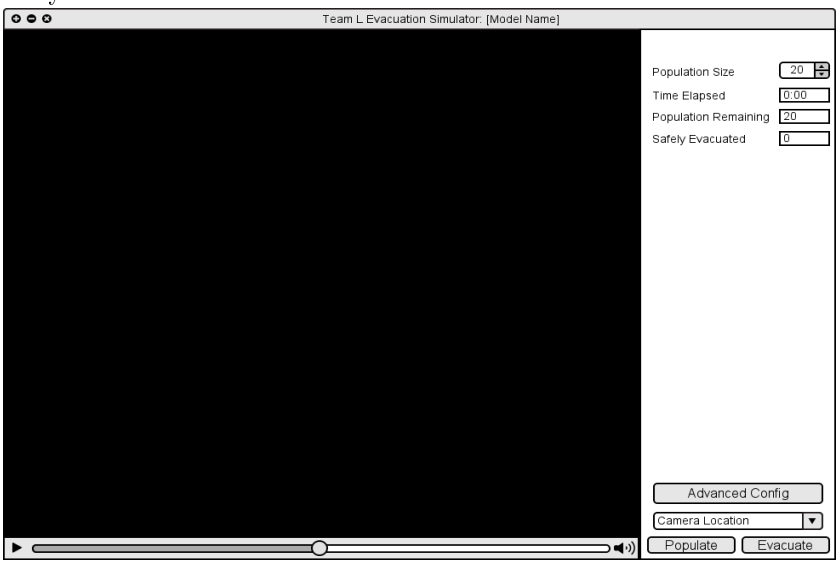
GUI Design and Implementation

The target userbase for the evacuator does not guarantee a good degree of computer-literacy. Hence a clear and easy to use GUI was required.

Initial Design

Based from initial system requirements a wireframe for the GUI could be generated. To keep the user experience simple it was decided to keep all the basic functionality within the main window. This allows unfamiliar users to gain from the software without being overwhelmed by endless options. For advanced configuration there was to be a detailed configuration panel allowing advanced users to set the majority of variables as they saw fit.

The wireframe below was drawn up to support these concepts. By sticking to conventions it gives a familiar feel to most who have used a typical windows based system within the last decade.

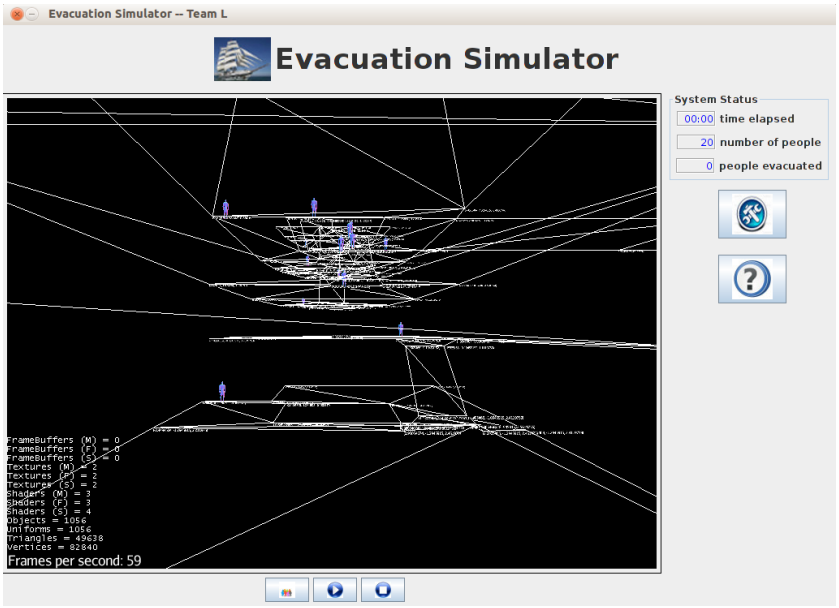


GUI Implementation and Revision

It was decided since the project was to be built in java and several members had knowledge of the library to use Swing for implementation of the GUI. The toolkit was more than capable of our needs and any advantages of alternatives seemed unlikely to counterbalance the time required to learn a new environment. SWT was briefly considered since the interface created generally allows a closer to native feel and higher performance (http://www.ibm.com/developerworks/grid/library/os-swingswt/). This however would come at a cost of portability as well as the time consideration mentioned above.

The GUI implementation was a quick and crude attempt towards the initial wireframe. Its purpose was more towards checking how well the JMonkey can-

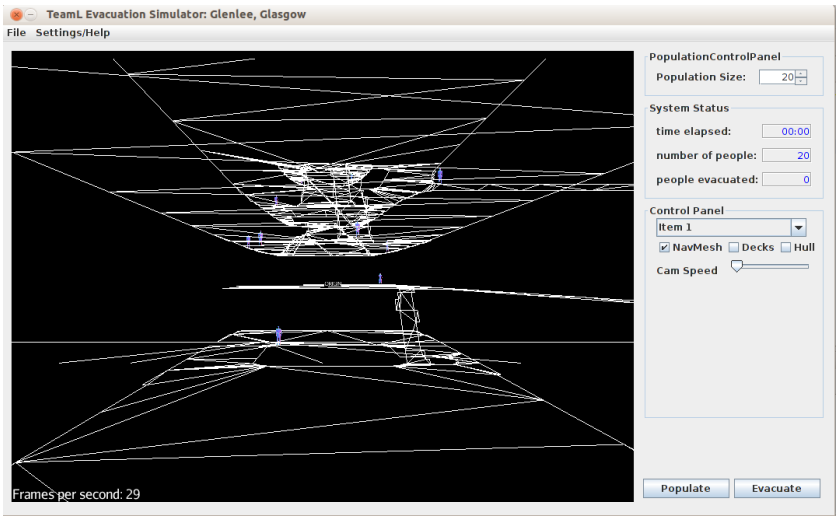
vas could be displayed within the Swing layout than producing a suitable end product.



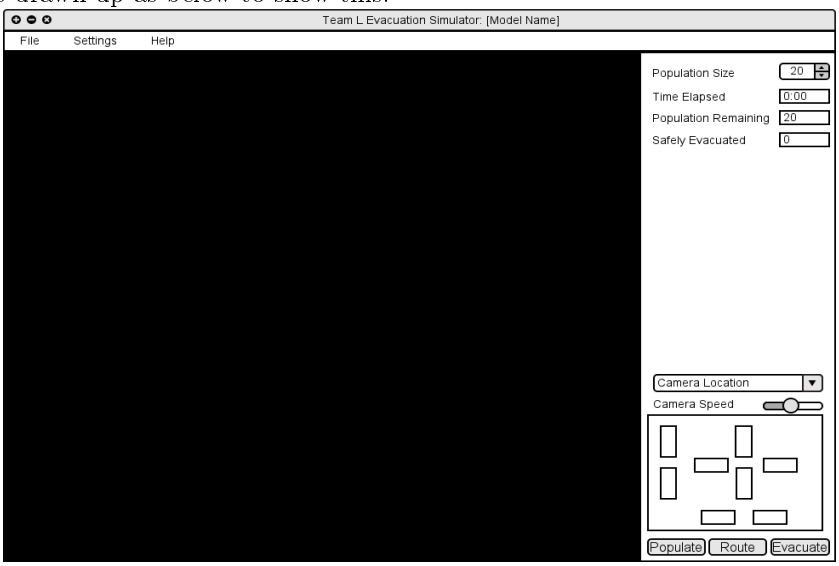
Whilst not particurly attractive a suitable platform for basic testing was then available.

At this point it became apparent that the back end would not easily allow for the evacuation to be "scrolled through" in the way we had initially anticipated. The play bar at the bottom of the screen was removed from designs. this Led to the control buttons being moved to the panel on the right. After speaking to some less computer-literate users it was further decided to use a traditional menu-bar. Some also found the nav-mesh didn't give an immediate understanding of what the model represented so toggles were added to allow a physical representation of the ship to be shown.

These changes manifested themselves as can be seen below. Due to more time being spent on this iteration it gives a much more complete feel and is considerably closed to the initial wireframe.



Further talks with users suggested that for those not used to computer-games the keyboard based camera controls were not immediately understandable so a control panel was added. Due to the way functionality was implemented a route button was added for testing purposes. It was then decided this would be of use to users so was left in. Finally, to allow a more native feel the theming was set to inherit from the system the code was run from. A new wireframe (below) was drawn up as below to show this.



Final GUI

The wireframe above was implmented to give our final GUI. It gives a polished finish and meets original aims well and takes into consideration user feedback.

Initial work was done using the Netbeans GUI builder. This allowed the layout we required to be built quickly, but not the functionality. To achieve this some work was needed on the raw swing code.

The final interface is shown below, both in linux and windows.