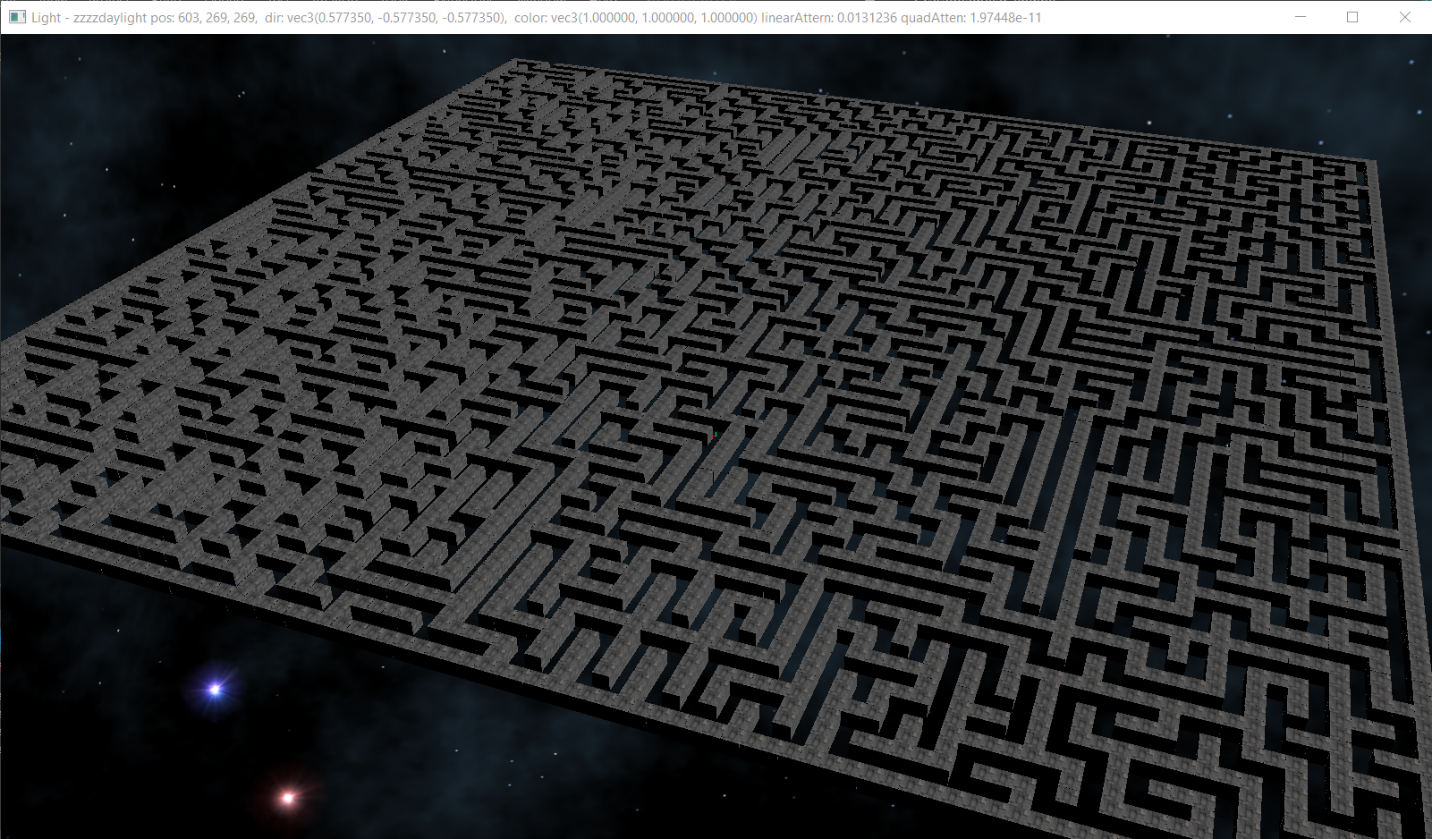
# Algorithms and Gems Midterm

# José Daniel Casado Aguilar - 0940167

# Showcase of different grouped steering behaviours.

## Answers

**Question 1:**

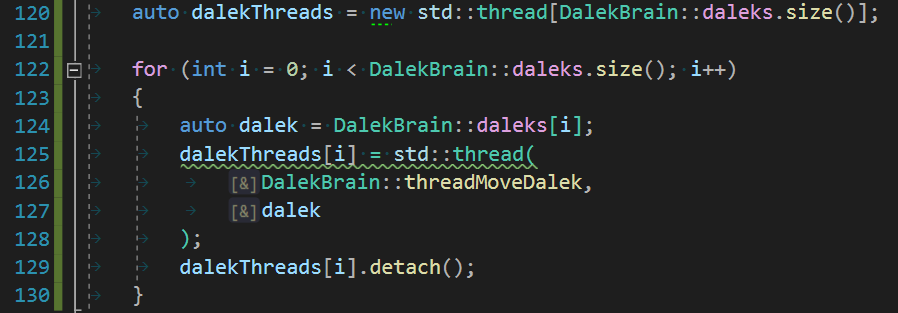
****

**Question 2:** *Question2.mp4* video in the project folder

**Question 3**

*Question3.mp4* showcases the final behaviour of the movement of the daleks. I will explain more in detail the code that creates the threads and makes the movement thread safe:

In the file *theMain.cpp****,*** in line 120-130 I initialize an array of threads; one for each generated Dalek:

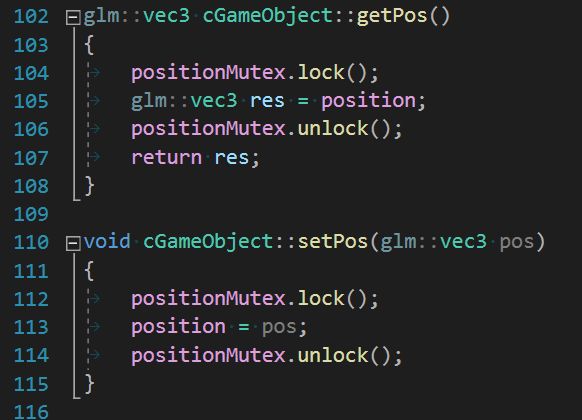


The function each thread runs, it’s an infinite loop that runs another function that moves the Dalek one “frame”, checks if it has arrived to the destination and generates a new destination if it has.

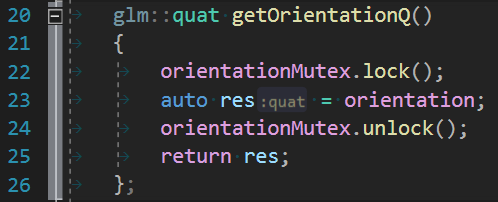
The *threadMoveDalek* function is defined in *DalekBrain.cpp* line 67-73. The function that actually moves each Dalek is defined on the same file, line 75-97.

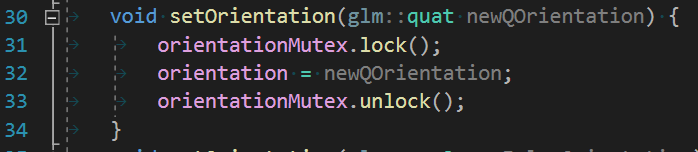
I made the position and orientation of each gameObject thread safe by using **std::mutex**.

You can see this in the file *cGameObject.cpp* on the functions *getPos* (102-108) and *setPos* (110-115):



Mutex are also used in the *aOrientable.h* file in the *getOrientationQ* (20-26) and *setOrientation(quat)* (30-34):





## Build instructions

# Open *j\_casadoaguilar\_AI\_P2.sln* with Visual Studio. Select *Release* and *x64* and then *Build Solution* (Ctrl + Shift + B)

## Run instructions

# Within Visual Studio, click on *Local Windows Debugger* (F5)

## Use instructions

The daleks start to go automatically. The camera fixes on the “first” dalek in the collection. You can go through the dalek list with the **up** and **down** arrows.

With the **a** and **b** keys you can turn the camera around. With the **w** and **s** keys you can jaw the camera up and down. With the **q** and **e** you can zoom in or out to the “selected” dalek.

The movement of the daleks is a little jumpy because I couldn’t figure out completely how to use the proper delta time with threads.