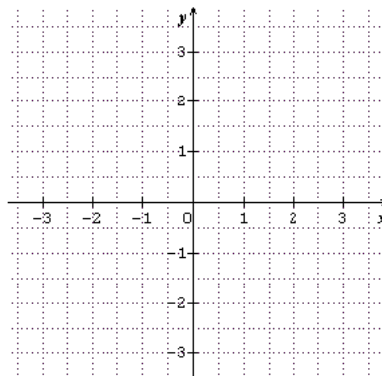


Cahier des charges du TP n°5 de C++ Adrien Lepic et Quentin Vecchio (B3323)

INSA Lyon - 3IF
Vendredi 5 Février 2016

I - Généralités

Le logiciel doit pouvoir gérer la construction, la modification et la suppression de figure simple comme des rectangles, des segments et des polygones convexes. Chaque figure est identifiée par un nom unique qui peut être composé de lettres et/ou chiffres. Le logiciel doit également pouvoir gérer des fonctions comme le déplacement de figure, savoir si un point appartient à une figure. Il doit également pouvoir gérer des ensembles (union et intersection) de figure. Chaque figure est composée de coordonnées entière. Nous avons choisi le repère orthogonale ci-dessous.

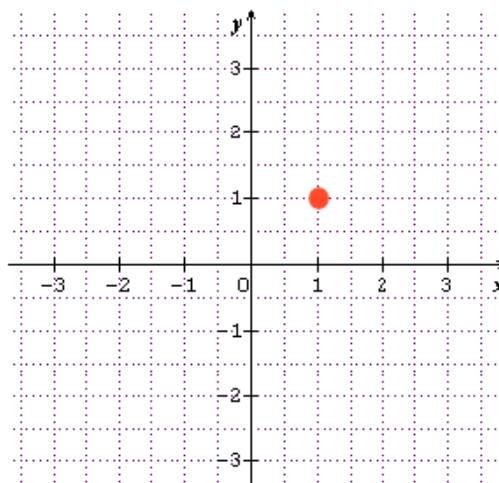


II - Les différentes figures

A - Point

Un point est représenté par ses coordonnées dans un repère orthogonale.

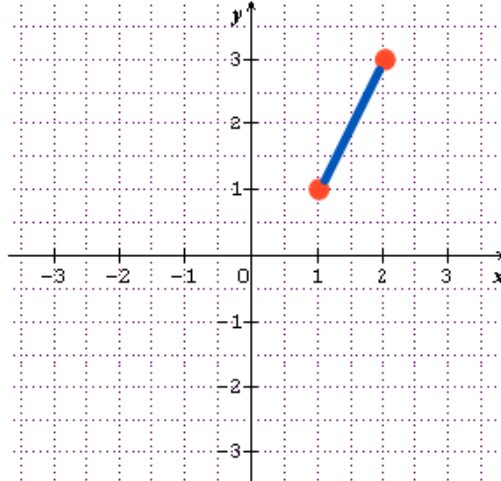
Affichage : X1 Y1 X2 Y2



B - Segment

Un segment est représenté par deux points et son nom. On doit pouvoir le déplacer et tester si un point est sur le segment.

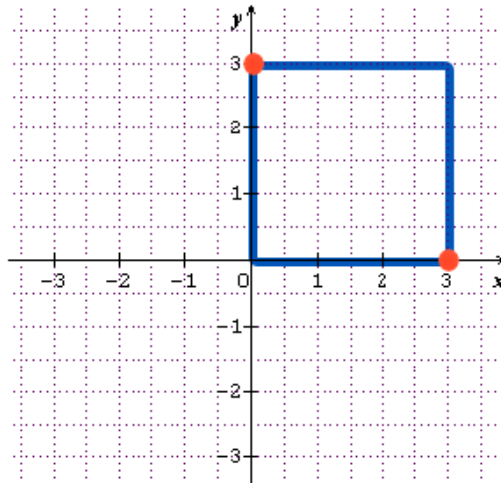
Affichage : S nom X1 Y1 X2 Y2



C - Rectangle

Un rectangle est représenté par deux points et son nom. On doit pouvoir le déplacer, tester si un point appartient au rectangle, connaître sa longueur et sa largeur. Son premier point est le point en haut à gauche et son deuxième point est celui en bas à droite.

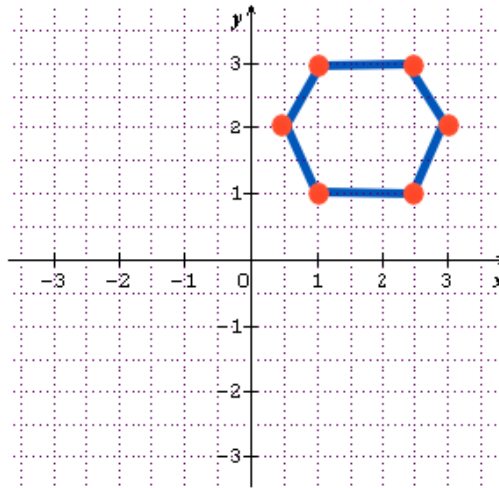
Affichage : R nom X1 Y1 X2 Y2



D - Polygone

Un polygone est représenté par ensemble de points et par son nom. On doit pouvoir le déplacer, tester si un point appartient au polygone. Cependant pour pouvoir être exploitable, le polygone doit être convexe.

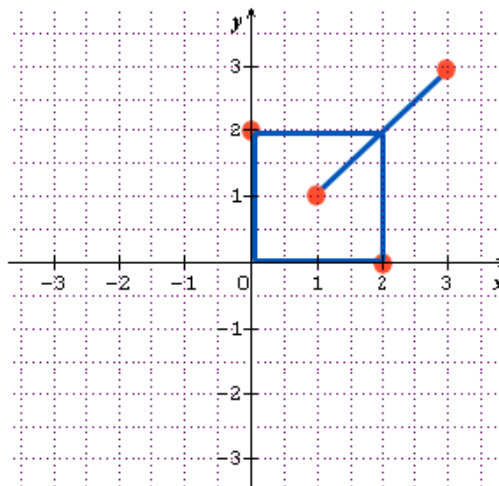
Affichage : PC nom X1 Y1 X2 Y2 X3 Y3 X4 Y4 X5 Y5



E - Union

Un union est représenté par ensemble de figures et par son nom. On doit pouvoir le déplacer, tester si un point appartient à une figure ou plusieurs figures contenu dans l'union. Pour pouvoir ajouter une figure dans l'ensemble, elle doit au préalable déjà exister. Ensuite elle sera copier dans l'union.

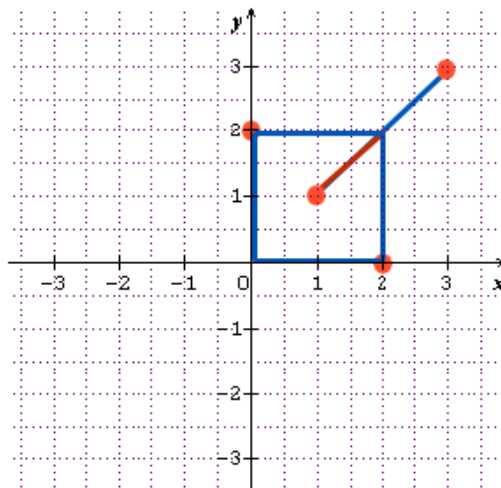
Affichage : OR nom nombreDeFigure
TypeFigure nom X1 Y1 ...



F - Intersection

Une intersection est représenté par ensemble de figures et par son nom. On doit pouvoir le déplacer, tester si un point appartient à toutes les figures contenu dans l'intersection. Pour pouvoir ajouter une figure dans l'ensemble, elle doit au préalable déjà exister. Ensuite elle sera copier dans l'union.

Affichage : OI nom nombreDeFigure
TypeFigure nom X1 Y1 ...



G - Dessin

Un Dessin est représenté par ensemble de figures, unions et intersections. C'est le conteneur principale du programme. On doit donc pouvoir lui ajouter une figure, mais aussi lui en supprimer. On peut également le déplacer et tester l'existence d'un point. On peut aussi afficher toutes les figures qui le compose.

III - Interactions

On doit pouvoir interagir avec le programme sur l'entrée et la sortie standard. Il existe donc des commandes permettant l'exécution de fonctions, ou d'ajout ou suppression de figure. De même, le programme doit répondre à l'utilisateur.

A - Création d'un segment

Commande : S nom X1 Y1 X2 Y2

Réponse : OK
ERR

Si la réponse est OK alors la création s'est bien passé, sinon c'est que le nom existe déjà.

B - Création d'un rectangle

Commande : R nom X1 Y1 X2 Y2

Réponse : OK
ERR

Si la réponse est OK alors la création s'est bien passé, sinon c'est que le nom existe déjà.

Attention : X1 Y1 sont les coordonnées du point en haut à gauche, X2 Y2 sont les coordonnées du point en bas à droite.

C - Création d'un polygone

Commande : PC nom X1 Y1 X2 Y2 ... Xn Yn

Réponse : OK
 ERR

Si la réponse est OK alors la création s'est bien passé, sinon c'est que le nom existe déjà ou alors que le polygone n'est pas convexe.

D - Création d'un union

Commande : OR nom nomFig1 ... nomFigN

Réponse : OK
 ERR

Si la réponse est OK alors la création s'est bien passé, sinon c'est que le nom existe déjà ou alors que l'une des figures n'existe pas.

E - Création d'une intersection

Commande : OI nom nomFig1 ... nomFigN

Réponse : OK
 ERR

Si la réponse est OK alors la création s'est bien passé, sinon c'est que le nom existe déjà ou alors que l'une des figures n'existe pas.

F - Test d'appartenance

Commande : HIT nom X1 Y1

Réponse : YES
 NO

Si la réponse est YES alors le point appartient à la figure, sinon c'est que le nom n'existe pas ou alors que le point n'appartient pas à la figure.

G - Suppression de figures

Commande : DELETE nomFig1 nomFig2 ... nomFig3

Réponse : OK
 ERR

Si la réponse est OK alors les figures ont été supprimé, sinon c'est que un des noms n'existe pas.

H - Suppression du dessin

Commande : CLEAR

Réponse : OK

I - Affichage

Commande : LIST

Réponse : Liste avec toutes les figures

J - Déplacement d'une figure

Commande : MOVE nom X1 Y1

Réponse : OK
ERR

Si la réponse est OK alors la figure a été déplacé, sinon c'est que le nom n'existe pas.

K - Sauvegarde

Commande : SAVE nomFichier

Réponse : OK
ERR

Si la réponse est OK alors la sauvegarde a été faite, sinon c'est qu'il y a une erreur avec le fichier.

L - Chargement

Commande : LOAD nomFichier

Réponse : OK
ERR

Si la réponse est OK alors le chargement a été fait, sinon c'est qu'il y a une erreur avec le fichier.

M - Undo

Commande : UNDO

Réponse : OK

N - Redo

Commande : REDO

Réponse : OK

O - Quitter

Commande : EXIT

IV - Sauvegarde et chargement

Afin de réutiliser au maximum notre code, nous avons choisi de formater les fichiers de sauvegarde et chargement comme les commandes d'exécution de notre programme. Par exemple voici le fichier de sauvegarde d'un dessin avec deux rectangles et un union de deux segments :

```
R rec1 2 2 0 4  
R rec2 2 2 0 7  
OR union1 2  
S seg1 1 1 4 4  
S seg2 2 2 5 5
```

La seule différence se trouve au niveau des unions et des intersections qui ont chacun un formatage différent de leur commande d'exécution.