|  |
| --- |
|  |
| CS 1632 – DELIVERABLE 3 |
| Web Testing with BDD: NeoGAF.com |

|  |
| --- |
|  |



Github URL: https://github.com/swanc12/deliv3.git

Colin Swan – CRS110@pitt.edu

March 3rd, 2016

# Why Neogaf

NeoGAF (New Gaming Age Force) is a community forum dedicated to providing fans of everything related to videogames a central location at which they can discuss the industry. It is one of the most active forums for video games around, and boasts many prominent professionals as members of the forum. It is often a source for news in the world of videogames, with several well publicized leaks originating with posts there. This has led to the meme “GAF -> Internet -> GAF”, referencing the cycle of news being posted on NeoGAF, then being picked up by videogame journalists, and then being reposted on NeoGAF when someone sees the article without realizing it sourced the forum.

The reason for its success is often attributed to its relatively strict moderation policies. It is one of the few sites that does not have automatic member creation; each membership request must be approved by an administrator, which means becoming a member can often take up to a month. And when one does become a member they are limited in what they can do; until they have been a junior member for three months, and have posted over three hundred times in threads, they are unable to create new topics on their own. Administrators are merciless about the terms of service as well, leading to several threads known as “graveyards” in which scores of people are banned for accidently breaking the TOS one after the other. The harsh system fosters a relatively civil atmosphere on the forum, and encourages people to discuss their ideas and opinions in depth.

The restrictions on membership creation were certainly a negative to choosing this site to test, as I knew going in that I would be unable to test any member functionality (as I don’t want to give up my account on a public github repository!). But I felt that the positives were enough to outweigh that. The site is relatively simple, with very little dynamic content. It is also free to use, and I have been using the site for years, giving me a level of familiarity with the layout of the site. I also did not want to choose a site that a lot of other people would be testing, though the reasons for that were probably mostly subconscious in nature and not really logical. That said, the only real negative to testing the site that I could find was the inability to log in, as mentioned, but I did not think it would prove to be too much of a hindrance for the purposes of the project.

# Difficulties

The biggest difficulty was the one that I already mentioned – that is, not being able to log in to test member functionality – as that somewhat limited the complexity of the user stories that I was able to work with. Most of the interesting functionality of the site is solely the domain of members, leaving only relatively straightforward tests available to me. The next big hurdle then became avoiding the pesticide paradox; there were a couple tests that seemed repetitive on the one hand, but then on the other hand I could easily envision scenarios were one test would break and the other would not. So I often just left the tests in even though they seemed potentially unnecessarily repetitive.

Another difficulty that seemed pretty common was figuring out how to select certain elements that lacked distinct id attributes, or other values by which they could be uniquely identified. This ended up making a few tests a bit more complex than I felt they should have been, especially where I was forced to use xpath expressions to pick out a button here or there. It was also somewhat more difficult to ascertain how to select an element inside of an element, though I eventually learned how to do this via some google searches. Going forward the biggest obstacle would once again be the lack of membership access, as there are only so many features of the site than can be tested without membership.

## Code locations:

My primary code can be found at <https://github.com/swanc12/deliv3.git>. The src folder has a test file for each user story, and they can all be run at the same time by compiling and running the SeleniumSuite.java file. Each file has the user story under test commented at the top, and each test case has a scenario commented out.

The website under test can be found at <http://www.neogaf.com/forum/>. I do not have direct access to the source code for the site.

# Test Execution Results

