Wed Apr 08 02:03:53 2015 bin/standalone.rb 1: require\_relative '../lib/crackle\_pop'
2: puts CracklePop.new.to\_s

```
Guardfile
              Wed Apr 08 01:17:58 2015
   1: # A sample Guardfile
   2: # More info at https://github.com/guard/guard#readme
   3:
   4: guard :minitest do
   5: # with Minitest::Unit
   6: watch(%r{^test/(.*)\/?test_(.*)\.rb$})
   7: watch(%r{^lib/(.*/)?([^/]+)\.rb$}) { |m| "test/#{m[1]}test_#{m[2]}.rb" }
   8: watch(%r{^test/test_helper\.rb$}) { 'test' }
   9:
  10: # with Minitest::Spec
  11: watch(%r{^spec/(.*)_spec\.rb$})
  12: watch(%r{^lib/(.+)\.rb$}) { |m| "spec/#{m[1]}_spec.rb" }
  13: watch(%r{^spec/spec_helper\.rb$}) { 'spec' }
  14:
  15: end
```

lib/crackle\_pop.rb Wed Apr 08 23:31:06 2015

```
1: class CracklePop
 2: def initialize(rule = { 3 => 'Crackle', 5 => 'Pop' }, rules: rule)
3: @rules = rules
4: end
5:
 6: # to_s takes an optional argument of a range otherwise simply plays the game
 7: # from 1 to 100
 8: def to_s(range = (1..100))
9: range.map { |n| what_is(n) }.join("\n")
10: end
11:
12: def what_is(integer)
13: result = @rules.each_key.map do |k|
14:
     (integer % k).zero? ? @rules[k] : ''
15: end.join
16: result.empty? && integer || result
17: end
18: end
```

```
spec/crackle_pop_spec.rb
                               Wed Apr 08 23:29:53 2015
```

```
1: require 'minitest/autorun'
              2: require_relative 'spec_helper'
              3: require 'crackle_pop'
              5: describe 'CracklePop' do
              6: before :each do
              7:
                              @cp = CracklePop.new
              8:
                                end
              9:
           10:
                                it 'must return "Crackle" for numbers divisible by 3' do
          11:
                                  @cp.what_is(3).must_equal 'Crackle'
         12:
         13:
         14:
                               it 'must return "Pop" for numbers divisible by 5' do
         15:
                                  @cp.what_is(5).must_equal 'Pop'
         16:
         17:
         18:
                               it 'must return "CracklePop" for 15' do
                                @cp.what_is(15).must_equal 'CracklePop'
         19:
         20:
         21:
         22: it 'must return 2 for 2' do
          23:
                               @cp.what_is(2).must_equal 2
         24: end
         25:
         26: it 'must output 1 - 100 according to rules' do
                                -> { puts @cp.to_s }.must_output "1\n2\nCrackle\n4\nPop\nCrackle\n7\n8\nCrackle\nPop\n11\nCrackle\n13\n14\nCracklePop\n16\n17\nCrackle\n19\nPop\nCrackle\n22\n23\nCrackle\nPop\n26\
         27:
nCrackle n28 n29 nCrackle n34 nPop nCrackle n34 nPop nCrackle n37 n38 nCrackle n43 n44 nCrackle n49 nPop nCrackle n52 n53 nCrackle nPop n56 nCrackle nPop n46 n47 nCrackle n49 nPop nCrackle n52 n53 nCrackle nPop n56 nCrackle nP
58 \times 10^{1} \times 10^{1
acklePop\n91\n92\nCrackle\n94\nPop\nCrackle\n97\n98\nCrackle\nPop\n"
         28: end
         29:
         30: it 'must output -15 - 15 according to rules' do
                                 -> { puts @cp.to_s(-15..15) }.must_output "CracklePop\n-14\n-13\nCrackle\n-11\nPop\nCrackle\n-8\n-7\nCrackle\nPop\n-4\nCrackle\n-2\n-1\nCracklePop\n1\n2\nCrackle\n4\nPop\nCrackle\n
n7\n8\nCrackle\nPop\n11\nCrackle\n13\n14\nCracklePop\n"
          32: end
         33:
         34: it 'must perform custom rules' do
         35: -> { puts CracklePop.new( 7 => 'Snap').to_s(5..8) }.must_output "5\n6\nSnap\n8\n"
          36: end
         37: end
```

spec/spec\_helper.rb Wed Apr 08 01:31:16 2015 1: %w[bin lib test].each do |path|
2: \$:.unshift('./' + path) 3: **end**