CSC 243, Spring 2020, Assignment 3

Purpose: Inheritance and Exceptions

Due: 11:59pm, Monday, March 2, 2020

Get the assignment code

These instructions assume that your course git repository is set up. Change into your course repository directory and enter the following commands.

```
git fetch assignments
git checkout assignments/master -- assignment3
git commit -a
```

This will copy the `assignment3` directory into your working directory, start tracking the files in the `assignment3` directory, and commit those files to your local git repository.

Assignment Description

In this assignment, you need to make the following changes to the updated code base:

- Create a new exception named `DrawException` that is a subclass of the newly added `GameException`.
- Complete the implementation of the `AIPlayer` class. The AI player does not need to be competitive; it only needs to choose a valid move. The `getValidMoves` of the `Board` class might be useful for implementing this method.
- Change the `playMove` method to have type void instead of boolean. Instead of returning false when a move is invalid, the method must throw a `GameException`.
- Change the `checkWinner` method to throw a `DrawException` in the event that there is no winner and there are no valid moves remaining.
- Change the `Board.Position` constructor to throw a `GameException` if the argument is outside the range of valid moves.
- Make any required changes for callers of methods that throw exceptions.

Compiling and Running

To compile the programs, execute the command:
`make`

To run the main method, execute the command:
`make run`

To run the main method with an AI player, execute the command:
`make runAI`

To build the Javadoc documentation, execute the command:

```
make doc
```

To remove the compiled class files, execute the command:

```
make clean
```

Turning in the Assignment

To turn in the assignment execute the following git commands:

```
git commit -a
git push origin master
```

Note: the most recent commit before the due date will be considered your official submission.

Grading Criteria

- Concise, accurate documentation following the CSC Department documentation guidelines
- Correct Javadoc is used for class and method documentation
- Correct implementation of the specification

Note: If your code does not compile, then you will receive a failing grade for this assignment. If the submission includes material that was not covered in class and the material is not properly cited, then you will receive a failing grade for this assignment.