

Fetching contributors...

 Cannot retrieve contributors at this time

32 lines (24 sloc) | 858 Bytes

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="utf-8">
5    <title>Jasmine Spec Runner v2.8.0</title>
6
7    <link rel="shortcut icon" type="image/png" href="lib/jasmine-2.8.0/jasmine_favicon.png">
8    <link rel="stylesheet" href="lib/jasmine-2.8.0/jasmine.css">
9
10   <script src="lib/jasmine-2.8.0/jasmine.js"></script>
11   <script src="lib/jasmine-2.8.0/jasmine-html.js"></script>
12   <script src="lib/jasmine-2.8.0/boot.js"></script>
13
14   <!-- include source files here... -->
15   <script src="src/Game.js"></script>
16   <!-- <script src="src/Frame.js"></script>
17   <script src="src/Bonus.js"></script> -->
18
19   <!-- include spec files here... -->
20   <script src="spec/GameSpec.js"></script>
21   <!-- <script src="spec/FrameSpec.js"></script>
22   <script src="spec/BonusSpec.js"></script>
23   <script src="spec/SpecHelper.js"></script> -->
24
25   </script>
26
27 </head>
28
29 <body>
30 </body>
31 </html>
```

Tree 856f6f6a25 ▾

bowling-challenge / index.html

Find file

Copy path

Fetching contributors...

Cannot retrieve contributors at this time

80 lines (72 sloc) | 2.19 KB

1

<!DOCTYPE html>

2

<!DOCTYPE html>

3

<html>

4

<head>

5

<meta charset="utf-8">

6

<link href="https://fonts.googleapis.com/css?family=VT323" rel="stylesheet">

7

<link rel="stylesheet" type="text/css" href="/main.css" />

8

<script src="https://code.jquery.com/jquery-3.2.1.min.js" integrity="sha256-hwg4gsxgFZh0sEEamd0YGBf13FyQuiTwlAQgxVSNgt4=" cross

9

<script src="/src/Game.js" type="text/javascript"></script>

10

<script src="/src/Interface.js" type="text/javascript"></script>

11

</head>

12

13

<body>

14

15

<table>

16

<tr>

17

<th>FRAME</th>

18

<th colspan="2">1</th>

19

<th colspan="2">2</th>

20

<th colspan="2">3</th>

21

<th colspan="2">4</th>

22

<th colspan="2">5</th>

23

<th colspan="2">6</th>

24

<th colspan="2">7</th>

25

<th colspan="2">8</th>

26

<th colspan="2">9</th>

27

<th colspan="3">10</th>

28

<th>Total</th>

29

</tr>

30

<tr>

31

<td>Player Name</td>

32

<td id="11">--</td>

33

<td id="12" class="second-roll">--</td>

34

<td id="21">--</td>

35

<td id="22" class="second-roll">--</td>

36

<td id="31">--</td>

37

<td id="32" class="second-roll">--</td>

38

<td id="41">--</td>

39

<td id="42" class="second-roll">--</td>

40

<td id="51">--</td>

41

<td id="52" class="second-roll">--</td>

42

<td id="61">--</td>

43

<td id="62" class="second-roll">--</td>

44

<td id="71">--</td>

45

<td id="72" class="second-roll">--</td>

46

<td id="81">--</td>

47

<td id="82" class="second-roll">--</td>

48

<td id="91">--</td>

49

<td id="92" class="second-roll">--</td>

50

<td id="101">--</td>

51

<td id="102" class="second-roll">--</td>

52

<td id="103" class="second-roll">--</td>

53

<td id="total" colspan="1" rowspan="2">--</td>

54

55

</tr>

56

<tr>

57

<td>Running Total</td>

58

<td id="frame-1"colspan="2">--</td>

59

<td id="frame-2"colspan="2">--</td>

60

<td id="frame-3"colspan="2">--</td>

61

<td id="frame-4"colspan="2">--</td>

62

<td id="frame-5"colspan="2">--</td>

63

<td id="frame-6"colspan="2">--</td>

64

<td id="frame-7"colspan="2">--</td>

65

<td id="frame-8"colspan="2">--</td>

66

<td id="frame-9"colspan="2">--</td>

67

<td id="frame-10"colspan="3">--</td>

68

</tr>

69

</table>

70

71

<form>

72

<input id="score" type="text" placeholder="Enter Score"/>

73

<button id="score-button" type="button" name="button">Submit</button>

74

</form>

75

76

</body>

77

78

79

</html>

Tree: 856f6f6a25 ▾

bowling-challenge / main.css

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

36 lines (29 sloc) | 430 Bytes

```
1  body {
2    background-color: black;
3  }
4
5
6  th {
7    font-weight: normal;
8    border: 1px solid white;
9    color: #efcb7c;
10 }
11
12 td {
13   width: 130px;
14   border: 1px solid white;
15   color: #efcb7c;
16   padding: 5px;
17 }
18
19 table {
20   font-family: 'VT323', monospace;
21   font-size: 30px;
22   margin: auto;
23   text-align: center;
24   border: 1px solid white;
25   color: #efcb7c;
26 }
27
28 form {
29   margin: auto;
30   text-align: center;
31 }
32
33 .second-roll {
34   color: #e25a2d;
35 }
```

Tree856f6f6a25

bowling-challenge / spec / BonusSpec.js

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

50 lines (40 sloc) | 1.12 KB

```
1 describe("Bonus", function(){
2
3   beforeEach(function(){
4     bonus = new Bonus()
5   });
6
7   describe("#isSpare", function(){
8     it("is set to false by default", function(){
9       expect(bonus.isSpare()).toEqual(false)
10    });
11  });
12
13  describe("#setSpareTrue", function(){
14    it("sets spare to be true", function(){
15      bonus.setSpareTrue()
16      expect(bonus.isSpare()).toEqual(true)
17    });
18  });
19
20  describe("#setSpareFalse", function(){
21    it("is set to false by default", function(){
22      bonus.setSpareFalse()
23      expect(bonus.isSpare()).toEqual(false)
24    });
25  });
26
27  describe("#isStrike", function(){
28    it("is set to false by default", function(){
29      expect(bonus.isStrike()).toEqual(false)
30    });
31  });
32
33  describe("#setStrikeTrue", function(){
34    it("changes #isSpareBonus from false to true", function(){
35      bonus.setStrikeTrue()
36      expect(bonus.isStrike()).toEqual(true)
37    });
38
39  });
40
41  describe("#setStrikeFalse", function(){
42    it("changes #isStrikeBonus from true to false", function(){
43      bonus.setStrikeTrue()
44      bonus.setStrikeFalse()
45      expect(bonus.isStrike()).toEqual(false)
46    });
47  });
48
49  });
```

Tree: 856f6f6a25 ▾

bowling-challenge / spec / FrameSpec.js

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

37 lines (31 sloc) | 814 Bytes

```
1 describe("Frame", function(){
2
3   beforeEach(function(){
4     frame = new Frame();
5   });
6
7   describe("#rollOne", function(){
8     it("stores the first roll", function(){
9       frame.setRollOne(3)
10      expect(frame.getRollOne()).toEqual(3)
11    });
12  });
13
14  describe("#rollTwo", function(){
15    it("stores the second roll", function(){
16      frame.setRollTwo(5)
17      expect(frame.getRollTwo()).toEqual(5)
18    });
19  });
20
21  describe("#bonus", function(){
22    it("stores the bonus score", function(){
23      frame.setBonus(7)
24      expect(frame.getBonus()).toEqual(7)
25    });
26  });
27
28  describe("#sumAllRolls", function(){
29    it("sums totals from roll 1, 2 and bonus", function(){
30      frame.setRollOne(3)
31      frame.setRollTwo(5)
32      frame.setBonus(7)
33      expect(frame.sumAllRolls()).toEqual(15)
34    });
35  });
36  });
```




Tree: 856f6f6a25 ▾

bowling-challenge / spec / GameSpec.js

Find file

Copy path


Fetching contributors...

 Cannot retrieve contributors at this time

166 Lines (140 sloc) | 4.27 KB

```
1 describe('Game', function(){
2
3   beforeEach(function(){
4     game = new Game();
5   });
6
7   function gutterGame(){
8     for (var i = 0; i < 20; i++){
9       game.addPins(0)
10    }
11  }
12
13  function roll145(){
14    for (var i = 0; i < 20; i++){
15      game.addPins(5)
16    }
17  }
18
19  function perfectGame(){
20    for (var i = 0; i < 12; i++){
21      game.addPins(10)
22    }
23  }
24
25  function normalGame(){
26    for (var i = 0; i < 20; i++){
27      game.addPins(4)
28    }
29  }
30
31
32  describe('#getCurrentScore', function(){
33    it('10 bowls of 1 returns a current score of 20', function(){
34      for (var i = 0; i < 20; i++){
35        game.addPins(1)
36      }
37      expect(game.getCurrentScore()).toEqual(20)
38    });
39
40    it('gutter game returns a current score of zero', function(){
41      for (var i = 0; i < 20; i++){
42        game.addPins(0)
43      }
44      expect(game.getCurrentScore()).toEqual(0)
45    });
46  });
47
48  describe('#moveToNextFrame', function(){
49    it('moves the frame on by one', function(){
50      game.moveToNextFrame()
51      expect(game.getCurrentFrame()).toEqual(2)
52    });
53
54    it('moves to the next frame after two normal rolls', function(){
55      game.addPins(1)
56      game.addPins(1)
57      expect(game.getCurrentFrame()).toEqual(2)
58    });
59
60    it('moves to the next frame when a 10 is rolled on the first turn of a frame', function(){
61      game.addPins(10)
62      expect(game.getCurrentFrame()).toEqual(2)
63    });
64  });
65
66  describe('#addPins', function(){
67    it('adds the number of pins knocked down to the current frame', function(){
68      game.addPins(1)
69      expect(game.scores[0]).toEqual([1])
70    });
71
72    it('throws an error if the number of pins entered is > 10', function(){
73      expect( function(){ game.addPins(20); }).toThrow("You can't knock down over 10 pins")
74    });
75
76    it('throws an error if the number of pins over two rolls is above 10', function(){
77      game.addPins(8)
78      expect( function(){ game.addPins(3); }).toThrow("You can't knock down over 10 pins")
79    });
80
81    it('throws an error when the game is over (spare or strike)', function(){
82      perfectGame()
83      expect( function(){ game.addPins(4) }).toThrow("The game is over")
84    });
85
86    it('throws an error when the game is over (normal game)', function(){
87      normalGame()
88      expect( function(){ game.addPins(4) }).toThrow("The game is over")
89    });
90  });
91
92  describe('#getCurrentTurn', function(){
93    it('returns 1 if it is the first turn in a frame', function(){
94      expect(game.getCurrentTurn()).toEqual(1)
95    });
96
97    it('returns 2 if it is the second turn in a frame', function(){
98      game.addPins(1)
99      expect(game.getCurrentTurn()).toEqual(2)
100    });
101
102    it('returns 1 after two normal rolls', function(){
103      game.addPins(1)
104      game.addPins(1)
105      expect(game.getCurrentTurn()).toEqual(1)
106    });
107  });
108
109  describe('#addSpare', function(){
110    it('adds the value of the next roll following a spare to the previous frame', function(){
111      game.addPins(1)
112      game.addPins(9)
113      game.addPins(3)
114      expect(game.scores[0][2]).toEqual(3)
115    });
116
117    it('returns a total score of 150', function(){
118      roll145()
119      expect(game.getCurrentScore()).toEqual(145)
120    });
121  });
122
123  describe('#addStrike', function(){
124    it('adds the value of the next two rolls to the previous frame if the first roll in that frame is 10', function(){
125      game.addPins(10)
126      game.addPins(2)
127      game.addPins(2)
128      expect(game.getCurrentScore()).toEqual(18)
129    });
130
131    it('adds roll to the previous two frames if both preceeding rolls are 10', function(){
132      game.addPins(10)
133      game.addPins(10)
134      game.addPins(5)
135      expect(game.getCurrentScore()).toEqual(45)
136    });
137  });
138
139  describe('rolling a spare in the last frame', function(){
140    it('has a total of 268', function(){
141      game.addPins(10)
142      game.addPins(10)
143      game.addPins(10)
144      game.addPins(10)
145      game.addPins(10)
146      game.addPins(10)
147      game.addPins(10)
148      game.addPins(10)
149      game.addPins(10)
150      game.addPins(3)
151      game.addPins(7)
152      game.addPins(5)
153      console.log(game)
154      expect(game.getCurrentScore()).toEqual(268)
155    });
156  });
157
158  describe('rolling a perfect game', function(){
159    it('has a total of 300', function(){
160      perfectGame()
161      console.log(game)
162      expect(game.getCurrentScore()).toEqual(300)
163    });
164  });
165 });
```

Fetching contributors...

 Cannot retrieve contributors at this time

177 lines (148 sloc) | 5.09 KB

```
1 describe("Game", function(){
2
3   function Roll4Then4(){
4     game.addRoll(4)
5     game.addRoll(4)
6   }
7
8   function RollSpare(){
9     game.addRoll(5)
10    game.addRoll(5)
11  }
12
13  function RollStrike(){
14    game.addRoll(10)
15  }
16
17  beforeEach(function(){
18    bonus = jasmine.createSpyObj('new Bonus()', ['setSpareTrue', 'setSpareFalse', 'setStrike', 'isSpare', 'isStrike']);
19    game = new Game(new Bonus());
20  });
21
22  describe("#getCurrentScore", function(){
23    it("returns a current score of 0", function(){
24      expect(game.getCurrentScore()).toEqual(0)
25    });
26
27    it("has a score of 40 after ten frames", function(){
28      for (var i = 0; i < 10; i++){
29        game.addRoll(4)
30      }
31      expect(game.getCurrentScore()).toEqual(40)
32    });
33  });
34
35  describe("#frames", function(){
36    it("is an object", function(){
37      expect(game.frames).toEqual(jasmine.any(Object))
38    });
39
40    it("returns an instance of Frame as the first element", function(){
41      expect(game.frames[1]).toEqual(jasmine.any(Frame))
42    })
43  });
44
45  describe("#getCurrentFrame", function(){
46    it("returns a current frame of 1", function(){
47      expect(game.getCurrentFrame()).toEqual(1)
48    });
49
50    it("doesn't increase the frame after only one roll (below 10)", function(){
51      game.addRoll(4)
52      expect(game.getCurrentFrame()).toEqual(1)
53    });
54
55    it("moves on to the next frame if the first roll is 10", function(){
56      game.addRoll(10)
57      expect(game.getCurrentFrame()).toEqual(2)
58    })
59
60    it("rolling twice (below 5) increases the current frame by 1", function(){
61      Roll4Then4();
62      expect(game.getCurrentFrame()).toEqual(2)
63    });
64
65    it("has a maximum of ten frames", function(){
66      for (var i = 0; i < 20; i++){
67        game.addRoll(4)
68      }
69      expect(game.getCurrentFrame()).toEqual(10)
70    });
71  });
72
73  describe("#isRoll0ne", function(){
74    it("is set to be true by default", function(){
75      expect(game.isRoll0ne()).toEqual(true)
76    });
77
78    it("is set to false after taking one roll", function(){
79      game.addRoll(4)
80      expect(game.isRoll0ne()).toEqual(false)
81    });
82
83    it("is set to true after making two rolls (below 5)", function(){
84      Roll4Then4();
85      expect(game.isRoll0ne()).toEqual(true)
86    })
87
88    it("is set to false after making three rolls (below 5)", function(){
89      Roll4Then4()
90      game.addRoll(4)
91      expect(game.isRoll0ne()).toEqual(false)
92    })
93  });
94
95  describe("#addRoll", function(){
96    it("adds a roll of 4 to the current score", function(){
97      game.addRoll(4)
98      expect(game.getCurrentScore()).toEqual(4)
99    });
100
101    it("throws an error if the number of pins hit is > 10", function(){
102      expect(function() { game.addRoll(11) }).toThrow("You can't knock down more than 10 pins")
103    })
104
105    it("throws an error if score for roll 2 would take total for frame over 10", function(){
106      game.addRoll(4)
107      expect(function(){ game.addRoll(7) }).toThrow("Your rolls can't sum over 10")
108    });
109  });
110
111  describe("#addFrame", function(){
112    it("adds one frame to the frame total", function(){
113      game.addFrame()
114      expect(game.getCurrentFrame()).toEqual(2)
115    });
116
117    it("doesn't add one frame when the number of frames is 10", function(){
118      for (var i = 0; i <15; i++){
119        game.addFrame()
120      }
121      expect(game.getCurrentFrame()).toEqual(10)
122    });
123  });
124
125  describe("#isSpareBonus", function(){
126    it("is set to false by default", function(){
127      game.bonus.isSpare = jasmine.createSpy('spare == false').and.returnValue(false)
128      expect(game.isSpareBonus()).toEqual(false)
129    });
130
131    it("is set to true if all pins are knocked down on the second roll", function(){
132      game.bonus.isSpare = jasmine.createSpy('spare == true').and.returnValue(true)
133      RollSpare()
134      expect(game.isSpareBonus()).toEqual(true)
135    });
136
137    it("resets to false after completing bonus roll", function(){
138      game.bonus.isSpare = jasmine.createSpy('spare == true').and.returnValue(true)
139      RollSpare()
140      game.bonus.isSpare = jasmine.createSpy('spare == false').and.returnValue(false)
141      game.addRoll(4)
142      expect(game.isSpareBonus()).toEqual(false)
143    });
144
145    it("resets to false after completing a bonus roll that is a strike", function(){
146      game.bonus.isSpare = jasmine.createSpy('spare == true').and.returnValue(true)
147      RollSpare()
148      game.bonus.isSpare = jasmine.createSpy('spare == false').and.returnValue(false)
149      RollStrike()
150      expect(game.isSpareBonus()).toEqual(false)
151    });
152  });
153
154  describe("#addSpareToBonus", function(){
155    it("adds the value of the first roll after a spare to the previous frame's bonus slot", function(){
156      RollSpare()
157      game.addRoll(3)
158      expect(game.getCurrentScore()).toEqual(16)
159    });
160  });
161
162  describe("#isStrikeBonus", function(){
163    it("is true if 10 is rolled on first roll", function(){
164      RollStrike()
165      expect(game.isStrikeBonus()).toEqual(true)
166    });
167  });
168  //DO ONCE SPARE BONUS IMPLEMENTED
169  // describe("#addStrikeBonus", function(){
170  //   it("adds the roll to previous Frame's bonus if strikeBonus is true", function(){
171  //     game.addRoll(10)
172  //     game.addRoll(4)
173  //
174  //   });
175  // });
176  });
```

Tree: 856f6f6a25 ▾

**bowling-challenge** / [spec](#) / **SpecHelper.js**

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

16 lines (14 sloc) | 320 Bytes

```
1  beforeEach(function () {
2    jasmine.addMatchers({
3      toBePlaying: function () {
4        return {
5          compare: function (actual, expected) {
6            var player = actual;
7
8            return {
9              pass: player.currentlyPlayingSong === expected && player.isPlaying
10           };
11         }
12       };
13     }
14   });
15 });
```



Tree: 856f6f6a25 ▾

bowling-challenge / src / Bonus.js

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

33 lines (24 sloc) | 483 Bytes

```
1  function Bonus(){
2    this.spare = false
3    this.strike = false
4  }
5
6  // SETTERS
7
8  Bonus.prototype.setSpareTrue = function(){
9    this.spare = true
10 }
11
12 Bonus.prototype.setSpareFalse = function(){
13   this.spare = false
14 }
15
16 Bonus.prototype.setStrikeTrue = function(){
17   this.strike = true
18 }
19
20 Bonus.prototype.setStrikeFalse = function(){
21   this.strike = false
22 }
23
24 // BOOLEAN
25
26 Bonus.prototype.isSpare = function(){
27   return this.spare
28 }
29
30 Bonus.prototype.isStrike = function(){
31   return this.strike
32 }
```

Tree: 856f6f6a25 ▾

bowling-challenge / src / Frame.js

Find file

Copy path


Fetching contributors...

 Cannot retrieve contributors at this time

45 lines (34 sloc) | 660 Bytes

```
1  function Frame(){
2    this.bonus = 0
3    this.rolls = {
4      1: 0,
5      2: 0,
6      bonus: 0
7    }
8  }
9
10 // GETTERS
11
12 Frame.prototype.getRollOne = function(){
13   return this.rolls[1]
14 }
15
16 Frame.prototype.getRollTwo = function(){
17   return this.rolls[2]
18 }
19
20 Frame.prototype.getBonus = function(){
21   return this.rolls['bonus']
22 }
23
24 // SETTERS
25
26 Frame.prototype.setRollOne = function(num){
27   this.rolls[1] = num
28 }
29
30 Frame.prototype.setRollTwo = function(num){
31   this.rolls[2] = num
32 }
33
34 Frame.prototype.setBonus = function(num){
35   this.rolls['bonus'] = num
36 }
37
38 // METHODS
39
40 Frame.prototype.sumAllRolls = function(){
41   return this.getRollOne()
42     + this.getRollTwo()
43     + this.getBonus()
44 }
```

Fetching contributors...

 Cannot retrieve contributors at this time

109 lines (92 sloc) | 2.49 KB

```
1  function Game(){
2      this.currentFrame = 1
3      this.scores = [[],[],[],[],[],[],[],[],[],[],[]]
4      this.currentTurn = 1
5      this.isEndOfGame = false
6  }
7
8  // GETTERS
9
10
11  Game.prototype.getCurrentScore = function(){
12      return this._sumGame()
13  }
14
15  Game.prototype.getCurrentFrame = function(){
16      return this.currentFrame
17  }
18
19  Game.prototype.getCurrentTurn = function(){
20      return this.currentTurn
21  }
22
23  // ADD PINS
24
25
26  Game.prototype.addPins = function(pins){
27      if (this.currentTurn === 4) {
28          throw "The game is over"
29      }
30      if (pins > 10) {
31          throw "You can't knock down over 10 pins"
32      } else {
33          if (this.scores[this.currentFrame - 1][0] + pins > 10 && this.currentFrame < 10) {
34              throw "You can't knock down over 10 pins"
35          } else {
36              this.scores[this.currentFrame - 1].push(pins)
37              this._addStrikeBonus(pins)
38              this._addSpareBonus(pins)
39              this._setCurrentTurn(pins)
40          }
41      }
42  }
43
44  // MOVE TO NEXT FRAME
45
46  Game.prototype.moveToNextFrame = function(){
47      this.currentFrame += 1
48  }
49
50  Game.prototype._setCurrentTurn = function(pins){
51      if (this.getCurrentTurn() === 1 && pins < 10) {
52          this.currentTurn = 2
53      } else if (this.getCurrentFrame() === 10) {
54          if (this.currentTurn === 1){
55              this.currentTurn = 2
56          }else if (this.currentTurn === 2) {
57              this.currentTurn = 3
58          }else if (this.currentTurn === 3) {
59              this.currentTurn = 4
60          }
61      } else {
62          this.currentTurn = 1
63          this.moveToNextFrame()
64      }
65  }
66
67  // PRIVATE
68
69  Game.prototype._sumFrames = function(){
70      var frameTotals = []
71      this.scores.forEach(function(frame){
72          var score = 0
73          frame.forEach(function(bowl){
74              score += bowl
75          });
76          frameTotals.push(score);
77      });
78      return frameTotals
79  }
80
81  Game.prototype._sumGame = function(){
82      var total = 0
83      this._sumFrames().forEach(function(frame){
84          total += frame
85      });
86      return total
87  }
88
89  Game.prototype._addSpareBonus = function(pins){
90      if (this.currentFrame > 1 && this.currentTurn < 3) {
91          if (this.scores[this.currentFrame - 2][0] + this.scores[this.currentFrame - 2][1] === 10 && this.currentTurn === 1) {
92              this.scores[this.currentFrame - 2].push(pins)
93          }
94      }
95  }
96
97  Game.prototype._addStrikeBonus = function(pins){
98      if (this.currentFrame > 1 && this.currentTurn < 3) {
99          if (this.scores[this.currentFrame - 2][0] === 10) {
100              this.scores[this.currentFrame - 2].push(pins)
101          }
102      }
103      if (this.currentFrame > 2 && this.currentTurn < 2) {
104          if (this.scores[this.currentFrame - 3][0] === 10 && this.scores[this.currentFrame - 2][0] === 10) {
105              this.scores[this.currentFrame - 3].push(pins)
106          }
107      }
108  }
```

Fetching contributors...

 Cannot retrieve contributors at this time

36 lines (28 sloc) | 782 Bytes


```
1  $(document).ready(function(){
2      var game = new Game();
3      var turn = 1
4
5      function SumTotal(){
6          $('#total').html(game.getCurrentScore());
7      }
8
9      function SumPreviousFrame(){
10         if (game.currentTurn === 1) {
11             return game.getCurrentScore()
12         }
13     }
14
15     function TotalFrameTen(){
16         if (game.currentFrame === 10 && game.currentTurn !== 1) {
17             $('#frame-'+(game.currentFrame)).html(game.getCurrentScore())
18         }
19     }
20
21     $('#score-button').on('click', function(){
22         var score = parseInt($('#score').val())
23         $('#'+game.currentFrame+game.currentTurn).html(score);
24         game.addPins(score);
25         TotalFrameTen()
26         $('#frame-'+(game.currentFrame - 1)).html(SumPreviousFrame())
27         SumTotal();
28         console.log(game)
29         console.log(game.isEndOfGame)
30     });
31
32
33
34     SumTotal()
35 });
```

Tree856f6f6a25

bowling-challenge / src / OriginalGame.js

Find fileCopy path

Fetching contributors...

 Cannot retrieve contributors at this time

126 lines (104 sloc) | 2.55 KB

```
1 // NEXT - Implement the addition of spare bonus to previous frame's bonus
2
3 function Game(bonus){
4   this.rollOne = true
5   this.bonus = bonus
6   this.currentFrame = 1
7   this.frames = {
8     1: new Frame(),
9     2: new Frame(),
10    3: new Frame(),
11    4: new Frame(),
12    5: new Frame(),
13    6: new Frame(),
14    7: new Frame(),
15    8: new Frame(),
16    9: new Frame(),
17    10: new Frame(),
18  }
19 }
20
21 // GETTERS
22
23 Game.prototype.getCurrentScore = function () {
24   var score = 0
25   for (var key in this.frames) {
26     if (this.frames.hasOwnProperty(key)) {
27       score += this.frames[key].sumAllRolls()
28     }
29   }
30   return score;
31 }
32
33 Game.prototype.getCurrentFrame = function () {
34   return this.currentFrame
35 }
36
37 // ADD SCORE
38
39 Game.prototype.addRoll = function (roll) {
40   if (roll > 10) {
41     throw("You can't knock down more than 10 pins")
42   }
43   if (this.currentFrame <= 10) {
44     if (this.isRollOne()) {
45       this._saveRollOne(roll)
46     } else {
47       this._saveRollTwo(roll)
48     }
49   }
50   // This shouldnt be here as is setting strike bonus regardless of whether 10 was rolled on first or second go
51   // if (roll == 10) {
52   //   this.setStrikeBonus()
53   // }
54 };
55
56
57 // BOOLEAN
58
59 Game.prototype.isRollOne = function(){
60   return this.rollOne
61 }
62
63 Game.prototype.isStrikeBonus = function(){
64   return this.bonus.isStrike()
65 }
66
67 Game.prototype.isSpareBonus = function () {
68   return this.bonus.isSpare()
69 };
70
71 // PRIVATE
72
73 Game.prototype.setSpareBonusTrue = function(){
74   this.bonus.setSpareTrue()
75 }
76
77 Game.prototype.setSpareBonusFalse = function(){
78   this.bonus.setSpareFalse()
79 }
80
81 Game.prototype.setStrikeBonusTrue = function(){
82   this.bonus.setStrikeTrue()
83 }
84
85 Game.prototype.setStrikeBonusFalse = function(){
86   this.bonus.setStrikeFalse()
87 }
88
89 Game.prototype.addFrame = function() {
90   if (this.currentFrame < 10){
91     this.currentFrame += 1
92   }
93 }
94
95 Game.prototype._setRollOneFalse = function(){
96   this.rollOne = false
97 }
98
99 Game.prototype._setRollOneTrue = function(){
100   this.rollOne = true
101 }
102
103 Game.prototype._saveRollOne = function(roll){
104   this.frames[this.currentFrame].setRollOne(roll)
105   this.setSpareBonusFalse()
106   if (roll == 10) {
107     this.setStrikeBonusTrue()
108     // think it should be here - LINE 48
109     this.addFrame()
110   } else {
111     this._setRollOneFalse()
112   }
113 }
114
115 Game.prototype._saveRollTwo = function(roll){
116   if (roll + this.frames[this.currentFrame].getRollOne() > 10) {
117     throw "Your rolls can't sum over 10"
118   }
119   if (roll + this.frames[this.currentFrame].getRollOne() == 10) {
120     this.setSpareBonusTrue()
121   }
122   this.frames[this.currentFrame].setRollTwo(roll)
123   this.addFrame()
124   this._setRollOneTrue()
125 }
```