

White Box Test Cases

-the following tests assumes the following initial conditions:

- standard board

- turn 1

-relevant variables will be printed onto Python console

Case 1:

Input: 3 moves made by player (blue player vs computer)

Expected Output: 6 turns will be tracked by turnCounter

Actual Output: result of 'print turnCounter' placed after every counter adding:

```
>>>
playing BPvC
1
2
3
4
5
6
```

Test Passed

Case 2:

Input: no action by user after game starts

Expected Output: blue side should have 7 possible moves, so there should be 7 elements in possible_moves array

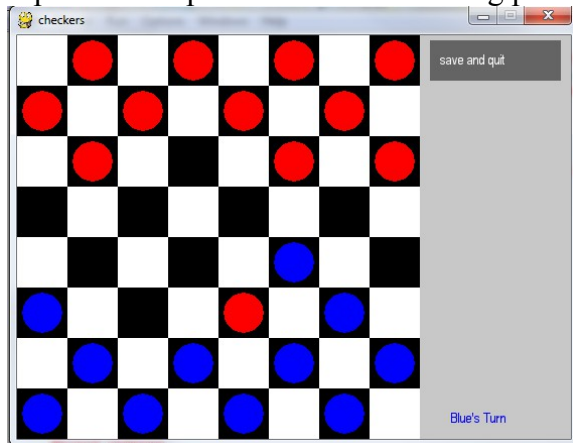
Actual Output: result of 'print len(possible_moves)' during the turn:

```
7
7
7
7
7
7
7
7
```

Test Passed (there's constant printing because print statement is within gameloop)

Case 3:

Input: moved pieces into the following position:



Expected Output: blue side only has the killing move available (the blue piece jumping to the top left of the red piece), so there should be only 1 element in possible_moves array

Actual Output: result of 'print len(possible_moves)' during the turn:

```
1
1
1
1
1
1
1
1
1
1
1
1
```

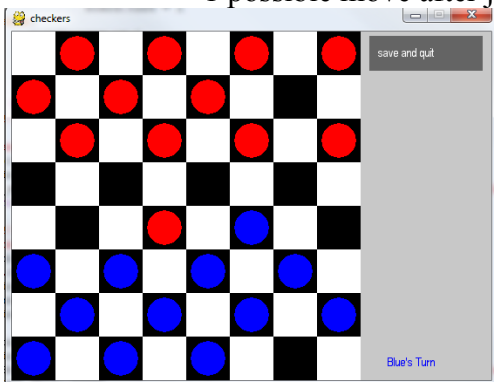
Test Passed (there's constant printing because print statement is within gameloop)

Case 4:

Input: attempt will be made to jump twice, there should be:

-2 possible moves before jump

-1 possible move after jump, still blue turn



Expected Output: with `len(possible_moves)`, the following will be printed:

- blue turn part 1: 2

- blue turn part 2: 1

Actual Output:

Part 1:

```
2
2
2
2
2
2
2
2
```

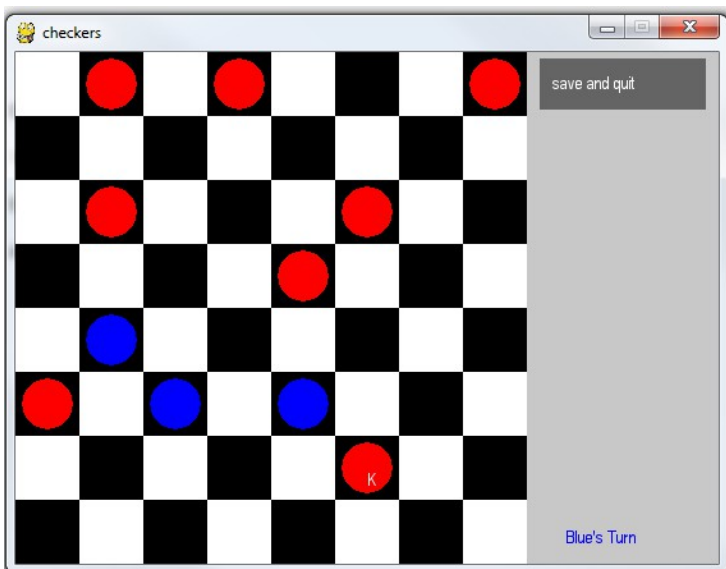
Part 2:

```
1
1
1
1
1
1
1
1
```

Test Passed (there's constant printing because print statement is within gameloop)

Case 5:

Input: moved pieces into the following position:



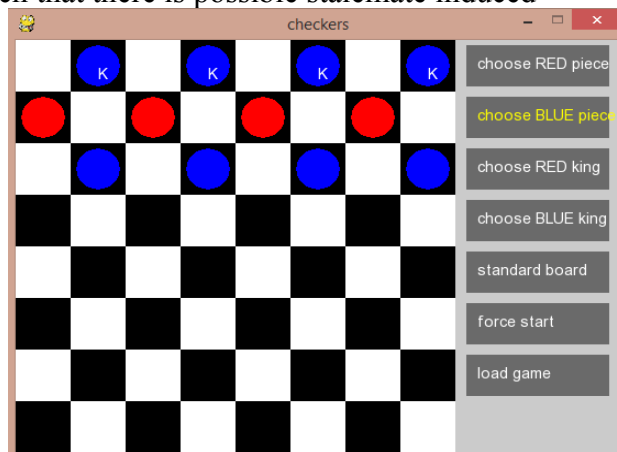
Expected Output: blue has 5 possible turns ($2 + 1 + 2$), and no kill moves are possible
Actual Output:

5
5
5
5
5
5
5
5
5
5

Test Passed (there's constant printing because print statement is within gameloop)

Black Box Test Cases

Input: Setup the board such that there is possible stalemate induced



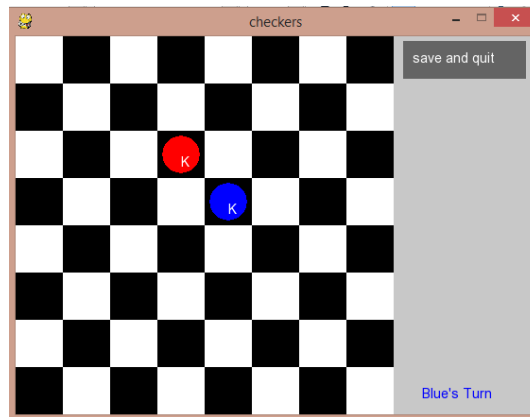
Expected Output: A first turn cannot be played

Output **Pass/Fail**: A first turn cannot be played screen appears and the program terminates

Input:testing forced jump with king pieces,specifically blue piece

Expected Output: Blue King cannot do anything but jump the red piece

Output **Pass**/Fail:



After jumping the game terminates with blue wins, and no moves besides the jump are legal.

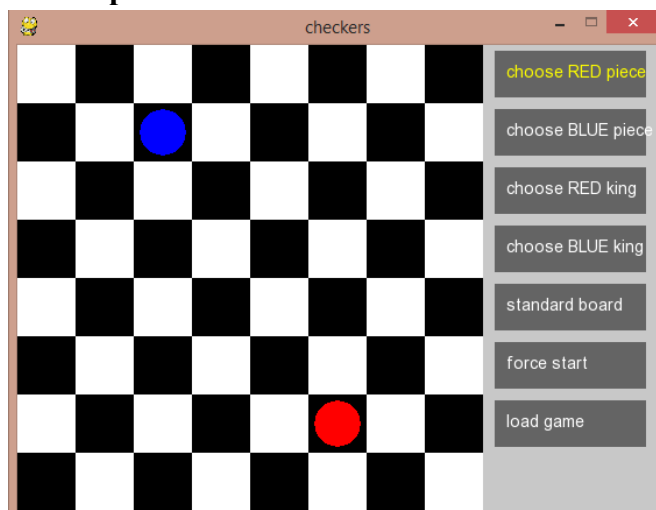
Recall that all the AI choices are Random except for the forced jumps

Input: Setup the board and test the AI and rerun the same scenario multiple time and see if the random results are produced. Checking if the computer is random. Initial setup:

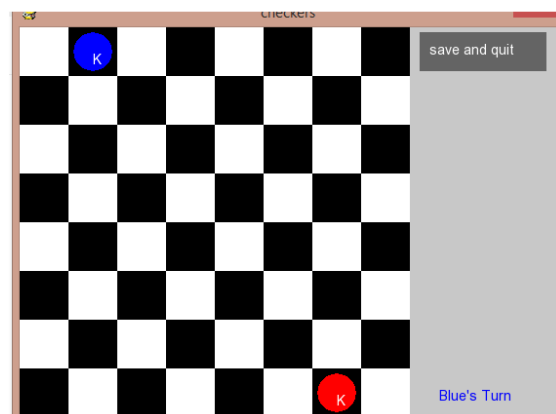
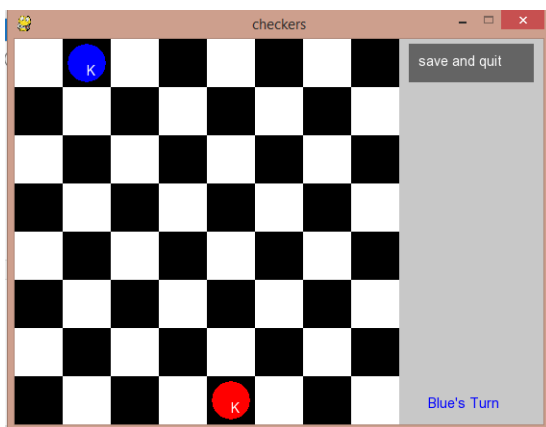
Expected Output: It should be random

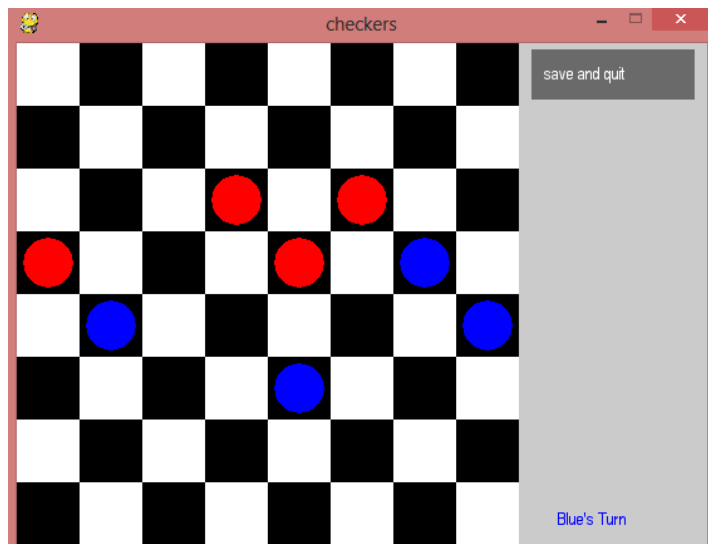
Output: Pass/Fail

Initial setup is:



Both results of AI move





Input:

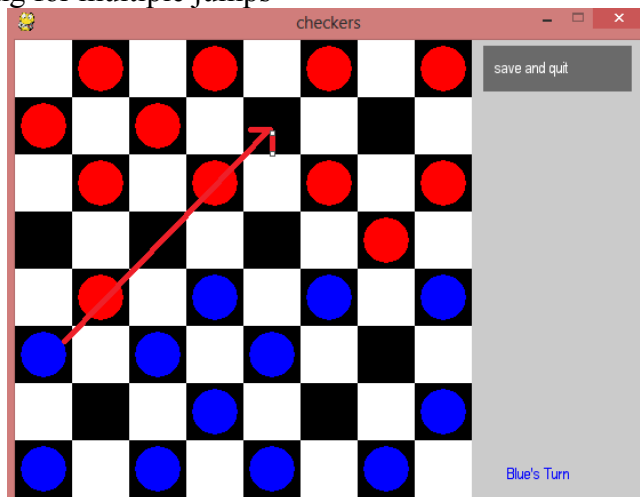
Press Save and quit

Expected Output: The game saves the game and shuts down the window.

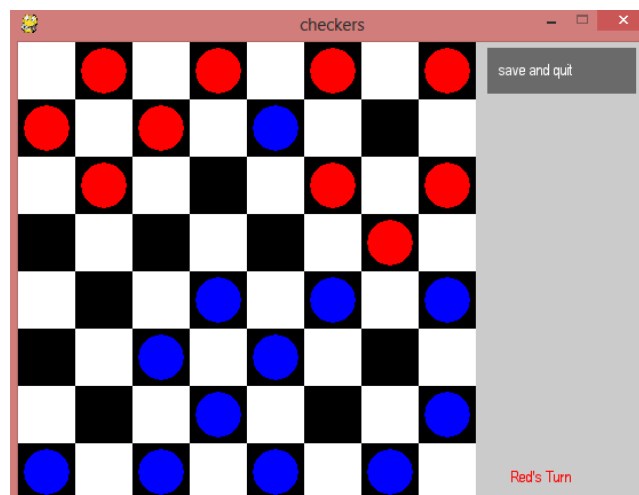
Output: The game saves the game and shuts down the window.

Result: Pass

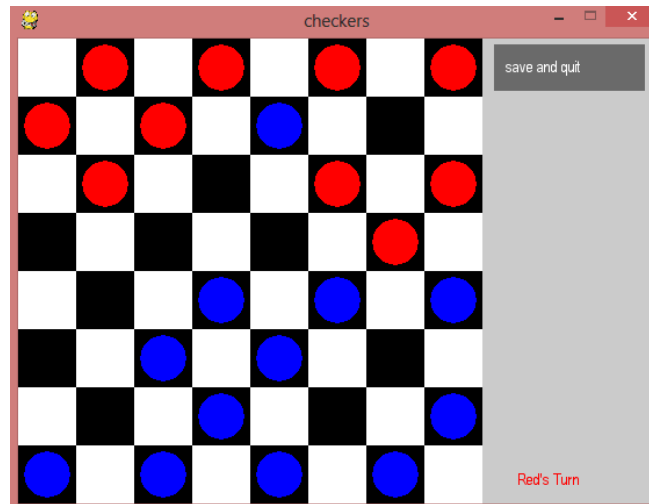
Input: Testing for multiple jumps



Expected output:

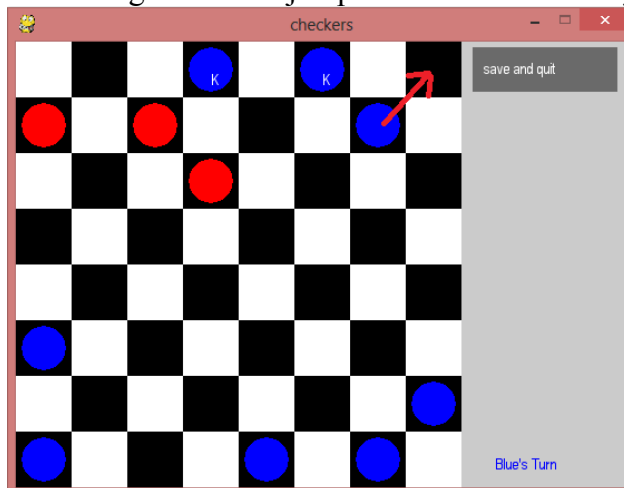


Output:

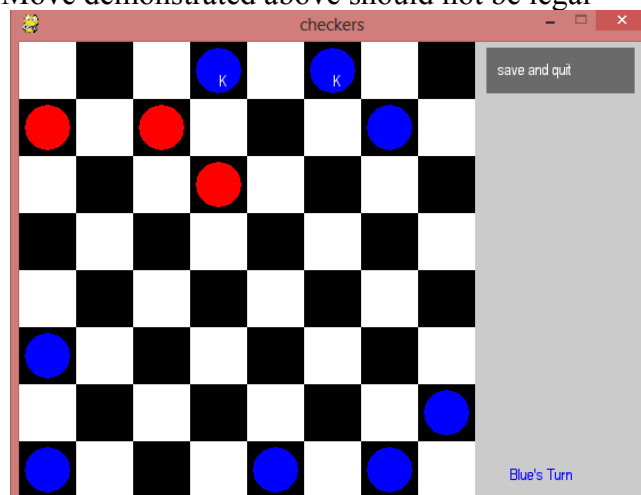


Result: Pass

Input: Testing for forced jumps blu has is forced to jump by the king piece



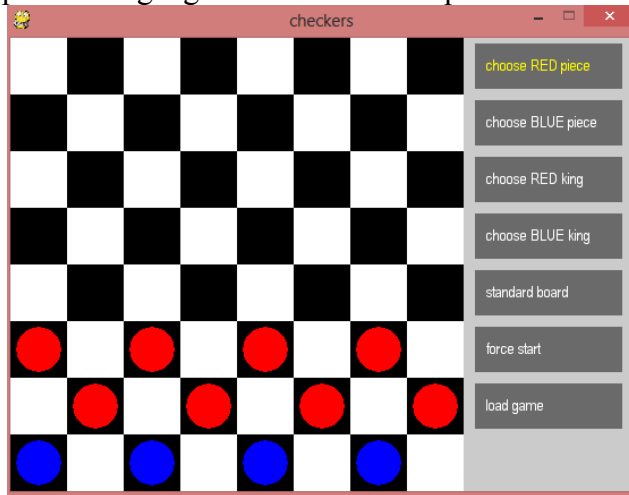
Expected Output: Move demonstrated above should not be legal



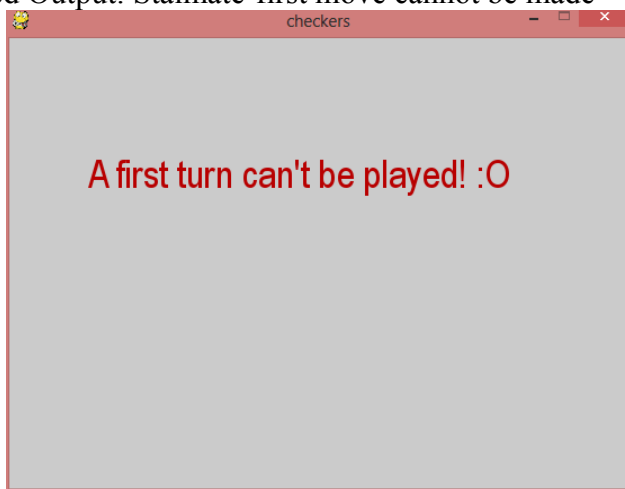
Output:

Result: PASS- unable to move that piece

Input: Starting a game in a stalemate position



Expected Output: Stalmate-first move cannot be made

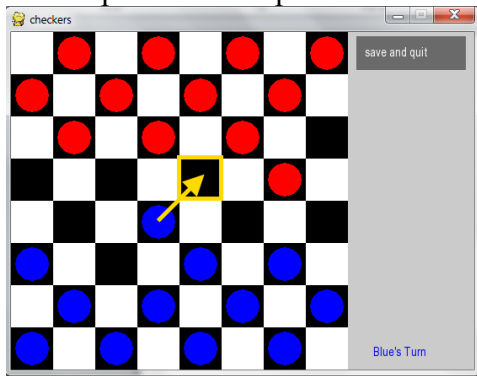


Output:

Result: Pass

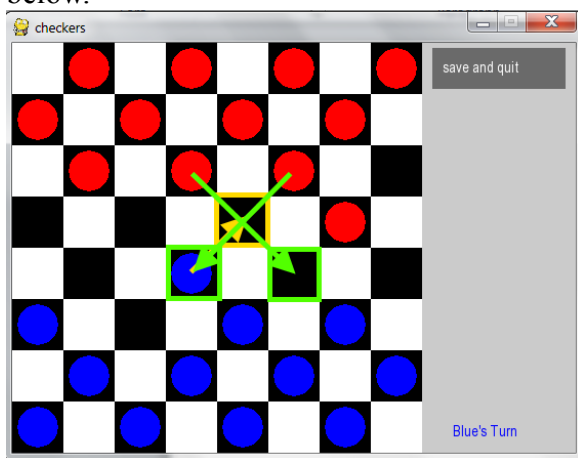
[Input]

Move specified blue piece forward.



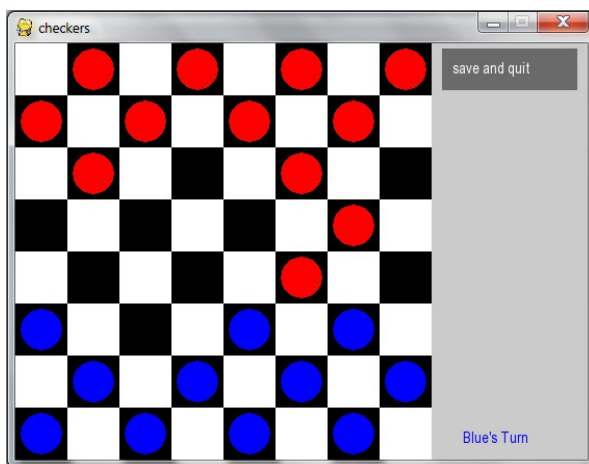
[Expected Output]

One of the specified red pieces of the AI will 'kill' the moved piece by making one of the moves below.



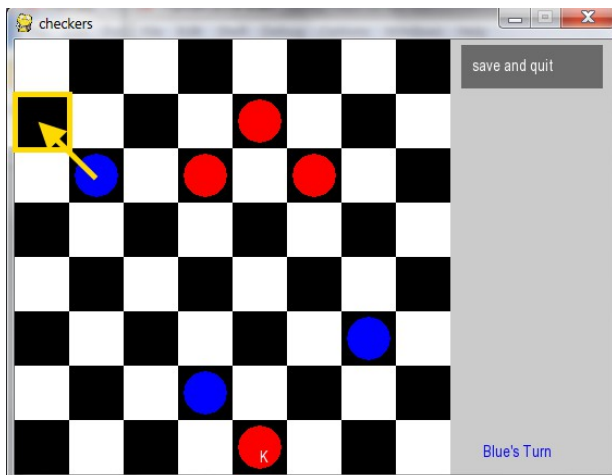
[Actual Output]

PASS



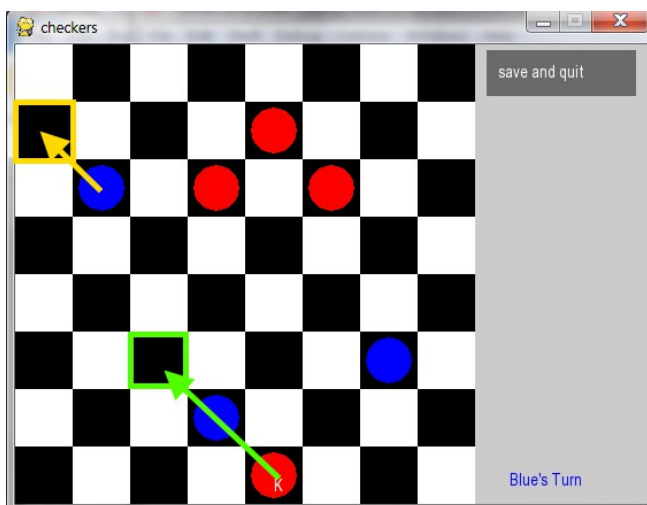
[Input]

Move specified blue piece forward.



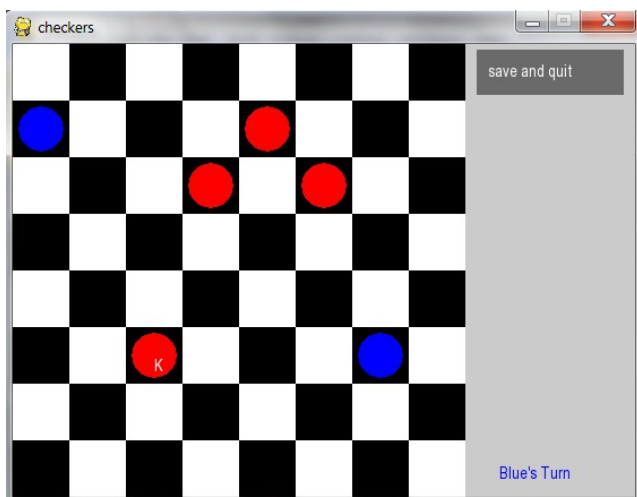
[Expected Output]

The red king piece of the AI 'kills' the blue piece nearby.



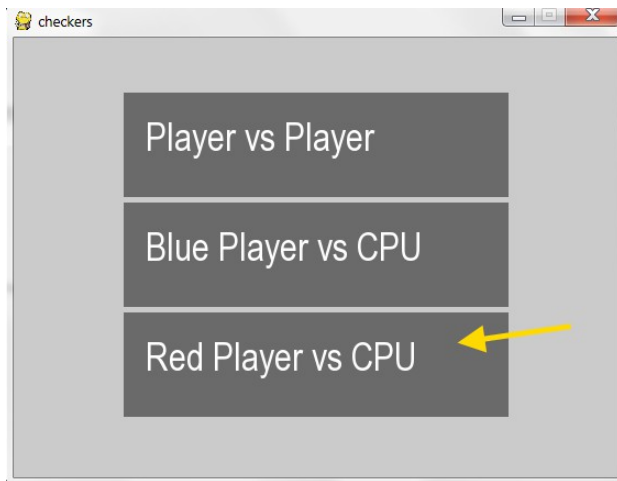
[Actual Output]

PASS



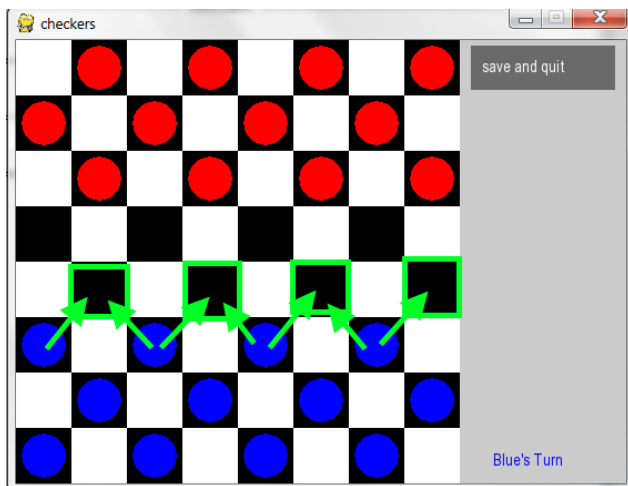
[Input]

Select 'Red Player vs CPU' mode with a standard board.



[Expected Output]

One of the blue pieces of the AI from the first row will be moved forward automatically.



[Actual Output]

PASS

