

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>

C07_d

问题集：

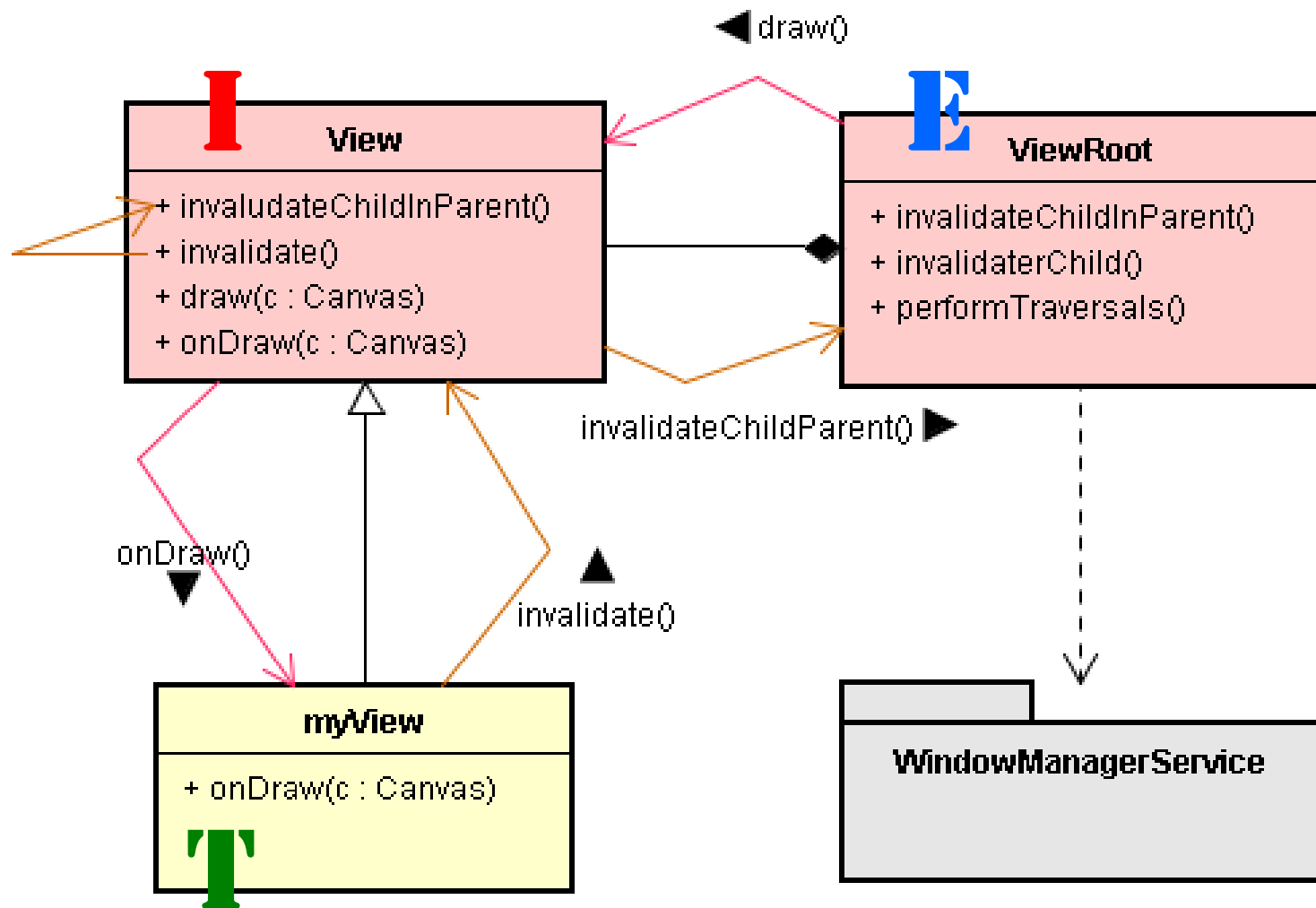
进程、线程和JNI架构(d)

By 高煥堂

問題

A2.6-thread

- 下图里的箭头，都是UI线程所执行的。就构成常见的游戏控制循环(Game Loop)。
- UI线程执行<T>插件时，调用基类的invalidate()函数，触发重新调用onDraw()，来更新画面上的绘图。
- 请问：小线程也可以调用此invalidate()函数吗？Why?



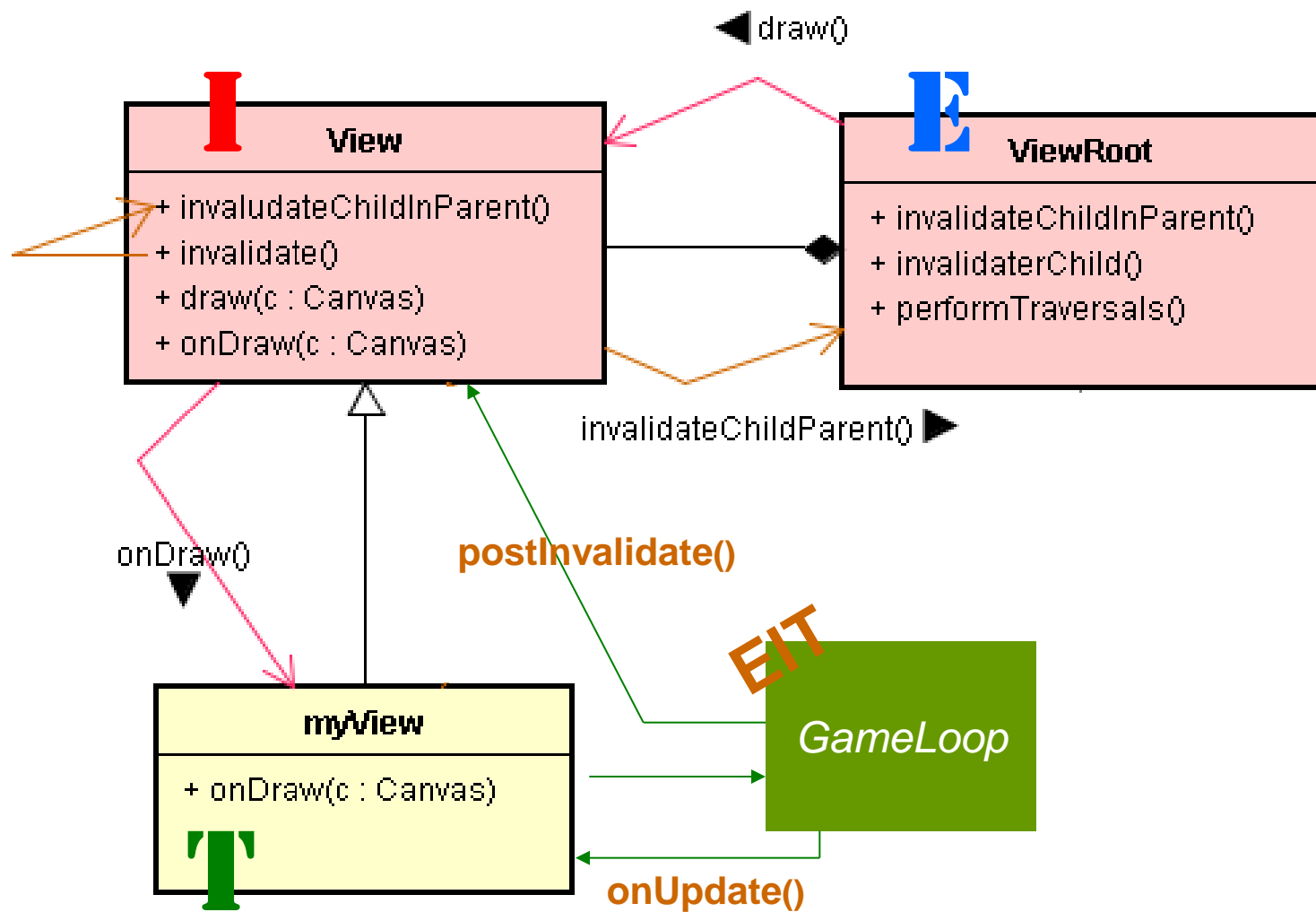
提示

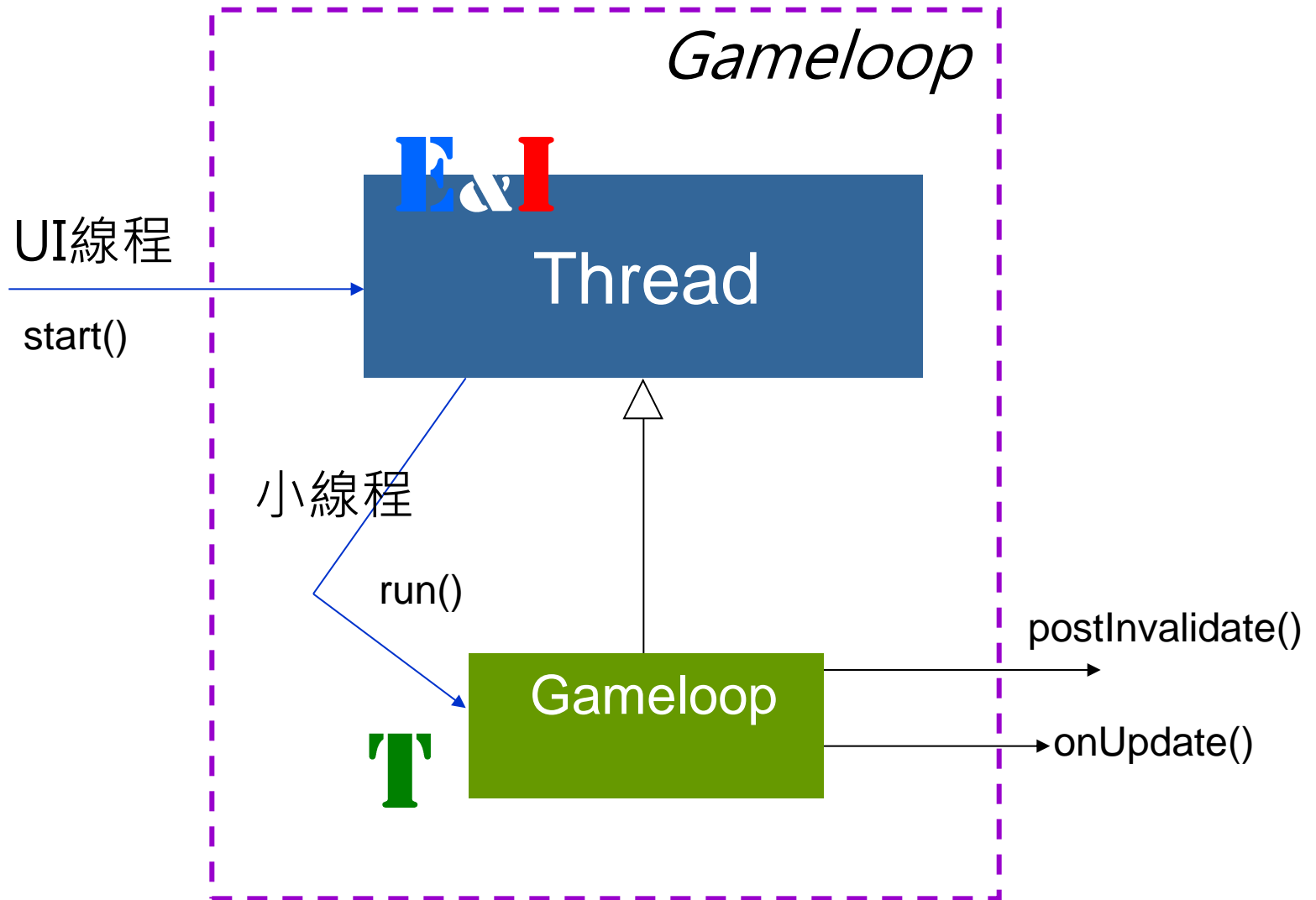
- 其中，`invalidate()`函数会处理到UI。
- 可是，UI是单线程环境。

問題

A2.7-thread

- 在上题里，UI线程执行<T>插件时，直接调用基类的invalidate()函数。
- 现在设计一个新类别：GameLoop类，内含一个EIT造形。如下图所示。





此GameLoop类别的代码如下：

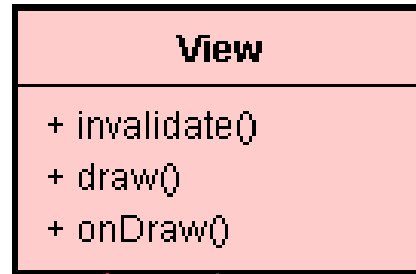
```
public class GameLoop extends Thread {  
    myView mView;  
  
    GameLoop( myView v )  
        { mView = v; }  
    public void run() {  
        mView.onUpdate();  
        mView.postInvalidateDelayed(1000);  
    }  
}
```

- 于是，小线程可调用View类别里的 `postInvalidate()` 函数来请求UI线程转而调用View类别的 `invalidate()` 函数。如下图所示。
- 请问：小线程执行 `postInvalidate()` 函数时，如何与UI线程通信呢？透过Message Queue吗？

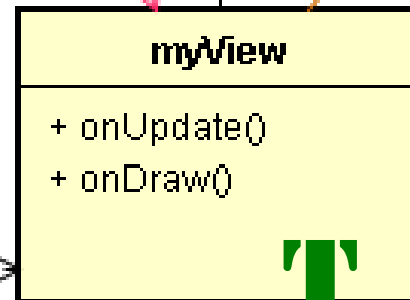
E&I



E&I



2. onDraw()



1.2 start()



1.2.2 postInvalidate()

1.2.1 onUpdate()

<<1. new>>

<<1.1 new>>

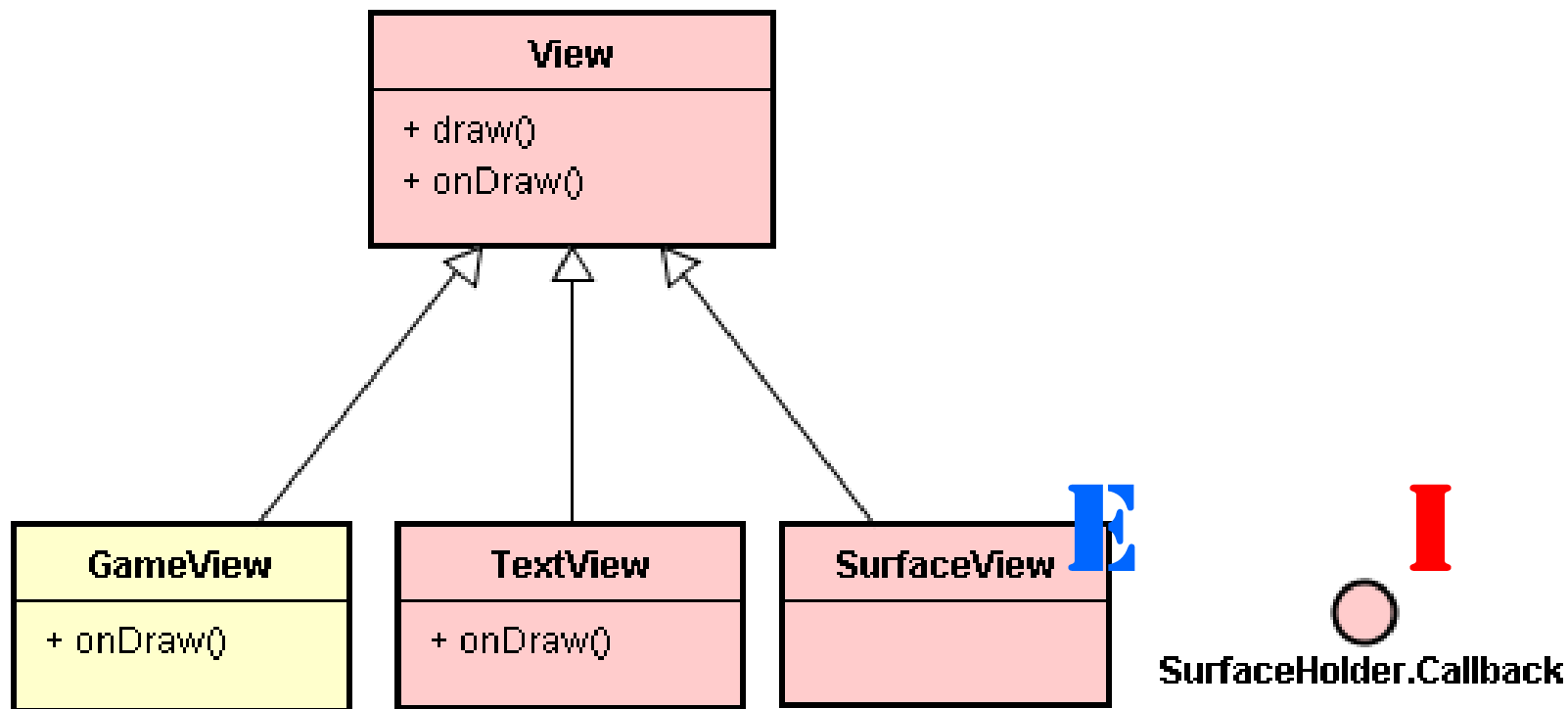
相关问题

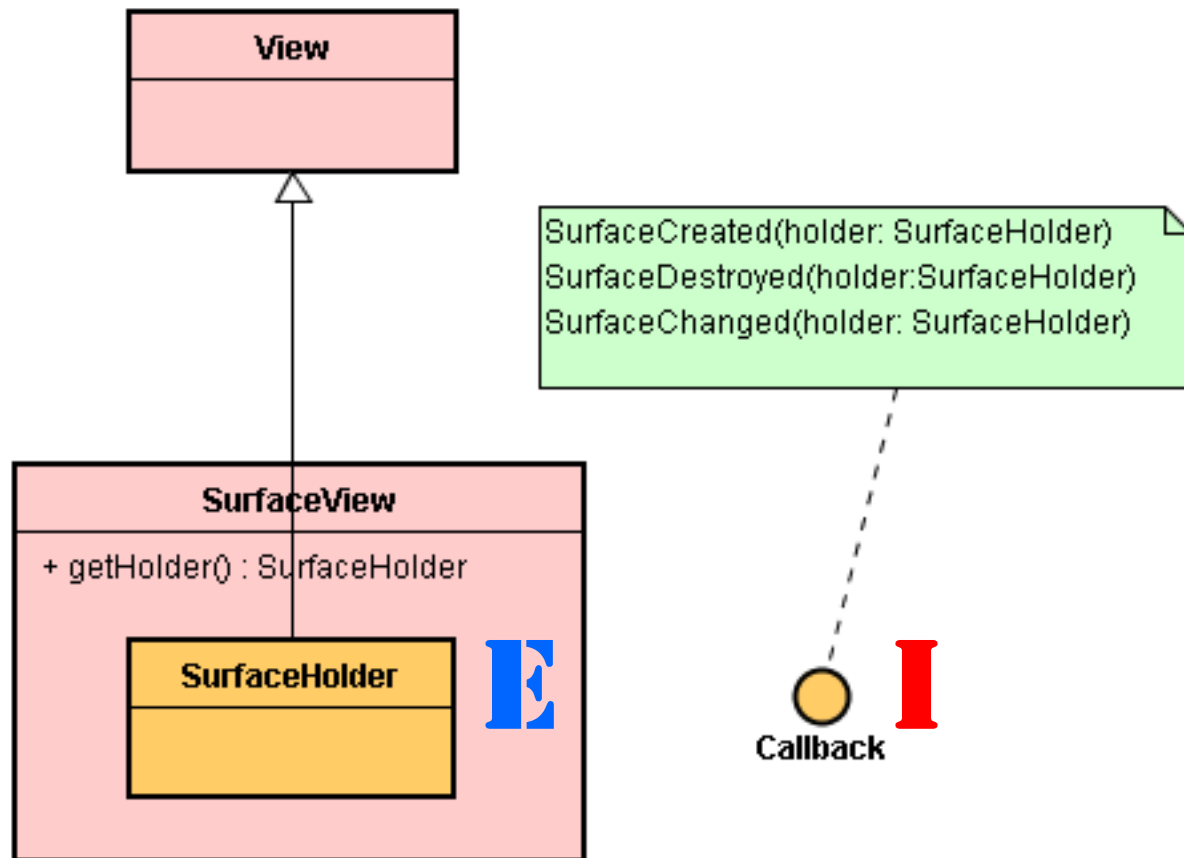
- 请问：这项架构是由几个EIT造形组成的呢？

問題

A2.8-thread

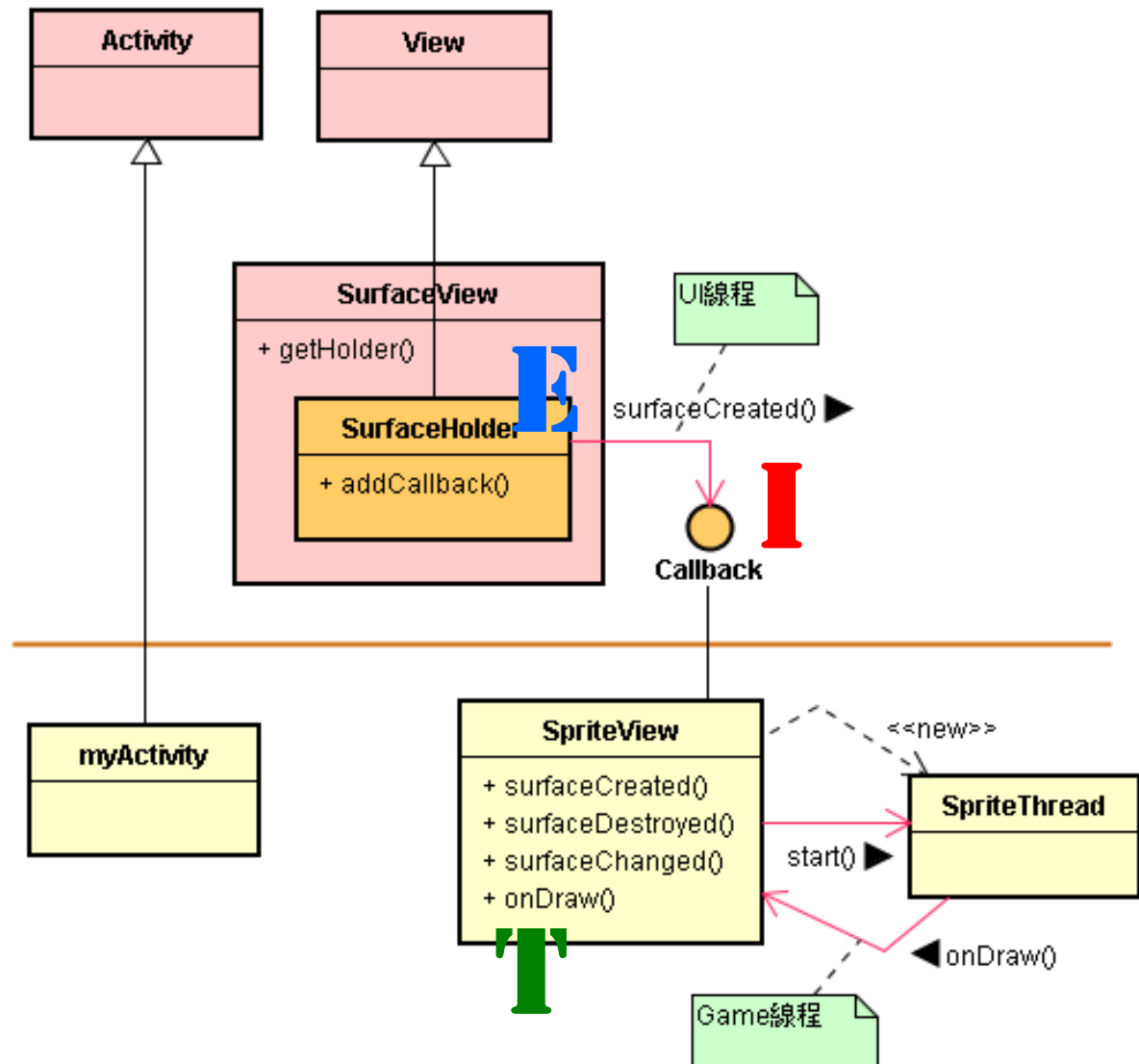
- 在View类别体系里的SurfaceView比较特别，小线程可以透过它来绘制UI上的图像。如下图所示。
- 请问：如何创建小线程来绘制SurfaceView里的图像呢？





提示

- 基于上图里的<E>和<I>来撰写其插件<T>。如下图所示。
- 依循上一题里的Gameloop类别写法，撰写一个SpriteThread(内含一个EIT造形)类别来创建一个小线程。



相关问题

- 在插件<T>里，该如何取得SurfaceView里的画布(Surface)呢？

```
public void surfaceCreated(SurfaceHolder holder) {  
    mHolder = holder;  
    Canvas cavans = mHolder.lockCanvas();  
    // .....
```



~ Continued ~