

Procédure pour créer un plugin

Plugin graphique

Afin d'ajouter un plugin graphique, il premièrement que celui-ci implémente l'interface 'IGraphisme' présente dans le projet 'interfaces' et les méthodes associées :

- `public void draw(IRobot r, Graphics g)`. Cette méthode est la méthode principale du plugin. C'est celle-ci qui sera chargée de dessiner les robots. Il vous faudra donc appelé les méthodes désirées sur l'objet `Graphics` passé en paramètre. L'objet `IRobot` vous donnera quant à lui les données nécessaires afin d'afficher le robot correctement (taille, coordonnées, couleur...)
- `public addNameIGraphisme(String name)`, cette méthode doit être appelée dans le constructeur de la classe. Elle a pour but d'ajouter le nom du plugin à la liste des plugins graphismes qui ont déjà été chargé. La méthode doit faire appel à la méthode `addNameIGraphisme(name)` du précédent décorateur.
- `public ArrayList<String> getListeNames()`, retourne la liste des plugins qui ont été chargés. La méthode doit faire appel au `getListeNames()` du précédent décorateur.

Plugin déplacement

Comme chaque plugin, celui-ci devra implémenter l'interface 'IDeplacement' dans le projet 'interfaces'. La méthode suivante lui sera associée :

- `public void move(IRobot r, ArrayList<IRobot> adversaires)`. Cette méthode prend en paramètre le robot sur lequel vous pourrez modifier les coordonnées et également une liste de robots. Cette liste est composée de tous les adversaires présents sur le terrain afin de faire bouger le robot concerné en fonction de la position des autres. Dans cette méthode, il n'y a nul besoin de refaire un affichage du robot après modification de ses coordonnées, le jeu rappellera simplement le plugin graphique pour les rafraichir.

Plugin attaque

Le plugin devra toujours implémenter l'interface 'IAttaque' du projet 'interfaces' avec les méthodes suivantes :

- public void attaque(Graphics gr, IRobot adversaire, IProjectile p). Méthode principale du plugin, cette méthode va calculer de quelle manière le robot doit attaquer en fonction de son adversaire désigné. Il y a cependant une différence avec le plugin déplacement, c'est que les attaques sont gérées par un plugin qui leur est dédié. Il sera créé lors de la création de votre plugin d'attaque.
- public void touche(IProjectile p). Cette méthode sera appelée si le projectile touche un adversaire, il vous suffira ensuite de mettre la ligne suivante pour enlever des points à l'adversaire :

```
p.getAdversaire().soustrairedelavie(/* mettre ici le nombre de points de vie à enlever */) ;
```

Veuillez faire attention à mettre un constructeur initialisant un attribut IGraphismeAttaque.

Plugin graphique pour les attaques

Toujours le même principe, vous devrez implémenter l'interface 'IGraphismeAttaque' du projet 'interfaces' et implémenter cette méthode :

- public void drawWeapon(IProjectile p, Graphics g, int direction). Cette méthode fonctionne dans le même principe que 'draw' du plugin graphique, elle dessinera une attaque selon vos envies sur l'objet Graphics. La direction est un indicateur pour préciser où se trouve l'adversaire (0 : à droite, 1 : en bas, 2 : à gauche, 3 : en haut)

IMPORTANT :

- Pour les plugins graphiques, il vous faudra un constructeur qui prend en paramètre un autre objet IGraphisme. Celui-ci sera donc attribut du plugin, cela sert au patron décorateur afin de pouvoir coupler plusieurs plugins graphiques. Et il vous faudra également appeler la méthode draw de cet attribut avec les mêmes arguments que sa propre méthode draw.