Monkey Test Automation on Android & iOS Native Apps

Lakshminarayanan Krishnamurthy

15/09/2022
Testing of Trustworthy Systems

The Context

Application Background:
- A server driven mobile application development platform
  - Based on Swift and Kotlin
  - 40+ Brands

Testing Objectives:
1) Need open-source based framework
   a) Generic, configuration driven test suite
   b) No hard dependency on programming language
   c) Should support both iOS and Android native apps
   d) CI/CD Integration
2) Uncover application crashes early in the development life cycle
1. Frequent app crashes on end user devices
3. Commercial/Open-Source Monkey test framework(s) are focused on either Android or iOS
4. Appium based Monkey test solution not available.
What is Monkey Testing?

Monkey testing is a type of testing where we send random inputs to an application while checking behavior or seeing whether the application or system will crash. Monkey test is a fast and effective method for testing the stability and robustness of software.

Any sequence of events which has a non-zero probability of happening, at least as long as it hasn’t occurred, will almost certainly eventually occur.
# Key Drivers

<table>
<thead>
<tr>
<th>Common Reasons for App Crash</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Poor Memory Management</td>
<td>❖ Bad User Reviews and Ratings</td>
</tr>
<tr>
<td>➢ Inadequate Testing</td>
<td>❖ Mobile App Uninstalls</td>
</tr>
<tr>
<td>➢ Unhandled Errors &amp; Exceptions</td>
<td>❖ Interrupted or Lost Transactions</td>
</tr>
<tr>
<td>➢ Excessive Code</td>
<td></td>
</tr>
<tr>
<td>➢ Device Incompatibility</td>
<td></td>
</tr>
<tr>
<td>➢ Network Issues</td>
<td></td>
</tr>
<tr>
<td>➢ Poor front-end optimization</td>
<td></td>
</tr>
<tr>
<td>➢ Bandwidth Constraints</td>
<td></td>
</tr>
</tbody>
</table>
Assessment Focus Areas

1. Android native app support
2. iOS native app support
3. Record and playback option
4. Option to write custom scripts
5. Commercial/ Open Source
6. Integration with CI/CD process
7. Framework development effort
8. Technical Skills required
9. Cloud Farm Availability
Tool Options

1. Firebase Robotest
2. UI/Application Exerciser (ADB)
3. UI Auto Monkey
4. SWIFT Monkey
5. Perfectomator
6. APPIUM ?
Custom Solution Blueprint

1. Test Case Management
   - JIRA + XRAY

2. Execution Platforms
   - bitrise
   - SAUCELABS
   - Local

3. Test Automation Framework
   - Property file
   - Configuration
   - testNG.xml

   - Android Widget
   - ImageButton
   - ImageView
   - View
   - ViewGroup
   - TextView
   - Event
   - Listener Services
   - Page Objects
   - Rotate, Horizontal Swipe, Vertical Swipe, Tap, Back, Home etc.,

4. Reporting
   - Extent HTML Report

Core Layer
- Utils
- Logging
- Reporting
- Helpers

Android Widget

View

ImageButton

TextView

ImageView

Listener Services

Events

Page Objects

Utils

Logging

Reporting

Helpers

Testing of Trustworthy Systems
Data Collected from Google Firebase

Total number of unique users: 17,71,708
Top 10 Devices where a greater number of events noticed

<table>
<thead>
<tr>
<th>iOS Devices</th>
<th>Android Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPhone XR</td>
<td>Samsung Galaxy S21 5G</td>
</tr>
<tr>
<td>iPhone 11</td>
<td>Samsung Galaxy S9</td>
</tr>
<tr>
<td>iPhone 8</td>
<td>Samsung Galaxy S21 Ultra 5G</td>
</tr>
<tr>
<td>iPad</td>
<td>Samsung Galaxy S10</td>
</tr>
<tr>
<td>iPhone 8 Plus</td>
<td>Samsung Galaxy S10e</td>
</tr>
<tr>
<td>iPhone X</td>
<td>Samsung Galaxy S10+</td>
</tr>
<tr>
<td>iPhone 7</td>
<td>Galaxy S8</td>
</tr>
<tr>
<td>iPhone 12</td>
<td>Samsung Galaxy Note9</td>
</tr>
<tr>
<td>iPhone 13 Pro Max</td>
<td>Samsung Galaxy S9 Plus</td>
</tr>
<tr>
<td>iPhone 12 Pro</td>
<td>Samsung Galaxy Note 20 Ultra 5G</td>
</tr>
</tbody>
</table>
1. Stress Testing and quick pseudo-random events (Appium vs ADB & Swift Monkey)
2. Handle Video, Images, Advertisement screen etc.,
3. Crashes uncovered during trial runs in CI/CD cycle: (Real Devices vs Emulator)
   - ANR – Application Not Responding
   - Native Crashes
4. Integration of Monkey Tests and Functional Tests
5. Common Framework and Programming language for iOS and Android
Any further questions?

Contact me:
Lakshminarayanan.Krishnamurthy@Epam.com