**原型系统介绍：安卓平台录制与重放工具**

1. **安卓平台录制与重放工具**

移动设备的迅猛发展为移动应用程序（即通常所称的app）的开发及测试都提出了较高的要求。为使得app在发布前得到充分的测试，测试人员通常需要在尽可能多的移动设备上进行测试，以排除不同尺寸或分辨率的屏幕、不同版本的操作系统等因素的影响。录制与重放工具在测试过程中扮演着重要的角色，因为其可以对用户的操作进行多次、跨设备的重放，为应用的调试，测试和维护提供了便利。app相比于传统桌面软件拥有更为丰富的输入（如手势、按键、传感器等）。因此，传统PC端平台上的录制与重放工具难以满足移动平台的需求。这些额外的输入同时为其测试带来了新的挑战。

近年来，学术界的研究人员和工业界的从业者提出了不少的移动设备上的录制与重放工具。然而，最近一份关于移动设备上的录制与重放工具的调查报告指出这些工具尚未能满足工业界的使用需求。具体地来说，微信安卓开发团队提出了以下4个功能需求：

1）录制中记录的数据应包括坐标信息；

2）录制中记录的数据应包括组件信息；

3）录制和重放功能不受应用状态影响；

4）录制时应记录操作间的时间间隔。

同时，他们还对工具提出了4个限制条件：

1）不建议定制操作系统；

2）不建议使用插桩技术；

3）不建议要求root权限；

4）不建议需要应用的源代码。

除此之外，使用者通常希望录制与重放工具满足以下3点要求：1）代码开源；2）记录的数据有可读性；3）支持在不同的屏幕尺寸及不同的操作系统的设备上回放。

以上来自于工业界的功能需求和限制，为安卓平台上录制与重放工具的实现带来了挑战，本文概括为以下三个方面。

1. **移动设备丰富的输入来源**

不同于传统软件，移动设备支持更加多样和灵活的交互方式。移动设备丰富的输入来源，主要包括手势输入、按键输入和传感器输入等，为录制和重放都带来了一定挑战。例如，录制或重放传感器输入（GPS等）通常需要定制操作系统并提供应用的源代码。以上条件在工业背景中均难以满足。

1. **组件信息的高效记录**

为提高重放过程的鲁棒性，录制过程中需要同时记录事件的坐标信息以及组件信息。此处的事件通常来自手势输入。坐标信息通常易于获得。然而，获取组件信息则相对困难、且更耗时。获取组件信息的方法多样，但往往都会带来很大的时间开销，使得录制过程变得十分缓慢，影响开发者/测试者的使用体验。例如，在录制过程中，工具Culebra需要耗时10才能对一个点击操作产生响应。这导致其在实际场景中无法大规模使用。由于在录制阶段高效地记录坐标和组件信息具有一定难度，因此现有的工具大都仅记录坐标信息。

1. **跨设备的事件重放**

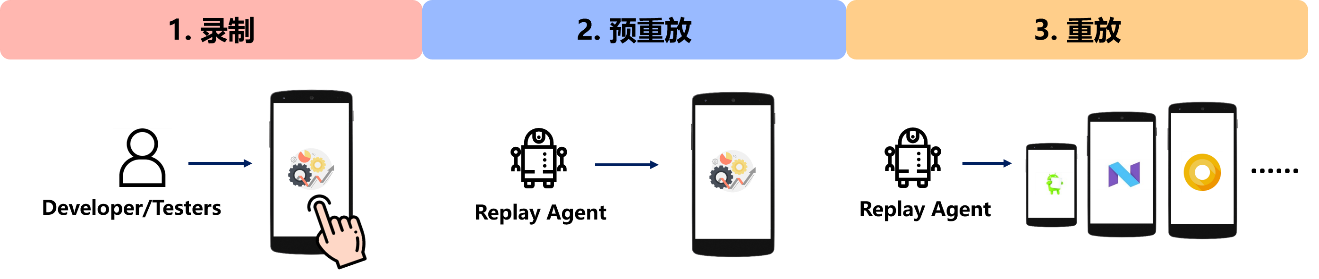
当app被唤起时，Android操作系统将根据设备的屏幕分辨率、尺寸等参数自动地在应用资源中查找合适的布局文件和资源进行渲染。Android官方文推荐开发者为不同分辨率和屏幕尺寸的设备定制对应的布局和资源文件。布局的多样性和灵活性带来了在不同设备上进行事件重放的需求，同时也给应用的测试增加了一定难度。不同设备上的事件重放仍是有待研究的开放性问题。

针对以上来自于工业界的需求和限制条件，我们设计并开发了该录制和重放工具，致力于在一定程度上解决以上三大挑战。**观看系统演示视频**<https://drive.google.com/file/d/1XD04LahI45gtrIhd6xlyp0NdAXN8Jpjh/view> 或 <https://pan.baidu.com/s/1aAiUOWqGoZZN1kvKYZUiOg> 。

1. **关键技术**
2. **动态插桩**
3. **预重放**
4. **适配性重放**

* 使用动态插桩技术，对移动设备中丰富的输入进行录制和重放，包括传感器输入等。这一技术的使用无需定制操作系统或提供应用源代码。
* 在录制和重放两大环节之间，引入“预重放”环节，以实现组件信息的高效记录。具体而言，在录制阶段，本系统仅记录手势输入事件的坐标信息，以保证应用较快的响应速度。在接下来的预重放环节中，系统在同一设备上基于坐标对事件进行自动的重放，同时记录所有相关的组件信息。由此即较为高效地实现了组件信息的记录，对工具的实用性不产生较大影响。
* 在不同设备上进行自适应的重放。自适应方法包括，启发式地对目标组件进行搜索以及基于设备参数的坐标变换，以实现不同屏幕尺寸、不同操作系统的设备上的准确重放。
* **触屏手势类型**

系统支持多种常见的触屏手势，包括：Tap, Long Tap, Double Tap, Swipe, Drag等。同时，系统支持常用的系统按键（如返回键、音量键等）和传感器输入的录制和重放。



**图1：系统实现**

* **事件记录方式**

系统在录制阶段记录的事件包括三类：Motion Event, Key Event以及Sensor Event。Motion Event记录了所有的触屏事件。获取安卓系统中的MotionEvent对象，并记录坐标、时间戳、操作代码等信息，解析后得到对应的触屏手势。Key Event记录所有的按键事件，如返回键、音量键等。Sensor Event记录传感器变化事件，如加速度传感器、GPS等。

* **事件重放方式**

系统重放事件主要分为两步。

预重放环节：根据录制阶段记录并初步解析后的事件，在同一设备、同一环境下进行基于坐标的事件重放，此过程由系统自动进行。本环节的主要目的是对事件记录进行信息增强，预重放的过程中，每次操作前通过坐标信息匹配对应的组件，并记录组件详细信息，同时通过基于坐标的操作对目标组件进行确认。除组件信息外，预重放环节还将记录界面截图、布局信息等。

重放环节：在不同的设备上进行事件重放，需要基于目标设备的配置进行坐标变换。具体来说，点击、长按等操作基于组件进行重放即可，而滑动、翻页等涉及轨迹的操作，需要进行坐标变换。本文采取的坐标变换原则是保持轨迹的物理距离相同，即用户感受到的距离。安卓系统中采用的距离度量单位之一density-independent pixel (dp)，能保持不同分辨率设备上的相同距离。dp与像素(pixel, px)间的变换公式如(1)式所示。

(1)

其中表示屏幕参数dots per inch。除坐标变换外，不同的设备还存在着屏幕尺寸大小不同的问题。例如，较大尺寸屏幕上显示的组件可能无法在较小尺寸的设备上显示，因而为事件重放带来了困难。根据实际经验，这种问题往往出现在可滑动的界面/组件上，因此本系统在重放时启发式地对目标组件进行搜索。若当前界面未找到目标组建，则加入滑动操作，直至组件出现。

1. **性能评估**
2. **评价指标**

为评价本系统实际使用中的性能，本文从Google Play的24个应用类型中选取了下载量最高的50个应用程序，下载量均超过百万次。每个应用程序中选择5个常用的应用场景作为录制和重放的目标。系统的性能评价分为三个方面：在相同设备上的录制回放、在不同设备上的录制回放、工具带来的时间和空间开销。实验选择的设备如表1所示。

**表1：实验设备**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **设备名称** | **屏幕分辨率** | **Dpi** | **屏幕尺寸/英寸** | **OS版本** |
| 红米1s | 720×1280 | 320 | 4.7 | 7.0.1 |
| Samsung Galaxy A8 | 1080×1920 | 480 | 5.7 | 6.0.1 |
| Samsung Galaxy A9 | 1080×2220 | 420 | 6.0 | 8.0.0 |

1. **性能评估**

* **相同设备上的录制与回放**

在录制的250个使用场景中，本系统成功回放场景214个，成功率85.6%。

* **不同设备的录制与回放**

实验中，本系统选择了35个场景进行跨设备的重放，共两轮。实验结果如表2所示。

**表2：不同设备录制与回放实验结果（共测试35个场景）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **录制设备** | 红米1s | | Samsung Galaxy A8 | |
| **重放设备** | Samsung Galaxy A8 | Samsung Galaxy A9 | 红米1s | Samsung Galaxy A9 |
| **成功场景个数** | 34 | 27 | 31 | 25 |
| **成功率** | **83.5%** | | | |

* **时间与空间开销**

时间开销指系统分别在录制和重放环节对应用程序使用产生的影响，主要表现为应用程序响应操作的时间。空间开销指系统录制时记录的日志文件的大小。实验选择了35个场景对系统的时间和空间开销进行评价。

**时间开销：录制环节4.19%，重放环节：6.99%**

**空间开销：记录日志平均大小79KB**

[1] Lam, Wing, et al. "Record and replay for Android: are we there yet in industrial cases?." *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*. ACM, 2017.