Report on connecting GUI to BlackJack model

Joakim Bergqvist

October 27, 2017

Introduction

In this short report there will be explaining on the task of connecting a Graphical User Interface(GUI) to a already existing Black Jack game. The decision to leave the model as it is was made early, even though some changes would have maybe felt more logical for a GUI, i.e. when a hand is updated all cards is sent as an Iterable. But as the GUI needs to keep all card it would have felt more logical to only send the new cards. The initial though was to try to save most of the controller as well witch unfortunately wasn't fulfilled.

Designing the GUI

Firstly a very simple GUI design was chosen,the view was split up in different sections (Panes) and more specifically mostly a mix of HBoxes and VBoxes. After the code was added for this, the Player/Dealer score was added since that was concidered to be the easiest task. This would update each time the observer recieved that the model was updated. In an attempt to launch the program it didn't even want to open the stage. So some research had to be done, and after the play game loop was started inside a Runnable the application would start and the score would update.

Adding the cards

As mentioned in the introduction the cards received had to be filtered to know which cards that was new, and only those were sent to the GUI to add the visual presentation of the card. After some fiddling with the location of the assets and linking the naming the program would run and the cards would appear on the screen.

Timing the cards

This was where the real problems started, and this was not solved in a completely satisfactoriness. A lot of time was spent trying different approaches. Trying to sleep the thread in a similar way as in the console base solution didn't work. Adding transition or timeline did animate the cards, but the animation for each card would start almost instantly. This is unfortunately the final solution that gave the best looking solution for me without changing the model side. I tried some different solutions without success. The one that was though would solve it was to put each timeline to a queue and at the end of the timeline trigger a new play timeline. But still all the cards at startup arrived with the same timing as before. Some reading about how to synchronize threads was done, but no soulutions that helped solved the problem was found. To make the card flip one change in the model was made, the observer got one more method to notify the subscripers, so now the controller knew if the call was a new card recieved or a showed hand call.

Conclusion

This task was much more troublesome than initially though. The reason for this was mainly that the Model is very well encapsulated and controller has very little input and output which makes it harder to make a pause between each card is dealt when not working on the same threads. Even if not having completely control of the timing I think I could make the animations timing somewhat satisfying. But I definitely will need to spend some more time mastering threads to know in detail what actually is going on.