Project 4 - Dots and Boxes

Due Oct 28 by 12pm **Points** 80

Overview

We're moving back in time, from the 1990's and Tetris, to the 1890's and the game of **Dots and Boxes (https://en.wikipedia.org/wiki/Dots_and_Boxes)** . This still-popular pencil-and-paper game can be found everywhere, from paper placemats and kids' menus in restaurants, to dozens of variants available for mobile and online play.

Objectives

Once again you must develop an implementation of an interface. Your implementation will need to keep track of all aspects of this game. There is no user interface, as we are building the underlying game engine. As in other projects, you'll be provided a few additional classes to assist with your implementation. The focus here is on somewhat advanced uses of loops and branches, in conjunction with more complex collections. We'll also touch on a few basics of exception management, which we'll cover in more detail in a few weeks.

You have a lot of freedom here to design and develop your code in any way you see fit, so your guidance will be somewhat more limited than in the past projects.

Requirements

The interface is called `edu.vt.cs5044.DotsAndBoxes` and your implementation must be called `edu.vt.cs5044.DABGame`. You must of course create a test class as well, called `edu.vt.cs5044.DABGameTest` and achieve full code coverage. You are expected (and required) develop at least one additional class, in the same package with any reasonable name. Your DABGame class should delegate a substantial amount of logic and data to this other class, in order to reduce the overall complexity of your implementation.

Downloads

dab5044.jar - library file containing the compiled utility classes and interface
dab5044-api.jar - library file containing the Javadocs for the utility classes and interface

Getting Started

Start a new project in Eclipse, then import the library and API files above exactly as you did with the library and API files for the Tetris project. You don't need any JavaFX libraries for this project (there is no user interface at all) but you will need JUnit 4.

You'll use a lot of what you've learned in previous assignments to complete this assignment. Be sure to thoroughly review the API details for all the provided classes. **Everything you need to know is in the API.**

Since there is no user interface, nor any other way to run your implementation beyond your own test class, a test-driven development strategy will be important to follow.

Submit to Web-CAT

Submit your solution to **Web-CAT** **(https://web-cat.cs.vt.edu/)** via the Eclipse plug-in. Please feel free to improve and resubmit your code as many times as you like (before the deadline) in order to improve your score.

Important Notes:

- See the **project grading rubric** for full details on the grading criteria applied to this assignment.