# Feature 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 8 | describe("Features", *function*(){ | | 9 | it("should show the initial sequence", *function*(){ | | 10 | *var* server=sinon.fakeServer.create(); | | 11 | *var* wordDisplayed=$('<div>'); | | 12 | *var* game=new Game({ | | 13 | wordDisplayed:wordDisplayed | | 14 | }); | | 15 |  | | 16 | server.respondWith('/secretword','{"word":"cocodrillo"}'); | | 17 | server.respond(); | | 18 |  | | 19 | wordDisplayed.html().should.equal('\_\_\_\_\_\_\_\_\_\_'); | | 20 | }); | | 21 | }); | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1 | describe("Game", *function*() { | | 2 | describe("show the a initial sequence with dashes", *function*() { | | 3 | it("should show a sequence with one character", *function*() { | | 4 | *var* wordDisplayed = $('<div>'); | | 5 | *var* game = new Game({ | | 6 | wordDisplayed: wordDisplayed | | 7 | }); | | 8 |  | | 9 | game.showInitialSequence('g'); | | 10 |  | | 11 | wordDisplayed.html().should.equal('\_'); | | 12 | }); | | 13 | it("should show a sequence with multiple characters", *function*() { | | 14 | *var* wordDisplayed = $('<div>'); | | 15 | *var* game = new Game({ | | 16 | wordDisplayed: wordDisplayed | | 17 | }); | | 18 |  | | 19 | game.showInitialSequence('ga'); | | 20 |  | | 21 | wordDisplayed.html().should.equal('\_\_'); | | 22 | }); | | 23 | }); | | 24 | it("should show the initial sequence after load", *function*(){ | | 25 | *var* server=sinon.fakeServer.create(); | | 26 | *var* wordDisplayed=$('<div>'); | | 27 | *var* game = new Game({ | | 28 | wordDisplayed: wordDisplayed | | 29 | }); | | 30 |  | | 31 | *var* called=false; | | 32 | *game*.showInitialSequence=*function*(*secretword*) { | | 33 | called=secretword=='gato'; | | 34 | }; | | 35 |  | | 36 | server.respondWith('/secretword','{"word":"gato"}'); | | 37 | server.respond(); | | 38 |  | | 39 | called.should.equal(true); | | 40 | }); | | 41 | }); | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1 | *function* Game(*attributes*) { | | 2 | this.wordDisplayed = attributes.wordDisplayed; | | 3 | this.loadSecretWord(); | | 4 | } | | 5 |  | | 6 | *Game*.prototype = { | | 7 | loadSecretWord: *function*() { | | 8 | *var* self = this; | | 9 | $.getJSON('/secretword', *function*(data) { | | 10 | self.showInitialSequence(data.word); | | 11 | }); | | 12 | }, | | 13 | showInitialSequence: *function*(*secretWord*) { | | 14 | *var* sequence = ''; | | 15 | for(*var* i = 0; i < secretWord.length; i++) { | | 16 | sequence += '\_'; | | 17 | } | | 18 | this.wordDisplayed.html(sequence); | | 19 | } | | 20 | }; | |

# Code Analisis JSHINT

# Feature 2

## Feature

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 35 | it("should be able to play", *function*() { | | 36 | playWith('cocodrillo'); | | 37 |  | | 38 | $('button:contains("o")', buttonList).click(); | | 39 | $('button:contains("d")', buttonList).click(); | | 40 |  | | 41 | wordDisplayed.html().should.be.equal('\_o\_od\_\_\_\_o'); | | 42 | }); | |

## Unit Tests

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 47 | describe("create letter buttons", *function*() { | | 48 | it("should show the first letter in alphabet", *function*() { | | 49 | game.createButtons(); | | 50 |  | | 51 | $('button:contains("a")', buttonList).should.have.length(1); | | 52 | }); | | 53 |  | | 54 | it("should show the last letter in alphabet", *function*() { | | 55 | game.createButtons(); | | 56 |  | | 57 | $('button:contains("z")', buttonList).should.have.length(1); | | 58 | }); | | 59 |  | | 60 | it("should show all letters in alphabet", *function*() { | | 61 | game.createButtons(); | | 62 |  | | 63 | $('button', buttonList).should.have.length(26); | | 64 | }); | | 65 | }); | | 66 |  | | 67 | describe("try letters", *function*() { | | 68 | it("should show the letter when the letter match", *function*() { | | 69 | playWith('g'); | | 70 |  | | 71 | game.tryLetter('g'); | | 72 |  | | 73 | wordDisplayed.html().should.equal('g'); | | 74 | }); | | 75 | it("should show dash when the letter does not match", *function*() { | | 76 | playWith('g'); | | 77 |  | | 78 | game.tryLetter('e'); | | 79 |  | | 80 | wordDisplayed.html().should.equal('\_'); | | 81 | }); | | 82 | it("should show only the letters that match", *function*() { | | 83 | playWith('ga'); | | 84 |  | | 85 | game.tryLetter('g'); | | 86 |  | | 87 | wordDisplayed.html().should.equal('g\_'); | | 88 | }); | | 89 | it("should show only the letters that match in different positions", *function*() { | | 90 | playWith('ga'); | | 91 |  | | 92 | game.tryLetter('a'); | | 93 |  | | 94 | wordDisplayed.html().should.equal('\_a'); | | 95 | }); | | 96 | it("should show only the letters that match of consecutive trials", *function*(){ | | 97 | playWith('gato'); | | 98 |  | | 99 | game.tryLetter('g'); | | 100 | game.tryLetter('a'); | | 101 |  | | 102 | wordDisplayed.html().should.equal('ga\_\_'); | | 103 | }); | | 104 | it("should show repetitive letters that match", *function*(){ | | 105 | playWith('gatito'); | | 106 |  | | 107 | game.tryLetter('t'); | | 108 |  | | 109 | wordDisplayed.html().should.equal('\_\_t\_t\_'); | | 110 | }); | | 111 |  | | 112 | it("should try the letter when click a letter button", *function*(){ | | 113 | *var* called=false; | | 114 | *game*.tryLetter=*function*() { | | 115 | called=true; | | 116 | }; | | 117 | game.createButtons(); | | 118 |  | | 119 | $('button:contains("a")', buttonList).click(); | | 120 |  | | 121 | called.should.equal(true); | | 122 | }); | |

## Implementation

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 24 | createButtons: *function*() { | | 25 | for(*var* i = 97; i <= 122; i++) { | | 26 | *var* button = $('<button>' + *String*.fromCharCode(i) + '</button>'); | | 27 | button.click($.proxy(this.onButtonClick,this)); | | 28 | this.buttonList.append(button); | | 29 | } | | 30 | }, | | 31 | onButtonClick: *function*(*e*) { | | 32 | *var* letter = $(e.target).html(); | | 33 | this.tryLetter(letter); | | 34 | }, | | 35 | tryLetter: *function*(*letter*) { | | 36 | for(*var* index = 0; index < this.secretWord.length; index++) { | | 37 | if(letter === this.secretWord[index]) { | | 38 | this.initialSequence = this.initialSequence.substr(0, index) + | | 39 | letter + this.initialSequence.substr(index + 1); | | 40 | this.wordDisplayed.html(this.initialSequence); | | 41 | } | | 42 | } | |

# Refactoring

* Propiedades
* Métodos
* Mover clase Game a un Nuevo archivo.

# Feature 3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 46 | it("should the number of failed attempts", *function*(){ | | 47 | playWith('cocodrillo'); | | 48 |  | | 49 | $('button:contains("a")', buttonList).click(); | | 50 | $('button:contains("e")', buttonList).click(); | | 51 |  | | 52 | failedTrials.html().should.be.equal('2'); | | 53 | }); | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 128 | it("should show the failed trials", *function*(){ | | 129 | //GIVEN Game.getFailedTrials return 2 | | 130 | /\*      gameController.game={ | | 131 | getFailedTrials:function() { | | 132 | return 2; | | 133 | } | | 134 | };\*/ | | 135 | *var* game=sinon.stub(gameController.game); | | 136 | game.getFailedTrials.returns(2); | | 137 |  | | 138 | gameController.tryLetter('a'); | | 139 |  | | 140 | failedTrials.html().should.equal('2'); | | 141 | }); | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 34 | tryLetter: *function*(*letter*) { | | 35 | this.game.tryLetter(letter); | | 36 | this.wordDisplayed.html(this.game.currentSequence); | | 37 | this.failedTrials.html(this.game.getFailedTrials()); | | 38 | } | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 3 | describe("Failed Trials", *function*() { | | 4 | it("should return 1 failed trial when try 1 incorrect letter", *function*() { | | 5 | *var* game = new Game(); | | 6 | game.secretWord = 'gato'; | | 7 |  | | 8 | game.tryLetter('e'); | | 9 |  | | 10 | game.getFailedTrials().should.equal(1); | | 11 | }); | | 12 |  | | 13 | it("should return 2 failed trials when try 2 incorrect letter", *function*() { | | 14 | *var* game = new Game(); | | 15 | game.secretWord = 'gato'; | | 16 |  | | 17 | game.tryLetter('e'); | | 18 | game.tryLetter('i'); | | 19 |  | | 20 | game.getFailedTrials().should.equal(2); | | 21 | }); | | 22 |  | | 23 | it("should return 2 failed trials when try 2 incorrect and 1 correct letter", *function*() { | | 24 | *var* game = new Game(); | | 25 | game.secretWord = 'gato'; | | 26 | game.currentSequence='\_\_\_\_'; | | 27 |  | | 28 | game.tryLetter('a'); | | 29 | game.tryLetter('e'); | | 30 | game.tryLetter('i'); | | 31 |  | | 32 | game.getFailedTrials().should.equal(2); | | 33 | }); | | 34 | }); | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 13 | tryLetter:*function*(*letter*) { | | 14 | for(*var* index = 0; index < this.secretWord.length; index++) { | | 15 | if(letter === this.secretWord[index]) { | | 16 | this.updateCurrentSequence(index,letter); | | 17 | } | | 18 | } | | 19 | if (this.secretWord.indexOf(letter)===-1) { | | 20 | this.failedTrials+=1; | | 21 | } | | 22 | }, | | 23 | updateCurrentSequence: *function*(*index,letter*) { | | 24 | this.currentSequence = this.currentSequence.substr(0, index) + | | 25 | letter + | | 26 | this.currentSequence.substr(index + 1); | | 27 | }, | | 28 | getFailedTrials:*function*() { | | 29 | return this.failedTrials; | | 30 | } | |

# Feature 4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 58 | it("should prevent to try the same letter twice", *function*(){ | | 59 | playWith('cocodrillo'); | | 60 |  | | 61 | tryWith("a"); | | 62 | tryWith("a"); | | 63 |  | | 64 | failedTrials.html().should.equal(1); | | 65 | }); | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 153 | it("should remove the button after click", *function*(){ | | 154 | sinon.stub(gameController.game); | | 155 | gameController.createButtons(); | | 156 | *var* button=$('button:contains("a")', buttonList); | | 157 |  | | 158 | button.click(); | | 159 |  | | 160 | button.is(':visible').should.equal(false); | | 161 | }); | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 9 | *function* loadFixture(){ | | 10 | fixtures.path='fixtures'; | | 11 | fixtures.load("game.html"); | | 12 | return $('#' + fixtures.containerId).contents().find('body'); | | 13 | } | | 14 |  | | 15 | beforeEach(*function*() { | | 16 | *var* fixture=loadFixture(); | | 17 | server = sinon.fakeServer.create(); | | 18 | wordDisplayed = $('#wordDisplayed',fixture); | | 19 | buttonList = $('#letters',fixture); | | 20 | failedTrials=$('#failedAttempts',fixture); | | 21 | gameController = new GameController({ | | 22 | wordDisplayed: wordDisplayed, | | 23 | buttonList: buttonList, | | 24 | failedTrials:failedTrials | | 25 | }); | | 26 | }); | | 27 |  | | 28 | afterEach(*function*(){ | | 29 | fixtures.cleanUp(); | | 30 | }); | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 30 | onButtonClick: *function*(*e*) { | | 31 | *var* letter = $(e.target).html(); | | 32 | this.tryLetter(letter); | | 33 | $(e.target).remove(); | | 34 | }, | |