Root Wizard

Seth Gilbert and Chris Johnson
UW-Eau Claire

Abstract

Since the beginning of the fall of 2016, I have been working on a game that helps intermediate level programmers have a better understanding of the data structures known as Trees—mainly Min Heaps, Max Heaps, Binary Search Trees, and AVL Trees. I realized the need for a game like this when I was taking a class in my second semester at UW Eau Claire on data structures. In that class, we go over a number of different data structures, and it becomes quite hard to keep track of them. This game intends to help students visualize what these data structures can look like and how they should act.

However, I could have just made a simulator so that students could plug in their numbers or values into a tree and the tree could fix itself or run through animations to show how it would act if something got removed or added, etc. This would be useful, but it would also be boring and students would likely not get very much out of it. There is no feedback involved, no way to mess up and learn from mistakes, and no drive to improve outside of the pressure to get a good grade. Thus, a game a great platform for learning a topic like this. It provides stimulation, a goal outside of simply learning for the sake of learning, and gives nearly instantaneous feedback on whether the player truly understands the topic.

In this game, the player will be presented with a tree, and their goal will be to move nodes around so that becomes the appropriate type of tree, or to perform actions on the tree like an add or a remove, or both. The first few levels will deal with Heaps, either Min or Max, where the player will be confined to swaps and removing only from the bottom-leftmost node. The levels will get more and more advanced through removing from BST's and searching for the proper replacement, and then finally using left and right rotations of sets of nodes to keep an AVL tree balanced. As a reward for each tree that the player successfully completes, they will receive a reward in the form of currency which that can use to buy objects that help them make money faster, and they will also gain understanding on how the data structures behave. The game will get much harder as they progress through it, so they will be continually confronting and conquering increasingly difficult concepts.

This game is still a work in progress, but I think it has great potential to be fun and to help my target audience of students who are trying to understand data structures.