Programmation Orientée Objets L3 MIAGE – Info M1 IMDS

TP 3 et 4 : dessins avec une tortue

On veut faire un logiciel de dessin qui utilise la tortue. Les classes concernant la tortue et son utilisation sont et devront être dans l'espace de nom `logo`.

Programme d'application

L'application de dessin est articulé autour de la classe `turtleDrawApplication` : (fichiers `turtledrawapplication.h` et `turtledrawapplication.cpp`). La boucle d'exécution lance un menu et appelle la méthode correspondant au choix.

Compilez et **testez** (il faut cliquer sur la fenêtre graphique après chaque action).

Manipulation d'une tortue

On veut pouvoir garder en mémoire les différentes manipulations effectuées sur la tortue pour ensuite pouvoir annuler la dernière manipulation.

Pour cela il faut représenter les manipulations par des objets. On crée alors une hiérarchie basée sur la classe `turtleManipulation` : les objets de cette hiérarchie manipulent la tortue `t` passée en paramètre grâce à leur méthode `manipulate(t)`. **Écrivez** la classe `turtleManipulation` (fichier `turtlemanipulation.h`).

Déplacement d'une tortue

On veut considérer les déplacements d'une tortue (déplacement, rotations) comme des manipulations de tortue. **Écrivez** la classe de base `turtleMover` de ces manipulations de déplacements, ainsi que les classes `turtleForward`, `turtleLeft` et `turtleRight` (dans les mêmes fichiers `turtlemover.h` et `turtlemover.cpp`) appliquant le déplacement correspondant à la tortue (`turtleForward` contient la distance de déplacement, qui est donnée dans le constructeur et modifiable par une méthode, même principe pour `turtleLeft` et `turtleRight`).

Pour tester, dans les méthodes de `turtleDrawApplication` qui déplacent la tortue, **remplacez** l'appel de méthodes de la tortue par sa manipulation par l'intermédiaire de ces déplacements de tortue. **Testez**.

Pinceau

On veut aussi considérer que lever ou baisser le pinceau est une manipulation de tortue. **Écrivez** dans les fichiers `turtlepen.h` et `turtlepen.cpp` les classes `turtleRaisePen` et `turtleLowerPen` qui représentent ces manipulations.

Pour tester, dans les méthodes de `turtledrawapplication` qui manipulent le pinceau de la tortue, **remplacez** l'appel de méthodes de la tortue par sa manipulation par l'intermédiaire de ces manipulations de tortue. **Testez**.

Appliquer une manipulation

On veut garder en mémoire dans un tableau les manipulations effectuées sur la tortue

Ajoutez le tableau de manipulations de tortue `d_manipulations` dans la classe `turtlePaint`, tableau qui gère lui-même le gestion mémoire de ses objets. **Ajoutez** la méthode `applyManipulation(manip)` qui applique la manipulation `manip` passée en paramètre à la tortue, l'ajoute au tableau et attend de cliquer sur la fenêtre.

Modifiez dans `turtleDrawApplication` les méthodes qui modifient la tortue (pinceau et déplacements) pour qu'elles créent en mémoire dynamique la manipulation correspondante et l'appliquent avec `applyManipulation`. **Testez** : vérifiez que tout continue de fonctionner.

Annuler la dernière manipulation

Annuler la dernière manipulation revient à effacer la fenêtre (fonction `cleardevice()`), réinitialiser la tortue et lui appliquer toutes les manipulations sauf la dernière qui est préalablement enlevée du tableau.

Ajoutez dans `turtleDrawApplication` la méthode `reinitializeTurtle` qui réinitialise la tortue (position (0,0) et direction 0) et la méthode `undoLastManipulation` qui annule la dernière manipulation (si cela est possible) et attend de cliquer sur la fenêtre. **Ajoutez** dans le menu le choix `(6) annuler la derniere manipulation` et modifiez la méthode `run` en conséquence. **Testez**.

On veut en profiter pour ré-initialiser le dessin. **Ajoutez** dans `turtleDrawApplication` la méthode `reset` qui enlève toutes les manipulations, réinitialise la tortue et efface la fenêtre. **Ajoutez** alors dans le menu le choix `(7) reinitialiser le dessin` et modifiez la méthode `run` en conséquence. **Testez**.

Refaire la dernière manipulation

On veut refaire la dernière manipulation. Pour cela il faut faire une copie de la dernière manipulation du tableau et l'appliquer. Il faut donc qu'on puisse demander à une manipulation de tortue de se cloner. Modifiez toutes les classes manipulation de tortue pour qu'une manipulation de tortue renvoie une copie d'elle-même (allouée dynamiquement) grâce à sa méthode `clone()`.

Ajoutez dans `turtleDrawApplication` la méthode `redoLastManipulation` qui refait la dernière manipulation (si cela est possible). **Ajoutez** dans le menu le choix `(8) refaire la derniere manipulation` et modifiez la méthode `run` en conséquence. **Testez**.

Tracé de figures (si vous avez le temps)

On veut pouvoir tracer des figures géométriques avec la tortue : polygones, cercles, …

On considère d'abord le tracé de polygones réguliers. Pour tracer un polygone régulier à n côtés inscrit dans un cercle de rayon r, il faut tracer n côtés de longueur $2*r*\sin(\pi/n)$ en tournant à gauche d'un angle de $360/n$ degrés à chaque fois, et finalement à la fin tourner à gauche d'un angle de $360\%n$.

Écrivez la classe `turtlePolygon` , manipulation de tortue qui lui fait tracer un polygone régulier.

Ajoutez dans `turtleDrawApplication` la méthode `drawPolygon` qui demande le nombre de côtés et la taille d'un côté et fait tracer le polygone correspondant. **Ajoutez** dans le menu le choix `(9) tracer un polygone` et modifiez la méthode `run` en conséquence. **Testez**.