



Projet du module Algorithmie I

filler

Résumé: Saurez-vous avoir être le meilleur ? :-)

Table des matières

I	Préambule	2
II	Règle du jeu	3
II.1	Le Filler	3
II.2	Le Plateau	4
II.3	Les pièces	4
III	Le Sujet	5
III.1	Le Player	5
III.2	Multi Players	6
III.3	Déroulement d'une partie	7
III.4	Championnat	8
IV	Consignes	9
IV.1	Le rendu	9
IV.2	Fonctions autorisées	9
V	Notation	10

Chapitre I

Préambule

L'Orgueil (*superbia* en latin) : attribution à ses propres mérites de qualités ou de comportements qui sont des dons de Dieu (intelligence, vertus, etc.).

L'Avarice (*avaritia* en latin) : accumulation des richesses recherchées pour elles-mêmes.

L'Envie (*invidia* en latin) : la tristesse ressentie face à la possession par autrui d'un bien, et la volonté de se l'approprier par tout moyen et à tout prix (à ne pas confondre avec la Jalousie).

La Colère (*ira* en latin) : produit des excès en paroles ou en actes : insultes, violences, meurtre.

La Luxure (*luxuria* en latin) : plaisir sexuel recherché pour lui-même.

La Gourmandise (*gula* en latin) : ce n'est pas tant la gourmandise au sens moderne qui est blâmable que la gloutonnerie, cette dernière impliquant davantage l'idée de démesure et d'aveuglement que le mot gourmandise. Par ailleurs, on constate que dans d'autres langues ce péché n'est pas désigné par un mot signifiant « gourmandise » (*gluttony* en anglais, par exemple).

La Paresse, anciennement l'acédie (*acedia* en latin). Le Catéchisme de l'Église catholique définit l'acédie, terme disparu du langage courant, comme « une forme de dépression due au relâchement de l'ascèse ». Il s'agit en effet de paresse morale. L'acédie, c'est un mal de l'âme qui s'exprime par l'ennui, l'éloignement de la prière, de la pénitence et de la lecture spirituelle.

(L'Ingratitude (*ingratitude* en latin) : manquement grave au devoir de reconnaissance pour des bienfaits reçus.)

Chapitre II

Règle du jeu

II.1 Le Filler

- Deux adversaires s'affronte dans ce jeu. Ils jouent chacun leur tour.
- L'objectif est de gagner le plus de point en remplissant le plateau de jeu avec le plus de pièces possible.
- Le plateau est défini par X colonnes et N lignes, il fera donc $X*N$ cases.
- Les pièces vous seront fournies tour par tour.
- Une pièce est définie par X colonnes et N lignes. Dans chaque pièce une forme de une ou plusieurs cases est représentée.
- Pour pouvoir poser une pièce, il faut qu'une case de la forme recouvre une case, et une seule, d'une forme précédemment posée.
- La forme doit rentrer intégralement dans le plateau.
- Le plateau contient une première forme pour initier la partie.
- La partie s'arrête à la première erreur : dès qu'une pièce ne peut plus être posée ou a été mal posé.

II.2 Le Plateau

- Un plateau 14 x 30.

```

Plateau 14 30:
012345678901234567890123456789
000 .....
001 .....
002 .....
003 .....
004 .....
005 .....
006 .....
007 .....
008 .....0.....
009 .....
010 .....
011 .....
012 .....
013 .....

```

II.3 Les pièces

- Un exemple de pièce 4 x 7.

```

Piece 4 7:
...*...
...*...
...*...
..****.

```

- Un exemple de pièce 4 x 5.

```

Piece 4 5:
.**..
.**.
.*..
....

```

- Un exemple de pièce 3 x 6.

```

Piece 3 6:
.****.
**....
*.....

```

Chapitre III

Le Sujet

III.1 Le Player

- L'exécutable permettant de jouer au **filler** vous sera accessible en pièce jointe du sujet.
- (Il marche sous ruby 2.0 et il se trouve ici :
/Volumes/DATA/nfs/sgoinfre/goinfre/filler :-))
- Pour ce projet vous allez devoir réaliser un joueur de **filler**. Votre but est de gagner :
 - Il lira le plateau et les pièces sur l'entrée standard.
 - A chaque tour le filler réécrit la carte et une nouvelle pièce à placer.
 - Le joueur devra écrire sur la sortie standard les coordonnées pour placer la pièce.
 - Le format sera sous la forme suivante : "X Y\n".
 - Chaque pièce placée fera gagner des points.

```
Plateau 14 30:
012345678901234567890123456789
000 .....
001 .....
002 .....
003 .....
004 .....
005 .....
006 .....
007 .....
008 .....0.....
009 .....
010 .....
011 .....
012 .....
013 .....
Piece 4 7:
...*...
...*...
...*...
..***..
```



Attention vous devez écrire les coordonnées de la pièce et non de la forme.

III.2 Multi Players

- Numéro du joueur :
 - Les deux premières lignes du programme du **filler** seront au format suivant :

```
$$$ exec pPLAYER_NUMBER : [PLAYER_NAME]
```
 - Le **filler** n'enverra que la ligne concernant votre programme. Vous devrez récupérer votre numéro de joueur.
 - Si vous êtes joueur 1 votre programme sera représenté par les caractères “o” et “O”. Si vous êtes le joueur 2, votre programme sera représenté par les caractères “x” et “X”. La première étape sera de récupérer votre numéro de joueur.
 - L’usage de la lettre minuscule (“x” ou “o”) met en avant la dernière pièce posée. Au tour suivant, elle ne sera plus la dernière donc elle sera représenté par les lettres majuscules (“X” ou “O”).
- Déroulement d’une partie :
 - A chaque tour, le **filler** enverra la carte mise à jour et une nouvelle pièce au joueur concerné.
 - Le joueur concerné écrira sur la sortie standard les coordonnées pour placer sa pièce.
 - Le **filler** enverra la carte et une nouvelle pièce au second joueur ...

III.3 Déroulement d'une partie

- Voici un exemple de déroulement d'une partie :

```
$>./filler -p1 user1 -p2 user2 -v -f samples/w1.flr
$$$ exec p1 : [user1]
$$$ exec p2 : [user2]
Plateau 14 30:
012345678901234567890123456789
000 .....
001 .....
002 .....
003 .....
004 .....X.....
005 .....
006 .....
007 .....0...
008 .....
009 .....
010 .....
011 .....
012 .....
013 .....
Piece 3 6:
.****.
**....
*....
<got (0) : [7 24] (7,24)
Plateau 14 30:
012345678901234567890123456789
000 .....
001 .....
002 .....
003 .....
004 .....X.....
005 .....
006 .....
007 .....oooo.
008 .....oo...
009 .....o...
010 .....
011 .....
012 .....
013 .....
Piece 3 8:
.....*.
.....**
.....*
<got (X) : [4 0] (4,0)
Plateau 14 30:
012345678901234567890123456789
000 .....
001 .....
002 .....
003 .....
004 .....x.....
005 .....xx.....
006 .....x.....
007 .....0000.
008 .....00...
009 .....0...
010 .....
011 .....
012 .....
013 .....
[...]
```

```
== X fin : 175 [1018918090]
== 0 fin : 168 [1018918090]
```


III.4 Championnat

- Un championnat sera organisé pendant ce projet. Les gagnants du concours seront récompensés !

Chapitre IV

Consignes

IV.1 Le rendu

- Vous devez rendre, à la racine de votre dépôt de rendu, un fichier **auteur** contenant votre login suivi d'un retour à la ligne :

```
$>cat -e auteur
xlogin$
$>
```

- L'exécutable doit s'appeller **filler**.
- Il doit se trouver à la racine du dépôt.
- Vous devez rendre un Makefile.
- Votre Makefile devra compiler le projet, et doit contenir les règles habituelles. Il ne doit recompiler le programme qu'en cas de nécessité.
- Si vous êtes malin et que vous utilisez votre bibliothèque **libft** pour votre player, vous devez en copier les sources et le **Makefile** associé dans un dossier nommé **libft** qui devra être à la racine de votre dépôt de rendu. Votre **Makefile** devra compiler la librairie, en appelant son **Makefile**, puis compiler votre projet.
- Votre projet doit être à la Norme.
- Vous devez gérer les erreurs de façon stricte. En aucun cas votre programme ne doit quitter de façon inattendue (Segmentation fault, etc...).

IV.2 Fonctions autorisées

- write, malloc, free, read, perror, strerror.

Chapitre V

Notation

- La notation s'effectue en deux temps :
 - Nous commençons par tester **las funcionalidades** (sur 15 points). Votre programme doit fonctionner.
 - La deuxième partie est **la competición**.
 - Pour participer au concours il faut que la première partie soit correcte.
 - Tous les players ayant validé l'intégralité de la première partie seront en compétition dans le championnat.
 - Le meilleur player remportera l'intégralité des points de ce concours.
 - Les projets suivants remporteront des points en fonction de leur classement.

Bon courage, BON FIGHT à tous !