|  |  |
| --- | --- |
|  | **第十七届全国大学生软件创新大赛** |
| **文档编号：SWC2024-Hi安卓** |



**按键精灵**

**AutoDroid**

**项目测试文档**

**Version: 0.6.0**



**Hi安卓**

**2024-03-19**

**All Rights Reserved**

目录

[1测试计划 1](#_Toc161744128)

[1.1 测试策略与目标 1](#_Toc161744129)

[1.2 测试范围 3](#_Toc161744130)

[1.3 测试环境 3](#_Toc161744131)

[2单元测试 5](#_Toc161744132)

[2.1 LLM交互模块 5](#_Toc161744133)

[2.1.1测试用例与结果分析 5](#_Toc161744134)

[2.1.2测试结果综合分析及建议 10](#_Toc161744135)

[2.1.3测试经验总结 10](#_Toc161744136)

[2.2 APP信息获取模块 10](#_Toc161744137)

[2.2.1测试结果综合分析及建议 11](#_Toc161744138)

[2.2.2测试经验总结 11](#_Toc161744139)

[3功能测试 12](#_Toc161744140)

[3.1 APP跳转与模拟动作执行功能 12](#_Toc161744141)

[3.1.1测试用例与结果分析 12](#_Toc161744142)

[3.1.2测试结果综合分析及建议 13](#_Toc161744143)

[3.1.3测试经验总结 13](#_Toc161744144)

[4系统测试 14](#_Toc161744145)

[4.1 模型性能测试 14](#_Toc161744146)

[4.1.1测试用例与结果分析 14](#_Toc161744147)

[4.1.2测试结果综合分析及建议 15](#_Toc161744148)

[4.1.3测试经验总结 15](#_Toc161744149)

文档修订历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **修订原因** | **版本号** | **作者** | **修订日期** | **备注** |
| **1** | **初稿** | **0.1.0** | **Hi安卓队** | **2024.01.15** |  |
| **2** | **对测试目标进行修改** | **0.2.0** | **Hi安卓队** | **2024.01.31** |  |
| **3** | **对硬件环境进行完善** | **0.3.0** | **Hi安卓队** | **2024.02.27** |  |
| **4** | **编写了单元测试的测试用例** | **0.6.0** | **Hi安卓队** | **2024.03.18** |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 测试计划

## 测试策略与目标

（1）测试策略

项目测试分括单元测试、功能测试和系统测试三个阶段。

|  |  |
| --- | --- |
| **单元测试** | |
| **测试目标** | 检测代码单元的逻辑功能 |
| **测试范围** | 最小模块 |
| **开始标准** | 编码完成 |
| **完成标准** | 成功地核实出各个软件组成单元都与 基准版本保持一致，  或符合可接受标准 |
| **测试重点** | Android APP、API接口测试、局部数据处理测试、路径测试、错误处理测试、边界测试。 |

表 1 单元测试表

|  |  |
| --- | --- |
| **功能测试** | |
| **测试目标** | 确保测试的功能正常，其中包括数据输入，动作生成，模拟GUI交互等功能。 |
| **测试范围** | 项目全部功能点 |
| **开始标准** | 功能开发完毕 |
| **完成标准** | 成功地核实出各个功能点都与基准版本保持一致，或符合可  接受标准 |
| **测试重点** | 用例或业务功能和业务规则的所有测试需求，通过图形界面  验证应用程序及其内部进程。 |

表 2 功能测试表

|  |  |
| --- | --- |
| **系统测试** | |
| **测试目标** | 对功能、性能以及软件所运行的硬软件环境进行测试。 |
| **测试范围** | 整个系统（软、硬件） |
| **开始标准** | 系统开发完成 |
| **完成标准** | 成功地核实系统运行满足的所有需求 点都与需求规格说明文档保持一致， 或符合可接受标准。 |
| **测试重点** | 系统功能、界面、可靠性、易用性、性能、兼容性、安全等 |

表 3 系统测试表

（2）测试目标：

| **测试目标** | **描述** |
| --- | --- |
| 功能测试 | 确保所有功能模块的正常运作，包括但不限于登录、注册、数据录入、搜索、支付等。 |
| 用户界面  测试 | 确保界面设计符合用户体验标准，包括布局、颜色、字体、图标等方面的一致性和美观性。 |
| 兼容性测试 | 确保应用在不同版本的 Android 操作系统上以及各种分辨率和屏幕尺寸的设备上能够正常运行。 |
| 性能测试 | 测试应用在各种网络条件下的响应速度、资源消耗情况，确保应用的流畅性和稳定性。 |
| 安全性测试 | 检查应用的数据传输加密、用户认证机制、防止恶意攻击和数据泄露的措施等安全功能是否有效。 |
| 国际化和  本地化测试 | 确保应用在不同语言环境下的文本、日期、时间、货币格式等显示和处理正确。 |
| 用户体验测试 | 通过用户行为和反馈数据评估用户在应用中的体验，包括易用性、可访问性和满意度等方面。 |
| 自动化测试 | 开发自动化测试脚本，提高测试效率和覆盖率，确保持续集成和持续交付过程的稳定性。 |
| 冒烟测试和  回归测试 | 在发布前执行冒烟测试确认核心功能是否可用，并在每次更新后执行回归测试以确保之前功能的稳定性。 |
| 用户反馈和  Bug跟踪 | 收集用户反馈并及时响应，追踪和管理 Bug，确保问题得到及时修复和处理。 |

表 4 测试目标表

| **质量等级** | **描述** |
| --- | --- |
| 高质量 | 所有功能经过全面测试，少于5%的缺陷，用户体验良好，性能稳定，安全性强，满足预期的所有需求。 |
| 中等质量 | 所有功能经过测试，5%-10%的缺陷，用户体验良好，性能较稳定，安全性一般，满足大部分需求。 |
| 低质量 | 所有功能经过测试，10%-20%的缺陷，用户体验受到一定影响，性能不稳定，存在安全隐患，需进一步优化。 |
| 不可接受 | 应用无法正常运行，超过20%的严重功能缺陷、性能问题或安全漏洞，无法满足用户需求，需立即修复。 |

表 5 质量等级表

## 测试范围

| **测试项目** | **测试广度** | **测试深度** |
| --- | --- | --- |
| 功能测试 | 所有主要功能模块及其子功能的覆盖 | 每个功能的各种可能情况下的测试，包括正常操作、边界情况和异常情况等 |
| 用户界面  测试 | 所有界面及其元素的覆盖 | 界面布局、颜色、字体、图标等细节的测试 |
| 兼容性测试 | 不同版本 Android 操作系统和各种分辨率、屏幕尺寸的设备覆盖 | 在各种设备上的功能性测试和界面适配测试 |
| 性能测试 | 不同网络条件下的响应速度和资源消耗测试 | 响应时间、内存占用、CPU占用等指标的详细测试 |
| 安全性测试 | 数据传输加密、用户认证机制、防止恶意攻击和数据泄露的测试 | 渗透测试、密码破解测试、数据篡改测试等 |
| 国际化和  本地化测试 | 不同语言环境下的文本、日期、时间、货币格式等显示和处理的测试 | 文本长度、特殊字符、不同日期格式、货币符号等测试 |
| 用户体验  测试 | 用户操作流程和反馈数据的全面覆盖 | 通过用户行为和反馈数据评估用户体验 |
| 自动化测试 | 核心功能和关键路径的自动化测试 | 自动化测试脚本的编写和执行，测试覆盖率和稳定性的评估 |
| 冒烟测试  和回归测试 | 核心功能的覆盖以及每次更新后的回归测试 | 核心功能的正常运行确认、之前功能的稳定性验证 |
| 用户反馈和 Bug 跟踪 | 用户反馈和 Bug 报告的收集和管理 | Bug的追踪和分类、用户反馈的整理和处理 |

表 6 测试范围表

## 测试环境

**（1）软硬件环境：**

1. 硬件环境：
   * 手机：OPPO手机。
   * PC机：处理器Intel i5-11，内存16GB，磁盘500GB。
   * GPU 服务器：8卡80GB A100 GPU。
2. 软件环境：
   * PC 操作系统：Windows10
   * Android 操作系统：Android 9，Android 10，Android 11.
   * 开发工具：Android Studio，用于应用的编译、构建和调试。
   * 测试辅助工具：ADB (Android Debug Bridge) 用于设备管理和调试，Android Emulator 用于虚拟设备测试。

**（2）网络环境：**

* 下载：4.42 Mbps
* 上传：6.49 Mbps

**（3）测试工具：**

1. 自动化测试工具：
   * DroidBot、Humanoid、Monkey等用于自动化功能测试。
2. 其他工具：
   * GitHub用于 Bug 管理和团队协作。

# 单元测试

## LLM交互模块

### 测试用例与结果分析

* 单元测试用例：UI界面转HTML形式

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 001 | | | |
| 测试单元描述 | 将Android端 xml形式的界面表示转换为等价的HTML描述 | | | |
| 用例目的 | 生成LLM的可以理解的GUI描述 | | | |
| 前提条件 | 任务开始时或AutoDroid执行完一次动作 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 具体步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | <android.widget.TextView>MAIN SCREEN </android.widget.TextView> | <input>MAIN SCREEN</input> | <input>MAIN SCREEN</input> | TextView标签属于输入标签，得到的输出就是input |
| 2 | <android.view.ViewGroup>我的视频</android.view.ViewGroup> | <p>我的视频</p> | <p>我的视频</p>  或者空白 | 除了可点击的button以及可输入的文本标签之外，都统一归类为无影响的<p> |
| 3 | <android.widget.ImageButton>展示图片</android.widget.ImageButton> | <button>展示图片</button> | <button>展示图片</button> | 可点击标签统一设置为<button> |

表 7 用例001表

测试结果分析：对以xml形式表示的用户图形界面，能够将其转化为等价的HTML形式。

* 单元测试用例：提示词生成

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 002 | | | |
| 测试单元描述 | 根据当前用户界面的HTML描述、用户任务以及 APP Memory生成提示词 | | | |
| 用例目的 | 生成LLM的可以理解的提示词，以此询问LLM来获取下一步的动作 | | | |
| 前提条件 | 任务开始时或AutoDroid执行完一次动作 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 具体步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | <p id=0 bound\_box=28,634,1052,756>Add a new note</p> <input id=1 bound\_box=72,804,1008,957>Label</input> <p id=2 bound\_box=72,1001,1008,1076>New note type:</p> <checkbox id=3 bound\_box=72,1076,1008,1197>Text note</checkbox> <checkbox id=4 status=selected bound\_box=72,1197,1008,1318>Checklist</checkbox> <button id=5 bound\_box=658,1329,843,1461>Cancel</button> <button id=6 bound\_box=843,1329,1019,1461>OK</button> | You are a smartphone assistant to help users complete tasks by interacting with mobile apps. Given a task, the previous UI actions, and the content of current UI state, your job is to decide whether the task is already finished by the previous actions, and if not, decide which UI element in current UI state should be interacted. Task: Create a new text note "todo".  Previous UI actions:  - Start the Notes app.  - Click the Add button.  Current UI state:  <p id=0 bound\_box=28,634,1052,756>Add a new note</p>  <input id=1 bound\_box=72,804,1008,957>Label</input>  ……  <button id=6 bound\_box=843,1329,1019,1461>OK</button>  Which action should you choose next? Fill in  the blanks about the next one interaction:  - id=<id number> - action=<tap/input> - input text=<text or N/A>. (if you think the task has been completed, the id should be -1. | You are a smartphone assistant to help users complete tasks by interacting with mobile apps. Given a task, the previous UI actions, and the content of current UI state, your job is to decide whether the task is already finished by the previous actions, and if not, decide which UI element in current UI state should be interacted. Task: Create a new text note "todo".  Previous UI actions:  - Start the Notes app.  - Click the Add button.  Current UI state:  <p id=0 bound\_box=28,634,1052,756>Add a new note</p>  <input id=1 bound\_box=72,804,1008,957>Label</input>  ……  <button id=6 bound\_box=843,1329,1019,1461>OK</button>  Which action should you choose next? Fill in  the blanks about the next one interaction:  - id=<id number> - action=<tap/input> - input text=<text or N/A>. (if you think the task has been completed, the id should be -1. | 无 |
| … | … | … | … | … |

表 8 用例002表

测试结果分析：能够根据当前用户界面的HTML描述、用户任务以及 APP Memory来生成一个通过 APP Memory prompt。

* 单元测试用例：基于LLM的动作生成

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 003 | | | |
| 测试单元描述 | 使用经过 APP Memory 增强过的 prompt来询问LLM，获取下一步执行的动作 | | | |
| 用例目的 | 生成完成任务所需要的下一步的动作 | | | |
| 前提条件 | 任务开始时或AutoDroid执行完一次动作 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 具体步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | You are a smartphone assistant to help users complete tasks by interacting with mobile apps. Given a task, the previous UI actions, and the content of current UI state, your job is to decide whether the task is already finished by the previous actions, and if not, decide which UI element in current UI state should be interacted. Task: Create a new text note "todo".  Previous UI actions:  - Start the Notes app.  - Click the Add button.  Current UI state:  <p id=0 bound\_box=28,634,1052,756>Add a new note</p>  <input id=1 bound\_box=72,804,1008,957>Label</input>  ……  <button id=6 bound\_box=843,1329,1019,1461>OK</button>  Which action should you choose next? Fill in  the blanks about the next one interaction:  - id=<id number> - action=<tap/input> - input text=<text or N/A>. (if you think the task has been completed, the id should be -1. | - id=1  - action=input  - input text=todo | - id=1  - action=input  - input text=todo | 无 |
| … | … | … | … | … |

表 9 用例003表

测试结果分析：能够根据当前用户界面的信息以及用户的任务生成匹配的动作。

### 测试结果综合分析及建议

该测试用例覆盖了Android后端中和LLM的交互过程。包括prompt生成的函数以及询问LLM来获取下一步动作的函数。能够准确提取出用户图形界面中的组件，并将其转换为与之对应的HTML组件，并形成一个HTML的界面描述。并且可以根据APP Memory对prompt进行增强，并通过其来询问LLM获取下一步的动作。

### 测试经验总结

本次测试经验表明，测试用例的设计要尽可能全面，覆盖各种场景，以确保功能的可靠性和准确性。在后续开发过程中，我们将继续积累测试经验，提高测试效率，为项目的顺利进行提供保障。

## APP信息获取模块

* 单元测试用例：用户手机APP信息收集与展示

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 004 | | | |
| 测试单元描述 | 获取用户手机上的所有app的名称、包名以及启动活动，并在前端界面展示所有app的列表 | | | |
| 用例目的 | 给用户提供一个可选择来执行任务的app列表，以及启动该app的活动，在用户点击执行任务按钮时，来跳转到对应的app | | | |
| 前提条件 | app首次启动 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 具体步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | packageManager | 百度地图：com.baidu.BaiduMap/com.baidu.baidumaps.WelcomeScreen.Alias0/com.baidu.baidumaps.WelcomeScreen  百度翻译：com.baidu.baidutranslate/com.baidu.baidutranslate.activity.SplashActivity/null  …… | 百度地图：com.baidu.BaiduMap/com.baidu.baidumaps.WelcomeScreen.Alias0/com.baidu.baidumaps.WelcomeScreen  百度翻译：com.baidu.baidutranslate/com.baidu.baidutranslate.activity.SplashActivity/null  …… | 无 |
| … | … | … | … | … |

表 10 用例004表

测试结果分析：能够准确得获取当前用户手机上的所有app的名称、包名以及启动活动的信息，并且能够在前端成功展示所有app的名称列表，供用户选择。

### 测试结果综合分析及建议

该测试用例覆盖了Autodroid启动时对用户手机上的app信息收集的函数，能够成功获取用户手机上的所有app的名称、包名以及启动活动的信息。并且能够在Android通过下拉框的形式成功展示所有的app的名称。

### 测试经验总结

本次测试经验表明，在开发过程中，针对关键功能进行单元测试是十分必要的。通过单元测试，我们可以发现并解决潜在的问题，提高代码质量和功能稳定性。

# 功能测试

## APP跳转与模拟动作执行功能

### 测试用例与结果分析

* 测试用例：跳转至用户选择的app

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | Function\_test\_001 | | | |
| 功能描述 | 从AutoDroid，跳转到用户选择的执行任务的app | | | |
| 用例目的 | 跳转到对应app去执行任务 | | | |
| 前提条件 | 用户点击“执行任务”按钮 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 具体步骤 | 输入 | 期望结果 | 实际结果 | 备注 |
| 1 | 京东 | com.jingdong.app.mall.main.MainActivity | com.jingdong.app.mall.main.MainActivity | 无 |
| 2 | 抖音 | com.ss.android.ugc.aweme.splash.SplashActivity | com.ss.android.ugc.aweme.splash.SplashActivity |  |
| … | … | … | … | … |

表 11 用例Function\_test\_001表

测试结果分析：能够根据用户所选择的app，跳转到对应的app的初始界面。

* 测试用例：执行模拟动作

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | Function\_test\_002 | | | |
| 功能描述 | 根据LLM的返回，在用户界面的对应组件上执行相应的动作 | | | |
| 用例目的 | 执行动作以逐步完成任务 | | | |
| 前提条件 | LLM产生对当前用户界面的回复 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 具体步骤 | 输入 | 期望结果 | 实际结果 | 备注 |
| 1 | InputEvent | True | True | 动作执行成功返回True，否则返回False |
| 2 | ClickEvent | True | True | 动作执行成功返回True，否则返回False |
| … | … | … | … | … |

表 12 用例Function\_test\_002表

测试结果分析：能够根据LLM的回复信息，提取出相应的组件以及相应的动作并且执行。

### 测试结果综合分析及建议

AutoDroid能够成功跳转到用户所选择的app当中，并且能够从LLM的输出中提取出完成当前任务所需要操作的组件和动作，并进行执行。

### 测试经验总结

首先，我们认识到测试用例需要尽可能全面，覆盖各种场景和情况。例如，在测试过程中，我们不仅关注了常规情况，还特别设计了针对边缘情况和异常情况的测试用例。这样做可以大大提高测试的覆盖率，从而更全面地评估系统的性能和稳定性。

再次，测试数据的充分性对于评估系统的稳定性和容错性具有重要意义。我们使用了多种不同的测试数据，包括正常收集的APP Memory数据、不全面的 APP Memory数据，以确保系统能够在各种情况下都能够正常运行。

最后，测试记录的详细性对于追踪和解决问题至关重要。我们详细记录了测试过程中的一切相关信息，包括测试人员、测试时间、测试用例、测试结果等。这样做可以使得问题追踪和解决更加高效，同时也有利于后续的测试工作。

综上所述，通过本次测试，我们深刻认识到了测试用例全面性、测试数据充分性以及测试记录详细性对于确保系统稳定性和可靠性的重要性。在未来的测试工作中，我们将继续坚持这些原则，不断提高测试质量，为项目的顺利进行提供有力保障。

# 系统测试

## 模型性能测试

### 测试用例与结果分析

* 预期性能指标测试用例：动作执行效率

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | System\_test\_001 | | | |
| 性能描述 | 每次动作执行的平均时间 | | | |
| 用例目的 | 验证AutoDroid在用户手机上执行动作时的性能表现，包括LLM响应的时间以及执行动作所需的时间。 | | | |
| 前提条件 | 用户手机网络通畅 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 具体步骤 | 输入/动作 | 期望的性能  （平均值） | 实际的性能  （平均值） | 备注 |
| 1 | 询问LLM获取动作 | ≈3s | ≈3s |  |
| 2 | 执行动作 | ≈1s | ≈1s |  |
| … | … | … | … | … |

表 13 用例System\_test\_001表

测试结果分析：单词动作的执行时间（包括LLM相应的时间以及执行动作所需的时间）在5s之内。

* 预期性能指标测试用例：任务完成成功率

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | System\_test\_002 | | | |
| 性能描述 | 完成用户任务的成功率 | | | |
| 用例目的 | 验证AutoDroid能够根据用户的移动任务，在对应app上执行相对应的动作来完成任务。 | | | |
| 前提条件 | 用户手机网络通畅 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 具体步骤 | 输入/动作 | 期望的性能  （平均值） | 实际的性能  （平均值） | 备注 |
| 1 | LLM判断任务完成，并且达到用户任务的要求 | 大于80%的任务完成率 | 90.9%的任务完成率 |  |
| … | … | … | … | … |

表 14 用例System\_test\_002表

测试结果分析：AutoDroid能够完成90.9%的用户任务，其中：路径和真实值不同，但是结果相同也算作成功完成。

### 测试结果综合分析及建议

AutoDroid能够以可接受的时间和成功率来完成用户所提出的任务。

### 测试经验总结

LLM的响应受网络影响，在测试时，应该保持网络通畅。