

Trabajo Práctico de Programación Orientada a Objetos

Jueves 9 de Noviembre de 2017

Paradigmas de Lenguajes de Programación

Integrante	LU	Correo electrónico
Francisco Demartino	348/14	demartino.francisco@gmail.com
Martín Mongi Badía	422/13	martinmongi@gmail.com
Grupo: "Demo"		



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellon I/Planta Baja)
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (54 11) 4576-3359
<http://www.fcen.uba.ar>

```
Object subclass: #Direction
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!Direction commentStamp: '<historical>' prior: 0!
Direction!

!Direction methodsFor: 'evaluating' stamp: 'PLP 10/9/2017 17:52'!
nextFrom: puntoDeOrigen
"Dado un punto, devuelve el resultado de moverse en la direccion correspondiente. "
^self class nextFrom: puntoDeOrigen ! !

"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --"!

Direction class
instanceVariableNames: ''!

!Direction class methodsFor: 'instance-creation' stamp: 'PLP 10/9/2017 17:53'!
north
^North new! !

!Direction class methodsFor: 'instance-creation' stamp: 'PLP 10/9/2017 17:52'!
south
^South new! !

!Direction class methodsFor: 'instance-creation' stamp: 'PLP 10/9/2017 17:59'!
southwest
^Southwest new! !

!Direction class methodsFor: 'instance-creation' stamp: 'PLP 10/9/2017 17:53'!
east
^East new! !

!Direction class methodsFor: 'instance-creation' stamp: 'PLP 10/9/2017 17:59'!
northeast
^Northeast new! !

!Direction class methodsFor: 'instance-creation' stamp: 'PLP 10/9/2017 17:59'!
southeast
^Southeast new! !

!Direction class methodsFor: 'instance-creation' stamp: 'PLP 10/9/2017 17:53'!
west
^West new! !
```



```
!Game class methodsFor: 'instance creation' stamp: 'MartinMongiBadia 11/3/2017 20:16'!  
newX: width Y: height trapLocations: trapCollection flagLocations: flagCollection  
| game terrain|
```

$$\hat{\text{game.}}$$

```
!GameTeam class methodsFor: 'instance creation' stamp: 'MartinMongiBadia 11/3/2017 17:24'!
```

```
Direction subclass: #North
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!North commentStamp: '<historical>' prior: 0!
North!
```

```
Direction subclass: #Northeast
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!Northeast commentStamp: '<historical>' prior: 0!
Northeast!
```

```
poolDictionaries: ''  
category: 'PLP'  
!Northwest commentStamp: '<historical>' prior: 0!  
Northwest!  
  
"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "  
  
Northwest class  
instanceVariableNames: ''!  
  
!Northwest class methodsFor: 'evaluating' stamp: 'PLP 10/9/2017 17:57'  
nextFrom: puntoDeOrigen  
  
^(puntoDeOrigen x - 1)@(puntoDeOrigen y + 1)! !  
  
Object subclass: #Player  
instanceVariableNames: 'team able terrain'  
classVariableNames: ''  
poolDictionaries: ''  
category: 'PLP'  
!Player commentStamp: '<historical>' prior: 0!  
Represents a player!  
  
!Player methodsFor: 'evaluation' stamp: 'MartnMongiBada 11/3/2017 11:17'  
team  
^team.! !  
  
!Player methodsFor: 'evaluation' stamp: 'MartnMongiBada 11/3/2017 16:33'  
location  
  
^terrain find: self.! !  
  
!Player methodsFor: 'as yet unclassified' stamp: 'MartnMongiBada 11/3/2017 16:16'  
interactWithAllElements  
| elements |  
elements := (terrain elementsAround: self) select: self target.  
elements do: self action.! !  
  
!Player methodsFor: 'path commands' stamp: 'MartnMongiBada 11/3/2017 17:09'  
moveTo: aDirection  
| nextPosition previousPosition |  
previousPosition := self location.  
nextPosition := aDirection nextFrom: self location.  
  
(self canMoveTo: aDirection) ifFalse: [ ^self ].
```

```

(terrain at: nextPosition) steppedOnBy: self.

terrain at: previousPosition put: Empty new.
terrain at: nextPosition put: self.

able ifTrue: [self interactWithAllElements].
! !

!Player methodsFor: 'path commands' stamp: 'MartnMongiBada 11/3/2017 17:11'!
move: aNumberOfSteps to: aDirection
aNumberOfSteps timesRepeat: [ self moveTo: aDirection ].! !

!Player methodsFor: 'capabilities' stamp: 'MartnMongiBada 11/3/2017 16:49'!
canMoveTo: aDirection
|nextPosition|
(self canMove) ifFalse: [ ^false].
nextPosition := aDirection nextFrom: self location.
^(terrain contains: nextPosition) and: [(terrain at: nextPosition) isVisible not].! !

!Player methodsFor: 'capabilities' stamp: 'MartnMongiBada 11/3/2017 11:39'!
sees: anObject
^anObject isVisible.! !

!Player methodsFor: 'capabilities' stamp: 'MartnMongiBada 11/3/2017 11:34'!
canHeal
self subclassResponsibility.! !

!Player methodsFor: 'capabilities' stamp: 'MartnMongiBada 11/3/2017 11:38'!
selectSeenFrom: aCollection
^aCollection select: [:each | self sees: each ].! !

!Player methodsFor: 'visibility' stamp: 'MartnMongiBada 11/3/2017 11:40'!
isVisible
^true.! !

!Player methodsFor: 'initialization' stamp: 'MartnMongiBada 11/3/2017 13:41'!
terrain: aTerrain
terrain := aTerrain.! !

!Player methodsFor: 'initialization' stamp: 'MartnMongiBada 11/3/2017 13:34'!
able: aBoolean
able := aBoolean.! !

!Player methodsFor: 'initialization' stamp: 'MartnMongiBada 11/3/2017 11:08'!

```



```

classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!Engineer commentStamp: '<historical>' prior: 0!
Represents an engineer!

!Engineer methodsFor: 'capabilities' stamp: 'MartnMongiBada 11/3/2017 11:50'!
sees: anObject
^anObject isVisible or: (anObject isKindOf: Trap). ! !

!Engineer methodsFor: 'capabilities' stamp: 'MartnMongiBada 11/3/2017 11:28'!
canHeal
^false. ! !

!Engineer methodsFor: 'accessing' stamp: 'MartnMongiBada 11/3/2017 15:25'!
target
^[:anObject | anObject isKindOf: Trap]. ! !

!Engineer methodsFor: 'accessing' stamp: 'MartinMongiBadia 11/9/2017 02:00'!
action
^[:aTrap | terrain at: (terrain find: aTrap) put: Empty new]. ! !

Player subclass: #Explorer
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!Explorer commentStamp: '<historical>' prior: 0!
Represents an explorer!

!Explorer methodsFor: 'capabilities' stamp: 'MartnMongiBada 11/3/2017 11:28'!
canHeal
^false. ! !

!Explorer methodsFor: 'accessing' stamp: 'MartnMongiBada 11/3/2017 15:22'!
target
^[:anObject | anObject isKindOf: Flag]. ! !

!Explorer methodsFor: 'accessing' stamp: 'MartnMongiBada 11/3/2017 16:08'!
action
^[:aFlag | aFlag team: self team]. ! !

Player subclass: #Medic

```

```
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!Medic commentStamp: '<historical>' prior: 0!
Represents a medic!
```

```
!Medic methodsFor: 'capabilities' stamp: 'MartnMongiBada 11/3/2017 13:36'!  
canHeal  
^able.!!
```

```
!Medic methodsFor: 'accessing' stamp: 'MartinMongiBadia 11/7/2017 19:28'!
target
^[ :anObject | (anObject isKindOfClass: Player) and:
    [anObject team = self team and:
    [anObject canMove not] ] ].! !
```

```
!Medic methodsFor: 'accessing' stamp: 'MartnMongiBada 11/3/2017 16:06'!  
action  
^[:anExplorer | anExplorer healedBy: self]!
```

```
Direction subclass: #South
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!South commentStamp: '<historical>' prior: 0!
South!
```

||-----||

```
South class
instanceVariableNames: ''!
```

```
!South class methodsFor: 'evaluating' stamp: 'PLP 10/14/2017 23:33'!  
nextFrom: puntoDeOrigen
```

$$\hat{(\text{puntoDeOrigen } x)@(\text{puntoDeOrigen } y - 1)! \quad !$$

```
Direction subclass: #Southeast
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!Southeast commentStamp: '<historical>' prior: 0!
```

||-----||

```
!Southeast class methodsFor: 'evaluating' stamp: 'PLP 10/9/2017 17:58'!  
nextFrom: puntoDeOrigen
```

```
Direction subclass: #Southwest
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!Southwest commentStamp: '<historical>' prior: 0!
Southwest!
```

||-----||

```
!Southwest class methodsFor: 'evaluating' stamp: 'PLP 10/9/2017 17:58'!  
nextFrom: puntoDeOrigen
```

```
Object subclass: #Terrain
instanceVariableNames: 'width height matrix'
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!Terrain commentStamp: '<historical>' prior: 0!
Represents the terrain of the game!
```

```
| index |
index := matrix indexOf: anObject ifAbsent: [^self].
matrix at: index x at: index y put: nil.!!
```

11

```

width: w height: h
"comment stating purpose of message"

width := w.
height := h.
matrix := Matrix rows: h columns: w element: Empty new.
^self.!!

!Terrain methodsFor: 'enumerating' stamp: 'MartnMongiBada 11/3/2017 10:36'!
select: aBlock
"comment stating purpose of message"

^matrix asBag select: aBlock.!!

!Terrain methodsFor: 'evaluation' stamp: 'MartnMongiBada 11/3/2017 13:14'!
elementsAround: aPlayer

^self elementsAround: aPlayer satisfying: [ :each | true ].!!

!Terrain methodsFor: 'evaluation' stamp: 'MartinMongiBadia 11/7/2017 19:15'!
contains: aPoint
"comment stating purpose of message"

^ (aPoint x between: 1 and: width) and: [aPoint y between: 1 and: height].!!

!Terrain methodsFor: 'evaluation' stamp: 'MartnMongiBada 11/3/2017 13:50'!
elementsAround: aPlayer satisfying: aBlock

| index elements|

index := self find: aPlayer.
elements := (self eightNeighbors: index) collect: [:point | self at: point].
^(aPlayer selectSeenFrom: elements) select: aBlock.!!

!Terrain methodsFor: 'evaluation' stamp: 'MartnMongiBada 11/3/2017 12:29'!
eightNeighbors: aPoint

| neighbors |

neighbors :=
Direction allSubclasses collect: [:direction | direction nextFrom: aPoint].
^neighbors select: [:point | self contains: point].!!

!Terrain methodsFor: 'evaluation' stamp: 'MartnMongiBada 11/3/2017 09:57'!
visibleAt: aPoint
"comment stating purpose of message"

```



```
!TerrainPosition methodsFor: 'as yet unclassified' stamp: 'MartnMongiBada 11/3/2017 16:40'!  
isVisible  
self subclassResponsibility ! !
```

```
TerrainPosition subclass: #Empty  
instanceVariableNames: ''  
classVariableNames: ''  
poolDictionaries: ''  
category: 'PLP'!  
!Empty commentStamp: '<historical>' prior: 0!  
Represents an empty position!
```

```
!Empty methodsFor: 'as yet unclassified' stamp: 'MartnMongiBada 11/3/2017 10:59'!  
isNotNil  
^self isNil not. ! !
```

```
!Empty methodsFor: 'as yet unclassified' stamp: 'MartnMongiBada 11/3/2017 10:58'!  
isNil  
^true. ! !
```

```
!Empty methodsFor: 'visibility' stamp: 'MartnMongiBada 11/3/2017 11:00'!  
isVisible  
^false. ! !
```

```
TerrainPosition subclass: #Flag  
instanceVariableNames: 'team'  
classVariableNames: ''  
poolDictionaries: ''  
category: 'PLP'!  
!Flag commentStamp: '<historical>' prior: 0!  
Representing a Flag that should be located in the Terrain!
```

```
!Flag methodsFor: 'accessing' stamp: 'MartnMongiBada 11/3/2017 04:41'!  
team  
"comment stating purpose of message"  
  
^team. ! !
```

```
!Flag methodsFor: 'setter' stamp: 'MartnMongiBada 11/3/2017 04:48'!  
team: aColor  
"comment stating purpose of message"  
  
team := aColor. ! !
```

```

!Flag methodsFor: 'visibility' stamp: 'MartnMongiBada 11/3/2017 04:30'!
isVisible
"comment stating purpose of message"

^true.!!

!Flag methodsFor: 'initialization' stamp: 'MartnMongiBada 11/3/2017 04:46'!
new
"comment stating purpose of message"
team := nil.
^self.!!

TerrainPosition subclass: #Trap
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!Trap commentStamp: '<historical>' prior: 0!
Represents a Trap that is supposed to go in the Terrain!

!Trap methodsFor: 'as yet unclassified' stamp: 'MartnMongiBada 11/3/2017 16:42'!
steppedOnBy: aPlayer
aPlayer disable.
^self.!!

!Trap methodsFor: 'as yet unclassified' stamp: 'MartnMongiBada 11/3/2017 04:36'!
value
"comment stating purpose of message"

^nil.!!

!Trap methodsFor: 'as yet unclassified' stamp: 'MartnMongiBada 11/3/2017 04:31'!
isVisible
"comment stating purpose of message"

^false.!!

Direction subclass: #West
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!West commentStamp: '<historical>' prior: 0!
West!

```


" _ _ _ _ _ " !

West class

```
instanceVariableNames: ''!
```

```
!West class methodsFor: 'evaluating' stamp: 'PLP 10/14/2017 23:34'!
```

```
nextFrom: puntoDeOrigen
```

$$^{\text{(puntoDeOrigen } x - 1) @ (\text{puntoDeOrigen } y)!}$$