

Matthew Bregg, Brandon Duong, Ian Fell, Ervis Shqiponja

CIS4930

214G

November 30, 2015

Final Project

The work I have done here is my own, and I have neither broken nor bent the honor code.

Learning Experience, Matthew Bregg : This project was something different from anything I've done far. Not actually implementing changed things, as we had to be very careful to think about would something work, as there was no implementation to test if our idea was viable. However, it did mean that some sections what would be awkward, or annoying to right, but known to be possible, we didn't have to worry about, and design is overall faster than code. The hardest part I found was for all of us to properly communicate our ideas, and to choose upon one idea. We spent too long, a good week or so, circling around what to do in the very beginning, and didn't make any real progress until we finally just picked one of the ideas we had in the first day or so, and decided to start expanding on it. The easiest part, I think, was writing the Final Document, once the uml and basic design was done.

Learning Experience, Brandon Duong : During this project, I learned more practical and hands on knowledge about using different design patterns. Not only that but I gained more experience in working in group programming projects. I think the hardest part of the project was trying to all agree and decide on a direction and goal together, since everyone had their own ideas about how to accomplish this. Once we got the ball rolling though, things went relatively smoothly, and overall it wasn't largely impossible since we stayed on top of everything as a team. I think we accomplished what the project was suppose to teach us, and learned alot about practical design pattern use outside theoretical or sample codes.

Learning Experience, Ian Fell : I learned that Parsers are just basically finite state machines and that we have lots of builders and stuff. The builders then call the visitor methods and generates the code. We finished at about 9:00 pm the night before its due and we will probably stay up another hour trying to print it. I also learned that the Design of a software system is a collection of UML diagrams and specifications. Jack Reeves was wrong and the source code is not the design. Our job was to design and not implement and that is what we did to the best of our ability. I hope you find our gobbled mess enjoyable to read. However, please do not try to

implement your SDCG as you might find that the design changes as you begin to implement it. After all, software developers are not engineers, but are a pack of monkeys designing software systems that will lead to utter failure and millions of dollars lost to those companies. If I had to do it over again, I would have come up with a simple/minimilistic design and then begin the development stage using all of the extreme programming software development techniques.

Learning Experiance, Ervis Shqiponja : The easiest part: After deciding the design of our framework, writing the documentations and explaining how each part works and what its purpose is. The most difficult part: Initially coming up with the design of our system. Also trying to design and understand how each part works with each other and how each part fits with one another. Educational Objectives: Better understand how some of the design patterns work and how they would be incorporated with other design patterns or other objects in a larger project. Learned how to use UMLET to generate UML diagrams. Better understand of objects and class relationships among each other.

Matthew Bregg, Brandon Duong, Ian Fell, Ervis Shqiponja

CIS4930

214G

November 30, 2015

Final Project

- \* Does the program compile without errors? Not applicable.
- \* Does the program compile without warning? Not applicable.
- \* Does the program run without crashing? Not applicable.
- \* Describe the ways in which the program does not meet assignment's specifications. Meets all design requirements as far as we know, did not do extra credit.
- \* Describe all known and suspected bugs. Not applicable.
- \* Does the program run correctly? Not applicable.