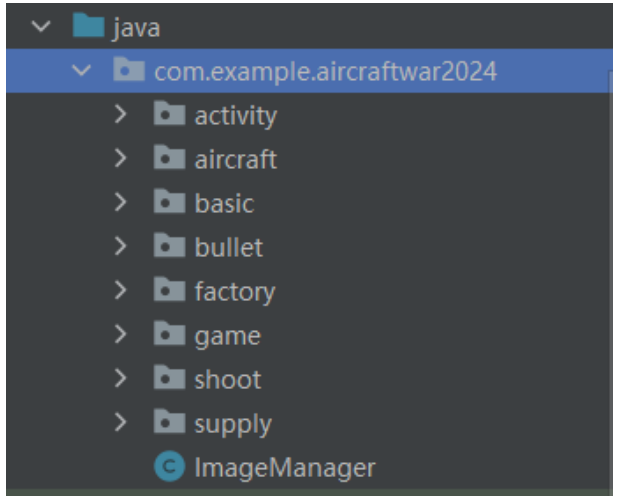
3-2 SurfaceView的开发

- 自定义SurfaceView绘制游戏界面；
- 根据用户选择的难度模式绘制不同的游戏界面；
- 使用图形处理API在游戏界面左上角绘制文本框显示英雄机分数和生命值。

3-1 导论代码移植

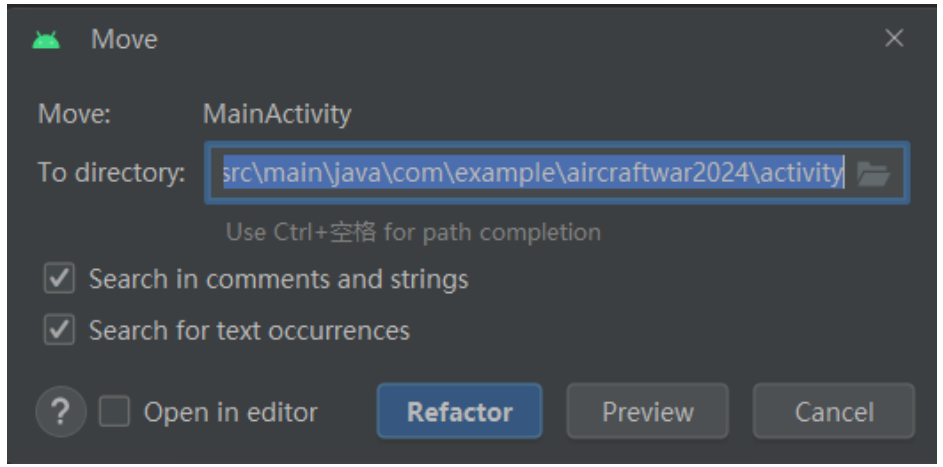
1. 将实验包中的图片资源放到res/drawable文件夹下

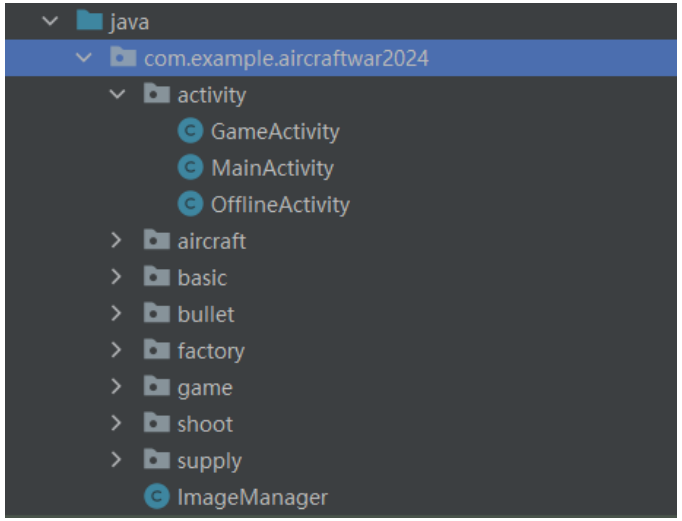
2. 将实验包中的代码解压后拷贝到com.example.AircraftWar2024文件夹下，项目结构如右图



实验步骤

3. 将MainActivity.java和OfflineActivity.java拖到activity文件夹中，选择Refactor





4. 在OfflineActivity.java和GameActivity.java中添加页面跳转代码，能够根据用户选择的模式显示对应的游戏画面

3-2 SurfaceView的开发

1. 在BaseGame.java中补充完整和SurfaceView生命周期有关的代码块。

2. 在BaseGame.java中使用Canvas和Paint在左上角绘制文本框显示英雄机分数和生命值，效果如右图。

```
private void paintScoreAndLife() {
    /*TODO:绘制文本框显示英雄机的分数和生命值*/
}
```