



上图一共五种交流方式

Controller持有Model和View，都可以直接对他们进行操作，在我们的App中Controller对View定义了一系列接口，Model层也可以针对Controller做接口（如某两个方块是否可以消去，如第几行第几列的方块是哪种type）目前有这么几种情形的数据流动：

1、初始化

初始化的时候应用从Controller启动，初始化Model（如让Model加载各种图片，生成棋盘），之后初始化View，对View使用DataSource的方式来初始化（每个小方块初始化的时候询问DataSource我这个小方块是什么type的（在我们的App里面就是Controller，它要遵循DataSource协议，Controller收到询问之后询问Model这个方块是什么））

2、View接到点击方块的事件

通过View中的Controller实例把事件传递给Controller（View自己判断一下选的是几行几列，给Button加上存储自己是几行几列的属性），Controller判断当前是选了1个块还是2个块，如果是1个块就让View把这个Button变为“选中状态”，在Controller保存下这个块的位置，如果是2个块，如果新选的块就是原来那个选的块，就告诉View把这个Button的选中状态取消。如果是两个不同的块，就去问Model是否选中的两个块可以消去，如果可以，就通知View把这两个块消掉，如果不可以，那就把这两个块的选中状态都取消。此时Model还需要判断是否为死局，如果为死局，Model自己把整个棋盘重置好，并发送一个广播“棋盘重置了”，Controller收到广播之后通知View初始化（重新加载所有Button，仍然使用DataSource方式）

3、重置游戏

View通知Controller，Controller让Model重置，重置成功之后通知View初始化（重新加载所有Button，仍然使用DataSource的方式）

4、Model层发现游戏时间已经结束

广播“游戏时间已经结束”事件，Controller收到广播之后让View弹出“游戏已经结束”的通知，那里面有一个“再玩一次”，如果点击了再玩一次，View层需要通知Controller，然后Controller让Model重置，基本上等同于1

5、Model发现游戏已经结束（消完了）

广播“游戏成功”事件，Controller收到之后基本上等同于4