The Benchmark

A benchmark is a universal tool to test the integration of a certain program. We are porting a specific program from JavaScript to Java and want to keep the outcome of the two programs the same. So a benchmark would be the perfect solution to this problem.

We started with the JavaScript version of Gamygdala from Joost Broekens for the Java port. We began to make an one to one port to Java and write tests for those lines of code. After the initial port, we saw that there we big SEM related problems, like god functions and god classes. A fine example was the Gamygdala class which had over 750 lines of code. There were also a lot of functions in the wrong place, like Agent functions that we written in the Gamygdala class. We first fixed that to get a better overview over the code. Then we took on the challenge of refactoring the Gamygdala port so that the program inCode wouldn’t give us very significant errors. By rewriting some classes and functions, and making new classes, we managed to get as little as 2 errors and they were about Data classes. Data classes are not very significant for code optimization and sometimes needed.

After all of the rewriting and making new code to optimize the working of the program, we wanted to know if our Java port still did the same as the JavaScript port. We asked Joost Broekens if there was any tool to check the working of the port. He emailed us an JavaScript-based emotion engine benchmark. We first ran it against the JavaScript Gamydala version and saw that the benchmark didn’t pass the benchmark completely. Comparing the outputs of the benchmark against our own results would be enough to see if our port worked the same.

Because the benchmark was in JavaScript, we needed to make a connection between Java and JavaScript. After some searching on the web, we found a Java Script Engine called Nashorn. With Nashorn you can call JavaScript function in Java and the other way around. Luckily for us, the benchmark was well written and we only needed to change the mapper.

After we changed the mapper we saw that our benchmark results differed from the JavaScript benchmark results. Some tests later we came found the error. It had to do with the change from Strings to Objects to keep track of Agents. In the JavaScript version the agents where saved by their name, but in the Java version is better to work with Objects. So after some changes in the Java port code it was all good to go.

Looking back on the purpose of the benchmark, it was very useful to check the integration of the Java port against the JavaScript version. The fault that had occurred by refactoring the Java port was not noticed until the benchmark. And because the integration of the port is very important to us, we were glad that the benchmark existed.