

projet RICOCHETS ROBOTS

Compte rendu de la première phase de travail.

Chaque membre de l'équipe a proposé un code pour répondre à la demande N°1.

I.Représentation du plateau de jeu

Nous avons deux propositions pour représenter les cloisons.

Proposition N°1 :

Un tableau **grille** de taille 6 x 6 .

grille [i][j] est un entier qui dont les 4 premiers bits codent les cloisons selon :

cloison du nord : bit de poids 1 | cloison à l'est bit de poids 2

cloison du sud : bit de poids 4 | cloison à l'ouest bit de poids 8

exemple : Dans le coin en haut à gauche il y a une cloison au nord et à l'ouest.

On a grille[0][0] = 9

La grille est une variable globale du module.

Proposition N° 2 et 3 :

Deux tableaux **mursHorizontaux**[6][7] et **mursVerticaux**[7][6] qui donnent pour chaque position la présence d'une cloison par un code.

La proposition 2 implémente ces deux tableaux comme attributs d'une classe **grille** sans instances.

La proposition 3 implémente ces deux tableaux comme attributs d'une classe **jeu**.

II Représentation des robots.

Les trois propositions implémentent une classe **robot**.

Les attributs diffèrent selon les propositions.

classe robot	classe robot	classe robot
<u>Attributs :</u> robot.x <i>int</i> robot.y <i>int</i> robot.colour <i>string</i>	<u>Attributs :</u> robot.position <i>(int,int)</i> robot.num <i>int</i> robot.code <i>int</i>	<u>Attributs :</u> robot.x <i>int</i> robot.y <i>int</i> robot.name <i>string</i>
<u>méthodes :</u> constructeur move(direction)	<u>méthodes :</u> constructeur	<u>méthodes :</u> constructeur move(direction)

III gestion des déplacements des robots.

Une proposition confie le déplacement des robots à une classe jeu.
Cette classe a pour attributs les cloisons ainsi que la liste des robots.

Les deux autres propositions confient le déplacement des robots à la classe robot.
Les robots échangent leurs positions au moyen d'une variable globale : un dictionnaire { string : (int , int) }

François Aubin a fait un essai de représentation avec Pygame.
Théo Giani a fait un essai de représentation avec Tkinter.

IV Bilan de la réunion :

1. Nous décidons de représenter la grille de jeu par un tableau.
Chaque case contient un entier dont les 4 bits de poids faibles indiquent la présence ou non d'une cloison.

bit	0001	0010	0100	1000
cloison au	Sud	Est	Nord	Ouest

La possibilité de sortir d'une case dépend de la direction de déplacement.
Si l'on se déplace vers le Nord il faut tester s'il y a une cloison au Nord.
On utilisera le masque

```
mask=0b0100
```

et le test

```
grid[x][y] & mask == 0
```

détermine si il n'y a pas de cloisons qui barre le passage.

2. Nous décidons de déléguer la gestion des mouvements de robots à une classe Game ayant la grille et la liste des robots comme attributs.
3. Norme pour le code :
Nous utilisons la convention snake_case et suivons les recommandations PEP8.
Les noms des variables, des classes et des fonctions sont en anglais.