```
 1 import random
 2
 3 # Board
 4 # Worked with Jyle
 5 row1 = ["1 ", "2 ", "3 ", "4 ", "5 "]
 6 row2 = ["6 ", "7 ", "8 ", "9 ", "10"]
 7 row3 = ["11", "12", "13", "14", "15"]
 8 row4 = ["16", "17", "18", "19", "20"]
 9 row5 = ["21", "22", "23", "24", "25"]
10
11 board = [
12     row1,
13     row2,
14     row3,
15     row4,
16     row5
17 ]
18
19 # Bomb board that is going to be altered to
   determine where the bombs should be randomized.
20 rowB1 = ["1 ", "2 ", "3 ", "4 ", "5 "]
21 rowB2 = ["6 ", "7 ", "8 ", "9 ", "10"]
22 rowB3 = ["11", "12", "13", "14", "15"]
23 rowB4 = ["16", "17", "18", "19", "20"]
24 rowB5 = ["21", "22", "23", "24", "25"]
25
26 bomb_Board = [
27     rowB1,
28     rowB2,
29     rowB3,
30     rowB4,
31     rowB5
32 ]
33 # Prints the board after every move
34 def print_rows(game_board):
35     for i in range(len(game_board)):
36         print("  ".join(game_board[i]))
37
38 def random_bombs():
39     for value in range(len(bomb_Board)):
40         random_row = random.randint(0, len(
```

```
40 bomb_Board) - 1)
41         random_element = random.randint(0, len(
   bomb_Board) - 1)
42         bomb_Board[random_row][random_element] = "
   Bb"
43
44 bombHit = False
45 value_in_board = False
46
47 # Defines the coordinates the user can select (1-25
   ), if they hit a bomb ton the board.hey lose (5
   bombs), user is
48 # prompted to enter a new coordinate if they input
   one that is not listed
49 # Worked with Jyle
50 def flag_place(play):
51     global bombHit, value_in_board
52     if play == "Yes":
53         choices = 5
54         while not bombHit:
55             while choices > 0:
56                 if bombHit == True:
57                     break
58                 print()
59                 print("You have " + str(choices) +
   " choices left!")
60                 row_coordinate = input("Please
   choose a spot on the board. ")
61                 print()
62                 row_coordinate_single =
   row_coordinate + " "
63                 # Worked with Jyle
64                 for elements in bomb_Board:
65                     if row_coordinate_single in
   elements:
66                         board[bomb_Board.index(
   elements)][elements.index(row_coordinate_single
   )] = "X "
67                         value_in_board = True
68                     elif row_coordinate in elements
   :
```

```
 69                          board[bomb_Board.index(
    elements)][elements.index(row_coordinate)] = "X "
 70                          value_in_board = True
 71                      else:
 72                          continue
 73                  if not value_in_board:
 74                      bombHit = True
 75                      continue
 76                  else:
 77                      value_in_board = False
 78                      print_rows(board)
 79
 80                  choices -= 1
 81              # Defines a win function if the user
    successfully avoids all bombs by the end of their
    moves
 82              if choices == 0:
 83                  print("No more choices so you win
    !")
 84                  break
 85
 86          if bombHit:
 87              print("You hit a bomb, you lose")
 88              # print_rows(bomb_Board)
 89          else:
 90              print("You win!")
 91      elif play == "No":
 92          print("Rerun Program to try again")
 93
 94 # Worked with Jyle
 95 if __name__ == "__main__":
 96      print_rows(board)
 97      print()
 98      random_bombs()
 99      play_game = str(input("Do you want to play the
     game? Yes or No: "))
100      # Calls place_flag function
101      flag_place(play_game)
102
```