华清远见
FARSIGHT
嵌入式培训专家

3G ANDROID手机
游戏开发典型案例分析

华清远见：范传奇

手机游戏产业

- 最早的手机游戏出现于1997年，经过十几年的发展，随着手机终端和移动通信网络的不断进步，手机游戏也正在经历由简单到复杂的进化过程。从全球来看，手机娱乐服务被公认为是带动移动数据业务快速发展的重要力量。作为手机娱乐服务的重要内容之一，近年来，伴随着移动网络和移动终端性能的不断提高与完善，手机游戏业务呈现快速增长的势头，成为一座名副其实的“金矿”。

手机游戏产业

国内手机游戏市场在过去几年里呈现快速发展的趋势，2008年中国手机游戏活跃用户数达698万户，手机游戏用户中玩手机网游产品的用户比例逐年快速递增。2008年手机网游用户占总体手机游戏用户的比例达40%左右，用户规模达280万户，这其中大部分用户是手机单机游戏的活跃用户。

手机游戏产业

2009年，手机游戏业务增长继续加速，全年手机游戏市场规模达到18亿元，同比增长38.5%。但是，整个手机游戏18亿市场跟整个网络游戏270亿市场相比还不到其1/10，手机游戏何时爆发成为人们关注的焦点。2010年，中国手机游戏用户规模突破1.3亿，同比增长52.11%，同期中国手机游戏市场规模达33亿。截至2010年底，手机网游产品累计已超过300款，其中2010年新上线60余款。

手机游戏产业

随着3G应用的快速推进以及智能机的普及，手机正逐渐成为“个人信息处理中心”，覆盖到生活的方方面面。随着手机带宽的不断提速，互联网从PC过渡到手机的趋势正在加速蔓延，这也为手机游戏业的兴起造就了新的商业机遇。手机游戏正在呈现一个巨大的市场，成为移动互联网领域的热门增长点。

游戏发展趋势

随着3G时代的到来，以及智能手机的普及，在手机等嵌入式设备上开发的游戏越发的多样化，近年来由apple以及google开发的两大智能手机平台ios,android上的游戏，画面越来越炫目，游戏的创新玩法也越来越新颖。以及它们带给游戏开发者的利益也越来越大。这使得，在智能手机上开发游戏的软件公司越来越多，甚至现在世界上大型的游戏开发公司都纷纷在智能手机平台上开发自家的大作。

华清远见
FARSIGHT
嵌入式培训专家

图层

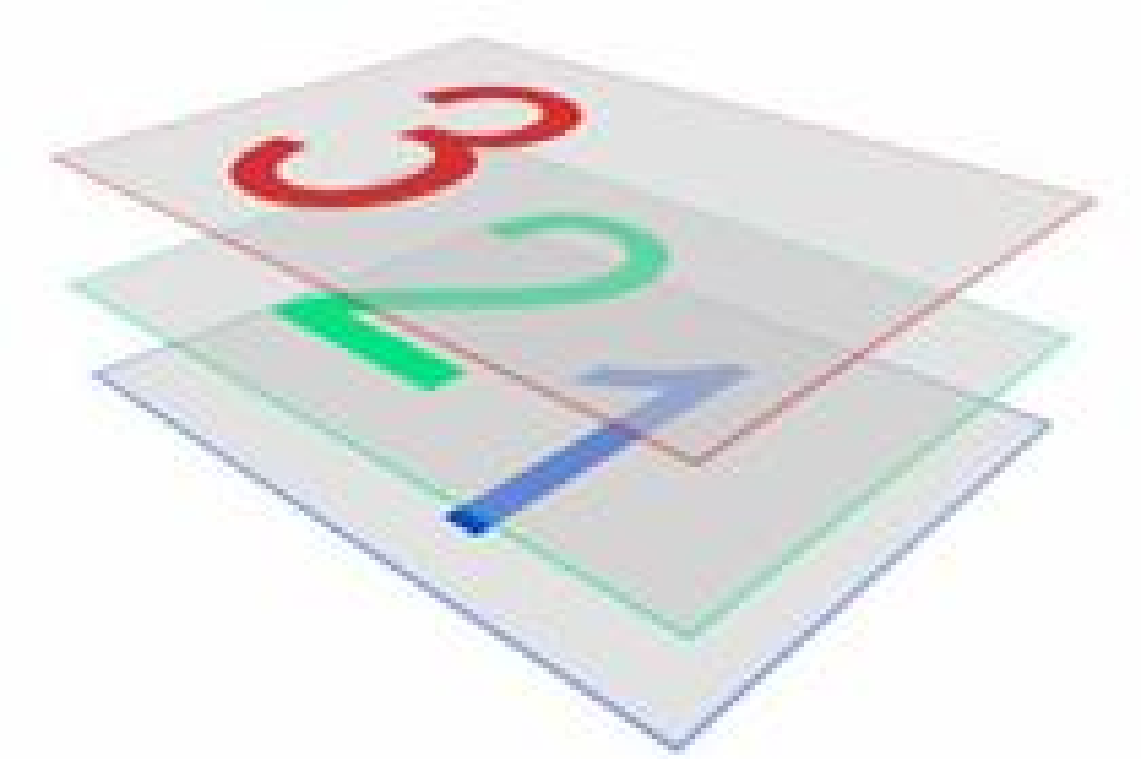
游戏图层的用途

- 为什么要引入图层的概念呢？

  我们可以想象一下一个游戏场景中拥有各种复杂的元素，这每个元素可能都具有不同的属性，功能，动画，特效，并且元素之间也有相互的遮挡，那么如果拥有图层的概念，这些处理就会变得很方便，我们可以把拥有相同属性，相同功能的多个元素放置到同一个场景中的一个图层上，这样一个场景中就可能包含多个图层，比如我们一个游戏场景中，天空和云可以是一个图层，山又可以是另一个图层，植物、障碍物、npc这些就可以是另外的一些图层，这样处理起来就方便很多了。

图层的用途

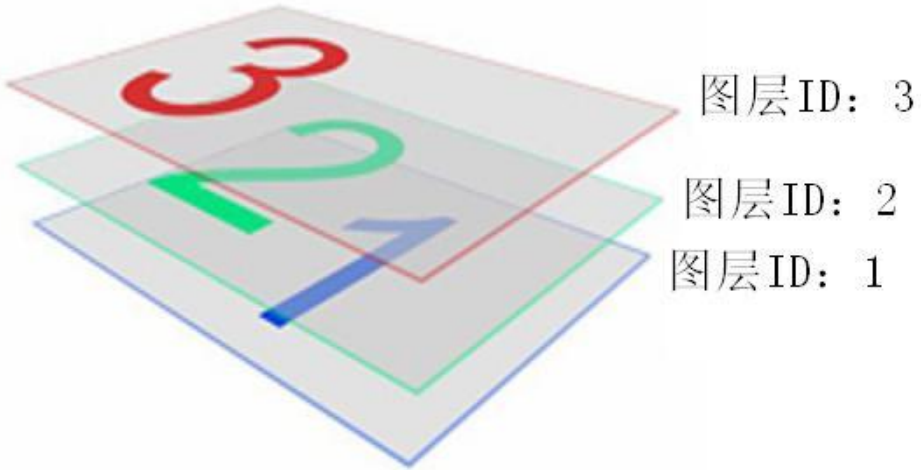
图层是一种没有厚度的、透明的电子画布



图层的实现

我们可以使用代码来模拟图层，实际上就是在同一张画布（Canvas）上绘制多遍不同的素材，这样，后绘制的内容就会覆盖之前绘制的内容，这样就能达到覆盖带来的层次感。

我们可以用某种数据结构来存放所处同一图层的可绘制的素材，然后为每一组图层的数据结构分配一个图层ID，我们可以按照某种顺序来绘制不同ID的数据结构下存放的素材，通常情况下ID是从大倒下排序的数字，那么ID小的图层素材会先绘制到画布上，这样ID大得图层素材会覆盖ID小的。



图层的实现

```
/**
 * 绘图方法，这个方法是由线程控制，周期性调用的
 */
public void onDraw(Canvas canvas) {
    //更新图层内容
    updatePicLayer(CHANGE_MODE_UPDATE,0,null);
    // 遍历所有图层，按图层先后顺序绘制
    for (int id : picLayerId) {
            for (Drawable drawable : picLayer.get(id)) {
                canvas.drawBitmap(drawable.getCurrentPic(),
                        drawable.getPicMatrix(), paint);
            }
    }
    //绘制LOGO
    canvas.drawText("farsight android game demo.  by XiloerFan", 0, 20, paint);
}
```

华清远见
FARSIGHT
嵌入式培训专家

多线程

游戏中的多线程

- 游戏中的多线程的用途？

  在游戏中，我们可以看到每个角色甚至是背景中的物体，都再发生这一些变化，有些相互影响，有些则不然，这些都是使用不同的线程来控制的。

  例如：

  在捕鱼游戏中，我们发射一枚炮弹，这枚炮弹会沿着发射轨迹移动。鱼群也会以不同的方向自由的移动，这些都是得力于多线程。

游戏中的多线程

- 多线程例子

  鱼儿游动：我们让线程以一个周期的方式循环变化鱼在屏幕上的坐标，这样，每次绘制周期中，绘制鱼的位置就再不断变换，看起来就像是鱼在游动一样。

游戏中的多线程

```
public void run() {
    isRun = true;
    // TODO Auto-generated method stub
    int fishX = 0;
    while(isRun){
        /**
         * 累加x，让鱼一直向右移动
         */
        fish.getPicMatrix().setTranslate(fishX,0);
        fishX++;
        if(fishX>screenWidth){
            /**
             * 减去宽度是是为了让鱼的起始位置位于屏幕左边没有出现的位置
             */
            fishX = 0-fish.getCurrentPic().getWidth();
        }
        try {
            Thread.sleep(50);
        } catch (Exception e) {
            // TODO: handle exception
        }
    }
}
```

游戏中的多线程

- 多线程的注意事项？

  在游戏中，我们可能将控制动作的线程体内容放置在一个while循环中，而循环条件不能是true,否则我们无法停止循环，这样游戏退出的时候得不到释放的话是会产生很多不良影响的。所以，我们应该设置一个可以控制的boolean变量，每当程序退出时，我们就主动设置这个值达到让线程停止的目的。

  游戏中经常会使用多线程来访问同一个数据，这样要避免多线程带来的问题，我们应采取线程锁的方式来保护多线程情况下操作同一数据带来的线程不安全操作。

华清远见
FARSIGHT
嵌入式培训专家

游戏素材

游戏素材

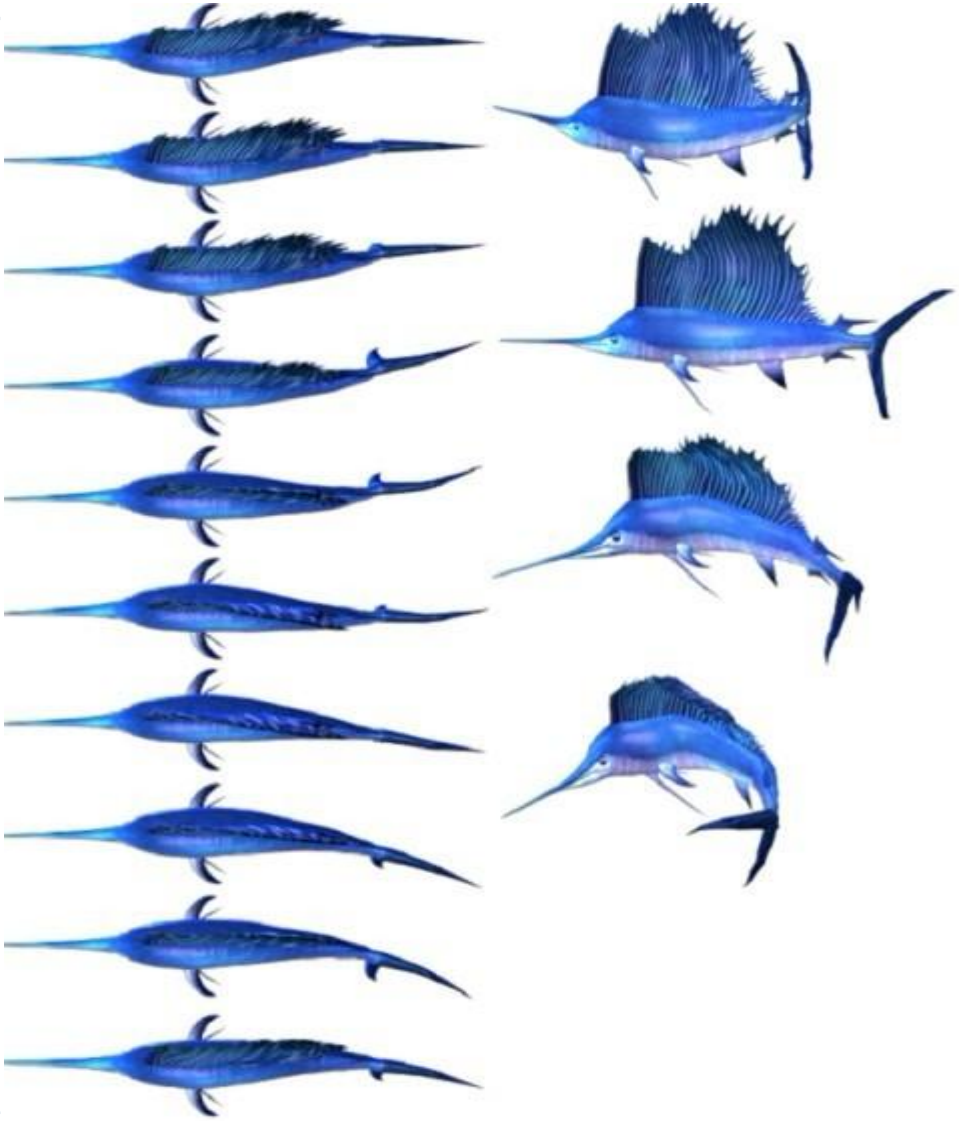
- 游戏中素材的标准？

  游戏中的素材，比如图片，大多数情况下，我们用一组图片来描述一个游戏中角色的连贯动作，但图片一般不是由多张图片文件来分别保存的。而是放在一张图片中，这样便于管理文件名，以及读取方式等。

  音乐方面，对于嵌入式设备的软件，比如android设备中的手机，资源是比较有限的，我们不应该使用很长的音乐来播放音效，我们可以循环播放某一段音效，来达到效果。

  我们应在不影响听觉效果的前提下尽量降低资源的占用为宜

游戏素材



http://www.3g-edu.org/

图片素材的剪裁

既然图片素材中一组表示角色连贯动作的图片都被放置在一张图文件中了，那么我们应该如何剪切呢？我们要在程序中使用通用方法剪切图片的话，那么图片的布局就应有一个标准

通常我们会将图片的尺寸大小告知美工，他们会按照这样的方式布置图片的摆放。每帧图片应有一样的宽度和高度，这样才便于使用通用的方式进行剪裁。当然，图片不一定非要绝对的值来描述宽度和高度，通常他们是有一个比值的，这样也方便我们在不同屏幕尺寸的硬件设备中缩放图片。

比值是宽高比以及与屏幕宽高的比值

在设置图片中每帧图片时，应当注意它与相对的屏幕宽度和高度，要将这两个值记住，才可以在程序中得知它应在运行环境下的屏幕中被缩放的尺寸

图片素材的剪裁

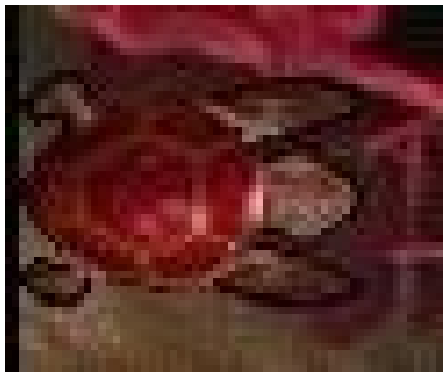
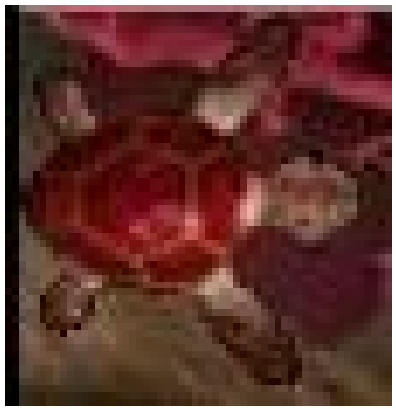
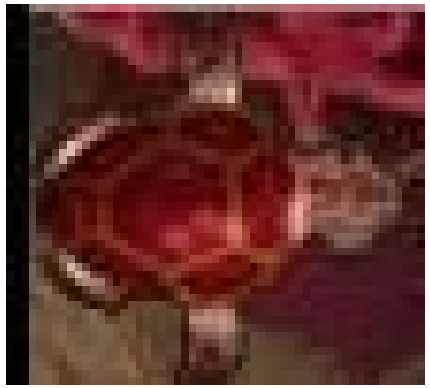
我们可以将素材的剪切写成一个xml文件来维护剪裁方式。

```
<dict>
        <key>texture</key>
        <dict>
                <key>width</key>
                <integer>1024</integer>
                <key>height</key>
                <integer>1024</integer>
        </dict>
        <key>frames</key>
        <dict>
                <key>fish17_06.png</key>
                <dict>
                        <key>x</key>
                        <integer>1</integer>
                        <key>y</key>
                        <integer>1</integer>
                        <key>width</key>
                        <integer>360</integer>
                        <key>height</key>
                        <integer>222</integer>
                        <key>offsetX</key>
                        <real>-4</real>
                        <key>offsetY</key>
                        <real>-1</real>
                        <key>originalWidth</key>
                        <integer>380</integer>
                        <key>originalHeight</key>
                        <integer>256</integer>
                </dict>
                <key>fish17_07.png</key>
                <dict>
                        <key>x</key>
                        <integer>1</integer>
```

让素材动起来

- 使用线程来控制

  鱼儿游动：我们让线程以一个周期的方式循环变化鱼的动作图片，这样，每次绘制周期中，绘制鱼的图片就再不断变换，看起来就像是鱼在游动一样。

华清远见
FARSIGHT
嵌入式培训专家

碰撞检测

碰撞的概念

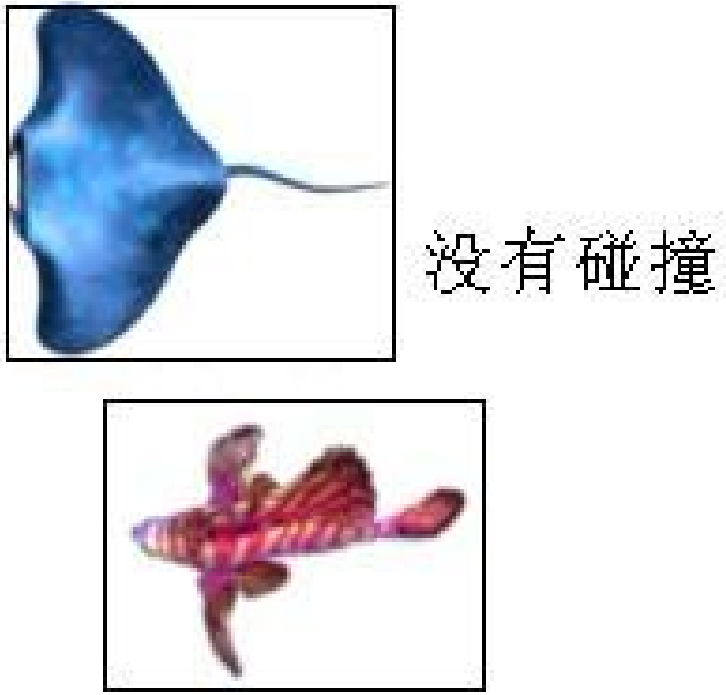
- 在Java 游戏开发中，经常需要进行碰撞检测算法的实现，例如判断前面是否有障碍以及判断子弹是否击中飞机，都是检测两个物体是否发生碰撞，然后根据检测的结果通过碰撞检测算法做出不同的处理。
- 进行碰撞检测算法的物体可能有些的形状和复杂，这些需要进行组合碰撞检测，就是将复杂的物体处理成一个一个的基本形状的组合，然后分别进行不同的检测。

游戏碰撞

碰撞实现

我们在检测碰撞时，一般使用的是矩形碰撞以及圆形碰撞。

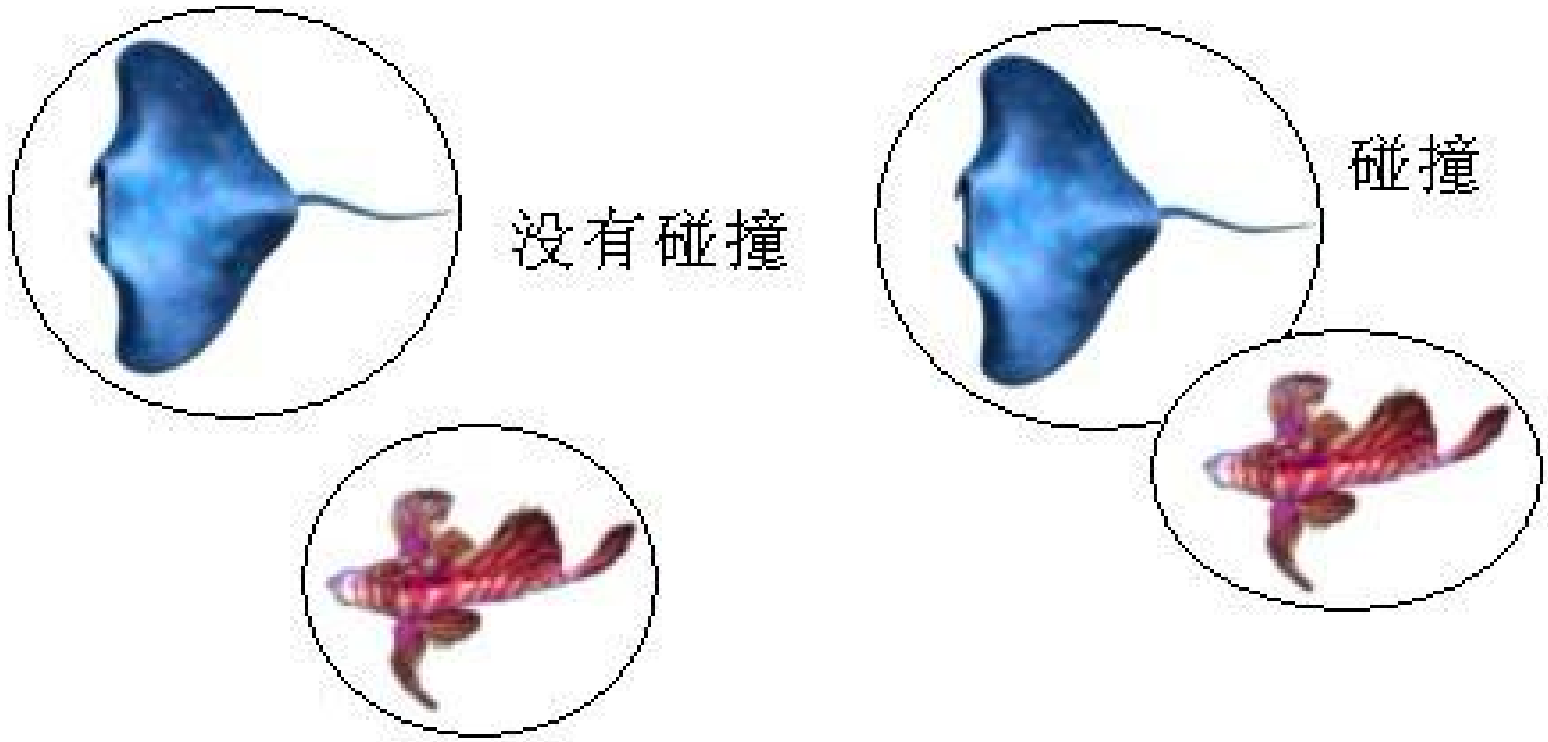
矩形碰撞是指两张图片的外接矩形是否有相交



游戏碰撞

- 碰撞实现

  圆形碰撞是指两张图片的外接圆形是否有相交



游戏碰撞

- 矩形和矩形进行碰撞检测算法
- 一般规则的物体碰撞都可以处理成矩形碰撞，实现的原理就是检测两个矩形是否重叠。我们假设矩形1的参数是：左上角的坐标是(x1,y1)，宽度是w1，高度是h1;矩形2的参数是：左上角的坐标是(x2,y2)，宽度是w2，高度是h2。
- 在检测时，数学上可以处理成比较中心点的坐标在x和y方向上的距离和宽度的关系。即两个矩形中心点在x方向的距离的绝对值小于等于矩形宽度和的二分之一，同时y方向的距离的绝对值小于等于矩形高度和的二分之一。下面是数学表达式：

x方向：| (x1 + w1 / 2) – (x2 + w2/2) | < |(w1 + w2) / 2|
y方向：| (y1 + h1 / 2) – (y2 + h2/2) | < |(h1 + h2) / 2|

游戏碰撞

- 圆形和圆形的碰撞检测算法
- 圆形和圆形的碰撞应该说是一种最简单的碰撞，因为在数学上对于两个圆形是否发生重叠，有计算两个圆心之间的距离的公式。那么条件就变为：计算两个圆心之间的距离是否小于两个圆的半径和。
- 假设圆形1的左上角坐标是(x1,y1)，半径是r1，圆形2的左上角的坐标是(x2,y2)，半径是r2。
- 下面是数学表达式：

$$(x1 - x2)^2 + (y1 - y2)^2 < (r1 + r2)^2$$

碰撞的概念

- 更加精确的碰撞

  精确到像素级，已经不能比这更精确了，相对的，效率也是最低的。怎样判断两个物体是否碰撞呢？

  现在有了PNG和XNA，逐像素检测就相对简单一些。首先仍然需要有一个矩形框包围物体，通过矩形检测得到重叠的矩形区域可以大大减少检测的像素点数量。然后在这个区域内，取两个图片的点逐行逐列迭代，如果遇到某个点两张图片均有颜色存在，即判为碰撞。

碰撞的概念

- 该用什么样的碰撞算法

  矩形和圆形碰撞的效率高，但是碰撞检测相对粗糙，在某个游戏中，如果大量的物体或图片需要进行相互碰撞检测并且不要求碰撞那么精确的时候，我们应当使用这两种碰撞方式，这样可以让我们的程序执行效率更高。在大规模碰撞检测中，我们应当尽量减少像素级碰撞，因为它的计算开销是非常大的。

  类似的，像格斗游戏，我们所检测的碰撞相对比较少，可能只需要判断两个角色是否有碰撞，并且拳脚相碰的检测是比较精确的，那么这个时候，我们应当使用像素级碰撞，这会时格斗游戏看起来更加逼真。

华清远见
FARSIGHT
嵌入式培训专家

音乐

游戏音乐

- 音乐在游戏中已经成为一个有机的、不可缺少的整体。一方面,音乐能烘托气氛,营造一个逼真的游戏氛围环境,并且能够让游戏者在游戏的过程中既得到乐趣,又得到美的艺术享受;另一方面,音乐在增强游戏软件的艺术感染力,推动剧情的发展、深化主题等方面有着特殊的作用,甚至起到了画龙点睛的作用。因此,游戏音乐在这些方面的重要作用与价值已成为游戏中不可缺少的重要组成部分。当然,在不同的游戏中,游戏音乐的作用也稍有差异。在某些以战争、射击等为题材内容的即时战略类、体育竞技类、动作射击类等游戏中,对音乐要求不是太多,音乐主要作为背景音乐使用,为游戏者营造一个良好的游戏氛围。它们中的很多音乐基本不参与游戏内容的发展,或者根据游戏场景内容的不同,选用不同类型的音乐作背景。

android音乐播放器

- Android中的音乐播放器
- MediaPlayer
- SoundPool

android音乐播放器

- MediaPlayer

  MediaPlayer适合比较长且对时间要求不高的情况。MediaPlayer同一时间只能播放一种音乐，所以我们经常会使用MediaPlayer来播放背景音乐。

```
mp = MediaPlayer.create(this, R.raw.fishbg2);
mp.setOnCompletionListener(new OnCompletionListener(){

    public void onCompletion(MediaPlayer arg0) {
        mp.seekTo(0);
        mp.start();
    }

});

mp.start();
```

android音乐播放器

- SoundPool

  SoundPool适合短促且对反应速度比较高的情况

  SoundPool同一时间可以播放多个音效，由于SoundPool的特性，我们不能使用它来播放较长的音效(通常是不超过5秒的音效)所以我们经常会使用它来播放游戏中的音效。

```
sp = new SoundPool(
        5,
        AudioManager.STREAM_MUSIC,
        0
        );
soundMap = new HashMap<Integer, Integer>();
soundMap.put(SOUND_BGM_FIRE, sp.load(SoundManager.context, R.raw.bgm_fire,1));
soundMap.put(SOUND_BGM_NET, sp.load(SoundManager.context, R.raw.bgm_net,1));
```

```
sp.play(soundMap.get(soundID), 100, 100, 0, 0, 0);
```

谢谢！