PC平台虚拟现实API自动化测试系统的研究与实现

徐怀宝 肖悦娱
(上海大学通信与信息工程学院 上海 200072)

摘 要：随着虚拟现实技术的发展，每天有大量运行在虚拟现实设备上的各类软件被开发，而对虚拟现实软件自动化测试系统的研究还很少。鉴于此，在研究自动化测试的基础上，结合工程实践设计并实现了一种PC平台的虚拟现实API自动化测试系统，同时针对虚拟现实API的测试要求，编写了相应的测试程序。系统在Windows操作系统上完成，通过编写批处理文件实现自动化测试。测试表明该系统高效的实现了从源程序下载到测试报告生成的全过程，有效提高了测试效率，能够更快更全的测出软件存在的问题。

关键词：虚拟现实；API；自动化测试；批处理
中图分类号：TP311.5；TN802 **文献标识码**：A **国家标准学科分类代码**：520.40

Research and implementation ofautomated virtual reality API test system based on PC platform

Xu Huaibao Xiao Yueyu
(School of Communication and Information Engineering, Shanghai University, Shanghai 200072, China)

Abstract: With the development of virtual reality technology, a large number of softwares running on virtual reality devices are developed every day. However, there is little research on virtual reality software automation testing system. Therefore, based on the study of automated testing, this paper designed and realized an automated virtual reality a PC-based API test system, combining with the engineering practice. At the same time, according to the test requirements of virtual reality APIs, the corresponding test program is written. The system is implemented on the Windows operating system, and the automatic test is realized by writing batch files. The test result shows that the system can realize the whole process from source code download to test report generation. Which improves the test efficiency effectively and faster and more fully find the software problems.

Keywords: virtual reality; application programming interface; automated testing; batch

0 引 言

2016年是虚拟现实(virtual reality，VR)元年，在这一年里，跟VR相关的软、硬件技术实现了质的飞跃[1]。一款好的VR设备不光需要好的硬件来支撑，更需要大量高质量的软件来驱动。因此，每天有大量运行在VR设备上的各类软件被开发，为了保证这些软件的质量，在正式发布之前，必须经过严格的测试。

在软件业较发达的国家，软件测试不仅成为软件开发的一个有机组成部分，而且在软件开发的系统工程中占据着相当大的比重。近些年来，国内也越来越重视软件的测试。由于手动测试费时费力、效率低下，因此自动化测试技术应运而生[2-4]。自动化测试就是用自动化测试工具或者其他手段，按照测试工程师预定计划进行自动测试，得到测试结果。

在现代的软件开发中，越来越多的使用C++等面向对象的高级语言，采用分层的软件体系结构。虚拟现实的软件开发工具包(software development kit，SDK)同样采用这种结构。在分层的结构中，应用程序编程接口(application programming interface，API)处在中间层，作为开发者和使用者的接口，它质量的好坏直接决定了整个软件质量的好坏[5]。一方面，每个VR公司都有自己版本的SDK，而这些SDK里包含了大量的API接口，采用人工手动测试的方法需要大量的时间和人力[6-8]。另一方面，现存的API自动化测试方法基本是针对Android移动平台而设

收稿日期：2017-01

计,而基于 Windows 操作系统针对 PC 平台的虚拟现实 API 自动化测试方法几乎是空白的[9-12]。目前,外接虚拟现实头戴式显示设备(简称 VR 头显)直接连接 PC,很多处理过程是在 PC 上进行,然后在 VR 头显上显示,而且绝大多数用户的操作系统为 Windows,这就要求在测试中要考虑 PC 处理器、显卡等硬件性能和 Windows 操作系统类型等,对于 Android 移动平台则无需关注这些 PC 配置。所以设计并实现针对 PC 平台的虚拟现实 API 自动化测试系统弥补了这方面的空白具有极大的实用价值和经济价值。

1 系统框架设计

PC 平台虚拟现实 API 自动化测试系统主要分为从 SVN 自动下载源程序、测试方法设计和测试测序编写、自动化编译和执行、测试结果生成和分析、网页文件生成、release 版本生成、获取测试环境等部分。它以 Batch 为引擎,在 Windows 操作系统上完成。整个测试系统框架如图 1 所示,首先从 SVN 把源程序下载到本地 PC 上,然后在分析源程序的基础上用 C++高级语言编写测试函数,API 的测试数据放在输入文本文档中,当 API 编译执行时自动从输入文本文档中读取,生成的测试结果保存在文本文档里,然后再用 Python 生成 XML 网页文件保存并呈现测试结果,最后用批处理文件自动生成软件的 release 版本。所有流程集成在一个 Batch 脚本里,只要双击这个脚本,就可以自动的完成以上所有流程,不但实现了高度的集成化,非常简单、易操作,而且各个部分通过编写批处理脚本调用执行,增加了各个部分的独立性,降低了耦合度[13]。

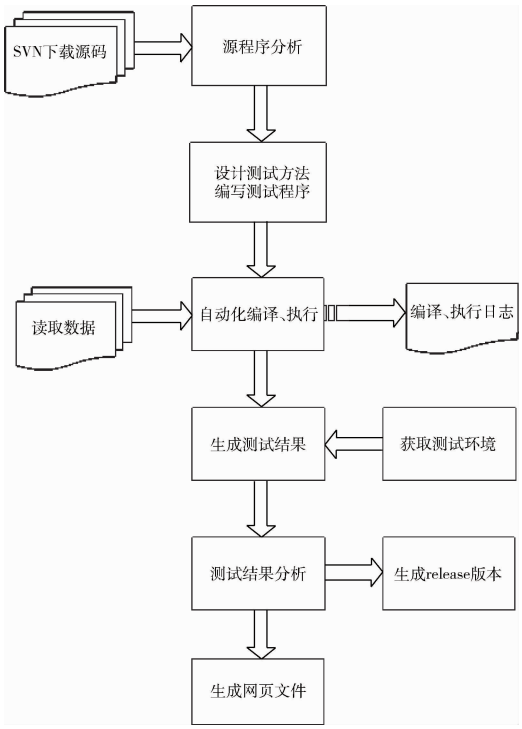

图 1 系统框架

2 系统各模块的实现

2.1 源程序下载模块

在分析、编译执行源程序前需要将源程序从 SVN 上下载下来,SVN 是 Subversion 的简称,是一个开放源代码的版本控制系统。下载用到的 Batch 语句为:

svncheckoutsvn://192.168.0.11/PlatformSDK/Api_Test ./Api_Test

这条语句的意思是从 IP 地址为 192.168.0.11 的主机上,把 PlatformSDK 文件夹下 Api_Test 里的源代码下载到本地当前路径下 Api_Test 文件夹里。

2.2 测试方法设计和测试程序编写模块

对于一般的 API,根据 API 的说明文档,编写对应的测试程序,调用相应 API,然后将需要传入 API 的相关数据(包括不同参数的组合、预期结果等)存放在输入文本文档中,在程序执行时打开输入文档自动读取对应的数据,传入 API 就可以了[14]。

不同于一般的 API,虚拟现实设备对延迟(20 ms 以内)有非常高的要求,测试必须要考虑到这一点。而且 VR SDK 里有很多一般 SDK 不具有的 API,针对这些 API,需要研究一些对应的测试方法,设计相应的测试用例。下面就介绍下这些特殊的 API 测试方法。

2.2.1 九轴传感器 API 测试方法

九轴传感器是 3 种传感器的组合,这 3 种传感器分别是 3 轴加速度传感器、3 轴陀螺仪和 3 轴地磁传感器(电子罗盘)。经数据融合后得到一个九轴的数据,主要反映 VR 设备的相对位置以判断 VR 设备的状态,根据这些状态系统会做出一些相应的反应。

要判定这个 API 是否符合要求,就要判断 API 返回的九个轴的数据是否正确。为此,设计一种自动测试装置,该装置包括 3 个部分:1)运动装置,包括用于承载并运送 VR 设备的传送皮带以及驱动传输皮带运行的步进电机,它有 X、Y、Z3 个方向的运动轴;2)定位传感器,设置在运送装置的初始端,用于检测传输带上的 VR 设备是否停留在预设零点和最大位置;3)控制装置,控制电机转速和转动角度,通过输入命令行的形式发送命令,使得 VR 装置运动到指定位置。

只要将 API 返回的 9 个轴的数据与自动测试装置上各个轴上返回的数据进行比较判断九轴传感器 API 返回的数据是否准确,就可以判断该 API 是否符合要求。

2.2.2 图像畸变率 API 测试方法

为了获得更好的沉浸感,VR 设备必须尽可能的覆盖人眼的视觉范围,为此,通常在 VR 设备上安装球面弧镜片。但是,经球面弧投射到人眼的图像是扭曲的,即产生了图像畸变,由于畸变的存在,图像根本没法正常观看。因此,在图像经球面弧之前,需要针对特定的镜片对图像进行反畸变变换。畸变的程度称之为畸变率并用专门的 API 实现并返回畸变率。对于畸变率是否准确,需要设计一个

专门的方法进行测试。

针对上面所述，将有一定畸变率的图像投在一个网格纸上，然后将经过畸变变换后的图像投到相同的网格纸上，比较两个图片的畸变程度是否相同，最后将经过球面弧的图像截取下来与原图像进行比较，以此来判断 API 是否符合设计要求。

2.2.3　声音信息 API 测试方法

VR 中，音频可以作为视觉反馈的辅助。为了有更好的沉浸感，不但图像要有 3D 效果，声音也需要 3D 的效果。传统的音频是提前渲染好的，不会根据头的移动而有所变化。而在 VR 中，VR 设备戴在头上，根据头的移动，声音的音源位置和强弱等应该产生相应的变化。这些不同的声音从不同的发音设备上播放出来，API 适时的返回这些信息。

为了验证 API 返回的信息是否准确，将每个播放设备上的音频的波形在示波器上显示出来，通过观察声音的波形，获取波形的振幅，可以得到声音强弱等信息，将得到的信息与 API 返回的信息进行对比，便可以迅速的判断该 API 是否符合设计要求。

2.3　自动化编译和执行模块

通过编写批处理脚本用 Microsoft Visual Studio 来编译执行，Microsoft Visual Studio 简称 VS，它是美国微软公司开发的工具包产品，包括了基本完整的开发工具集。自动化编译的 Batch 语句是：

devenv Api_Test.sln　/Build "debug|Win32"　/Project Api_Test

ping /n 1 /w 3000 9.9.9.9 1>nul

devenv 是 VS 的可执行程序，一般安装在“C:\Program Files (x86)\Microsoft Visual Studio 10.0\Common7\IDE”下。用来在命令行或 GUI 方式运行 VS。通常.sln 中的多个 Projects 间有依赖关系，所以虽然你只是 build 一个.sln 中的某个 Project，但是还是需要指定 Project 所在.sln，然后通过/Project 来指定 Project 的名字。/Build "debug|Win32"用来指定编译模式。ping 是用来测试本机网卡的好坏，/n 用来指定 ping 的次数，加 nul 就是不显示结果。编译完成后会生成一个 exe 文件。要执行程序只要运行生成的 exe 文件即可，相应的 Batch 语句为：

start/waitE:\ApiTest\OutPut\ \Debug\Api_Test.exe。

2.4　测试结果生成和分析模块

在每个 API 调用结束后会返回对应的测试数据，我们将 API 返回的测试数据与期望的结果进行对比分析，由于延时和刷新率等是 VR 质量好坏的重要指标，所以还要获得这些测试数据，最后根据这些数据进行综合分析，判断每一个 API 是否符合要求。通过则标记为 OK，不通过则标记为 ERROR，然后将这些测试数据保存在输出文本文档中。

2.5　网页文件生成模块

为了便于查看测试结果，通过编写 Python 程序将保存在文本文档里的测试结果自动的生成 XML 格式的网页文件，并且将最终生成好 XML 文件自动发送到指定的邮箱给相关人员查看。考虑到简单且便于重复利用，生成 XML 文件的方法是先写好需要的模板 XML 文件[15]，然后通过 Python 程序读取保存测试结果的文本文档里的测试数据，用读取到的数据替换模板 XML 文件中需要替换的地方则生成最终需要呈现在大家面前的网页文件。由于篇幅所限，图 2 是截取的测试报告总体测试情况的一部分。

API测试报告

项目：API test
部门：测试部
执行：徐怀宝
日期：2016-08-20 11:46:18

全部API

API	测试结果
GetVersion	OK
Init	部分测试用例未通过
GetDeviceInfo	OK
RecordPose	OK
UpdatePose	开发中
Compose	OK
Deinit	OK
GetUserData	OK
Resume	目前没有实现
SetIntValue	部分测试用例未通过
GetTime	OK

API总数	24	测试通过数	15	部分测试用例未通过数	4	开发中	3	目前没有实现	2

图 2　总体测试结果

2.6 release 版本生成模块

release 版本生成只要编写 Batch 文件将需要对外发布的源文件拷贝到对应文件夹,然后设置版本号即可,由于非常简单,这里不再赘述。

2.7 测试环境获取模块

由于 VR 设备对 PC 的软、硬件配置有相当高的要求,有时候在不同配置的 PC 上设备运行会出现不同的状况,所以有必要获取测试环境以作参考。通常需要关注的几个配置有操作系统、处理器、内存大小、显卡和显存大小。通过编写 Batch 脚本获得 PC 的配置并保存在文本文档中,然后通过 Python 程序将这些配置信息添加到生成的测试结果网页中以便查看。网页文件中显示的 PC 配置信息如图 3 所示。

测试环境	操作系统: Windows 7 旗舰版 64-bit (6.1, Build 7601) Service Pack 1 (7601.win7sp1_gdr.150202-1526) 处理器: Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz (4 CPUs), ~3.2GHz 内存:8GB 显卡: NVIDIA GeForce GTX 970 显存:4GB

图 3 PC 配置信息

3 应用实例

以一个实际 API 作为例子来简要说明测试系统的测试过程和结果。被测 API 的定义如下,它的功能是通知 runtime 记录头盔位置作为当前纹理所对应的摄像头位置已便 time warping 使用。

```
bool RecordPose(Instance instance, EyeTypeeye_type)
{
if (instance == NULL)
return FALSE;
returnRecordPoseP(instance, eye_type);
}
```

首先在分析源程序的基础上编写测试程序,由于篇幅所限,只列出部分代码,如下:

```
intRecordPoseTest()
{
ifstreaminfile("f:\\InPut\\RecordPose.txt");
ofstreamoutfile("f:\\OutPut\\api_out.txt", ios::app);
    ...
 Inst = Init(1, UM_DEFAULT, ctx-> gDev.GetInterfacePtr(), DEVICE_DX11, NULL);
if (RecordPose(instance, eye_type))
{
str1 = "OK";
outfile<< "OK" << "\t";
}
else
{
str1 = "ERROR";
outfile<< "ERROR" << "\t";
}
if (instance == NULL)
{
str2 = "ERROR";
outfile<< "ERROR" << "\t";
}
else
{
str2 = "OK";
outfile<< "OK" << "\t";
}
}
```

然后将输入的数据保存在输入文本中,设置 5 个测试用例,将测试结果保存在输出文本文档中,如图 4 所示。

```
【5 RecordPoseTest】 Notify runtime record helmet position
instance        eye_type                Actual result   Desired result  Final result

Inst            EYE_LEFT                ERROR           ERROR           OK
NULL            EYE_LEFT                ERROR           ERROR           OK
Inst            EYE_RIGHT               ERROR           ERROR           OK
Inst            EYE_CENTER              ERROR           ERROR           OK
Inst            EYE_LEFT_RIGHT          ERROR           ERROR           OK
Time delay: 18ms
```

图 4 保存在输出文本文档中的测试结果

最后用 Python 生成 XML 网页文件，如图 5 所示。

从测试结果可以看出，5 个测试用例输出与预期输出完全一致，延时 18 ms 也在预期之内，所以该 API 的设计是完全符合要求的。

API bool RecordPose(Instance instance, EyeType eye_type) 上一级

函数说明	功能：通知runtime记录头盔位置为当前纹理所对应的摄像头位置已便time warping使用 参数：instance　初始化函数返回值 eye_type

用例数	失败断言
5	0

2016-08-23 15:17:13

用例：1

instance	eye_type	Actual result	Desired result	Final result
Inst	EYE_LEFT	OK	OK	OK

用例：2

instance	eye_type	Actual result	Desired result	Final result
NULL	EYE_LEFT	ERROR	ERROR	OK

用例：3

instance	eye_type	Actual result	Desired result	Final result
Inst	EYE_RIGHT	OK	OK	OK

用例：4

instance	eye_type	Actual result	Desired result	Final result
Inst	EYE_CENTER	OK	OK	OK

用例：5

instance	eye_type	Actual result	Desired result	Final result
Inst	EYE_LEFT_RIGHT	OK	OK	OK

延时

延时	18ms

图 5　测试结果

4　结　　论

本文在结合实际工作经验和对自动化测试研究的基础上了，设计并实现了一种 PC 平台的虚拟现实 API 的自动化测试系统，此次设计在 Windows 操作系统上通过编写 Batch 脚本实现，整个测试只要双击一个 Batch 脚本即可以自动完成，实现了高度的集成化。该系统大大提高了人工测试的效率和成本，为 VR 软件的开发质量提供了强有力的保障，使开发者能够更好更快的完成软件开发的工作，具有巨大的经济和实用价值。

参考文献

[1]　杨欢，刘小玲. 虚拟现实系统综述[J]. 软件导刊，2016(4)：35-38.

[2]　贺晋宁，杜伟伟，高静. 软件自动化测试的探索实践[J]. 国外电子测量技术，2016(7)：1-4.

[3]　陈克浚. 国内软件自动化测试的现状及原因探究[J]. 电子技术与软件工程，2014(11)：81-81.

[4]　杨巍. 软件自动化测试系统的设计[J]. 科技传播，2014(18).

[5]　崔红军，饶若楠，邵培南. 一种 API 自动化测试工具的设计与实现[J]. 计算机工程，2007，33(4)：270-271.

[6]　孔丽文，薛召军，陈龙等. 基于虚拟现实环境的脑机接口技术研究进展[J]. 电子测量与仪器学报，2015，29(3)：317-327.

[7]　詹秦川，弓宸. 虚拟现实技术的应用及发展现状分析[J]. 产业与科技论坛，2014(7)：75-76.

[8]　张伟，东袁昊，周荫清，等. 基于虚拟仪器的电子测量系统[J]. 电子测量技术，2006，29(4)：44-45.

[9]　董娜娜，詹惠琴. 软件测试自动化技术应用研究[J]. 电子测试，2010(11)：47-50.

[10]　徐进. 自动化软件测试的分析[J]. 信息技术，2010(3)：152-155.

[11]　刘慕涛，张磊，王艳，等. 基于 XML 的 API 自动化测试工具设计与实现[J]. 计算机工程，2007，33(13)：96-98.

[12]　SPINELLIS D. Softwarebuilders. IEEE Software，2008，25(3)：22-23.

[13]　黄哲. 基于 Windows 的 API 自动化测试框架的设计与实现[J]. 科技传播，2010(18)：264，267.

[14]　王琨. 嵌入式计算机软件测试关键技术探讨[J]. 科技创新与应用，2016(7)：87-88.

（下转第 41 页）