

UNIVERSITY OF NOTTINGHAM
SCHOOL OF COMPUTER SCIENCE & I.T.

INDIVIDUAL DISSERTATION
G53IDS

as part of G400
COMPUTER SCIENCE (BSC HONS)

**A HTML5 Framework for
Multi-Agent Algorithms
Project Proposal**

by

Ben JENKINSON

bxj08u / 4082995

October 28, 2011

1 Aims and Objectives

The aim of this project shall be to produce a framework to both animate and inspect the progress of a multi-agent algorithm. My intention is to create this framework leveraging HTML5.

The visualiser client would be distinct from the application or process that is actually running the algorithm simulation, with nothing to connect them but a defined data-format to describe the state of the algorithm simulation. This separation would mean that there is no difference between running the simulation and the client on the same computer, and running the simulation remotely. If the client has been written in HTML5 as a single-page web application, then there is no reason why the client cannot be hosted publicly as well and accessed from anywhere in the world.

This would open up the possibility of running a complex simulation on a supercomputer, with restricted access, and then providing public access to a visualisation that would be updated in real-time as the simulation progresses. This visualisation could then be designed to be as attractive and intuitive as possible so as to better illustrate the process of the simulation to the user, who may not be of a technical mindset. This arrangement would work quite well for showing a non-technical stakeholder the process and the progress made while the simulation is running over a long time-period.

Of course there is nothing that says the datastream has to be located on another computer, it could just as easily be a local source for a local client.

This client could then effectively 'tune-in' to a stream of algorithm state-data output by a completely different machine. This would allow multiple researchers or clients to view the algorithm progress in real time while the intensive computation is done elsewhere. Independantly, each client could then pause and inspect the data of the visualisation at any time without interfering with any of the other clients.

All the while, the client could continue downloading the live stream as it is output so that the user can then skip around the timeline of the simulation from whenever they first joined the stream.

2 Project Plan

2.1 Tasks

- Define a protocol, API or data-format for communicating the state of a multi-agent algorithm simulation.
 - Research types of multi-agent algorithm.
 - Research a variety of problems that can be solved by multi-agent algorithms.
 - Acquire some example problems and example algorithms to use in development of the visualisation client.
 - Define the protocol/format ensuring it would successfully describe the example problems and algorithms.
 - Write explanatory documentation for the protocol/format.

It would be wise to limit the scope of my project to a sub-group of problems that would suit visualisation. Perhaps those areas for which an in-progress inspection of the algorithm is particularly useful.

- Investigate the potential for a server-side component for the production of the algorithm data.
 - Research existing software for the simulation of multi-agent algorithms.
 - Identify whether any of the existing software could be adapted to generate the data-format that has been defined.
 - If it is possible, adapt an existing simulation to provide the required output.
- (*Further*) Develop my own component to generate the required output stream of simulation data.
- Design the visualisation client, capable of reading a stream of the defined data-format and buffering it for playback through an ideally attractive and intuitive interface.

- Develop the ability to read in the data-stream from a defined location.
- Use HTML5 offline storage to buffer the data-stream for playback.
- Develop a clear and attractive visualisation for the simulation data.
- Develop an 'inspector' for the visualisation, whereby playback can be paused and any particular agent can be inspected.
- (*Further*) Draw upon my research of existing visualisation software to determine other useful features for the client to possess, then add them.

2.2 Milestones

*All dates are rough estimates, plan is subject to change. Official dates are shown in **bold***

2011

28 OCT Project proposal & plan

- 4 NOV Research - Multi-agent algorithms.
- 11 NOV Development - Definition and documentation of protocol/format.
- 25 NOV Research - HTML5 technologies to read and store the protocol/format into the client.

28 NOV - 2 DEC Presentation

- 16 DEC Development - Basic client capable of reading or receiving the protocol/format.
- 30 DEC Development - Client is fully capable of storing the data from the protocol/format and skipping forwards and backwards through time.

2012

- 6 JAN Research - Use of existing simulation software to generate the protocol/format/feed.
- 27 JAN Development - Design and develop the visualisation of the simulation to be both attractive and intuitive.
- 10 FEB Development - Adapt/develop server-side component to generate the protocol/format from the simulation.
- 24 FEB Outline & first chapter**
- 30 MAR Development - System should be fully working.
- 7 MAY Complete dissertation**
- 16 MAY Demonstration**