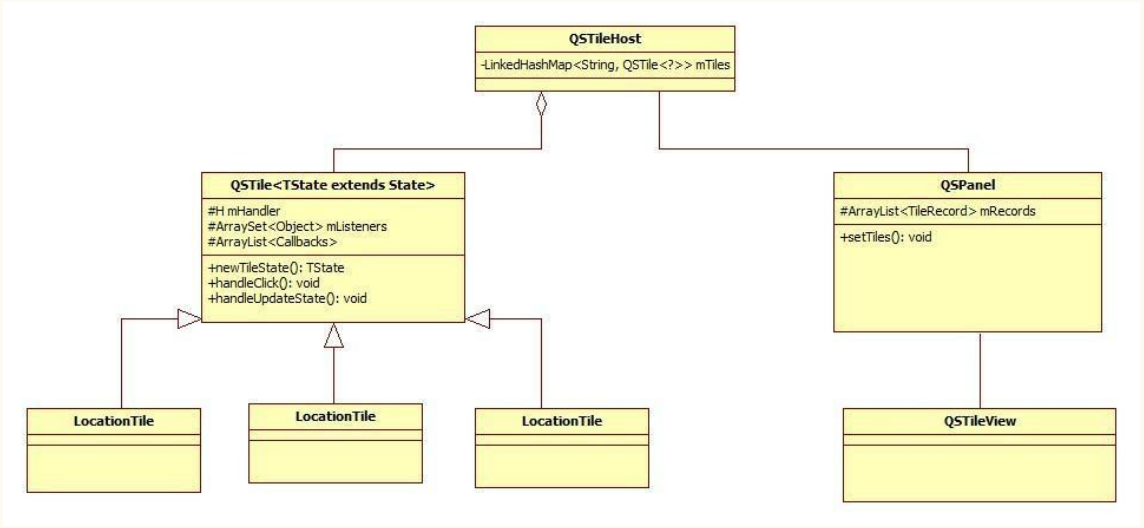
贴上 SystemUI/QS 的 UML 类图：



ClassDiagram1.jpeg

代码设计

抽象出了两个关键的类：QSTile, QSTileView + QSTileHost

QSTile：实际处理各种事件，包括 QSTileView.

```
public void init(OnClickListener click, OnLongClickListener longClick) {

        setClickable(true);

        //click longClick    实现是 QSTile

        setOnClickListener(click);

        setOnLongClickListener(longClick);

}
```

QSTileView:接受 view 的点击时间并交由 QSTileView 做实际的逻辑处理

QSTileHost:就是 host，类似 manager，管理 QSTile 的（如何初始化，初始化多少个具体的 QSTile）。

并为 QSTile 提供了运行环境。

代码实现

QSTile

1.QSTile 是如何处理各种事件的？

答：Thread+Handler

线程环境由 QSTileHost 提供

```
public QSTileHost(...) {
    final HandlerThread ht = new HandlerThread(QSTileHost.class.getSimpleName(), Process.THREAD_PRIORITY_BACKGROUND);
    ht.start();
    //类图中已表明 QSTileHost 和 QSTile 是聚合关系，QSTile 持有 QSTileHost 的引用，通过引用得到 looper，创建 mHandler
    mLooper = ht.getLooper();
}
```

Handler 在 QSTile 中创建，用来处理各种事件

```
mHandler = new H(host.getLooper);
```

所以，我想说的的结论是快速设置所有的逻辑业务都在 mHandler 中。他的类型是 H，继承于 Handler

2.QSTile 既然是处理各种事件的，那么如果某一事件的发生需要改变内容显示，我们如何关联到 QSTileView 呢？

哈哈，就如类图所示。其实，QSPanel，QSTileHost 相互关联，也就实现了 QSPanel 和 QSTile 的相互关联（QSPanel 持有 QSTile 的引用）。而 QSPanel 和 QSTilView 是关联的，所以，你懂的。中间起到桥梁作用的一个关键接口是 Callback

```
public interface Callback {

    void onStateChanged(State state);

    void onShowDetail(boolean show);

    ...

}
```

所以，我想说的结论是，有几种更新界面的方式，你看 Callback 接口里面提供的方法就可以了。

了解以上 2 个问题，基本就可以了。

QSTileView

它是 view，所以各种资源，内容更新就在这，无需多讲

QSTileHost

管理 QSTile 的。很老套了，既然是管理的，那肯定提供了 QSTile 的创建，更新等等了。没什么东西，不多讲了