Regular Test Write-Up

CSC-260

WordHuntGame Project

Simon, Baibhav, Neil, Jacob

Test name: testGenerateRandomBoard

Rationale: Ensure that the generateRandomBoard method of WordHuntGame produces a board with the correct number of rows and columns.

Test method: JUnit assertEquals

Result: Passed

Test name: testIsValidMove

Rationale: Verify that the isValidMove method correctly determines the validity of a move.

Test method: JUnit assertTrue and assertFalse

Result: Passed

Test name: testGetFoundWords

Rationale: Check that the getFoundWords method returns a non-null list and that it is initially empty.

Test method: JUnit assertNotNull and assertTrue

Result: Passed

Test name: testGetFoundBonusWords

Rationale: Validate that the getFoundBonusWords method returns a non-null list and is initially empty.

Test method: JUnit assertNotNull and assertTrue

Result: Passed

Test name: testIsValidWord

Rationale: Confirm that the isValidWord method correctly identifies an invalid word.

Test method: JUnit assertEquals

Result: Passed

Test name: testAddFoundWord

Rationale: Ensure that the addFoundWord method adds a word to the found words list.

Test method: JUnit assertTrue

Result: Passed

Test name: testGetPossibleWords

Rationale: Check that the getPossibleWords method returns a non-null list.

Test method: JUnit assertNotNull

Result: Passed

Test name: testTearDown

Rationale: Verify that the tearDown method empties the board and the found words list.

Test method: JUnit assertTrue and isEmpty check

Result: Passed

Test name: testFoundWordBonusFlag

Rationale: Confirm that adding a word with the bonus flag results in the word being in the found bonus words list but not in the found words list.

Test method: JUnit assertTrue and assertFalse

Result: Passed

Test name: testGetBoardAfterGenerateRandomBoard

Rationale: Check that the getBoard method returns a non-null board after calling generateRandomBoard.

Test method: JUnit assertNotNull

Result: Passed

Test name: testGetFoundWords

Rationale: Ensure that the list of found words is not null, is initially empty and contains a new found word when it is added.

Test method: JUnit assertNotNull, assertTrue and assertEquals

Result: Passed

Test name: testTearDown

Rationale: Verify that the tearDown method empties the list of found words, making the number of found words zero.

Test method: JUnit assertTrue and assertEquals

Result: Passed

Test name: testIsValidWord

Rationale: Confirm that the isValidWord method correctly identifies an invalid word.

Test method: JUnit assertEquals

Result: Passed

Test name: testGetPossibleWords

Rationale: Check that the getPossibleWords method returns a non-null list that is initially empty.

Test method: JUnit assertNotNull and assertTrue

Result: Passed

Test name: testGetFoundWords

Rationale: Check that the getFoundBonusWors method returns a non-null list that is initially empty.

Test method: JUnit assertNotNull and assertTrue

Result: Passed

Test name: testGetFoundBonusWords

Rationale: Validate that the getFoundBonusWors method returns a non-null list that is initially empty.

Test method: JUnit assertNotNull and assertTrue

Result: Passed

Test name: testAddFoundWord

Rationale: Ensure that the addFoundWord method adds a word to the found words list, increasing the number of found words by one.

Test method: JUnit assertTrue and assertEquals

Result: Passed

Test name: testFoundWordBonusFlag

Rationale: Confirm that adding a word with the bonus flag results in the word being in the found bonus words list but not in the found words list.

Test method: JUnit assertTrue and assertFalse

Result: Passed

Test name: testTearDown

Rationale: Verify that the tearDown method empties the board and the found words list.

Test method: JUnit assertTrue and isEmpty check

Result: Passed

GUI Tests

Test name: GUI Click and Drag tests

Rationale: We needed the ability to click and drag over tiles in the game board

Test method: Trial and error, since it is user input, there isn't a very easy way to automate this test with something like JUnit

Result: Passed, after a lot of testing

Test name: GUI Testing In-Game Menu

Rationale: Ensure that the in-game menu works with all functionality (new game, save game, and quit) properly functioning

Test method: Trial and error

Result: Passed

Test name: GUI Main Menu testing

Rationale: The buttons in the main menu do what they are supposed to do, i.e. Quit button quits game, new game generates new board, load game loads a board from a file.

Test method: Trial and error, since it is user input, there isn't a very easy way to automate this test with something like JUnit

Result: Passed

Test name: GUI Add Word Update Testing

Rationale: Ensure that the bonus words are highlighted in the found word list and separated from the regular work

Test method: Trial and error

Result: Passed

Test name: GUI Testing For Current Word

Rationale: Ensure that the current word display at the bottom of the screen updates as the user drags over tiles in the game board

Test method: Trial and error

Result: Passed