Final Design Outline

Jianan Chen
Jiaming Zhang

Spring 2016

Genearal Idea

The chess project we implemented used the MVC modelling. The entire project consist of three major components. The controller handles the user input and responsible for error handling of the wrong user commands as well as game instructions. The model stores the internal status of the game. The view is responsible for print the chess board.

Class Description

- Controller:
- View:
  Observer pattern was implemented here.
  - TextDisplay:
    - print the chessboard by ascii art
  - GraphicDisplay:
    - visualize the chessboard status graphically
  - UnicodeDisplay:
    - print the chessboard by unicode character, which shows the actual shapes of the pieces
- Model:
  - Board:
    - contains the whole board
    - support/restrict to make each movement of the chesspiece
  - ChessPiece:
    - include six subclasses in total
    - represent all of the chesspiece appeared on the chessboard
    - rule the move range and the attack range of each chesspiece

Timetable

Step one: creation of the big structure of the project (3 hours):

o The connection of all files, follows the observer pattern

o Definition of all classes, included base classes and the subclasses, filled in by their fields and the methods' initiation:

o Leave the place of the implementation with comments

Step two: class implementation (4 hours):

o The chess_piece class and its subclasses

o The observer class

Step three: project main structure implementation (8 hours):

- Main class
- Makefile
- Controller
- Level one implementation

Step four: Display-component implementation (6 hour):

- DsiplayByText
- DisplayByWindows

Step five: Testing and debugging (10 hours).

Step six: enhancement of the program(3 hours):

- Level two, three and four implementation
- Clean up the dead code
- Partially code revisions

Work Distribution

j548chen: controller, display, and Ai
j585zhan: chesspiece board

Cohesion and coupling

The program achieve high cohesion and low couping by using mvc modelling. All the function in the Model have a common purpose. The controller handles all user inputs, including receving and parsing inputs. This is a good example of high cohesion, since the fucntionality of control is simple and consistant. Same for the model and view parts. And since most functions are handle by only one part of the program, low compling is also achieved.

Questions

Question: Chess programs usually come with a book of standard opening move sequences, which list accepted opening moves and responses to opponents' moves, for the first dozen or so moves of the game. Although you are not required to support this, discuss how you would implement a book of standard openings if required.

We will refer to the classical opening moves online, and save them in a text file. When the game start, check if there is corresponding moves in the file until there is no reference to use. If any of the players stop to follow the moves sequences, the AI will use their normal way of deciding how to move.

Question: How would you implement a feature that would allow a player to undo his/her last move? What about an unlimited number of undoes?

To allow the player to undo his/her last move, we choose to use a stack of grid (theChessBoard), which shallow copy all of the smart pointer into it. It allows undo unlimited times. Each time when user undo one step, the last chessboard is popped out the stack.

Then the current chessboard shallow copy the old chessboard saved in the history. Because of the changing of position of each chesspiece, the chessboard will corrrect all of their position by the position of the chesspiece pointer in the stack.
In this question, we decided to change slightly regard our previous solving strategy. The reason of that is we found this way saves more space for the program, because most of the pointers in the history are pointing to the same thing(shared_ptr).

Question: Variations on chess abound. For example, four-handed chess is a variant that is played by four players (search for it!). Outline the changes that would be necessary to make your program into a four-handed chess game.

In the controller, simply add two more players to the game. Also, we need to change the Board class and View class to fit the new configuration of the board. In addition, we need to make changes to AI class to accommodate the extra two players.

1. What lessons did this project teach you about developing software in teams? If you worked alone, what lessons did you learn about writing large programs?

- always do the documentation to make others easy to read the code
- coding after thinking throughtly; otherwise, it is time-wasting to change the idea midway
- maintain communication with each other during the programming, to avoid misunderstanding
- start earlier to meet deadline

2. What would you have done differently if you had the chance to start over?

Firstly, we would start as early as we can, so we wouldn't be rushing at the night ahead the deadline. Secondly, we will do the AI part and the Undo part first since at the end of the project we found these two features are extremely helpful in many circumstances. Lastly, we would debug new functionalityies thoroughly after implementing, instead of trying to debug when most of the implementations are finished.