**Unit 6 Reflection**

Date:       May 31, 2017

To:          Mr. Peck

From:      Jessica Peng

Subject:   “APCS-Boggle” Project Reflection

**Accomplishments**

I was never really great at Boggle, but I thought coding it would be a good experience because it would allow me to understand the underlying complexities– such as how computers checked through hundreds and thousands of words and how the board was able to display different activities.

When we were first handed over the Boggle project, I was a little disappointed that I was not able to code an entire game from scratch, but upon starting the project, I realized it was not an easy task. Before even starting to write a line of code, I spent hours reading and re-reading over the instructions and then over each of the classes. I had to tear apart line-by-line each the classes that were provided to truly understand what each class did and how they connected with each other. Even then, it was hard to think of how exactly we were supposed to implement the data structures to use. I was finally able to understand the roles of lexicons in our Boggle Game and how there were different data strucutres, such as TrieLexicon, SimpleLexicon, BinarySearchLexicon, and CompressedTrieLexicon, and how all of these essentially served the same function but each in their unque ways. I was able to complete triedlexicon and write methods that reported the wordstatus in these lexicons. Essentially, what a wordstatus did was allow the search algorithm to determine if what the user entered was a word, not a word, or a prefix. If it was a word, it would return the word. If it was not a word, it would give an “error” or “stop” indication and if it was a prefix it would keep searching. I was also able to implement algorithms to create a lexicon similar to trielexicon, but compressing it and making it essentially more time and space efficient. I am proud that I was able to implement the different data structures and sorting forms of the lexicons and have it be implemented by the other main classes such as GoodWordOnBoardFinder and AutoPlayer.

**Learning Experience**. Like I mentioned before, it took a lot of scrutinizing over lines and lines of code and breaking them down, drawing diagrams, and re-reading the guidelines to truly understand what each class did and what my task was to do. When I was first writing the WordStatus methods I was confused on what a positive and negative return value from the Collections Binary Search method did and why it contributed to the final return statement. I was also not sure exactly what I was supposed to return. But after breaking down code, I understand that when a word is being searched through the datastrucutre, the WordStatus will return whether the word it is looking for is not a word, a word, or a prefix. In addition, trielexicon completely stumped me at first. The data structure was strange– there were ArrayLists, Nodes, and also Maps. They also created such things as “children” (which was the map) and “parent” (which was the node). In writing CompressedTrieLexicon, I had to fully understand these concepts before finishing, and it took a lot of patience and work. Finally, I was able to implement the idea of children and parent and eventually I was able to create the CompressedTrieLexicon which was both time and space efficient.

**Objectives**.

·            Challenge. I think that I was thoroughly challenged in this process by the new types of data structures and complexities of how classes interacted with one another.

·            Effort. I think I worked fairly hard. The hardest part was learning and understanding all of the code that was given to us by Mr. Peck. Then came actually coding the data structures and methods and it was a little easier because there were ideas in other classes such as TrieLexicon with the children and parent that could be used.

·            Quality. I think I did my work fairly well. We met all the criteria given to us in the assignment of Boggle from Duke University, including their compressedtrielexicon extra credit. Everything works, so I think we did a good job.

·            Problem Solving. I think I was fairly resourceful and was able to figure out the code by consulting in Mr. Peck, asking my teammates, and breaking down code line-by-line. I was also able to debug my errors.

·            Results. Witout the lexicon classes, the board would not be able to find words and the game would be pointless, so I think I made a significant contribution.

·            Teamwork. I think I was more of a hustler and was simply able to get things done. Overall, I think that the whole team worked very well together and our tasks were divided equally in accordance to the assignment objectives.

**Overall Assessment**. I think I deserve a 95 because I finished my part successfully in the project and worked well with my team.