Cahier des charges détaillé TP4 C++

**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°). Aucune implémentation d’une interface graphique n’est demandée. Les différentes opérations que l’on pourra faire sur l’application sont détaillées ci-dessous.

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 », 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. 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.

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. 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.

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° 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. 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 avec un commentaire spécifiant l’erreur. Si la commande est bien exécutée, la réponse sera OK.

Opération de réunion

**Commande**: OR Name Name1 … NameN

**Réponse**: OK ou ERR

**Description**: Permet de créer une figure 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 ou figure. 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 avec un commentaire spécifiant l’erreur. De même, si la même forme est passée deux fois en paramètre (réunion d’une forme avec elle-même) la réponse retournée sera ERR. Si la commande est bien exécutée, la réponse sera OK.

Opération d’intersection

**Commande**: OI Name Name1 … NameN

**Réponse**: OK ou ERR

**Description**: Permet de créer une figure 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 ou figure. 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 avec un commentaire spécifiant l’erreur. De même, si la même forme est passée deux fois en paramètre (intersection d’une forme avec elle-même) la réponse retournée sera ERR. Si la commande est bien exécutée, la réponse sera OK.

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.

Suppression

**Commande**: DELETE Name1 … Name2

**Réponse**: OK ou ERR

**Description**: La commande permet de supprimer les formes (segments, rectangles, polygones) ou figures (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.

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.

Enumération

**Commande**: LIST

**Réponse**: Affichage sous le format d’un …

**Description**: Cette commande permet d’afficher une description de tous les objets (segments, rectangles, polygones, intersections et réunions) déjà créés. L’affichage se fera sous la forme du format texte choisi pour enregistrer le modèle en mémoire. Ce format est ….

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 aucune opération n’a eu lieu ou 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.

Reprendre la dernière modification

**Commande**: REDO

**Réponse**: OK ou ERR

**Description**: Cette commande permet de revenir au modèle précédent la commande UNDO. Cette opération ne peut être effectuée 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 entre. Si la commande UNDO n’a pas été appelée ou qu’une modification du modèle a eu lieu entre, la réponse sera ERR. Si la commande est bien exécutée, la réponse est OK.

Charger en mémoire un modèle

**Commande**: LOAD filename

**Réponse**: OK ou message d’erreur

**Description**: Cette commande permet de charger un fichier qui contient un ensemble d’objets décrits sous un format texte. Le format choisi est …..

Sauvegarder le modèle courant

**Commande**: SAVE filename

**Réponse**: OK ou message d’erreur

**Description**: Cette commande permet d’enregistrer en mémoire, sous le format défini ci-dessus, le modèle courant de l’application. Par défaut, le fichier est enregistré dans le même répertoire que l’exécutable. Un chemin pour enregistrer le fichier dans un autre répertoire peut être mis dans le « filename ». Si la commande est bien exécutée, la réponse est OK.

Vider le modèle actuel

**Commande**: CLEAR

**Réponse**: OK ou ERR

**Description**: Cette commande permet de supprimer tous les objets qui composent le modèle. Si le modèle est vide ou si la commande est bien exécutée et que tous les objets ont été supprimés, la réponse est OK. Si la commande est mal exécutée et que tous les objets ne sont pas supprimés, la réponse retournée sera ERR.

Fermer l’application

**Commande**: EXIT

**Réponse**: Aucune

**Description**: Cette commande permet de fermer l’application sans sauvegarder le modèle courant.