Multithreading the update and draw loops

The update look will use displacement structures ... even these can be toned down from what i have

The draw loop will use DrawingStructures and are very simple.

Though about several Solutions ... the one im most happy with at the moment is this

We do not mind if the draw loop lags slightly ... the update loop should always be faster than the draw loop ... if not the draw loop will simply redraw the same image (or even better ... do nothing)

Because of this the update loop will always operate on the central objects and the draw loop will operate upon a list of structures generated by the update loop.

These structures are everything necessary for the draw loop to draw. They need to contain a reference to a draw method which can be achieved through an interface (IDrawable), this method needs to accept the DrawingStructure as an argument. This allows each object to have responsibility for its own rendering. The DrawingStructure also contains the models position and rotation as vector3 and quaterion as values. In the draw method.

The draw loop in the DrawingLoopComponent will contain a list of DrawingStructures, each representing a and have access to the camera object to draw through. It will perform the logic to determine which objects are in view (if the camera moves than all objects need to be tested for FrustrumVisibility again, if not only changes matter in comparison to existing FrustrumVisibility state).

This list will get updates when the update loop adds to a queue of changed objects, new objects and objects to be destroyed.

Both of the threads get a reference to the queues passed to them on creation. The only thing that need to be ‘multi-threaded’ here is the queue ... but as one thread will add and the other remove there isn’t a problem here either.

Multithreading inside these threads can be considered – however i am cautious as context switching is a \*bad\* thing. If you exceed an optimum amount (for the program – load will vary with time) the performance plummets. With no way to precalculate this varying edge things are very difficult.

The main use for it will be File IO / networking etc anything we need asynchronously and is order independant.