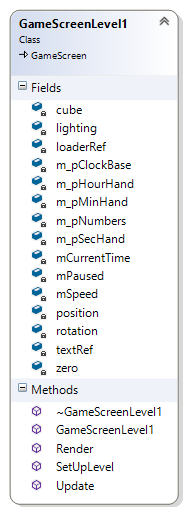
Robert Pullen – Clock

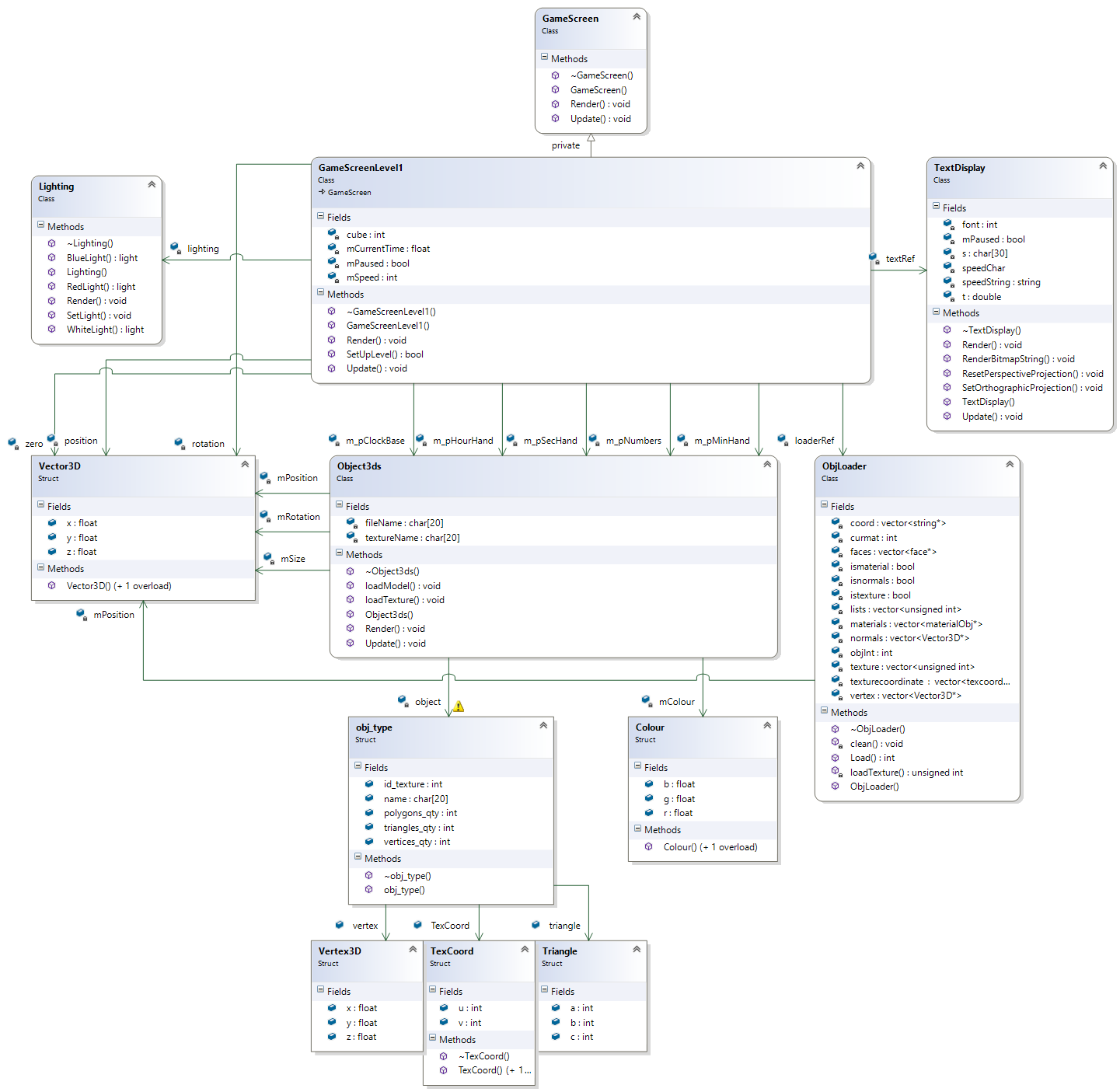
The aim of this project is to re-create an analogue clock in C++. In the simulation, the clock moves in real-time with 3 hands; a second hand, a minute hand and an hour hand. There is an adjustable speed on the clock, which can be used to speed up/down the flow of time. The clock can also be rotated around to view it from different angles.

The controls are:

|  |  |
| --- | --- |
| Key | Action |
| A | Rotate the clock on the Z axis |
| D | Rotate the clock on the Z axis |
| Q | Rotate the clock on the Y axis |
| E | Rotate the clock on the Y axis |
| W | Move the clock closer |
| S | Move the clock further |
| R | Reset the speed |
| Left Arrow | Decrease speed |
| Right Arrow | Increase speed |
| Space | Pause |



This is an overview on my code in UML format. As you can see from the diagram, I held variables in my GameScreenLevel1 class. This is where I would set/change the Object3ds/Obj parameters. I held variables, such as position and rotation, in a Vector3D struct. I set colour using a Colour struct as well. This made adjusting them much easier than and kept my code neater. I would then pass through these variables into the classes via the Render() function, which would be applied there. I used the Update() functions to control time for my clock, with the deltaTime variable in GameScreenLevel1. In Object3ds, it held an obj\_type struct to load in models. It would hold variables such as the amount of triangles and vertices, which were kept in their relative structs.



Overall, I’m pleased with the outcome of the project but I wish I could’ve done more with it. With the coding, I’m pleased with my use of structs and classes. At first I wasn’t sure of the benefits with structs, but I quickly learned how useful they were after starting the project. With my classes, I think they all have efficiency and all are easily used.

I would say the efficiency of my code is average, as I came up with numerus bugs. These were most notable in the ObjectObj class. As you can see from the screenshot, the cube on the right is missing triangles from the model and this was a bug I could not fix. If I had more time, I would re-write my Obj classes and make them much more similar to the 3ds classes I have. Another problem I encountered was loading in models initially; they would not be visible to the camera. After debugging I realised the models were being created at an increased scale, and the camera was inside of them, meaning they would not be seen.

If I had more time, there would be several things I would add or improve to this project (as well as the Obj classes). One problem I feel the scene has is that it’s quite basic. My initial idea was to have a plane that could fly around a clock tower, which would run at real-time. The problem I had (apart from the big scope of it), was that I underestimated the code I would need to write. I would also like to implement a hierarchy for my scene, but unfortunately I ran out of time.

**References:**

ObjLoader Class/ ObjectObj Class: <https://www.youtube.com/watch?v=XIVUxywOyjE>

This tutorial was linked on the [AGEC\_Prac8\_Loading\_3DS\_Max\_Models](https://blackboard.staffs.ac.uk/bbcswebdav/pid-1815039-dt-content-rid-5085914_1/xid-5085914_1) tutorial, and I used this code as a base when I created mine.