

粉粉项目天气功能Code Review

nate

why

- 了解团队成员工作内容
- 发现潜在问题，减少bug
- 吸收好的代码，丰富自己

how

- 抽取项目中某一功能模块
- 作者先需求分析，出设计方案
- 其他同事思考自己的方案
- 讨论，优化方案和实现细节

参考依据

- 常规
- 安全
- 文档

常规

- 代码能够工作么？它有没有实现预期的功能，逻辑是否正确等。
- 所有的代码是否简单易懂？
- 代码符合你所遵循的编程规范么？这通常包括大括号的位置，变量名和函数名，行的长度，缩进，格式和注释。
- 是否存在多余的或是重复的代码？
- 代码是否尽可能的模块化了？
- 是否有可以被替换的全局变量？
- 是否有被注释掉的代码？
- 循环是否设置了长度和正确的终止条件？
- 是否有可以被库函数替代的代码？
- 是否有可以删除的日志或调试代码？

安全

- 所有的数据输入是否都进行了检查（检测正确的类型，长度，格式和范围）并且进行了编码？
- 在哪里使用了第三方工具，返回的错误是否被捕获？
- 输出的值是否进行了检查并且编码？
- 无效的参数值是否能够处理？

文档

- 是否有注释，并且描述了代码的意图？
- 所有的函数都有注释吗？
- 对非常规行为和边界情况处理是否有描述？
- 第三方库的使用和函数是否有文档？
- 数据结构和计量单位是否进行了解释？
- 是否有未完成的代码？如果是的话，是不是应该移除，或者用合适的标记进行标记比如‘TODO’？

code review不是

审判庭

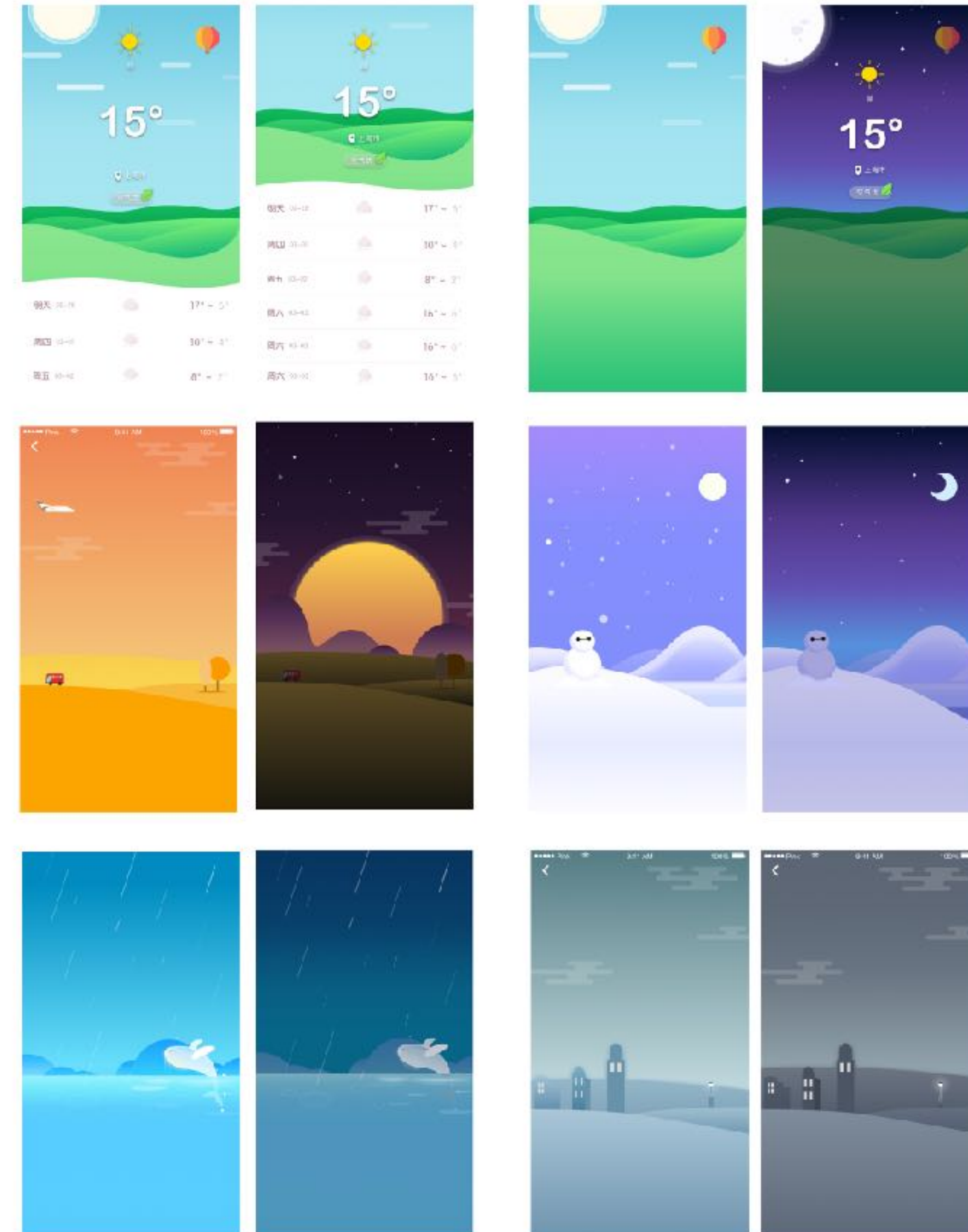
分锅

更不是怼人



列表栏最少显示三天天气（如图1），最多显示6天（如图2）

天气背景共有五组，每组左边是白天，右边是黑夜，分别为晴天，多云，下雪，下雨，雾霾



天气动态元素表格

天气类型	白天动态元素	动态参数	夜晚动态元素	动态参数	备注	元素备注	对应的服务器代号	服务器代号特殊备注
晴天	云	小云 随意飘 共40s	星星	同时最多只可出现 3个 不同程度的星星 ✨ 闪烁 大小：8–16 显示高度：0–0.4	最少一片 最多同片天空出现5片 云可以在背景0–0.45 飞来飞去	具体每个元素 的高度大小参 数 请参考标注 图	“晴”：“00”	"冻雨"：“19”，短粗雨 为粗细为3px长度为普 通雨的1/3 有时间优化
	热气球	从左往右飘 共35s	同左		上下运动 按背景比例 0.1-0.3 时间长18s 距离右边55			
	太阳	光晕	月亮	光晕				
	飞机	从左往右开 共25s	同左		从左往右飞 按背景比例 0.1-0.4 时间长24+3s 飞机航班长度：-250+375+48 / 375+48			
多云	大云	从左往右飘 共35s	同左		云1 云3：156:40 云可以在背景0.1-0.5 随意飞		01/20/29-31 “多云”：“01”，"沙尘 暴"：“20" / "浮尘"：“29", "扬 沙"：“30", "强沙尘暴"：“31",	"无"：“99"
	飞机	从左往右开 共25s	星星	不闪				
	小汽车	从左往右开 共15s	同左					
雪天	雪	下雪范围高度0-0.85	同左	4s+0/1 宽度 / 长度相同：3—16 透明度0.3-0.9		下雪的大小根据 具体天气，雪花 的大小和密集度 会变化	13-17/雪的等级 "阵雪"：“13", "小雪"：“14", "中 雪"：“15", "大雪"：“16", "暴 雪"：“17", 26–28过渡雪 "小到中雪"：“26", "中到大 雪"：“27", "大到暴 雪"：“28",	
	雪人	❄	同左					
	太阳	光晕	月亮	光晕				
			星星	3个 不同程度的闪烁	特大雪没有星星			
雨天	雨	下雨范围高度0-0.85	同左	1.2s+0/1 速度 宽度：1随机 长度：40–55px		下雨的大小根据 具体天气，雨水 的长短和密集度 会变化，特例： 冻雨是普通雨水的 1/3长，5x粗细	3-12不同程度的雨 "阵雨"：“03", "雷阵雨"：“04", "雷 阵雨伴有冰"：“05", "雨夹 雪"：“06", "小雨"：“07", "中雨"：“08", "大 雨"：“09", "暴雨"：“10", "大 暴雨"：“11", "特大暴 雨"：“12", / 21–25过度雨 "小到中雨"：“21", "中到大 雨"：“22", "大到暴 雨"：“23", "暴雨到大暴 雨"：“24", "大暴雨到特 大"：“25",	
雾霾	大云	从左往右飘 共40s	同左				“阴天”：“02”， "雾"：“18" "霾"：“53"	



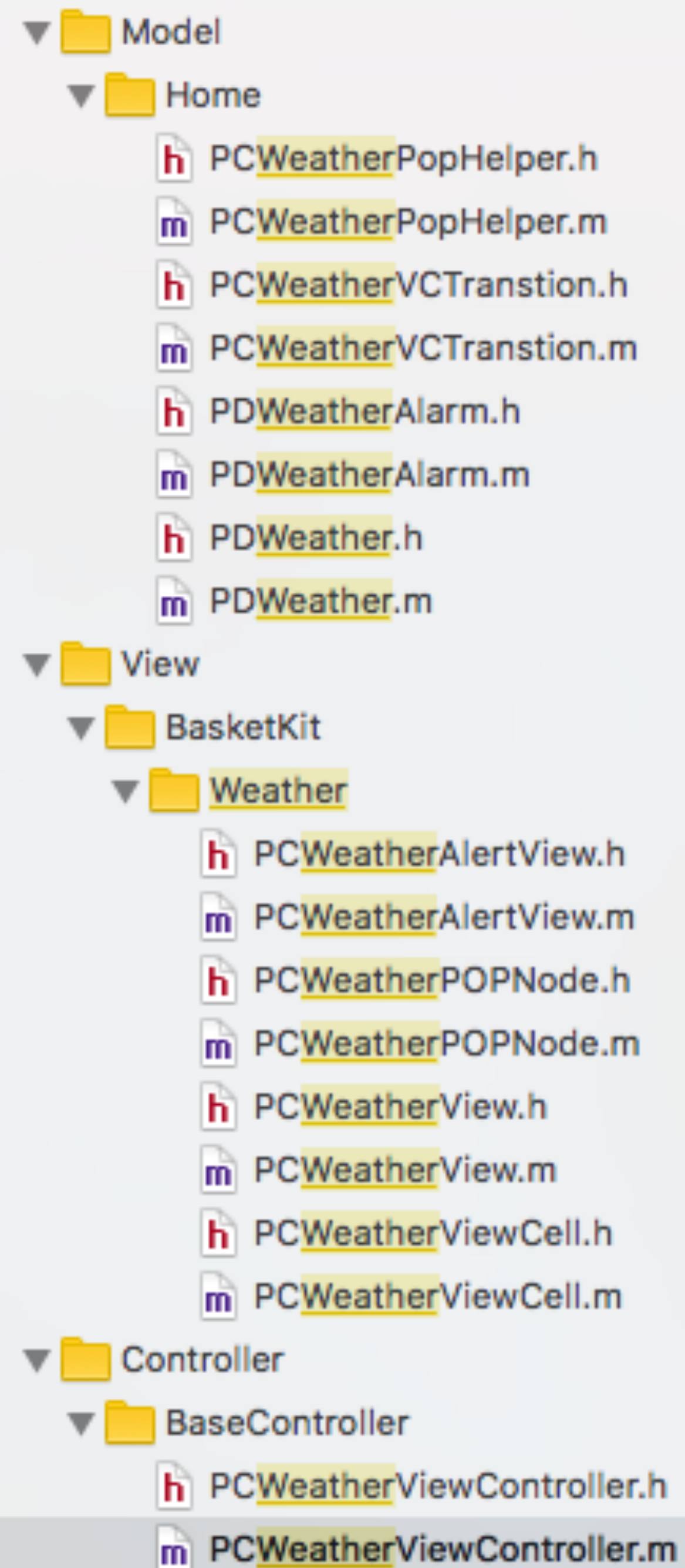
粉粉日记
我的专属贴心闺蜜

00:00

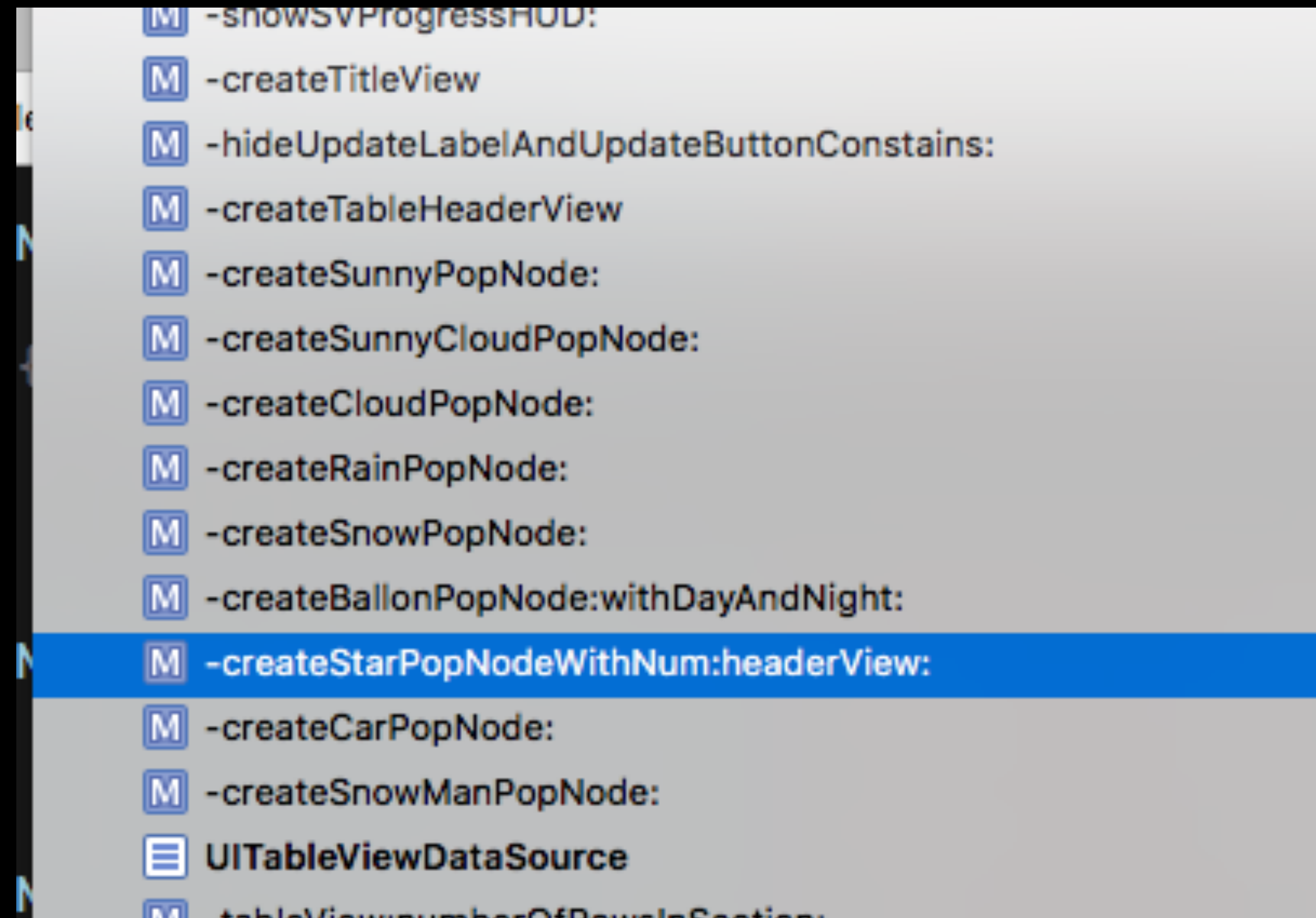
结束录制

代码片段分析

一个VC，其他视图通过subview添加



统一的view添加



```
-(void)createSunnyPopNode:(UIView *)headerView{
    PCSunnyCloudPOPNode *cloudNode = [[PCSunnyCloudPOPNode alloc] initWithFrame:CGRectMake(0, 0, headerView.width, headerView.height*0.45)];
    [headerView addSubview:cloudNode];

    PCPlanePOPNode *planeNode = [[PCPlanePOPNode alloc] initWithFrame:CGRectMake(0, headerView.height*0.1, headerView.width, headerView.
        height*0.5)];
    [headerView addSubview:planeNode];
}
```

头文件梳理

```
37
38 #import "PCWeatherViewController.h"
39 #import "PDProvinceController.h"
40
41 #import "PCWeatherView.h"
42 #import "PCWeatherViewCell.h"
43 #import "PCWeatherPOPNode.h"
44 #import "PCShareDiaryView.h"
45 #import "PCWeatherAlertView.h"
46
47 #import "PDWeather.h"
48 #import "PDForecast.h"
49 #import "APISession+Weather.h"
50 #import "PCWeatherVCTransition.h"
51
52 #import <ReactiveCocoa.h>
53 #import <YYModel.h>
54 #import <Masonry.h>
55 #import <PINImageView+PINRemoteImage.h>
56 #import <PINRemoteImageMacros.h>
57
```


无意义命名

```
58 #define waveHeight 7
59
60 @interface PCWeatherViewController ()<UITableViewDelegate,
61     CGFloat _offset;
62     CGFloat _waveSpeed;
63     CGFloat _waveRate;
64     UIColor *_waveColor;
65
66     CAShapeLayer *_aShapeLayer;
67     CAShapeLayer *_bShapeLayer;
68 }
69
70 @property (nonatomic, strong) UITableView *tableView;
71 @property (nonatomic, strong) PCWeatherView *weatherView;
72 @property (nonatomic, strong) UIScrollView *titleLabel;
73 @property (nonatomic, strong) UIButton *titleLabelButton;
74 @property (nonatomic, strong) UIView *titleLabelView;
75
76 @end
```

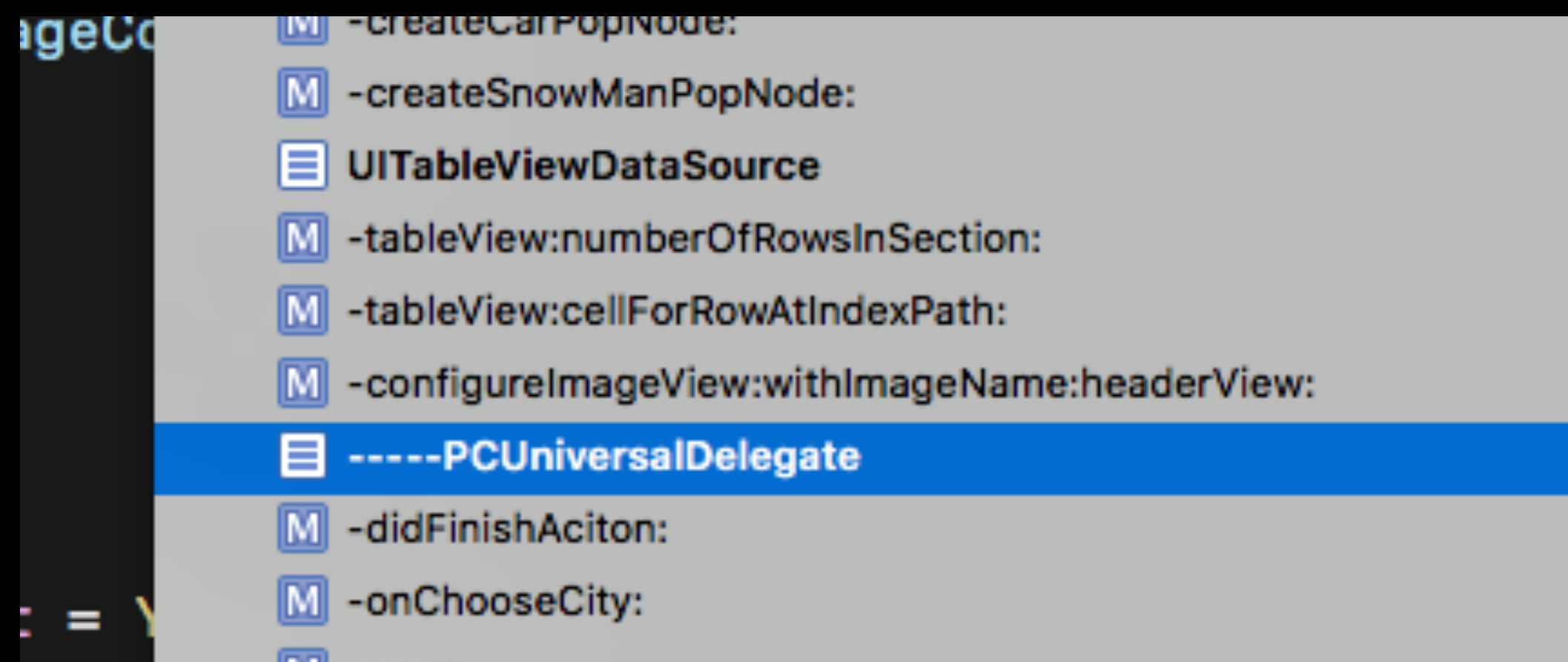

硬编码+硬编码

```
38
39 -(void)viewWillAppear:(BOOL)animated{
40     [super viewWillAppear:animated];
41
42     [[[self.navigationController.navigationBar su
43
44     CGFloat weatherViewHeight = self.weatherView.
45     @weakify(self);
46     [[RACObserve(self.tableView, contentOffset)
47         @strongify(self);
48     CGRect rect = [self.weatherView convertRe
49     if (rect.origin.y < 64+20+20) {
50         if (self.titleView.contentOffset.y == 0
51             [UIView animateWithDuration:0.4 a
52             @strongify(self);
53             self.titleView.contentOffset
54             self.weatherView.alpha = 0;
55         }];
56     }
57     }else{
58         if (self.titleView.contentOffset.y == 4
59             [UIView animateWithDuration:0.4 a
60             @strongify(self);
61             self.weatherView.alpha = 1;
62             self.titleView.contentOffset
63         }];
64     }
65 }
```

```
[titleLabel addSubview:containerView];  
  
UIImageView *weatherImgView = [UIImageView new];  
weatherImgView.tag = 4000;  
UILabel *weatherLabel = [UILabel new];  
weatherLabel.tag = 4001;  
UILabel *temperatureLabel = [UILabel new];  
temperatureLabel.tag = 4002;  
[containerView addSubview:weatherImgView];  
[containerView addSubview:weatherLabel];  
[containerView addSubview:temperatureLabel];
```

```
[self.titleLabel setTitle:self.model.city forState:UIControlStateNormal];  
UIImageView *weatherImgView = [self.titleHolderView viewWithTag:4000];  
UILabel *weatherLabel = [self.titleHolderView viewWithTag:4001];  
UILabel *temperatureLabel = [self.titleHolderView viewWithTag:4002];  
[weatherImgView pin_setImageFromURL:[NSURL URLWithString:self.model.weatherImgurl]];  
weatherLabel.text = self.model.weather;  
temperatureLabel.text = [NSString stringWithFormat:@"%s°",self.model.temp];  
[self.titleLabel mas_updateConstraints:^(MASConstraintMaker *make) {  
    @strongify(self);  
    make.right.equalTo(self.titleLabel.mas_centerX).offset(-8);  
}];
```

合理使用pm分区



```
44 #pragma mark - life cycle
45
46 #pragma mark - private methods
47
48 #pragma mark - event response
49
50 #pragma mark - UITableViewDelegate
51
52 #pragma mark - getters / setters
53
```


for循环操纵i

```
-(void)loopPopFly{
    NSInteger num = arc4random()%5+1;//self.cloudImgArray.count;
    CGFloat longestTime = 40;
    for (int i=0; i<num; i++) {
        NSInteger index = arc4random()%self.cloudImgArray.count;
        if ([self.cloudInScreenArray containsObject:@(index)]) {
            i -=1;
            continue;
        }
        [self.cloudInScreenArray addObject:@(index)];

        BOOL shouldAutoReveal = arc4random()%2;
        BOOL cloudDirectionLeft = arc4random()%2;
        UIImageView *cloudIma = self.cloudImgArray[index];
    }
}
```

代码重复

```
@end
@implementation PCSunnyCloudPOPNode

-(instancetype)initWithFrame:(CGRect)frame{
    if (self = [super initWithFrame:frame]) {
        self.cloudImgArray = [[NSMutableArray alloc] init];
        self.cloudInScreenArray = [[NSMutableArray alloc] init];
        for (int i=0; i<6; i++) {
            UIView *cloudImg = [[UIView alloc] initWithFrame:frame];
            cloudImg.backgroundColor = [UIColor whiteColor];
            cloudImg.layer.cornerRadius = 5;
            [self addSubview:cloudImg];
            [self.cloudImgArray addObject:cloudImg];
        }

        [self loopPopFly];
    }
    return self;
}

-(void)loopPopFly{
    NSInteger num = arc4random()%5+1; //self.cloudImgArray.count;
    CGFloat longestTime = 40;
    for (int i=0; i<num; i++) {
        NSInteger index = arc4random()%self.cloudImgArray.count;
        if ([self.cloudInScreenArray containsObject:@(index)]) {
            i -= 1;
            continue;
        }
        [self.cloudInScreenArray addObject:@(index)];

        BOOL shouldAutoReveal = arc4random()%2;
        BOOL cloudDirectionLeft = arc4random()%2;
        UIImageView *cloudImg = self.cloudImgArray[index];
        cloudImg.y = arc4random()%(int)floor(self.bounds.height);
        cloudImg.width = arc4random()%35+50;
        cloudImg.alpha = (arc4random()%7)*0.1+0.3;
        POPBasicAnimation *basicAnimation = [POPBasicAnimation animation];
        basicAnimation.autoreverses = shouldAutoReveal;
        basicAnimation.timingFunction = [CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionDefault];
        basicAnimation.repeatCount = 1;
        if (cloudDirectionLeft) {
            basicAnimation.fromValue = @(kScreenWidth);
        }
    }
}
```

```
@interface PCCloudPOPNode ()

@property (nonatomic, strong) NSMutableArray<UIImageView*> cloudImgArray;
@property (nonatomic, strong) NSMutableArray<NSNumber*> cloudInScreenArray;

@end
@implementation PCCloudPOPNode

-(instancetype)initWithFrame:(CGRect)frame{
    if (self = [super initWithFrame:frame]) {
        self.cloudImgArray = [[NSMutableArray alloc] init];
        self.cloudInScreenArray = [[NSMutableArray alloc] init];
        for (int i=0; i<9; i++) {
            UIImageView *cloudImg = nil;
            if (i%3==0) {
                cloudImg = [[UIImageView alloc] initWithFrame:frame];
            }else if (i%3==1){
                cloudImg = [[UIImageView alloc] initWithFrame:frame];
            }else if (i%3==2){
                cloudImg = [[UIImageView alloc] initWithFrame:frame];
            }
            //45+5*i%3, 18+i%3*5];
            cloudImg.image = [UIImage imageNamed:@"cloud.png"];
            [self addSubview:cloudImg];
            [self.cloudImgArray addObject:cloudImg];
        }

        [self loopPopFly];
    }
    return self;
}

-(void)loopPopFly{
    NSInteger num = arc4random()%3;
    CGFloat longestTime = 30;
    for (int i=0; i<num; i++) {
        NSInteger index = arc4random()%self.cloudImgArray.count;
        if ([self.cloudInScreenArray containsObject:@(index)]) {
            i -= 1;
            continue;
        }
        [self.cloudInScreenArray addObject:@(index)];

        BOOL shouldAutoReveal = arc4random()%2;
        BOOL cloudDirectionLeft = arc4random()%2;
    }
}
```

调试log

```
26 - (void)onChooseCity:(NSDictionary *)data{  
27     CCLog(@"data:%@",data);  
28     NSString *cityId = data[@"cityid"];  
29     UILabel *label = [self.view viewWithTag:10000];  
30     label.text = cityId;  
31     [self fetchData:cityId];  
32 }
```


循环引用

```
#ifdef DEBUG
UILabel *cityIdLabel = [UILabel new];
cityIdLabel.tag = 10000;
cityIdLabel.text = self.model.cityid;
cityIdLabel.userInteractionEnabled = YES;
cityIdLabel.frame = CGRectMake(50, 100, 200, 100);
[self.view addSubview:cityIdLabel];
UIPanGestureRecognizer *panG = [UIPanGestureRecognizer new];
[[panG rac_gestureSignal] subscribeNext:^(UIPanGestureRecognizer *x) {
    switch (x.state) {
        case UIGestureRecognizerStateBegan:
        {
            break;
        }
        case UIGestureRecognizerStateChanged:
        {
            // [cityIdLabel.layer pop_removeAllAnimations];
            CGPoint translation = [x translationInView:self.view]; //.viewForLastBa
            CGPoint center = cityIdLabel.center;
            center.x +=translation.x;
            center.y +=translation.y;
            cityIdLabel.center = center;

            [x setTranslation:CGPointZero inView:cityIdLabel];
        }
        break;
    }
}];
[cityIdLabel addGestureRecognizer:panG];
#endif
```

破窗理论

一幢有少许破窗的建筑为例，如果那些窗不被修理好，可能将会有破坏者破坏更多的窗户。最终他们甚至会闯入建筑内，如果发现无人居住，也许就在那里定居或者纵火。一面墙，如果出现一涂鸦没有被清洗掉，很快的，墙上就布满了乱七八糟、不堪入目的东西；一条人行道有些许纸屑，不久后就会有更多垃圾，最终人们会视若理所当然地将垃圾顺手丢弃在地上。这个现象，就是犯罪心理学中的破窗效应

不要留着“破窗户”（低劣的设计、糟糕的代码）

“任何一个傻瓜都能写出计算机可以理解的代码。惟有写出人类容易理解的代码，才是优秀的程序员。”

–Martin Fowler

谢谢