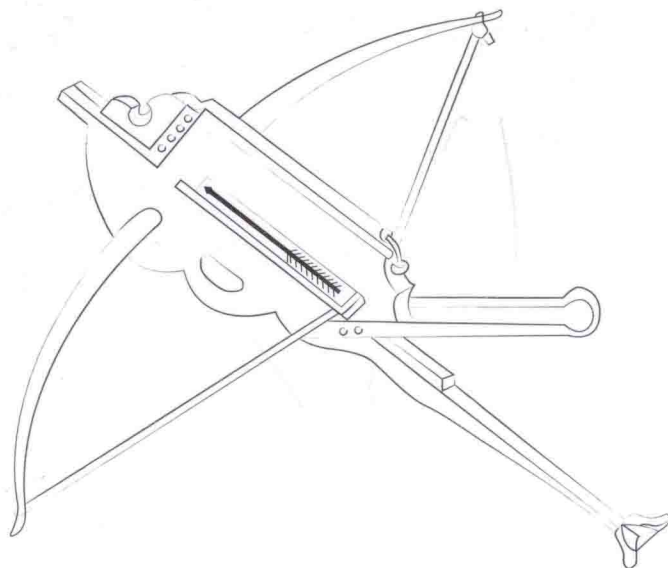


首度全面解析游戏自动化测试

Broadview[®]
www.broadview.com.cn



游戏自动化测试实践

本书适合：喜欢游戏，又喜欢测试，还相对较懒的人
独创性强，技术全面，实战为要

陈大卫 李建玲 著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

游戏自动化测试实践

陈大卫 李建玲 著

电子工业出版社
Publishing House of Electronics Industry

内 容 简 介

本书中所探讨的主题,业界目前尚无统一理论与实现体系,因此在书籍的第一部分(1~3章)阐述完游戏测试的特点、工具及自动化测试在游戏测试中的引入与理解等主旨思想后,更多的篇幅放在了实践上,从相对孤立的自动化测试实现各阶段技术点与思路掌握(4~6章),到完整的测试任务实现(第7章),实例贯穿始终。

实例代码会由书籍内容本身和可下载的资源包两部分构成,两部分都是相对完整的。即使缺少该资源包也不会导致书籍体系的不完整,但是测试代码总要在被测对象上才有用武之地。因此,除测试代码外的实际被测游戏也不应忽视,该资源包正是为了促进读者学习理解而准备的。

本书适合计算机软件测试从业人员,尤其是游戏软件测试人员及希望拓展自动化测试的测试人员阅读。此外,也可供游戏开发人员、游戏项目经理及游戏玩家(因为书中很多技术可被用于游戏外挂的制作)参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

游戏自动化测试实践/陈大卫,李建玲著. —北京:电子工业出版社,2015.1
ISBN 978-7-121-25216-7

I. ①游… II. ①陈… ②李… III. ①游戏程序—测试技术 IV. ①TP311.51

中国版本图书馆CIP数据核字(2014)第302722号

策划编辑:孙学瑛

责任编辑:李云静

印 刷:三河市双峰印刷装订有限公司

装 订:三河市双峰印刷装订有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

开 本:787×980 1/16 印张:15 字数:249千字

版 次:2015年1月第1版

印 次:2015年1月第1次印刷

定 价:59.00元



凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

前 言

本书介绍的是游戏软件的自动化测试实现，讲解内容以实践为主，理论为辅。实例将会贯穿整书。虽然本书不会过多讲解测试理论，但是并不能说明我们的自动化测试实现可以脱离测试思想而单独存在。反而实际情况是，游戏软件的自动化测试首先应该是软件测试，然后是游戏软件的测试，最后才是游戏软件测试的自动化实现。专注程度不断提高，实践性逐渐加强。因此，建议各位读者应首先对于软件测试进行适当了解和实践后再阅读和体会此书；否则可能会被更多的具体技术实现所吸引，而偏离了测试这一根本性主旨。

本书主要由三部分构成，各部分各有侧重、相对独立，读者可根据需要跳跃式阅读。第一部分（第1~3章）主要探讨游戏测试的特点、工具以及自动化测试在游戏测试中的引入与理解等；第二部分（第4~6章）按照自动化测试实现三阶段要点详细介绍各技术实现的思路和特点；第三部分（第7章）以完整实例分析，抛砖引玉式地引发各位读者更优的游戏自动化测试实践。同时本书还有放于 www.broadview.com.cn/25216 的资源包可供下载（拒绝因商业目的而对本书及该资源包一切内容的引用。若在教学或技术交流中引用该资源包中的案例，则需完整明确注明案例出处），里面包含了第三部分综合实例中的所有实例（含源码），还包含了第一、二部分中的大部分案例，以及其他少量相关内容。该资源包体现了本书实践的主旨：引用伟人的话就是“实践是检验真理的唯一标准”。

无论是游戏测试还是自动化测试，都包含了广博的知识，所以各种理论和实践层出不穷。给大家所呈现的也只能是“花海”中的一朵“小花”而已，若有讲解不周或纰漏之处还望谅解和指正。

虽然书中所提到的实现在很大程度上可以被用于游戏外挂的开发，但是读者一定要明白，技术本无好坏，不过却会因使用者的心态不同而发挥不同的作用。

思想远比实现重要得多，笔者在后续例子中不厌其烦地介绍了很多实现的细节，那也只是为了更加充实对于思想体现的理解。

笔者为什么选择游戏实例介绍测试自动化？原因之一是游戏的灵活性使得各种自动化技术得以被应用，而远比一些行业软件要丰富多彩；原因之二是目前游戏自动化在一定程度上属于空白领域，所以笔者愿意作为少数较早“吃螃蟹”的人与大家探索这一领域的技术；原因之三是因为自己比较喜欢游戏，也很喜欢测试。那为何不将游戏和测试结合起来呢？主要是本人比较懒，这就成为笔者游戏自动化测试的原动力。

目 录

| | |
|-------------------|----|
| 第 1 章 游戏测试 | 1 |
| 1.1 游戏测试的特点 | 2 |
| 1.2 游戏测试的三个层次 | 3 |
| 1.2.1 常规软件测试 | 4 |
| 1.2.2 游戏参数调节 | 4 |
| 1.2.3 游戏性测试 | 5 |
| 1.3 游戏自动化测试的可行性 | 5 |
| 第 2 章 软件测试工具运用 | 9 |
| 2.1 什么是软件测试工具 | 9 |
| 2.2 幸福五子棋 9 连测试 | 10 |
| 2.3 拖拉机纸牌游戏测试 | 13 |
| 2.4 斗地主游戏测试 | 17 |
| 2.5 测试工具在测试中的思想体现 | 20 |

| | |
|---------------------|----|
| 第 3 章 自动化测试思想 | 21 |
| 3.1 目标决定了实现方式与思考方法 | 21 |
| 3.2 脚本语言在自动化测试中的优越性 | 22 |
| 3.3 自动化测试实现的工具无关性 | 23 |
| 3.4 功能自动化测试实现三要点 | 24 |
| 第 4 章 自动执行 | 27 |
| 4.1 直接控件操作 | 27 |
| 4.1.1 控件操作 | 27 |
| 4.1.2 窗口/控件信息的获取 | 28 |
| 4.2 模拟键盘和鼠标 | 30 |
| 4.2.1 模拟键盘和鼠标操作 | 30 |
| 4.2.2 窗口的定位与激活 | 31 |
| 4.2.3 绝对坐标与相对坐标 | 31 |
| 4.2.4 坐标的获取 | 32 |
| 4.2.5 录制脚本 | 34 |
| 4.2.6 有效的延时等待 | 35 |
| 4.3 收发网络消息交互 | 36 |
| 4.3.1 TCP 连接的建立 | 37 |
| 4.3.2 UDP 连接的建立 | 38 |
| 4.3.3 局域网游戏——拱猪 | 39 |
| 第 5 章 状态识别 | 42 |
| 5.1 不进行状态识别 | 42 |
| 5.2 直接获取所需状态信息 | 45 |
| 5.3 通过状态伴随表现间接识别 | 45 |
| 5.4 进行状态推算 | 46 |
| 5.5 通过屏幕的像素扫描 | 49 |
| 5.5.1 打苍蝇 | 50 |

| | |
|---------------------|-----------|
| 5.5.2 扫雷 | 52 |
| 5.5.3 连连看 | 54 |
| 5.6 直接读取内存 | 55 |
| 5.6.1 扫雷 | 59 |
| 5.6.2 九宫格 | 61 |
| 5.7 程序后门 | 66 |
| 第 6 章 逻辑处理 | 68 |
| 6.1 顺序执行 | 68 |
| 6.2 使用简单算法完成常规处理 | 70 |
| 6.2.1 扫雷 | 71 |
| 6.2.2 局域网拱猪 | 73 |
| 6.3 利用计算机优势, 得到较优解 | 74 |
| 6.3.1 点灯游戏 | 75 |
| 6.3.2 连连看 | 77 |
| 6.4 移花接木, 利用外部逻辑 | 80 |
| 第 7 章 综合实例演示 | 84 |
| 7.1 中国象棋 | 84 |
| 7.1.1 测试策略的选定 | 84 |
| 7.1.2 实例源码 | 86 |
| 7.1.3 实例点评 | 105 |
| 7.2 幸福跳棋 | 106 |
| 7.2.1 测试策略的选定 | 108 |
| 7.2.2 实例源码 | 109 |
| 7.2.3 自动化测试实现的工具无关性 | 114 |
| 7.2.4 脚本的优化与完善 | 116 |
| 7.2.5 实例点评 | 121 |
| 7.3 幸福五子棋 | 122 |

| | |
|----------------------|-----|
| 7.3.1 测试策略的选定 | 122 |
| 7.3.2 实例源码 | 123 |
| 7.3.3 实例点评 | 141 |
| 7.4 手机模拟器 | 142 |
| 7.4.1 翻方块 | 143 |
| 7.4.2 打地鼠 | 146 |
| 7.4.3 实例点评 | 150 |
| 7.5 手机游戏 | 151 |
| 7.5.1 一刻到底 | 152 |
| 7.5.2 2048 | 161 |
| 7.6 九宫格 | 183 |
| 7.6.1 测试策略的选定 | 183 |
| 7.6.2 实例源码 | 184 |
| 7.6.3 Nokia 数独游戏 | 195 |
| 7.6.4 实例点评 | 199 |
| 7.7 仙剑 Online 防外挂码识别 | 200 |
| 7.7.1 备选答案识别 | 201 |
| 7.7.2 防外挂码识别 | 202 |
| 7.7.3 防外挂码识别的正确率 | 203 |
| 7.7.4 实例源码 | 205 |
| 7.7.5 实例点评 | 214 |
| 附录 A Autolt v3 简介 | 216 |
| 附录 B 中国象棋棋谱规则 | 220 |
| 附录 C 因果图分析之中国象棋走马 | 222 |
| 后记 | 231 |

第 1 章

游戏测试

何谓游戏，在辞海中给出的定义是：以直接获得快感为主要目的，且必须有主体参与互动的活动。很显然我们后续所谈及的游戏，都指的是伴随计算机发展而出现的电子游戏软件。

何谓软件测试，IEEE 在 1983 年给出的定义是：使用人工或自动手段来进行或测定某个系统的过程，其目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别。同样我们后续谈到的测试也并非全集，而仅限定在了功能自动化测试的范畴内。

游戏测试中，变是绝对的，不变是相对的。变是因为游戏类型的多样性、游戏制作技术的多样性、游戏本身的创意性等诸多因素所决定的；所谓不变，只是因为便于我们研究和实现所总结出来的“规律”，而且这种规律除了下面所谈到的几点外，在本书中更多地体现在了功能自动化测试实现三要点的技术上。

1.1 游戏测试的特点

本书讨论的是游戏软件的自动化测试，那么就首先要了解游戏软件测试的特点，然后根据其特点，合理地开展游戏自动化测试。

游戏软件虽然也是软件，但是在我们的印象中，它却远远有别于常规的软件，测试的难度也往往要远高于同等规模的其他类型软件，即使是在很多人眼中的小游戏，也是“麻雀虽小，五脏俱全”，要想做到测全、测精，很不容易。游戏测试的特点主要表现在以下几个方面。

1. 游戏测试的灵活性

由于游戏种类繁多，而且在同一款游戏中也是变化无穷，所以不断地变化，就成为游戏最大的特点，常规测试中的固定输入、输出的做法受到了很大的冲击。此时测试不能再循规蹈矩地做事情了，而应该也跟随变化，甚至比游戏变化得更快，提前推测出游戏后续的发展变化。这样测试自然是有一定难度的，但是万物皆规律，例如各种游戏的游戏规则的结合就是游戏测试很好的入手点。

游戏软件技术实现的多样性也否定了很多传统的测试实现方式，仅以我们本书所讨论的自动化测试就可以看出，在其他软件自动化测试领域取得成功的那些自动化工具，在面对游戏的自动化测试时却纷纷落马。而一些并不为人所熟知的小工具，却可能收到令人意想不到的效果。

2. 游戏测试主观度的把握

游戏软件测试不仅要实现常规软件测试中所重点关注的功能逻辑，与各种环境的兼容，以及软件效率等测试内容；还有自身特别的关注点，就是要充分关注游戏的游戏性。关于游戏性尚有颇多不同观点，但是无人质疑它对游戏所起到的重要作用。而目前尚没有可被测试人员很好使用的评价模型和量化指标，因此更多的是强调测试人员自己的经验和主观评判。

当今游戏软件无不追求着美，显示出艺术的光芒；而艺术向来是很难以统一标准被评价和量化的，所以在测试中尽量避免以个人的喜好而假想为所有玩家的喜好，需要慎重把握主观评判的度。

而另外一个需要慎重评判的事情是游戏均衡性的调节。很多时候，就因为一种特定战术的运用，而彻底颠覆了看似平衡的游戏属性。所以不要轻易改变相对稳定的游戏参数，除非可以证明新的系数确实优于之前的系统。

3. 游戏测试之艰辛的极限测试

在游戏中没有最强，只有更强，测试人员的游戏能力永远不能覆盖住所有玩家。而Bug就出现在了“骨灰级”玩家的极端情况下。这种事例在游戏测试中屡见不鲜，因此测试人员必须覆盖住这些极端情况，这通常成为测试人员最头痛的事情之一了。

将自身打造成为顶尖的骨灰级玩家，或者借助各种作弊手段达到骨灰级玩家所不能，是在实际测试中两条可行的有效途径。但是“作弊”在多数玩家们看来是一种很无耻的行为，而优秀的游戏测试人员又通常拥有很强的玩家心态，因此必须要提醒自己是一名测试人员，而非纯粹的玩家。

4. 游戏测试的知识广博性

在游戏软件中除了计算机知识、美学、心理学等通用知识外，还会涉及上到天文，下到地理，古有神化传说，今有科学幻想等各个领域的知识。若不具有相关知识背景，往往事倍功半，会极大地影响测试效果。

1.2 游戏测试的三个层次

为了能够让各位对于层次划分的理解尽量少产生歧义，我在这里首先要和大家聊一聊“游戏性”(Gameplay)的含义。该词最早见于1999年Geoff Howland发表的《游戏性的焦点》(The Focus Of Gameplay)一文。由于时间久远，当时作者所指的游戏性实际上泛指“游戏的玩法”或者“游戏”，与现在的普遍概念略有出入。到目前为止，在

游戏界中尚没有给出“游戏性”的明确定义，但是归结起来大体上是两方面的意见。

- 其一是延续游戏玩法的理解，也就是强调游戏的具体操作、游戏的实现效果等。因此置入感和交互性越强的，其游戏性也越强，例如，极品飞车游戏中超真实的赛车体验。
- 而另一种理解则是更符合中国人务虚传统的理解，即游戏的核心性质，如可玩性或趣味性等。这种性质在很大程度上是不依赖于具体实现的，这种理解典型的例子就是棋牌类游戏，它甚至不需要任何介质，就可以达到游戏的目的——盲棋。

这里我并不想讨论究竟哪种理解更合适，但是这两种理解对于游戏测试发展提出了两个大方向。下面我们就来说说游戏测试的三个层次。

1.2.1 常规软件测试

将游戏软件作为常规软件进行测试，这是游戏测试的第一个层次，也许很多人认为将此作为单独的一个层次是多余的，但这经常是被游戏测试人员所忽略的。游戏软件本身也是软件，这一点毋庸置疑，因此若想得到理想的游戏，就首先要保证能够得到一个符合需求的软件。

既然游戏软件也是软件，那么软件测试的各种方法和工具就没有道理不在游戏软件测试上延续。我们应该做的是在常规软件测试的基础上进行补充和完善；而不是彻底否定常规软件测试的内容，完全创造一个“全新的游戏软件测试”。

在抛开游戏的上层表现后，我们看到的就是一行一行的代码，它和其他软件在这点上没有任何差别，所以白盒测试完全可以照常开展；而最终的游戏软件也可以像其他软件一样，分解为一个个相对独立的功能，因此很多黑盒测试工作也可以在不考虑游戏特殊性的情况下进行。

1.2.2 游戏参数调节

在进行了常规软件测试的基础上，我们必须针对游戏的特点补充和完善测试内容，这就是游戏测试的第二个层次。相对于上述游戏性的阐述，这里主要进行的就是游戏具

体操作的测试。

这一部分测试会因为游戏的多样性而导致测试实现的表现多样性。但是排除明了的功能（如操作等）是否实现的检查外，留给测试人员更多的是具体细节的微调。游戏玩家通常都有这样的感觉，一款游戏流行后，就会出现很多模仿的同类游戏，但是仿品很难给玩家带来原版游戏一样的体会，这主要就是细节的微小差异造成的。

在一些游戏公司针对此部分内容已经设定了专门的岗位，如游戏数值平衡师等。不过这些岗位多数是出于设计目的，而非测试的检查目的。

1.2.3 游戏性测试

这里的游戏性指的是第二种理解，即游戏核心的性质。身为测试人员，从质量保证角度出发，应该尽量检测到软件的所有方面；自然对于游戏软件，游戏性的测试也是应该要覆盖的。但是关于这一层次的测试恐怕只能作为理想去追求，而不能奢望仅靠些努力就可以实现。

不过世事无绝对，自身的能力虽然总是有限度的，但是如果我们能够有效地借助外部资源，与对的人合作，就能够突破个人能力的极限，取得“意外”的效果。通过组织最终玩家试玩、试测，多听听最终玩家的意见，更多的做一个玩家的倾听者、游戏开发的建议者，其收效很可能比你个人的单打独斗好得多。

1.3 游戏自动化测试的可行性

我们谈到了游戏测试可以分为三个层次，最基本的就是将游戏看成一般的软件对其进行测试。常规软件的自动化测试概念中既包含了基于白盒思想的内部实现的自动检查，也包含了基于黑盒思想的外部功能表现的自动检查。

游戏测试对应地也可以通过上述两种思路开展自动化测试。下面一段脚本就是针对联想早期推箱子游戏地图中存在箱子数量和箱子目的地数量是否一致问题而编写的。基

本原理是游戏各个关卡的地图都存在于程序的 map.dat 文件中。直接读取地图文件，并统计各关中箱子的数量和箱子目的地的数量，不一致则报告。

map.dat 文件的结构是：第 0 关地图的宽，第 0 关地图的高，第 0 关地图按照从左到右、从上到下的格子类型序列；第 1 关地图的宽，第 1 关地图的高……其中格子类型用 02 表示该位置仅为箱子，03 表示该位置仅为箱子目的地，05 表示该位置为人和箱子目的地重合，07 表示箱子和箱子目的地重合。AutoIt3 工具实现的具体脚本如下：

```

$file=FileOpenDialog("请选择推箱子地图文件","", "地图文件 (*.dat) | 所有文件 _
(*.*)", 3, "map.dat")
$level=67 ;游戏中总关卡数为 0-67，如果关卡数改变，则对应改变该数值
$level=$level+1 ;因为游戏是从 0 关开始的，而不是从 1 开始的
$handle=FileOpen($file,16) ;以二进制文件方式打开
$log=""
For $i=1 To $level
    $width=Dec(StringRight(FileRead($handle,1),2)) ;得到地图的宽
    $high=Dec(StringRight(FileRead($handle,1),2)) ;得到地图的高
    $s_map=$width * $high
    $s_map=FileRead($handle,$s_map) ;读取地图
    $num_2=0
    $num_3=0
    $num_5=0
    $num_7=0
    For $len=1 to StringLen($s_map) Step 2 ;依次搜索统计相关格子数
        $s=StringMid($s_map,$len,2)
        Switch $s
            Case 2 ;仅为箱子
                $num_2+=1
            Case 3 ;仅为箱子目的地
                $num_3+=1
            Case 5 ;人和箱子目的地重合
                $num_5+=1
        EndSwitch
    Next
Next

```



```
Case 7 ;箱子和目的地重合
    $num_7+=1
EndSwitch
Next
$box=$num_2+$num_7 ;得到总箱子数
$Destinations=$num_3+$num_5+$num_7 ;得到总箱子目的地数
;如果该关的箱子总数不等于箱子目的地总数,则记录相关错误信息
If $box<>$Destinations Then $log=$log & "第" & $i-1 & "关" & @CRLF
& "箱子数:" & $box & @TAB & "目标数:" & $Destinations & @CRLF & @CRLF
Next
FileClose($handle)
MsgBox(0,"箱子与目的地数量错误",$log)
```

上述案例中游戏测试通过自动化直接检查地图文件内部数据的做法上,与常规软件的对应实现并无明显区别。可是若想通过游戏外部操作,像真正的玩家一样,自己玩游戏的自动化脚本却与常规的功能自动化测试有很大的区别。举例如下。

- 常规软件的构成控件多为标准控件,所以很多专业的自动化测试工具,可以直接基于控件的识别和操作,达到良好的自动化测试效果。而游戏软件经常使用自定义控件,在控件的状态识别和操作上,通常需要测试人员自己编写脚本来实现。虽然多数专业测试工具也提供了脚本编辑的功能,但是实现相对复杂,而且也浪费了这些专业工具的优势所在。
- 常规软件的自动化实现都是线性的,按照固定的操作顺序去执行被测试软件。虽然通过参数化使用数据池,或者插入检查点做一些判断,但是我们基本上可以把它归为不变化的一类。而游戏的特点就是很开放性的操作,不断变化的布局,在游戏里,变是绝对的,不变是相对的。
- 游戏自动化总是伴随着人工智能的讨论和类似游戏外挂实现的技术,所以需要比使用常规专业自动化测试工具更广的知识做基础,以更灵活的实现技术做支持,还需要更多的探索和尝试。

我们看到的这些区别，都是我们要实现游戏自动化测试的“拦路虎”。虽然困难重重，但是我们静下心来，有针对性地逐一解决，会发现各处都闪现着生机。

针对控件的识别与操作，能直接识别和操作的，自然就直接处理；若不能，至少我们可以退回到最原始的状态，通过屏幕显示来识别状态，模拟键盘和鼠标来控制游戏操作。

对于游戏的变，我们亦可以换个角度来思考，游戏中的变是万变不离其宗，所有的变化都要依据游戏规则。而计算机游戏中的规则就犹如“石头、剪子、布”游戏规则一样，清晰明了。这种清晰明了的规则，既适合编写游戏程序所使用，也同样适合编写自动化游戏脚本所使用。所以我们只需要加强一点开发的技能（优秀的测试人员，本身就应具备相当的开发能力），就完全有可能洞悉游戏变化，并跟随和适应游戏的变化。

而谈及人工智能和游戏外挂技术，自然没有那么简单。但是，谁说人没有翅膀就不可以飞？我们完全可以借助外部资源来辅助我们实现。并且通过合理地限定问题范围，仅实现满足测试需求的可接受解，将大大降低自动化测试的技术难度。

所以不仅是如普通程序一样自动化测试游戏是可行的，就如同玩家一样去自动化操作并测试游戏也是可行的。本书更想探讨的是后者的自动化测试实现，所以后面所讨论的范围主要是与玩家行为类似的自动化测试游戏的案例。