

ECOLE DES MINES DE DOUAI

DECREUSE Renals

OUALLET Robin

TOUREAU Jean-Baptiste

Documentation Projet IA 2012

# Table des matières

[1 Table des matières 2](#_Toc349057126)

[2 Gameboard 3](#_Toc349057127)

[2.1 Diagramme 3](#_Toc349057128)

[2.2 Descriptif des classes 4](#_Toc349057129)

[3 Stratégie 5](#_Toc349057130)

[3.1 Diagramme 5](#_Toc349057131)

[3.2 Descriptif des classes 6](#_Toc349057132)

[4 Beacon 7](#_Toc349057133)

[4.1 Diagramme 7](#_Toc349057134)

[4.2 Patterns 7](#_Toc349057135)

[4.3 Descriptif des classes 8](#_Toc349057136)

# Gameboard

## Diagramme

### 

## Descriptif des classes

### Classe GameBoardPainter

Contient toutes les méthodes pour dessiner les éléments.

### Package gameboard.elements

Ce package contient des classes qui seront dessinées, qui gèrent les bordures et les trajectoires, ainsi qu’une classe DrawHelper qui permet de dessiner de façon simplifiée des cercles et des rectangles.

### Package org.cen.ui.gameboard.web

Ce package contient toutes les classes qui forment les onglets de l’interface web. Ces views sont affichées quand l’utilisateur clique sur l’onglet correspondant.

### Classe AbstractGameBoardElement

Superclasse de tous les éléments du gameboard qui doivent être représentés. Cette classe implémente l’interface IGameBoardElement qui contient la méthode paint(). C’est grâce à cet héritage que tous les éléments implémentent cette méthode et qu’un tableau d’éléments pourra être déclaré puis dessiné.

### Classe GameBoard2013

C’est la classe centrale du gameboard à définir chaque année car elle contient un tableau de tous les éléments à dessiner ainsi que leurs positions respectives.

### Classe GameBoardPainter

Cette classe permet de dessiner la totalité du gameboard. Elle contient entre autre les méthodes suivantes :

* Paint
* paintElement
* paintNavigationPoint
* paintPath
* paintRobot
* paintShapes
* paintTargets.

# Stratégie

## Diagramme



## Descriptif des classes

### Classe IGameStrategy

La classe IGameStrategy est l’interface de la classe qui représente une stratégie à adopter lors d’un match.Une stratégie se compose de plusieurs StrategyItems réunis dans une liste IGameStrategyItemList. La méthode getItems permet de récupérer cette liste.

### Classe IGameStrategyItem

La classe IGameStrategyItem est l’interface de la classe qui représente un item de stratégie, à savoir l’association d’une cible implémentant l’interface ITarget, et d’un gain implémentant l’interface ITargetGain. Ces items sont les briques de base d’une stratégie.

### Classe ITarget

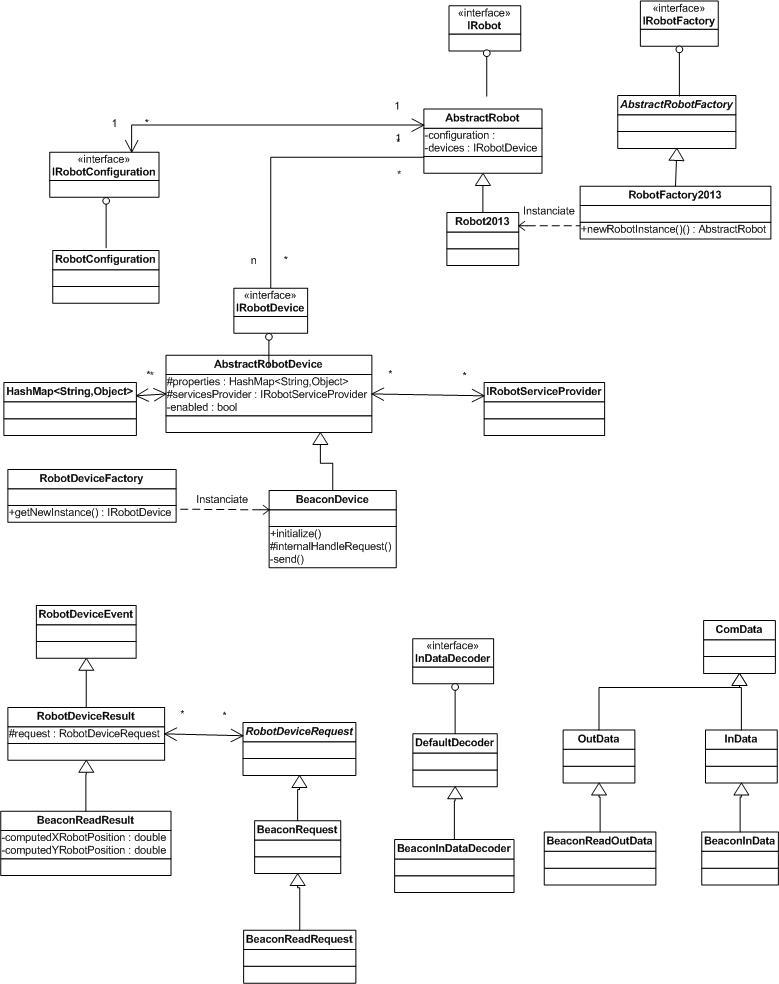
La classe ITarget est l’interface de la classe qui représente une cible stratégique. La cible se compose d’une liste d’action à réaliser qui implémentent ITargetAction. Ces actions sont réunies dans une implémentation de ITargetActionList

### Classe ITargetAction

La classe ITargetAction est l’interface de la classe qui représente une action complexe à réaliser pour atteindre l’objectif d’une ITarget. Elle est composée d’une liste de ITargetActionItem qui représente des actions élémentaires du robot, réalisables par une seule requête. Ces actions élémentaires sont réunies dans une implémentation de ITargetActionItemList.

# Beacon

## Diagramme



## Patterns

On remarque la présence de plusieurs paterns Factory. Le premier est la factory de Robot, avec la factory spécialisée RobotFactory2013. Le second semble se trouver autour des devices. Cependant, la factory RobotDeviceFactory n’est ni générique ni spécialisé. Le patern est donc incomplet.

## Descriptif des classes

### Classe AbstractRobot

La classe AbstractRobot représente un Robot générique, qui sera spécialisé chaque année. Il contient notamment la configuration du Robot, ainsi que la liste des Devices qui lui sont attachées.

### Classe AbstractRobotDevice

La classe AbstractRobotDevice représente un composant générique qui peut faire parti du robot. Ce composant doit avoir un serviceProvider, une liste de configuration, et être capable de répondre au message internalHandleRequest.