CSS 605
Monopoly Competition

Due: December 10, 2014 by 9:00am

For the assignment, you are to write the behavior of an agent playing a game of Monopoly. The interface you should implement is the `MonopolyPlayer` interface (you may also opt to extend the `AbstractMonopolyPlayer` class which provides some convenience methods):

```
public interface MonopolyPlayer extends Steppable {

     public void step(final SimState state);

     public boolean considerTrade(TradeProposal tradeProposal);

     public boolean considerCounter(TradeProposal
                                         tradeCounterProposal);
}
```

There is more documentation in the source code, but here is a brief overview of each of the methods. In the `step` method, your player should purchase property, build houses, and propose trade to others. You should also check to make sure your account balance is not below zero to prevent the simulation from mortgaging properties and selling houses for you. If your account balance is negative and there are no additional assets that can be sold, then the player will go bankrupt and be removed from the game.

Trading with the other players is important for doing well in the game and one of the more interesting aspects of the game with regards to computational social science. When a player proposes a trade, the other players' `considerTrade` method is called. The other players can accept the trade as is, accept the trade with modification, or reject the trade proposal. If the trade is accepted as is, then the trade is processed. If the trade is accepted with modification, then the `considerCounter` method of the original proposer is called to see if the player agrees to the modification. The original proposer can then only accept or reject the counterproposal, but cannot counter the counterproposal.

The key to doing well in this competition is getting an entire set of properties so that you can build houses/hotels and get paid higher rents. Trading with the other players can make it much easier to get an entire set of properties, so spend some time designing your trading strategy.

Finally, make sure that your `MonopolyPlayer` implements the `toString` method so that a unique name is displayed in the simulation. Also, if you develop multiple player implementations and have them play against one another, then the `toString` can be helpful in determining how each of them are behaving and for debugging. To have your player implementation used, add your class name to the `monopoly_players.txt` file.

Competition Description

Each game of Monopoly will be played with four `MonopolyPlayers`. For the competition, there are two possible configurations:

1. Four different people wrote each of the four `MonopolyPlayers`.
2. Two people have two instances of their `MonopolyPlayer`.

Many games of Monopoly will be played to determine the winner of the competition.

What to Submit

Submit your ***well-documented*** Java class file that contains your implementation of a `MonopolyPlayer`. This file should be in a package named something like: `edu.gmu.<your username>.competition.monopoly`.

Also, submit a description of your player in a PDF file (approximately 1 to 2 pages).

Week 13 Assignment

Due: December 3, 2014 by 9:00am
(Competition submission due: December 10, 2014 by 9:00am)

Either submit an initial version of your `MonopolyPlayer` or write a paragraph about how you plan to approach the competition. The goal of this assignment is to get you started working on your competition code earlier rather than later.