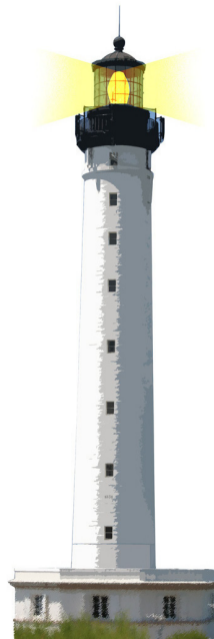# Double dispatch

S. Ducasse

Goals

- In the quest of dispatch
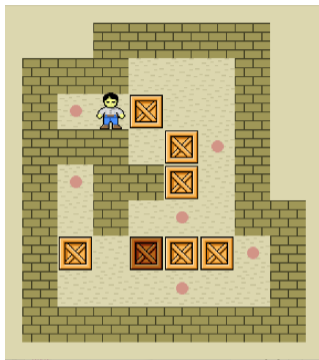- Double dispatch does not have to be symmetrical

Remember

```
>>> (Stone new vs: Paper new)
#paper
```

```
>>> 1 + 3.3
4.3
```

```
>>> 3.3 + 1
4.3
```

Imagine a game model

- Block
  - Box
  - BoxOnTarget
  - EmptyBlock
  - Player
  - Wall

Too many ifs....

```
GameView >> drawBlock: aBlock on: aCanvas
  aBlock isWall
    ifTrue: [ self drawWall: aCanvas ]
    ifFalse: [ aBlock isEmptyBlock
      ifTrue: [ aBlock hasPlayer
          ifTrue: [ aBlock hasTarget
            ifTrue: [ self drawTargetAndPlayer: aCanvas ]
            ifFalse: [ self drawPlayer: aCanvas ]]
        ifFalse: [ aBlock hasBox
          ifTrue: [ aBlock hasTarget
            ifTrue: [ self drawTargetAndBox: aCanvas ]
            ifFalse: [ self drawBox: aCanvas ]]
          ifFalse: [
            aBlock hasTarget
              ifTrue: [ self drawTarget: aCanvas ]
            ifFalse: [ self drawEmptyBlock: aCanvas ]]]
```

A nicer solution

```
GameView >> drawBlock: aBlock on: aCanvas
  aBlock isWall ifTrue: [ self drawWall: aCanvas ].
  aBlock isEmptyBlock ifTrue: [
  aBlock hasPlayer ifTrue: [ ...
```

Becomes

```
GameView >> drawBlock: aBlock on: aCanvas
  aBlock drawOn: aCanvas view: self

Wall >> drawOn: aCanvas view: aView
  aView drawWall: aCanvas

EmptyBlock >> drawOn: aCanvas view: aView
  aView drawEmptyBlock: aCanvas
```

Double dispatch

Each block tells the view how to draw it.

```
GameView >> drawBlock: aBlock on: aCanvas
  aBlock drawOn: aCanvas view: self

Wall >> drawOn: aCanvas view: aView
  aView drawWall: aCanvas

EmptyBlock >> drawOn: aCanvas view: aView
  aView drawEmptyBlock: aCanvas
```

- We **tell** a block to draw itself and it **tells** how to the canvas
- Sending messages is powerful
- Modular

Conclusion

- Double dispatch is creating a varation point without hardcoding the path
- Modular
- Can be asymmetrical

A course by

S. Ducasse, G. Polito, and Pablo Tesone