| Function Name | # | Test Description | Sample Input | Expected Result | Actual Result | P/F |
|---|---|---|---|---|---|---|
| displayMenu() | 1 | Input 1 for "Play"<br>Input 2 for "View Scores"<br>Input 3 for "Quit" | 1 | Play | Play | P |
| | 2 | | 2 | View Scores | View Scores | P |
| | 3 | | 3 | Quit | Quit | P |
| | 4 | character input | a | Error; loops | Error; loops | P |
| displayFirst() | 1 | return a random number between 1 and 2 | 1<br>(from displayMenu()) | 1 or 2 | 2 | P |
| | 2 | | 1 | 1 or 2 | 2 | P |
| | 3 | | 1 | 1 or 2 | 1 | P |
| | 4 | | 1 | 1 or 2 | 2 | P |
| | 5 | | 1 | 1 or 2 | 1 | P |
| askMove() | 1 | return inputs as coordinates | 1 2 3 4 | Starting space: (1, 2)<br>Landing space: (3, 4) | compile error | F |
| | 2 | fixed error; used pointers to return multiple values from the function | 1 2 3 4 | Starting space: (1, 2)<br>Landing space: (3, 4) | Starting space: (1, 2)<br>Landing space: (3, 4) | P |
| | 3 | | 4 3 5 6 | Starting space: (4, 3)<br>Landing space: (5, 6) | Starting space: (4, 3)<br>Landing space: (5, 6) | P |
| | 4 | character input | 3 m 7 8' | Error; loops | Error; loops | P |
| checkMove() | 1 | checks if a move is legal. if yes, updates the board | 4 3 5 4 | legal move, correct board update | legal move, correct board update | P |
| | 2 | | 5 6 6 5 | legal move, correct board update | legal move, correct board update | P |
| | 3 | | 8 3 7 4 | legal move, correct board update | legal move, correct board update | P |
| | 4 | | 6 5 4 3 | legal capture, correct board update | legal capture, <u>incorrect</u> board update | F |
| | 5 | fixed logical error; calculation for coordinate of captured piece was wrong | 6 5 4 3 | legal capture, correct board update | legal capture, correct board update | P |
| | 6 | testing invalid input | 9 9 9 9 | illegal move | illegal move | P |
| | 7 | | 9 1 4 5 | illegal move | illegal move | P |
| | 8 | white squares | 1 1 2 2 | illegal move | illegal move | P |
| | 9 | | 3 5 4 6 | illegal move | illegal move | P |
| | 10 | landing space not empty | 2 1 1 2 | illegal move | illegal move | P |
| | 11 | jump 2 spaces w/o capture | 6 3 8 5 | illegal move | illegal move | P |
| | 12 | move Light piece as Dark | 5 6 6 5 | illegal move | illegal move | P |
| | 13 | move Dark piece as Light | 4 3 5 4 | illegal move | illegal move | P |
| updateTurn() | 1 | Current player: Player 1 (Dark) | (complete turn) | update current player to Player 2 (Light) | update current player to Player 2 (White) | P |
| | 2 | Current player: Player 2 (Light) | (complete turn) | update current player to Player 1 (Black) | update current player to Player 1 (Black) | P |
| | 3 | Current player: Player 2 (Dark) | (complete turn) | update current player to Player 1 (Light) | update current player to Player 1 (Light) | P |
| | 4 | Current player: Player 1 (Light) | (complete turn) | update current player to Player 2 (Dark) | update current player to Player 2 (Dark) | P |
| updatePieceCount() | 1 | start of game | (start game) | Dark pieces: 12<br>Light pieces: 12 | Dark pieces: 12<br>Light pieces: 12 | P |
| | 2 | capture Light piece | (capture) | Dark pieces: 12<br>Light pieces: 11 | Dark pieces: 12<br>Light pieces: 11 | P |
| | 3 | capture Dark piece | (capture) | Dark pieces: 11<br>Light pieces: 11 | Dark pieces: 11<br>Light pieces: 11 | P |
| doSurrender() | 1 | Player 1 surrenders | (surrender) | Player 2 wins<br>Player 2 score +1 | Player 2 wins<br>Player 2 score +1 | P |

| | | | | Player 1 wins Player 1 score +1 | Player 1 wins Player 1 score +1 | P |
|---|---|---|---|---|---|---|
| | 2 | Player 2 surrenders | (surrender) | Player 1 wins Player 1 score +1 | Player 1 wins Player 1 score +1 | P |
| checkReachEndRow() | 1 | Dark piece reaches end row | | Dark player wins | Dark player wins | P |
| | 2 | Light piece reaches end row | | Light player wins | Light player wins | P |
| checkAllCaptured() | 1 | Dark piece count = 0 | | Light player wins | Light player wins | P |
| | 2 | Light piece count = 0 | | Dark player wins | Dark player wins | P |
| checkPossibleMoves() | 1 | Dark possible moves = 0 | | Light player wins | Light player wins | P |
| | 2 | Light possible moves = 0 | | Dark player wins | Dark player wins | P |
| displayQuit() | 1 | Confirm YES | Y | ends program | ends program | P |
| | 2 | Confirm NO | N | continues | continues | P |
| | 3 | invalid input | O | asks again | asks again | P |
| BONUS: checkMulti() | | additional 1 capture for Dark | | multiple capture automated | multiple capture automated | P |
| | | additional 2 captures for Dark | | multiple capture automated | multiple capture automated | P |
| | | additional 1 capture for Light | | multiple capture automated | multiple capture automated | P |
| | | additional 2 captures for Light | | multiple capture automated | multiple capture automated | P |