Michael Lieu and Will Schreiner

CS Senior Design II

Assignment 1

Test Plan

When developing the tests for our Sudoku Solver program, we decided that the best first course of action was to modularize our testing approaches. As a result, we split our main testing efforts into the basic scan test, the hidden sets test, the naked sets test, and the total Sudoku test. The basic scan tests were designed to test the accuracy of the basic scan functions; the hidden sets tests were designed to test the accuracy of the hidden doubles functions; the naked sets tests were designed to naked sets and doubles functions; and the Sudoku tests were designed to test all of those functions together for varying difficulties ranging from “Easy” to “Evil.”

Each of these tests required filling in the board object and running the algorithm we wanted to test for each respective test. There were multiple test cases for each unit test, with each test case testing for certain conditions that could be encountered during a program’s runtime. For example, in the case of our naked sets tests for vertical evaluations:

test_naked_set_vertical: Tests first column

test_naked_set_vertical_2: Tests last column

test_naked_set_vertical_3: Tests to ensure two triples aren’t found as a pair

test_naked_set_vertical_4: Tests for detecting multiple naked pairs

test_naked_set_vertical_5: Tests for detecting naked triples

The following table will detail 10 of our test case descriptions:

Test ID	Purpose	Description	Inputs	Expected Outputs	Normal/ Abnormal/Boundary Case Indication	Blackbox/Whitebox test indication	Functional/Performance Test Indication	Integration/Unit Test Indication
test_hor_comp_1	Tests horizontal comparison function; Test 1	Basic test to ensure that the horizontal comparison can be solved based on existing values	Row with each square filled with a value from 1-9	Row with each square filled with a value from 1-8	Normal	Whitebox	Functional	Unit
test_vert_comp_1	Tests vertical comparison function; Test 1	Basic test to ensure that the vertical comparison can solve based on existing values	Column with each square filled with a value from 1-8	Column with each square filled with a value from 1-9	Normal	Whitebox	Functional	Unit
test_square_check_1	Tests Square Check Function; Test 1	Basic test to ensure that square compari	Board with first 3x3 square filled with	Board with first 3x3 square filled with	Normal	Whitebox	Functional	Unit

		son can solve based on existing values	values except for the last value; each square is filled with values from 1-8	values except for the last value; each square is filled with values from 1-9				
Test_hidden_doubles	Tests matches_h finding hidden doubles; Test 1	Test to ensure that hidden doubles comparison solves based on existing values	Possible values list of [1,2,3], [1,2,3], [3,4], and [3,5]	Possible values list of [1,2], [1,2], [3,4], and [3,5]	Normal	Whitebox	Functional	Unit
Test_hidden_double_2	Tests matches_h finding hidden doubles; Test 2	Test to ensure that hidden doubles comparison solves based on existing values	Possible values list of [1,3,5], [5,6,7], [1,3,7], and [8,9]	Possible values list of [1,3, [5,6,7], [1,3], and [8,9]	Normal	Whitebox	Functional	Unit
test_naked_set_horizontal	Test 1 of naked set horizontal scan; tests	Test to ensure that horizontal comparison for	For sequential squares in first row of board	For sequential squares in first row of board	Normal	Whitebox	Functional	Unit

	first row	naked sets solves based on existing values	starting from the first, possible lists of values are [1,2], [1,2]. [8], [2,4], [2,3], [1,5], [9], [6], [7]	starting from the first, possible lists of values are [1,2], [1,2]. [8], [4], [3], [5], [9], [6], [7]				
test_naked_set_vertical	Test 1 of naked set vertical scan; tests first column	Test to ensure that vertical comparison for naked sets solves based on existing values	For sequential squares in first column of board starting from the first, possible lists of values are [1,2], [1,2]. [8], [2,4], [2,3], [1,5], [9], [6], [7]	For sequential squares in first column of board starting from the first, possible lists of values are [1,2], [1,2]. [8], [4], [3], [5], [9], [6], [7]	Normal	Whitebox	Functional	Unit
test_naked_double_square	Test 1 of naked double square scan; tests	Test to ensure that square check comparison for	In a 3x3 square on the board, the squares – in	In a 3x3 square on the board, the squares – in	Normal	Whitebox	Functional	Unit

	first square	naked sets solves based on existing values	sequential order – are filled in with these possible inputs lists: [1,2,3], [1,2,3], [8,9], [2,4], [2,3], [1,5], [8,9], [6,8,9], [7]	sequential order – are filled in with these possible inputs lists: [1,2,3], [1,2,3], [8,9], [2,4], [2,3], [1,5], [8,9], [6], [7]				
Test_Easy_Sudoku	Tests easy sudoku puzzle 1	Integration testing multiple test cases for "Easy" boards from Sudoku.com	Boards with squares filled in; taken from "Easy" board values from Sudoku.com	Solved boards from "Easy" boards from Sudoku.com	Normal	Blackbox	Functional	Integration
Test_Medium_Sudoku	Tests medium sudoku puzzle 1	Integration testing multiple test cases for "Medium" boards from Sudoku.com	Boards with squares filled in; taken from "Medium" board values from Sudoku.com	Solved boards from "Medium" boards from Sudoku.com	Normal	Blackbox	Functional	Integration

Test_Hard_Sudoku	Tests hard sudoku puzzle 1	Integration test testing multiple test cases for “Hard” boards from Sudoku.com	Boards with squares filled in; taken from “Hard” board values from Sudoku.com	Solved boards from “Hard” boards from Sudoku.com	Normal	Blackbox	Functional	Integration
Test_Expert_Sudoku	Tests expert sudoku puzzle 1	Integration test testing multiple test cases for “Expert” boards from Sudoku.com	Boards with squares filled in; taken from “Expert” board values from Sudoku.com	Solved boards from “Expert” boards from Sudoku.com	Normal	Blackbox	Functional	Integration
Test_Evil_Sudoku	Tests evil sudoku puzzle 1	Integration test testing multiple test cases for “Evil” boards from Sudoku.com	Boards with squares filled in; taken from “Evil” board values from Sudoku.com	Solved boards from “Evil” boards from Sudoku.com	Normal	Blackbox	Functional	Integration

Below is the Test Case Matrix for the 10 test items:

Test ID	Normal/Abnormal/Boundary Case Indication	Blackbox/Whitebox test indication	Functional/Performance Test Indication	Integration/Unit Test Indication
test_hor_comp_1	Normal	Whitebox	Functional	Unit
test_vert_comp_1	Normal	Whitebox	Functional	Unit
test_square_check_1	Normal	Whitebox	Functional	Unit
Test_hidden_doubles	Normal	Whitebox	Functional	Unit
Test_hidden_double_2	Normal	Whitebox	Functional	Unit
test_naked_set_horizontal	Normal	Whitebox	Functional	Unit
test_naked_set_vertical	Normal	Whitebox	Functional	Unit
test_naked_double_square	Normal	Whitebox	Functional	Unit
Test_Easy_Sudoku	Normal	Blackbox	Functional	Integration
Test_Medium_Sudoku	Normal	Blackbox	Functional	Integration
Test_Hard_Sudoku	Normal	Blackbox	Functional	Integration
Test_Expert_Sudoku	Normal	Blackbox	Functional	Integration
Test_Evil_Sudoku	Normal	Blackbox	Functional	Integration