Tecnología de la Programación - Curso 2014/2015
Grado en Ingeniería Informática, Grado en Ingeniería de Computadores
Grado en Ingeniería del Software, Doble grado en Matemáticas-Informática
Examen de junio - Parte práctica - Duración: 2,5 horas.

Instrucciones

- Lee cuidadosamente el examen hasta el final. Algunas decisiones de diseño tomadas para implementar las primeras partes pueden afectar a la complejidad en la implementación de las últimas.
- En el examen deberás partir de la *práctica 5* realizada durante el curso. **En breve se indicará el procedimiento de acceso a dicho código.**
- El código entregado ***debe compilar***.
- En la corrección del examen se valorará el funcionamiento, la claridad del código, el uso conveniente de los medios proporcionados por la Programación Orientada a Objetos (herencia, polimorfismo...) y comentarios. Para la evaluación se tendrá en cuenta tanto lo pedido en el examen como el código del que se parte (implementación de la práctica 5). Romper la encapsulación de las clases (acceso a atributos privados y protegidos desde clases externas, utilización de atributos públicos, etc.) implica suspender el examen.
- Incluye, además del código, una descripción de los pasos y modificaciones que has realizado para implementar lo que se pide en el examen. Este informe de cambios puedes hacerlo en papel o adjuntarlo como un fichero de texto con el resto de los archivos de código fuente. Adjunta este fichero de texto `descripcion.txt` con el resto de los archivos de código fuente. Deja el fichero en "el raíz" del .zip entregado.
- La puntuación de cada apartado está calculada *sobre 10 puntos*, aunque supondrá el 50% de la nota final.

Consejos

Es preferible realizar el examen en el orden siguiente:

1. Implementar cada uno de los apartados. Tras la implementación de cada parte, comprobar que el código compila y funciona y guardar una copia del mismo. De este modo siempre tendremos algo que entregar que sabemos que compila y que funciona (aunque no esté completo).
2. Revisar y corregir los posibles errores o deficiencias que se arrastren de la práctica base, pues ese código también se evalúa.

Instrucciones de entrega

- Para entregar la solución al examen, crea un fichero zip. En él debes incluir todo el proyecto una vez limpiado de archivos intermedios, y con el fichero `alumnos.txt` con tu nombre completo.
- Nombra al fichero `NN_Apellido1Apellido2.zip`, donde `NN` indica el número de grupo (con dos dígitos).
- La estructura del fichero será la misma que la utilizada durante el curso.
- Si optas por escribir la descripción de los pasos y modificaciones realizadas durante el examen en un fichero de texto, inclúyelo en el raíz del archivo comprimido.
- Para entregar el examen, se utilizará el mecanismo de entregas disponible en el laboratorio. En particular, cerca del final del examen se habilitará la unidad `U:` en la que deberás dejar la solución al examen. Si deseas entregarlo antes, indícaselo al profesor para que habilite la unidad.
- Antes de abandonar el laboratorio debes pasar por el puesto del profesor para asegurarte de que lo que se ve en el puesto del profesor es lo que has entregado y firmar en la hoja de entregas.

Tecnología de la Programación - Curso 2014/2015
Grado en Ingeniería Informática, Doble grado en Matemáticas-Informática
Practical Exam - June 2015 - Duración: 2,5 horas.

Exam

We are going to extend the 5^{th} assignment to include the following functionality: (1) provide the user with information on the number of moves performed so far, by each player, during the game; (2) modify the Reversi game such that players have a limit on the number of moves that each can perform; and (3) allow players to yield their turn.

[0.5 points] Part A

Extend the model such that it tracks (i.e., stores in some way) the number of *valid* moves performed by each player. Recall that this information should be updated when executing any operation on the model.

[1.5 points] Part B

Modify the console and window views to show the number of moves performed by each player during the game.

[4 puntos] Parte C

Modify the application to allow setting a limit on the number of moves per player. Note that this number will be used in the next part, in this part we only need to store it. Implement the following:

- Add a command-line parameter `-n <num-movs>` or `--numMovs <num-movs>`, where `<num-movs>` is a non-negative number indicating the limit on the number of moves per player. When this number is 0, or the parameter is not provided, there is no limit.
- Modify the `PLAY` command, in the console mode, to allow setting a limit on the number of moves, e.g.,

  `"PLAY CO 10"` plays a Complica game with 10 moves limit per player.

  `"PLAY GR 7 7 15"` plays a Gravity game with 15 moves limit per player.

  When the provided limit is 0, there is no limit.
- Add to the window mode the possibility to set a limit on the number of moves when changing to a new game, e.g., using `JTextField`.
- Modify the window and console views so they show the limit on the number of moves. If there is no limit nothing should be shown (you can show an appropriate message as well in this case).

Note: Observe that you have to modify the classes of the model to store this limit in some way. This limit will be used in the next part.

[2 points] Part D

Implement the following rule in the Reversi game: "Every player can perform a maximum number of moves as indicated in the previous part. A player that reaches this limit will not be able to make moves anymore. The game terminates when both players reach their move limits or when the corresponding (current) termination conditions are satisfied."

Note: If you did not implement part C, adhoc your code to include a limit on the number of move for the Reversi game, as a constant for example.

[2 puntos] Parte E

Perform the following modifications so players can yield their turn (i.e., pass the turn to the other player, if possible, without making a move):

- Add to the model the possibility to yield the turn of the current player.
- Add a command `YIELD`, to the console mode, to yield the turn of the current player.
- Add a "Yield" button to the window mode that can be used to yield the turn of the current player. The button should be deactivated when it is not applicable, i.e., when the current player is automatic or the game is over.

Note: The Undo operation should not be affected by the yield operation, i.e., the undo stack should keep only the performed moves as before.