# Card切换方案

## 切换信息保存

看了一下kitchensink的card切换方案，也是在view层加了一个direction进行控制方向，但是它只考虑了切换card，没有考虑被切换card，iOS中，切换有时候切换card与被切换card相同，有时候又不相同，并不能一并控制，所以在view层把direction写死了并不能很好得控制，目前考虑将direction写在controller层，有动作进行具体切换时，再读取相应切换性息，保证card切换的灵活性。切换时，需传入两个方向值，第一个为切换card的方向，第二个为被切换card的方向，第二个通常情况下可以省略，先确定切换card的方向，然后再根据此得出默认方向，若设置了第二个方向，则不会进行计算。可以在公用部分增加一个反方向映射表，获取相应信息。

Var oppositeDirection = {

“left” : “right”,

“right” : “left”,

“bottom” : “top”,

“top” : “bottom”

}

Function changeCard(tarCardDirection, curCardDirection){

If(!tarCardDirection){

throw(‘error’);

}

curCardDirection = curCardDirection || oppositeDirection[tarCardDirection];

…….

}

## 切换类型

主要还是slider，在多进程程序切换间有用到slider加pop的效果。”left”和”right”两个切换大多为两个card同时切换，很多种情况下，”top”和”bottom”切换往往只运动一个card。同时，切换时还存在一个问题，一般card切换时同时伴随着card的bar切换，而左右切换的时候还伴随着透明度的改变，上下切换的时候并没有透明度的改变。

1. **Slide**
2. “left”，
3. “right”，
4. “bottom”，
5. “top”
6. **Pop**