Final Project

Due Date: Sunday, 31 Aug 2014, 23:59:59

We will make one last variation of the fantasy combat program. You will make a program to run a tournament. The first player can pick a list of creatures to fight. The second player gets to pick her own list of creatures. The only real requirement for that is that each must pick the same number of fighters. You also need to keep track of the standings. Basically, the last 3 creatures to exit are in first, second, and third place.

By list I mean something like a batting order in baseball or softball. It is NOT a requirement to use a list data structure! This means you provide 2 lists of fighters. The head of each list fights. The winner gets put at the back of his "team's" list; the other is out of play. The order cannot be changed once the tournament starts.

CLARIFICATION:

I also did not specify how creatures recover, if they do at all. Even if a creature wins she may have taken damage. Do you allow the creature to recover for any hits received? If yes, then it's the same as fighting with new creatures all the time. If it's no, then a win might be a pyrrhic victory. Do you restore some percentage of the damage when they get back in line? Do they recover a fixed percentage every "turn" they spend in the line waiting for their next fight? It's up you.

You need to maintain the losers in order to help determine the final standings. Think of a dead pile, or the pile of bad ones after Popeye gets finished.

For this program you must use 2 different data structures discussed in the lectures/chapters assigned. There is s standard container that is appropriate to hold the lineup. Whoever you have in the first slot fights whoever is first in the other lineup. The winner of that combat gets put at the end of the appropriate lineup. The loser will go into the other container so that you can maintain the order they're defeated for the final standings. Part of the assignment is selecting an appropriate tool (container) for your program.

To make sure you are on the right track, do your analysis and preliminary design. Then check with me or a TA to make sure you are thinking in the right direction. ☺ This doesn't need to be a full design document. More of a simple note to the effect of telling us which data structure you propose to use for each of these lineups. Again, I'm avoiding the word list to keep you from thinking that is the appropriate data structure. Since the bulk of the code you will need already exists as containers this design element is a key piece of the assignment.

Remember that this is a programming assignment. Make sure you test your program with instances of all creature types you've created. You may not be able to do an exhaustive test. That would require running lineups in such a way that every creature type fought every other creature type and each type ran through the lineup structure at least one time.

At the end of the tournament you should display the first, second, and third place winners of the tournament. It should also indicate which player "owns' them, and which player won the tournament. This is another element you must define in the analysis and design stage. I recommend doing some research on different standings systems. For example, one of the complaints about NASCAR is that it would be possible for one player to have the first second and third placed fighters, but still lose the tournament. You could also keep track of the individual fights. And if a weaker creature beats a strong creature they get more points than if the stronger whomps on a hobbit. ☺ In your final design document clearly state the system you are going to use and how you will implement it.

You may not be able to just use the winner of the last combat and the top two fighters on the loser pile. It's possible the last 3 or more fighters are from the same team. So you will need some points system to order the fighters when this happens. And a tie-breaker if 2 fighters of the last 3 have the same number of points.

I suggest you just use a simple random draw. There are some professional sports that use that as the final tie-breaker. ☺

To be clear, you only need to send a note explaining which data structures you propose to use. The design document itself just needs to be submitted with the assignment itself.

NOTE: This is a programming assignment! You are demonstrating programming skills. On this extended project some students get distracted by trying to make it fair or interesting. That is NOT a requirement. It is also not an interactive game. Again, some want to allow "players" to change the order of the lineup. If you are using containers that can be a complicated change, maybe requiring significant redesign. You are required only to produce the behavior in these requirements. You can do more if you want, but make sure you save a copy of a base program that meets these requirements in case something doesn't work out.

Enjoy.