PA2552 - Software Testing

GUI Testing

Christoffer Bohman
MScEng: Game Engineering
Blekinge Institute of Technology

Contact:
chbh22@student.bth.se / student@skyh1gh.dev

Revisions and git history:
GitHub

February 16, 2026

Notice of AI usage:

No AI was used for this assignment

1 Test Cases & Acceptance Criteria

- As a user I want to be able to access the Godot webpage.

  Acceptance: The website is accessible through a standard web browser.

- As a User I want to be able to download the latest version of Godot to my machine

  Acceptance: The software has been successfully downloaded to the local machine.

- As a contributor I want to be able to know how to contribute to the project.

  Acceptance: The relevant learning resources are available and accessible to the contributor.

- As a corporate sponsor I want to have a way for users to access my tools from the Godot website.

  Acceptance: The information of the sponsors as well as their related links can be found and accessed through the website.

- As a user I want to be able to access the Godot Documentation

  Acceptance: The website is accessible through a standard web browser.

- As the Godot foundation I want to expose a way for users to show support through donations

  Acceptance: A donation webpage is available and a donation can be sent through it.

- As a user I want to be able to search the documentation according to my needs

  Acceptance: A search box should be usable and the results should be relevant to the search keyword.

- As a contributor I want to be able to report bugs

  Acceptance: The Godot webpage has a feature that allows users to report defects in the Godot project.

- As a user I want to be able to rollback or download older versions of the engine.

  Acceptance: A webpage with the older versions should be accessible and the links should be working.

- As a user I want to be able to use a translated version of the webpage

  Acceptance: An option to change the language of the webpage should be accessible. The most common languages should also be available.

Table 1: Abstract and Concrete Lean Testing Principles

Test Case	Development Time in m (approx)	Execution Time in s (approx)
Access Godot Webpage	10	5
Download Latest Version	5	7
Access Contribution Page	10	10
Verify Sponsors Pages Work	25	26
Access Engine Documentation	10	9
Access Donation Page	20	17
Search For Specific Item In Documentation	15	17
Find Resources For Bug Reporting	25	20
Finding Older Versions of Godot	20	20
Changing To Translated Version	10	10

Table 1: Values from the GUI testing suite

2 Discussion

The experience I had using EyeAutomate and GUI testing was relatively pleasing with a fair few problems in the beginning. I'd like to begin by addressing the issues I have had using EyeAutomate together with Linux. From my brief research and troubleshooting, it seems as though the rendering API Wayland does not interact well with either EyeAutomate nor Sikulix as the ability for automated cursor movement seems to be disabled for security reasons. This would not have been a problem if modern Linux distributions still used the legacy X11 API. This all eventually required me to repartition one of my hard drives and install Windows 11 causing a smaller headache.

On the other hand, GUI testing has proved to be a useful tool when automating specifically visual and UX-specific use cases. For me as a programmer that focuses on 3D-graphics, I can see certain areas where this kind of testing may be applicable. I could for example find this tool to be exceptionally powerful for website and usability testing in dynamic settings. However, due to the nature of visual testing, some types of minor changes like moving a button to a different page or changing the colour scheme or theme of the test subject may completely break the tests.

My test suite is based on testing the functionality and usage of the Godot foundations webpage for the Godot Game Engine. The reasoning behind this decision was partially due to the fact that I am studying a game engine focused programme, and partially because it is easy to reason about the different requirements of an open source development project.

In the list below, I will describe and explain why I chose these specific features for my suite, why they are relevant to test which resultingly also partly implies why they are of high quality.

- Access Godot Webpage

  - Accessing the main component of any application is a requirement before being able to test anything else. That same principle applies here and this theme can be tracked throughout my testing as this specific test case serves as the isolating member that restores the original state of the Godot webpage after every test. This case is fairly simple all in all making it easy to read and by implication also rewrite should the agile environment require it. This scenario only encompasses the opening of the firefox web browser, search of the godot website and handling the closure of the web browser.

- Download Latest Version

  - As with any website that hosts any kind of file, a way to download or retrieve that file is also generally a requirement. This test case handles the scenario where the user wants to download the latest version of the Godot Engine. A limitation with my test implementation is that the file does not get downloaded as it should be per the defined requirements above, this is to avoid any rate limitation and unecessary retrieval of the same large file over several test suite runs.

- Access Contribution Page

  - Free Open Source Software is heavily reliant on active maintainers and contributors in order to stay alive and up to date. Resources for anyone that wants to take part in this should be available in order to create an environment where new contributors may join in a systematic and productive way. This case covers the availability of a contribution page as well as makes sure the documentation may be accessed through the main webpage.

- Verify Sponsors Pages Work

  - Third-party financial aid, especially for non-profit foundations, can boost the possibilities of reaching certain goals that are being aimed for. Consequently, a means to garner sponsorships and financial support is through advertisement of each sponsors personal page. In this test, the ability to see and access the different webpages is being tested. For a more accurate result, a variety sponsor links are being tested to ensure that no link is invalid or doesn't work.

- Access Engine Documentation

  - For this software to be usable, the Godot foundation needs to guarantee access to materials that enable developers to work with the engine. The responsibility of this test is to ascertain the possibility of finding and reading such materials.

- Access Donation Page

  - Similarily to sponsoring, monetary stability may be achieved through smaller contributions from individuals that want to support the Godot foundations work. This also enables people that may not have the required knowledge or experience to aid financially rather than submit code. This was a more complex test, not only checking whether there is a possibility to donate, but also ensuring both monthly as well as one-time payments are accepted.

- Search For Specific Item In Documentation

  - A fundamental part of documentation and encyclopedia-like documents is the ability to search and find the required information through minimal effort from the user. The test written for this case encases the scenario where you find the documentation page, use the search box for finding the sought after information, and test the functionality of opening a found result.

- Find Resources For Bug Reporting

  - Even thoroughly-tested codebases and programs do inevitably end up with bugs. Allowing users to report these bugs can not only further cover cases that the tests have failed to flag, but also develop and better the tests for increased coverage in the future. This test confirms that a way of reporting these bugs is available to the users of the game engine.

- Finding Older Versions Of Godot

  - It is important to acknowledge the need for a versioning system in combination with a way to download earlier versions of the product. This is a feature that is especially important for applications that themselves interact with other systems as conflicts may happen. For our use case, this ensures the user can continue working on their game in an older version of the engine should their project not be portable to newer versions due to bugs or project-breaking changes.

- Changing To Translated Version

  - For a wider usage of the product, it is crucial to have correct localisation in case the user doesn't understand the english language. This case in isolation may not be very relevant, however in combination with the previous nine cases, helping non-english speaking individuals partake in the project may increase the collective value in all other use cases.