

reducer

其中：每个action除了负责刷新state（就是我们界面的数据源），还要负责调用parser中的相应的函数，如：

```
case .changecolor(let colorid):  
    inparser.changeTemplateid(input: colorid)  
    return AppState(keyboardout: self.keyboardout, nlpout: self.nlpout, brainmapout: self.brainmapout, pencilout:  
        self.pencilout)
```

- 可见这货收到changecolor的action就调用了parser中修改template的函数数据自然刷新了

reducer

```
case .editinline(let lineNumber, let newstring):  
    // 这里随便写了parser的调用的函数  
    inparser.chageOutline(lineid: lineNumber, pageid: 0, newstring: newstring)  
    inparser.drawSingleSlide()  
    return .inline(lineNumber: lineNumber)
```

- 再比如，这货收到editline的action就调用了parser中changeoutline（修改大纲的函数），和drawSingleSlide（重绘页面函数）