Design Reflection

Program name : Project4_Design_Reflection_Schaefer_Kristin.pdf
Author : Kristin Schaefer
Date : 05-26-2019

Design Changes and Problems Encountered

Project 4 was more challenging in some senses than the previous projects due to that Project 4 built on the program from Project 3, but some changes and rewrites of functions were necessary to adapt to the new project requirements. I found it difficult to restructure the program to accommodate the Team and Loser doubly-linked lists. I had to rewrite several Game functions, and as I was rewriting instead of writing from scratch, it made it more difficult to incrementally test the program.

It took me time to create the design plan, however, I was very satisfied with the results. I made very few changes in the original plan. The most mind-bending part of the design plan was determining how to restructure the program to handle node objects and to create doubly-linked lists to handle those objects. After reading through the discussions on Piazza, I decided it was easiest for me to change the Character class to treat each Character object like a node. Thus, I had to create next and previous pointers for private member variables as well as getter and setter functions for the pointers. All of these were defined int he Character class, and inherited by the derived classes. Surprisingly this worked out during my testing, and I didn't have to do any rewrites. After redefining the Character class to behave like a node, I created the classes Team and Loser to contain define the doubly-linked lists. Thanks to the previous lab assignments I did not have to rewrite any of the Team and Loser functions!

The main flaw in my design plan was that I did not specify where to delete the dynamically allocated memory. I was not sure whether it would be better to delete the memory in the Game class or the Team and Loser classes, but after writing most of the program, it became clear to me that it would be the most ideal to delete the memory in the Team and Loser destructors, reusing code from previous labs. This worked smoothly, until I tried to play the game again. I realized that the same list was being used! I then created a separate delete function in the Team and Loser classes to delete the memory, and then I did not need to delete anything in the destructors. I then switched from using Team and Loser objects in the Game class to using pointers to new Team and Loser objects created in the gameSetup() function. After the tournament ended, I called the delete functions for the nodes of the Team and Loser objects and then deleted the Team and Loser objects themselves. This seemed to solve all of the issues I was having with replaying the game.

In Project 3 I lost points due overusing the main function to call Menu functions. For Project 4, I took extra time to understand how to place these function calls within the Game class functions. Like previously mentioned, sometimes rewriting is more challenging than writing something from scratch, but with a little effort and experimenting I placed the Menu functions startMenu, menuCharacter and menuTeamSize within the gameSetup function (Game class). I then placed the menuPrintLoserPile in the playTournament function (Game class).