###### //1.模拟圈圈动画,自身旋转

CABasicAnimation \*animation2 = [CABasicAnimation animationWithKeyPath:@"transform.rotation.z"];

//默认是顺时针效果，若将fromValue和toValue的值互换，则为逆时针效果

animation2.fromValue = [NSNumber numberWithFloat:0.f];

animation2.toValue = [NSNumber numberWithFloat: M\_PI \*2];

animation2.duration = 3;

animation2.autoreverses = NO;

animation2.fillMode =kCAFillModeForwards;

animation2.repeatCount = HUGE\_VALF;

[self.buttonTransform.layer addAnimation:animation2 forKey:nil];

###### //2.模拟船的动画,位置来回走

CAKeyframeAnimation \* animation = [CAKeyframeAnimation animationWithKeyPath:@"transform.translation.y"];

animation.values=@[@100,@200,@100];

animation.duration = 3;

animation.repeatCount = HUGE\_VALF;

animation.timingFunction = [CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionLinear];

animation.autoreverses=NO;

[animation setCalculationMode:kCAAnimationPaced];

[self.buttonPath.layer addAnimation:animation forKey:@"kViewShakerAnimationKey"];

//3.创建一个可变路径

CGMutablePathRef circleKeyframePath = CGPathCreateMutable();

//创建用于转移坐标的Transform，这样我们不用按照实际显示做坐标计算，以这个坐标做基准点。坐标为下半个弧的中心点

CGAffineTransform circleKeyframeTransform = CGAffineTransformMakeTranslation(300, 260);

CGPathMoveToPoint(circleKeyframePath, &circleKeyframeTransform, 0, 0);

//CGPathAddLineToPoint(circleKeyframePath, &circleKeyframeTransform, -100, 0)

//创建一个1/4弧（圆的左下角弧）

CGPathAddArc(circleKeyframePath, &circleKeyframeTransform, 0, -100, 100, (CGFloat)(0.5 \* M\_PI), (CGFloat)(M\_PI), false);

CGPathAddLineToPoint(circleKeyframePath, &circleKeyframeTransform, -100, -100);

CGPathAddLineToPoint(circleKeyframePath, &circleKeyframeTransform, -50, -100);

//创建一个以半径为50的两条切线的内切圆弧

CGPathAddArcToPoint(circleKeyframePath, &circleKeyframeTransform, 0, -200, 50, -100, 50);

CGPathAddLineToPoint(circleKeyframePath, &circleKeyframeTransform, 50, -100);

CGPathAddLineToPoint(circleKeyframePath, &circleKeyframeTransform, 100, -100);

//CGPathAddLineToPoint(circleKeyframePath, &circleKeyframeTransform, 100, 0)

//创建一个1/4弧(圆的右下角弧)

CGPathAddArc(circleKeyframePath, &circleKeyframeTransform, 0, -100, 100, 0, (CGFloat)(0.5 \* M\_PI), false);

//关闭路径

CGPathCloseSubpath(circleKeyframePath);

CAShapeLayer \*backgroundLayer = [CAShapeLayer layer];

backgroundLayer.path = circleKeyframePath;

backgroundLayer.strokeColor =pchRedColor.CGColor;

backgroundLayer.strokeColor = [UIColor redColor].CGColor;

backgroundLayer.lineWidth = 3;

backgroundLayer.fillColor = pchCyanColor.CGColor;

[self.view.layer addSublayer:backgroundLayer];

//4.path运动

CGMutablePathRef path = CGPathCreateMutable();

//将路径的起点定位到 （50 120）

CGPathMoveToPoint(path,NULL,50.0,120.0);

//下面5行添加5条直线的路径到path中

CGPathAddLineToPoint(path, NULL, 60, 130);

CGPathAddLineToPoint(path, NULL, 70, 140);

CGPathAddLineToPoint(path, NULL, 80, 150);

CGPathAddLineToPoint(path, NULL, 90, 160);

CGPathAddLineToPoint(path, NULL, 100, 170);

//下面四行添加四条曲线路径到path

CGPathAddCurveToPoint(path,NULL,50.0,275.0,150.0,275.0,70.0,120.0);

CGPathAddCurveToPoint(path,NULL,150.0,275.0,250.0,275.0,90.0,120.0);

CGPathAddCurveToPoint(path,NULL,250.0,275.0,350.0,275.0,110.0,120.0);

CGPathAddCurveToPoint(path,NULL,350.0,275.0,450.0,275.0,130.0,120.0);

//以“position”为关键字 创建 实例

CAKeyframeAnimation \*animation = [CAKeyframeAnimation animationWithKeyPath:@"position"];

//设置path属性

[animation setPath:path];

[animation setDuration:3.0];

animation.repeatCount=HUGE;

//这句代码 表示 是否动画回到原位

// [animation setAutoreverses:YES];

CFRelease(path);

[self.buttonPahtValues.layer addAnimation:animation forKey:NULL];

###### //5.鸟飞行,云移动

@property (nonatomic, strong) NSMutableArray \*imageArr;//鸟图片数组

@property (nonatomic, strong) UIImageView \*birdImage;//鸟本体

@property (nonatomic, strong) UIImageView \*sunCloudImage;//晴天云

-(NSMutableArray \*)imageArr {

if (!\_imageArr) {

\_imageArr = [NSMutableArray array];

for (int i = 1; i < 9; i++) {

NSString \*fileName = [NSString stringWithFormat:@"ele\_sunnyBird%d",i];

UIImage \*image = [UIImage imageNamed:fileName];

[\_imageArr addObject:image];

}

}

return \_imageArr;

}

//鸟 本体

-(void)loadBrid{

\_birdImage = [[UIImageView alloc]initWithFrame:CGRectMake(-30, LH\_ScreenHeight \* 0.2, 70, 50)];

[\_birdImage setAnimationImages:self.imageArr];

\_birdImage.animationRepeatCount = 0;

\_birdImage.animationDuration = 1;

[\_birdImage startAnimating];

[self.view addSubview:\_birdImage];

[\_birdImage.layer addAnimation:[self birdFlyAnimationWithToValue:@(LH\_ScreenWidth+30) duration:10 ] forKey:nil];

}

//动画横向移动方法

- (CABasicAnimation \*)birdFlyAnimationWithToValue:(NSNumber \*)toValue duration:(NSInteger)duration{

CABasicAnimation \*animation = [CABasicAnimation animationWithKeyPath:@"transform.translation.x"];

animation.toValue = toValue;

animation.duration = duration;

animation.removedOnCompletion = NO;

animation.repeatCount = MAXFLOAT;

animation.fillMode = kCAFillModeForwards;

return animation;

}

-(void)loadSunCloudImage{

//晴天云

\_sunCloudImage = [[UIImageView alloc]initWithImage:[UIImage imageNamed:@"ele\_sunnyCloud2"]];

CGRect frame = \_sunCloudImage.frame;

frame.size = CGSizeMake(LH\_ScreenHeight \*0.5, LH\_ScreenWidth\*0.2);

\_sunCloudImage.frame = frame;

\_sunCloudImage.center = CGPointMake(LH\_ScreenWidth \* 0.25, LH\_ScreenHeight\*0.5);

[\_sunCloudImage.layer addAnimation:[self birdFlyAnimationWithToValue:@(LH\_ScreenWidth+30) duration:50] forKey:nil];

[self.view addSubview:\_sunCloudImage];

}

###### //6.火车效果

@property (nonatomic, strong) CAShapeLayer \*yellowPath;

//---

[self loadYellowPath];

for (int i=0; i<10;i++) {

[self addYellowCarPathAnimation:(CACurrentMediaTime() + 0.2 \*(float)(i))];

}

//----

-(void)loadYellowPath{

CAShapeLayer \*calayer= [[CAShapeLayer alloc] init];

calayer.backgroundColor = [UIColor redColor].CGColor;

calayer.lineWidth = 5;

calayer.strokeColor=pchColor222.CGColor;//轨道边缘

UIBezierPath \*path=[[UIBezierPath alloc] init];

[path moveToPoint:CGPointMake(0, LH\_ScreenHeight-70)];

[path addCurveToPoint:CGPointMake(LH\_ScreenWidth/1.5, 200) controlPoint1:CGPointMake(LH\_ScreenWidth/6, LH\_ScreenHeight-200) controlPoint2:CGPointMake(LH\_ScreenWidth/2.5, LH\_ScreenHeight+50)];

[path addQuadCurveToPoint:CGPointMake(LH\_ScreenWidth+10, LH\_ScreenHeight/3) controlPoint:CGPointMake(LH\_ScreenWidth-100, 50)];

[path addLineToPoint:CGPointMake(LH\_ScreenWidth+10, LH\_ScreenHeight+10)];

[path addLineToPoint:CGPointMake(0, LH\_ScreenHeight+10)];

calayer.fillColor =[UIColor colorWithPatternImage:[UIImage imageNamed:@"yellow"]].CGColor;

calayer.path=path.CGPath;

//------------------------------------------------------------------------------------------------------------------------------------------------

CAShapeLayer \*lineLayer=[[CAShapeLayer alloc] init];

lineLayer.lineCap = kCALineCapRound;

lineLayer.strokeColor = [UIColor redColor].CGColor;//轨道中心部分

lineLayer.lineWidth = 2;

lineLayer.fillColor=[UIColor clearColor].CGColor;

lineLayer.path=path.CGPath;

lineLayer.lineDashPattern=@[@5,@10];//虚线

[calayer addSublayer:lineLayer];

self.yellowPath= calayer;

[self.view.layer addSublayer:self.yellowPath];

}

-(void) addYellowCarPathAnimation:( CFTimeInterval )beginTime {

CALayer \*carLayer=[[CALayer alloc] init];

carLayer.frame=CGRectMake(0, 0, 17,11);

[carLayer setAffineTransform:CGAffineTransformMakeTranslation(0, -7)];

[carLayer setContents:[UIImage imageNamed:@"car"].CGImage];

CAKeyframeAnimation \*animation=[CAKeyframeAnimation animationWithKeyPath:@"position"];

animation.path = self.yellowPath.path;

animation.timingFunction =[CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionLinear];

animation.duration = 12;

animation.beginTime = beginTime;

animation.repeatCount = MAXFLOAT;

animation.autoreverses = false;

animation.calculationMode = kCAAnimationCubicPaced;

animation.rotationMode = kCAAnimationRotateAuto;

[self.yellowPath addSublayer:carLayer];

[carLayer addAnimation:animation forKey:@"carAnimation"];

}

###### //7.圆圈虚线运动

-(void)loadgradientLayer{

// 贝塞尔曲线(创建一个圆)

UIBezierPath \*path = [UIBezierPath bezierPathWithArcCenter:CGPointMake(100/2.f, 100/2.f) radius:100/2.f startAngle:0 endAngle:M\_PI \* 2 clockwise:YES];

// 创建一个shapeLayer

CAShapeLayer \*layer = [CAShapeLayer layer];

layer.frame=CGRectMake(250, 150, 100, 100);

layer.strokeColor = [UIColor greenColor].CGColor; // 边缘线的颜色

layer.fillColor = configColor\_alpha(100, 100, 1, 0.2).CGColor; // 闭环填充的颜色

layer.lineCap = kCALineCapSquare; // 边缘线的类型

layer.path = path.CGPath; // 从贝塞尔曲线获取到形状

layer.lineWidth = 4.0f; // 线条宽度

layer.lineDashPattern = @[@6, @10];//画虚线

self.circlePath=layer;

[self.view.layer addSublayer:self.circlePath];

CABasicAnimation \*animation2 = [CABasicAnimation animationWithKeyPath:@"transform.rotation.z"];

//默认是顺时针效果，若将fromValue和toValue的值互换，则为逆时针效果

animation2.fromValue = [NSNumber numberWithFloat:0.f];

animation2.toValue = [NSNumber numberWithFloat: M\_PI \*2];

animation2.duration = 10;

animation2.autoreverses = NO;

animation2.fillMode =kCAFillModeForwards;

animation2.repeatCount = HUGE\_VALF;

[layer addAnimation:animation2 forKey:nil];

}