Lab 4 README

ECE 13

Sean Manger

Prof: Steve McGuire

F24

Author: <Sean Manger> (smanger@ucsc.edu)

Collaborators: For this lab, I discussed concepts of Matrix Math with Donny Tang. We discussed approaches to the various functions and how they might be able to be implemented. For instance, we discussed how MatrixSubmatrix() might be implemented & I suggested that some for loops, if(), & continue statements might be the way to go in implementing this function. However, our discussions were on a purely conceptual level and all code in Lab 4 is my own original work which I began on 10 /18 and completed on 10 /20. We also discussed debugging and compiling errors along with where hardcoding in Matrix Math may or may not be appropriate. Again this was on a purely conceptual level. I found our discussions engaging and they ultimately helped me succeed in the lab.

Lab Summary: The goal of this lab was to implement a Matrix Math library in C. This included functions for things such as finding the determinant and inverse of a 3x3 matrix. I thought the most important part of this lab was making sure you were indexing through the array properly. For instance, when doing matrix multiplication I initially thought that two for loops would suffice however I ended up going with three nested for loops. The first 2 simply iterated through the array while the third kept track of what numbers I was at which allowed me to successfully implement multiplication. I also thought that the continue statement was invaluable in finding the submatrices of a 3x3 matrix. The continue statement turned what could've been a nasty if-else tree into an elegant looking nested for loop. I think the manual does a good job at discussing indexing but I wish it would've stressed indexing tricks, and techniques a bit more.

Overall Approach to Lab & Difficulties: For this lab, I thought that a disorderly approach was the best approach. What I mean by this is rather than coding the matrix functions in order I decided to code what functions seemed challenging or interesting to me at a given time. For instance, I did the MatrixSubmatrix() function before working on the MatrixDeterminant() function. I could've easily just sorta hardcoded in the determinant for various matrices and gotten the same result. However, by doing the submatrix function first I was able to reuse it in the determinant and inverse functions which saved me a lot of time. I think that there is great benefit in tackling challenging problems first rather than last. Namely, the lab becomes easier as it goes on as opposed to increasing in difficulty. If there were some things that I think could be improved in my lab I would start with the MatrixPrint() function. While it outputs fine I wish it would look a bit nicer/cleaner. However, I didn't think it was worth wasting precious time on something trivial like this when more important functions needed to be coded. If I were to do this lab again, I'd start a day earlier, edit MatrixPrint() & clean up my code a bit. Thankfully however, I was able to successfully implement all functions

Feedback: I spent probably about 24 hrs in this lab. I didn't find it horribly challenging but it was time consuming coding/ testing each function. I thought the lab was a great experience and I certainly feel like a much better coder after completing it. I thought that the lab manual was excellent and very detailed as to what I needed to do and how I might possibly implement it. I also thought that in class examples greatly prepared me for this lab. At the moment I can't think of any improvements but I will suggest them if I think of them. Overall, I thought the hardest part of this lab was just getting everything to work. Everything thankfully was successful in the end and I passed all of my tests for each function.