

## Plan of Attack

### Breakdown of Project(two hours):

- 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
  - o Distribution: half for each person
- Step two: class implementation (4 hours):
  - o The chess\_piece class and its subclasses
  - o The observer class
  - o Distribution: half for each person
- Step three: project main structure implementation (4 hours):
  - o Main class
  - o Makefile
  - o Controller
  - o Level one implementation
  - o Distribution:
    - J585zhan: Controller + Makefile
    - J548chen: Main class + Level one implementation
- Step four: Display-component implementation (3 hour):
  - o DsisplayByText
  - o DisplayByWindows
  - o Distribution: One for each person
- Step five: Testing and debugging (6 hours).
  - o Distribution: Together
- Step six: enhancement of the program(3 hours):
  - o Level two, three and four implementation
  - o Clean up the dead code
  - o Partially code revisions
  - o Distribution: Together

## 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 contain a pointer to the deep copy the gaming state(Board class) at each step, and save it in the controller. For the unlimited number of undoes, we will use the easiest way to achieve it, which is save all of the steps in the stack. When we need to undo, use the last element in the stack and pop it.

**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.