

QUALITY
FORGE 2024

A virtual conference dedicated to
software quality assurance

Register
now

 **TestDevLab**

Services ▾

R&D ▾

Industries ▾

Company ▾

Know

Quality
Assurance

Audio
&
Video

Development

Company

August 23, 2024

A Practical Guide to Test Automation Tools for Mobile Games

Nikita Jamrovskis
QA Engineer



Mobile games are one of the most popular forms of entertainment today, with billions of users around the world. In fact, as of 2024 there are around 1.7 billion mobile gamers worldwide and this number is expected to grow in the coming years. According to Statista, the Mobile Games market is projected to generate a revenue of US\$98.74bn worldwide this year. With such a huge user base and revenue, it's essential for game developers to ensure that their games are bug-free and run smoothly on all devices. That's where test automation for mobile games comes in.

Automated testing can help ensure that a game runs as intended, without the need for human intervention. Automated tests are basically like your manual tests, but done for you by a computer. Imagine having to do hundreds of tests manually,

like tap one button here, swipe there, drag elsewhere, and so on. Quite a burden, isn't it? And you have to do it daily. That's where automation shines. You just write a test suite once, and you can launch it anytime, how many times you want, and focus on doing something else while tests are executed. And to make this process even easier and faster, you can use a test automation tool for mobile game testing.

To help you better understand mobile game test automation and which test automation tools are worth using, we've prepared a practical guide with all the information you need to get started. Learn more about test automation for mobile games, find out more about AltTester, a test automation tool for mobile games, and discover alternatives that you can use to help you choose the best one for mobile game testing.

What is test automation for mobile games?

Test automation for mobile games involves using automated tools and scripts to conduct various testing tasks

on mobile games, ensuring they function correctly, are free of bugs, and provide a good user experience. Automated testing offers benefits like increased speed, efficiency, consistency, and early bug detection. Namely, one of the main struggles game developers face is ensuring that their game runs smoothly across a range of devices. With so many different smartphones and tablets on the market, it can be challenging to ensure that a game runs correctly on each one. This is where automation tools come in. Automated testing tools can simulate different devices and test the game on each one to ensure it runs as intended. This saves developers a significant amount of time and resources, as they don't need to manually test the game on each device.

You might be interested in: [Possible Ways to Automate In-game Movements Using AI](#)

However, there are also some disadvantages to using test automation tools for mobile games testing. One of the main drawbacks is that automated testing tools can only test for known issues. They can't identify new or unknown issues that may arise when a game is played by real users. For example, an automated testing tool may not be able to detect that a user can

cheat in a game, which could ruin the experience for other players, if it's a multiplayer game. Or a player can fall through textures or objects, where it's not intended by the game design. Such cases are not covered by automated testing.

Another limitation of automated mobile game testing is that it can be difficult to set up and maintain. Many automated testing tools require a significant amount of technical expertise to use effectively. Developers may need to invest in training or hire new staff to manage the testing process. Also, the cost of test automation tools might not be worth it at all.

In this blog post we take a closer look at some test automation tools for mobile games and their pros and cons—based on our experience in automated mobile game testing. Let's start with AltTester.

AltTester—A game test automation tool



Based on our experience, a great game test automation tool is AltTester. AltTester is a Unity mobile games testing automation tool that enables game developers to test their games on a range of devices without the need for manual testing. AltTester works by simulating different devices and testing the game on each one to ensure it runs correctly. It also allows the use of real devices, both iOS and Android. One of the advantages of using AltTester is that it is simple to set up and use, even for developers with little technical expertise. Another important advantage is—it's free. The basic Unity plugin comes as a free package, which is fairly easy to install (more on that later). There is also a good addition to make automation even easier—AltTester Desktop, but it's free only for a limited time. Although AltTester Desktop helps to speed up test writing, by making the process of finding elements, their names, components and properties easier, it's not strictly necessary.

But as any software, AltTester has both pros and cons.

Pros:

- Free.
- Excellent choice for games built with Unity.
- Allows testing any platform supported by Unity itself.
- Tests can be run on both real and emulated devices.
- Shows test results with passes and fails (stating why test failed).
- Easy-to-understand UI.
- Has option to work with Appium.
- Documentation (with flaws, like poor examples, but still explains a lot).
- Fast test execution speed (compared to other tools, like Airtest, Repeato, etc.).
- Multiple languages supported (C#, Java, Python).
- Allows for deep interaction with game objects (if AltTester SDK is built in the app).
- Discord server, where you can ask for

help/explanations.

Cons:

- Installation instructions in documentation don't include possible issues (which we encountered most of the time).
- The more complex a test is, the more chaotic and random the results are.
- Only works if AltTester SDK is implemented in the app.
- Related to the previous point - Requires a close collaboration with the development team and developers have to first implement the SDK into their project, if AltTester is going to be used.
- Sometimes test failure is not explained properly, leading to a lot of time being wasted on pretty simple issues. iOS tests are true pain. Expect lots of new errors, port forwarding issues, and the need to rewrite parts of tests in most cases, as the input system might need an alternative approach.
- Free technical support is sporadic and often non-existent. Discord server has various channels, but many questions remain unanswered

or unresolved after a long time. If you seek proper tech support for AltTester, you will have to pay 300 dollars for a 1-on-1 session with one of their team members.

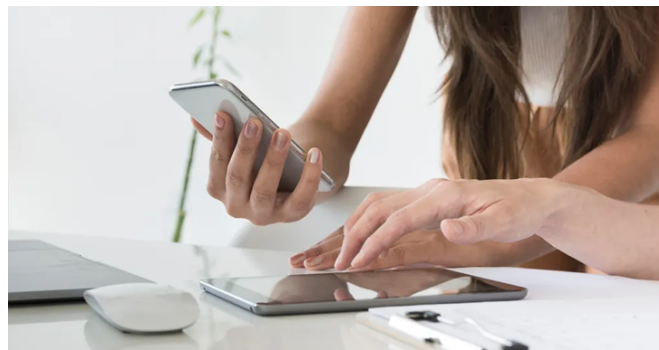
- Critical errors often have little to no information on how to fix/avoid them. The documentation only describes a few general issues. The Discord server, on the other hand, has a 90% chance that it has something written about specific errors, however, this doesn't necessarily mean that issues were resolved.
- Different devices will behave differently if you try to create an "ultimate test template" with the intention to copy-paste it. Let's say, you have a level with obstacles you want to pass. The player has to jump within a certain distance range, with a specific speed. If you try to make a test with an "AI" approach, where an obstacle gets defined and the player seeks it, goes towards and jumps over it, regardless of what kind of obstacle it is, it will be tied closely to devices' FPS (frames per second). In that case, a hardcoded approach (e.g. when the player is at position X - do jump, depending on obstacle type) will be much more consistent, but will require manually finding the jump

position, speed and distance.

- The longer you use AltTester in one session, the more likely it is that some random errors will start to appear. Restarting the Unity engine usually helps. It's unknown where those issues come from.

Regardless of its flaws, AltTester is currently the most reliable tool for Unity games. We would know. It was the best choice for automating our game, "The Lost Treasures of Buggalo", a 2D platformer game developed by TestDevLab. You can download the game on the [Google Play Store](#) or [App Store](#).

Alternative test automation tools for mobile games



Throughout our journey of finding the right test automation tool for mobile games that meets our needs, we also came across various tools and services:

- **Repeato.** An automation tool based on JavaScript that is free, but with limitations, and allows you to create tests via manual screen touch. Perfect for non-programmers if the app under test doesn't require multi-touch or precise timing. It allows custom JS scripts, which can test multi-touch as well, but performance won't allow accurate timings.
- **AirtestIDE.** A free test automation tool that is very similar to Repeato in terms of recording tests and creating test scripts from recordings. Additionally, it allows Python usage for custom scripts. The downside is that documentation is only available in Chinese. Performance is also not friendly towards timing-sensitive tests.
- **Kobiton.** A web-based tool with a public list of cloud devices available that offer users a free trial. However, it comes with a major flaw. Namely, the majority of available devices are constantly locked by other users due to incorrect password input (and there's no information on the trial page of what those passwords

exactly are). Through our attempts, we weren't really able to check this one out within the trial period.

- **Katalon.** Free with monthly limitations, Katalon uses image recognition techniques for tests and is not timing-friendly. Therefore, it's not really suitable for our task, but it would be a great tool for some e-commerce apps, as it doesn't require programming knowledge to create tests. Also, it has some good tips and a helpful community.
- **TestRigor.** An interesting option that is free with limitations. It is a web-based tool that allows you to write tests using pseudocode. The documentation explains everything in an accessible manner. The service uses emulators instead of real devices, limiting testing coverage. When we checked it with our app, it couldn't even render it at all, freezing at the very intro. But overall, it is easy to use and a good option for simple apps, just not for most games.
- **Perfecto.** A flexible tool that can be used for mobile/web, desktop app testing. It offers a 14-day trial and this trial version gives you access to some cloud-based devices, both iOS and Android. As a feature, when you upload your app, it can automatically

generate tests, according to what your app has. But similar to the other tools, it is a good option for simple apps, but a bad one for games. It allows play-and-record test creation, as some previous tools we've seen, and supports JavaScript, Java, C#, Python and Ruby. Probably due to the trial version, but we couldn't get past the issue where connection to their servers was always timing out, due to a "no available sessions" error, whenever we launched tests.

- **Appium.** One of the most used tools for mobile automation, Appium is great for both emulated and real devices. But, as some of previously mentioned tools, it uses the app's hierarchy of elements to interact with. Simple apps can be tested easily, while more complex games... not so much. It allows custom scripts in Ruby, Python, Java, JavaScript, PHP, C# and RobotFramework. Recently, Appium started integration with AltTester, to mix features of both tools, but it's still in progress.
- **TestGrid.** An Appium based tool that offers a 3-day trial, TestGrid uses TestCaseGenerator, which scans the screen for elements and creates test cases. Unfortunately, for our task it was useless, as every frame of the game was counting as a separate

“page”, and yet no elements could be found by the generator. To make things worse, the game was running at mere 3-4 fps.

The bottom line

To conclude, test automation for mobile games is something that should be implemented in the very early stages of development, as most of the test automation tools are not meant for automating complex apps. Of course, depending on the task at hand, some tools that didn't work well in the past, might become a better option in the future. Yet with decent knowledge and programming skills, even currently available tools allow us to create test suites for almost any need. In our case, AltTester was the only reasonable solution, with good performance, relatively easy and quick setup, and good capabilities for what we needed to test.

Do you have a mobile game that could use a bit of testing to ensure it performs as expected and delivers a smooth gaming experience? We can help you with both manual and automated mobile game testing. [Get in touch](#) to learn more.

Tags: Mobile App Testing Game Testing
Software Testing Software Quality Assurance
Test Automation Automated Testing
Test Automation Tools

Subscribe to our newsletter

Sign up for our newsletter to get regular updates and insights into our solutions and technologies:



Subscribe

You may also like



5 Critical
Reasons to...

[Learn more >](#)



Integration
Testing 10...

[Learn more >](#)



Manual
Testing: A...

[Learn more >](#)



Full-Stack
Testing: A...

[Learn more >](#)

info@testdevlab.com
+371 25423562

Subscribe to our newsletter

Sign up for our newsletter to receive regular updates and insights into our solutions and technologies:

Your business e-mail *

Subscribe

bm

certification

ISO 9001

ISO 22301

ISO 27001

System certification

Services

QA

Management

Performance

Testing

Test Automation

Manual Testing

Accessibility Testing

UX & Usability Review

Security Testing

CX Testing

CI/CD Services

Compatibility Testing

Competitive Analysis & Benchmarking

ISO Advisory Services

Functional Testing

Non-Functional

4.8

★★★★★

5

★★★★★

- Platforms
- Mobile App Testing

Web App Testing

Desktop App Testing

API Testing

SDK Testing

AR / VR Testing

- R&D
- Audio & Video Quality Testing

Battery & Data Usage Testing

- Location Testing

Network Simulation

Loadero

TestRay

- Industries
- Fintech & Banking

- Company
- Why Choose Us

How We Work

Clients

Case Studies

About Us

- Knowledge base
- Blog

Podcasts

Webinars

Whitepapers

Press Center

We're hiring!

Visit Careers page

Testing	Communications
Testing levels	
Integration	Hardware & IoT
Testing	Security & Authentication
System Testing	
User	Travel & Hospitality
Acceptance	Healthcare
Testing	Gaming
	Artificial Intelligence