CS 3331 - Advanced Object-Oriented Programming

HOMEWORK 4: UI AND DESIGN

This assignment is to done in your designated team. You need to fill out the contribution form (see the course blackboard).

In this assignment, you are to extend your HW3 program to support several new features including the artificial intelligence for Player 2:

Provide a one player mode game where there are two difficulty levels: Easy & Intermediate

R1. Implement the "Easy" one-player mode as random move.

R2. Implement the "Intermediate" one-player mode based on the heuristic function discussed in class.

R3. Add feature where the user can select his/her disc color.

R4. Improve the user interface by providing a menu and a tool bar to perform at least the above features (R1, R2 & R3). Also start a "New Game".
For each menu item, provide an icon, a mnemonic and an accelerator.
For each tool bar button, use an icon and provide a tool tip.

R5. Your design should adhere to the MVC architectural pattern. Clearly identify the Model, View, and Controller classes.

R6. Document your code using JavaDoc. Generate the JavaDoc HTML.

1. Design your application and document your design by drawing a UML class diagram [Chapter 4 of 1]. You should focus on designing those classes that are modified (from your HW2 design) or newly introduced; highlight them in your diagram.

   - Your class diagram should show the main components (classes and interfaces) and their relationships.
   - Your model (business logic) classes should be clearly separated from the view/control (UI) classes with no dependencies [2].
   - For each class in your diagram, define key (public) operations to show its roles or responsibilities in your application.
   - For each association (aggregate and composite), include at least a label, multiplicities and navigation directions.
   - You should provide a short, textual description of each class appearing in your class diagram.

3. Code your design.

   For R1 & R2 above, use the Strategy design pattern to support a family of solving algorithms.

   For R4, use JMenuBar and JToolBar; refer to the following two online documents:

   - "How to use menus" available from https://docs.oracle.com/

javase/tutorial/uiswing/components/menu.html
- "How to use tool bars" available from https://docs.oracle.com/javase/tutorial/uiswing/components/toolbar.html

4. (Bonus points) Introduce additional interesting features, e.g.,
-Improve the Heuristic function provided in class and add an addiontal level "Advanced"
-Use sounds when a Disc is placed, or invalid move, and when there is a winner.

HINTS

Reuse your HW3 design and code as much as possible. For this, you may need to refactor
HW3 classes to open up their features or make them extensible.

TESTING

Your code should compile and run correctly under Java 8 or later
versions.

WHAT AND HOW TO TURN IN

You should submit a single PDF document of your UML diagrams along with
accompanying documents on the due date.
You should submit a single zip file that contains:

- design.pdf (UML class diagram along with descriptions)
- contribution-form.docx
- hw4.jar, a runnable jar containing bytecode and support files
(e.g., images and audio clips)
- src directory of source code files
- html generated from JavaDoc

You should submit your program through Blackboard.

The submission page will ask you to zip your program and upload a single zip file. Your zip file should include only a single directory named YourFirstNameLastName containing all your source code files and other support files needed to compile and run your program. DO NOT INCLUDE BYTECODE (.class) FILES. There is a limit on upload file size and the maximum file size is 2MB. You should turn in your programs by 11:59 pm on the due date.

GRADING

You will be graded on the quality of the design and how clear your code is. Excessively long code will be penalized: don't repeat code in multiple places. Your code should be reasonably documented and sensibly indented so it is easy to read and understand.

Be sure your name is in the comments in your code.

REFERENCES

[1] Martina Seidl, et al., UML@Classroom: An Introduction to Object-Oriented Modeling, Springer, 2015. Free ebook through UTEP library.

[2] Holger Gast, How to Use Objects, Addison-Wesley, 2016. Sections 9.1 and 9.2. Ebook available from UTEP library.