

# Cahier des charges détaillé TP4 C++

B3330

CHAPELLE Victoire

SOTIR Paul-Emmanuel

**Problème** : implémenter une application console qui édite des formes géométriques et les manipule. Les formes géométriques à gérer sont le segment, le rectangle et les polygones convexes (c'est-à-dire dont chaque angle est inférieur à  $180^\circ$ ). Aucune implémentation d'une interface graphique n'est demandée. Les différentes opérations que l'on pourra faire sur l'application sont détaillées ci-dessous.

Les coordonnées seront considérées comme des entiers signés et l'axe des X sera 'vers la droite' et celui des Y 'vers le haut' (cette orientation est précisée car elle influe sur la validation des rectangles créés : voir opération '2')

1) Ajout d'un segment

**Commande** : S Name X1 Y1 X2 Y2

**Réponses** : OK ou ERR

**Description** : Permet de créer un segment entre les points 1 et 2 de coordonnées respectifs (X1,Y1) et (X2,Y2). Le segment prendra pour nom « Name », une string avec lettres et/ou chiffres sans séparateur. Le nom est unique et ne peut pas avoir été utilisé précédemment pour nommer une quelconque autre forme (mis à part si cette forme a été supprimée ou fait maintenant partie d'une intersection ou union). Si les points 1 et 2 sont confondus ou si le nom est déjà employé, la réponse retournée sera ERR avec un commentaire spécifiant l'erreur. Si la commande est bien exécutée, la réponse sera OK.

2) Ajout d'un rectangle

**Commande** : R Name X1 Y1 X2 Y2

**Réponse** : OK ou ERR

**Description** : Permet de créer un rectangle défini par son point en haut à gauche (X1,Y1) et son point en bas à droite (X2,Y2). Le rectangle prendra pour nom « Name », un string avec lettres et/ou chiffres sans séparateur. Le nom est unique et ne peut pas avoir été utilisé précédemment pour nommer une quelconque autre forme (mis à part si cette forme a été supprimée ou fait maintenant partie d'une intersection ou union). Si les points 1 et 2 ne sont pas respectivement les coins supérieurs gauche et inférieur droit ou si le nom est déjà employé, la réponse retournée sera ERR. Si la commande est bien exécutée, la réponse sera OK.

3) Ajout d'un polygone convexe

**Commande** : PC Name X1 Y1 X2 Y2 ... Xn Yn

**Réponse** : OK ou ERR

**Description** : Permet de créer un polygone convexe (c'est-à-dire dont tous les angles sont inférieurs à  $180^\circ$  de minimum 3 côtés. Le polygone prendra pour nom « Name », un string avec lettres et/ou chiffres sans séparateur. Le nom est unique et ne peut pas avoir été utilisé précédemment pour nommer une quelconque autre forme (mis à part si cette forme a été supprimée ou fait maintenant partie d'une intersection ou union). Si deux points sont confondus, si le polygone n'est pas convexe ou si le nom a déjà été employé la réponse retournée sera ERR. Si la commande est bien exécutée, la réponse sera OK.

4) Opération de réunion

**Commande** : OR Name Name1 ... NameN

**Réponse** : OK ou ERR

**Description** : Permet de créer une forme qui symbolisera la réunion de deux formes géométriques ou plus, spécifiées dans la commande par leur nom. La réunion prendra pour nom « Name », un string avec lettres et/ou chiffres sans séparateur. Le nom est unique et ne peut pas avoir été utilisé précédemment pour nommer une quelconque autre forme (mis à part si cette forme a été supprimée ou fait maintenant partie d'une intersection ou union). Si un ou plusieurs des noms entrés en paramètres n'existe pas ou si le nom donné à la figure est déjà employé, la réponse retournée sera ERR. Si la commande est bien exécutée, la réponse sera OK.

5) Opération d'intersection

**Commande** : OI Name Name1 ... NameN

**Réponse** : OK ou ERR

**Description** : Permet de créer une forme qui symbolisera l'intersection de deux formes géométriques ou plus, spécifiées dans la commande par leur nom. L'intersection prendra pour nom « Name », un string avec lettres et/ou chiffres sans séparateur. Le nom est unique et ne peut pas avoir été utilisé précédemment pour nommer une quelconque autre forme (mis à part si cette forme a été supprimée ou fait maintenant partie d'une intersection ou union). Si un ou plusieurs des noms entrés en paramètres n'existe pas ou si le nom donné à la figure est déjà employé, la réponse retournée sera ERR. Si la commande est bien exécutée, la réponse sera OK.

6) Opération d'appartenance

**Commande** : HIT Name X Y

**Réponse** : YES ou NO

**Description** : la commande renvoie YES si le point de coordonnées (X,Y) est contenu dans la forme « Name ». Le point est contenu dans la forme « Name » s'il est dans la forme ou sur son contour pour les rectangles et polygones convexes et sur le segment pour les segments. Elle renvoie NO si le point ne remplit pas les conditions ci-dessus.

7) Suppression

**Commande** : DELETE Name1 ... Name2

**Réponse :** OK ou ERR

**Description :** La commande permet de supprimer les formes (segments, rectangles, polygones, intersection, réunion) désignées par leur nom. Si un des noms ne correspond à aucune figure, aucun objet n'est supprimé et la réponse retournée est ERR. Si la commande est bien exécutée la réponse est OK. Les formes ne seront supprimées que si tous les noms spécifiés sont corrects.

8) Déplacer un objet

**Commande :** MOVE Name dX dY

**Réponse :** OK ou ERR

**Description :** La commande permet de déplacer une forme ou une figure, désignée par son nom « Name », de dX sur l'axe des abscisses et dY sur l'axe des ordonnées. Si le nom ne désigne aucun objet, la réponse retournée est ERR, sinon OK.

9) Enumération

**Commande :** LIST

**Réponse :** Une description facilement lisible des formes. Par exemple :

\* poc3 : { (0, 0); (0, 1); (1, 1); (1, 0) }  
\* seg : { (0, 2); (2, 3) }  
\* rzc : { (0, 0); (1, 0) }  
\* poc1 : { (0, 0); (0, 10); (10, 8); (8, 4) }  
\* int : Intersection de 5 formes

**Description :** Affiche les formes en haut de la hiérarchie (formes n'étant pas dans une intersection ni une union). Si aucunes formes n'est présente, la commande affiche « Aucunes formes ».

10) Annuler la dernière opération

**Commande :** UNDO

**Réponse :** OK ou ERR

**Description :** Cette commande permet d'annuler la dernière opération qui a entraînée des modifications sur le modèle telle que les déplacements, la suppression, l'insertion d'objet ou le chargement d'un fichier. Cette commande permet d'annuler 20 opérations d'affilées maximum. Si la commande UNDO a été utilisée pour annuler plus de 20 opérations en arrière, la réponse sera ERR. Si la commande est bien exécutée, la réponse est OK.

11) Reprendre la dernière modification

**Commande :** REDO

**Réponse :** OK ou ERR

**Description** : Cette commande permet de revenir à l'état précédent la commande UNDO. Cette opération ne peut être effectuée (sans ERR) que si une commande UNDO a été appelée et qu'aucune commande qui entraîne des modifications (un déplacement, une suppression, une insertion d'objet) n'a été appelée après. Si la commande UNDO n'a pas été appelée ou qu'une modification du modèle a eu lieu après, la réponse sera ERR. Si la commande est bien exécutée, la réponse est OK.

12) Charger en mémoire un état

**Commande** : LOAD filename

**Réponse** : OK ou ERR

**Description** : Permet de charger un fichier au format XML spécifié (de nom « filename »). Si le nom du fichier est invalide, si le contenu est invalide, ou si le fichier ne peut pas être ouvert, ERR est affiché et l'état de l'application reste cohérent. Cette commande est annulable avec UNDO.

13) Sauvegarder le modèle courant

**Commande** : SAVE filename

**Réponse** : OK ou ERR

**Description** :

14) Vider le modèle actuel

**Commande** : CLEAR

**Réponse** : OK

**Description** : Supprime toutes les formes composant l'état courant de l'application. Si la commande est bien exécutée, la réponse est OK.

15) Fermer l'application

**Commande** : EXIT

**Réponse** : OK

**Description** : ferme l'application

16) Activer les messages plus détaillés

**Commande** : EN\_ERROR\_MESS

**Réponse** : OK

**Description** : Active l'affichage de message plus précis au lieu de « ERR » lors d'appel à d'autres commandes.

17) Désactiver les messages plus détaillés

**Commande :** DIS\_ERROR\_MESS

**Réponse :** OK

**Description :** Désactive l’affichage de message plus précis au lieu de « ERR » lors d’appel à d’autres commandes.