14/01/2022

DOCUMENTATION

Projet Anamorphoses

Table des matières

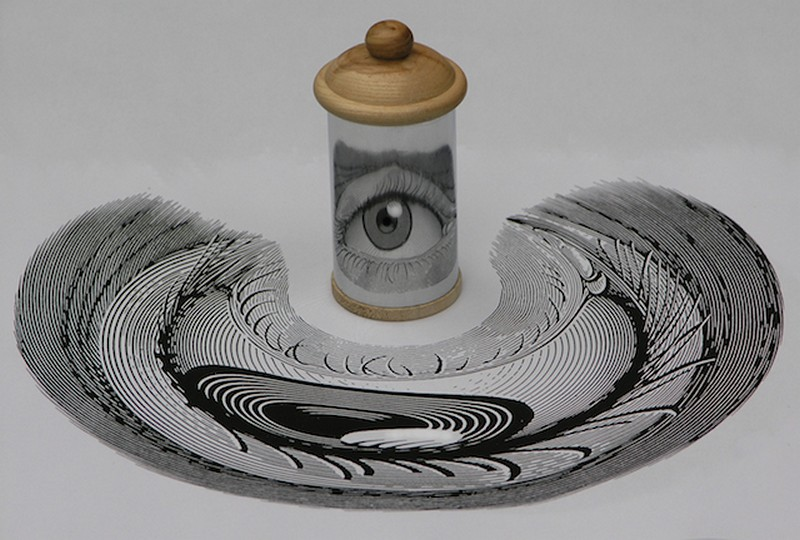
[Les calcules 2](#_Toc92711641)

[Le code 2](#_Toc92711642)

[Organisation du code 5](#_Toc92711643)

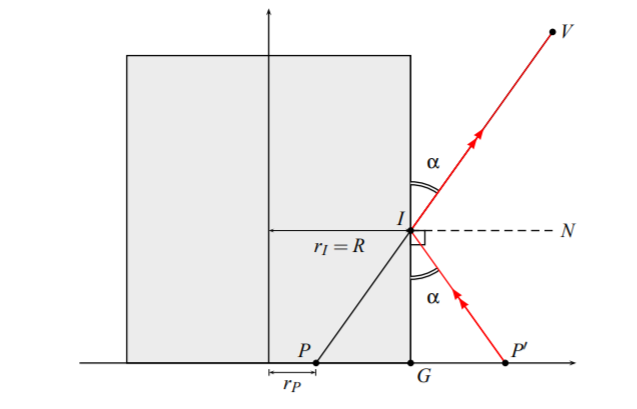
# Les calcules

Pour ce sujet nous sommes parties sur l’anamorphose cylindrique



Pour pouvoir reproduire l’anamorphoses cylindrique, nous nous sommes rendu compte qu’il était extrêmement compliqué de calculer les points de l’image non déformer devant apparaitre dans le miroir cylindrique en partant de l’image déformer sur le plan.

Nous sommes donc parties d’un autre point de vue, nous allons partir du principe que l’image non déformer est déjà présente dans le miroir cylindrique puis calculer l’image déformer.



Dans cette image, le point V correspond à la position de la caméra et le point P est un point de l’image non déformer qui apparait dans le cylindre.

# Le code

Le code possède plusieurs fonctions principales permettant la génération et l’affichage des différentes figures.

***GénérerObjet()***

Cette fonction est différente en fonctions du fichier sélectionner, cette fonction a pour but de générer les points des de l’image non déformer qui apparaitre dans le miroir cylindrique

***GenererPlan()***

Cette fonction a pour but de générer le plan sur lequel sera poser le miroir cylindrique ainsi que l’image déformer.

La façon dont le plan est réalisé est que l’on génère 4 points différents puis on les créer une face les reliant ce qui créer le plan.

Une image contenant texte, reçu

Description générée automatiquement

***GenererCylindrique()***

Afin de créer le cylindrique, nous avons décider de le réaliser manuellement en calculant l’ensemble des points du cercle du bas puis du haut puis de créer les différentes faces

Pour cela nous avons utiliser la formule de la coube d’un cercle qui est :

X = R\*cos(theta)

Y = R \* sin(Theta)

Z = 0 ou 12 (en fonction du cercle du haut ou du bas)

***GenererFaceCylindreQuadra()***

Cette fonction a pour but de générer les faces quadratiques du cylindre en reliant les points des deux cercles. Pour cela, on définit les différents points de chacune des faces dans le bon ordre afin que l’ordinateur puisse afficher les faces correctement.

Une image contenant texte

Description générée automatiquement

***GenererImageDeforme()***

Cette fonction à pour but de générer les points et les faces de l’image déformer qui apparait sur le plan. La façon dont les points sont générés dans cette fonction sont lié au calculs différents calculs que nous avons réalisés. Cette fonction va donc générer autant de points que pour l’image non déformer en suivant les différents calculs réaliser.

Les faces seront ensuite créées de la même façon que pour l’image non déformer.

Par exemple, si l’image non déformer possède une face [1,2,4,3] alors l’image déformer possèdera aussi une face [1,2,4,3].

***Affichage ()***

La fonction Affichage à pour but d’afficher l’ensemble des points des différents figure qui apparaitront à l’écran tel que le cylindre, le plan et l’image déformer et non déformer.

***processSpecialKeys ()***

Cette fonction permet de déplacer la caméra en appuyant sur les fléche directionnelle.

Les flèche du haut et du bas permette d’avancer et de reculer respectivement.

Les flèche de gauche et de droit permette de se déplacer sur la droite ou sur la gauche.

Cependant comme la caméra fixe un point précis, si l’on se déplace sur la gauche ou la droite cela donne plus un effet de rotation que de déplacement mais la caméra bouge vraiment.

***Mouse() et mouseMotion()***

Ces deux fonctions travaillent ensemble afin de permettre de déplacer la caméra en appuyant sur le clique gauche et bougeant la souris.

***Clavier()***

Cette fonction permet de modifier l’affichage en appuyant sur les touches du clavier.

P => affichage des figures pleine

F => affichage en mode fil de fer

S => affichage des sommet seul

Q => Quitter le programme

***Main()***

La fonction main initialise et lance le programme. Elle va d’abord appeler toutes les fonctions permettant de calculer les points des différents figures.

Une image contenant texte

Description générée automatiquement

Elle initialise ensuite Le programme OpenGLUne image contenant texte

Description générée automatiquement

Puis pour finir elle associe les différentes fonctions tel que Affichage() ou clavier() au programme OpenGL.

Une image contenant texte

Description générée automatiquement

# Organisation du code

Le projet est divisé en trois fichiers différents

* Figure1.cpp
* Figure2.cpp
* Figure3.cpp

La majorité du code de ces trois fichiers est similaire. La seul différence est la partie fonction genererObjet()