2 Requirements Specification

J. Mitchard

2.1 High Level Requirements

We would like to create a program that simulates a possible zombie apocalypse using crowd-swarming algorithms in a concurrent manner. We aim for this to be realistic as far as we can make something that is still science-fiction realistic, based on research of the topic in science fiction and realistic biological effects considered. This simulation application would allow the user to change different parameters, such as changing the density of the populated area or the speed in which the Zombies can move, that would affect the way the simulation ran.

The software should be able to display a visual representation of a humans and zombies swarming together, attempting to survive. For each type of character, Zombie or Human, this will be achieved differently. Though different by nature, these two different characters will share similar qualities, or states, such as being hungry. However, they will differ in that Humans become tired, whereas a zombie uses less brain power and internal systems and act for longer period of time.

In terms of the intelligence of these two different entities, the human entities will be considerably more intelligent when it comes to things like finding the route of best fit to a location, or navigating obstacles. On the other hand, a zombie would just try and get from A to B without thinking about it in a particularly logistical or strategic manner. This should be represented within the simulation. We would like this application to be cross platform, though until further research has been carried out we will not limit ourselves to defining how this is achieved. It has been considered that we display the visualisation of the simulation through the means of a web browser or through a Java like system that uses a virtual machine.

2.2 Low Level Requirements

Based on the inherent lack of graphical apis with the majority of concurrent languages, we have considered that a decoupled client-server approach could be the best way in which to run the simulation. For example, carrying out the decision making and work in a concurrent language and sending this to a front end client written in a different language.

A message passing model of concurrency is going to be more efficient for our concept to be modelled than something like a thread based system, as quick communication between the individual processes will be a central point of our system. Certain requirements we have made about the languages used are that the language;

- Does it's own memory management
- Is efficient,in context to it's role, to a reasonable level

'Visual Representation Client'

- Has access to a well documented,complete 2D graphical animation and window management library.

'Simulation'

- Is concurrent
- Can deal with a large amount of separate processes
- uses message passing