

MICROOH 麦可网

Android-从程序员到架构师之路

出品人：Sundy

讲师：高焕堂（台湾）

<http://www.microoh.com>

B03_c

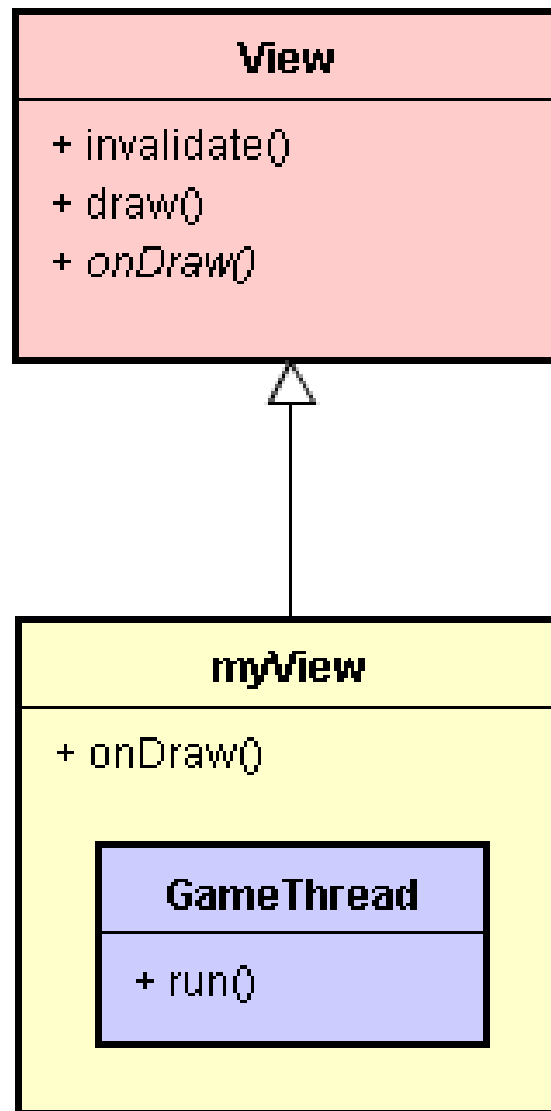
应用Android的UI框架(c)

-- 以设计游戏循环(*GameLoop*)为例

By 高焕堂

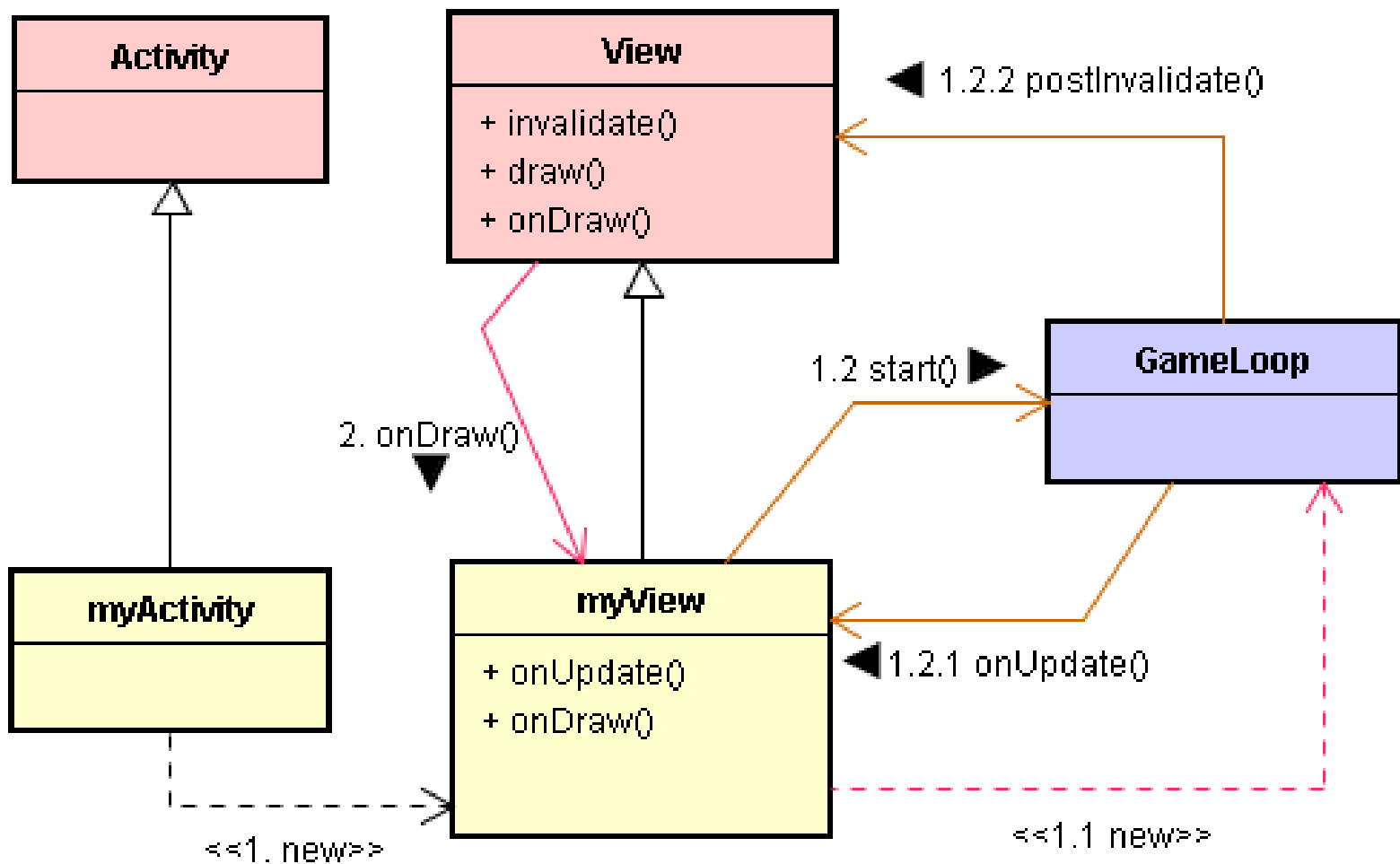
6、设计一个 GameLoop类别

- 刚才的小线程，其实就扮演了游戏线程 (Game thread) 的角色，它负责控制游戏的循环。



```
// myView.java
//.....
public class myView extends View {
    //.....
    @Override protected void onDraw(Canvas canvas) {
        //.....
        myThread t = new myThread();
        t.start();
    }
    class myThread extends Thread{
        public void run() {
            postInvalidateDelayed(1000);
        };
    }
}
```

- 于是，我们将刚才的小线程部分独立出来，成为一个独立的类别，通称为游戏线程 (Game Thread) 或游戏循环(Game Loop)。




```
// GameLoop.java
```

```
// .....
```

```
public class GameLoop extends Thread {  
    myView mView;
```

```
    GameLoop(myView v){  
        mView = v;  
    }
```

```
    public void run() {  
        mView.onUpdate();  
        mView.postInvalidateDelayed(1000);  
    }
```

```
}
```

```
// myView.java
```

```
// .....
```

```
public class myView extends View {  
    private Paint paint= new Paint();  
    private int x, y;  
    private int line_x = 100;  
    private int line_y = 100;  
    private float count = 0;  
  
    myView(Context ctx) {  
        super(ctx);  
    }  
    public void onUpdate(){  
        if( count > 12) count = 0;  
        x = (int) (75.0 * Math.cos(2*Math.PI * count/12.0));  
        y = (int) (75.0 * Math.sin(2*Math.PI * count/12.0));  
        count++;  
    }  
}
```

```
@Override protected void onDraw(Canvas canvas) {  
    super.onDraw(canvas);  
    canvas.drawColor(Color.WHITE);  
    paint.setColor(Color.BLUE);  
    paint.setStrokeWidth(3);  
    canvas.drawLine(line_x, line_y, line_x+x, line_y+y, paint);  
  
    paint.setStrokeWidth(2);  
    paint.setColor(Color.RED);  
    canvas.drawRect(line_x-5, line_y - 5, line_x+5, line_y + 5, paint);  
    paint.setColor(Color.CYAN);  
    canvas.drawRect(line_x-3, line_y - 3, line_x+3, line_y + 3, paint);  
    //-----  
    GameLoop loop = new GameLoop(this);  
    loop.start();  
}  
}
```

- 首先由myActivity来诞生myView对象，然后由Android框架调用myView的onDraw()函数来绘图和显示。绘图完毕，立即诞生一个GameLoop对象，并调用start()函数去启动一个小线程去调用postInvalidate()函数。就触发UI线程重新调用myView的onDraw()函数。



~ Continued ~