PGCert IT: Programming for Industry

Challenges 04 - Noughts and Crosses Game

This challenge requires you to develop a simple game called Noughts and Crosses. Before starting the challenge, read through and gain an understanding of the requirements.

Notes:

- This challenge is worth **2%** of your final Programming for Industry grade. Each exercise is worth **1%** of the total grade.
- For this challenge, there is *no starter project* - you must create everything yourself. These instructions can help you create a new Java project within IntelliJ.
- After completing the challenge, you should have a functional Java program with some number of classes (depending on your own design).
- Zip your IntelliJ project for the challenge, including all the source files and submit the single Zip file before the due date.
- **IMPORTANT**: Read the instructions carefully before attempting each task.

Introduction

The Noughts and Crosses, or Tick-Tack-Toe, is designed for two players playing on a 3 x 3 square board. Each player in turn places their counter on the board. The first player to have three of their counters in a row wins. If no player has a row of three of their counters and there is no more move, then the game is a draw. More information about the game itself can be found here.

Exercise One: Design the Game

Design your classes and methods (i.e. create UML class and sequence diagrams) before implementing the game. The game could be played either via console or GUI. Don't have everything in one class, and try to promote code reuse as much as possible.

Discuss with the tutors or lecturers about your design before moving on to the next exercise.

Exercise Two: Implement the Game

Based on your design in exercise one, implement the GUI version of the Noughts and Crosses game.

There are different ways of visualising the game. You could use buttons to represent the counters on the board and allow players to interact with the board directly. You could also use a text field that allows players to enter their moves and a text area that represents the board.

Marking Guide

This challenge will be marked based on the correctness of your program and your programming style. Here are some questions to consider for the programming style:

- Is the code well-structured?
- Is the code self-explanatory?
- Can you understand the code easily?
- Are all variables properly defined with meaningful and clear code?
- Is there any commented out code?

Marks will be deducted if your code cannot be compiled or has bad programming style.

All required functions implemented correctly including proper exception handling; Good software design and programming style (appropriate classes, code reuse, good structure, variable names, comments, etc).	*5 marks*
All required functions implemented correctly; Adequate software design and programming style.	*3 marks*
All basic functions implemented correctly; Poor software design and programming style.	*1 marks*
Code cannot be compiled; Most functions not implemented;	*0 marks*