

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | Implementation Log – Graphics II | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Sam Croft100485878 |  | | | |
|  | | | | 04/11/2019—5CC510: Graphics II— |  | | | |
|  | | |  | | |  | | |

# Starting Point

Leading up to this point, we were provided with a basic Direct3D 11 framework in which to begin adding code and utilizing overrides of Initialise, Update, Render and Shutdown methods.

# Week 3 – commencing 11/02/2020

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Implementation | Problems found | Solution |
| 11/02/2020 | Collection used to store Scene Nodes. Vectors are efficient at accessing specific elements, like arrays and are dynamically allocated, reducing complexity. However, usually allocate extra storage and thus use more memory. Lists have better iterators, but also can consume more memory. |  | Chosen to use a vector for simplicity and reasonable efficiency |
| 11/02/2020 | Implemented SceneGraph.cpp containing methods of Initialise, Update, Render, Shutdown, Add, Remove & Find. |  |  |
| 11/02/2020 | Included a class, inherited from SceneNode, which creates a cube. |  |  |
| 16/02/2020 | Added SceneNodes with seperate transforms in update method to animate a cube robot in order to test. |  |  |
| 07/04/2020 | Introduced a Resource Manager and the ASSIMP library. |  |  |
| 08/04/2020 | Created a TerrainNode Class with a method to generate a terrain grid. | Not finished | Need to create vertex and index buffers, build shaders to generate |
| 14/04/2020 | Code to create Vertex and Index Buffers. | Indices incorrect | Rewritten |
| 17/04/2020 | Added terrain shaders and methods to render terrain | Not displaying |  |
| 18/04/2020 | Bug Testing |  |  |
| 20/04/2020 | Bug Testing |  |  |
| 21/04/2020 | Continued bug testing |  |  |
| 28/04/2020 | Bug testing – Progress | Not displaying anything | Missed the call to build renderer states in initialise |
| 29/04 | Continuing with week 5 | Displays a diagonal line across the screen. | Rewriting the terrain grid algorithm. |
| 29/04/20 | Week 5 Terrain | Diagonal line / nothing appearing | Removed unnecessary code which interfered with rendering. |
| 29/04/20 | Week 5 Terrain | Displays, but follows the plane model | Didn’t include a constant buffer, so it attached itself to the last scenenode. |
| 29/04/20 | Height Map |  | Height map is loaded and used to update terrain y values. |
| 04/05/20 | Week 5 Terrain Lighting Implemented | Bad use of UINTs. Would enter negative values, sending array indices out of range. | Casting to ints.  Also, American-english spelling of colour in shader file. |
| 04/05/20 | Camera Class Implemented | Set Camera Position function throws an exception. | Camera position was being set before constructed. |
| 04/05/20 | Keyboard Control Implemented. |  |  |
| 07/05/20 | Written code to load textures and algorithm to apply a blend map to the terrain texture. | Terrain renders very dark. Doesn’t seem to be textured. |  |
| 18/05/20 | Investigating dark textured terrain | Could be an issue with the cBuffer struct, it doesn’t match the struct used for the shaders.  Blend Map also doesn’t seem to blend textures. |  |
| 18/05/20 | Implementation of SkyNode | Not rendering.  Downloaded RenderDoc to help diagnose the issue. |  |
| 19/02/20 | Investigating SkyNode not displaying, blendmap and lines across terrain using RenderDoc.  A close up of a logo  Description automatically generated  A screenshot of a cell phone on a table  Description automatically generated | Seems that at the edge of the terrain, the next polygon is being drawn from opposite ends of the mesh, creating the strange lines. They’re so thin, I wouldn’t have seen them in wireframe without the heightmap. | Looked again at the code that generates the index buffer. Used the terrain width, rather than depth in the algorithm. Now renders all polygons correctly, although texturing is still very dark. |
| 19/02/20 |  |  |  |