ixie WHERE GAMES LEVEL UP

GAME QA

Game Test Automation Tools – A Comprehensive Review and Comparison

By iXie September 1, 2023

1. Introduction:

Game automation testing is a crucial aspect of game development that involves using specialized tools and techniques to automate the testing process. It helps game developers ensure the quality, stability, and functionality of their games by automating repetitive tasks, identifying bugs, and validating game behavior. Traditionally, game testing has been a manual process, where testers play the game, perform actions, and observe the results to find issues. While manual testing is still valuable, it can be time-consuming, labor-intensive, and prone to human error. Game automation testing aims to address these challenges by leveraging automation tools and scripts.

Automated game testing involves creating scripts or using testing frameworks to simulate player interactions, such as button presses, mouse movements, and UI interactions. These scripts can be designed to perform specific actions, test scenarios, or execute a series of predefined test cases. The automation tools execute these scripts and provide feedback on the game's behavior, detecting any anomalies, bugs, or issues.

Game automation testing offers several advantages:

- It significantly reduces the time and effort required for testing, as scripts can perform tasks much faster than manual testers.
- It enables testers to run tests repeatedly, ensuring consistent results and allowing for regression testing to catch any issues introduced during development.
- Automation testing can cover a wide range of scenarios, including stress testing, performance testing, and compatibility testing across different platforms and devices.
- Automation tools can perform repetitive actions with precision, helping expose hidden bugs, boundary cases, or rare scenarios that might go unnoticed otherwise.
- Automation testing allows for easy retesting and verification of bug fixes, increasing the overall efficiency of the development process.

With the wide spectrum of tools available in the market, it is always important to assess your specific requirements, consider the capabilities and limitations of each approach, and evaluate how well they align with your game automation needs before making a decision. This blog compares the game automation testing tools on the aspect of Functionality, Ease of use, Compatibility and Community Support. Further downs we have classified the game automation testing tools based on the underlying working principle.

2. Game Automation Technique Classifications:

In general Game Test automation is not as same as Mobile or Web UI Automation as in game automation we are dealing with animations and rendered Game Engines which cannot be directly interacted by the test scripts. To overcome this issue, the Game Test automation tools have incorporated certain techniques and underlying principles of kind and based on which they can even be classified. Those are as follows:

- Image Base Automation Technique tools

- Game Elements Exposing APIs Technique Tools

- Game Bots Technique tools

2.1. Image Base Automation Techniques Tools:

Principle of Operation:

Image-based automation tools use image recognition techniques to interact with game elements. They capture screenshots, analyse the images, and identify specific elements based on their visual appearance.

Advantages:

- **User Interface Independence:** Image-based tools can interact with games regardless of the underlying technology or programming language used in the game's user interface.
- **Visual Flexibility:** These tools can recognize and interact with game elements that may not have easily accessible attributes or identifiers.

Limitations:

- **Fragility:** Image-based automation relies heavily on visual appearance, making it sensitive to changes in graphics, resolution, or UI layout. Even minor changes can break automation scripts.
- **Performance Impact:** Image recognition can be computationally expensive, potentially affecting the performance of the automation process.

Example:

Sikuli, Zap Test, Airtest

2.2. Game Elements Exposing APIs Technique Tools:

Principle of Operation:

Elements Exposing API-based automation tools establish a certain communication channel between the automation tool and the game using the certain API protocols. They exchange messages and commands to interact with the game in real-time.

Advantages:

- **Real-time Interaction:** These tools allow for direct, real-time communication with the game, enabling immediate response and interaction.
- **Precise Control:** These tools can send specific commands and receive game state updates, providing fine-grained control over game interactions.

Limitations:

- **Integration Requirements:** This automation requires game developers to implement Communication API support within the game. It may not be readily available or feasible for all games.
- **Dependency on Game APIs:** These tools rely on the game exposing APIs, limiting their usability to games that support this communication method.

Examples:

Game Driver, Poco Driver, Alt Unity

2.3. Game Bots Technique tools:

Principle of Operation:

Game Bot automation tools leverage various techniques, such as AI algorithms or rule-based systems, to simulate player behaviour and automate game actions. They aim to replicate human-like interaction within the game environment.

Advantages:

- **Flexibility and Adaptability:** Game Bot automation tools can be designed to adapt to different game scenarios and dynamically respond to changes in the game environment.
- **AI Capabilities:** Advanced Game Bot tools can incorporate AI algorithms to learn and improve gameplay strategies.

Limitations:

- **Learning Curve:** Developing effective Game Bots often requires expertise in AI or rule-based systems and a deep understanding of the game mechanics.
- **Game-specific Implementation:** Game Bots may need to be tailored or customized for each game, making them less versatile for use across different games.

Examples:

AutoHotkey, Pulover's Macro Creator, X-Mouse Button Control

Ready to score big in the gaming industry?
Our Game QA services can help you achieve a flawless launch!
Talk to our experts!

3. Game Automation Tools Review and Comparisons:

There are various Game automation tools available in the market and for this comparison we are going to take AltUnity, Poco Driver, Airtest, Unium and AutoHotKey as these are predominantly used by Automation Testing Community and Preferred by the companies for automating their Games and introduce the tests created by these games into the Delivery pipeline for faster and bug free release.

3.1. Alt Unity:

Features:

- Alt Unity provides a wide range of features for Unity automation, including object identification, UI interaction, and automated testing.
  - This tool is limited to only Unity Games and does not support other Game engines.
  - This tool comes in both premium and community where the premium version comes with inbuilt inspector which helps ease of

  - development And Even support Record and Playback option.
  - Appium plugins are available for this tool which helps to handle the native popups with ease.
  - Can Run automated tests on PC, Mobile and Consoles.
  - Availability of source code helps users to customise the SDK as per needs.

Ease of Use:

- Alt Unity offers a user-friendly interface and supports scripting languages like C#, Python, Java and JavaScript making it relatively easy to integrate and use.

Community Support:

- Although Alt Unity has a smaller user community, it has an active support team that aids and updates to users.
- Clear Documentation helps the first-time users to master the tools easily.

3.2. Airtest:

Features:

- Airtest is an open-source automation framework that supports multiple platforms, including Unity. It offers features for UI testing, performance testing, and gameplay automation.
- Support all platforms like Consoles, PCs, and mobile devices.

- Supports all Gaming Engines
- Incorporates OpenCV for image detection to interact with games under test.

Ease of Use:

- Airtest provides a simple scripting language and a user-friendly interface for recording and playing back test scenarios.
- Supports Only Python for scripting automated tests.

Community Support:

- Airtest has a limited community of users and developers.
- Online forums and resources are available for support and knowledge sharing.

3.3. PocoDriver:

Features:

- PocoDriver focuses on Unity UI automation, using the Poco UI framework to interact with UI elements.
- Currently compatible with Python 2.7 and Python 3.3-3.6.
- Allows easy and straightforward SDK integration to the game.
- Supports popular game engines such as Unity3D, Unreal, cocos2dx-js, cocos2dx-lua, Android/iOS native apps.

Ease of Use:

- PocoDriver offers a straightforward API for automating UI tests.
- Supports recording and playback which helps faster test development.
- Supports Only Python for scripting automated tests.

Community Support:

- PocoDriver has a smaller user base.
- Not very active Online forums and resources can provide support and guidance.

3.4 Unium:

Features:

- Unium is a Unity-specific automation tool that offers features for unit testing, integration testing, and automated testing of Unity projects.
- No new Builds are required for creating new auto test.
- Super-fast and Impact free to game

Ease of Use:

- Uses Http requests for interaction with the Game.
- Less user friendly when compared to other gaming tools.
- Only Developers can work with this as it needs Game source code for creating auto test.

Community Support:

- No Community Support
- Documentation and support resources to assist users are available.

3.5 AutoHotKey:

Features:

- AutoHotkey is a general-purpose automation scripting language that can be used for game automation.
- It offers features for automating keyboard and mouse inputs, making it suitable for macro-based testing.
- Supports only windows-based games.

Ease of Use:

- Not suitable for dynamic games and games involving complex actions.
- Mainly used for automation mundane tasks and macros in games.
- Only Developers can work with this as it needs Game source code for creating auto test.

Community Support:

- Limited community usage and support for the users.
- Documentation and support resources to assist users are available.

4. Conclusion:

The best tool for game automation testing depends on various factors, including the specific requirements of your testing project and the nature of the games you are testing. When selecting the best tool for your automated game testing it is essential to evaluate the documentation, community support, and compatibility with your testing environment when making your decision. Other factors such as the specific requirements of your testing project, the type of games you are testing, the desired level of integration, and the available scripting capabilities.

#Game Automation #game QA #Game Testing

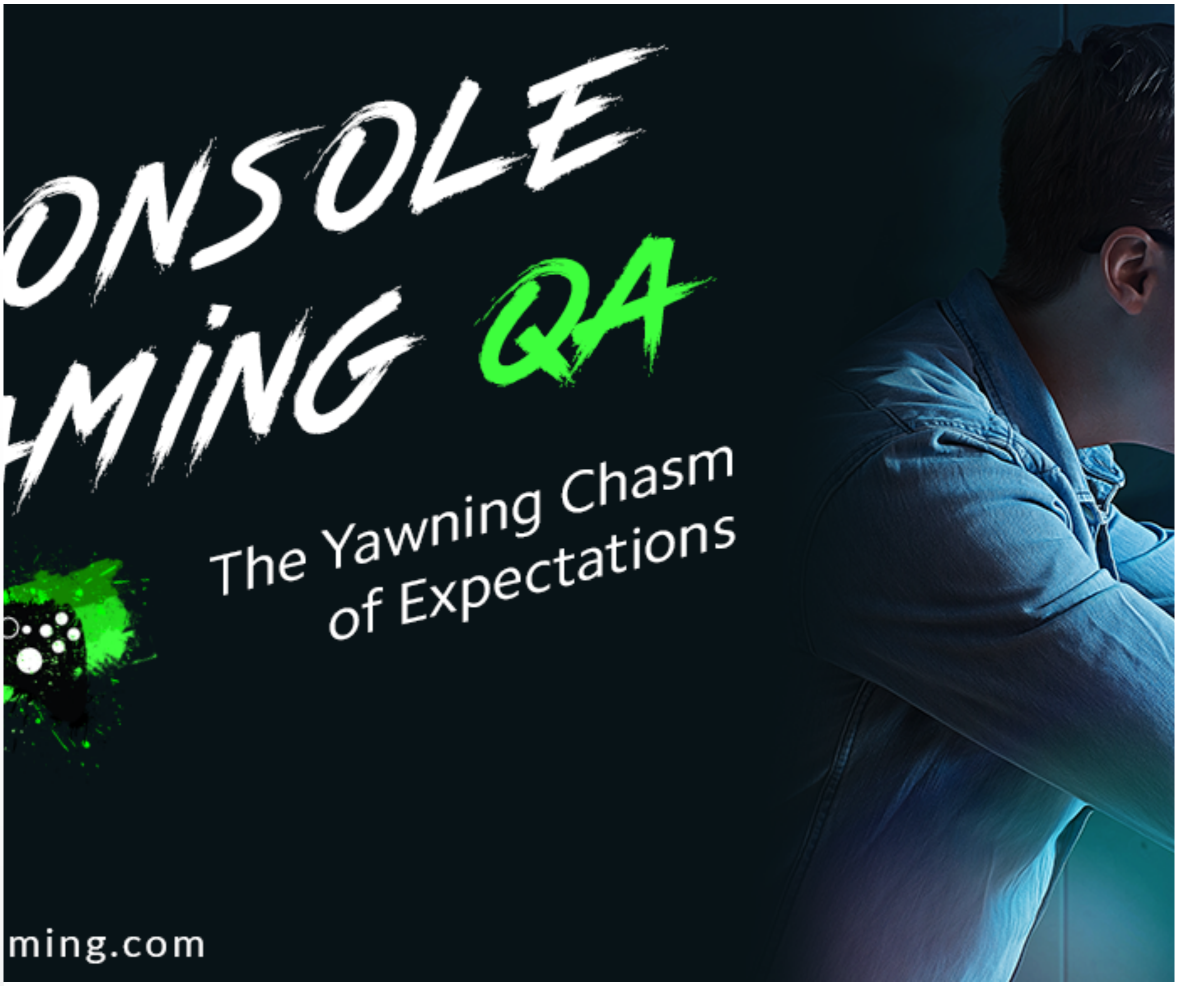
iXie

← PREVIOUS

Is Game Automation the Next Generation of Testing?

NEXT →

2D Game Automation using Airtest



Console Gaming QA – The Yawning Chasm of Expectations

By Jaaziel January 5, 2018

●○○○○○

Search

Search

Recent Posts

- Ensuring Game Quality with Black Box and Grey Box Testing in Automation
- The Ethical Implications of Procedural Content Generation in Gaming
- The Evolution of Game Testing: How Modern QA Teams Adapt to Complex Gaming Landscapes
- Enhancing Mobile Game Performance: Essential VFX Techniques for Visual Excellence
- Stress Testing: Pushing Games to Their Limits Before They Reach Yours

WHERE GAMES LEVEL UP

iXie is the gaming division of Indium. We are an end-to-end gaming services company that provides holistic services across the game development lifecycle, using world-class

IXIE GAMING

About Us

iXie Academy

Careers

Contact Us

Prevention of Modern

GAMING SERVICES

Game Development

Game QA

Gaming Analytics

Game

SPECIALIZED QA

Functionality Testing

Device-based Testing

Localization QA

tools and technology. We have extensive experience working on 450+ titles across various genres and platforms.

Slavery

Localization

Art

Production

Software

Testing

Services

Stay In Touch