Criterion E – Evaluation

Meeting of the criteria for Success:

1. Interface is representation of the board showing each piece in its intended position – *this criterion is met*
2. All game data and moves are saved and can be undone – *game data and moves are saved, however one must enter a specific string on the opponent's turn to undo the last move and moves cannot be redone. One cannot undo moves after a winner is declared.*
3. Players set up their pieces on their respective side of the board, according to the rules. – *criterion is met*
4. Player can easily move each piece by inputting board coordinates – *this criteria is partially met as players can move pieces by inputting board coordinates. However data must be entered in a specific format on console which can make it harder for the user.*
5. There is an accurate determination of which piece(s) gets eliminated when pieces collide – *works*
6. The rules are correctly implemented. – *correctly implemented*
7. Winner is announced when the opponent flag is captured or bombed. – *criterion met*

Recommendations for Future Improvements:

This game works well for players who want to experiment with new strategies, as the undo function allows players to try different combinations of moves to test different outcomes. However as my client suggested, this game can be tedious to play, as new players not only need to figure out the rules but also put a lot of time in finding out the coordinates of piece they need to move and then the coordinates it needs to be moved to. To make interactions easier between the player and the program, further GUI could be used to allow the player to click the piece that needs to be moved and its destination. I could user event listeners to listen for clicks in certain areas of the board when the console prompts for a form of input. Since I already have the x,y coordinates, width and height for each JLabel, I can use those values for the implementation of a clicking system as opposed to console input.

Furthermore, I can also implement a game saving system. Where a game is saved and can be recalled similar to that of the undo function. To do this, I can make the stack recordOfTurns a 2D array, so I can use the first set of indexes to store the game number and the second to store the Boards within the game. A main menu will be implemented to ask the user for the game to be recalled and the game will be recalled to the point where the players last left off.

My choice of Java was a good one, since it is the language I am most comfortable with. However, GUI can be difficult with Java and I have definitely struggled displaying certain pieces that have been manipulated. Sometimes the piece of the state would change but there would be no change in display.

Word Count: 489