

Programmation 3

L2 Informatique 2012-2013

Fiche de TP 2

Notions abordées : gestion des tableaux ; définition de types structurés ; bibliothèque graphique MLV.

L'objectif de ce TP est d'implanter le *Chomp*, un jeu mathématique à deux joueurs. Les règles sont les suivantes. Deux joueurs se disputent une tablette de chocolat de dimension $n \times m$ où n et m sont des entiers supérieurs à un (voir figure 1). Chaque joueur choisit

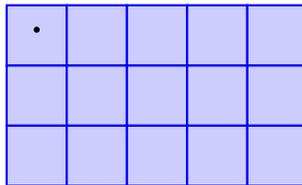


FIGURE 1 – La tablette de chocolat du *Chomp* 3×5 dans sa configuration initiale. Le carré empoisonné contient un point •.

alternativement un carré de chocolat, le mange, et mange aussi tous les carrés qui se trouvent en bas et à sa droite (voir figure 2). La partie s'arrête lorsque l'un des deux joueurs mange

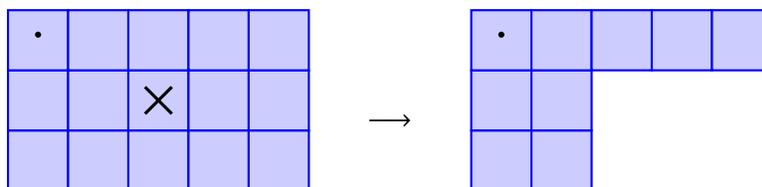


FIGURE 2 – Le joueur mange un carré de chocolat — celui qui contient une croix ×. Ceci a l'effet de manger tous les carrés de chocolat situés en bas et à sa droite.

le carré de chocolat en position $(0, 0)$. Ce carré est en effet empoisonné et ce joueur perd la partie.

L'un des objectifs de ce TP est d'apprendre à utiliser la bibliothèque graphique MLV, qui permet d'afficher la tablette de chocolat et de jouer au *Chomp* à la souris.

Exercice 1. (La bibliothèque graphique MLV)

1. Visiter la page

<http://www-igm.univ-mlv.fr/~boussica/mlv/index.html>,

télécharger et installer la bibliothèque graphique MLV.

2. Parcourir sa documentation et s'assurer qu'elle a bien été installée en créant un programme `Rectangle.c` qui affiche un rectangle rouge dans une fenêtre.
3. Créer un programme `RectangleClic.c` qui affiche un rectangle rouge dans une fenêtre. Lorsque l'utilisateur clique sur le rectangle, il change de couleur (il passe du rouge au bleu et du bleu au rouge).

Exercice 2. (Définition des types)

1. Définir un type `Tablette` qui représente une tablette de chocolat de dimension $n \times m$. Utiliser un tableau $n \times m$ qui contient des entiers. Un contenu à 1 signifie que le carré de chocolat correspondant existe encore tandis qu'un contenu à 0 signifie qu'il a été mangé. Toute variable de type `Tablette` connaît sa dimension *i.e.*, les entiers n et m .
2. Définir un type `Joueur` qui permet de modéliser les deux joueurs. On demande ici un type énumération.
3. Définir un type `Position` qui permet de représenter une position de jeu de *Chomp*. Une position est déterminée par une tablette de chocolat et le joueur dont c'est le tour de jouer.
4. Définir un type `Coup` qui permet de modéliser un coup joué. Un coup est entièrement spécifié par les coordonnées x et y du carré de chocolat que le joueur souhaite manger.

Exercice 3. (Manipulation des objets)

1. Écrire une fonction

```
1 Tablette creer_tablette(int n, int m);
```

qui crée et retourne une variable de type `Tablette` de dimension $n \times m$. La tablette retournée possède tous ses carrés de chocolat.

2. Écrire une fonction

```
1 void manger(Tablette *t, int x, int y);
```

qui modifie la tablette `t` de sorte à manger le carré de chocolat en position (x, y) ainsi que tous ceux qui sont en dessous de lui et à sa droite.

3. Un coup dans une position donnée est légal s'il ordonne de manger un carré de chocolat qui existe encore. Écrire une fonction

```
1 int est_legal(Position *pos, Coup *coup);
```

qui retourne 1 si le coup `coup` est légal dans la position `pos` et 0 sinon.

4. La partie est terminée lorsque le carré de chocolat empoisonné vient d'être mangé. Dans ce cas, c'est le joueur qui vient de jouer qui a perdu et l'autre qui a gagné. Écrire une fonction

```
1 int est_jeu_termine(Position *pos, Joueur *joueur_gagnant);
```

qui retourne 1 si la partie représentée par la position `pos` est terminée et 0 sinon. Dans le cas où la partie est terminée, la fonction affecte à la variable pointée par `joueur_gagnant` le joueur qui gagne la partie.

5. Écrire une fonction

```
1 void jouer_coup(Position *pos, Coup *coup);
```

qui joue le coup `coup` dans la position `pos`. Il ne faut pas oublier de modifier le champ qui contient le joueur dont c'est le tour de jouer.

Exercice 4. (Assemblage du jeu)

1. Écrire une fonction

```
1 void afficher_position(Position *pos);
```

qui affiche en utilisant la bibliothèque MLV la position `pos`.

2. Écrire une fonction

```
1 Coup lire_coup(Position *pos);
```

qui attend un clic de l'utilisateur sur la tablette de chocolat dans la fenêtre graphique et calcule et retourne le coup spécifié par l'utilisateur. Si l'utilisateur ne clique pas sur la tablette, ou bien clique sur un carré de chocolat déjà mangé, la fonction ne fait rien et attend de traiter le prochain clic.

3. Utiliser les fonctions précédentes pour construire le programme `Chomp.c` qui permet de jouer au *Chomp*. Utiliser pour cela l'algorithme suivant :

(1) lire la dimension de la tablette de chocolat sur laquelle le jeu va se dérouler (passée en argument au programme);

(2) initialiser la position `pos`;

(3) tant que la position `pos` ne représente pas une partie terminée :

(a) afficher la position `pos` sur la fenêtre graphique ;

(b) lire un coup `c` sur la fenêtre graphique ;

(c) jouer le coup `c` dans la position `pos` ;

(4) afficher le numéro du joueur gagnant.