

## 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 Autonoma de Buenos Aires - Rep. Argentina Tel/Fax: (54 11) 4576-3359 http://www.fcen.uba.ar

## 1 TP3.st (sin tests)

```
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!!
```

```
!Direction class methodsFor: 'instance-creation' stamp: 'PLP 10/9/2017 17:59'!
northwest
^Northwest new! !
!Direction class methodsFor: 'evaluating' stamp: 'PLP 10/9/2017 17:54'!
nextFrom: puntoDeOrigen
^ self subclassResponsibility !!
Direction subclass: #East
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!East commentStamp: '<historical>' prior: 0!
East!
"-- -- -- -- -- -- -- -- -- -- -- -- "!
East class
instanceVariableNames: ''!
!East class methodsFor: 'evaluating' stamp: 'PLP 10/14/2017 23:32'!
nextFrom: puntoDeOrigen
^(puntoDeOrigen x + 1)@(puntoDeOrigen y)! !
Object subclass: #Game
instanceVariableNames: 'terrain teams currentTeamIndex'
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!Game commentStamp: '<historical>' prior: 0!
Represents a game!
!Game methodsFor: 'accessing' stamp: 'MartinMongiBadia 11/3/2017 19:36'!
currentTeam
^teams at: currentTeamIndex.! !
!Game methodsFor: 'accessing' stamp: 'MartinMongiBadia 11/3/2017 19:01'!
terrain: aTerrain
terrain := aTerrain.! !
!Game methodsFor: 'accessing' stamp: 'MartinMongiBadia 11/3/2017 20:05'!
terrain
^terrain.!!
```

```
!Game methodsFor: 'accessing' stamp: 'MartinMongiBadia 11/3/2017 19:28'!
teams: aTeamCollection
teams := aTeamCollection.! !
!Game methodsFor: 'accessing' stamp: 'MartinMongiBadia 11/7/2017 19:09'!
currentWinners
| claimedFlags teamFlags scores |
claimedFlags :=
terrain select: [:pos | (pos isKindOf: Flag) and: [pos team isNotNil] ].
teamFlags := claimedFlags groupedBy: [:flag | flag team ].
scores := teamFlags keys groupedBy: [ :aTeam | (teamFlags at: aTeam) size.].
scores ifEmpty: [ ^OrderedCollection new. ].
^(scores at: scores keys max).!!
!Game methodsFor: 'accessing' stamp: 'MartinMongiBadia 11/3/2017 19:37'!
currentTeamIndex: aNumber
^currentTeamIndex := aNumber.! !
!Game methodsFor: 'as yet unclassified' stamp: 'MartinMongiBadia 11/7/2017 19:10'!
registerTeam: aColor playerLocations: anArray
|members team|
members := OrderedCollection new.
1 to: anArray size by: 2 do:
[:i | (anArray at: i+1) new: terrain with: (anArray at: i) and: aColor.
  members add: (terrain at: (anArray at: i))].
team := (GameTeam new: aColor with: members).
teams add: team.
!!
!Game methodsFor: 'as yet unclassified' stamp: 'MartinMongiBadia 11/3/2017 20:17'!
moveCharacter: anIndex to: aDirection steps: aNumber
| player |
player := (teams at: currentTeamIndex) members at: anIndex.
player move: aNumber to: aDirection.
self pass.!!
!Game methodsFor: 'handling' stamp: 'MartinMongiBadia 11/3/2017 19:43'!
currentTeamIndex := currentTeamIndex \\ teams size + 1.! !
```

Game class

```
instanceVariableNames: ''.
!Game class methodsFor: 'instance creation' stamp: 'MartinMongiBadia 11/3/2017 20:16'!
newX: width Y: height trapLocations: trapCollection flagLocations: flagCollection
| game terrain|
game := self new.
terrain := (Terrain dimX: width dimY: height).
flagCollection do: [:aPoint | terrain at: aPoint put: Flag new].
trapCollection do: [:aPoint | terrain at: aPoint put: Trap new].
game terrain: terrain.
game teams: OrderedCollection new.
game currentTeamIndex: 1.
^game.
!!
Object subclass: #GameTeam
instanceVariableNames: 'teamName members'
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!GameTeam commentStamp: '<historical>' prior: 0!
Represents a team in the game!
!GameTeam methodsFor: 'accessing' stamp: 'MartinMongiBadia 11/3/2017 17:26'!
members
^members.! !
!GameTeam methodsFor: 'accessing' stamp: 'MartinMongiBadia 11/3/2017 17:25'!
teamName: aColor
teamName := aColor.! !
!GameTeam methodsFor: 'accessing' stamp: 'MartinMongiBadia 11/3/2017 17:25'!
members: aCollection
members := aCollection.! !
!GameTeam methodsFor: 'accessing' stamp: 'MartinMongiBadia 11/3/2017 17:27'!
name
^teamName.!!
GameTeam class
instanceVariableNames: ''!
!GameTeam class methodsFor: 'instance creation' stamp: 'MartinMongiBadia 11/3/2017 17:24'!
```

```
new: aColor with: aCollectionOfMembers
|team|
team := self new.
team teamName:aColor.
team members: aCollectionOfMembers.
^team.!!
Direction subclass: #North
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!North commentStamp: '<historical>' prior: 0!
North!
"-- -- -- -- -- -- -- -- -- -- -- -- "!
North class
instanceVariableNames: ''.!
!North class methodsFor: 'evaluating' stamp: 'PLP 10/14/2017 23:33'!
nextFrom: puntoDeOrigen
^(puntoDeOrigen x)@(puntoDeOrigen y + 1)!!
Direction subclass: #Northeast
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!Northeast commentStamp: '<historical>' prior: 0!
Northeast!
Northeast class
instanceVariableNames: ''!
!Northeast class methodsFor: 'evaluating' stamp: 'PLP 10/9/2017 17:57'!
nextFrom: puntoDeOrigen
^(puntoDeOrigen x + 1)@(puntoDeOrigen y + 1)! !
Direction subclass: #Northwest
instanceVariableNames: ''
classVariableNames: ''
```

```
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'!
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'!
```

```
team: aTeam
team := aTeam.! !
!Player methodsFor: 'testing' stamp: 'MartnMongiBada 11/3/2017 13:34'!
canMove
^able.!!
!Player methodsFor: 'setter' stamp: 'MartinMongiBadia 11/7/2017 19:11'!
healedBy: aPlayer
(aPlayer canHeal) ifFalse: [Exception signal].
(aPlayer = self) ifTrue: [Exception signal].
able := true.! !
!Player methodsFor: 'setter' stamp: 'MartnMongiBada 11/3/2017 13:34'!
disable
able := false.! !
!Player methodsFor: 'accessing' stamp: 'MartnMongiBada 11/3/2017 15:21'!
target
self subclassResponsibility.! !
!Player methodsFor: 'accessing' stamp: 'MartnMongiBada 11/3/2017 15:28'!
self subclassResponsibility.!!
"-- -- -- -- -- -- -- -- -- -- -- -- "!
Player class
instanceVariableNames: ''!
!Player class methodsFor: 'initialization' stamp: 'MartnMongiBada 11/3/2017 13:43'!
new: aTerrain with: aPoint and: aTeam
"comment stating purpose of message"
| player |
player := self new.
player team: aTeam.
player terrain: aTerrain.
player able: true.
aTerrain at: aPoint put: player.
^player.!!
Player subclass: #Engineer
instanceVariableNames: ''
```

```
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'!
^[ :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 isKindOf: Player) and:
   [anObject team = self team and:
   [anObject canMove not] ] .! !
!Medic methodsFor: 'accessing' stamp: 'MartnMongiBada 11/3/2017 16:06'!
^[: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
^(puntoDeOrigen x)@(puntoDeOrigen y - 1)! !
Direction subclass: #Southeast
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!Southeast commentStamp: '<historical>' prior: 0!
```

```
Southeast!
"-- -- -- -- -- -- -- -- -- -- -- -- "!
Southeast class
instanceVariableNames: ''!
!Southeast class methodsFor: 'evaluating' stamp: 'PLP 10/9/2017 17:58'!
nextFrom: puntoDeOrigen
^(puntoDeOrigen x + 1)@(puntoDeOrigen y - 1)!!
Direction subclass: #Southwest
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!Southwest commentStamp: '<historical>' prior: 0!
Southwest!
"-- -- -- -- -- -- -- -- -- -- -- -- "!
Southwest class
instanceVariableNames: ''!
!Southwest class methodsFor: 'evaluating' stamp: 'PLP 10/9/2017 17:58'!
nextFrom: puntoDeOrigen
^(puntoDeOrigen x - 1)@(puntoDeOrigen y - 1)!!
Object subclass: #Terrain
instanceVariableNames: 'width height matrix'
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!Terrain commentStamp: '<historical>' prior: 0!
Represents the terrain of the game!
!Terrain methodsFor: 'setter' stamp: 'MartnMongiBada 11/3/2017 10:51'!
remove: anObject
| index |
index := matrix indexOf: anObject ifAbsent: [^self].
matrix at: index x at: index y put: nil.! !
!Terrain methodsFor: 'setter' stamp: 'MartnMongiBada 11/3/2017 10:58'!
```

```
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 all Subclasses 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"
```

```
^(self at: aPoint) isVisible ifTrue: true ifFalse: nil.!!
!Terrain methodsFor: 'evaluation' stamp: 'MartnMongiBada 11/3/2017 13:49'!
find: anObject
|point|
point := matrix indexOf: anObject ifAbsent: [^self].
^point y@point x.! !
!Terrain methodsFor: 'as yet unclassified' stamp: 'MartnMongiBada 11/3/2017 09:42'!
at: aPoint
"comment stating purpose of message"
^matrix at: aPoint y at: aPoint x.! !
!Terrain methodsFor: 'as yet unclassified' stamp: 'MartnMongiBada 11/3/2017 09:46'!
at: aPoint put: anObject
matrix at: aPoint y at: aPoint x put: anObject.! !
Terrain class
instanceVariableNames: ''!
!Terrain class methodsFor: 'initialization' stamp: 'MartnMongiBada 11/3/2017 05:25'!
dimX: aDimension dimY: anotherDimension
"comment stating purpose of message"
| terrain |
terrain := self new.
terrain width: aDimension height: anotherDimension.
^terrain.!!
Object subclass: #TerrainPosition
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'PLP'!
!TerrainPosition commentStamp: '<historical>' prior: 0!
Represents a terrain position
!
!TerrainPosition methodsFor: 'as yet unclassified' stamp: 'MartnMongiBada 11/3/2017 16:41'!
steppedOnBy: aPlayer
^self.! !
```

```
!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'!
"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'!
"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!
```