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

A09_a

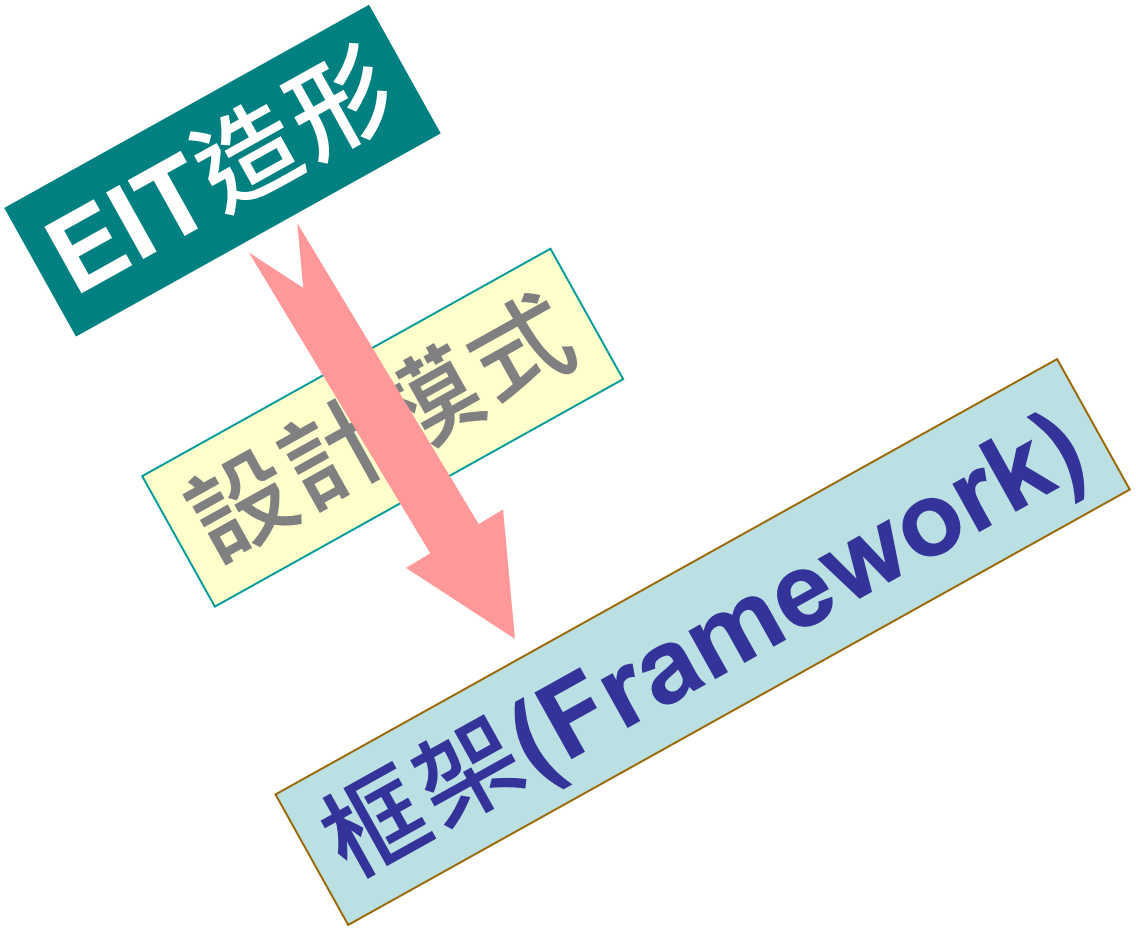
认识框架(Framework) (a)

By 高煥堂

内容

1. <E&I>是框架的核心要素
2. 框架是EIT造形的组合

1、<E&I>是框架的核心要素


EIT造形
設計模式
框架(Framework)

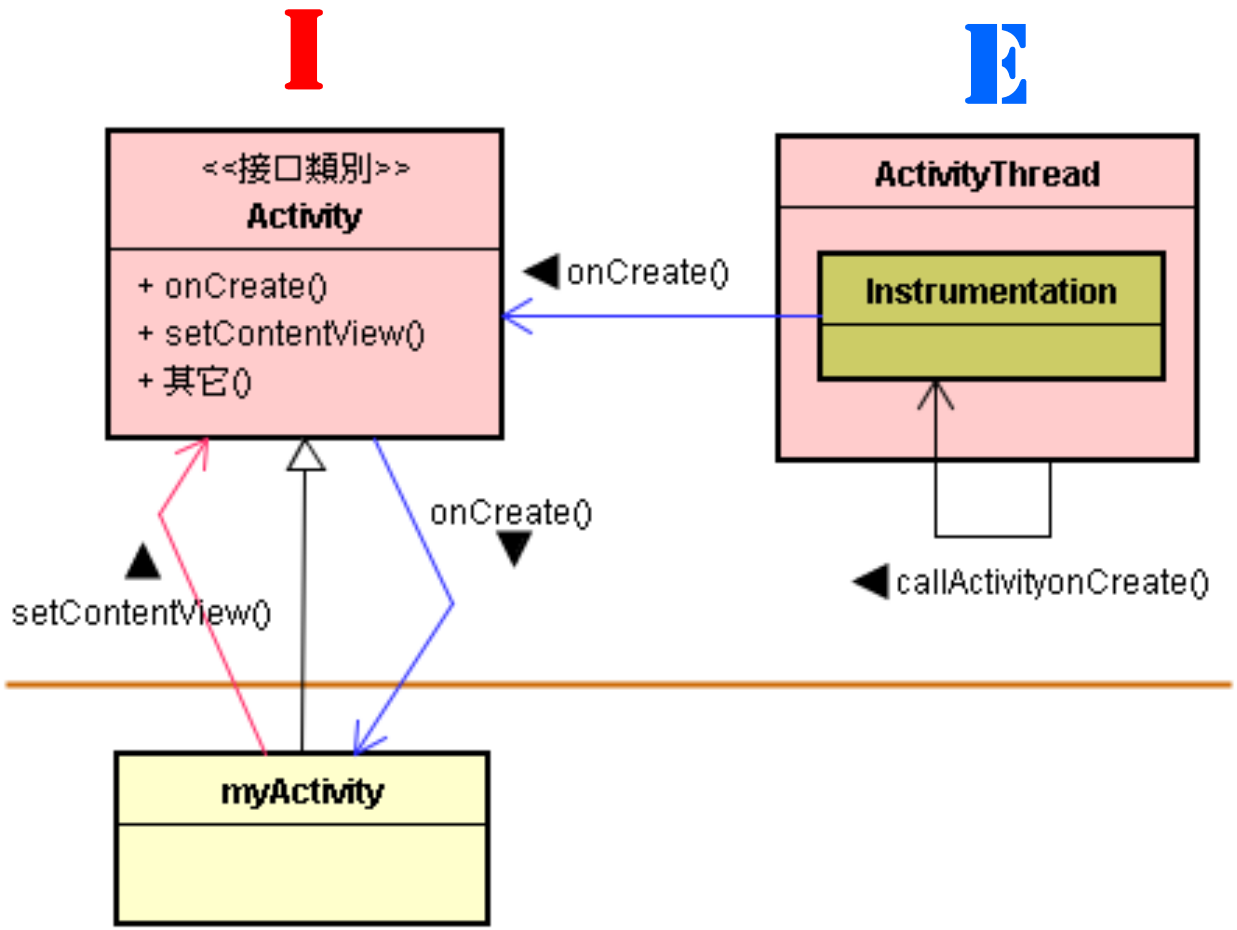
- 在特定领域(Domain)里，将EIT造形的<E&I>部份有意义地组合起来，就成为框架(Framework)了。
- 基本的分工模式：

  -- 强龙定义<I>，并开发<E>

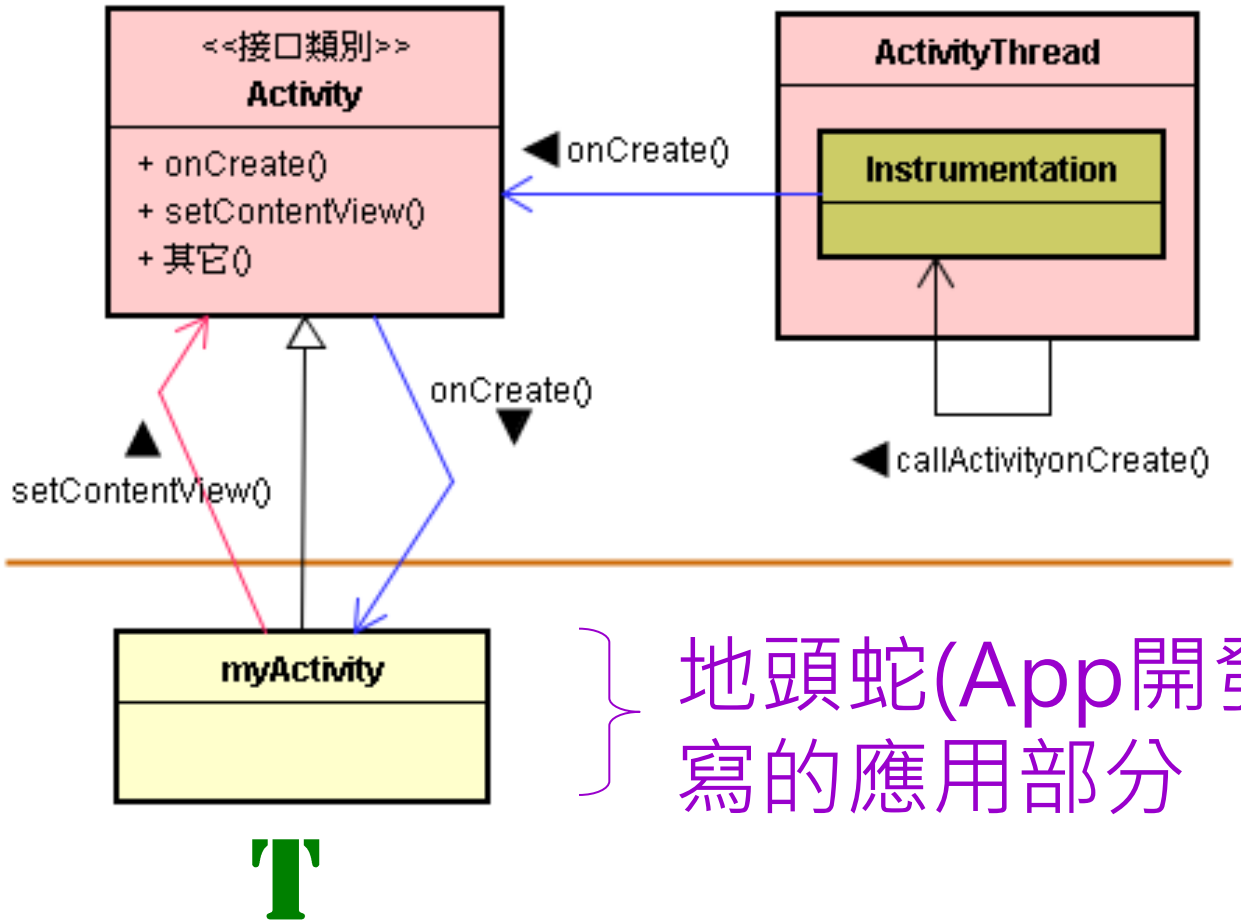
  -- 地头蛇开发<T>

- <E&I>组合起来成为框架本身
- <T>组合起来成为框架的应用(Application)
- 强龙做框架；地头蛇做应用
- 以Android的Java层应用框架为例，在Android框架里，处处可见EIT造形；其<E&I>部分就是框架的核心要素。如下述范例：

範例(一)

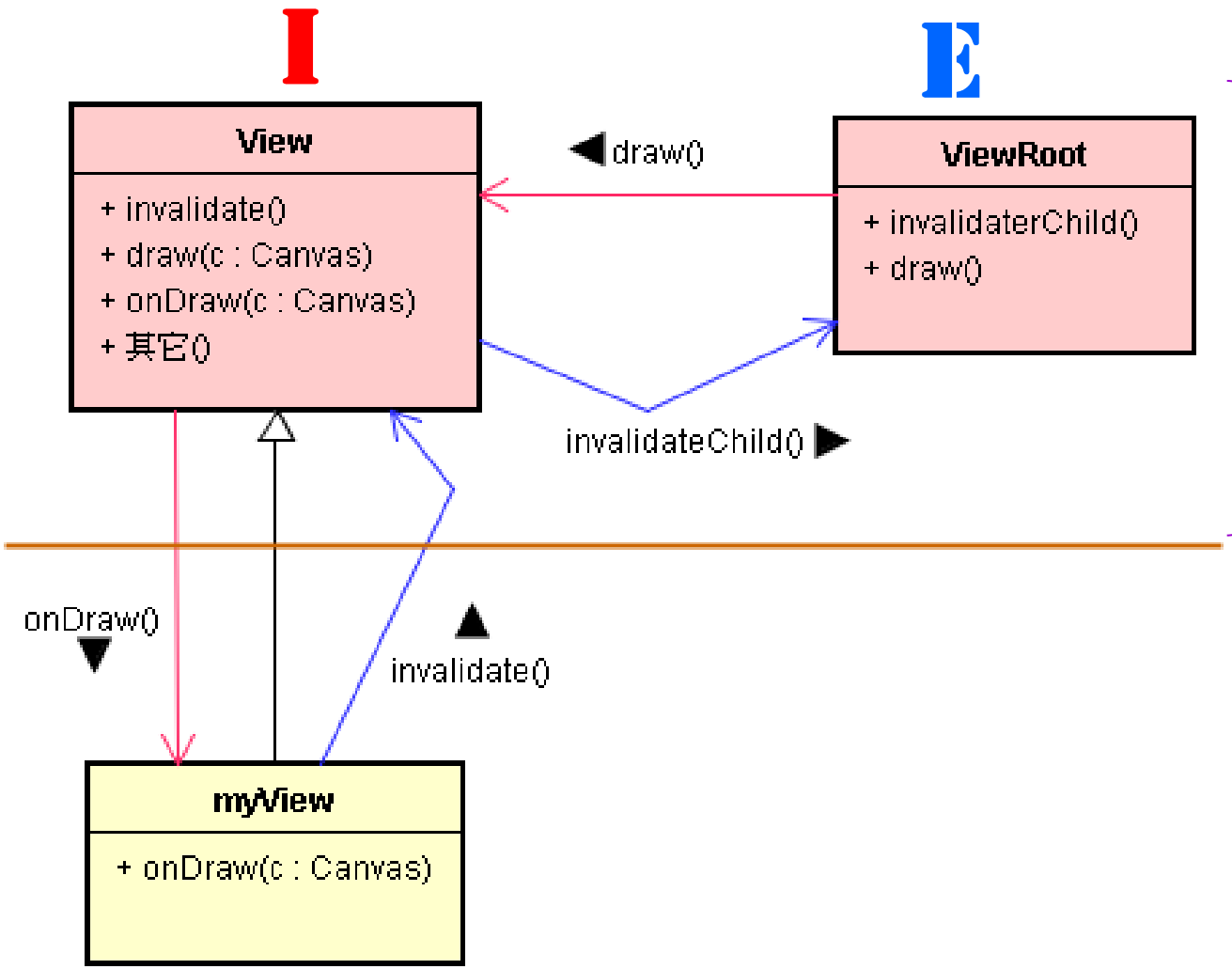


Google
強龍寫的
框架部分

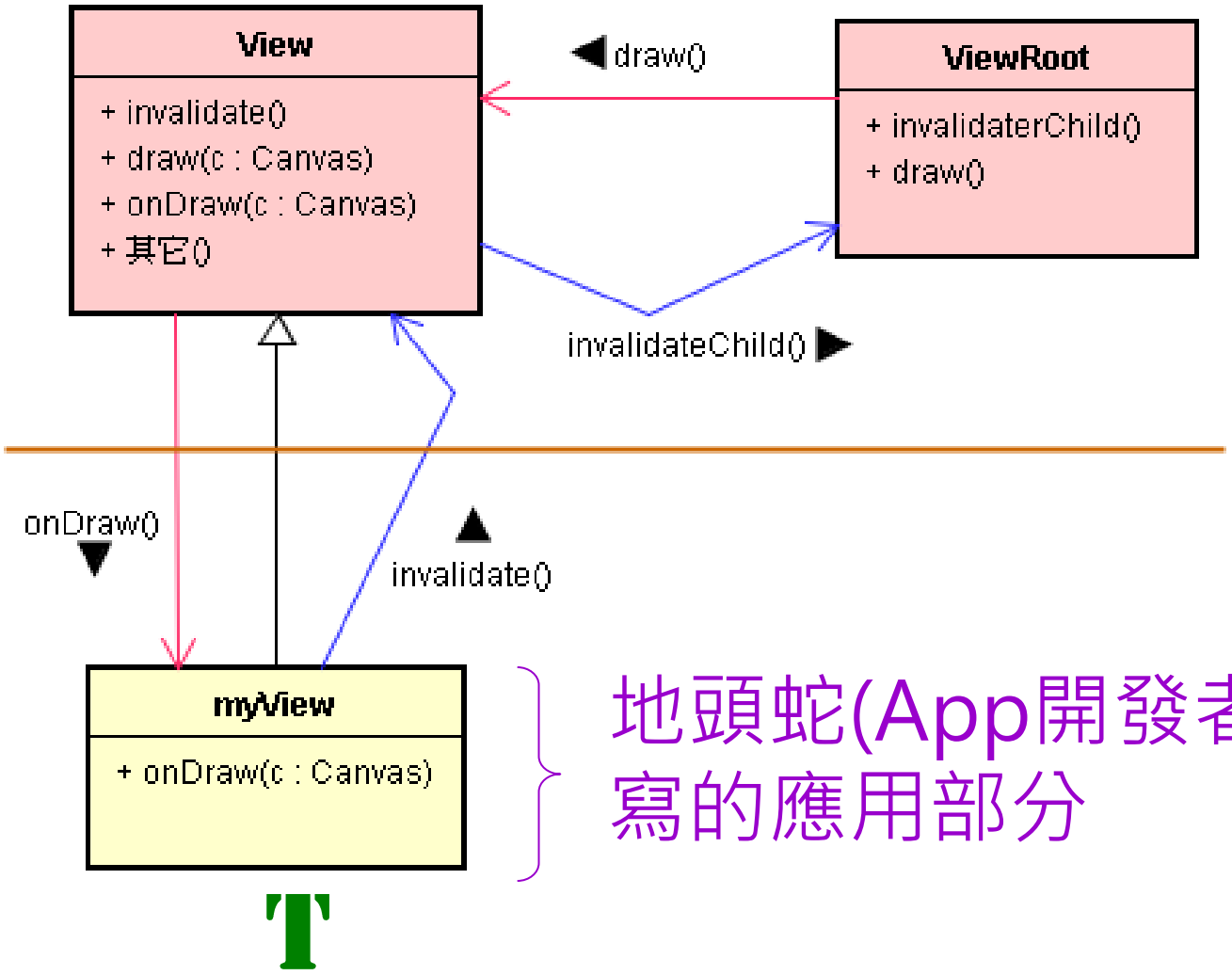


地頭蛇(App開發者)
寫的應用部分

範例(二)



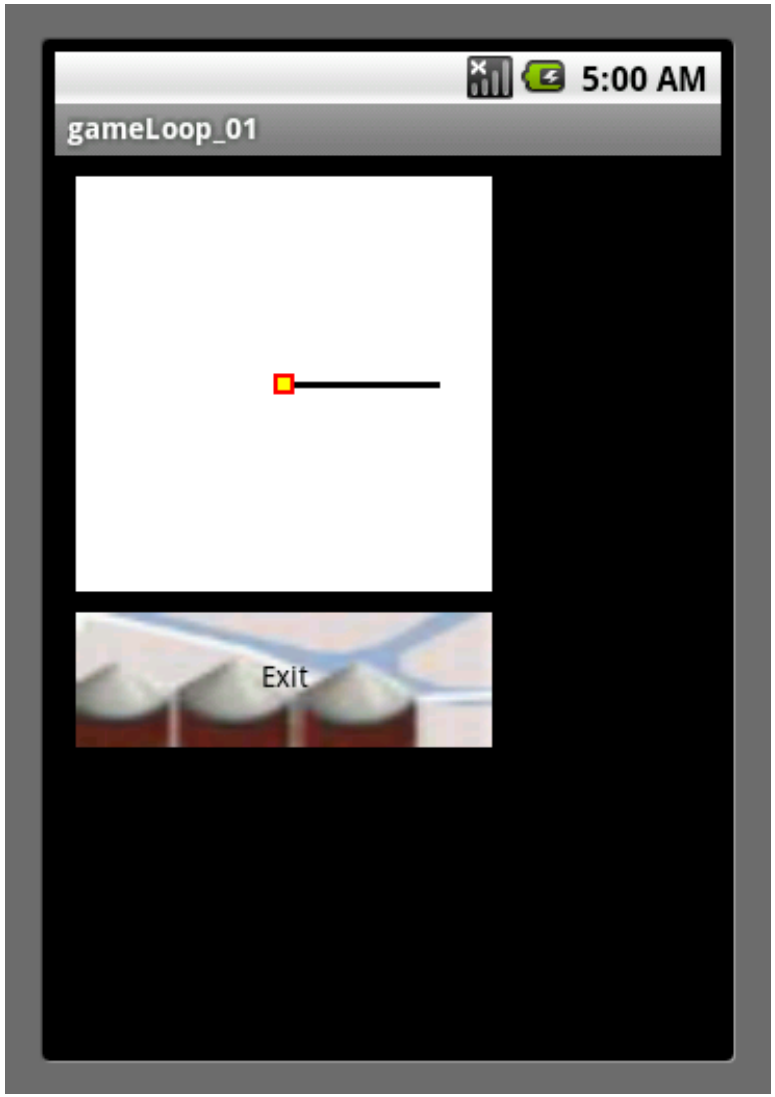
Google
強龍寫的
框架部分



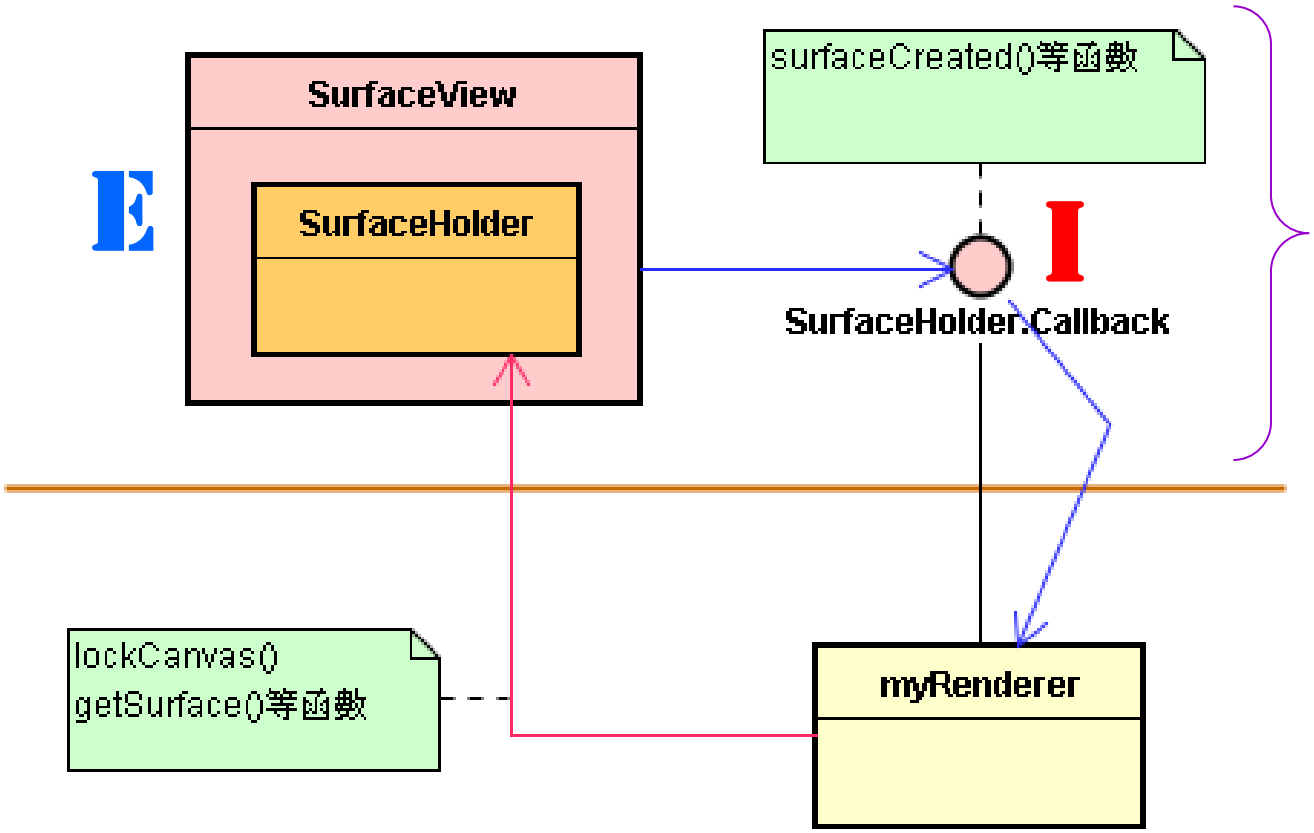
地頭蛇(App開發者)
寫的應用部分

```
// myView.java
//…….

public class myView extends View {
        private Paint paint= new Paint();
        private int line_x = 100, line_y = 100;
        private float count = 0;

      myView(Context ctx) {  super(ctx);  }
      @Override protected void onDraw(Canvas canvas) {
            super.onDraw(canvas);
         if( count > 12)
               count = 0;
         int x = (int) (75.0 * Math.cos(2*Math.PI * count/12.0));
         int y = (int) (75.0 * Math.sin(2*Math.PI * count/12.0));
         count++;
```

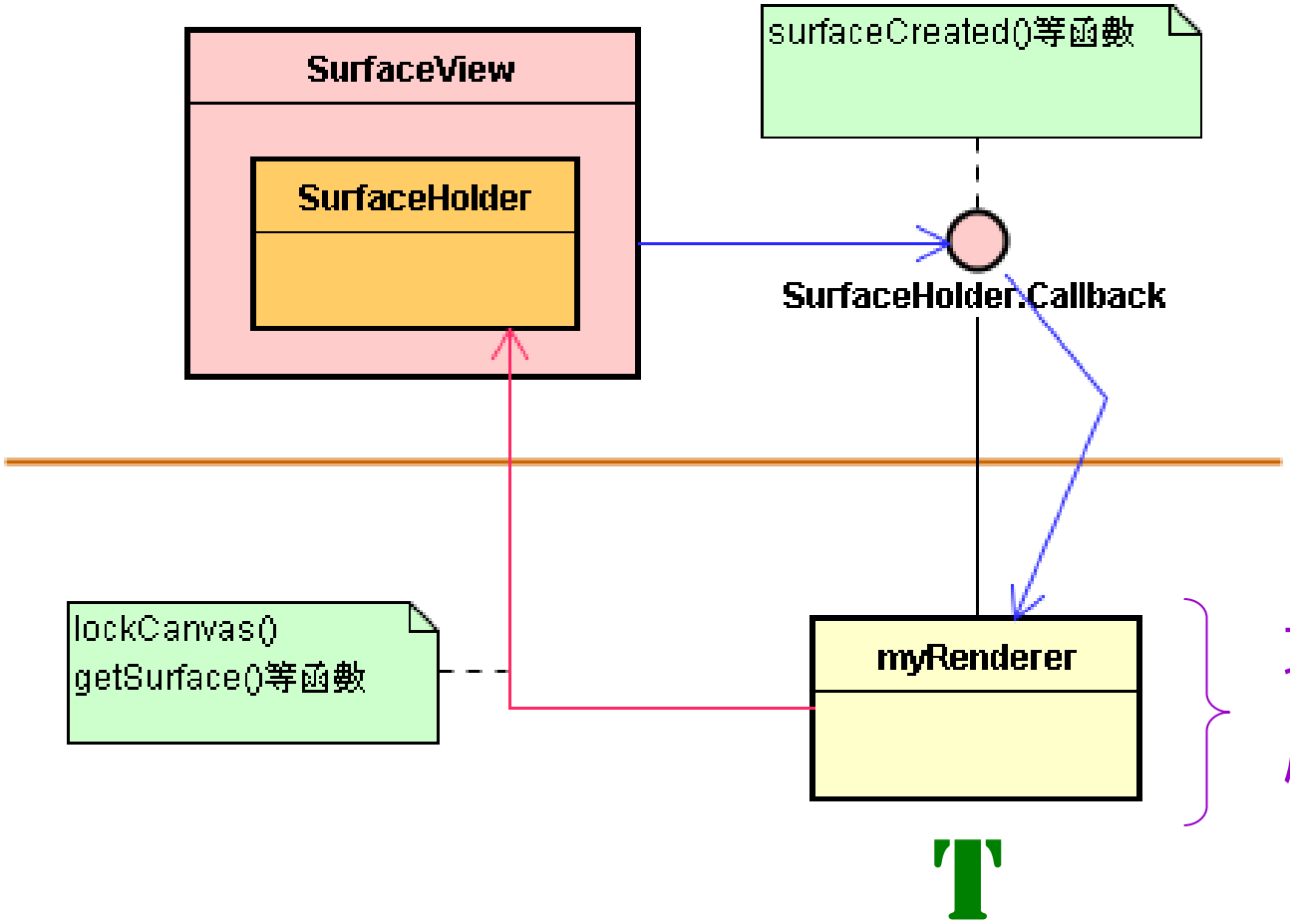
```
            canvas.drawColor(Color.WHITE);
            paint.setColor(Color.BLACK);
            paint.setStrokeWidth(3);
            canvas.drawLine(line_x, line_y, line_x+x, line_y+y, paint);
            paint.setStrokeWidth(2);
            paint.setColor(Color.RED);
            canvas.drawRect(line_x-5, line_y - 5,
                                    line_x+5, line_y + 5, paint);
            paint.setColor(Color.YELLOW);
            canvas.drawRect(line_x-3, line_y - 3, line_x+3,
                                    line_y + 3, paint);
        }
}
```


5:00 AM
gameLoop_01
Exit

範例(三)



Google
強龍寫的
框架部分


SurfaceView
SurfaceHolder
surfaceCreated()等函數
SurfaceHolder.Callback
lockCanvas()
getSurface()等函數
myRenderer

地頭蛇寫的
應用部分

```
// myRenderer.java
class myRenderer implements SurfaceHolder.Callback {
      private SurfaceHolder mHolder;
      private DrawThread mThread;

      public void surfaceCreated(SurfaceHolder holder) {
           mHolder = holder;    mThread = new DrawThread();
           mThread.start();
       }
      public void surfaceDestroyed(SurfaceHolder holder) {
           mThread.finish();     mThread = null;
       }
      public void surfaceChanged(SurfaceHolder holder,
                                        int format, int w, int h) { }
```

```
// ----------------------------------------------------------------------
    class DrawThread extends Thread {
        int degree = 36;
        boolean mFinished = false;
        DrawThread() {  super();  }
         @Override public void run() {
            Bitmap bmp =
                BitmapFactory.decodeResource(getResources(),
                        R.drawable.x_xxx);
            Matrix matrix;
            degree = 0;
            while(!mFinished){
                Paint paint = new Paint();
                paint.setColor(Color.CYAN);
                Canvas cavans = mHolder.lockCanvas();
                cavans.drawCircle(80, 80, 45, paint);
```

```
            //------ rotate ----------------------------
            matrix = new Matrix();    matrix.postScale(1.5f, 1.5f);
            matrix.postRotate(degree);
            Bitmap newBmp = Bitmap.createBitmap( bmp, 0, 0,
                        bmp.getWidth(), bmp.getHeight(), matrix, true);
            cavans.drawBitmap(newBmp, 50, 50, paint);
            mHolder.unlockCanvasAndPost(cavans);
            degree += 15;
            try { Thread.sleep(100);
            } catch (Exception e) {}
          }
        }
        void finish() { mFinished = true;  }
    }
}
```

```
// ac01.java
// ……..
public class ac01 extends Activity {
    private SurfaceView sv = null;
    @Override protected void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        sv = new SurfaceView(this);
        myRenderer mr = new myRenderer();
        sv.getHolder().addCallback(mr);

        LinearLayout layout = new LinearLayout(this);
        layout.setOrientation(LinearLayout.VERTICAL);
        LinearLayout.LayoutParams param =
            new LinearLayout.LayoutParams(200, 150);
        param.topMargin = 5;
        layout.addView(sv, param);
        setContentView(layout);
  }}
```

~ Continued ~