

Mini Projet #1

Noms:

Cours: C++ et programmation orientée objet – Professeur: Thibaut Tortosa
Date de remise: 08/11/2019

Introduction

Ce mini projet vise à développer un logiciel de dessin simple.

Le dessin permettra de représenter plusieurs formes dans une image. Le choix et les coordonnées des formes pourront être choisis à l'aide d'une interface en ligne de commande, ou encore à l'aide d'un fichier de configuration.

L'enregistrement des images se fera à l'aide de la librairie stb image.

Critères de notation

Le projet sera dans un premier temps guidé par un ensemble de questions. Ceci permettra de mettre en place une base minimale de développement. Une fois cette base mise en place, il sera demandé à chaque étudiant de choisir les fonctionnalités à ajouter à son projet. Par ailleurs, il est demandé à chaque étudiant de commenter de manière détaillée l'ensemble du code, en mettant en correspondance le code développé avec les éléments théoriques du cours.

Le projet sera évalué selon les critères suivants :

- Présence des fonctionnalités de base de l'application
- L'implémentation des fonctionnalités de base de l'application : On cherchera à utiliser de manière pertinente les notions du cours
- Présence de commentaires permettant de justifier les choix d'implémentation
- Choix et implémentation des fonctionnalités supplémentaires permettant de mettre en avant les notions du cours

Mise en place

Prise en main du code source

Procédure. Dans un premier temps, vérifier l'installation du paquet build-essential

```
sudo apt-get install build-essential
```

Celui-ci installe plusieurs compilateurs de C et C++, ainsi que la commande make
Installer ensuite le paquet zip

```
sudo apt-get install zip
```

Une fois les paquets installé, choisir un répertoire de travail et y copier le projet miniproject.zip Déziper ensuite le projet puis entrer dans le répertoire créé

```
unzip miniproject.zip  
cd miniproject-isen
```

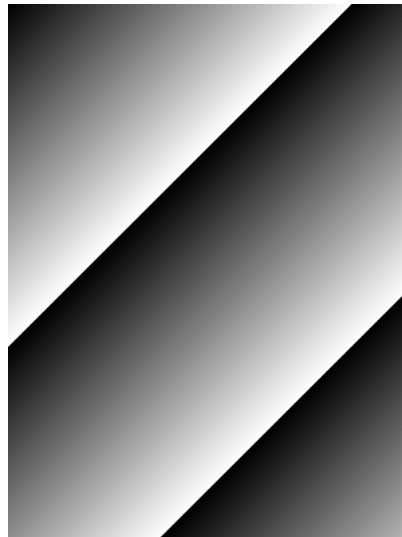
Vérifier la compilation du programme :

```
make
```

Vérifier la compilation du projet puis exécuter le programme résultant :

```
./FigureCreator
```

Le programme a alors produit une image nommée **test_image.bmp**. L'ouvrir afin de vérifier son contenu. On doit obtenir l'image suivante :



Votre environnement de travail est maintenant prêt.

Présentation du projet

Arborescence du projet. Dans le répertoire de travail, on trouve les fichiers et répertoires suivants :

assignment : Contient ce document,

extern : Contient les bibliothèques externes,

src : Contient le code source des classes du projet,

makefile : Fichier permettant de configurer la compilation,

miniproj.cpp : Fichier contenant le **main**.

Lors du développement du projet, les classes ajoutées au projet seront ajoutées dans le répertoire src. Il s'agira alors de modifier le fichier makefile pour ajouter la nouvelle classe au projet.

Architecture du projet. Actuellement, le projet est constitué des éléments suivants :

1. Une classe **Drawing** permettant de représenter le dessin que nous allons réaliser, contenus dans les fichiers *Drawing.h* et *Drawing.cpp*
2. Un **main** contenu dans le fichier *miniproj.cpp*

La classe **Drawing** a les attributs suivants :

width : largeur de l'image. Cet attribut est constant et défini à la construction de l'objet

height : hauteur de l'image. Cet attribut est constant et défini à la construction de l'objet

image : contenu de l'image, stocké dans un vecteur de *char*

Elle a les méthodes suivantes :

Drawing : Constructeur de l'objet. Permet d'initialiser **width** et **height**, puis d'allouer la mémoire nécessaire au stockage de l'image.

Drawing : Destructeur de l'objet

clearImage : Remplit l'image de pixels à 0

createTestImage : Remplit l'image avec une image de test. Ceci permet de vérifier le fonctionnement de la classe

save : Permet d'enregistrer l'image créée. Pour le moment, cette méthode crée une image de test avant de l'enregistrer.

Le **main** crée une instance de **Drawing** puis enregistre une image de test dans un fichier.

Question 1

Créer une classe **Figure** permettant de représenter une Figure quelconque dans l'espace de la classe **Drawing**. On pourra s'inspirer d'un exercice que nous avons fait précédemment.

Précisions.

1. La classe **Figure** devra être écrite dans deux fichiers séparés, *Figure.h* et *Figure.cpp*
 2. Le Makefile devra alors être modifié pour inclure ces fichiers
 3. La classe **Figure** aura un attribut permettant de représenter son image au même format que l'image contenue dans **Drawing**
 4. La classe **Figure** aura un attribut de largeur et de hauteur permettant de représenter sa taille
-

5. La classe **Figure** aura une méthode virtuelle pure visant à dessiner la forme dans un tableau passé en paramètre
6. La méthode *setPoint* mettra le pixel visé à 255 et non à 1
7. Une méthode devra être implémentée pour retourner l'image de la figure, sans que celle-ci ne puisse être modifiée.

Question 2

Créer des classes dérivées de la classe **Figure**.

1. Créer la classe **FigCross**
2. Créer la classe **FigRectangle**
3. Créer la classe **FigSquare**
4. Créer la classe **FigSegment**

Précisions.

- Chaque classe devra être écrite dans un nouveau fichier
- Il pourra également être utile d'ajouter la classe **Point** et la classe **Segment** au projet

Question 3

Réaliser les modifications suivantes dans la classe **Drawing** :

1. Ajouter la possibilité de stocker un ensemble d'objets de type **Figure**
2. Ajouter la possibilité de stocker les coordonnées correspondant ces objets
3. Ajouter une méthode *draw* permettant de dessiner l'ensemble de ces figures dans l'attribut image
4. Modifier la méthode *save* pour qu'elle remplace son appel à la méthode *createTestImage* par un appel à la méthode *draw*

Question 4

Réaliser une classe **Menu** implémentant un menu interactif en ligne de commande permettant d'interagir avec un objet de type **Drawing**. Pour cela, il implémentera une fonction **run** sous forme d'une boucle, qui interragira avec (cin, cout) l'utilisateur et lui permettra de réaliser les opérations suivantes :

1. Ajouter une figure et choisir son type, ses caractéristiques et ses coordonnées
2. Sauvegarder l'image dans un fichier au format BMP
3. Quitter le programme

Le *main* pourra créer une instance de cet objet et lancer la méthode *run* afin de configurer et de sauvegarder des dessins.

Précisions. Cette question pourra éventuellement impliquer la création de plus d'une classe

Question 5

Ajouter de nouvelles fonctionnalités au logiciels. Plusieurs axes d'enrichissement du logiciel peuvent être envisagés :

1. Ajout de figures
2. Création de figures à partir d'autres figures
3. Ajout de moyen de transformation de figure ou du dessin (rotation, translation)
4. Ajout de caractéristiques pour les figures (épaisseur de trait, intensité de pixel)
5. Utilisation d'un fichier de configuration pour la création des figures

Le *main* pourra créer une instance de cet objet et lancer la méthode *run* afin de configurer et de sauvegarder des dessins