6 Languages research

J. Mitchard

In order to achieve our goal of creating this simulation, we will need two separate systems. One that is written in a concurrent language that will act as a message switchboard and run the algorithms that control the intelligence behind the nodes in the the simulation, and a second that has a good, well documented, but not too heavy 2D animation library that we can display a visual representation of the simulation.

6.1 Simulation Languages

6.1.1 Go(Golang)

Though, in terms of programming languages, Go is very young, this has quickly become a popular concurrent language because of its similarities, in coding style, to C and because of its open source nature. This would be a good language to use for the simulation because of its asynchronous nature, but none of the members of the project have used it before so it would be a big risk to the completion of the project if it took longer to learn than expected.

6.1.2 Java

Java is an Object Orientated Programming language, so it may seem strange to include it in a list of concurrent languages. However, Java is a very adaptable language and through the use of Threads it is possible to synchronise multiple tasks so that they happen concurrently. For the scope of our project, this is not ideal however, as it doesnt allow proper communication between the threads.

6.1.3 Erlang

Erlang is a functional and concurrent programming language that deals very well with a high load of processes. Concurrency is handled through message passing between individual processes, which makes for very flexible and dynamic concurrency designs possible in simple manner. This is the language we have decided on using for this part of our project because the members of the project group are all competent Erlang programmers and we feel that this language meets our requirements.

6.2 Visualisation Languages

6.2.1 Java

Java meets most of our requirements quite well here; it is cross platform through the use of its virtual machine, we all know it and it has access to a lot of different graphics and animation languages. However, most of these libraries are very heavy duty and may be a little overkill for what is needed for the program.

6.2.2 Javascript(and the D3 Library

Being a web-based scripting language, Javascript has the inherent advantage of being totally cross platform as all modern browsers can run Javascript. On top of this is includes a lot of individual libraries that allow for animation, object control and networking abilities. This makes it a very strong candidate for our project as it meets all our requirements and is a language that we all have a lot of experience in using.

D3 is a Javascript library that allows you to manipulate documents based on data. It is lightweight and is entirely cross platform as it is web-based and follows strict compatibility with web standards. We have decided to use Javascript, and other web based markup languages such as HTML and CSS to build our visual client as we feel that it is the most effective means to achieve our goal.

6.2.3 Python

Similar to Java, Python is primarily an Object Orientated Programming language. Its open sourced and community based ethos has led to an abundance of very good graphical and animation libraries. One library in particular that we have looked into is calle Pygame, whilst this library is very well documented and maintained, it is far more than we really need for this program. Another problem is that Python is not crossplatform in the same way that the previous two languages are, however compilers for all widely used platforms are available.

6.3 Conclusion

So, now it has been decided that we will use an Erlang based server that will deal with all simulation handling and process management, and will be represented in Javascript to create a cross platform system that negates the problem of having to install it onto multiple operating systems.