AR Firework

白盒测试计划

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <12/04/2022> | <1.0> | 白盒测试计划 | 康艺潇，林舒怀，林祺龙，徐惠东 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 背景 4

1.3 范围 4

2. 资源 5

2.1 角色 5

2.2 测试模型 5

2.3 测试日志 6

2.4 缺陷报告 6

测试计划

# 简介

ARFirework项目由一款Android客户端教育应用程序和供科技馆使用的网页组成。该项目与上海科技馆达成合作，主要使用者是参观科技馆的儿童。其中的客户端App是一个交互式增强现实的教育应用，将AR技术用于模拟危险的实验室环境和可视化抽象概念。在与展品互动的过程中，用户创作出以烟花为特色的DIY纪念品，也能更深刻的学习科学知识。虚拟AR模型与实体礼盒绑定，参观者可以将其作为礼物赠送给他人，也可以在数字画廊中展示。这个纪念品通过数字媒体在游客和他们的个人参观体验之间建立了长久的联系。该研究促进了对纪念品个性化、社会化、AR和STEAM教育的理解。

提供给科技馆使用的网页主要用于生成二维码，提供给游客进行烟花数据的上传和下载，实现烟花场景的分享。

燃放烟花的过程中涉及到许多易燃易爆的危险化学反应，也产生了对人体和环境有危害的污染废气废料，不适宜在中学实验室中实际操作观察。我们针对这一痛点，提出在上海科技馆内用AR技术来完成实验，既将实验结果可视化、可感知化，又确保了实验安全。

## 目的

这一“测试计划”文档有助于实现以下目标：

本文档针对AR Firework的白盒测试，记录相关测试内容，测试方法和测试环境。

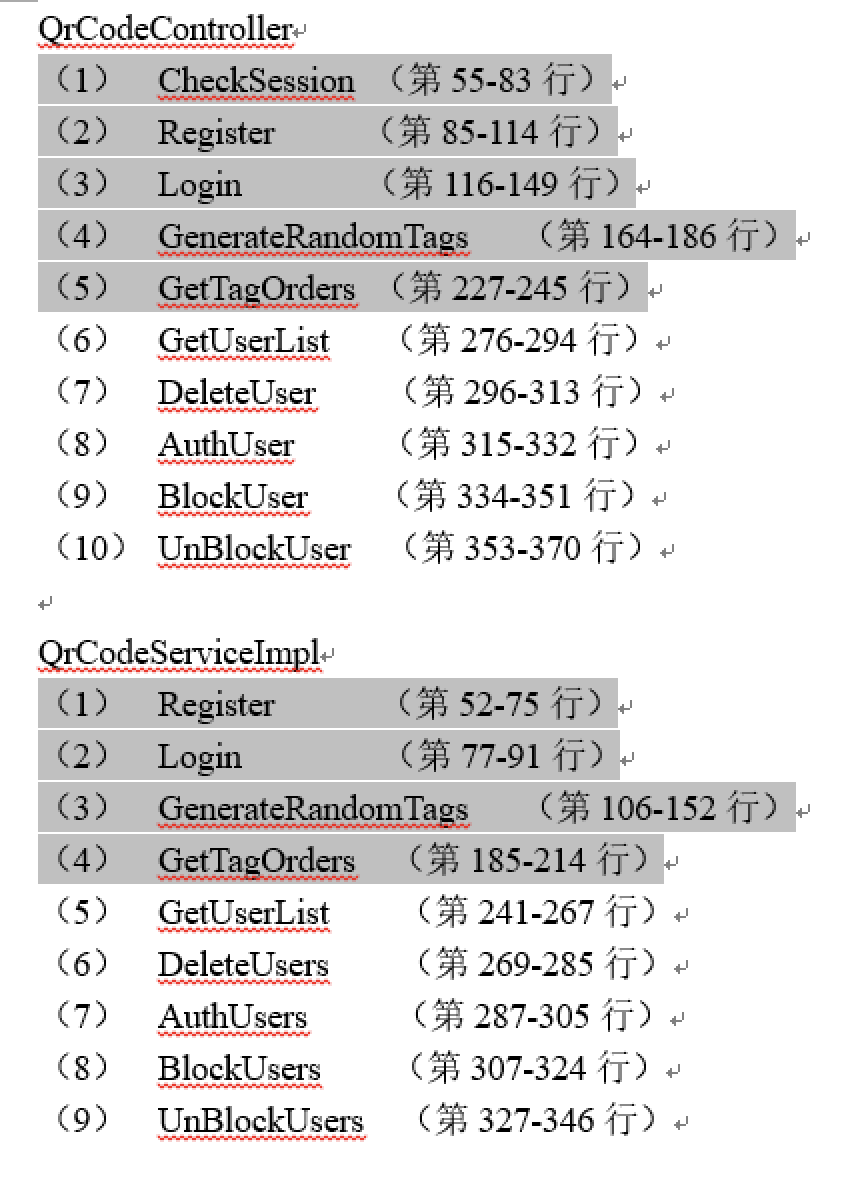
* 列出推荐的测试需求（高层次）。
* 本次测试需要在测试环境部署该项目，需要4台4核8G内存的服务器，需要测试服务器1台4和8G服务器。
* 本次测试分别使用的方法为基于DD路径的测试方法、和基于数据流的测试方法.
* 同时，我们会在测试报告中详细说明了这些测试用例设计的过程、原因和得出的结论。

## 背景

增强现实技术在教学中的应用潜力巨大、前景广阔，主要体现在运用增强现实技术具有激发学习动机、创设学习情境、增强学习体验、感受心理沉浸、跨越时空界限、动感交互穿越和跨界知识融合等多方面的优势。增强现实技术的应用，能够为教育工作者提供全新的教学工具，同时，能激发学生学习新知识的兴趣，让学生在动手体验中迸发出创新的火花。随着人工智能时代的到来，学生在基础教育阶段具备学科核心素养和创新思维能力，树立终身学习的方法和意识变得尤为重要。STEAM代表科学（Science），技术（Technology），工程（Engineering），艺术（Arts），数学（Mathematics）。STEAM教育就是集科学，技术，工程，艺术，数学多领域融合的综合教育。STEAM 教育源于美国，强调学科知识的有机融合和学生综合素养的培育。通过比较研究和实践探索，构建基于STEAM教育理念的创新课程体系，并为教师实施跨学科整合、开展创新教育实践等教学环节提供借鉴。

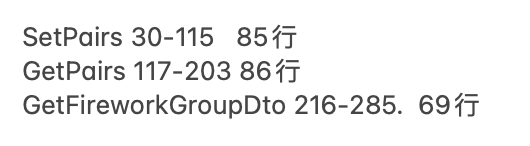
我们在调研中学实验中发现，部分实验具有一定的危险性，且存在难以直接观察的现象。采用增强现实技术进行虚拟实验可以克服该问题。我们搭建的基于增强现实技术的STEAM教育及虚拟实验平台可以对抽象概念可视化，让同学们有更直观的理解。虚拟实验也弥补了一些危险的化学、电学实验的弊端，从而给中学实验带来了新思路。

## 范围



测试分为后端+前端应用代码。测试了后端的9个功能，分别对应QrCodeController和QrCodeServiceImpl的18个函数。其中图中灰色部分主要为登陆和二维码相关功能，白色部分为用户管理功能。

前端测试了Unity的逻辑实现代码如下：



针对每个函数我们都设计基于DD路径的测试和基于数据流的测试。

# 资源

* 服务器资源：使用4个4核8G内存服务器进行部署，1个4核8G服务器进行测试。

## 角色

角色分工略去，每个人都参与到具体的用例设计之中。

下表表列出了测试项目所需的系统资源。

|  |  |
| --- | --- |
| 系统资源 | |
| 资源 | 名称/类型 |
| 数据库服务器 | 无 |
| 网络或子网 | 不使用网络 |
| 服务器名 | 无 |
| 数据库名 | 无 |
| 客户端测试 PC | Windows 10 |
| 包括特殊的配置需求 | 无 |

项目里程碑

|  |  |  |  |
| --- | --- | --- | --- |
| **里程碑任务** | **工作量** | **开始日期** | **结束日期** |
| 制定测试计划 | 10 | 2022年4月5日 | 2022年4月5日 |
| 设计测试 | 20 | 2022年4月5日 | 2022年4月5日 |
| 实施测试 | 20 | 2022年4月10日 | 2022年4月12日 |
| 执行测试 | 10 | 2022年4月10日 | 2022年4月12日 |
| 评估测试 | 10 | 2020年4月13日 | 2020年4月13日 |

可交付工件

* 交付的文件如下：

1. 《测试计划》
2. 《需求规格说明书》
3. 《测试报告》
4. 其他附件

交付时间为2022年4月15日

* 交付的工具如下：

测试代码

交付时间为2022年4月15日

## 测试模型

### 基于DD路径的测试

语句覆盖即测试用例覆盖所有执行语句；

条件覆盖即测试用例覆盖每一个条件；

多条件覆盖即测试用例覆盖每种条件组合；

循环覆盖即测试用例覆盖不循环、1次循环、2次循环、多次循环等情况；

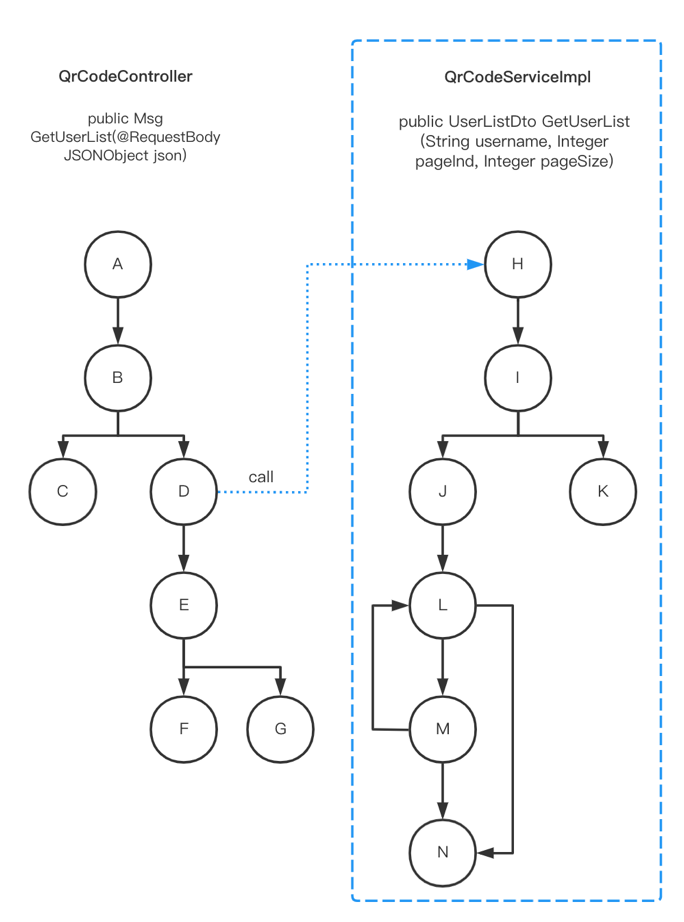
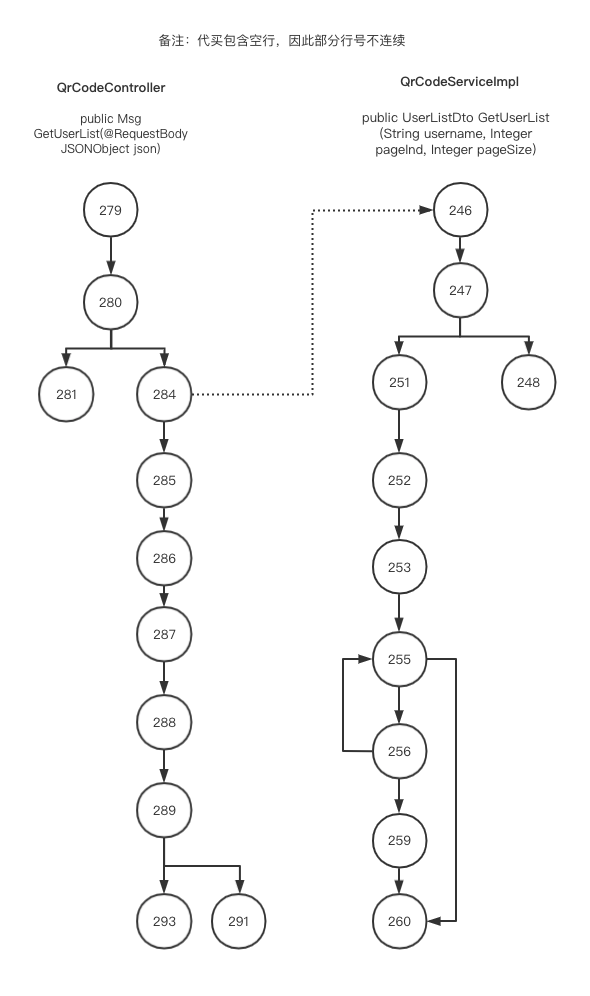
包含最多2次循环的所有路径意义显然；

路径测试即测试用例要覆盖所有可能执行的路径。

测试用例见附件【DD测试用例表.xlsx】

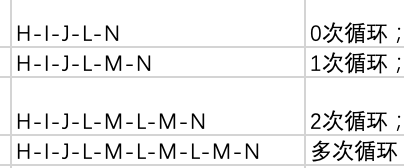
下以GetUserList功能举例说明：

首先路径图，之后进行合并和简化



程序路径图 DD路径图

对这个功能采用循环覆盖+语句覆盖，即覆盖所有的节点，对于循环L-M, 采取0次循环，1次循环，2次循环，多次循环的测试



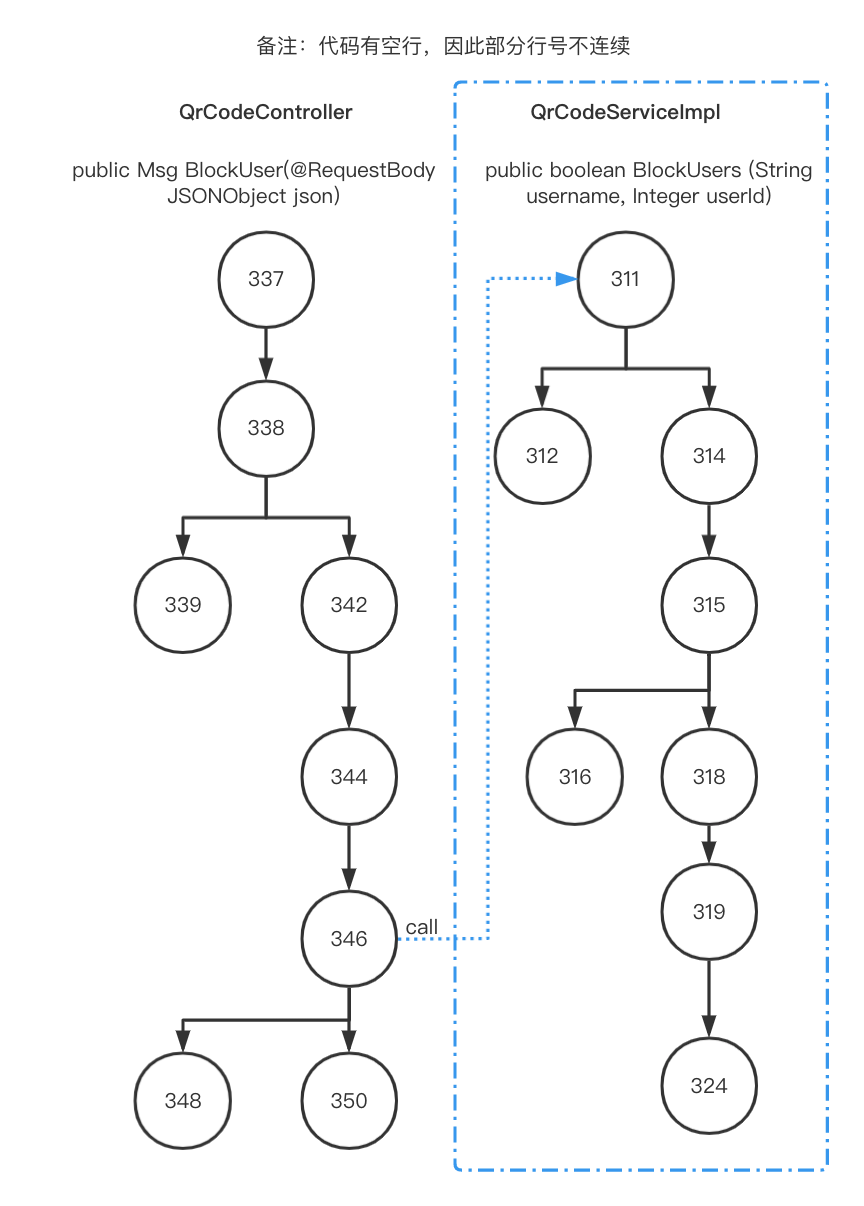
### 基于数据流的测试

测试用例见附件【数据流测试用例表.xlsx】

对于每个函数，我们又采用基于数据流的测试用例生成方法。我们选取其中的QrCodeController. BlockUsers函数作为代表，详细介绍基于数据流的测试用例的生成过程。而其他函数的测试用例生成过程略去，具体材料我们放在附件之中。

需要注意的是，为了方便测试和展示，在本小节中，我们以行号作为程序图的节点，由于函数中存在空行，图中相邻节点的行号不一定连续。

1. 绘制目标函数的程序图



1. 统计变量的定义/使用节点

在对程序图进行绘制后，我们可以对QrCodeController. BlockUsers中的所有变量的定义/使用节点进行统计：

|  |  |  |
| --- | --- | --- |
| 变量名 | 定义节点 | 使用节点 |
| json | 335 | 344 |
| sessionJson | 337 | 342 |
| username | 342 | 346 |
| userId | 344 | 346 |

1. 列出变量的定义-使用路径

在列出所有变量的定义/使用节点后，我们可以依据简化程序图的节点情况，依次列出所有变量的定义-使用路径，因为部分路径存在重合部分，因为我们对路径进行了整理，图中列出各个变量的测试路径：

|  |  |  |  |
| --- | --- | --- | --- |
| 变量名 | 定义节点 | 使用节点 | 测试路径 |
| json | 335 | 344 | 335-344 |
| sessionJson | 337 | 342 | 337-342 |
| username | 342 | 346 | 342-246 |
| userId | 344 | 346 | 344-346 |

1. 选取测试覆盖指标，生成测试用例

在这一步，我们选用全使用准则作为测试覆盖指标，即选择每一个定义节点到所有使用节点以及其后续节点的定义-清除路径。而对于所有的定义-使用路径，我们用尽可能少的测试用例去覆盖尽可能多的目标路径。具体的生成的测试用例，我们放在附件中予以展示，此处略去。

1. 编写代码，进行测试

步骤1-4就是基于数据流的测试用例生成过程，接下来我们可以编写代码，进行测试。值得一提的是，因为对于数据流测试的测试覆盖指标本身就是我们测试用例生成的依据，因此按照全使用准则，我们的覆盖率必然是100%，但是在其他覆盖率（比如条件覆盖率、分支覆盖率等）语境下，覆盖率会有所下降。

## 测试日志

详见测试报告

## 缺陷报告

详见测试报告