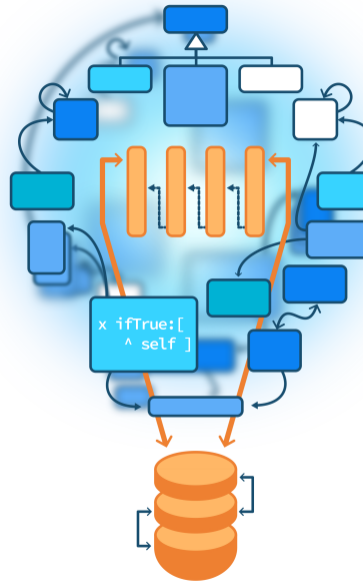


Double dispatch

S.Ducasse, L. Fabresse, G. Polito, and P. Tesone



Goals

- Look at double dispatch
- Double dispatch does not have to be symmetrical



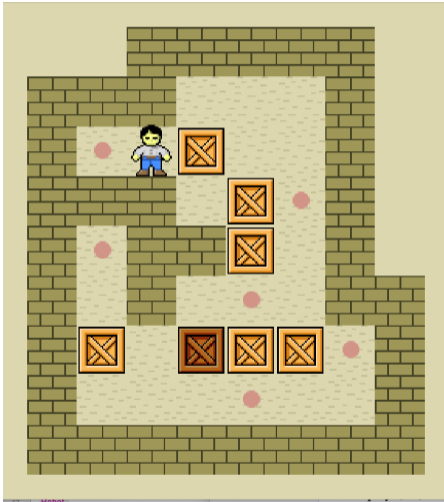
Remember

> (Stone new vs: Paper new)
#paper

> (Scissors new vs: Paper new)
#scissors



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 variation point without hardcoding the path
- Modular
- Can be asymmetrical



Produced as part of the course on <http://www.fun-mooc.fr>

Advanced Object-Oriented Design and Development with Pharo

A course by

S.Ducasse, L. Fabresse, G. Polito, and P. Tesone



Except where otherwise noted, this work is licensed under CC BY-NC-ND 3.0 France
<https://creativecommons.org/licenses/by-nc-nd/3.0/fr/>