

M2 Informatique : Image, Logiciels

Projet C++11 minimum

Dessiner, déplacer et questionner une image

Cet exercice manipule des images constituées de figures géométriques, liées par des relations de composition et d'héritage public.

- On souhaite modéliser un ensemble de figures géométriques.
- Parmi les figures géométriques, on s'intéresse plus particulièrement aux lignes droites (définies par deux points), aux cercles (définis par un centre et un rayon), aux triangles (définis par trois points), aux rectangles (définis par deux points).
- On souhaite pouvoir placer les figures géométriques sur une image (*les doublons ne sont pas autorisés*).
- Une image peut également contenir des images de plus petite taille (aire).
- Les calculs du périmètre et de l'aire sont requis. Pour une ligne, considérer qu'elle a une épaisseur de 1.
- Les formes géométriques peuvent se transformer par homothétie, se déplacer par translation, par rotation, par symétries centrale et axiale. Une classe matrice2D est à développer.
- On souhaite pouvoir trier les formes selon plusieurs critères: leur périmètre, leur aire et leur distance à l'origine.
- On souhaite pouvoir rechercher, compter, supprimer des figures selon certains critères :
 - leur forme (ligne, cercle, triangle, rectangle)
 - leur périmètre inférieur à un seuil donné.
- Dans un premier temps, un ensemble représentatif de figures peut être entré en dur dans le programme.
- Dans un deuxième temps, les données sont à lire dans un fichier dont le format sera expliciter dans le rapport.
- BONUS : Afin de visualiser graphiquement les images et leurs déplacements, l'utilisation de la MLV pourra être envisagée.

La classe Image est en relation d'agrégation (composition forte) avec un container STL de figures (pointeurs intelligents sur des figures).

Une gestion d'exception est parfois nécessaire.

Un respect des principes S.O.L.I.D. est demandé.

Un maximum de fonctionnalités C++11 sont à mettre en oeuvre, par exemple :

- espaces de noms
- délégation de constructeurs
- move semantics (constructeur et assignation par déplacement)
- algorithmes de la STL
- prédicats "if"
- lambda-fonctions
- mots-clés : override, auto, decltype
- ...

CONSIGNES DE RENDU PROJET

Le projet est à réaliser par **binôme ou trinôme pour les filières "Logiciels" et "Image"**.

Le rendu est sous la forme **NOM1_NOM2[_NOM3].zip**, archive contenant le makefile, les .h et les .cpp et autres fichiers nécessaires, sans oublier le rapport au format pdf.

Les projets sont à déposer sur l'espace de rendu de l'ENT, ceux rendus uniquement par mail seront pénalisés.

Merci de votre sens des responsabilités.

Afin de bien mettre en évidence les relations entre les classes de votre projet, vous pouvez utiliser un logiciel qui vous générera un diagramme de classes à partir de vos sources.

L'idéal est d'utiliser un logiciel libre permettant la rétro-conception.

StarUML : <https://www.projet-plume.org/mots-cles-proposes-par-lauteur/retro-ingenierie>

BOUML : <https://www.projet-plume.org/fiche/bouml>

Une documentation des sources doit être générée automatiquement (utiliser doxygen pour documenter les sources C++).

Votre rapport pourra contenir des graphes d'appels.

Utiliser des outils pour générer des graphes d'appels :

<http://odellconnie.blogspot.fr/2012/07/free-c-software-call-graph-generators.html>

Un rapport doit fournir votre analyse et votre conception.

Les éléments suivants doivent notamment y figurer :

- une introduction exposant clairement les objectifs, limites, choix du projet
- un mode d'emploi de l'application
- des schémas décrivant l'architecture fonctionnelle
- des schémas décrivant les structures de données utilisées
- une explication en français et/ou pseudo-code et/ou langage d'implémentation choisi des principales fonctions (code à fournir et à commenter)
- une conclusion résumant le travail effectué et ouvrant des perspectives
- une bibliographie utilisée
- une table des matières

Les classes conçues doivent être détaillées aussi bien leur interface que leur implémentation.

Une bonne idée est de réaliser le rapport en même temps que le développement.

Exemple de répartition des tâches :

- fonctionnalités demandées sur les figures + Rapport : NOM1 (beaucoup), NOM2 et NOM3
- classe matrice pour les déplacements + Rapport : NOM2
- lecture des figures dans un fichier + Rapport : NOM3