# How was the program designed?

# What issues were encountered and how were they resolved?

# What special features are in your program?

# What issues, if any, arose when you were doing the maintenance part of the assignment?

# Testing Details

# Suggestions for Improvement

The biggest improvement that could be made is not related to the actual construction of the program, but is instead to do with time management. The major working periods on this program were upon initially receiving the assignment and during the few weeks leading up to due date, this was due to underestimating the time required to do tasks and the number of tasks involved. Doing it differently we would ensure progress is made each week thus enabling us to better judge what exactly is required to be added and the sort of time required to complete those tasks.

There were of course many program related areas in which we could greatly improve also, the largest of these would be memory management, as we were low on time much our code is functional rather than well written and set out, therefore many objects are not deleted after their use is fulfilled so more memory is continually taken up as the program runs. The biggest memory sink was related to reading in models using GLM, every time a model was required it would be read in by GLM then displayed, even if the same model had already been used before. This was a huge waste of resources as nearly every in game model is used very often. What we would have liked to have had was a class that managed the models so that one model was only read in once and then could be used anywhere.

Multiple classes were also poorly written, with some instances of public variables being used simply to get things working. Had we the time we would very much like to overhaul most of the classes to get them running in compliance with low coupling and high cohesion.

The Finite State Machine AI also needs improvement as currently each AI uses only one behaviour, this behaviour handles everything that AI does instead of transitioning between multiple behaviours. Splitting these single behaviours would have made the code much more readable and understandable. This also currently defeats the purpose of having the finite state machine class.

Given more time we would also increase the current scope of our game, by adding more AI and more abilities so that the game play could be mixed up more to keep it fresh, currently the only change is difficulty as the AI come out at a faster and faster rate, this means the game gets stale very quickly.

# How is code reuse taken care of?

# Can the assets/objects/items be reused easily in other programs?

How would this reuse happen?