Programming Assignment 3

Henry Z. Lo

Due: Tuesday, July 8 at midnight

1 Goals

This assignment will help you with the following:

1. Write a game-playing AI.
2. Learn the game package.
3. Review dynamic programming (memoization).

2 Overview

You will write an intelligent AI for playing games in general. There are several games in the `Game` package, but by writing a `Player` class, you will be able to play any of them in a general way.

You will need to fill in the `findBest` method in `Backtrack` to use the minimax algorithm to find the best move for the given player. You will also write another player `Dynamic`, which does the same thing, but uses memoization to save computations.

3 Instructions

1. Replace `findBest` in `Backtrack` in the cs310 folder, using a minimax algorithm.
   - You may need a recursive subfunction to return ints, rather than moves.
   - Remember to make moves on Game copies, so that you can backtrack.
   - You will need to find out whose turn it is (`game.whoseTurn()`) to find the right move.
2. Use `PlayGame` to make sure your `Backtrack` player works, and write down times for several games.

3. Create another `Player` subclass called `Dynamic`, which memoizes results in the computation tree.
    - Use a map to memoize results for subtrees.
    - Your code should look very similar to BackTrack.
4. Use `PlayGame` to test `Dynamic` and `Backtrack` for several games.
    - Show that they perform the same moves.
    - Show that dynamic saves time.
5. Classes must compile.
6. Put all deliverables in your cs310/pa3 folder. Deliverables include:
    - Backtrack.java
    - Dynamic.java
    - memo.txt
7. Answer the questions in section 4, put answers in memo.txt.

4 Questions

1. Play 2 games (you choose which) with `Backtrack`, save the result.
2. Play the same two games with `Dynamic`.
3. Show the runtime difference between `Backtrack` and `Dynamic`. Use games where this difference is noticeable.
4. Play two AI players against each other.