

Fetching contributors...

 Cannot retrieve contributors at this time

26 lines (19 sloc) | 667 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/bowling.js"></script>
16
17   <!-- include spec files here... -->
18   <script src="spec/SpecHelper.js"></script>
19   <script src="spec/bowlingSpec.js"></script>
20
21 </head>
22
23 <body>
24 </body>
25 </html>
```

Tree: a24e53a8da ▼

bowling-challenge / [public](#) / **index.html**

Find file

Copy path

Fetching contributors...



Cannot retrieve contributors at this time

11 lines (7 sloc) | 102 Bytes

```
1  <html>
2    <head>
3      <script src='../src/bowling.js'></script>
4    </head>
5
6    <body>
7
8    </body>
9
10 </html>
```

Tree: a24e53a8da ▾

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 });
```


Fetching contributors...

 Cannot retrieve contributors at this time

91 lines (80 sloc) | 2.41 KB

```
1 describe('Bowling', function(){
2   var bowling;
3
4   beforeEach( function(){
5     bowling = new Bowling();
6   });
7
8
9   describe('initialize', function(){
10     it('has starting score of zero', function(){
11       expect(bowling.total()).toEqual(0);
12     });
13
14     it('has a maximum score of 300', function(){
15       expect(bowling.maxScore()).toEqual(300);
16     });
17   });
18
19   describe('add score', function(){
20     it('should add 10 to the total', function(){
21       expect(bowling.addScore(10)).toEqual(10);
22     });
23
24     it('cannot add more than 10 to score at a time', function(){
25       expect(bowling.addScore(11)).toEqual('Cannot add more than 10');
26     });
27   });
28
29   describe('frame', function(){
30     it('adds to counter if less than 2', function(){
31       bowling.frame(10);
32       bowling.frame(10);
33       expect(bowling._counter).toEqual(2);
34     });
35     it('return counter to 0 if played more than twice', function(){
36       bowling.frame(10);
37       bowling.frame(10);
38       bowling.frame(10);
39       expect(bowling._counter).toEqual(0);
40     });
41     it('adds score if counter less than 2', function(){
42       bowling.frame(10);
43       expect(bowling._currentScore).toEqual(10);
44     });
45   });
46
47   describe('game', function(){
48     it('adds 10 to total', function(){
49       bowling._counter = 0;
50       bowling.game(10);
51       expect(bowling._total).toEqual(10);
52     });
53     it('adds 1 to frame if set complete', function(){
54       bowling.game(10);
55       expect(bowling._frame).toEqual(1);
56     });
57     it('counter does not change if strike', function(){
58       bowling.game(10);
59       expect(bowling._counter).toEqual(0);
60     });
61     it('resets counter if 2 rolls',function(){
62       bowling.game(4);
63       bowling.game(3);
64       expect(bowling._counter).toEqual(0);
65     });
66     it('adds score to current score if first roll', function(){
67       bowling.game(4);
68       expect(bowling._currentScore).toEqual(4);
69     })
70     it('adds current score to total if second roll', function(){
71       bowling.game(4);
72       bowling.game(3);
73       expect(bowling._total).toEqual(7);
74     });
75     it('cannot play more than 12 frames', function(){
76       for(var i = 0; i<12; i++){
77         bowling.game(10);
78       }
79       expect(bowling.game(10)).toEqual('Cannot play more than 12 frames');
80     });
81   })
82
83   // describe('current score', function(){
84   //   it('adds score to current score', function(){
85   //     bowling.currentScore(10);
86   //     expect(bowling._currentScore).toEqual(10);
87   //   });
88   // });
89
90 });
```

Fetching contributors...

 Cannot retrieve contributors at this time

67 lines (59 sloc) | 1.43 KB

```
1  const MAXFRAME = 12;
2
3  function Bowling(){
4    this._total = 0;
5    this._maxScore = 300;
6    this._counter = 0;
7    this._currentScore = 0;
8    this._frame = 0;
9    this._bonus = 0;
10 }
11
12 Bowling.prototype.total = function(){
13   return this._total;
14 };
15
16 Bowling.prototype.maxScore = function(){
17   return this._maxScore;
18 }
19
20 Bowling.prototype.addScore = function(score){
21   if(score > 10){
22     return 'Cannot add more than 10';
23   }
24   return this._total += score;
25 }
26
27 Bowling.prototype.frame = function(score){
28   if (this._counter < 2){
29     this._currentScore += score;
30     this._counter += 1;
31   } else {
32     this._frame += 1;
33     this._counter = 0;
34   }
35 }
36
37
38 Bowling.prototype.game = function(score){
39   if(this._frame < MAXFRAME) {
40     if(this._counter === 0 && score === 10){
41       // strike
42       this._total += score;
43       this._counter = 0;
44       this._frame += 1;
45     } else if(this._counter === 1 && (this._currentscore + score === 10) ){
46       //spare
47       this._total += this._currentScore;
48       this._currentScore = 0;
49       this._counter += 1;
50       this._frame += 1;
51     } else if(this._counter == 0){
52       //normal first roll
53       this._currentScore += score;
54       this._counter += 1;
55     } else if(this._counter === 1){
56       //normal second roll
57       this._currentScore += score;
58       this._total += this._currentScore;
59       this._currentScore = 0;
60       this._counter = 0;
61       this._frame += 1;
62     }
63   } else {
64     return 'Cannot play more than 12 frames';
65   }
66 }
```