

TP C++ 4 : Gestionnaire de formes

Cahier des charges

B3334 : VICENTE Olivier, SORIN Nicolas
Rendu : Vendredi 5 février 2016

L'application à réaliser est un gestionnaire de formes géométriques fonctionnant en ligne de commande. Cet application permettra de créer un modèle constitué de différents objets simples ou complexes. Différentes opérations pourront être réalisées sur ces objets. Il faudra aussi mettre en place un système de sauvegardes de modèles qui pourront ainsi être chargés indépendamment par la suite à la demande de l'utilisateur.

1 – Types d'objets

Chacun des objets créé est défini par un nom sous forme d'une chaîne de caractère contenant uniquement des lettres et des chiffres sans séparateurs. Chaque nom est unique (il est impossible de créer deux objets de même nom).

a) Objets simples

- Segment : Un segment est caractérisé par deux points (X1,Y1) et (X2,Y2) symbolisant ses extrémités.
- Rectangle : Un rectangle est caractérisé par deux points (X1,Y1) et (X2,Y2) correspondant aux angles "haut-gauche" et "bas-droite".
- Polygone convexe : Un polygone convexe est caractérisé par une série de points (X1,Y1), (X2,Y2), ... , (Xn, Yn) symbolisant l'ensemble de ses sommets. Comme l'indique son nom, les angles de ce polygones doivent tous être inférieurs à 180°.

b) Objets complexes

- Réunion d'objets : Une réunion inclut l'ensemble des objets qui la composent. Elle est constituée d'une liste d'objets, simples ou complexes.
- Intersection d'objets : Une intersection inclut uniquement les portions communes à tous les objets qui la composent. Elle est constituée d'une liste d'objets, simples ou complexes.

2 – Opérations sur les objets

Les opérations seront effectuées via une ligne de commande en respectant une syntaxe précises en entrée et en sortie. Il sera possible d'annuler une commande précédemment exécutée, et de restaurer la dernière commande annulée (uniquement si elle vient d'être annulée). Jusqu'à 20 annulations consécutives seront possibles, ce qui implique de conserver en mémoire les 20 dernières commandes exécutées. Le statut de l'opération (succès ou échec) sera retourné sous la forme suivante : OK en cas de succès, ERR en cas d'échec (suivi d'un éventuel commentaire).

a) Ajouter un segment

Un segment est ajouté en indiquant ses caractéristiques sous la forme suivante :
S <name> <X1> <Y1> <X2> <Y2>

b) Ajouter un rectangle

Un rectangle est ajouté en indiquant ses caractéristiques sous la forme suivante :
R <name> <X1> <Y1> <X2> <Y2>

c) Ajouter un polygone convexe

Un polygone convexe est ajouté en indiquant ses caractéristiques sous la forme suivante :
PC <name> <X1> <Y1> <X2> <Y2> ... <XN> <YN>
Une erreur devra être retournée si le polygone n'est pas convexe.

d) Opération de réunion

Une opération de réunion est effectuée en indiquant les noms des objets concernés sous la forme suivante :
OR <name> <name1> <name2> ... <nameN>
Cette opération crée un nouvel objet.

e) Opération d'intersection

Une opération d'intersection est effectuée en indiquant les noms des objets concernés sous la forme suivante :
OI <name> <name1> <name2> ... <nameN>
Cette opération crée un nouvel objet.

f) Opération d'appartenance

Une opération de vérification d'appartenance à une forme est effectuée en indiquant le nom de l'objet et les coordonnées du point sous la forme suivante :
HIT <name> <X> <Y>
La réponse du programme devra être sous la forme YES or NO.

g) Suppression

Une opération de suppression est effectuée en indiquant les noms des objets à supprimer sous la forme suivante :
DELETE <name1> <name2> ... <nameN>
Cette opération pouvant être annulée, il faudra être capable de restaurer les objets supprimés.

h) Déplacement

Une opération de déplacement est effectuée en indiquant le nom de l'objet et son déplacement en X et en Y sous la forme suivante :
MOVE <name> <dX> <dY>

i) Énumération

Cette opération doit permettre d'afficher dans un format à définir la liste des objets. Elle est effectuée de la manière suivante :

LIST

j) Annulation

Cette opération doit permettre d'annuler la dernière commande effectuée. Il doit être possible de réaliser jusqu'à 20 annulations consécutives, ce qui implique de retenir les 20 dernières commandes exécutées. L'opération annulation est effectuée de la manière suivante :

UNDO

k) Restauration

Cette opération doit permettre de restaurer la dernière commande annulée, ce qui implique de retenir la dernière opération annulée. Ceci n'est possible que si aucun changement n'a été apporté au modèle depuis la dernière annulation. Il n'est donc pas possible d'effectuer plusieurs restaurations successives. La restauration est effectuée de la manière suivante :

REDO

l) Chargement

Cette opération doit permettre de restaurer le modèle dans l'état où il était lors de sa sauvegarde. Il faut pour cela spécifier un nom de fichier, qui doit être sous le bon format (à définir), de la manière suivante :

LOAD <filename>

m) Sauvegarde

Cette opération doit permettre de sauvegarder l'état exact du modèle (donc tous les objets qui le composent) au moment de son exécution. Le format de sauvegarde est à définir. L'opération de sauvegarde s'effectue de la manière suivante :

SAVE <filename>

n) Vidage du modèle

Cette opération doit permettre de supprimer l'intégralité des objets du modèle. Elle devra pouvoir être annulée, il doit donc être possible de restaurer le modèle dans l'état où il était juste avant cette opération. L'opération s'effectue de la manière suivante :

CLEAR

o) Fermeture de l'application

Cette opération permettra simplement de fermer l'application depuis la ligne de commande. L'opération s'effectue de la manière suivante :

EXIT