Plan:

July 17th: finish a5-uml, pp9kplan.pdf, deadlinex.txt
July 18th: finish GameNotification, Controller, View, TextDisplay, GraphicDisplay, Xwindow, some parts of Game
July 18th: finish rest of Game, Chessmen(all subclass), Player(Human, Computer levev1)
July 19th: test game initialization part(setup the whole board, setup some particular position, setup from a saved file) test if the display board works well, test player's moving part(capture correctly, check, checkmate, pawn capture en passant, pawn promotion, castling)
July 20th: create computer level 2-4
July 21st: test computer level 2-4
July 22nd: add some check condition to check invalid input, test it
July 23rd: update a5-uml-final, compare with a5-uml, overview of all aspects of our project, answer questions for due day 2, create final design document, create Makefile.
July 24th: select a demo time slot
July 25th-27th: final check and test the program
July 28th: submit all required files

Questions:

Q1

This is most likely a load-game method, but the load-game method is used to update the chessboard, the standard opening move sequence is like a guide that both player could follow to make the next move.

So, the book is like a file that contains some starting valid move sequence. Instead making a move by using cin(for human player) or using random/AI(for computer player), both player could read the next moving sequence from the book file.

Since these starting valid move sequence are just for the game that both players have not moved yet. So one of play already moved, reading from this file will not be allowed.

Also, during the reading process form the book file, once one player decide to move by their own will(human player by cin, computer player by random/AI mechanism), the reading process form the book file should be terminated and cannot be restart until the new game starts.

Q2

Create a vector in Game Class,if game is notified that there is a valid move occurs,then store all the information vector.(cannot undo,if game is over after you make move)

For example,

i,If white makes a rook e1 eats a rook e3,then store 000115153

ii,If black make a castling and rook at a8,then store 111000000

1.Fist bit is indicating which player is making move

0 is white

1 is black

2.Second bit is indicating whether there is a castling

0 no

1 yes

3.Third bit is indicating the position of rook if there is a castling

i. If there is no castling we ignore this bit

ii. If there is castling we know king and one of rook haven't move yet therefore we know the position of king. Since we know which player is make move. The only since I have to know is which rook is used.

1 is left

2 is right

Once we know which one is used in castling, will know what's the final position of the king and the rook.

4.Fourth bit is indicating if the player is capturing the other.

1 is capturing

0 is no capturing

5.Fifth bit is indicating which one the player are capturing.

i. If no capturing it will be 0

ii. 1 is K 3 is B 5 is N 2 is Q 4 is R 6 is P

6 Sixth~Seven bits are indicating the position before move

11 is a1

21 is b1

…

since the board is 8x8 two bits are enough to indicating the position

7 Eighth~Ninth bits are indicating the position after move like 6

Therefore Ten bits contain all information about what is happened after player makes move. If undo occurs, we can take out the last input from the vector, and decipher the information bit by bit and reverse the move's order, and pop_bach the last element.
For the unlimited undo, just keep reading the last element, decipher the information, refer the move's order, and pop_bach the last element. If there is no element in vector, then we print error message.

Q3

Two more player classes should be added in our game class.

The group type should be expand. Suppose for two-player game, group type are represent by integers, such that 1 represent group-1(black chessmen), and 2 represent group-2(white chessmen), group = {1,2}. Now we add group-3 and group-4 to indicate the player-3 and player-4, such that group = {1,2,3,4}.

The capturing mechanism should be changed. For example, if group-1 and group-2 are alliance, group-3 and group-4 are alliance, then the chessmen for group-1 and group-2 can only capture the chessmen which belong group-3 and group-4. So, in each chessmen, besides the group field, we should add another field called alliance to represent the alliance group number.

The game board initialization should be changed(the chessboard size, the chessmen position)

Suppose group-1 and group-2 are alliance, king of group-1 is captured, then the player class for group-1 should notify the rest of chessmen to change their group type to 2, this means the rest of player-1's chessmen now are belong to player-2, and can be control by player-2.

The winning mechanism should be changed. If only one king is checkmated for one group of alliance, the game is draw.

Q4

Set two coordinates x=0 y=0, starts at the most left of the top corner, and I enter the setup mode, then keep reading a char at a time until I have read a W or B. Since I already have the setup mode, each time I only have to update the coordinates so that it matches the coordinates on the board,and set color to whatever I read at last.