Lab 05 AStarNodes

**Fufillment**

* I finished all the objectives!
* I think I may have done extra, but the requirements were semi-ambiguous/very open ended, so I’m not entirely sure what to consider extra and what to consider just well-done.

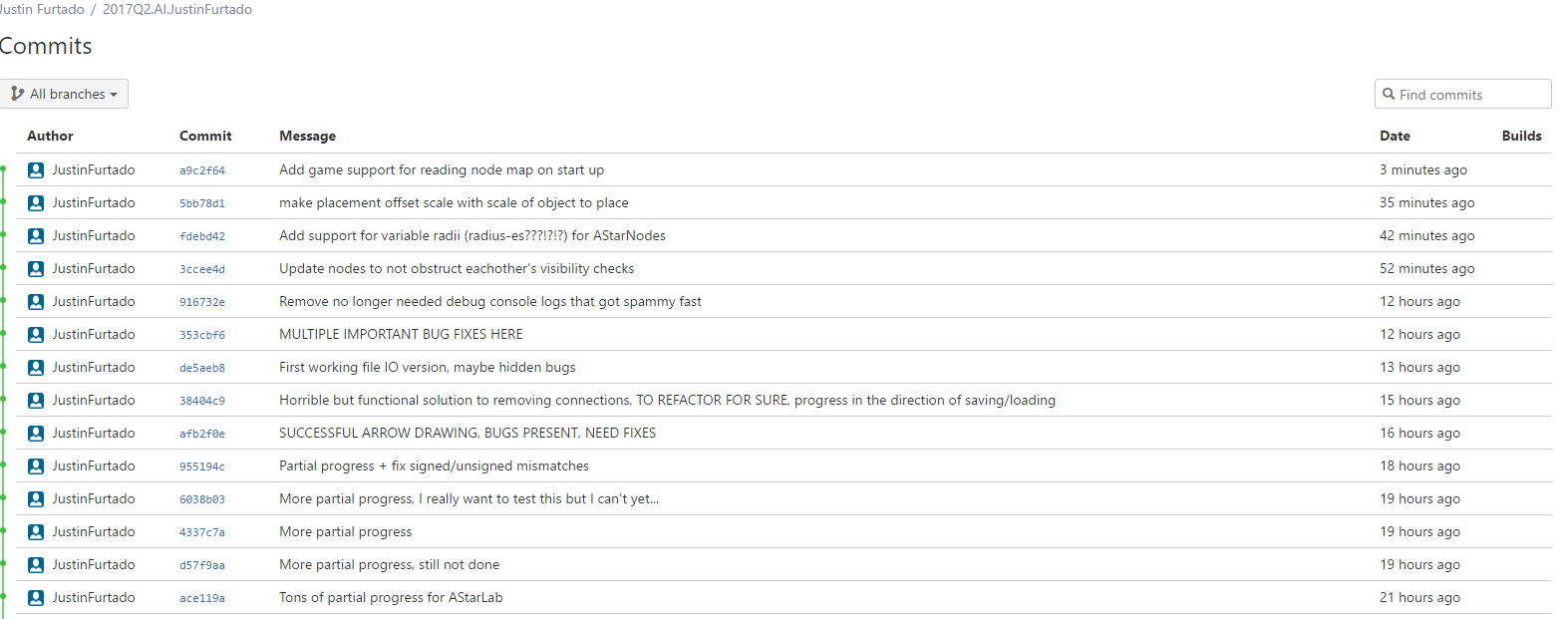
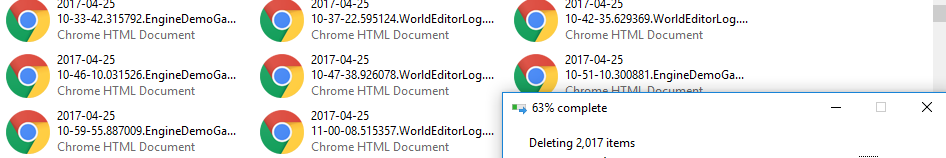
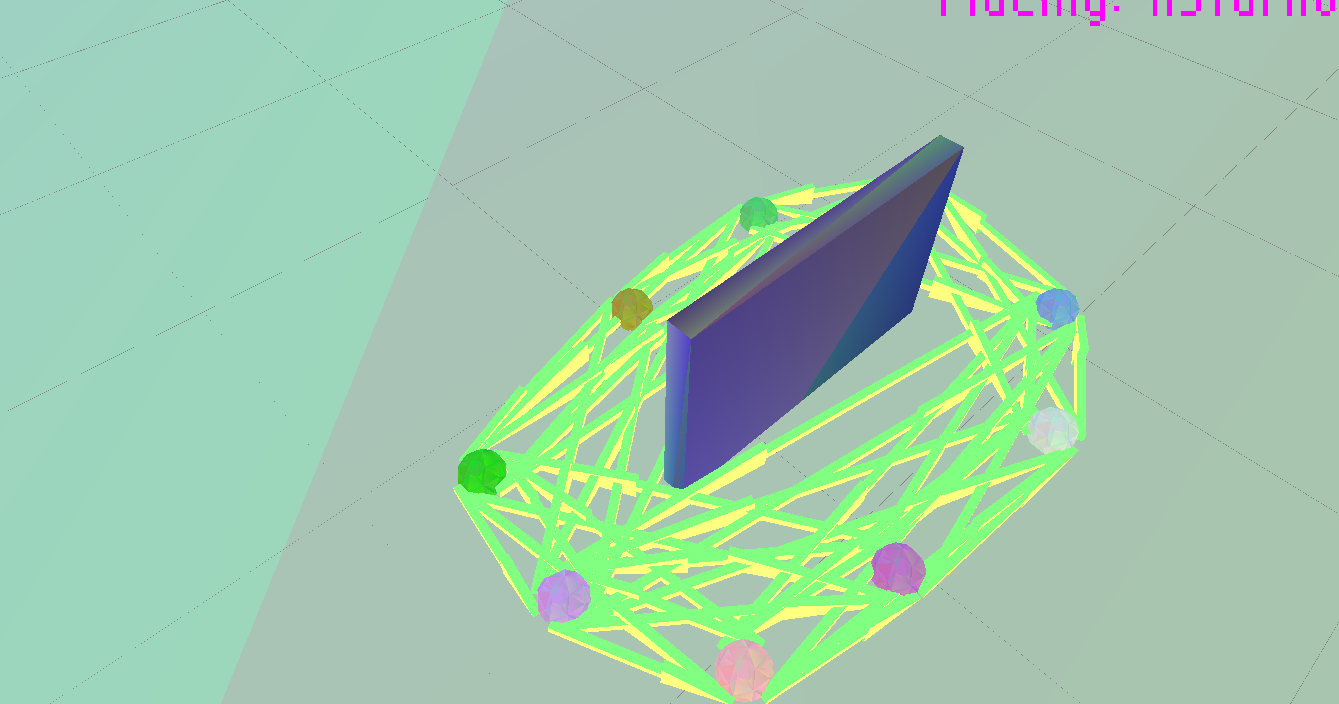
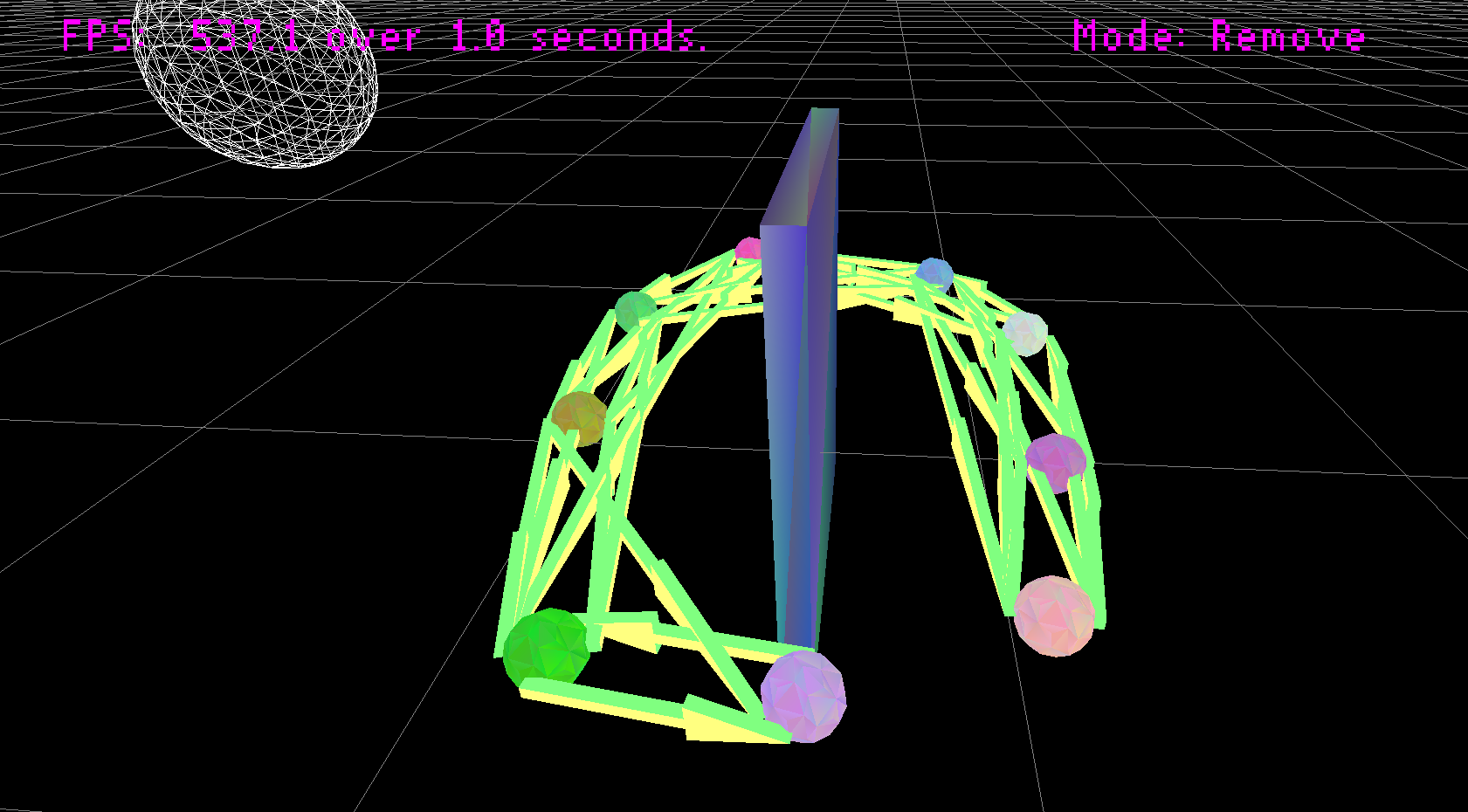
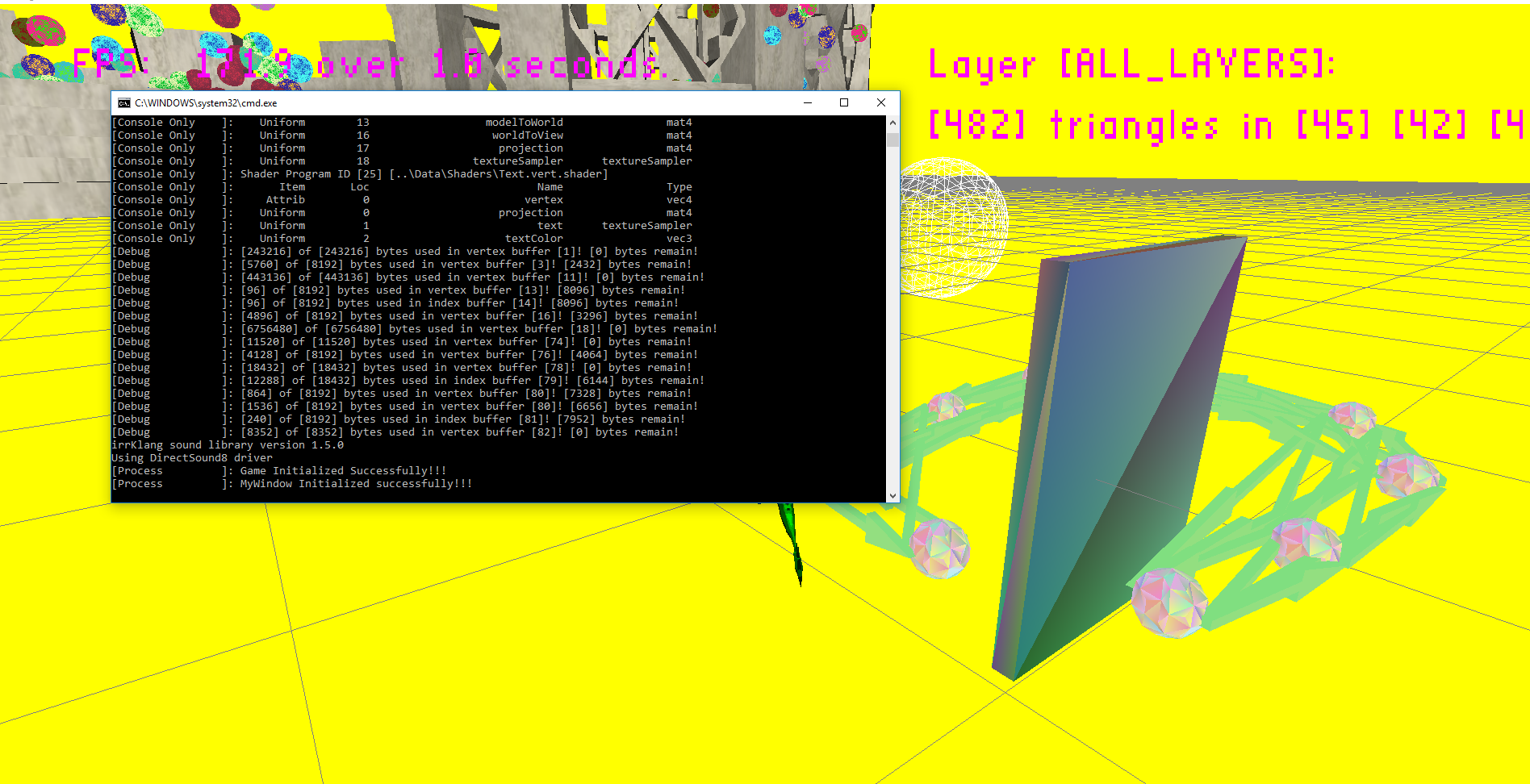
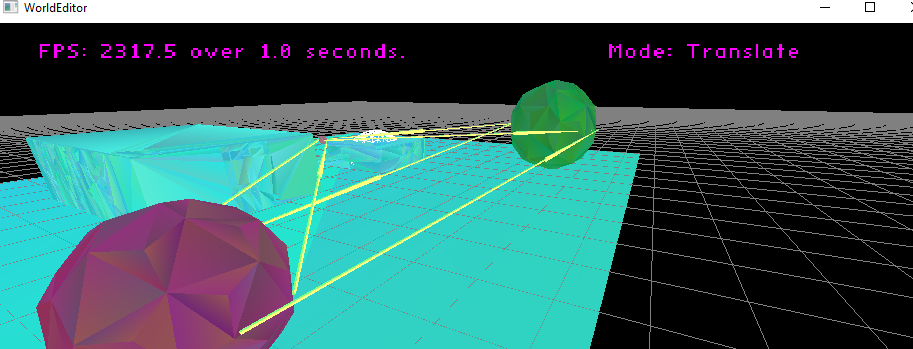
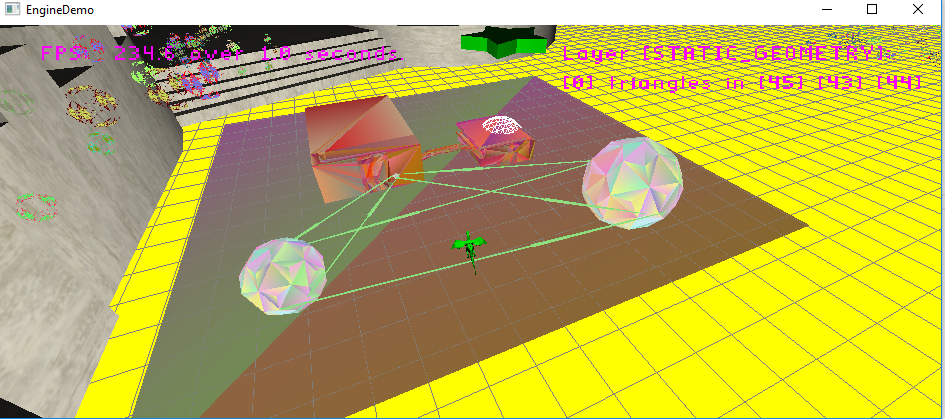
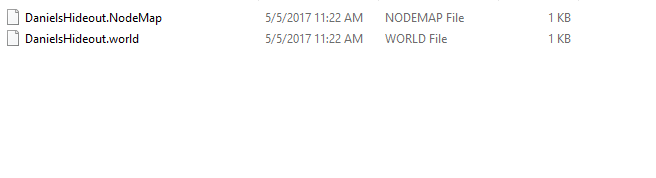
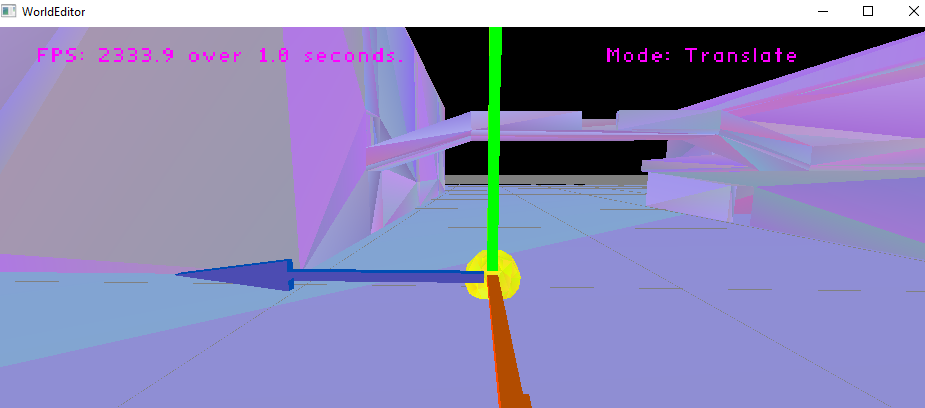
**Execution**

* I do not believe there should be any unexpected requirements for running
  + Actually, if you count setting the start-up project in visual studio to world editor or game (or back and forth) that could be a requirement

**Controls (World Editor)**

* Pressing the X key will close the application
* Pressing W will move the camera forward
* Pressing S will move the camera backward
* Pressing A will move the camera left
* Pressing D will move the camera right
* Holding right click and moving the mouse will turn the camera
* Pressing 1 will swap to object place mode
  + The selected object to be placed (displayed in the top right as text) will be placed with its origin at the intersection point of whatever you click on, unless no objects are present then it will be placed at 0 0 0 when you click
* Pressing 2 will swap to object remove mode
  + Objects will be highlighted red when moused-over, and removed when clicked
* Pressing 3 will swap to object translate mode
  + Clicking and dragging on the arrows will move the object in the respective direction
* Pressing 4 will swap to object rotate mode
  + Clicking and dragging on the arrows will rotate the object in a semi-kinda-sorta-maybe-not-but-still-better-than-nothing-rotation-method
* Pressing 5 will swap to object scale mode
  + Clicking and dragging on the arrows will scale the object in those respective axes
    - There is sometimes a reversal with rotated objects, so if this bugs you, scale them before rotating them!
    - NOTE: AStarNodes only support uniform scaling, because I’m the developer and I say so!
* Pressing 7 will move the object to place forward
* Pressing 8 will move the object to place backward
* Shift + k will calculate all possible routes between the placed AStarNodes based on visibility checks and update the GOBs accordingly
* Pressing Shift+9 will save the scene, including BOTH the World file AND the Node Map
* Pressing Shift+0 will load the scene, including BOTH the World file AND the Node Map

**Screenshots**

* + **My version control history, as of about 11:00am – I coded almost all day yesterday and finished up this morning – this only shows the second iteration, not the hundreds of lines of code I threw away earlier… (Actually I have done more bug fixes since then, but this was the bulk of it)**
  + 
  + **Oh my goodness, a new record! 2K log files! I didn’t expect that many, but I guess I did do a ton more “run, see bug, fix it, run again, find new bug… etc.” this time around.**
  + **Nodes with all visible paths calculated and drawn**
  + 
  + **Node removal**
  + 
  + **Loading the nodes into the game – note that I do not save the node graphical objects, and instead re-calculate them from my data structure. Also note the removed nodes are not saved/replaced**
  + 
  + **World editor with variable sized nodes, about to be saved out**
  + 
  + **Game with variable sized nodes loaded in**
  + 
  + **The very tiny files produced for the above read/write**
  + 
  + **Moving nodes in the world editor**
  + 

**Post-Mortem**

* **Well, first things first. I had to start over. I vastly overcomplicated the issue by trying to make my nodes too aware of each other and their connections, and tried too hard to make it overly dynamic and have support for removal of connected nodes without re-calculating and moving the nodes making it auto update and so many more unnecessary things that I now realize was just silly.**
* **Having approached the problem a second time using the methods described in class (plus my own modifications, of course) I now realize that it is a superior method and I wish that I had thought of something similar before spending so much time wasted on a failed attempt.**
* **I spent nearly twenty hours on this lab in total, about half and half on each attempt**
* **I ran into so many issues that were tiny and silly, such as**
  + **Forgetting to draw the objects I created**
  + **Misplacing the arrows**
  + **Making them too long/short**
  + **Forgetting to re-calculate my spatial grid when things are added/removed**
  + **Decrementing a counter twice by accident**
  + **Forgetting to increment a counter**
  + **Failing to condense holes in the array properly**
  + **Accidentally removing the wrong connections on click**
  + **Accidentally clearing the node map on load**
  + **Trying to write out pointers to files**
  + **Deleting things twice**
  + **Trying to set the color of a deleted object**
  + **And tons more…**
    - **And from this I learned, take more breaks, don’t try to code everything in one sitting, you get tunnel vision and make mistakes you would not make if you were coding at full brain power**
* **Overall I’m not too sure why this one was difficult because in retrospect it doesn’t seem like it should have been, maybe it was just a misunderstanding of requirements/expectations or a really lousy work session (I’ll try more coffee next time?)**
  + **Or maybe my engine wasn’t ready for this kind of thing?** 
    - **I still have no idea how Brandon could think this is a one-hour sized assignment**
      * **10 hours for my successful attempt, minus like 6 of those that were mostly debugging leaves like 4 absolute minimum**