

IFT1025 - Travail Pratique 1 (partie 1 / 2)

Concepts appliqués : Programmation orientée objet, algorithmique graphique 2D, héritage, développement d'application de taille moyenne

1 Contexte

Pour ce premier travail pratique, vous devrez concevoir un petit programme pour faire des dessins sur une ligne de commande, par exemple :

[illegible]

2 Spécification

Le programme s'exécute interactivement et doit répondre à certaines commandes. Un exemple d'exécution serait :

```
init 30 15
car .
ajouter rectangle 0 0 30 15
car #
ajouter rectangle 5 5 20 10
dessiner
```

qui va successivement :

1. Initialiser la surface de dessin à une surface de largeur 30 et de hauteur 15 (donc 15 lignes de 30 caractères)
2. Définir le caractère de dessin comme étant .
3. Ajouter un rectangle à la position (0,0) large de 30 caractères et haut de 15 caractères, qui sera dessiné avec des . (les coordonnées (x,y) sont données par rapport au coin supérieur gauche, qui a la position (0,0)).
4. Définir le caractère de dessin comme étant #
5. Ajouter un rectangle à la position (5,5) de taille 20 par 10 (dessiné avec des #)
6. Dessiner le tout sur la console, ce qui donnerait finalement :

```
.....
.....
.....
.....
.....
.....#####.....
.....#####.....
.....#####.....
.....#####.....
.....#####.....
.....#####.....
.....#####.....
.....#####.....
.....#####.....
.....#####.....
.....#####.....
```

2.1 Commandes

2.2 `init LARGEUR HAUTEUR`

Va initialiser la surface de dessin avec une surface vide de taille `LARGEUR` par `HAUTEUR`, par exemple :

```
init 70 25
```

définit que la surface de dessin sera de 25 lignes de 70 caractères chacune.

2.3 `car CARACTERE`

Va définir le caractère de dessin actuel à `CARACTERE`, par exemple :

```
car #
```

Définit que le caractère utilisé sera `#` pour les prochaines formes ajoutées.

2.4 `ajouter FORME ...`

Va ajouter une nouvelle forme à la surface de dessin. Les paramètres dépendent de la forme à ajouter (voir plus bas pour les différentes formes disponibles).

On pourrait par exemple ajouter un rectangle au dessin avec la commande :

```
ajouter rectangle 5 5 20 10
```

La forme ajoutée sera dessinée avec le caractère de dessin actuel. Si on faisait quelque chose comme :

```
car #
ajouter rectangle 5 5 20 10
car 8
ajouter rectangle 2 1 5 5
ajouter rectangle 4 2 8 7
```

Ça ajouterait d'abord un rectangle qui serait dessiné avec des `#`, puis ça ajouterait deux rectangles dessinés avec des `8`.

S'il n'est pas spécifié, le caractère de dessin est initialement un dièse (`#`).

2.5 dessiner

Va simplement afficher sur la console l'ensemble des formes ajoutées. Les formes sont dessinées dans l'ordre dans lequel elles sont ajoutées.

Par exemple la suite de commandes : Affichera sur la console :

init 10 10
car .	.8888.....
ajouter rectangle 0 0 10 10	.8888.....
car #	.8888##...
ajouter rectangle 3 3 4 4	.8888##...
car 8	...####...
ajouter rectangle 2 2 4 4	...####...
dessiner

Ici, le troisième rectangle ajouté s'affiche "devant" les autres.

2.6 brasser

Va déplacer toutes les formes de ± 1 caractère sur l'axe des X, des Y ou les deux, où le déplacement positif ou négatif et les axes sont choisis de manière aléatoire et différemment pour chaque forme du dessin.

Par exemple la suite de commandes : Pourrait donner :

init 10 10	8888
car .	8888.....
ajouter rectangle 0 0 10 10	8888.....
car #	8888.....
ajouter rectangle 3 3 4 4	..####...
car 8	..####...
ajouter rectangle 2 2 4 4	..####...
brasser	..####...
dessiner

Dans cet exemple, le premier rectangle (.) a été déplacé de (0, +1) par rapport à sa position originale, le deuxième (#) a été déplacé de (-1, 0) et le troisième (8) a été déplacé de (-1, -1).

Notez que même si une forme peut avoir des coordonnées qui dépassent les bornes de la zone de dessin, on n'affiche toujours qu'une surface de la taille de la surface de dessin.

Dans le cas où il n'y a aucune forme en un point (par exemple, à droite du rectangle sur la première ligne), on affiche un espace.

2.7 renverser

Renverse verticalement le dessin.

Par exemple :

```
init 10 10
car .
ajouter rectangle 0 0 10 10
car #
ajouter rectangle 3 3 4 4
car 8
ajouter rectangle 2 2 4 4
renverser
dessiner
```

Donnerait le résultat :

```
.....
.....
.....
...####...
...####...
.8888##...
.8888##...
.8888.....
.8888.....
.....
```

Notez qu'utiliser **renverser** deux fois de suite est équivalent à ne rien faire.

2.8 Formes dessinables

Plusieurs types de formes peuvent être ajoutées au dessin. On vous rappelle que les coordonnées (x, y) sont données par rapport au coin supérieur gauche, qui a la position $(0, 0)$.

2.8.1 ajouter rectangle X Y LARGEUR HAUTEUR

Ajoute un rectangle de taille **LARGEUR** par **HAUTEUR** dont le coin supérieur gauche est sur le point (X, Y) .

Par exemple :

```
init 10 10
car .
ajouter rectangle 0 0 10 10
dessiner
```

Donne le résultat :

```
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
```

2.8.2 ajouter carre X Y COTE

Ajoute un carré dont le coin supérieur gauche est à la position (X, Y) dont la taille d'un côté est de COTE caractères.

La commande `ajouter carre 0 0 10` aura donc le même effet que `ajouter rectangle 0 0 10 10`.

2.8.3 ajouter ligne X1 Y1 X2 Y2

Ajoute une ligne allant du point $(X1, Y1)$ inclusivement jusqu'au point $(X2, Y2)$ inclusivement.

Par exemple :

```
init 10 10
car .
ajouter ligne 0 0 9 9
dessiner
```

Donne le résultat :

```
#.....
.#.....
..#.....
...#.....
....#.....
.....#....
.....#...
.....#..
.....#.
.....#
```

2.8.4 ajouter cercle X Y RAYON

Ajoute un cercle de rayon RAYON dont le centre est à la position (X, Y) .

Par exemple :

```
init 10 10
car .
ajouter rectangle 0 0 10 10
car #
ajouter Cercle 4 4 3
dessiner
```

Donne le résultat :

```
.....
...###...
..#...#...
.#....#..
.#....#..
.#....#..
..#...#...
...###...
.....
.....
```

2.8.5 ajouter caractere X Y CARACTERE

Ajoute le caractère `CARACTERE` sur la surface à la position (X, Y) .

Par exemple :

```
init 8 11
car .
ajouter rectangle 0 0 8 11
car #
ajouter lettre 0 0 B
dessiner
```

Donne le résultat :

```
.....
.#####.
..#...#.
..#...#.
..#...#.
..#...#.
..#####.
..#...#.
..#...#.
..#...#.
..#...#.
.#####.
.....
```

Note: Les représentations des caractères possibles vous seront fournies plus tard. Vous n'aurez pas à coder chaque lettre à la main.

2.8.6 ajouter texte X Y TEXTE

Ajoute la séquence de caractères `TEXTE` sur la surface à la position (X, Y) .

Par exemple :

```
init 30 11
car #
ajouter texte 0 0 Java
dessiner
```

Donne le résultat :

```
###
#
#
#   ####   #   #   ####
#       #   #   #       #
#   #####   #   #   #####
#   #   #   #   #   #   #
#   #   #   #   #   #   #
###   ### #   #   ### #
```