

AUTOMATION TESTING

# How to Automate Video Game Tests

30 June 2020, 09:30 AM

By Test Guild

Share:

Join the Guild for FREE



💡 Learn more about our  
Strategic Partner



Chances are, you enjoy playing video games.

And if you're reading this, I'm guessing you're also a tester.

Wouldn't it be cool to combine your love of gaming with automation testing?

Be sure to read this entire article to discover how you can get into video game automation using [GameDriver.io](#).

## What is Game Driver?

GameDriver was created by Shane Evans.



After doing some research on game development, he realized that beyond unit testing, most everything else was still done manually.

Having worked for a massive test tool provider, and being a gamer himself, Shane thought there had to be a better way. Hence GameDriver.io was born.

Utilizing the GameDriver agent enables video game developers, testers, and producers to automate playtesting. This, in turn, improves their time to market, test coverage, and overall quality revenue.

Can't wait to see it, right?

Before we get into how it all works, you might be wondering why gaming companies would even care about the automation of video games.

## Why is automation in game development so important now?

Simply put, software games have a much longer lifespan these days than they did in the past.

For example, **World of Warcraft** has been in production since 2004, and since that time, they've had numerous patches, releases, and expansions.

Gamers are heavily invested in these games and expect things to work—and these changes are not expected to break anything.

When Shane spoke to the head of QA at **Blizzard** (the maker of World of Warcraft), he was shocked to learn that they don't do any test automation to speak of.

Their developers do unit testing, but their integration testing and playtesting are 100% manual, and more and more games are becoming long-lived titles like WoW.

Also, many game companies are now making the shift to more CI/CD practices that require them to release more quickly, while still being able to test faster.

Another problem is that there are armies of manual testers out there; thousands of them working in shifts at some of the larger studios. And they don't have a whole lot available to them in terms of career growth opportunities.

So it's just manual playtesting, and either you do that until you become a manager or you get burnt out and switch roles because testing can be quite arduous.

Automating just some of the simple things that game developers have to do for every release and every build is a huge win.

Of course, the intent of this tool is not to completely eliminate the need for manual testing, but to reduce that dependency.

So if you are an experienced tester, this is an excellent opportunity for you to find a job in gaming, since the industry is not very mature in test Automation practices.

And also, you're probably already familiar with using test automation tools like Selenium to find bugs.

I think we can all agree that it makes sense to automate every day, repetitive tasks that game

developers need to perform all the time, like:

- Creating a character
- Starting the game
- Performing some primary tasks for sanity tests before jumping into gameplay

But what does this look like?

## The Architecture of the Driver

Now, how do you start automating some of these game testing tasks using GameDriver.io? What skills do you need?

GameDriver is a framework that provides developers an API agent that they can embed into their games so that when the developers are creating the build, they embed the driver game object within their game.

00:57

That allows GameDriver to connect into the game, control it, and interrogate the game while running both in the development stage and when you've put together something like a standalone build.

This approach allows you to send in commands like movements or query objects like:

- What is the score?
- What's happening on the screen right now?
- What's the position of this player object, etc.?

It essentially allows you to communicate with the game by creating a backchannel to the driver's API and send commands that perform those actions asynchronously or synchronously.

These commands end up providing the same type of input that a real user would perform—except it's done through a different mechanism.

Besides the user input functionality you get with GameDriver, there is much more power to be had.

For example, you're not limited to what the user interface can do in any game.

That means that if you're testing with, let's say, Appium, or some other kind of user interface—

level testing tool, that UI-based testing tool is going to be limited to testing only what's visible to the user at runtime.

That's fine for some approaches, but with the driver you can really expose all the internals of the game so that you can interrogate objects, get their information and then perform much more comprehensive testing, aside from just UI-based screen manipulation.

## How would a developer use the API for automation?

If you already have experience with API automated testing using tools like **Rest-Assured**, it's the same with GameDriver.

Primarily, your developers would code API calls into your game.

To get started, there is a small set of libraries under ten megabytes that needs to be added to the game folders under the Assets directory. Then you add one script and you're done.

That script needs to be launched as part of the initial launch of the game, but any scene that launches within the game and contains that script gives the driver access to it.

Here is an example of some Game Driver API code in action.

12:49

## Does it have a game test recorder?

Like automation tools such as **Selenium IDE**, **GameDriver** does have a recorder in Alpha right now that's in the early stages of being productized.

But the idea would be for you to record the basic actions.

And like most automation recorders, it's not meant to create a comprehensive test suite. But it's helpful to give you a starting point in terms of the actions that were performed by the user. You would then go about inserting your control statements, data driving it, and adding non-user types of commands like querying, or adding assertions to the test.

## What gaming engines are supported?

*Test Guild New Podcast: Testing Castlevania, Playwright to Selenium Migrat***TEST GUILD**  
By Joe Colantonio[Learn Automation Testing](#) ▾ [Free Test Tools](#) ▾[Testing Confer](#)

once the unity market has all the features they're looking for to do some level of automated testing, they'll begin looking to support other engines.

The next engine to be supported will probably be **Unreal Engine**, which has been around for a long time and is widely used in the larger studios.

Also, AWS recently jumped in with **Amazon Lumberyard**, and that might be a supported option at some point as well.

The driver is still relatively new, so I'm sure more functionality and supported platforms will be added over time. Other supported platforms are:

- Supports PC & macOS—build and run tests from macOS or Windows
- Multi-host—build local, run remote tests on private lab or public Cloud providers like **AWS**, **Azure** and **Google Cloud (GCP)**
- iOS—high performance mobile game testing

## Can You Automate Virtual Game Play Test?

Shane did mention that one of the early adopters of GameDriver had a complicated VR project they needed to automate. They were surprised to find that a lot of their manual testing, which required somebody to put on a helmet and do those types of validations, could be automated through GameDriver with very few gaps.

I see this as a great start.

If you're serious about testing and gaming, you should check this out and be one of the first to make the move to a job in gaming automation as a game tester.

## Do you know that the TestGuild has Free Online Meetups?

Check out our Test Guild Meetup with Shane Evans to learn even more about GameDriver.io. Also, make sure to check out and register for our **upcoming meetups**.



**Mike Harrell**

July 31, 2020 at 12:31 pm

As an SDET who's been working in web development for years but would much rather be working in games, this is an AWESOME project! Can't wait to find out more!

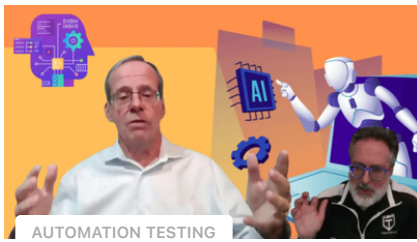


**Reprise Assistant**

January 21, 2021 at 4:59 pm

Try Reprise Assistant at [reprass.com](https://reprass.com), it can automate any task in couple of minutes

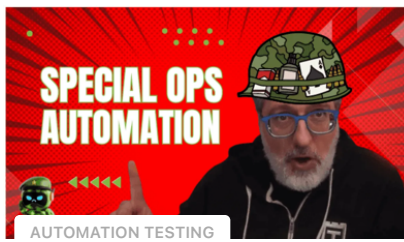
Comments are closed.



## Symbolic AI vs. Gen AI: The Dynamic Duo in Test Automation

Posted on 09/23/2024

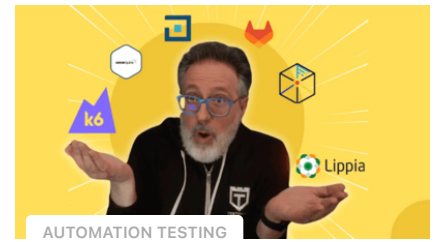
You've probably been having conversations lately about whether to use AI for testing. ...



## 8 Special Ops Principles for Automation Testing

Posted on 08/01/2024

I recently had a conversation, with Alex "ZAP" Chernyak about his journey to ...



## Top 8 Open Source DevOps Tools for Quality 2024

Posted on 07/30/2024

Having a robust Continuous Integration and Continuous Deployment (CI/CD) pipeline is crucial. Open ...

## Create Less Flaky Automation Tests. Quality content right in your inbox

Automation Testing can be frustrating. Don't do it alone. Register now to get automation, performance, and security testing tips from some of the top experts in the industry.

[Click to Get 10 Quick Test Automation Tips Now](#)

“

I appreciate the information you send over email. It has really helped me to enhance my testing skills!

*Alisha Sheikh (Test Engineering Senior Analyst at Accenture)*

Thank you for all the information provided in your newsletter. I am new in automation testing; hence, it is very helpful.

*Audrey Pruvot (TagPlay)*

(865) 272-6088



Copyright 2024 by Joe Colantonio | Test Guild  
LLC

[Work With Us!](#) | [Privacy Policy](#)