



软件测试基础与实践

实验报告

实验名称： 黑盒测试实验二

实验地点： 软件学院机房

实验日期： 2020 年 12 月 9 日

学生姓名： 叶宏庭

学生学号： 71118415

东南大学 软件学院 制



一、实验目的

- (1) 能根据待测软件的特点, 选择合适的方法对软件进行黑盒测试(功能测试);
- (2) 了解随机测试, 巩固白盒测试和黑盒测试方法;
- (3) 了解 JUnit 测试开发框架及其应用;
- (4) 能对一些特定的程序进行蜕变测试。

二、实验内容

(一) 题目 1: 随机测试 VS 黑盒测试 VS 白盒测试

1. 实验背景

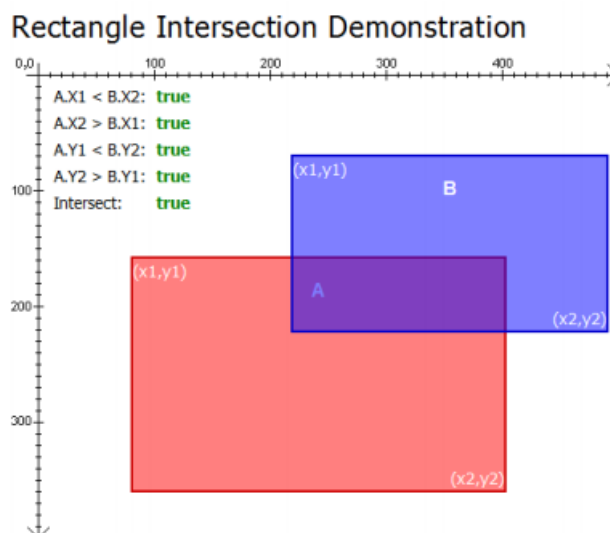
在游戏引擎开发中, 检测物体碰撞是一项重要的基础功能, 比如 DOTA 和王者荣耀等游戏中的各种华丽大招的伤害波及范围计算等。

为简单起见, 我们这里只考虑二维平面空间的情况, 并用 RectManager 程序判断平面上任意两矩形的相交关系 (A:不相交, B:相交: B1:相交为一个区域, B12:包含, B13:完全重合, B2:交点为 1 个点, B3:交点为 1 条线段), 如果相交, 则同时给出相交部分的面积。

这里的二维平面限定为 iphone4 屏幕(640*960 分辨率), 且所有矩形的边都与坐标轴平行。

计算机图形学中, 通常用左上角和右下角的坐标来表示一个矩形。

任意两个矩形的关系可借用这个工具来辅助分析: <http://silentmatt.com/rectangle-intersection/>
坐标系请参照下图:



2. 实验要求

(1) 请编写一简单程序, 随机生成两个矩形的数据作为测试用例, 请用这些测试用例对 RectManager 进行测试。

要求:



a) 编写程序，实现用随机函数生成大量测试用例（10 万-100 万个），对上述问题进行随机测试。

b) 注意随机测试用例产生的范围应比屏幕范围稍微大一点。屏幕范围：x 取值范围[0-639]，y 取值范围[0-959]；

c) 在测试用例生成程序中，同时调用 RectManager 中的方法直接驱动测试自动执行。

d) 对随机测试结果进行统计，分析随机测试用例对两矩形相交的各种关系的覆盖情况（统计上的命中概率）。

e) 给出源代码和测试运行结果。

熟悉 JUnit 的同学，可用 JUnit 实现上述随机测试。

(2) 请用黑盒测试方法，设计相应的测试用例来测试程序（可参考并重用实验四中设计的测试用例）；

提示：程序运行命令行：java -jar RectManager.jar

(3) 请分析 RectManager 的实现源代码，利用基本路径测试方法对程序进行白盒测试；

只要求针对 solve() 方法进行测试（只给出基本路径，不用具体设计测试用例）。

(4) 在上述实验的基础上分析三种测试方法发现缺陷的能力上有何差别。

3. 实验过程与结果

(1) 解答问题 1

测试结果：

矩形关系类型	用例数量	命中概率
矩形不相交！	554577	0.5546
矩形相交于一个区域！	386498	0.3865
矩形相交于一个区域且为包含关系！	55759	0.0558
矩形相交于一个区域且正好重合！	0	0.0000
矩形相交于一个区域且交点为 1 个点！	4	0.0000
矩形相交于一个区域且交点为 1 条线段！	3162	0.0032

注：测试源代码见附录。

(2) 解答问题 2

采用等价类划分方法先给出矩形关系类型对应的编号：

矩形关系类型	编号
矩形不相交！	0
矩形相交于一个区域！	1
矩形相交于一个区域且为包含关系！	2
矩形相交于一个区域且正好重合！	3
矩形相交于一个区域且交点为 1 个点！	4
矩形相交于一个区域且交点为 1 条线段！	5

测试用例：(left, top, right, bottom)

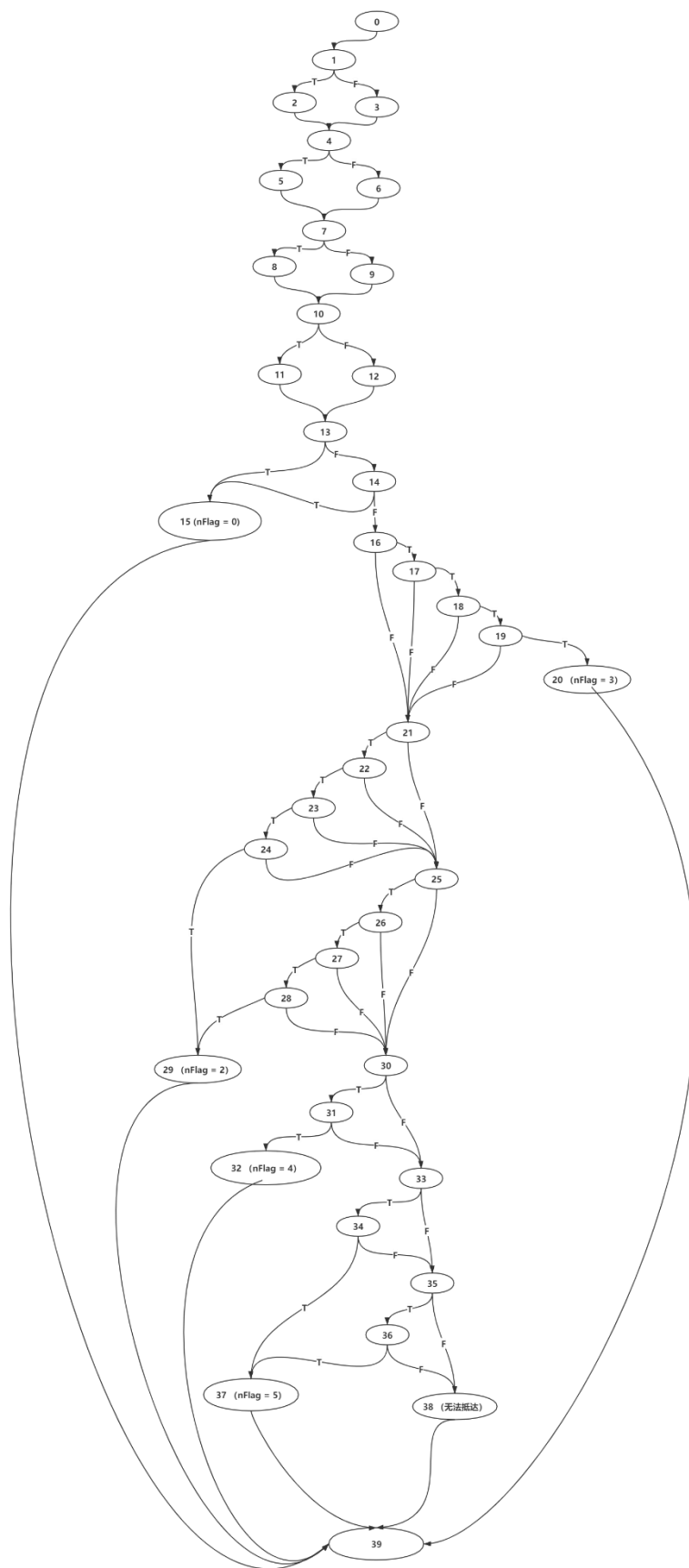
矩形一	矩形二	矩形关系类型	预期输出	实际输出
(1, 1, 2, 2)	(3, 3, 4, 4)	矩形不相交！	0	0
(3, 3, 4, 4)	(1, 1, 2, 2)		0	0



(1, 3, 2, 4)	(2, 1, 3, 2)		0	0
(2, 1, 3, 2)	(1, 3, 2, 4)		0	0
(1, 1, 3, 3)	(2, 2, 4, 4)	矩形相交于一个区域!	1	1
(2, 2, 4, 4)	(1, 1, 3, 3)		1	1
(1, 3, 4, 6)	(2, 2, 5, 5)		1	1
(2, 2, 5, 5)	(1, 3, 4, 6)		1	1
(3, 3, 4, 7)	(1, 5, 6, 6)		1	1
(1, 1, 5, 5)	(2, 2, 3, 3)	矩形相交于一个区域且为包含关系!	2	2
(2, 2, 3, 3)	(1, 1, 5, 5)		2	2
(1, 1, 3, 3)	(1, 1, 3, 3)	矩形相交于一个区域且正好重合!	3	3
(1, 1, 3, 3)	(3, 3, 4, 4)	矩形相交于一个区域且交点为1个点!	4	4
(3, 3, 4, 4)	(1, 1, 3, 3)		4	4
(1, 3, 3, 5)	(3, 2, 4, 3)		4	4
(3, 2, 4, 3)	(1, 3, 3, 5)		4	4
(2, 2, 3, 3)	(2, 1, 3, 2)	矩形相交于一个区域且交点为1条线段!	5	5
(2, 2, 3, 3)	(1, 2, 2, 3)		5	5
(2, 2, 3, 3)	(2, 3, 3, 4)		5	5
(2, 2, 3, 3)	(3, 2, 4, 3)		5	5

(3) 解答问题 3

程序流程图:



基本路径集: $V(G) = 63 - 40 + 2 = 25$

编号	路径
P1	0-1-2-4-5-7-8-10-11-13-15-39
P2	0-1-3-4-5-7-8-10-11-13-15-39
P3	0-1-2-4-6-7-8-10-11-13-15-39
P4	0-1-2-4-5-7-9-10-11-13-15-39
P5	0-1-2-4-5-7-8-10-12-13-15-39
P6	0-1-2-4-5-7-8-10-11-13-14-15-39
P7	0-1-2-4-5-7-8-10-11-13-14-16-17-18-19-20-39
P8	0-1-2-4-5-7-8-10-11-13-14-16-21-22-23-24-29-39
P9	0-1-2-4-5-7-8-10-11-13-14-16-17-21-22-23-24-29-39
P10	0-1-2-4-5-7-8-10-11-13-14-16-17-18-21-22-23-24-29-39
P11	0-1-2-4-5-7-8-10-11-13-14-16-17-18-19-21-22-23-24-29-39
P12	0-1-2-4-5-7-8-10-11-13-14-16-21-25-26-27-28-29-39
P13	0-1-2-4-5-7-8-10-11-13-14-16-21-22-25-26-27-28-29-39
P14	0-1-2-4-5-7-8-10-11-13-14-16-21-22-23-25-26-27-28-29-39
P15	0-1-2-4-5-7-8-10-11-13-14-16-21-22-23-24-25-26-27-28-29-39
P16	0-1-2-4-5-7-8-10-11-13-14-16-21-25-30-31-32-39
P17	0-1-2-4-5-7-8-10-11-13-14-16-21-25-26-30-31-32-39
P18	0-1-2-4-5-7-8-10-11-13-14-16-21-25-26-27-30-31-32-39
P19	0-1-2-4-5-7-8-10-11-13-14-16-21-25-26-27-28-30-31-32-39
P20	0-1-2-4-5-7-8-10-11-13-14-16-21-25-30-33-34-37-39
P21	0-1-2-4-5-7-8-10-11-13-14-16-21-25-30-31-33-34-37-39
P22	0-1-2-4-5-7-8-10-11-13-14-16-21-25-30-33-35-36-37-39
P23	0-1-2-4-5-7-8-10-11-13-14-16-21-25-30-33-34-35-36-37-39
P24	0-1-2-4-5-7-8-10-11-13-14-16-21-25-30-33-35-38-39
P25	0-1-2-4-5-7-8-10-11-13-14-16-21-25-30-33-35-36-38-39

(4) 解答问题 4

三种测试方法各有优缺点，总结如下表：

测试方法	优点	缺点
随机测试	能够自动产生大量测试用例，测试速度快	能够覆盖大多数测试情况，但是对于特殊的情况无法做到覆盖
黑盒测试	相比随机测试，能够较为全面的考虑测试的情况，设计的用例覆盖情况较多	做不到测试情况的全覆盖，设计测试用例较为困难
白盒测试	能够覆盖测试的全部情况，考虑每一种特殊情况	流程复杂繁琐，花费时间多



(二) 题目 2:蜕变测试问题

1. 实验背景

在很多软件测试活动中，人们发现给出一个测试用例的期望输出是一件很困难的事情，在一些情况下，人们甚至无法给出测试用例的预期输出。这种测试预测输出无法获得的问题，就称为测试预言问题(Test Oracle Problem)。为解决 Test Oracle 问题，1998 年 T.Y. Chen 提出的蜕变测试的概念。

游戏引擎中需要高效的计算，所以其中的数学函数都需要重新进行快速实现，不能调用系统的自带数学函数。如下的程序片段是一个 $\sin(x)$ 函数的实现代码示例：

```
//always wrap input angle to -PI..PI
if (x < -3.14159265)
    x += 6.28318531;
else
if (x > 3.14159265)
    x -= 6.28318531;

//compute sine
if (x < 0)
    sin = 1.27323954 * x + .405284735 * x * x;
else
    sin = 1.27323954 * x - 0.405284735 * x * x;
```

2. 实验要求

Sin.exe (该执行程序见课程主页) 是一个用某种数值计算方法求解数学函数 $f(x)=\sin(x)$ 的程序。请设计测试用例，实现对该程序的黑盒测试。

要求：

- (1) 给出每个测试用例设计的理由；
- (2) 这个实验中展示的测试用例输出值无法预测的情形还可能出现在哪些实际应用中。

提示：

- a) 查找蜕变测试资料，了解测试 Oracle 问题和蜕变测试简单原理；
- b) 理解蜕变关系，并知道某些特定测试问题的蜕变关系。
- c) 测试过程中，应在 cmd 命令行终端环境下执行，避免程序闪退看不到结果。

3. 实验过程与结果

(1) 解答问题 1

蜕变关系： $\sin(\pi - x) = \sin(x)$

测试用例 x	输出结果	测试用例 x	输出结果	设计理由
15°	0.258819	165°	0.258819	采用蜕变关系，进行测试。（正值）
30°	0.5	150°	0.500001	
45°	0.70106	135°	0.707107	
60°	0.866025	120°	0.866025	
75°	0.965926	105°	0.965925	
90°	1	90°	1	
-165°	-2.87979	-15°	-0.261799	采用蜕变关系，进行测试。
-150°	-2.61799	-30°	-0.523599	
-135°	-2.35619	-45°	-0.785398	



-120°	-2.0944	-60°	-1.0472	试。(负值)
-105°	-1.8326	-75°	-1.309	
-90°	-1.5709	-90°	-1.5709	

测试结果：正值度数基本准确，存在少量误差积累问题。负值度数存在严重算法问题，没有考虑 \sin 函数的周期性，出现 $\sin(x) < -1$ 的情况，这是严重错误。

(2) 解答问题 2

测试输出无法预测的蜕变测试问题应用举例：

- 测试 $\exp(x)$ 问题
- 无向图的最短路径问题
- 桥梁应力测试

三、实验思考

1. 通过测试，是否发现程序中存在的缺陷？

答：发现缺陷，对于 `Sin.exe` 存在严重的 BUG，负数的 \sin 值计算错误。

2. 经过一系列的实验，请谈谈你所感受的软件测试中存在哪些挑战性的难题。

答：测试用例无法覆盖所有可能的情况，不同的测试方法存在各自的优劣性，对于时间开销与资源开销无法做好把控，如何完成高效率的自动化测试。

3. 随机测试中，对很多同学的随机数生成范围来说，矩形不相交的统计概率不是 0.5，你的实验是否存在这个现象？尝试用概率知识对结果进行分析一下。

答：首先假设矩形 A 已经确定，那么我们需要考虑矩形 B 随机生成是否会与矩形 A 形成相交，矩形 B 随便生成的四个参数，需要其中两个参数落入矩形 A 的覆盖范围内，因此对于该落入的概率不是简单的等于 0.5。

4. 蜕变关系是否容易发现？蜕变测试的可适用场景有哪些？如果是 $\tan(x)$, $\log_2(x)$ 等函数的实现，其蜕变测试怎么做呢？需要寻找什么蜕变关系？

答：不容易发现，适用于输出无法准确预测的场景。 $\tan(\pi - x) = -\tan(x)$, $\log_2(x) + \log_2(1/x) = 0$ 。

四、实验体会

通过本次实验，我更加深入理解了多种测试方法的目的与意义，动手实践过程中，对每一种测试的方法、过程有了初步了解与掌握，希望在未来的学习与工作中，能够继续学习，深入掌握，设计出更好的测试用例，做一个合格的测试工程师。