azPNG

C++ PNG Creator

Créer par
AZAR Majed Saliou
majed.azar7@gmail.com
majedazar.com

Lomé, Togo
17 December 2020

Table des matières

Aperçu

Langage : C++

Plateformes : Windows, Linux

Résumé : Une bibliothèque permettant de créer des images .png à l'aide de formes (pixels, lignes, rectangles, etc ...)

Description :

La bibliothèque azPNG a été principalement écrite pour créer des images au format .png. Les images peuvent être dessinées par le programmeur en utilisant des pixels, des lignes, des rectangles ou d'autres formes géométriques. Les fonctions utilisées pour dessiner sont assez élémentaires mais simples à saisir et à comprendre.

Étant donné que cette bibliothèque se concentre uniquement sur le dessin de l'image, elle ne dispose d'aucune fonctionnalité pour afficher la sortie (si vous souhaitez voir les résultats, ouvrez l'image .png créée à l'aide d'un autre logiciel). Ici, le format .ppm est utilisé pour créer l'image, puis la sortie est convertie en .png. De ce fait, vous pouvez utiliser des couleurs 24 bits (rouge, vert et bleu de 0 à 255), mais la transparence n’est pas prise en charge.

Mise en place

Pour utiliser la bibliothèque azPNG dans un projet C ++, vous devez d'abord inclure le fichier d'en-tête qui lui est associé :

```
#include "azPNG.h"
```

Ensuite, l'étape suivante dépend de votre système.

Sous Linux : Vous devez installer **netpbm** pour pouvoir convertir vos images .ppm en .png :

```
sudo apt-get install netpbm
```

Sous Windows : Vous devez copier le fichier **convert.exe** à côté de l'exécutable de chaque programme qui utilise la bibliothèque azPNG. De cette façon, il l'utilisera pour convertir vos images .ppm en .png.

Lorsque vous avez terminé la configuration, vous pouvez enfin utiliser les objets à l'intérieur de l'espace de noms azPNG :

```
azPNG::Image newImage(640,480,azPNG::Color::White());
```

Objet Color

L'objet Color est l'objet utilisé pour représenter les pixels. Il utilise 3 variables pour représenter l'intensité de la lumière rouge, verte et bleue pour former une couleur. 0 est le niveau minimum d'intensité (noir) et 255 est le maximum (blanc).

Constructeurs

L'objet Color est livré avec 3 constructeurs différents. Le premier crée la couleur noire, le deuxième crée une variante grise et le troisième peut créer n'importe quelle couleur.

```
Color()
```
Description : Crée un pixel noir

Paramètres : aucun

```
Color(short const& I_GREY)
```
Description : Crée un pixel gris en utilisant des niveaux de gris

Paramètres :

- I_GREY : entier représentant l'intensité du gris (de 0 à 255)

```
Color(short const& I_RED, short const& I_GREEN, short const& I_BLUE)
```
Description : Crée un pixel coloré en utilisant les normes RVB

Paramètres :

- I_RED : entier représentant l'intensité du rouge (de 0 à 255)
- I_GREEN: entier représentant l'intensité du vert (de 0 à 255)
- I_BLUE: entier représentant l'intensité du bleu (de 0 à 255)

Getters

Les getters liés à l'objet Color peuvent fournir un moyen d'obtenir l'intensité du rouge, du vert et du bleu utilisé pour l'image.

```
unsigned short GetRed() const
```
Description : Renvoie l'intensité du rouge utilisé à l'intérieur du pixel

Paramètres : aucun

Retour : un entier représentant l'intensité du rouge

```
unsigned short GetGreen() const
```
Description : Renvoie l'intensité du vert utilisé à l'intérieur du pixel

Paramètres : aucun

Retour : un entier représentant l'intensité du vert

```
unsigned short GetBlue() const
```
Description : Renvoie l'intensité du bleu utilisé à l'intérieur du pixel

Paramètres : aucun

Retour : un entier représentant l'intensité du bleu

```
std::string GetColor() const
```
Description : Renvoie toute l'intensité de toutes les couleurs utilisées à l'intérieur du pixel

Paramètres : aucun

Retour : une chaîne représentant l'intensité de chaque couleur séparée par un espace (ex: "0 127 255")

Setters

Les setters liés à l'objet Color peuvent fournir un moyen de définir l'intensité du rouge, du vert et du bleu utilisé pour l'image.

```
bool SetRed(short const& I_INTENSITY)
```
Description : Définit l'intensité du rouge utilisé à l'intérieur du pixel à une valeur donnée

Paramètres :

- I_INTENSITY : entier représentant la valeur de l'intensité (de 0 à 255)

Retour : true si l'intensité a été modifiée ou **false** si la valeur donnée est incorrecte

```
bool SetGreen(short const& I_INTENSITY)
```
Description : Définit l'intensité du vert utilisé à l'intérieur du pixel à une valeur donnée

Paramètres :

- I_INTENSITY : entier représentant la valeur de l'intensité (de 0 à 255)

Retour : true si l'intensité a été modifiée ou **false** si la valeur donnée est incorrecte

```
bool SetBlue(short const& I_INTENSITY)
```
Description : Définit l'intensité du bleu utilisé à l'intérieur du pixel à une valeur donnée

Paramètres :

- I_INTENSITY : entier représentant la valeur de l'intensité (de 0 à 255)

Retour : true si l'intensité a été modifiée ou **false** si la valeur donnée est incorrecte

```
bool SetColor(short const& I_RED, short const& I_GREEN, short const& I_BLUE)
```
Description : Définit la couleur à l'intérieur du pixel sur une valeur donnée

Paramètres :

- I_RED : entier représentant la valeur de l'intensité du rouge (de 0 à 255)
- I_GREEN : entier représentant la valeur de l'intensité du vert (de 0 à 255)
- I_BLUE: entier représentant la valeur de l'intensité du bleu (de 0 à 255)

Retour : true si la couleur a été modifiée ou **false** si la couleur donnée était incorrecte

```
bool SetColor(Color const& CL_COLOR)
```

Description : Définit la couleur à l'intérieur du pixel sur une valeur donnée

Paramètres :

- CL_COLOR : Objet couleur représentant la couleur à appliquer

Retour : true si la couleur a été modifiée ou **false** si la couleur donnée était incorrecte

Couleurs prédéfinies

L'objet Color propose également des couleurs prédéfinies que vous pouvez utiliser sans avoir à donner manuellement l'intensité des composants rgb (ex: azPNG::Color::Black()).

```
Color static RandColor()
```

Description : Renvoie une couleur aléatoire

Paramètres : aucun

Retour : un objet couleur représentant une couleur aléatoire

```
Color static RandRed()
```

Description : Renvoie une variante aléatoire du rouge

Paramètres : aucun

Retour : un objet couleur représentant une variante aléatoire du rouge

```
Color static RandGreen()
```

Description : Renvoie une variante aléatoire du vert

Paramètres : aucun

Retour : un objet couleur représentant une variante aléatoire du vert

```
Color static RandBlue()
```

Description : Renvoie une variante aléatoire de bleu

Paramètres : aucun

Retour : un objet couleur représentant une variante aléatoire du bleu

```
Color static RandYellow()
```

Description : Renvoie une variante aléatoire du jaune

Paramètres : aucun

Retour : un objet couleur représentant une variante aléatoire du jaune

```
Color static RandCyan()
```
Description : Renvoie une variante aléatoire de cyan

Paramètres : aucun

Retour : un objet couleur représentant une variante aléatoire du cyan

```
Color static RandMagenta()
```
Description : Renvoie une variante aléatoire de magenta

Paramètres : aucun

Retour : un objet couleur représentant une variante aléatoire du magenta

```
Color static RandGrey()
```
Description : Renvoie une variante aléatoire du gris

Paramètres : aucun

Retour : un objet couleur représentant une variante aléatoire du gris

```
Color static Black()
```
Description : Renvoie une couleur noire

Paramètres : aucun

Retour : un objet couleur représentant la couleur noire

```
Color static Red()
```
Description : Renvoie une couleur rouge

Paramètres : aucun

Retour : un objet couleur représentant la couleur rouge

```
Color static Green()
```
Description : Renvoie une couleur verte

Paramètres : aucun

Retour : un objet couleur représentant la couleur verte

```
Color static Blue()
```
Description : Renvoie une couleur bleue

Paramètres : aucun

Retour : un objet couleur représentant la couleur bleue

```
Color static Yellow()
```

Description : Renvoie une couleur jaune

Paramètres : aucun

Retour : un objet couleur représentant la couleur jaune

```
Color static Cyan()
```
Description : Renvoie une couleur cyan

Paramètres : aucun

Retour : un objet couleur représentant la couleur cyan

```
Color static Magenta()
```
Description : Renvoie une couleur magenta

Paramètres : aucun

Retour : un objet couleur représentant la couleur magenta

```
Color static White()
```
Description : Renvoie une couleur blanche

Paramètres : aucun

Retour : un objet couleur représentant la couleur blanche

```
Color static DarkRed()
```
Description : Renvoie une couleur rouge foncé

Paramètres : aucun

Retour : un objet couleur représentant la couleur rouge foncé

```
Color static DarkGreen()
```
Description : Renvoie une couleur vert foncé

Paramètres : aucun

Retour : un objet couleur représentant la couleur vert foncé

```
Color static DarkBlue()
```
Description : Renvoie une couleur bleu foncé

Paramètres : aucun

Retour : un objet couleur représentant la couleur bleu foncé

```
Color static DarkYellow()
```
Description : Renvoie une couleur jaune foncé

Paramètres : aucun

Retour : un objet couleur représentant la couleur jaune foncé

```
Color static DarkCyan()
```
Description : Renvoie une couleur cyan foncé

Paramètres : aucun

Retour : un objet couleur représentant la couleur cyan foncé

```
Color static DarkMagenta()
```
Description : Renvoie une couleur magenta foncé

Paramètres : aucun

Retour : un objet couleur représentant la couleur magenta foncé

```
Color static Grey()
```
Description : Renvoie une couleur grise

Paramètres : aucun

Retour : un objet couleur représentant la couleur grise

```
Color static VeryDarkRed()
```
Description : Renvoie une couleur rouge très foncé

Paramètres : aucun

Retour : un objet couleur représentant la couleur rouge très foncé

```
Color static VeryDarkGreen()
```
Description : Renvoie une couleur vert très foncé

Paramètres : aucun

Retour : un objet couleur représentant la couleur vert très foncé

```
Color static VeryDarkBlue()
```
Description : Renvoie une couleur bleu très foncé

Paramètres : aucun

Retour : un objet couleur représentant la couleur bleu très foncé

```
Color static VeryDarkYellow()
```
Description : Renvoie une couleur jaune très foncé

Paramètres : aucun

Retour : un objet couleur représentant la couleur jaune très foncé

```
Color static VeryDarkCyan()
```
Description : Renvoie une couleur cyan très foncé

Paramètres : aucun

Retour : un objet couleur représentant la couleur cyan très foncé

```
Color static VeryDarkMagenta()
```
Description : Renvoie une couleur magenta très foncé

Paramètres : aucun

Retour : un objet couleur représentant la couleur magenta très foncé

```
Color static DarkGrey()
```
Description : Renvoie une couleur gris foncé

Paramètres : aucun

Retour : un objet couleur représentant la couleur gris foncé

Objet Image

L'objet image est utilisé pour créer notre image. Il s'agit d'une grille de I_WIDTH x I_HEIGHT pixels (l'objet Color). Cette grille peut être modifiée pixel par pixel mais elle prend également en charge les formes géométriques telles que les lignes, les rectangles, les triangles, le trapèze et le cercle. Ces formes peuvent être dessinées vides (seul le contour sortira alors) ou elles peuvent être dessinées remplies.

Constructeurs

L'objet Image est livré avec 2 constructeurs. Le premier crée une image noire et le second crée une image remplie avec la couleur que vous avez définie.

```
Image(short const& I_WIDTH, short const& I_HEIGHT)
```
Description : Construit une image remplie de la couleur noire

Paramètres :

- I_WIDTH : entier représentant la largeur de l'image
- I_HEIGHT : entier représentant la hauteur de l'image

```
Image(short const& I_WIDTH, short const& I_HEIGHT, Color const& CL_COLOR)
```
Description : Construit une image remplie de la couleur donnée en paramètre

Paramètres :

- I_WIDTH : entier représentant la largeur de l'image
- I_HEIGHT : entier représentant la hauteur de l'image
- CL_COLOR : Objet de couleur représentant la couleur à appliquer

Getters

Dans cette classe, il existe des getters classiques pour obtenir la largeur, la hauteur ou même un pixel sur l'image mais il existe également des getters pour obtenir une version recadrée ou mise à l'échelle d'une image.

```
unsigned short GetWidth() const
```
Description : Renvoie la largeur de l'image

Paramètres : aucun

Retour : un entier représentant la largeur de l'image

```
unsigned short GetHeight() const
```
Description : Renvoie la hauteur de l'image

Paramètres : aucun

Retour : un entier représentant la hauteur de l'image

```
Color GetPixel(short const& I_X, short const& I_Y) const
```
Description : Renvoie un pixel à l'intérieur de l'image

Paramètres :

- I_X : entier représentant la coordonnée x du pixel
- I_Y : entier représentant la coordonnée y du pixel

Retour : un objet couleur représentant un pixel à l'intérieur de l'image

```
Image GetCrop(short I_X1, short I_Y1, short I_X2, short I_Y2) const
```
Description : Renvoie une version recadrée de l'image

Paramètres :

- I_X1 : entier représentant la coordonnée x du 1er point
- I_Y1 : entier représentant la coordonnée y du 1er point
- I_X2 : entier représentant la coordonnée x du 2ème point
- I_Y2 : entier représentant la coordonnée y du 2ème point

Retour : un objet image représentant l'image recadrée

```
Image GetScale(short const& I_RATIO) const
```
Description : Renvoie une version mise à l'échelle de l'image

Parameters :

- I_RATIO : entier déterminant le nombre de pixels à mettre à l'échelle de l'image

Return : un objet image représentant l'image mise à l'échelle

Méthodes de dessin

L'objet Image est livré avec diverses fonctions pour dessiner des formes géométriques. Vous pouvez dessiner des pixels, des lignes, des rectangles, des cubes, des triangles, des trapèzes et des cercles remplis ou non.

```
void DrawPixel(float const& F_X, float const& F_Y, Color const& CL_COLOR = Color(255))
```

Description : Dessinez un pixel à l'intérieur de l'image

Paramètres :

- F_X : flottant représentant la coordonnée x du pixel
- F_Y : flottant représentant la coordonnée y du pixel
- CL_COLOR (defaut = Color::White()) : Objet couleur représentant la couleur du pixel

Retour : aucun

```
void DrawLine(float const& F_X1, float const& F_Y1, float const& F_X2, float const& F_Y2, Color const& CL_COLOR = Color(255))
```

Description : Tracez une ligne à l'intérieur de l'image

Paramètres :

- F_X1 : flottant représentant la coordonnée x du 1er point utilisé pour dessiner
- F_Y1 : flottant représentant la coordonnée y du 1er point utilisé pour dessiner
- F_X2 : flottant représentant la coordonnée x du 2ème point utilisé pour dessiner
- F_Y2 : flottant représentant la coordonnée y du 2ème point utilisé pour dessiner
- CL_COLOR (defaut = Color::White()) : Objet couleur représentant la couleur de la ligne

Retour : aucun

```
void DrawRectangle(float F_X1, float F_Y1, float F_X2, float F_Y2, Color const& CL_COLOR = Color(255))
```

Description : Dessinez un rectangle à l'intérieur de l'image

Paramètres :

- F_X1 : flottant représentant la coordonnée x du 1er point utilisé pour dessiner
- F_Y1 : flottant représentant la coordonnée y du 1er point utilisé pour dessiner
- F_X2 : flottant représentant la coordonnée x du 2ème point utilisé pour dessiner

- F_Y2 : flottant représentant la coordonnée y du 2ème point utilisé pour dessiner
- CL_COLOR (defaut = Color::White()) : Objet couleur représentant la couleur de la forme

Retour : aucun

```
void FillRectangle(float F_X1, float F_Y1, float F_X2, float F_Y2, Color const& CL_COLOR = Color(255))
```

Description : Dessinez un rectangle rempli à l'intérieur de l'image

Paramètres :

- F_X1 : flottant représentant la coordonnée x du 1er point utilisé pour dessiner
- F_Y1 : flottant représentant la coordonnée y du 1er point utilisé pour dessiner
- F_X2 : flottant représentant la coordonnée x du 2ème point utilisé pour dessiner
- F_Y2 : flottant représentant la coordonnée y du 2ème point utilisé pour dessiner
- CL_COLOR (defaut = Color::White()) : Objet couleur représentant la couleur de la forme

Retour : aucun

```
void DrawCube(float F_X, float F_Y, float F_C, Color const& CL_COLOR = Color(255))
```

Description : Dessinez un cube à l'intérieur de l'image

Paramètres :

- F_X : flottant représentant la coordonnée x du point utilisé pour dessiner
- F_Y : flottant représentant la coordonnée y du point utilisé pour dessiner
- F_C : flottant représentant la longueur du cube
- CL_COLOR (defaut = Color::White()) : Objet couleur représentant la couleur de la forme

Retour : aucun

```
void FillCube(float F_X, float F_Y, float F_C, Color const& CL_COLOR = Color(255))
```

Description : Dessinez un cube rempli à l'intérieur de l'image

Paramètres :

- F_X : flottant représentant la coordonnée x du point utilisé pour dessiner
- F_Y : flottant représentant la coordonnée y du point utilisé pour dessiner
- F_C : flottant représentant la longueur du cube
- CL_COLOR (defaut = Color::White()) : Objet couleur représentant la couleur de la forme

Retour : aucun

```
void DrawTriangle(float const& F_X1, float const& F_Y1, float const& F_X2, float const& F_Y2, float const& F_X3, float const& F_Y3 , Color const& CL_COLOR = Color(255))
```

Description : Dessinez un triangle à l'intérieur de l'image

Paramètres :

- F_X1 : flottant représentant la coordonnée x du 1er point utilisé pour dessiner
- F_Y1 : flottant représentant la coordonnée y du 1er point utilisé pour dessiner
- F_X2 : flottant représentant la coordonnée x du 2ème point utilisé pour dessiner
- F_Y2 : flottant représentant la coordonnée y du 2ème point utilisé pour dessiner
- F_X3 : flottant représentant la coordonnée x du 3ème point utilisé pour dessiner
- F_Y3 : flottant représentant la coordonnée y du 3ème point utilisé pour dessiner
- CL_COLOR (defaut = Color::White()) : Objet couleur représentant la couleur de la forme

Retour : aucun

```
void FillTriangle(float const& F_X1, float const& F_Y1, float const& F_X2, float const& F_Y2, float const& F_X3, float const& F_Y3 , Color const& CL_COLOR = Color(255))
```

Description : Dessinez un triangle rempli à l'intérieur de l'image

Paramètres :

- F_X1 : flottant représentant la coordonnée x du 1er point utilisé pour dessiner
- F_Y1 : flottant représentant la coordonnée y du 1er point utilisé pour dessiner
- F_X2 : flottant représentant la coordonnée x du 2ème point utilisé pour dessiner
- F_Y2 : flottant représentant la coordonnée y du 2ème point utilisé pour dessiner
- F_X3 : flottant représentant la coordonnée x du 3ème point utilisé pour dessiner
- F_Y3 : flottant représentant la coordonnée y du 3ème point utilisé pour dessiner
- CL_COLOR (defaut = Color::White()) : Objet couleur représentant la couleur de la forme

Retour : aucun

```
void DrawTrapeze(float const& F_X1, float const& F_Y1, float const& F_X2, float const& F_Y2, float const& F_X3, float const& F_Y3, float const& F_X4, float const& F_Y4 , Color const& CL_COLOR = Color(255))
```

Description : Dessinez un trapèze à l'intérieur de l'image

Paramètres :

- F_X1 : flottant représentant la coordonnée x du 1er point utilisé pour dessiner

- F_Y1 : flottant représentant la coordonnée y du 1er point utilisé pour dessiner
- F_X2 : flottant représentant la coordonnée x du 2ème point utilisé pour dessiner
- F_Y2 : flottant représentant la coordonnée y du 2ème point utilisé pour dessiner
- F_X3 : flottant représentant la coordonnée x du 3ème point utilisé pour dessiner
- F_Y3 : flottant représentant la coordonnée y du 3ème point utilisé pour dessiner
- F_X4 : flottant représentant la coordonnée x du 4ème point utilisé pour dessiner
- F_Y4 : flottant représentant la coordonnée y du 4ème point utilisé pour dessiner
- CL_COLOR (defaut = Color::White()) : Objet couleur représentant la couleur de la forme

Retour : aucun

```
void FillTrapeze(float const& F_X1, float const& F_Y1, float const& F_X2, float const& F_Y2, float const& F_X3, float const& F_Y3, float const& F_X4, float const& F_Y4 , Color const& CL_COLOR = Color(255))
```

Description : Dessinez un trapèze rempli à l'intérieur de l'image

Paramètres :

- F_X1 : flottant représentant la coordonnée x du 1er point utilisé pour dessiner
- F_Y1 : flottant représentant la coordonnée y du 1er point utilisé pour dessiner
- F_X2 : flottant représentant la coordonnée x du 2ème point utilisé pour dessiner
- F_Y2 : flottant représentant la coordonnée y du 2ème point utilisé pour dessiner
- F_X3 : flottant représentant la coordonnée x du 3ème point utilisé pour dessiner
- F_Y3 : flottant représentant la coordonnée y du 3ème point utilisé pour dessiner
- F_X4 : flottant représentant la coordonnée x du 4ème point utilisé pour dessiner
- F_Y4 : flottant représentant la coordonnée y du 4ème point utilisé pour dessiner
- CL_COLOR (defaut = Color::White()) : Objet couleur représentant la couleur de la forme

Retour : aucun

```
void DrawCircle(float const& F_X, float const& F_Y, float const& F_R, Color const& CL_COLOR = Color(255))
```

Description : Dessinez un cercle à l'intérieur de l'image

Paramètres :

- F_X : flottant représentant la coordonnée x du point utilisé pour dessiner
- F_Y : flottant représentant la coordonnée y du point utilisé pour dessiner

- F_R : flottant représentant le rayon du cercle
- CL_COLOR (defaut = Color::White()) : Objet couleur représentant la couleur de la forme

Retour : aucun

```
void FillCircle(float const& F_X, float const& F_Y, float const& F_R, Color const& CL_COLOR = Color(255))
```

Description : Dessinez un cercle plein à l'intérieur de l'image

Paramètres :

- F_X : flottant représentant la coordonnée x du point utilisé pour dessiner
- F_Y : flottant représentant la coordonnée y du point utilisé pour dessiner
- F_R : flottant représentant le rayon du cercle
- CL_COLOR (defaut = Color::White()) : Objet couleur représentant la couleur de la forme

Retour : aucun

```
void Clear(Color const& CL_COLOR = Color(255,255,255))
```

Description : Remplissez l'image avec une couleur

Paramètres :

- CL_COLOR (defaut = Color::White()) : Objet couleur représentant la couleur appliquée à l'image

Retour : aucun

Sauvegarde

Enfin, la fonction suivante peut être utilisée pour enregistrer un objet Image une fois le travail terminé.

```
bool Save(std::string const& S_PATH = "image.ppm", bool const& B_CONVERT = true)
```

Description : Enregistre l'image en .ppm et la convertit en .png

Paramètres :

- S_PATH (defaut = image.ppm) : chaîne représentant le chemin et le nom où l'image doit être enregistrée (ex: blank.ppm)
- B_CONVERT : booléen déterminant si l'image est convertie après avoir été enregistrée ou non

Retour : true si l'image peut être enregistrée sinon **false**