Jack Thake
Karla Fant
CS 202
10/24/2021

Efficiency Writeup

This assignment was a real challenge for me, I underestimated the time it would take and fell victim to having a cool but ultimately way too complex design. I made many mistakes through the making of this program. My plan was to model how a futuristic airport would operate, having the gates planes and ground vehicles communicating. I ended up underestimating the time it would take to turn this idea into reality and ended up having to make last minute sacrifices and then ultimately having to call it as to not sacrifice time for program two. The data structures were implemented, those being a LLL, and a CLL with the LLL living in an array in the test program. The codebase itself at least in my opinion isn't bad. I fully employed recursion wherever possible and the LLL and CLL seemed to be very reliable. The downfall being none of it was built on a sturdy foundation. My location class was meant to be the foundation for all the physical objects in my program. Although it seemed to be working, I forgot to use valgrind through my implementation and found out at the very end that the location class was causing catastrophic memory corruption. This was the start of the downfall of my development cycle, I could not fix this memory problem. The only viable solution with respect to time was to just remove the location's dynamic memory, even though it was a part of the assignment. The next critical failure was the use of an std::vector, at first I had it as a member of the vehicle_manager class. Having the vector only served to further complicate the memory problems, so ultimately, I decided to scrap it – valuing a program that wouldn't segfault over one that would, even if that went against the assignment outline. With this next week having both the proficiency demonstration as well as the midterm and the start to program three, I decided it would be best to stop work on program 1 to be better prepared for both midterms and the second program. Through making this assignment I've learned a lot just about the development cycle that I had forgotten. I've relearned how important it is to use valgrind over other memory tools as valgrind goes in depth about where and what the errors are, rather than just if a leak occurred. As well as employing better time management to these assignments.