# **TP2 C++**

#### Introduction

Je vous présenterai dans ce document le compte rendu de mon TP2, qui est composé de 19 fichiers qui implémentent un labyrinthe. J'ai pour cela crée un objet graphique fixe (un plateau composé de murs, de couloirs, d'une sortie et d'une entrée), un objet graphique mobile (un personnage). Dans mon fichier labyrinthe, je superpose les deux afin de faire évoluer mon personnage dans le labyrinthe.

Dans ce compte rendu, je m'attarderai sur les fichiers labyrinthe et plateau. Je ferai ensuite des tests sur mon labyrinthe afin de montrer qu'il fonctionne dans tous les cas. Pour conclure je ferai un bilan avec les difficultés que j'ai rencontré et ce que j'ai appris.

### I. Classe Plateau

Ma classe plateau est composée de 3 attributs :

- Le nombre de ligne du plateau : m nbl
- Le nombre de ligne du plateau : m\_nbc
- D'un pointeur d'objet graphique fixe : m plateau

Ma classe plateau est composée de 4 méthodes :

- setCase : prend en argument les coordonnées de la case et un pointeur d'objet graphique fixe (un mur/couloir/sortie). Elle permet de modifier une case du labyrinthe
- getCase : prend un argument les coordonnées de la case. Elle retourne la case du labyrinthe correspondante aux coordonnées en argument.
- Afficher(void): Cette méthode affiche le plateau, c'est-à-dire l'objet graphique fixe.
- Afficher(ObjetGraphiqueMobile& o) : Cette méthode affiche le plateau fixe ainsi que le personnage. Elle prend un argument le personnage.

#### II. <u>Classe Labyrinthe</u>

Ma classe labyrinthe est composée de 4 attributs :

- Le nombre de ligne du plateau : m\_nbl
- Le nombre de ligne du plateau : m nbc
- D'un pointeur de plateau : m\_plateau

- D'un pointeur de personnage : m personnage

Ma classe labyrinthe est composée de 6 méthodes :

- Afficher : cette méthode affiche le labyrinthe, avec le plateau et le personnage.
- 4 méthodes pour déplacer le personnage vers la gauche/la droite/le haut/le bas. J'ai vérifié que le déplacement était possible, c'est-à-dire qu'il ne menait pas vers un mur ou qu'il ne faisait pas sortir du plateau.
- Fini : cette méthode renvoie un booléen : vrai si le personnage est arrivé à la sortie, faux sinon. On regarde si les coordonnées du personnage sont égales aux coordonnées de la sortie.

#### > Constructeur de labyrinthe :

Dans ce TP, j'ai utilisé des fichiers afin de créer mes labyrinthes. J'ai donc donné en argument de mon constructeur de labyrinthe le nom du fichier contenant un labyrinthe. J'ai ensuite ouvert et lu ce fichier dans mon constructeur (je traite le cas où il y a une erreur dans l'ouverture du fichier), en construisant au fur et à mesure mon plateau grâce à la méthode plateau setCase. De cette manière, j'initialise les attributs de mon labyrinthe : le nombre de ligne et de colonne sont sur les deux premières lignes de mes fichiers labyrinthe, j'initialise le plateau grâce à la grille, et personnage grâce à la position du P sur la grille.

Mes fichiers labyrinthe sont :

		15
5	10	15
	10	*********P***
	10	*** ** **
5	****P***	** ****
)	* ** * **	***** ******
*P***	at a de ate ate ate ate ate ate ate	*** ******
	**** ****	*****
* *	****	*** ******
	******	***** ******
* *	****	***
	****	******
	*** **	******** ***
* **	****	*****
		*******
**S**	*****	******
	*******	********

#### III. <u>Tests</u>

• Test labyrinthe niveau 1:5x5

• Test labyrinthe niveau 2 : 10x10

```
****p****
* ** * **
****
     ****
****
        *
******
****
***
       **
****
*****
*******5*
Quel deplacement ? Bas(B) , Haut(H), Droite(D), Gauche(G) :
****
* ** P* **
****
****
****
       **
****
       **
*****
*******
Quel deplacement ? Bas(B) , Haut(H), Droite(D), Gauche(G) :
**** ****
* **p * **
**** ****
****
       *
****
      **
****
      **
*****
*******5*
```

• Test labyrinthe niveau 3 : 15x15 :

```
**********p***
**** **
**
     ****
*****
*** ******
*****
*** *****
*****
****
      ***
*****
********
*****
*******
******
********5
Quel deplacement ? Bas(B) , Haut(H), Droite(D), Gauche(G) :
*******
**** ** P***
    ****
**
*****
*** ******
*****
*** ******
*****
****
      ***
******
*******
******
*****
*********$
Quel deplacement ? Bas(B) , Haut(H), Droite(D), Gauche(G) :
******
**** p ***
**
    ****
****
*** *****
*****
*** *****
*****
****
     ***
*****
*******
*****
******
*****
******
```

• Test mauvaise saisie de la direction

```
Niveau 1, 2, 3 ?
1
*p***
*     *
*     *
*     **
***S**

Quel deplacement ? Bas(B) , Haut(H), Droite(D), Gauche(G) :
F
Quel deplacement ? Bas(B) , Haut(H), Droite(D), Gauche(G) :
```

• Test déplacement impossible : mur

• Test déplacement impossible : hors du plateau

Test gagné

A noter que si on modifie le nom du fichier labyrinthe dans le code, le cas de l'ouverture impossible du fichier est traité :

```
Niveau 1, 2, 3 ?
1
Impossible d'ouvrir le fichier !
```

## IV. Bilan

Pour clôturer ce compte rendu de TP, je dirai que j'ai l'impression que mon code fonctionne au vu des tests effectués. J'ai rencontré des difficultés à écrire le constructeur de la classe labyrinthe avec les fichiers. J'ai bloqué longtemps sur cette étape car j'ai mis du temps à comprendre la différence entre un objet graphique fixe et un objet graphique mobile. Je n'avais pas compris qu'il fallait lire chaque caractère du fichier et remplir le plateau en fonction du type de la case.

Ce TP m'a appris à ouvrir et lire un fichier, il m'a également permis de mieux comprendre la manipulation des pointeurs.