Lab 01 Shader Reboot

**Fufillment**

* All of the requirements are met with minimal extra conveniences added (BetterDargon can fly around and look at the scene, some useful things like movement and camera speed are config values)

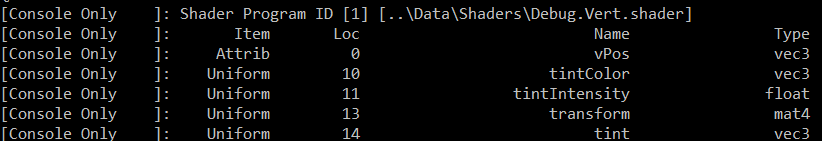
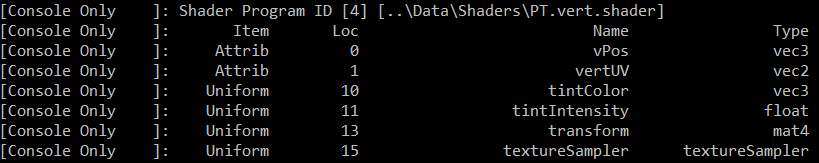
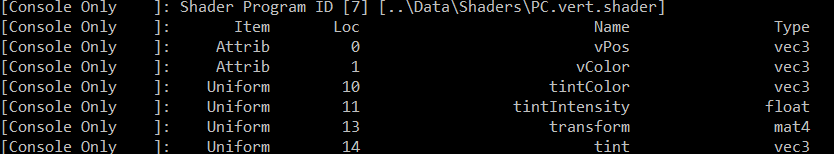
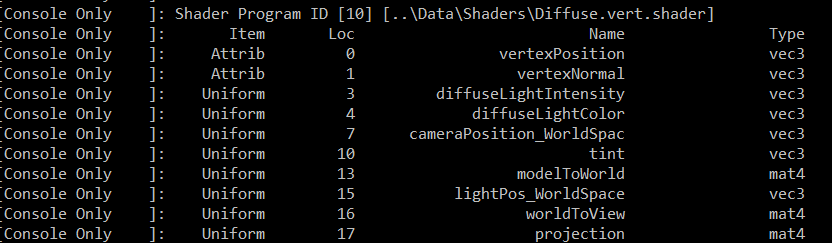
**Execution**

* I do not believe there should be any unexpected requirements for running

**Controls**

* Pressing the X key will close the application
* Pressing the P key will pause the application, pressing it again will un-pause the application
* Pressing numpad 0 will re-read the config file
* Pressing M, L, T or C will dump engine info to the console, this is pretty much exclusively used for debugging
* Pressing W will rotate BetterDargon to the left, S will rotate him to the right
* Pressing A will tilt BetterDargon forward, D backward
* Pressing Q will roll BetterDargon to the left, E to the right
* Holding space will move BetterDargon forward, in the direction he is facing, releasing will halt movement
* Right clicking and dragging the mouse will turn the camera around BetterDargon
* Scrolling in or out with the mouse wheel should zoom the camera accordingly, up to a minimum or maximum distance

**Screenshots**

* This is the output for the debug shader from the shader parser
* 
* This is the output for the PT shader, which was used for Regular Dargon in the demo, but I let it parse for testing purposes anyway
* 
* This is the output for the PC shader, which BetterDargon is using in the demo
* 
* This is the output for the diffuse shader, used by the teapots
* 
* Following is a screenshot of all three objects and all three lights
* 
* A close up of the first teapot (reflectivity = (0.0f, 0.0f, 0.9f), light color = (1.0f, 1.0f, 1.0f))
  + Note that when hit by a white light, it reflects only blue!
* 
* A close-up of the next teapot (reflectivity = (0.9f, 0.9f, 0.0f), light color = (1.0f, 1.0f, 1.0f))
  + Note that when hit by white light, it reflects only red and green light, making it appear yellow
* 
* A close up of the final teapot (reflectivity = (0.75f, 0.75f, 0.75f), light color = (1.0f, 0.0f, 0.0f))
  + Note that when hit by a red light, it reflects a portion of the red
* 

**Post-Mortem**

* I encountered little-to-no difficulties with this lab – it may have taken more time than was expected, but at no point did I have an error I could not rectify within five minutes.
* I would have done nothing differently for this lab, I went straight for the requirements and have met them all, perhaps I would move some code around for organization purposes, but I’m not sure if it’d be worth it depending on how the future labs are structured – it may not be necessary
* The lab doc could have gone more into detail explaining the methods we had to use for the shader parser we had never seen before. The documentation in the textbook and on the internet was awful, and I had to combine like five sources with trial and error before actually figuring out what was going on.
  + Of course, now that I have done it once, it will be a million times easier in the future
* The only thing I felt was relatively cool is that each object had its own light. We hadn’t done that before – it was easy, but still nice to look at.
* If all you were going for was a simple get-us-back-in-the-groove-of-things-while-doing-something-we’ve-done-before-but-introducing-three-new-opengl-methods-along-the-way sort of thing, you hit the head on the nail, perfect job. If you were going for something a bit more instructive – perhaps the lab could be re-worked from the ground up to provide a more beneficial experience.