

1. Copy and paste relevant areas of homework 5 to the files of homework 6
2. Define a structure containing the enums of color and ranks and a boolean
3. Change board from an array of pieces to an array of the structure
4. Replace all areas that use board[i][j] or stuff like that with board[i][j].rank
5. Create function for updating the board
6. Create loop for actual gameplay
  - a. Check for valid coordinates for selected piece and piece movement
  - b. Check for game end
7. Turn board into a dynamic 2d array
  - a. Change all the function definitions and stuff to accommodate the dynamic array

### Hand-calculations (showing the board works)

```

or the GENERAL (2). When pieces have equal rank, then both pieces are
removed.
3. The FLAG and BOMBS are not moveable while all of the other pieces may
move one square at a time forward, backward, or sideward, but not di-
agonally to an open square.
4. A player must either move or strike on his/her turn.
5. The game ends when a player strikes his/her opponent's flag.
-----
  1 2 3 4 5
+-----+
A | X X X X X |
B | X X X X X |
C |           |
D | B 8 7 S 6 |
E | B 4 F B 1 |
+-----+
RED MOVE: Enter current coordinates of piece (e.g., D2, or QQ to quit): D2
RED MOVE: Enter new coordinates of piece at D2: C2
RED 8 moves from D2 to C2.
  1 2 3 4 5
+-----+
A | X X X X X |
B | X X X X X |
C | 8       |
D | B 7 S 6 |
E | B 4 F B 1 |
+-----+
RED MOVE: Enter current coordinates of piece (e.g., D2, or QQ to quit): C2
RED MOVE: Enter new coordinates of piece at C2: B2
RED 8 at C2 defuse BLUE B at B2.
  1 2 3 4 5
+-----+
A | X X X X X |
B | X X X X X |
C |           |
D | B 7 S 6 |
E | B 4 F B 1 |
+-----+
RED MOVE: Enter current coordinates of piece (e.g., D2, or QQ to quit): B2
RED MOVE: Enter new coordinates of piece at B2: A2
RED 8 at B2 captured by BLUE 3 at A2.
  1 2 3 4 5
+-----+
A | X X X X X |
B | X X X X X |
C |           |
D | B 7 S 6 |
E | B 4 F B 1 |
+-----+
RED MOVE: Enter current coordinates of piece (e.g., D2, or QQ to quit):

```