Ethan Johanson
301076978
Spring 2012

IAT 452 - Final Project
Individual Report

On this course's project, my role within the team was largely in working to develop, refine, and implement the underlying data structure model that facilitates the rest of the project elements' functionalities in terms of being able to have data to operate on and gather information from. Working in Java I developed a series of classes that aligned closely with our structure developed in our proposed class diagrams, adjusting specifics of methods and relationships based on feedback from the instruction team and the needs of other elements of the project. Looking at our early documents from the initial phases, a refinement can be seen in definitions of classes of data and especially in what data can maintain references to what. Our early decisions, while making sense to us at the time based on our understanding of the problem, quickly escalated into extremely complex interwoven systems when the process of creating them was underway. Removing many of these relationships and enforcing a stricter hierarchy of types provided a much simpler system for propagation of information downwards through the system and centralization of controls at the upper level.

Very related to this, the key non-model feature that was my primary task was developing the saving and loading mechanisms. This was accomplished, again through propagation of information requests from a strong hierarchal model down to all children objects, and brought back up to combine into a cohesive and structured set of XML based data. While the saving of this information was a good first step, the loading of that same information proved to be a greater challenge, with limited knowledge of DOM parsing becoming error prone when issues of recursion arose. Switching to a SAX based parser system allowed for tracking of both the beginning and ends of tags, allowing a sense of where in the recursive model the code was at any given point in the loading process, and provided it with a consistently accurate assignment of constructed objects to their proper parents. Working on this XML data structure and what the syntax should look like also allowed us to evaluate the data models of the system and be sure of what the important elements of the system were and what order and structure provided the best sense of cohesion to the data.

Overall I feel that while we may have been slightly ambitious in our initial concepts and ideas, that my team and I have developed a good sense of the development process from this project and managed to complete a satisfying result, particularly considering our various workloads from other concurrent classes. I feel that perhaps a suggestion for future versions of the class would be some more practical exercises in the early weeks developing data structures. While I know this area is covered extensively in IAT 352, and the class diagram exercises are useful, I think that having to actually think through, analyze, and document the differences between attributes of an object and object types is very useful. In our project for instance, our initial concepts had some classes sharing certain attributes and functions that really did not need to be doing so. While in our final implementation we still see remnants of this polymorphism start, it has been restructured to a more comprehensible system. (i.e. character, location, action, and written elements were changed to have a 'type' string that could determine their form rather than all including a variety of boolean flags). Otherwise, I feel the course was well structured and while a bit open ended which led to some apprehension in choosing a project amongst all the possibilities, in context it helped provide a more interesting variety of group projects and helped us challenge ourselves. I have very little ideas on how to improve the current system.