

TP5 – Spécifications

B3125

Romain BRUNAT & Merlin NIMIER-DAVID

Description générale de l'application

`geometry` est une application de gestion d'objets géométriques en ligne de commande. L'application permet de créer, modifier et supprimer des cercles, rectangles, lignes, polygones et « objets agrégés » (décrits plus bas). À tout moment, il est possible d'annuler les dernières opérations grâce à une gestion complète de l'historique. Enfin, l'application permet de sauvegarder le document en cours puis de le charger lors d'une session ultérieure.

Commandes

Pour toutes les commandes du logiciel, décrites plus en détails ci-dessous, on a les contraintes et comportements suivants :

Contraintes :

- Les coordonnées sont des entiers.
- Les noms sont constitués de lettres et / ou chiffres, sans séparateur.
- Les noms sont uniques. Si l'utilisateur essaie de créer un objet avec un nom déjà existant, une erreur est retournée.
- La description d'un objet est la ligne de commande qui permet sa création.

Réponse :

- Si la syntaxe est exacte et que l'opération s'est bien déroulée, la réponse est "OK".
- Sinon, la réponse est "ERR".

Objets géométriques

Cercle

Commande : `C Name X1 Y1 R`

Création d'un cercle de nom `Name` , de rayon `R` et ayant pour centre le point de coordonnées (`X1`, `Y1`).

Remarque : Le cercle peut être réduit à un point (cas où $R=0$).

Contraintes :

- Le rayon `R` est un entier positif.

Rectangle

Commande : `R Name X1 Y1 X2 Y2`

Création d'un rectangle de nom `Name`, défini par les deux points $(X1, Y1)$ et $(X2, Y2)$. Le premier point définit le coin supérieur gauche, le second le coin inférieur droit.

Remarques :

- On accepte des coordonnées « invalides », par exemple un coin inférieur-droit situé au-dessus du coin supérieur-gauche. Aucune vérification n'est effectuée par l'application.
- Le rectangle peut être réduit à un point (cas où $X1=X2$ et $Y1=Y2$) ou une ligne (cas où $X1=X2$ ou $Y1=Y2$).

Polyligne

Commande : `PL Name X1 Y1 X2 Y2 ... Xn Yn`

Une polyligne est une suite de segments consécutifs. Le second point d'un segment coïncide avec le premier point du segment suivant.

Création d'une polyligne de nom `Name`, défini par les n points $(X1, Y1), (X2, Y2) \dots (Xn, Yn)$. L'ordre des points est celui indiqué dans la commande.

Contraintes :

- Au minimum deux points doivent être donnés.

Remarques :

- Si trois points alignés sont donnés, on garde ces trois points tels quels : l'application n'effectue pas de simplification automatique de la polyligne.
- On autorise qu'un même point soit utilisé plusieurs fois dans la définition d'une polyligne.

Ligne

Commande : `L Name X1 Y1 X2 Y2`

Création d'une ligne de nom `Name`, défini par les deux points $(X1, Y1)$ et $(X2, Y2)$. Une ligne est un segment.

Remarque : Exactement deux points doivent être donnés. Cependant, aucune vérification n'est effectuée, ces deux points peuvent donc être identiques.

Objets agrégés

Commande : `OA Name Name1 Name2 ... NameN`

Création d'un objet agrégé formé des N différents objets nommés `Name1 ... NameN`.

Remarques :

- Un objet peut appartenir à plusieurs objets agrégés.
- Un objet agrégé peut contenir d'autres objets agrégés.
- Un objet donné plusieurs fois ne sera inclus qu'une seule fois dans l'objet agrégé (on ignore les doublons).
- On accepte la construction d'agrégats vides.

Contraintes :

- Les objets donnés doivent impérativement exister préalablement.

- Un objet agrégé ne peut pas appartenir à un objet agrégé qu'il contient.

Modification d'objets existants

Toutes les commandes ci-dessous s'appliquent aussi bien à des objets géométriques qu'à des objets agrégés. En général, lorsqu'appliquée à un objet agrégé, la commande s'applique récursivement à tous les objets contenus dans l'agrégat.

Déplacement

Commande : MOVE Name X Y

Translate l'objet Name d'un vecteur $v(X, Y)$. S'il s'agit d'un objet agrégé, l'ensemble des objets contenus dans celui-ci est déplacé de ce même vecteur v .

Contraintes :

- Si un même objet apparaît plusieurs fois dans la composition d'un agrégat il n'est déplacé qu'une seule fois.
Exemple : Si agregat3 contient agregat2 et objet1 ; agregat2 contient objet1. Le déplacement de agregat3 d'un vecteur $v(10, 10)$ ne déplacera objet1 qu'une seule fois de ce vecteur v .
- Il n'est pas interdit de déplacer un objet (ou agrégat) d'un vecteur nul (cas où $X=0$ et $Y=0$).

Suppression

Commande : DELETE Name1 Name2 ... NameN

Supprime les N objets Name1 ... NameN.

Contraintes :

- Si un nom d'objet donné est invalide, aucun élément n'est supprimé et une erreur est retournée.
- On tolère que le nom d'un même objet soit cité deux fois dans la même commande.
- La suppression d'un objet agrégé n'entraîne pas la suppression des objets le composant.
- En revanche, la suppression d'un objet faisant partie d'un agrégat entraîne la mise à jour de l'agrégat. On peut donc, suite à une suppression, obtenir un ou plusieurs agrégats vides.

Gestion de l'historique

Les commandes faisant partie de l'historique sont :

- C
- R
- L
- PL
- OA
- DELETE
- MOVE

- `CLEAR`
- `LOAD`

Annuler la dernière commande de l'historique

Commande : `UNDO`

Annule la dernière opération (incluse dans la liste ci-dessus) ayant été réalisée. S'il n'y a rien à annuler, l'application ne fait rien et renvoie tout de même "OK".

Contraintes :

- On doit être capable d'assurer au minimum vingt opérations `UNDO` consécutives.

Refaire la dernière commande annulée

Commande : `REDO`

Refait la dernière opération (incluse dans la liste ci-dessus) ayant été annulée. Un `REDO` s'effectue obligatoirement à la suite d'un `UNDO` ou d'une opération ne modifiant pas l'historique. S'il n'y a rien à refaire, l'application ne fait rien et renvoie tout de même "OK".

Contraintes :

- On doit être capable d'assurer autant d'opérations `REDO` qu'il y a eu d'opérations `UNDO`.
- Il est impossible d'effectuer un `REDO` si une nouvelle opération modifiant l'historique a été effectuée depuis le dernier `UNDO`.

Sauvegarde et chargement

Sauvegarde

Commande : `SAVE filename`

Sauvegarde l'ensemble des objets dans un fichier `filename` sous le format suivant :

- Chaque ligne décrit un objet ou un agrégat ou est un commentaire.
- Une ligne de commentaire commence par un symbole dièse (#).
- La description d'un objet est la ligne de commande qui permet sa création.

Contraintes :

- Si le fichier `filename` existe déjà, son contenu est écrasé.
- Si le fichier `filename` n'existe pas, il est automatiquement créé.
- Si le fichier n'est pas accessible en écriture, la sauvegarde n'est pas effectuée et une erreur est retournée.

Chargement

Commande : `LOAD filename`

Charge l'ensemble des objets décrit dans le fichier `filename`.

Contraintes :

- Le fichier `filename` doit suivre le format décrit à la section précédente. L'utilisateur est

responsable de la bonne syntaxe et sémantique du fichier donné.

- Si n'importe quelle commande contenue dans le fichier chargé génère une erreur, l'ensemble du chargement est annulé.
- Les lignes de commentaires sont ignorées.
- Si le fichier `filename` n'existe pas ou n'est pas accessible en lecture, une erreur est retournée.

Autres commandes

Afficher le contenu du document

Commande : `LIST`

Affiche les descripteurs d'objets existants, y compris les agrégats. Un descripteur est affiché par ligne. Cet affichage se fait par ordre alphabétique sur les noms d'objets. Un descripteur d'objet est la commande nécessaire à sa construction. Aucun message « OK » n'est affiché.

Si le document est vide, rien n'est affiché.

Vider le document

Commande : `CLEAR`

Supprime tous les objets composant le modèle actuel, y compris les agrégats. Cela correspond à un retour à l'état initial.

Remarque : Même si aucune sauvegarde n'a été effectuée, le contenu n'est pas perdu grâce à la gestion de l'historique.

Quitter

Commande : `EXIT`

Ferme l'application. Si aucune sauvegarde n'a été effectuée, le contenu est définitivement perdu car l'historique est détruit.

Réponse : aucune.