Artur Janowiec, Tom Plano, Cliff Anderson, Will Lawrence

Card game Lab – Lab 4

March 17, 2016

About

1. In general, how useful is the list data structure? Can you think of other uses aside from those discussed in the applications chapter?

It is extremely useful because you have a lot of control over the data in the structure. You can go forwards and backwards, and you can easily search the list.

The list data structure can be used in a lot of applications where the user wants to immediately know if the data is in the list or where it is in the list. An example of this would be a bank in a RPG. You want to know whether an item is in the bank by simply typing in the name.

1. If we didn’t have the list data structure, how else could you implement the list concept?

You could make a database, and just store everything like that.

Summary

1. About your team:
   1. How did you “divide up” the work so that each student still met the objectives for the assignment (i.e., learn, understand and apply the concepts). Include details of who did what.

Cliff and Tom wrote the main part of the game, while Will and Artur worked on the ADT.

* 1. How did you coordinate code changes/testing?

We used Github, and we made sure that certain classes are only worked on by one person at a time. This is to prevent merge conflicts, which are not fun. So really only use Github if you have multiple classes to link together, or when you cooperate with someone very really when working on the same class. In this case, that’s exactly what we did.

* 1. Other observations about working with a partner?

Make sure your partner doesn’t rush code, otherwise you’re just going to have to fix all the bugs that your partner makes, and that’s never fun. Always bug test code properly, and if you don’t bug test, then you aren’t done. In fact, a lot of the time software is never done in the business world.

1. Where did you have trouble with this assignment? How did you move forward? What topics still confuse you?

I (Artur) had trouble fixing bugs. The way to move forward is to definitely take breaks when trying to fix a tricky bug. You have to keep a very clear view of the program as a whole otherwise you get focused on a single aspect of the bug even though the bug is being caused by something outside of what you are looking for.

1. What did you learn from this assignment? (Please by specific).

I (Artur) learned that about how double linked lists are created, and even though it’s a bit fuzzy still, I can easily look back on my work as a reference now.

1. How could this assignment be improved in the future?

It would be interesting to have all team members work on the game part of the lab instead of splitting up half the group to do the ADT and the other half has to make the actual game using the ADT and other classes.