

-3 -

- to do list
 - to do list
- GridTest Acceptance Test . Acceptance Test .
- Acceptance Test pseudo . . .
- - rubymine . . .
 - eclipse . . .
- - step by step . . .
- - pass . . .
 - . . .
 - . . .
 - . . .

- square

```
def near_squares(x, y)
  arr = Array.new

  (x-1..x+1).each do |row|
    (y-1..y+1).each do |col|
      arr << get_square(row, col) unless out_of_index?(row, col)
    end
  end

  return arr
end
```

- square 0 open . . near_squares . .
- .near_squares for . . .

```
def open (x, y)
  raise IndexError.new if out_of_index?(x, y)
  raise GameOverError.new if get_square(x, y).mined

  return if get_square(x, y).opened
  get_square(x, y).open
  return unless get_square(x, y).zero_near_mine_num

  (x-1..x+1).each do |row|
    (y-1..y+1).each do |col|
      open(row, col) unless out_of_index?(row, col)
    end
  end
end
```

- pass . . .
- pass . . . for , . . code block()
- . . .

```

def open (x, y)
  raise IndexError.new if out_of_index?(x, y)
  raise GameOverError.new if get_square(x, y).mined
  return if get_square(x, y).opened
  get_square(x, y).open
  return unless get_square(x, y).zero_near_mine_num

  near_squares_each(x, y) do |row, col|
    open(row, col) unless out_of_index?(row, col)
  end
end

def near_squares(x, y)
  arr = Array.new

  near_squares_each(x, y) do |row, col|
    arr << get_square(row, col) unless out_of_index?(row, col)
  end

  return arr
end

def near_squares_each(x, y)
  (x-1..x+1).each do |row|
    (y-1..y+1).each do |col|
      yield row, col
    end
  end
end

```

- .

,

- .
- .
- code block , closure .
- .
- .