

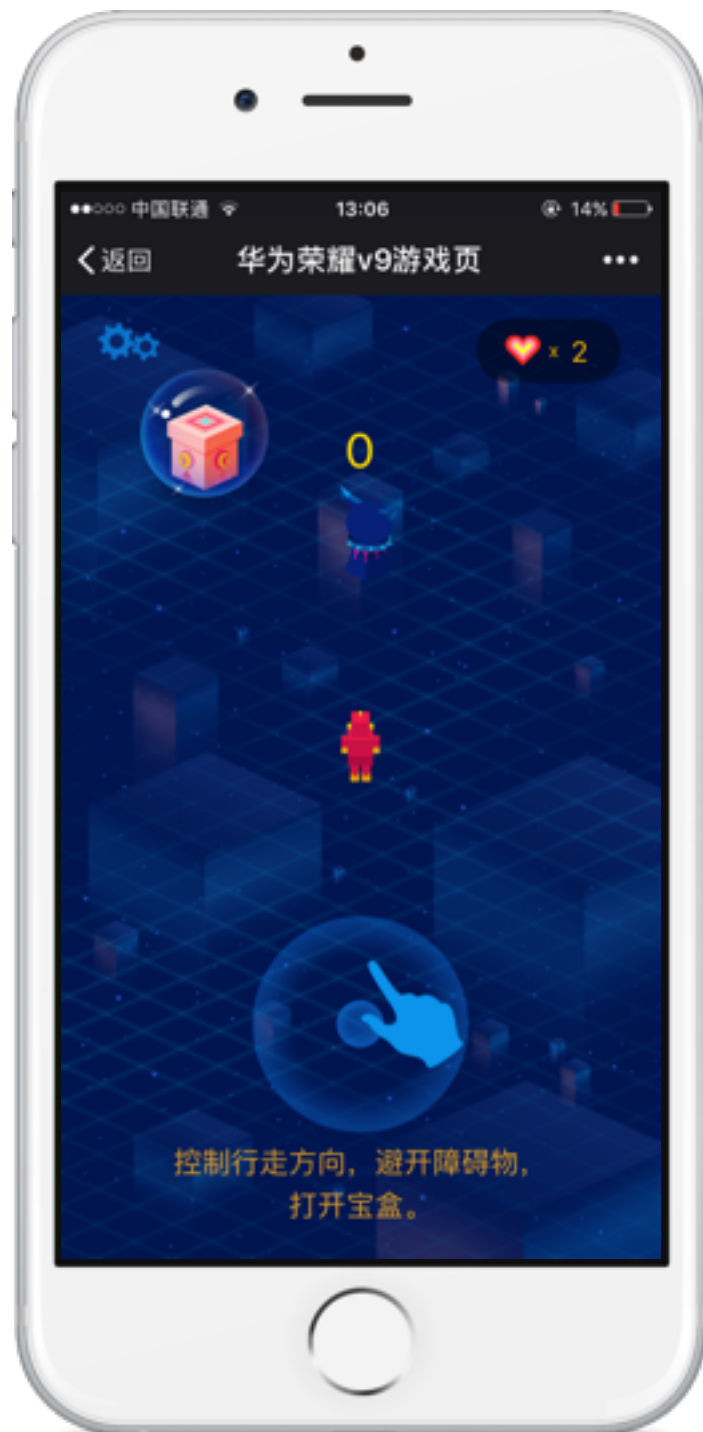
02

OPEN ORIENTED

凹凸实验室

# H5躲避类游戏开发

# 最近做的一些案例

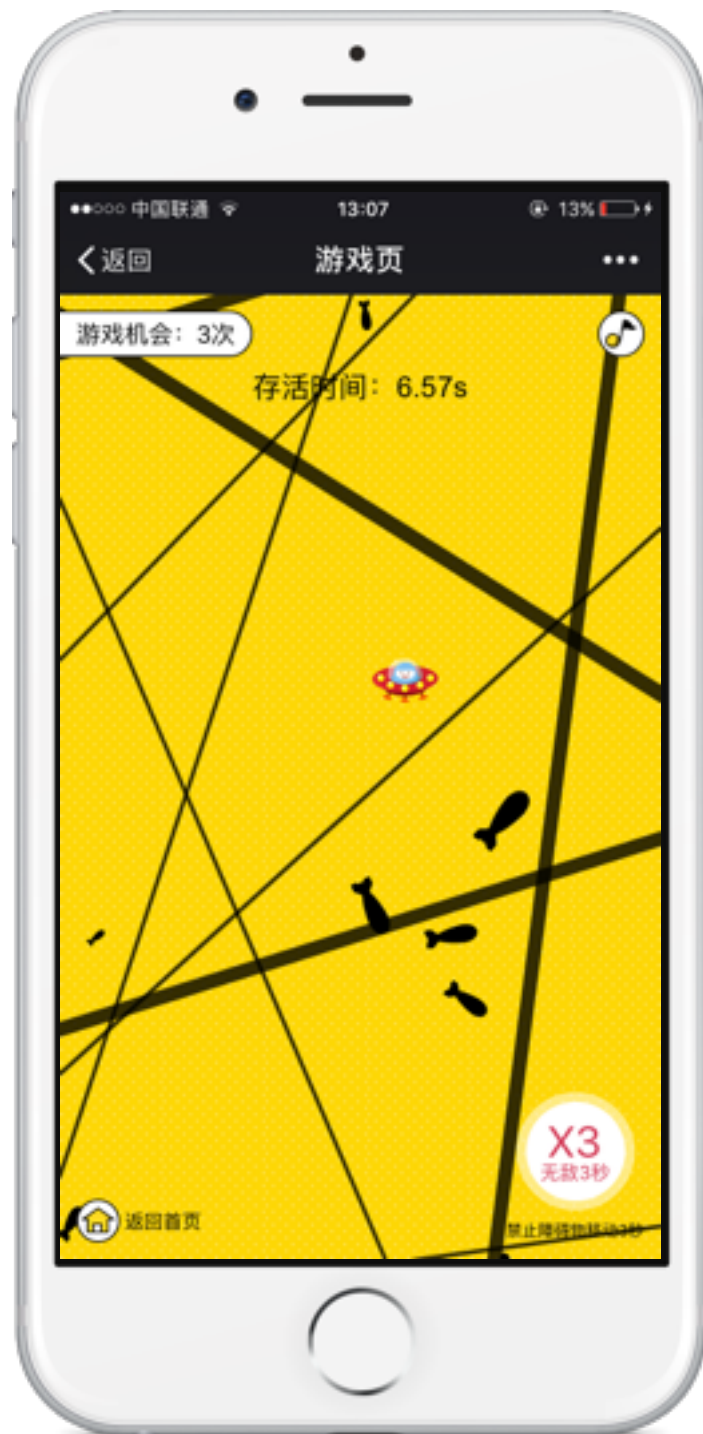


华为荣耀v9首发



# 最近做的一些案例

02

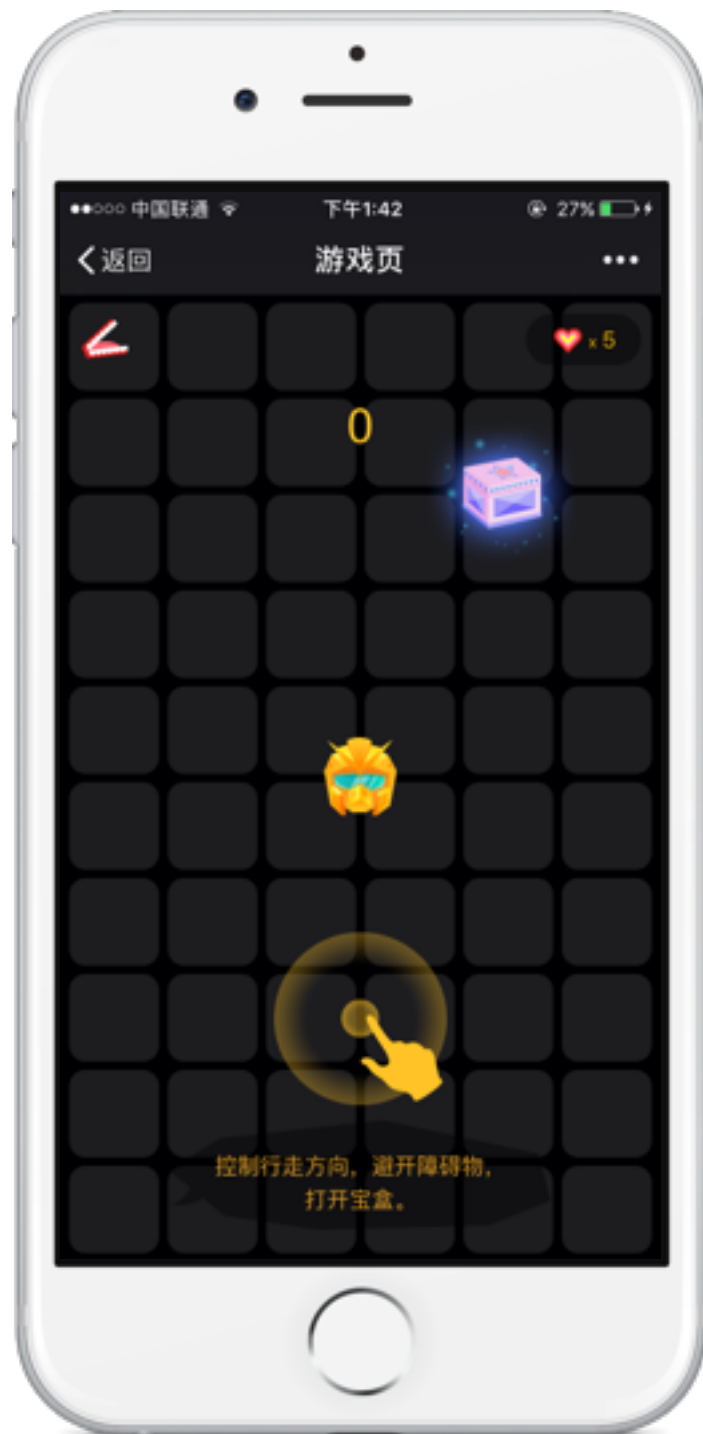


手机新品季SNS



# 最近做的一些案例

02



显示器狂欢季SNS



通过触摸屏幕控制主角运动，躲避出现的障碍物，以存活时间或获得的物品作为成绩。



**CREATEJS**



**EASELJS**



**TWEENJS**



**SOUNDJS**



**PRELOADJS**

# 代码实现

02

```
//主逻辑
var Game = {
}

// 飞船
var Ship = function() {
  this.init();
}

Ship.prototype = {

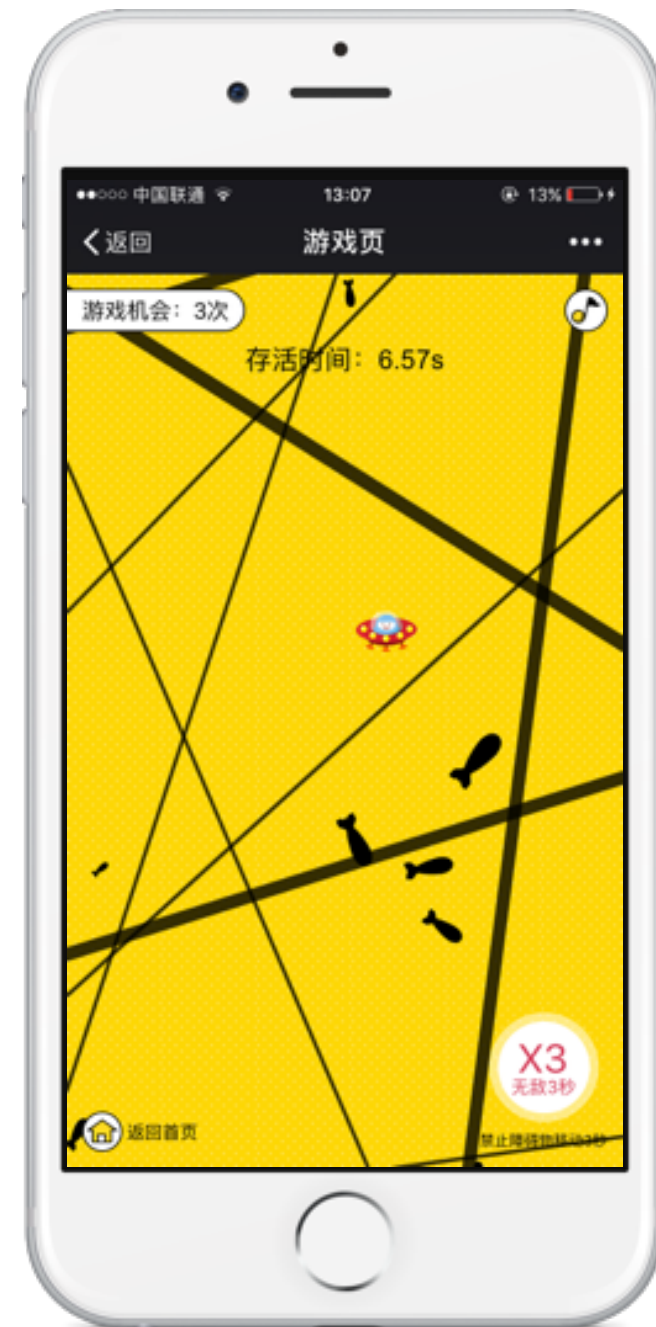
}

//电线
var Line = function() {
  this.init();
}

Line.prototype = {

}
.....
```

游戏主逻辑



障碍物 / 主角

## 游戏主逻辑

资源加载

配置：障碍速度 / 个数、结束回调等

进程控制：start / pause / resume / stop / reset

主角运动控制

障碍物控制

碰撞检测

舞台更新

...



障碍物 / 主角

出现 / 消失

运动（速度、轨迹、角度等）

Tween动画

sprite动画



图形



像素

# 碰撞检测 - 图形

## 1. 圆形 vs 圆形

```
//圆的xy为圆心  
getCirclesCollision: function(circle1, circle2) {  
    var dx = circle1.x - circle2.x;  
    var dy = circle1.y - circle2.y;  
    return Math.sqrt(dx * dx + dy * dy) <= circle1.r + circle2.r;  
}
```

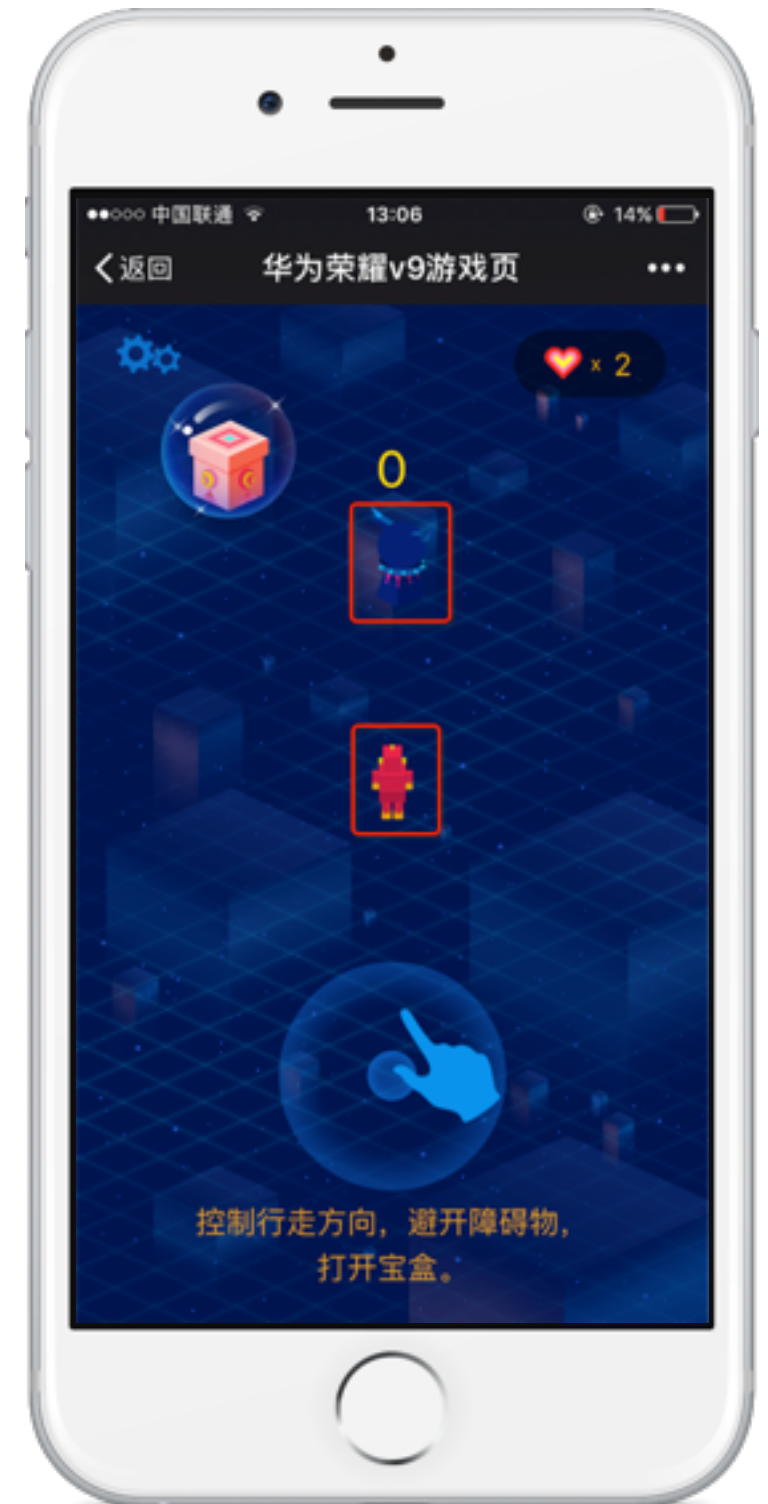


# 碰撞检测 — 图形

02

## 2. 矩形 vs 矩形

```
getRectCollision: function(rect1, rect2) {  
    return rect1.x < rect2.x + rect2.w  
        && rect1.x + rect1.w > rect2.x  
        && rect1.y < rect2.y + rect2.h  
        && rect1.h + rect1.y > rect2.y;  
}
```



# 碰撞检测 — 图形

02

## 3. 圆形 vs 矩形 (旋转)

<http://www.migapro.com/circle-and-rotated-rectangle-collision-detection/>



# 碰撞检测 — 像素

1. 通过将两个物体在 offscreen canvas 中判断交集下是否同时存在非透明的像素。

2. 利用 canvas 的 `globalCompositeOperation = 'destination-in'` 属性。该属性会让两者的重叠部分会被保留，其余区域都变成透明。

<https://github.com/olsn/Collision-Detection-for-EaselJS>



# 一些优化

1.资源优化

2.实例回收，如障碍物消失后放到数组，需要时再添加至舞台。

# 一些几何知识

1. 角度弧度转换
2. 直角三角形求角度边长等
3. 直线方程
4. 两点求角度
- .....



THANKS  
FOR YOUR WATCHING