CronusGL (FOGGS Semester 2 Assignment)



Conecpt for my application was to make a city/town like a scene where player can roam it either from inside of a tank or float around it. Application in current form allows player to switch on the fly camera mode as well as control tank from within tank camera.

# Controls (floating camera):

* WASD – navigate around the scene
* Mouse movement – rotation around the scene
* Right mouse button – menu that allows to quit game or switch camera mode
* ESC – quits game

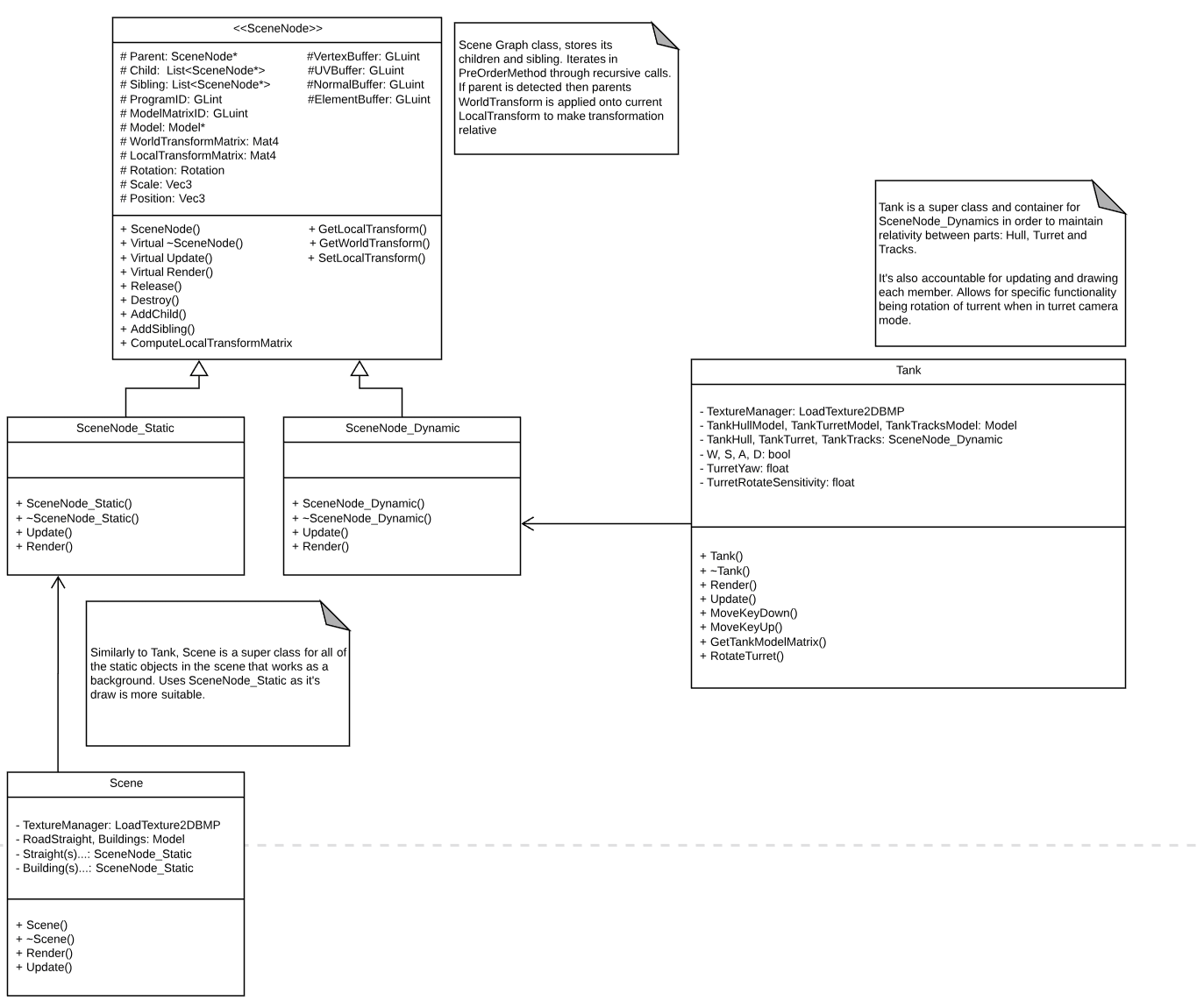
# Controls (tank camera):

* WASD – navigate tank around the scene
* Mouse movement – rotate around tank as well as rotate turret meanwhile
* Right mouse button – menu that allows to quit game or switch camera mode
* ESC – quits game

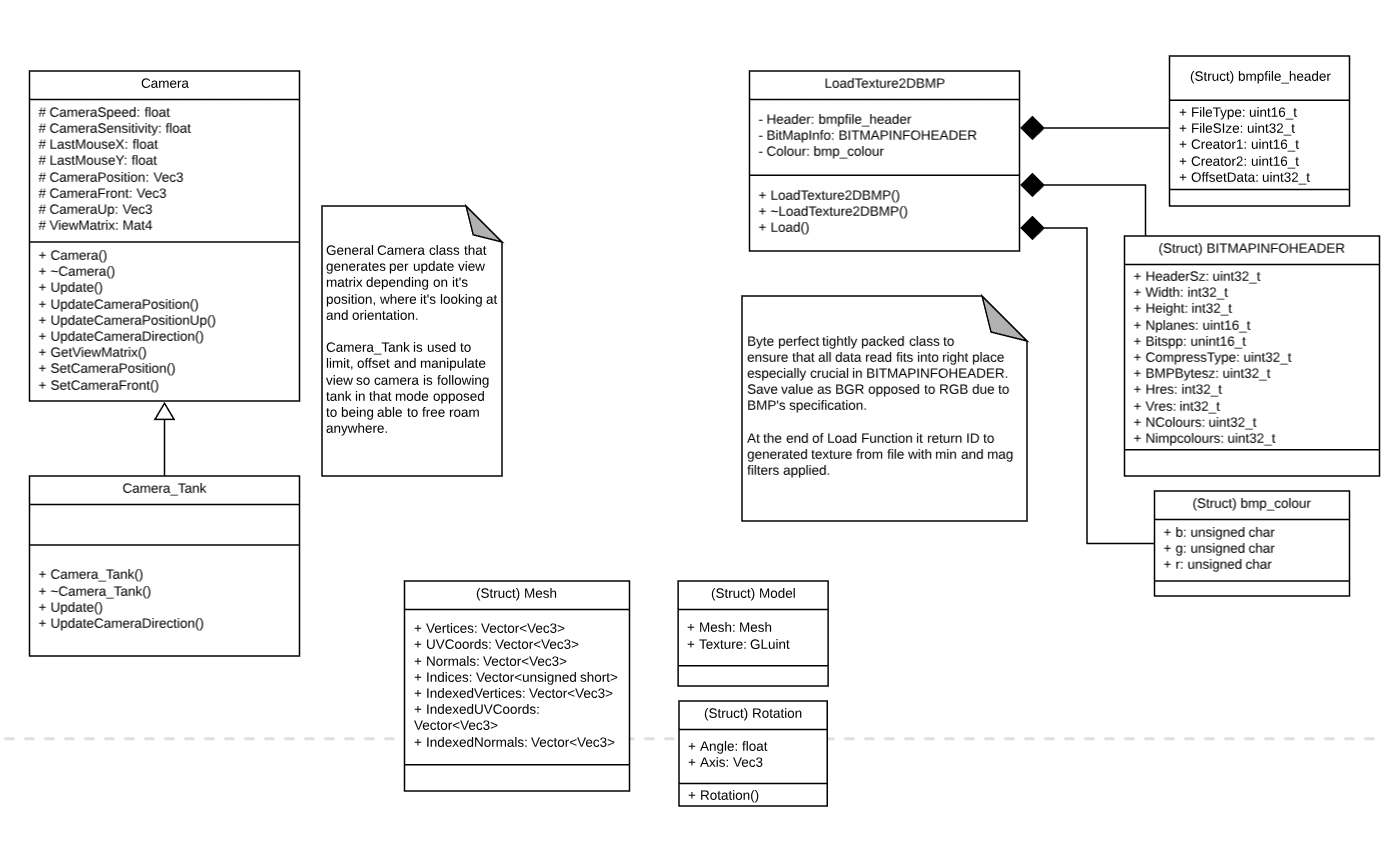
Reflection

Development process of this application was extremely hard for me as had to grasp a lot of new concepts at once not only from OpenGL field but Game Development overall. Didn’t make my job at all easier by deciding to use modern OpenGL opposed to legacy which was in tutorial documents. On Brightside I do feel like I’ve earned much more by going through this path instead. I’m particularly proud of my SceneGraph class as well as OBJ loader, SceneGraph because I’ve managed to make it so transformations get passed down the hierarchy therefore achieving relativity between objects.

As for OBJ Loader it will be because I’ve managed to achieve not a trivial task at all relatively well disregarding it being strict about what can be passed in and what not. I think it’s overall see-through and easy to read which mattered to me a lot as there’s a lot of things that happen in there. Regard what could’ve went better definitely will be my super classes for scene graph as feel like they could’ve been created in much more managed and maintainable way opposed to just getting stuff done. Another thing that could’ve been improved would be my pipeline of communicating and sending data down to shader, I feel like I could’ve made it more centralised which would make it much easier to manage opposed to current way which is stitching data together in places far apart from each other or even isolated.

UML Diagrams

UML Diagrams (cont.)



References

Research.ncl.ac.uk. (n.d.). *Scene Graphs*. [online] Available at: https://research.ncl.ac.uk/game/mastersdegree/graphicsforgames/scenegraphs/Tutorial%206%20-%20Scene%20Graphs.pdf [Accessed 3 May 2019].

Webglfundamentals.org. (n.d.). *WebGL - Scene Graph*. [online] Available at: https://webglfundamentals.org/webgl/lessons/webgl-scene-graph.html [Accessed 3 May 2019].

Archive.gamedev.net. (n.d.). *GameDev.net -- Understanding and Implementing Scene Graphs*. [online] Available at: http://archive.gamedev.net/archive/reference/programming/features/scenegraph/page2.html [Accessed 3 May 2019].

Mufti, A. (2019). *Implementation of Binary Trees in C++ - TutorialsPanel*. [online] TutorialsPanel. Available at: http://www.tutorialspanel.com/implementation-of-binary-trees-in-cplusplus/ [Accessed 4 May 2019].

de Vries, J. (n.d.). *Learn OpenGL, extensive tutorial resource for learning Modern OpenGL*. [online] Learnopengl.com. Available at: https://learnopengl.com [Accessed 1 May 2019].

Opengl-tutorial.org. (2011). *Opengl-tutorial*. [online] Available at: https://www.opengl-tutorial.org [Accessed 1 May 2019].

En.wikibooks.org. (n.d.). *OpenGL Programming/Modern OpenGL Tutorial Load OBJ - Wikibooks, open books for an open world*. [online] Available at: https://en.wikibooks.org/wiki/OpenGL\_Programming/Modern\_OpenGL\_Tutorial\_Load\_OBJ [Accessed 29 Apr. 2019].

Solarianprogrammer.com. (2018). *C++ reading and writing BMP images | Solarian Programmer*. [online] Available at: https://solarianprogrammer.com/2018/11/19/cpp-reading-writing-bmp-images/ [Accessed 29 Apr. 2019].

Free3d.com. (2015). *Residential Buildings Free 3D Model - .obj .blend .fbx - Free3D*. [online] Available at: https://free3d.com/3d-model/residential-buildings-83244.html [Accessed 4 May 2019].

Free3d.com. (2018). *WWII Tank France AMR 35 v1 Free 3D Model - .obj .stl - Free3D*. [online] Available at: https://free3d.com/3d-model/wwii-tank-france-amr-35-v1--291595.html [Accessed 4 May 2019].

Turbosquid.com. (n.d.). *free obj model road street cities intersection*. [online] Available at: https://www.turbosquid.com/3d-models/free-obj-model-road-street-cities-intersection/611350 [Accessed 4 May 2019].