Cory Hershman

DuckDuckGoose Post-Mortem

This post-mortem will discuss the steps that were taken to complete the DuckDuckGoose program. The requirements pdf “DuckDuckGoose_fa17.pdf” was dissected and the individual requirements were listed out on whiteboard and a picture was taken. The classes and methods that were required were then listed on whiteboard with pictures taken of those. Then, pseudo-code was written up for each of the classes and methods, with pictures taken of those. Then those pictures were transferred to the computer that was used to code the program.

There were three proposed classes: Node, cLinkedList, and DuckDuckGoose. There were no methods proposed for the Node class, only the variables “element” to hold the data of the node,“next” to serve as a pointer to the next node, and a constructor. There was one variable and five methods proposed for the cLinkedList class: curNode variable to keep track of the node that is currently being pointed to in the list; nextNode method for traversing to the node after the current node; hasNext method to determine if a node had a node after it; append would serve as a method to add nodes to the list; Remove would serve as a method to remove nodes from the list; And isEmpty to determine whether the list was empty. There were five proposed methods for the DuckDuckGoose class: GooseSelection to choose who the next goose was; “isBored” to calculate if a kid was bored; Pokemon would call isBored on each kid and would remove bored kids; chase to determine who would win each race around the circle; and the main method to run the program.

Most of the pseudo-code was easily transferred to java. nextNode and hasNext methods of the cLinkedList class were moved to the Node class as they were more convenient there. This led to some simplifying of the code. However, some methods that were necessary for the program to work. The setNext, getName, getAge, and getGender methods were added to the Node class. A getSize method was added to the cLinkedList class. And the DuckDuckGoose class had the most changes with a listKids method added and some reworking of the main and chase methods. The main method needed the order of which the other methods would be called each round, and the chase method needed a different approach to appending and removing than originally proposed in pseudo-code.

Prior experience making a musical chairs program, and proper design before coding led to minimal problems coding with minimal bugs. Spending a liberal amount of time designing before coding cut down on the overall time spent making the program and is the biggest lesson to bring forward to future projects.