# Plan of attack of PawnPusher9000

We have separated the project into two major parts, which are the AI system and Board related. Basically, the Model-View-Controller (MVC) design pattern is applied to this program. We will begin with the Model part (class Board and Pieces) and textdisplay part .Next, we will implement the controller part. After that, AI logic and the graphicdisplay model will be finished. Finally, we will add on DLCs after the debugging process if possible.

1. Implement Board, Pieces and textdisplay part.
   Deadline: 2015-7-18
2. Implement Controller.
   Deadline: 2015-7-20
3. Add on ability of checking invalid move. Allow multiplayer mode.
   Deadline: 2015-7-22
4. Create AI system. Implement 3 difficulties of single player mode.
   Deadline: 2015-7-24
5. Allow singleplayer mode and movie mode.
   Deadline: 2015-7-25
6. Implement Xwindow display.
   Deadline: 2015-7-26
7. Debug, DLC (Different mode, Higher AI) and documentation.
   Deadline: 2015-7-28

# Questions and Answers

Question 1

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.

Answer 1

First, we can find some standard opening moves sequence online and store them in standard notations. We store the moves of our board also use the same notation, so that the moves can be matched with the standard openings we store. When the opponent make a move, we search through all the openings in our library and match one, then perform the next move according to the standard openings we store.

Question 2

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

Answer 2

Creating a vector field in Board class is preferred. The vector stores every move of each player in the form of "string". For instance, "e2 f3 P" means a move of a piece from e2 to f3 capturing another piece P(Pawn). In each player's round, push_back() method is needed to push the string containing the necessary information of the move into the vector. To undo the moves of a player, we need to call the pop_back() method of the vector class which contains the information about the specific moves. If the string being popped ends with a uppercase letter, then a piece has been captured in the last move. Otherwise, no piece has been captured in the move. For example, the string being popped is "e2 f3 P", we move the piece on f3 back to e2 and put an opponent piece Pawn at f3. Since vector will automatically allocate and manage memory so the vector could store infinite move; therefore, enabling unlimited number of undos. :)

Question 3

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.

Answer 3

1. Add necessary cells to the board (four 3*8 areas on each side).
2. Add corresponding fields of pieces indicating the player it belongs to.
3. If team-mode is enabled, check algorithm should be changed to prevent friendly-check.
4. The Pawn promotion should also be applied when a Pawn reaches the King's row to the left, right or directly across.
5. :)