sievenna Development Log

Onni Aarne

January 16, 2018

1 First Week

Thursday 21.12
Locked down the topic and named the project.
Created this document and the GitHub repository.
Wrote a basic draft of the specification document.
Wrote this log.
Registered on labtool.

Next week I'll start to actually write the program.

This week I spent maybe 5h on the project.

2 Second Week

I did project setup sporadically throughout the Holidays.

As I traveled on 27.12 I started researching the actual algorithms.
Learned some stuff about information theory and the theory of compression.
Kinda learned how Huffman coding works.
Need to look deeper into all of that stuff.
Decided to implement a simple byte-level Huffman coding algorithm as a first step.

Over the next two days I worked on implementing said functionality.
Ran into difficulties with bit-level I/O.

I've been terrible about organizing my time so it's a bit tough to say how many hours I've spent on this. Traveling and the Holidays certainly haven't helped with that.
I've spent at least 10 hours on this project this week.

With a few hours before the deadline I'll try to learn how to unit test and implement some tests for incomplete functionality. Yay.
Wrote tests for my HuffNode class. Next I need to figure out how to test my I/O class.
Next week I'll implement decoding the Huffman-coded files.

3 Third Week

Sunday 24.12
Wrote some tests for my main method. The program now runs without crashing! The output is incomplete, though.

Friday 5.1. Apologies for being late, should've realized I wouldn't be able to get stuff done in time on the road.
Over the past few days I made the program use Apache Commons CLI to parse arguments and changed the architecture of the program to make more sense.
It's taken more hours than it should have, yet I've put in fewer hours than I should have.

4 Fourth Week

Tuesday 9.1
After days of struggle, the "basic huffman coding" that was supposed to be so easy finally works. I've learned a lot of life lessons about debugging, and that HUMAN MEMORY IS ABSOLUTELY NEVER TO BE TRUSTED. Hopefully now I'll be able to get on with this project to something which will compress an image more than 20%. Next thing tomorrow I'll benchmark the working version some more and decide what improvement to add next. I spent maybe 20-30h of actual active programming time on this this week? This metric will hopefully gain precision and increase significantly from now on.

5 Fifth Week

Wednesday 10.1 Tested the functionality with the Large Text Compression Benchmark, achieved 36% compression.
Added unit test to test compressing and decompressing an image file.
Found that an IO bug resulted in the loss of some bytes at the end of any file, fixed that.

Thursday 11.1
Replaced PriorityQueue with my own min heap implementation. Learned a thing or two about heaps and java generics.
Added some simple prints to diagnose why my program was so slow.
Made all of my I/O classes use buffered streams, which improved speed.

Friday 12.1
Did unconsequential adjustments.

Sunday 14.1
Started to write testing documentation. Realized I need to make a conscious effort to log my time use continuously if I want to have any idea about anything.

6 Final Week

Monday 15.1
Made some more I/O optimizations, cleaned up I/O exception handling.

Tuesday 16.1
Replaced print statements with logging that can be enabled with a flag.

TODO:
For extra credit make it easy to do the above with arbitrary files.
Add some kind of transform, LZW is under consideration.
Make modeling more sophisticated.
Add citations for benchmark files used.