Suggestions of Extensions

1. Adding an option to save the game would be a good extension. To give the choice of a save option I would allow the user the enter ‘s’ at anytime. At that point when the ‘s’ is entered then I would break from main while loop and call a save function which takes no parameters. The function will prompt the user for a name that he or she would like to save it as. The function would open a stream in the same folder with the name and extension “.txt “. Then it would take the board\_array, and current player and write it to the output stream. The advantage of having player and board as a global variable is that there is no confusion between passing arguments by value and possibly starting one move before the one that was intended to be saved.
2. With a save option there must also be an option to open a previously saved game. That would be done with the user entering the name of the whole file in the command line. The main would have to check if it is a valid file and that it is starting at the beginning of the file. After that process, it will proceed to open a stream from the input file and begin reading the file in the same fashion that it was saved. Finally it will change the global variables to the input that was read and the game will start where it was left off at.
3. Another possible extension is to allow the player to undo a move. The general idea is allow the player to go back one move if he or she accidentally put in an unintended move. The process would first show the player the consequences of the move, then ask if the user would like to proceed and a simply ‘y’ or ‘n’ would suffice. I would imagine the process involving the malloc() and free() function so that excess memory would not be taking up. The process would store the move before the change in memory and if the user decided that he or she would like to go back then current array would be overwritten with that stored in memory.
4. Another extension could be the option to play against some form of AI. The player\_can\_move function could be a skeleton for the AI algorithm. For example while sweeping the board the program could take note of the coordinate that results in the most flips of the opposite player’s piece. Then when it comes for the AI to play, the coordinate would be the input for the AI and the turn would switch to the human player. If there was an option to implement a level of difficulty for example one to five, that number will determine the number of turns that the AI player can see ahead. This would probably require a lot of memory since it is predicting every combination of moves that will be made and not made. Then it has to go through and decide which path will convert the most pieces.
5. I could also go through and optimize for performance and clean up memory where it can be cleaned.
6. I think that another improvement would be to convert this to a graphical interface and allow the players to click on the spot that they would like to place their piece on. Most of the source code would still be the same, I think that it would just be a matter of converting the mouse clicks to an input that can be interpreted by the algorithm.