Victoria University of Wellington
School of Engineering and Computer Science

SWEN222: Software Design

Assignment 2

Due: Wednesday 30th August @ Midnight

1 Introduction

The goal of this assignment is to provide a graphical user interface for the game of Sword and Shield. You can reuse all the code you wish from the former assignment, but you are also free to modify it or to rewrite parts of it as it suits your needs.

You are required to strictly use the MVC pattern, as shown in the lectures.

In the description of the requirement we will use the following tags:

- Base: The feature is part of the bulk of the assignment, we suggest you start by implementing just the base features. Base features need to work well so that you can build the other features on top. By only implementing the Base features, you may get up to 70% of the marks.
- Hard: The feature is challenging and is contributing to get you a better mark. By completing Base and Hard features, you may get up to 90% of the marks.
- Completion: We suggest you to leave those features for last. They are related to rotation and reactions: If you have chosen 'Partial completion' for Assignment 1, you may not have those features in your current Sword and Shield implementation and you may be unable to do this part. By completing Base, Hard and Completion features, you may get up to 100% of the marks.
- Extra: If you chosen 'Partial completion' for Assignment 1, or otherwise you prefer doing 'Extra' **instead of** 'Completion', you may approach the (more boring) Extra features. By completing Base, Hard and Extra features, you may get up to 100% of the marks.

NOTE: you need to chose to do EITHER extra OR completion. You can not harvest points from both.

We suggest you to read carefully the whole assignment 2 handout before starting coding.

2 Menu phase

[Base] Menu content (4 Marks)
The game need to start with a menu, showing the title of the game. A 'Info' button will show the author name and some other information. A 'Begin new game' button will start a new game. A 'Quit' button will close the application.

3 General game set up

[Base] Display the game (4 Marks)
When the game is started by pressing the Begin button, the following elements are shown:

0 A JToolBar containing buttons ‘Undo’, ‘Pass’, ‘Surrender’

1 The board

2 The Green pieces that can still be created

3 The Yellow pieces that can still be created

4 The Green pieces in the cemetery

5 The Yellow pieces in the cemetery

[Hard] Resizeable panels (5 Marks)
The areas 1 to 5 should be independently resizeable, for example using using SplitPane (you may want to read https://docs.oracle.com/javase/tutorial/uiswing/components/splitpane.html). In any moment, all the content of all the 6 panels should be visible, so the drawings have to be resized.

4 Creation phase

[Base]Creation input (10 Marks)

a The user can click on a piece in the panel 2 (or 3).

b The content of the panel 2 (or 3) is replaced to show the selected piece in the 4 possible orientations.

c The user click on the selected orientation.

d The content of the panel 2 (or 3) shows the pieces that can still be created.

The selected piece (in the selected orientation) is now displayed in the creation grid.

[Hard]Creation animation (3 Marks)
Make a smooth graphical transition between a-b and c-d.
[Hard]Creation animation (3 Marks)
Make so that the selected piece smoothly flies into position instead of just appearing there. This is **very hard** since it requires to move a piece ”on top” of different panels/splitpane. We suggest you leave this feature last.
[Base]Skipping creation (1 Marks)
The user may click on the ‘pass’ button to go directly to the movement phase. Note that this is the only possible option when the creation grid is occupied. If the creation grid is occupied, and the user tries to click on a piece in the panel 2 (or 3), nothing should happen.
[Extra] Warning the user (4 Marks)
If the creation grid is occupied, and the user tries to click on a piece in the panel 2 (or 3), an ugly beeping sound should advise the user that is attempting an invalid move. At the second attempt to click on a piece, the ugly beeping sound should be repeated, longer. Then the ‘pass’ button should become golden and shiny to attract the attention. If the user tries a 3rd time, a ‘JDialog’ should be used to explicitly explain to the user what they are doing wrong.

5 Moving phase

[Base]Skipping Moving (1 Marks)
Players can press the ‘pass’ button to stop their turn.
[Base]Moving input (12 Marks)

a The user will click on a piece on the board.

b Some graphical effect will show clearly that the piece is selected.

Already moved pieces can not be selected. Selected pieces can be moved in two ways:

- Press an arrow key to move the piece
- Click on the edge of the pieces in the direction of the desired movement. You should consider the area inside of the piece, but near the ’edge’, to be part of the edge. Make it so that it is reasonably easy to click on the ’edge’, do not make it 1 single pixel tick.

c The board will update accordingly.

Some graphical effect show what pieces have already moved.

Both ways (clicking and arrows) are required.

[Hard] Moving animation (5 Marks)
When the piece moves, perform an animation showing the movement in a smooth way. Notice that multiple pieces may have to be animated at the same time.
[Hard] Disappearing animation (2 Marks)
When a piece fall off the board (or dies in any other way if you have implemented reactions), animate the demise of the piece in some clear way. For example you may make the piece shrink to a minuscule size, slowly become transparent or start rotating very fast. At the end of the animation, the piece will have to disappear.
[Hard] Sound effects (2 Marks)
Add sound effects to the ’Disappearing animation’.
[Completion] Rotation input (5 Marks)
After ‘b’ (the selection of the piece) the user can chose to rotate the piece instead. Rotation is selected by clicking in the centre of the piece (while movement happens by clicking the sides).

- The piece will become larger and the rest of the board will look more grey.
- The user will click on the piece, and at every click the piece will rotate 90 degrees.
- When the user has selected their desired rotation they will click out of the piece to commit the change.
- The board will update to show the new configuration, and the game will continue.

[Completion] Reactions input (5 Marks)
After any movement or rotation, reactions may happen.

- If there are reactions, the reactions are going to be marked on the board with some clearly visible graphic effect, like a red sign around/under the reacting pieces.
- The user will click on the *border between* two pieces that are reacting.
- The selected reaction will happen, and the game will continue.

Display some nice graphical effect to show the reaction happening.

6 Terminating the game

[Base] End game (1 Mark)
If in any moment a player clicks on the 'Surrender' button, or the winning condition reaction happens, a JDialog will notify of the victory; Then the game will go back to the the Menu.
[Extra] Display Winner and Loser (4 Marks)
The winning and the losing

- The two faces will become larger and the rest of the board will look more grey.
- After half a second, the area around the winning face will become less greyed out, while the area around the losing face will become more greyed out. You can look at RadialGradientPaint and LayerUI, and to http://docs.oracle.com/javase/tutorial/uiswing/misc/jlayer.html

[Extra] End game statistics (2 Marks)
When a game is terminated, you should provide a JDialog with informations about the game: How many moves happened, how long the game took (in minutes and seconds), and how many pieces died. Finally, report how many times the Undo feature has been used.

7 Other marks

In addition of the marks for correctly completing the described features, marks will be assigned for

- **[Base] Swing Components (5 Marks)** Marks will be awarded for programs which make appropriate use of Swing components.
- **[Base] Model-View-Controller (5 Marks)** Marks will be awarded for programs which make appropriate use of the *Model-View-Controller* pattern, as discussed in lectures. In order to be sure to follow the MVC, A good starting point for your code could be the code for WheatRun (lab3). You can adapt it to turn it into your assignment 2.
- **[Base] Code Style (5 Marks)** Overall code style which includes: good use of naming for methods, fields, classes and variables, good division of work into methods, so as to avoid long and complex chunks of code.

8 Submission

Your program code should be submitted electronically via the *online submission system*, linked from the course homepage. Your submitted code should be packaged into a jar file, including the source code (see the export-to-jar tutorial linked from the course homepage). Your program should include appropriate Javadoc comments, and provide JUnit tests where appropriate.

In addition to the source code, various pieces of design documentation are required. These should be submitted as PDF files; other formats will not be accepted. The required documents are:

1. **[Base] A 300 words design report (10 Marks)** giving an overview of the design for your graphical user interface. This should include brief discussions on how the GUI is organised and, in particular, how the main display is drawn and how the animations are handled. While you can mention the logic of the game, keep the focus on the code implementing the GUI and the user interaction.

2. [**Base**] **Good code highlight (12 Marks)** A 200 words description commenting a 2-50 lines fragment of source-code you wrote (as part of your final submission for this assignment) that you believe to be brilliant, well designed or otherwise worthy of praise. This code need to be related with the GUI, not with the game logic.

3. **Complete project code (78 Marks total)** Submit an exported copy of your eclipse project. This will contain

   - A *readme.txt* file.
   - The complete source code.
   - Any other dependency needed to run your code.
   - Your test code.

   If we fail to easily run the code, you may lose a lot of marks.