Tree: **248a7c7635** ▾　**bowling-challenge** / **SpecRunner.html**　　　　Find file　Copy path

Fetching contributors…

⊘ Cannot retrieve contributors at this time

26 lines (19 sloc) | 674 Bytes

```html
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
17    <!-- include spec files here... -->
18    <script src="spec/GameSpec.js"></script>
19      <!-- <script src="spec/web_spec.js"></script> -->
20
21  </head>
22
23  <body>
24  </body>
25  </html>
```

66 lines (55 sloc) │ 1.95 KB

```javascript
describe('Bowling', function(){
  var game;

  beforeEach(function(){
    game = new Game();
    game.addFrame();
  });

  describe('Start a new game', function(){
    it('starts a game with a frame and turn', function(){
      expect(game.currentGame()).toEqual({frame: 1, round: 0, score: 0})
    });
  });

  describe('Adds score to the frame', function(){
    it('rolls a five in the first round, changes the round', function(){
      game.roll(5);
      expect(game.currentGame()).toEqual({frame: 1, round: 1, score: 5})
    });
  });

  describe('After two rounds, points are added up', function(){
    it('rolls a 3 in th second round, points are added up', function(){
      game.roll(5);
      game.roll(3);
      expect(game.rounds[0]).toEqual({frame: 1, round: 2, score: 8})
    });
  });

  describe('You can play a gutter game', function(){
    it('if player really sucks and cannot bowl for his life', function(){
      for (var i = 0; i < 29; i++){
        game.roll(0)
      }
      expect(game.currentGame()).toEqual({frame: 10, round: 2, score: 0})
    });
  });


  describe('When turn is 2, you then will be added to another frame', function(){
    it('When you have past one frame and in the second frame, after one turn', function(){
      game.roll(5);
      game.roll(3);
      game.roll(4);
      expect(game.currentGame()).toEqual({frame: 2, round: 0, score: 12})
    });
  });

  describe('If one roll is a strike(ten), turn is skipped', function(){
    it('a strike is rolled', function(){
      game.roll(10);
      expect(game.rounds[0]).toEqual({frame: 1, round: 2, score: 10})
      expect(game.currentGame()).toEqual({frame: 2, round: 1, score: 20})
    });
  });

  // describe('If one roll is a strike(ten), frame is changed with plus ten from a strike', function(){
  //   it('a strike is rolled', function(){
  //     game.roll(10);
  //     game.roll(1);
  //     expect(game.currentGame()).toEqual({frame: 2, round: 2, score: 22})
  //   });
  // });

});
```

Tree: **248a7c7635 ⌄** | **bowling-challenge** / spec / **web_spec.js**

Find file | Copy path

Fetching contributors…

Cannot retrieve contributors at this time

0 lines (0 sloc) | 0 Bytes

Fetching contributors…

⊘ Cannot retrieve contributors at this time

58 lines (51 sloc) | 1.16 KB

```javascript
function Game() {
  this.rounds = [];
  this.frame = 0
  this.currentround = 0
  this.currentpoints = 0
  this.strikebonus = 0
  this.change = 0
  this.game = {}
};

Game.prototype.addFrame = function(){
  this.frame += 1
  this.currentround = 0
  this.rounds.push({frame: this.frame, round: this.currentround, score: this.currentpoints})
};

Game.prototype.roll = function(points){
  this.currentpoints += points;
  this.currentGame().score = this.currentpoints
  this.strike()
  this.checkRounds()
  console.log(this.rounds)
};

Game.prototype.currentGame = function(){
  return this.rounds[this.frame - 1];
};

Game.prototype.checkRounds = function(){
  if(this.currentGame().round > 2){
      return false }
  else if(this.currentGame().round === 2){
    this.addFrame()
    } else {
      this.addRounds()
    }
};

Game.prototype.addRounds = function(){
  if(this.currentGame().score >= 0){
      this.currentGame().round += 1
    }
  else{
      return false
    }
};

Game.prototype.strike = function(){
  if(this.currentGame().score === 10){
      this.currentGame().round = 2
      this.addFrame()
      this.currentGame().score += 10
    }
  else{
      return false
    }
};
```