# 2d物理世界的碰撞检测

分类：圆，矩形，多边形

其中多边形默认为凸多边形，在游戏引擎里，非凸多边形会变成多个凸多边形处理，碰撞检测的任务是判断两个图形有没有碰撞，如果有碰撞要按照顺序输出碰撞点的列表，重叠部分图形的顶点列表(如果重叠部分为非多边形，输出圆弧所在圆的圆心半径和夹角)

## 点a与多边形

判断点与凸多边形的关系：返回0代表在多边形上，-1代表外，1代表内

1. 将多边形的第i条边的第一个顶点指向点P得到向量 v1，然后将从第一个顶点指向第二个顶点得到向量v2，叉乘这两个向量。

2.如果叉乘结果与上一条边的叉乘结果的乘积大于0则继续执行，如果乘积小于0，表示点P不在凸多边形内，直接返回即可。

3. 对于每一对点(p,a,b),ab是多边形边的顶点，还要判断他们是否共线，如果共线，看p是否在线段ab上，如果在返回0，不在返回-1

## 圆与圆

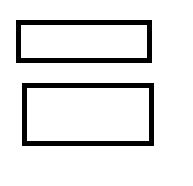
很简单，只要已知半径和圆心，看圆心的距离是否大于半径加和

## 圆与矩形

## 圆与多边形

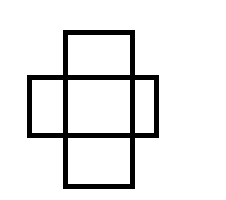
## 矩形与矩形

1. 矩形A和矩形B的顶点互相包含个数为0，且A中点不在B内，B中点不在A内：



此时重叠面积为0

1. 矩形A和矩形B的顶点互相包含个数为0，且A中点在B内，B中点在A内：



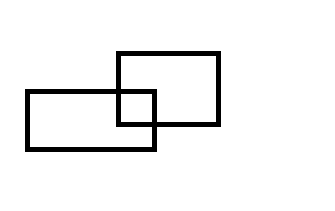
此时计算四个交点，然后用四个交点的组成的矩形求面积。

这里可以用两条线段位置关系求出4个交点，但其顺序不一定是顺时针或逆时针的，有两种解决办法：

一是暴力求解，判断每对交点的和矩形A的每条边是垂直还是平行或共线，由于矩形A在输入时候保证4个顶点有顺序(顺时针或逆时针)，所以也一定可以求出4个交点的顺序，知道顺序就很容易求面积了。

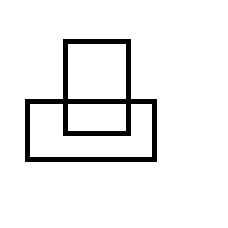
二是用凸包算法，把4个点整理成有序的，凸包算法适用于任意的点求凸多边形并且按照顺时针或逆时针排好

3. 矩形A和矩形B的顶点互相包含个数为1



此时取任意包含的点比如A中的点b1,这个b1是B内的，求这个b1在B的顶点列表两侧的点b0和b2,然后分别求直线b0-b1和A的交点以及直线b2-b1和A的交点，设这两个交点分别为c0和c1,则相交面积是dis(b1,c0)\*dis(b1,c1)

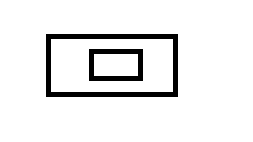
4. 矩形A的顶点包含B或矩形B的顶点包含A个数为2



以矩形A的顶点包含B个数为2为例子：

假设包含的节点为b0和b1,则寻找b1在B的顶点列表中的上一个或下一个不是b0的节点，称为b3, 则相交面积是dis(b1,b0)\*dis(b1,b3)

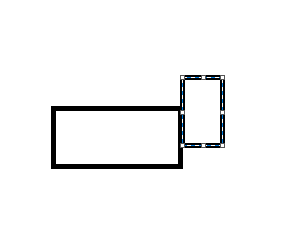
5. 矩形A的顶点包含B或矩形B的顶点包含A个数为4



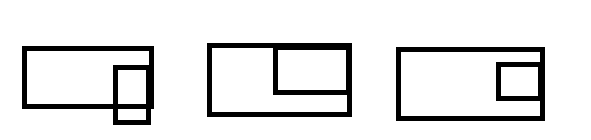
此时直接返回被包含的那个矩形的面积

6.处理两个矩形正好有顶点在另一个的边上的情况

6.1 两个矩形有重叠边，且包含点为0，结果是0



6.2两个矩形有重叠边，且包含点不为0，把重叠点算作在内部，按照前面的方法做



## 矩形与多边形

## 多边形与多边形

## 子弹类物体的碰撞检测

## 四叉树优化

## 例题

<https://ac.nowcoder.com/acm/problem/19905>

两个凸多边形交集

# 3D物理世界的碰撞检测

分类：圆柱体，球体，立方体