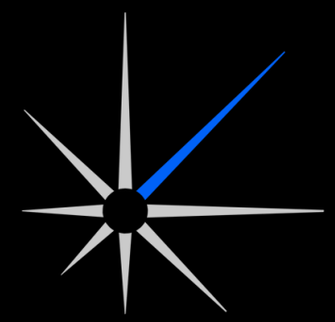
Test Driven Development in Ruby

Micah Martin
Software Craftsman



Ruby

- Powerful! but....
- Can be Dangerous

Dangers of Ruby

- No Compiler
- No Type Safety
- Unchecked Definitions

Runtime Errors

```
my_variable = 7
puts my_variable + 1
puts my_varaible + 2
```

Runtime Errors

```
my_variable = 7
puts my_variable + 1
puts my_varaible + 2
```

```
8
tmp.rb:4: undefined local
variable or method `my_varaible'
for main:Object (NameError)
```

No Type Safety

```
def increment(value)
  return value + 1
end

puts increment(6)
puts increment("six")
```

No Type Safety

```
def increment(value)
  return value + 1
end

puts increment(6)
puts increment("six")
```

```
7
tmp.rb:2:in `+': can't convert Fixnum into
String (TypeError)
   from tmp.rb:2:in `increment'
   from tmp.rb:6
```

Unchecked Definitions

```
class Foo
end

puts Foo.new
puts Bar.new
```

Unchecked Definitions

```
class Foo
end

puts Foo.new
puts Bar.new
```

```
#<Foo:0x28460>
tmp.rb:6: uninitialized constant
Bar (NameError)
```

Runtime Errors

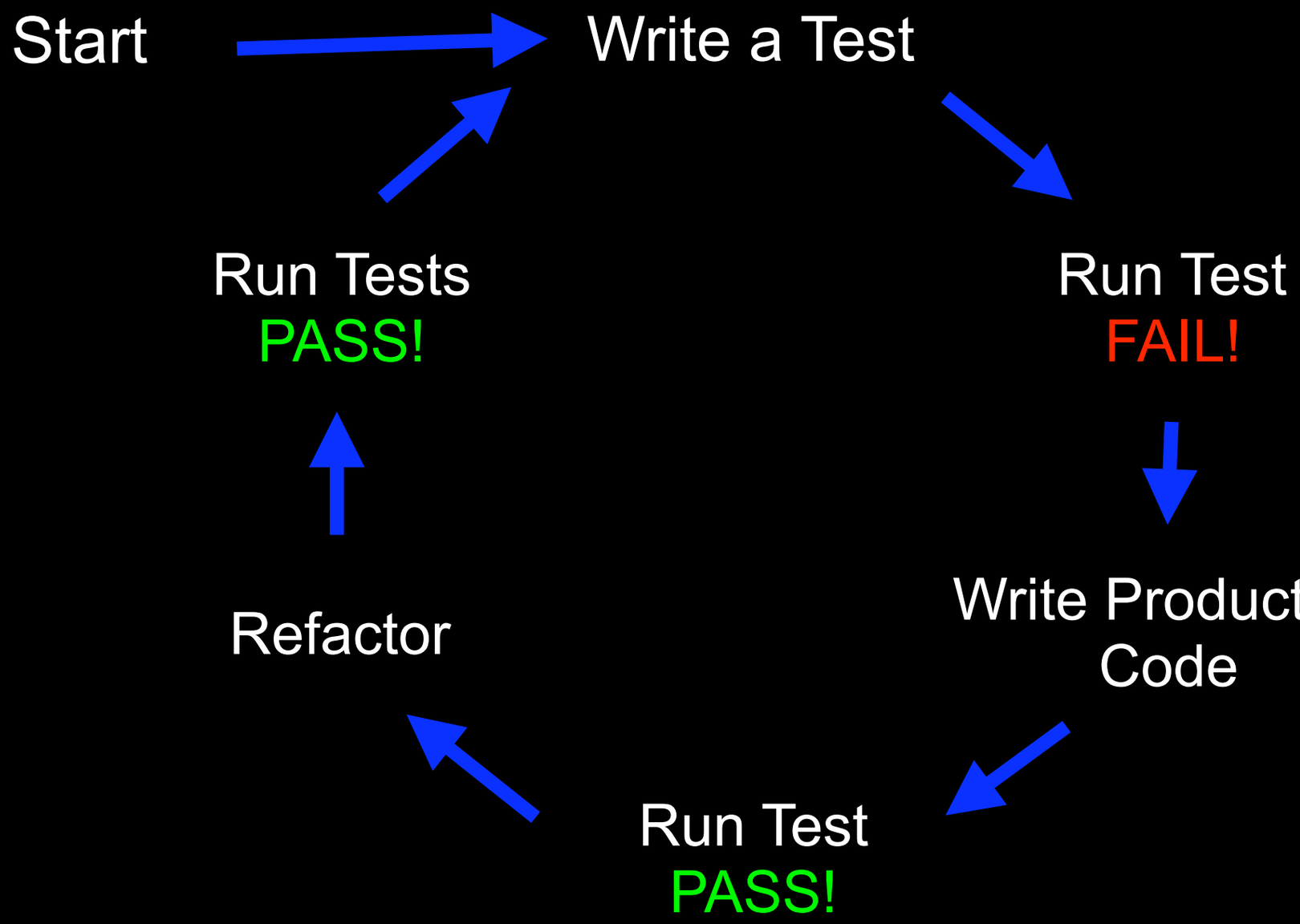
- Errors Appear at Runtime
- Changing Code is Risky
- Solution:
  - Execute Code As You Write It.
  - Unit Tests!

2 Rules of TDD

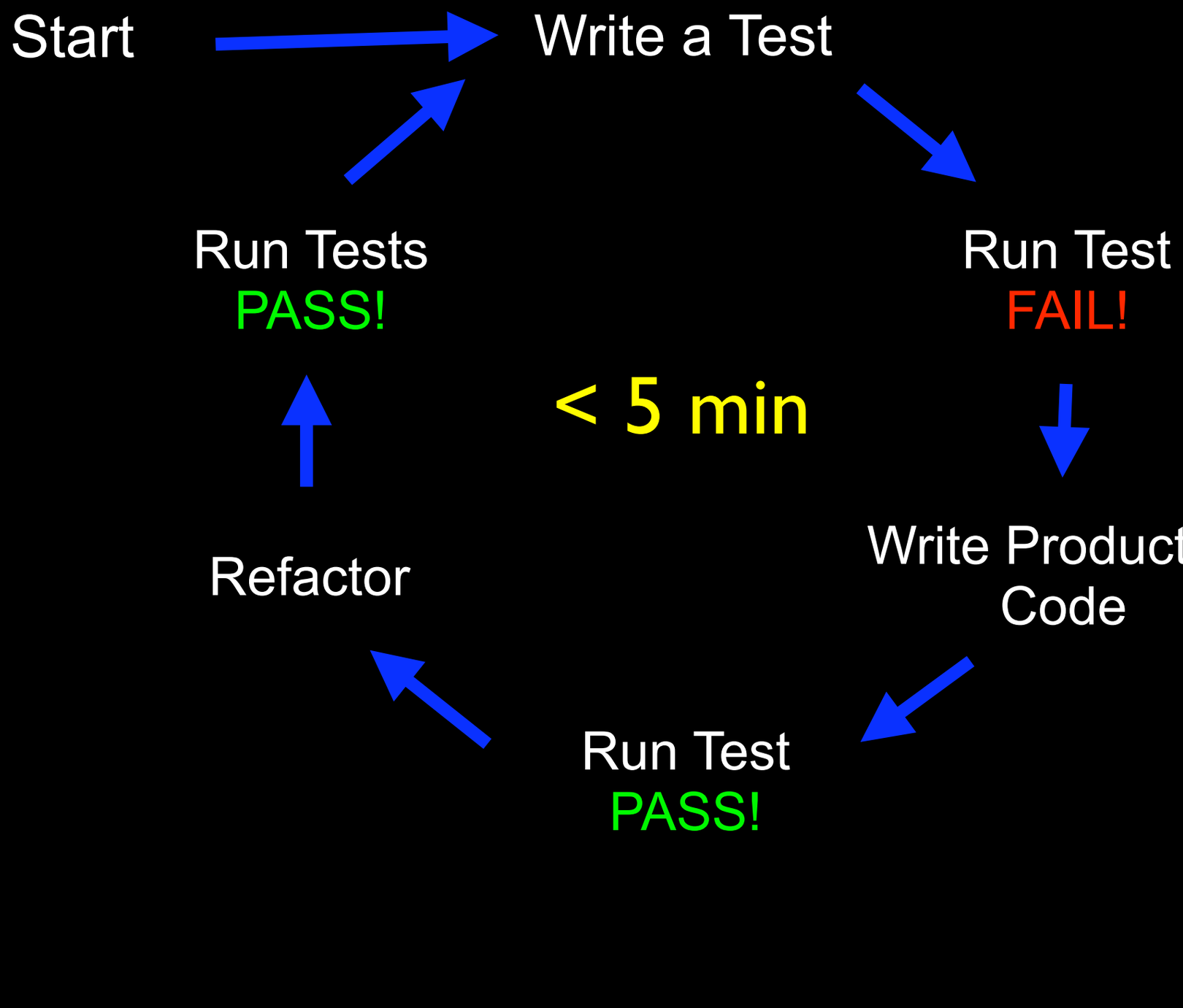
(Beck)

1. Do not write a line of production code until you have a failing unit test
2. Refactor all duplicate code

TDD Cycle


Start
Write a Test
Run Test
FAIL!
Write Production Code
Run Test
PASS!
Refactor
Run Tests
PASS!

TDD Cycle

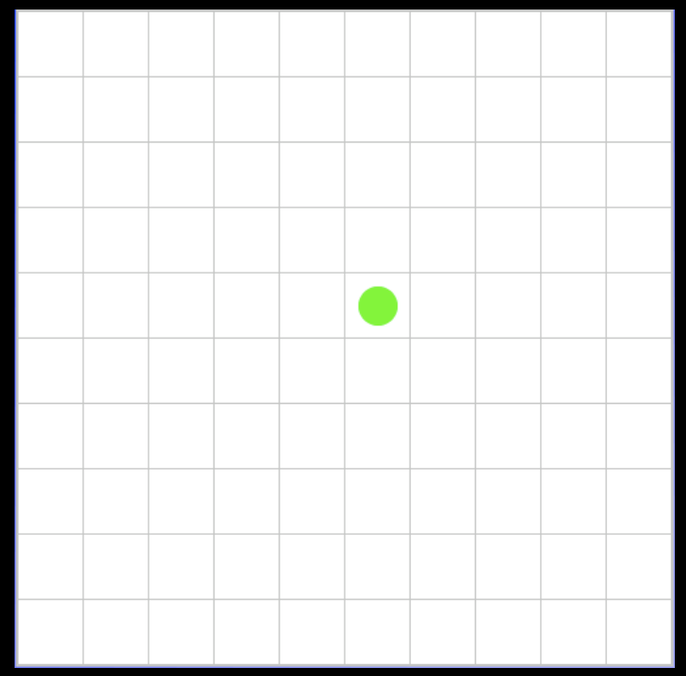

Start
Write a Test
Run Test
FAIL!
Write Production Code
Run Test
PASS!
Refactor
Run Tests
PASS!
< 5 min

Behavior Driven Development

- RSpec
- A Change in Language
- ...A Change in Thinking
- “TDD Done Right”

Demo

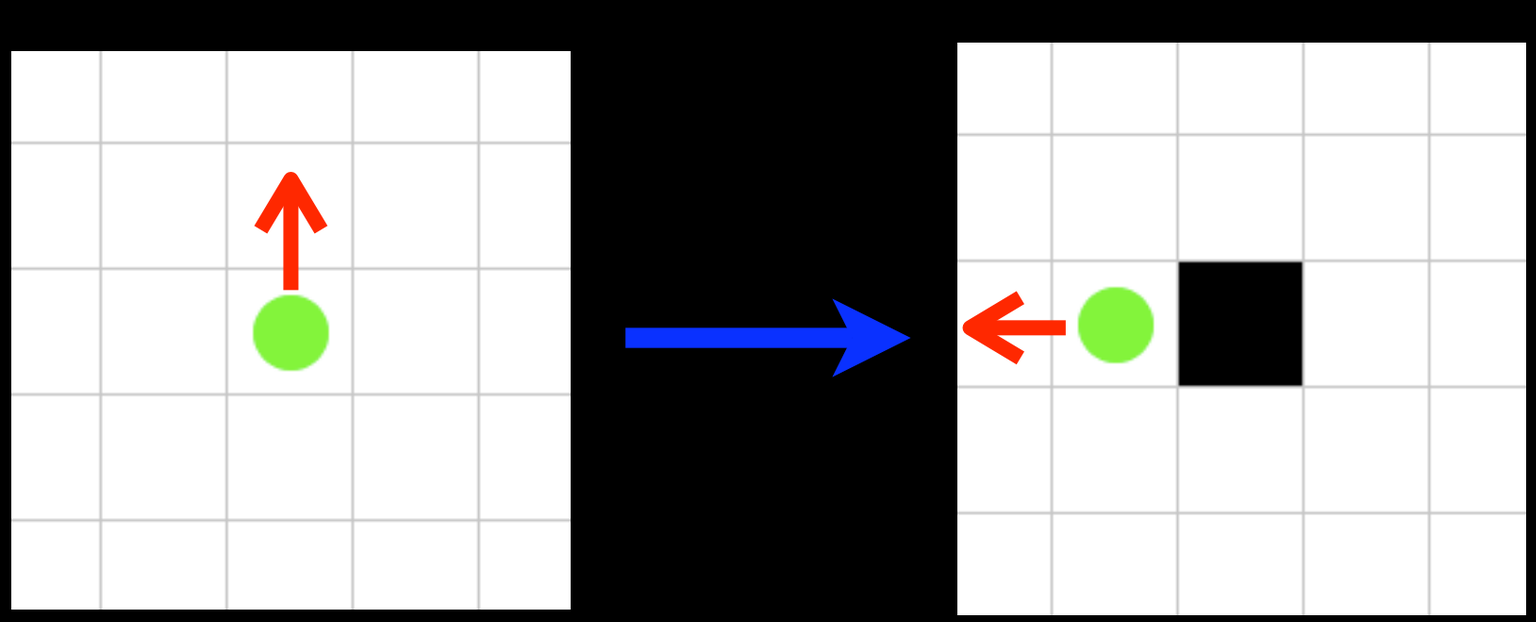
Langston's Ant



- A Classic Simulation
- Ant lives in 2D world.
- 2 Simple Rules
  - If it's on a Black square, paint it White and go Right
  - If it's on a White square, paint it Black and go Left

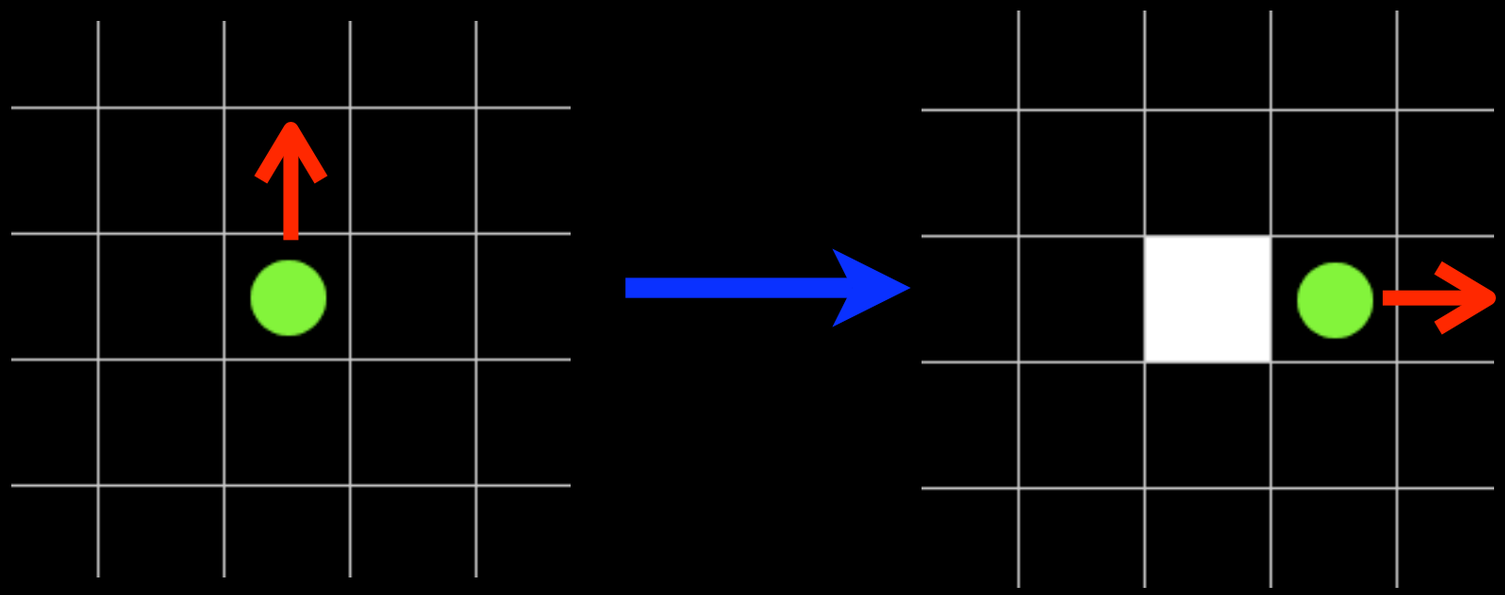
White Square

If it's on a White square,
paint it Black and go Left

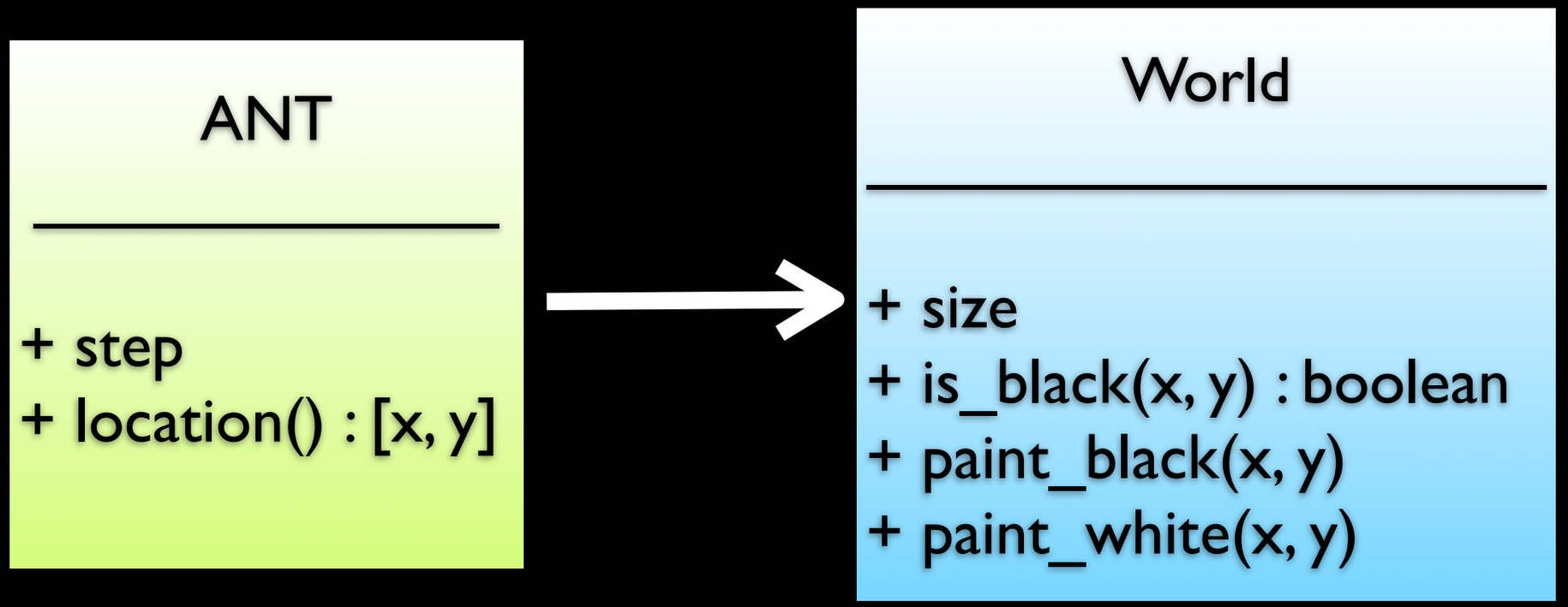


Black Square

If it's on a Black square,
paint it White and go Right



Design


ANT
+ step
+ location() : [x, y]
World
+ size
+ is_black(x, y) : boolean
+ paint_black(x, y)
+ paint_white(x, y)