Lab 8

Due 18 April

Handout 8
CSCI 136: Spring, 2004
12 April

Hex-a-Pawn

This week's lab uses trees to play a small chess-like game. You will build a tree representing all possible states that the game board can be in. You will then implement several different players that consult this tree to make moves as they play Hex-a-Pawn. The players include: a human player that asks the user for moves to make; a random player that picks possible moves at random; and a computer player that improves its strategy by learning from past mistakes. In the end, you will be able to run the different players against each other.

The lab is outlined in Section 11.11 of *Bailey*.

Notes

1. The starter files are in the directory `FilesToCopy/HexAPawn`. Familiarize yourself with these files *before* starting to work on your lab. They are described briefly in the book, and the javadoc documentation is available on the assignments page:
`http://www.cs.williams.edu/~kim/cs136/s04/Assignments/index.html`.

2. You need to design the `GameTree` class. This is a tree structure with potentially many children instead of just two. Think about the methods you will need in this class and how you can represent the structure. **Come to lab prepared to talk about how you will implement this class.**

3. You can play a game of Hex-a-Pawn by running `java HexBoard` after compiling the starter files.

4. Turn in your project to the dropoff folder on Cortland as usual. Answer thought questions 1 and 2 in comments at the beginning of your class `GameTree`.

5. The starter directory contains Gardner's original paper on Hex-a-Pawn. There are other learning algorithms described in that paper if you are interested in experimenting further.