makersacademy / **bowling-challenge**

Tree: a24e53a8da ▾

bowling-challenge / **SpecRunner.html**

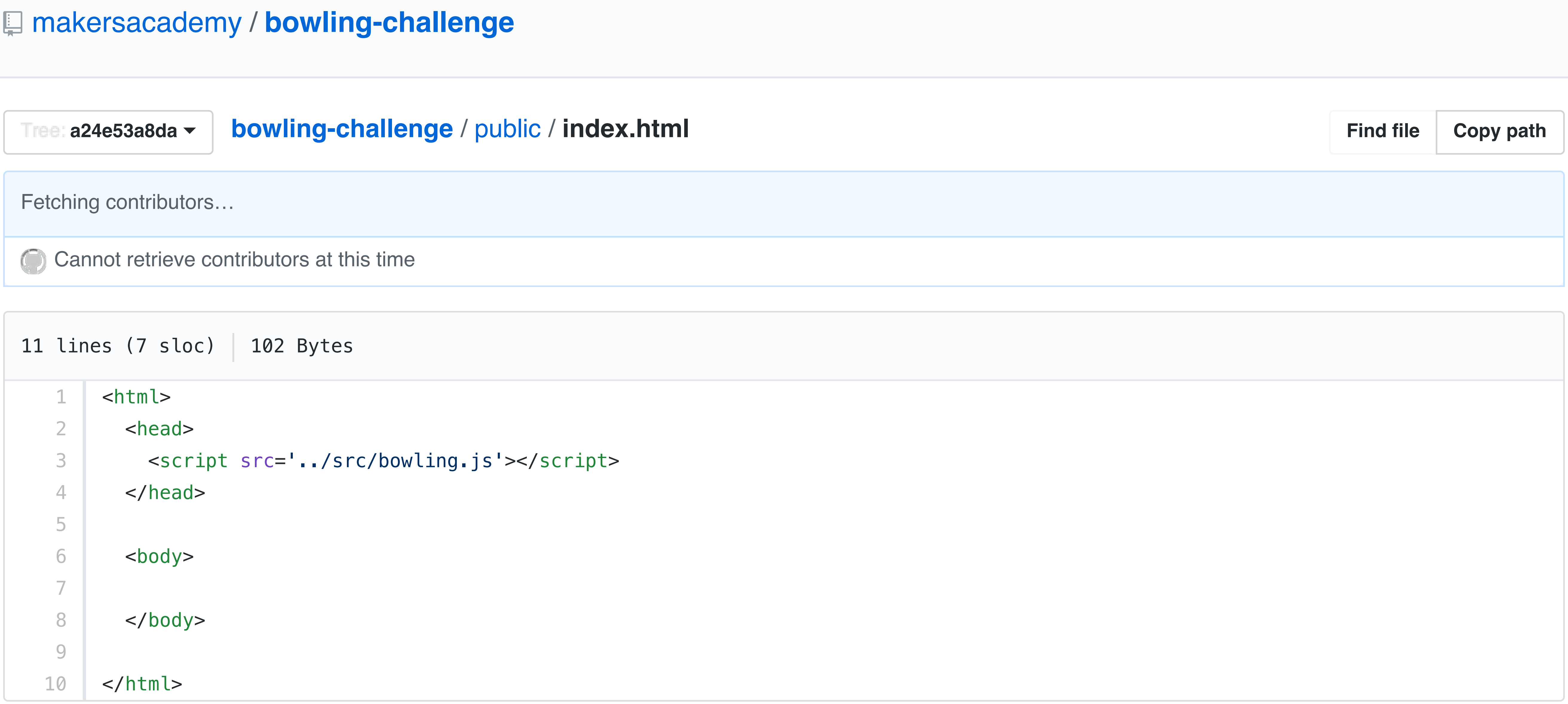
Find file | Copy path

Fetching contributors…

Cannot retrieve contributors at this time

26 lines (19 sloc) | 667 Bytes

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Jasmine Spec Runner v2.8.0</title>

  <link rel="shortcut icon" type="image/png" href="lib/jasmine-2.8.0/jasmine_favicon.png">
  <link rel="stylesheet" href="lib/jasmine-2.8.0/jasmine.css">

  <script src="lib/jasmine-2.8.0/jasmine.js"></script>
  <script src="lib/jasmine-2.8.0/jasmine-html.js"></script>
  <script src="lib/jasmine-2.8.0/boot.js"></script>

  <!-- include source files here... -->
  <script src="src/bowling.js"></script>

  <!-- include spec files here... -->
  <script src="spec/SpecHelper.js"></script>
  <script src="spec/bowlingSpec.js"></script>

</head>

<body>
</body>
</html>
```



makersacademy / **bowling-challenge**

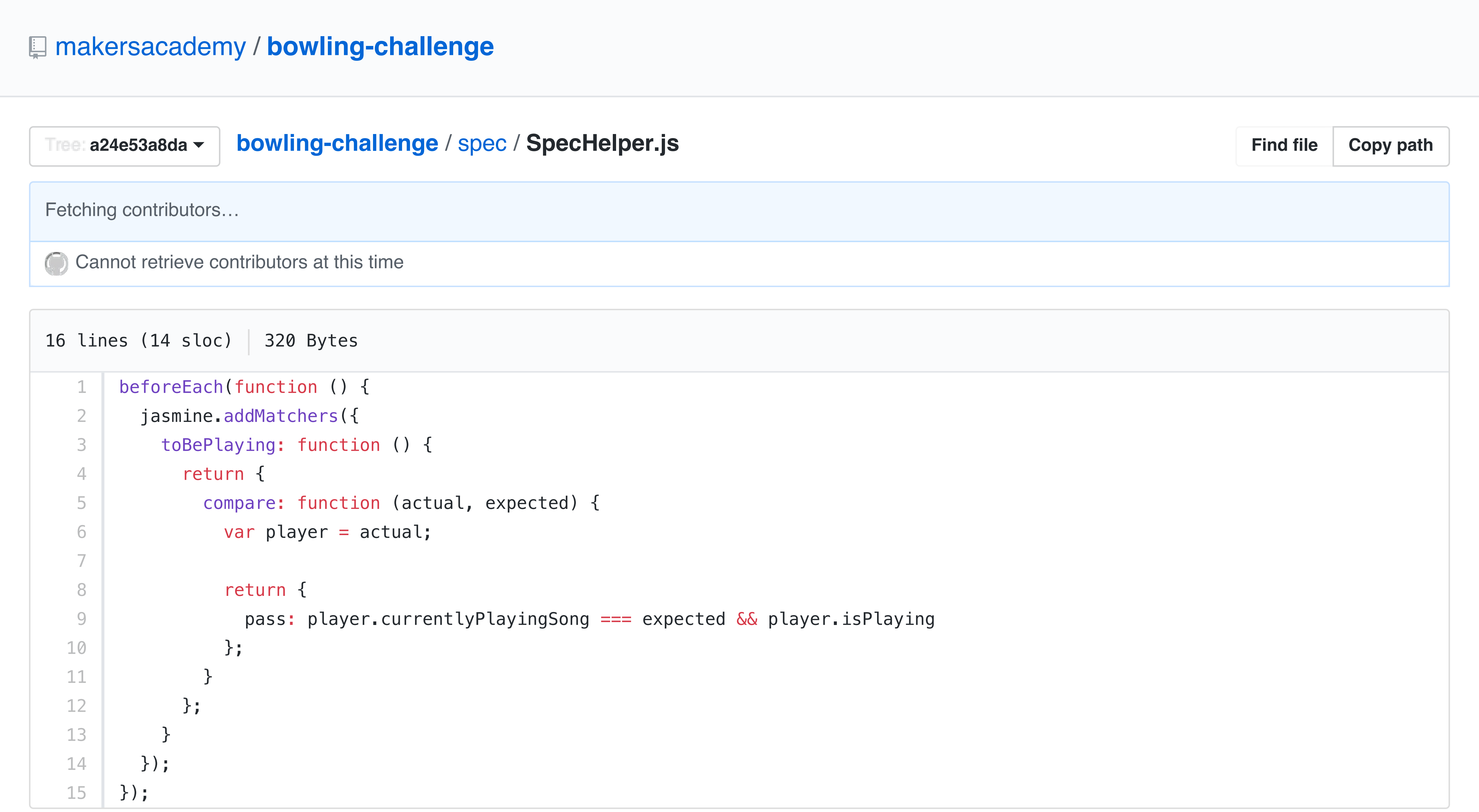
bowling-challenge / public / **index.html**

Fetching contributors…

Cannot retrieve contributors at this time

11 lines (7 sloc) | 102 Bytes

```
<html>
  <head>
    <script src='../src/bowling.js'></script>
  </head>

  <body>

  </body>

</html>
```



makersacademy / bowling-challenge

bowling-challenge / spec / SpecHelper.js

Fetching contributors…

Cannot retrieve contributors at this time

16 lines (14 sloc) | 320 Bytes

```
beforeEach(function () {
  jasmine.addMatchers({
    toBePlaying: function () {
      return {
        compare: function (actual, expected) {
          var player = actual;

          return {
            pass: player.currentlyPlayingSong === expected && player.isPlaying
          };
        }
      };
    }
  });
});
```

makersacademy / **bowling-challenge**

Tree: a24e53a8da ▾ **bowling-challenge** / spec / **bowlingSpec.js** Find file Copy path

Fetching contributors…

Cannot retrieve contributors at this time

91 lines (80 sloc) | 2.41 KB

```
describe('Bowling', function(){
  var bowling;

  beforeEach( function(){
    bowling = new Bowling();
  });


  describe('initialize', function(){
    it('has starting score of zero', function(){
      expect(bowling.total()).toEqual(0);
    });

    it('has a maximum score of 300', function(){
      expect(bowling.maxScore()).toEqual(300);
    });
  });

  describe('add score', function(){
    it('should add 10 to the total', function(){
      expect(bowling.addScore(10)).toEqual(10);
    });

    it('cannot add more than 10 to score at a time', function(){
      expect(bowling.addScore(11)).toEqual('Cannot add more than 10');
    });
  });

  describe('frame', function(){
    it('adds to counter if less than 2', function(){
      bowling.frame(10);
      bowling.frame(10);
      expect(bowling._counter).toEqual(2);
    });
    it('return counter to 0 if played more than twice', function(){
      bowling.frame(10);
      bowling.frame(10);
      bowling.frame(10);
      expect(bowling._counter).toEqual(0);
    });
    it('adds score if counter less than 2', function(){
      bowling.frame(10);
      expect(bowling._currentScore).toEqual(10);
    });
  });

  describe('game', function(){
    it('adds 10 to total', function(){
      bowling._counter = 0;
      bowling.game(10);
      expect(bowling._total).toEqual(10);
    });
    it('adds 1 to frame if set complete', function(){
      bowling.game(10);
      expect(bowling._frame).toEqual(1);
    });
    it('counter does not change if strike', function(){
      bowling.game(10);
      expect(bowling._counter).toEqual(0);
    });
    it('resets counter if 2 rolls',function(){
      bowling.game(4);
      bowling.game(3);
      expect(bowling._counter).toEqual(0);
    });
    it('adds score to current score if first roll', function(){
      bowling.game(4);
      expect(bowling._currentScore).toEqual(4);
    })
    it('adds current score to total if second roll', function(){
      bowling.game(4);
      bowling.game(3);
      expect(bowling._total).toEqual(7);
    });
    it('cannot play more than 12 frames', function(){
      for(var i = 0; i<12; i++){
        bowling.game(10);
      }
      expect(bowling.game(10)).toEqual('Cannot play more than 12 frames');
    });
  })

  // describe('current score', function(){
  //   it('adds score to current score', function(){
  //     bowling.currentScore(10);
  //     expect(bowling._currentScore).toEqual(10);
  //   });
  // });

});
```

makersacademy / **bowling-challenge**

Tree: a24e53a8da ▾ **bowling-challenge** / src / **bowling.js**

Find file | Copy path

Fetching contributors…

Cannot retrieve contributors at this time

67 lines (59 sloc) | 1.43 KB

```javascript
const MAXFRAME = 12;

function Bowling(){
  this._total = 0;
  this._maxScore = 300;
  this._counter = 0;
  this._currentScore = 0;
  this._frame = 0;
  this._bonus = 0;
}

Bowling.prototype.total = function(){
    return this._total;
};

Bowling.prototype.maxScore = function(){
  return this._maxScore;
}

Bowling.prototype.addScore = function(score){
  if(score > 10){
    return 'Cannot add more than 10';
  }
  return this._total += score;
}

Bowling.prototype.frame = function(score){
  if (this._counter < 2){
    this._currentScore += score;
    this._counter += 1;
  } else {
    this._frame += 1;
    this._counter = 0;
  }
}


Bowling.prototype.game = function(score){
  if(this._frame < MAXFRAME) {
    if(this._counter === 0 && score === 10){
      // strike
      this._total += score;
      this._counter = 0;
      this._frame += 1;
    } else if(this._counter === 1 && (this._currentscore + score === 10) ){
      //spare
      this._total += this._currentScore;
      this._currentScore = 0;
      this._counter += 1;
      this._frame += 1;
    } else if(this._counter == 0){
      //normal first roll
      this._currentScore += score;
      this._counter += 1;
    } else if(this._counter === 1){
      //normal second roll
      this._currentScore += score;
      this._total += this._currentScore;
      this._currentScore = 0;
      this._counter = 0;
      this._frame += 1;
    }
  } else {
    return 'Cannot play more than 12 frames';
  }
}
```