

# Testing Document

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>PLAY</b>	<b>1</b>
<b>3</b>	<b>CUSTOM</b>	<b>2</b>
3.1	No Input . . . . .	2
3.2	Input Incorrectly Formatted . . . . .	2
3.3	Invalid Location - Not on Solid Square . . . . .	3
3.4	Invalid Location - Out Of Board Bounds . . . . .	3
3.5	Too Many White Pieces . . . . .	3
3.6	Too Many Black Pieces . . . . .	4
3.7	Overlapping Pieces . . . . .	4
3.8	Accepted Board Configurations . . . . .	4
<b>4</b>	<b>SAVE</b>	<b>4</b>
<b>5</b>	<b>LOAD</b>	<b>5</b>
5.1	Save File . . . . .	5

## 1 Introduction

This document will outline testing procedures for the Checkers game. When first starting the game, the user is shown a menu with two(2) options - both of which correspond to a method of board setup

1. PLAY
2. CUSTOM
3. LOAD

## 2 PLAY

When the user clicks on PLAY, a standard 8x8 checkers board with all 24 pieces (12 white, 12 black) should be generated. To test this functionality click on the play button and verify the correct board has been generated.

### 3 CUSTOM

When the user clicks on CUSTOM, they will be prompted to enter positions for all of their pieces.

#### 3.1 No Input

In the event that the user inputs nothing, the console will display an incorrect input message and prompt the user for a correct input

##### Test Cases

- Return Key

#### 3.2 Input Incorrectly Formatted

In the event that the user inputs the piece information incorrectly, the console will display an appropriate incorrect input message and prompt the user for a correct input.

##### Test Cases

- c
- A1=e
- W=A1
- A1 = W
- A1 = KW
- 2=2
- =====
- !@\*\*
- A(5-4)=W
- AA3=W
- A1=W, C1=B
- A1=W,      C1=B
- G1=34,A7=B

### 3.3 Invalid Location - Not on Solid Square

In the event that the user inputs a location that corresponds to a light square instead of a solid square, the console will display an appropriate incorrect input message and prompt the user for a correct input

#### Test Cases

- A2=W
- B1=B
- C4=W
- F5=W
- H3=B

### 3.4 Invalid Location - Out Of Board Bounds

In the event that the user inputs a location that does not exist on the board, the console will display an incorrect input message and prompt the user for a correct input.

#### Test Cases

- A9=B
- B12=W
- I1=W
- J10=B

### 3.5 Too Many White Pieces

In the event that the user inputs too many white pieces (>12), the console will display an appropriate input message (along with how many pieces you inputted) and prompt the user for a correct input.

#### Test Cases

- A1=W,C1=W,E1=W,G1=W,A3=W,A5=W,A7=W,B8=W,B6=W,B4=W,  
B2=W,E1=W,E3=W
- E5=B,A1=W,C1=W,E1=W,G1=W,A3=W,A5=W,A7=W,B8=W,B6=W,  
B4=W,B2=W,E1=W,E3=W

### 3.6 Too Many Black Pieces

In the event that the user inputs too many black pieces (>12), the console will display an appropriate input message (along with how many pieces you inputted) and prompt the user for a correct input.

#### Test Cases

- A1=B,C1=B,E1=B,G1=B,A3=B,A5=B,A7=B,B8=B,B6=B,B4=B,B2=B,E1=B,E3=B
- A1=B,C1=B,E1=B,G1=B,A3=B,A5=B,A7=B,B8=B,B6=B,B4=B,B2=B,E1=B,E3=B,E5=W

### 3.7 Overlapping Pieces

In the event that the user inputs a location that is already filled with a piece, the previous piece will be overwritten by the new piece

#### Test Cases

- A1=B,A1=W
- A1=W,A1=WK
- A1=W,A3=B,A1=B,A3=W
- A1=W,A1=W

### 3.8 Accepted Board Configurations

In the event that the user inputs the correct format, the console will display a confirmation message and the custom game board will be generated

#### Test Cases

- A1=W
- a1=w
- A3=W,B2=B

## 4 SAVE

The user can press the save button at any time while playing. Currently, the save file is saved into a folder on the desktop. If the user doesn't have write permissions to the desktop, a message will appear saying "Save Unsuccessful".

## 5 LOAD

When the user clicks on LOAD, if the savefile exists, the game will switch to the playing state with the board set up. Loading of board configurations is handled internally by the same system as SETUP so the same limitations are shared, such as: the user cannot load a board with more than 12 pieces for each player.

### 5.1 Save File

If the save file exists, but the text inside doesn't hold legal parsable information, the game will stay in the menu and tells the user a save cannot be found. Only the first two lines of the save file are parsed. The first line contains who's turn it is to go out of BLACK, WHITE. And the second line contains the board setup as a string in the same format as inputted by a user. The user should not be writing in this file manually.

#### Test Cases

- Default board

```
1 BLACK
2 A1=W, A3=W, A7=B, B2=W, B6=B, B8=B, C1=W, C3=W, C7=B, D2=W, D6=
  B, D8=B, E1=W, E3=W, E7=B, F2=W, F6=B, F8=B, G1=W, G3=W, G7
  =B, H2=W, H6=B, H8=B"
```

- Custom board

```
1 BLACK
2 A1=W, A3=W, A7=B, B4=B, B8=B, C1=BK, C5=B, C7=W, D4=BK, D6=W,
  E3=W, F4=B, F6=B, F8=WK, G1=W, G3=W, G5=W, G7=WK, H6=B
```

- Empty board

```
1 blAcK
2
```

- Extra information

```
1 WHITE
2 A1=W, A3=W, A7=B, B4=B, B8=B, C1=BK, C5=B, C7=W, D4=BK, D6=W,
  E3=W, F4=B, F6=B, F8=WK, G1=W, G3=W, G5=W, G7=WK, H6=B
3 extra information and junk here
4 is allowed in case we want to save score and such
5 later on
```