Projet de *Codage numérique*

Sujet : Saturne et ses anneaux

Le but de ce sujet est d'écrire un programme C++ comme vu en cours dans le chapitre 6 (chap. couleurs, pixels et images). Ce programme doit permettre de produire l'image ci-dessous :

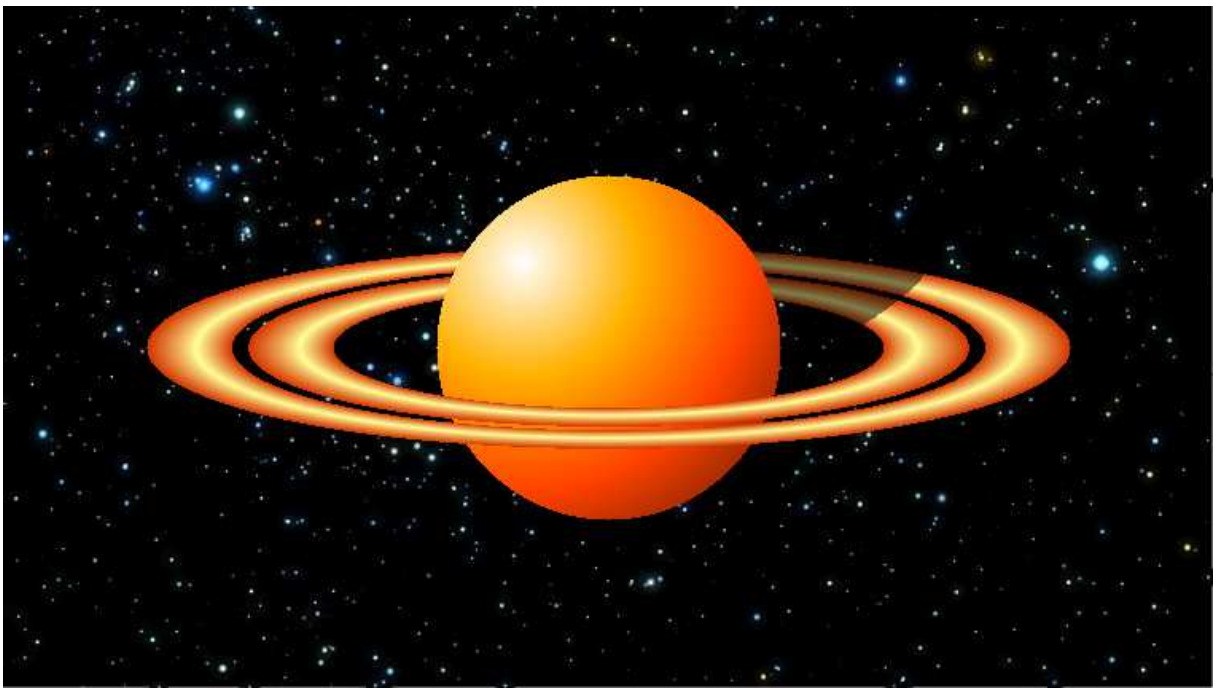


Image calculée (760x428) de la planète Saturne avec ses anneaux

Détaillons les étapes de la construction de cette image.

1. Construction de la planète avec sa tache de lumière et sa zone de pénombre

Le disque représentant la planète peut être construit comme dans l'exercice 3. chap. 6 (drapeau impérial japonais). Notons $centre_{Saturne}(x_{centreSaturne}, y_{centreSaturne})$ la position du centre de la planète et $rayon_{Saturne}$ le rayon de la planète.
La tache blanche diffuse et le dégradé de couleurs sur la planète peuvent être obtenus par alpha-blending comme vu en cours (cf. cours, chapitre 6, couleurs, pixels et images, exercices 11 et 12).
Le mélange de couleurs est effectué en fonction de la distance des pixels au centre de la tache.
Détaillons cette idée :
Soit $p(x,y)$ un pixel situé à l'intérieur du disque (avec x n° de ligne et y n° de colonne) et $centre_{Tache}(x_{centreTache},y_{centreTache})$ la position du centre de la tache.
$centre_{Tache}$ peut être construit en fonction de $centre_{Saturne}$ et de $rayon_{Saturne}$.
On considère alors la distance d entre p et $centre_{Tache}$.

d est utilisée (cf. fig. 1), comme dans l'exercice 12 (chap. 6), pour obtenir la nuance de couleur à inscrire dans le pixel p.

☞ Vous devez écrire, en langage C, la fonction

```
double distance(  const double  x1, const double  y1,
                  const double  x2, const double  y2)
```

qui calcule la distance euclidienne entre les points (x_1,y_1) et (x_2,y_2).

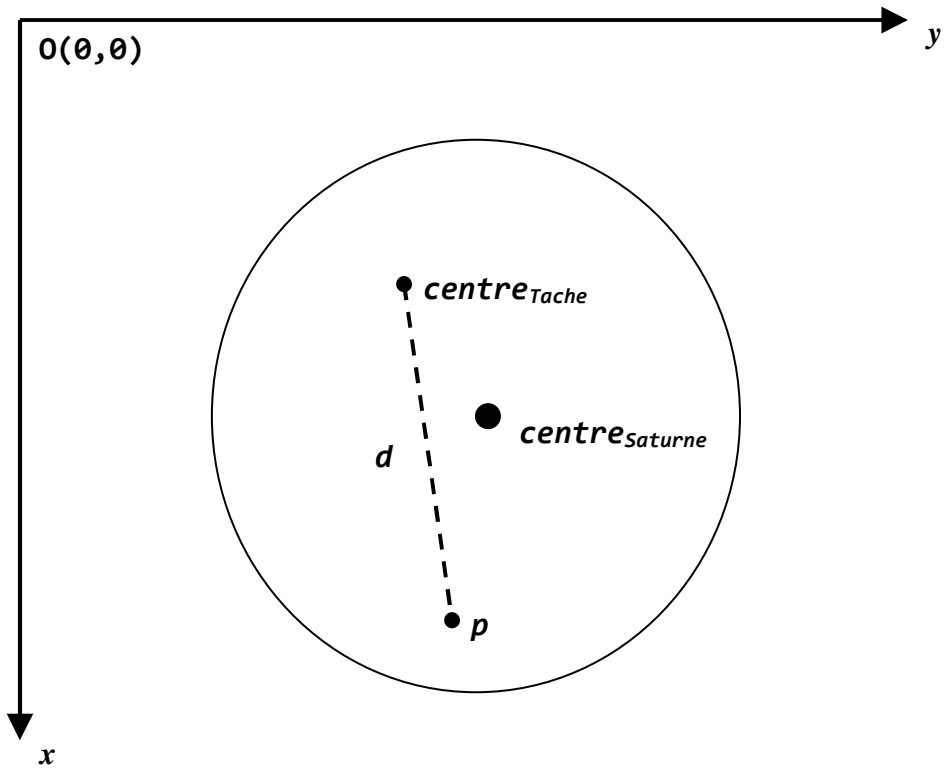


fig. 1. La couleur du pixel *p* dépend de la distance *d*.

2. Construction (géométrique) des anneaux

Cette construction est essentiellement basée sur la notion d'ellipse. Une ellipse est une courbe fermée plane qui généralise la notion de cercle.

Une ellipse *E* est définie par un centre $C_0(x_0,y_0)$ et par deux rayons *a* et *b* tels que $0 < a$ et $0 < b$.

La plus grande des deux valeurs est notée grand rayon et l'autre petit rayon. Si $a = b$, l'ellipse est tout simplement un cercle de centre (x_0,y_0) et de rayon $r = a = b$.
Dans la suite (paragraphes 2 et 3), on suppose que l'ellipse admet pour axes de symétrie l'axe horizontal et l'axe vertical. Dans ce cas l'ellipse prend l'aspect suivant (fig. 2) :

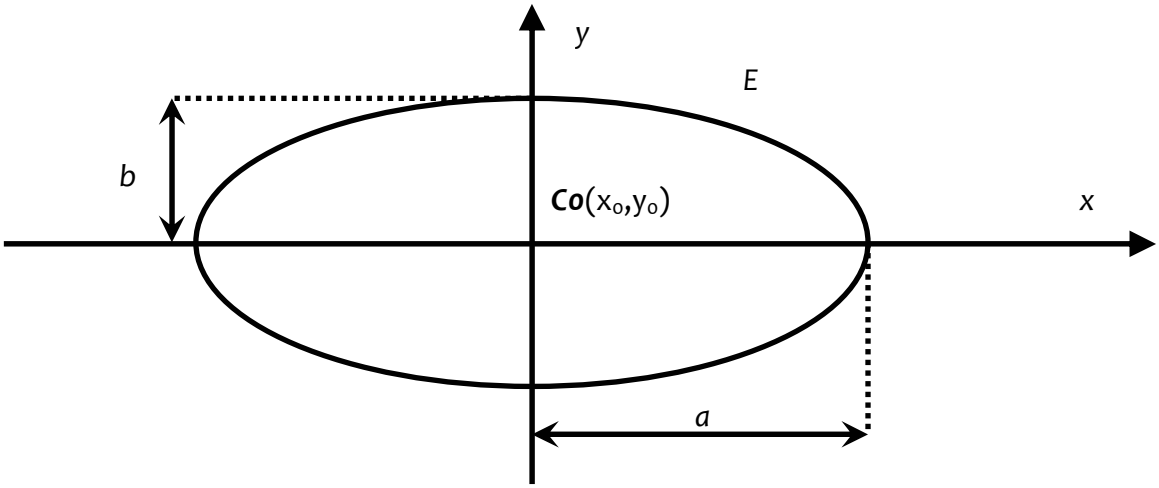


fig. 2

et l'équation cartésienne de l'ellipse *E* devient très simple :

(x,y) ∈ *E* si et seulement si :

$$\left(\frac{x-x_0}{a}\right)^2+\left(\frac{y-y_0}{b}\right)^2-1=0 \qquad (1)$$

Corollaire :
(x,y) est situé à l'intérieur de *E* si et seulement si :

$$\left(\frac{x-x_0}{a}\right)^2+\left(\frac{y-y_0}{b}\right)^2-1\le 0 \qquad (2)$$

De ces résultats on déduit le plan suivant pour construire l'image de Saturne :

1. Ecrire une fonction qui indique si un point *(x,y)* est situé à l'intérieur d'une ellipse de centre (x_0,y_0) et de rayons *a* et *b*.
2. Ecrire une fonction qui indique si un point *(x,y)* est situé à l'intérieur d'un anneau elliptique *A* formé par deux ellipses E_1 et E_2 concentriques de centre (x_0,y_0) et de rayons respectifs a_1, b_1 et a_2, b_2. On suppose que $a_1 < a_2$ et que $b_1 < b_2$.
3. Ecrire une fonction qui indique si un point *(x,y)* est situé à l'intérieur d'un demi-anneau elliptique gauche A_G formé par deux ellipses E_1 et E_2 concentriques de centre (x_0,y_0) et de rayons respectifs a_1, b_1 et a_2, b_2. On suppose que $a_1 < a_2$ et que $b_1 < b_2$.
A_G est défini comme l'ensemble des points $\boldsymbol{p}(x,y)$ tels que $\boldsymbol{p} \in A$ et $x \le x_0$.
4. Ecrire une fonction qui indique si un point *(x,y)* est situé à l'intérieur d'un demi-anneau elliptique droit A_D formé par deux ellipses E_1 et E_2 concentriques de centre (x_0,y_0) et de rayons respectifs a_1, b_1 et a_2, b_2. On suppose que $a_1 < a_2$ et que $b_1 < b_2$.
A_D est défini comme l'ensemble des points $\boldsymbol{p}(x,y)$ tels que $\boldsymbol{p} \in A$ et $x_0 \le x$.

Note : les fonctions d'appartenance d'un pixel à A_G et à A_D sont utiles lors de l'assemblage de la planète et de ses anneaux (cf. paragraphe 5).

☞ Il est obligatoire d'écrire les fonctions (1), (2), (3) et (4) mentionnées ci-dessus.

3. Réalisation du dégradé de couleurs sur les anneaux

Là encore on exploite la notion d'ellipse :
Reprenons les notations du paragraphe précédent et posons :

$$f(x,y) = \left(\frac{x-x_0}{a}\right)^2 + \left(\frac{y-y_0}{b}\right)^2 - 1 \qquad (3)$$

où $f(x,y)$ est la quantité qui intervient dans les équations (1) et (2) du paragraphe 2.
$f(x,y)$ a une signification géométrique et peut être utilisée comme une sorte de distance.

Soit $\boldsymbol{p_1}(x_1,y_1)$ un point situé à l'extérieur de l'ellipse E et $\boldsymbol{p_2}(x_2,y_2)$ un point situé à l'intérieur de l'ellipse E.

Notons $\boldsymbol{q_1}$ (respectivement $\boldsymbol{q_2}$) le point d'intersection de la droite ($\boldsymbol{C_0}$, $\boldsymbol{p_1}$) (respectivement ($\boldsymbol{C_0}$, $\boldsymbol{p_2}$)) et de E (cf. fig.3).

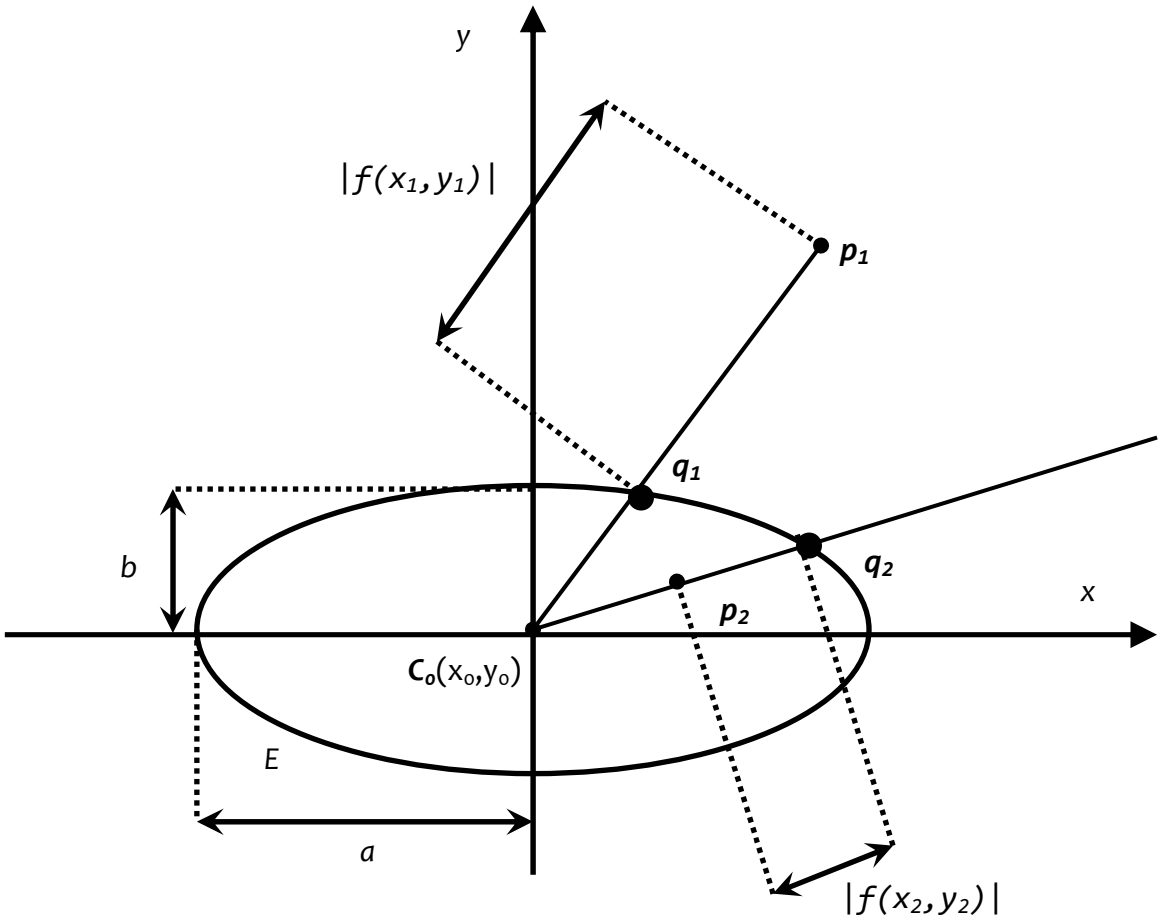


fig.3. Illustration de la notion de *distance elliptique*

Alors :

1. $f(x_1, y_1) > 0$ et $f(x_1, y_1)$ croît si $\boldsymbol{p_1}$ s'éloigne de $\boldsymbol{q_1}$ tout en restant sur la droite ($\boldsymbol{C_0}$, $\boldsymbol{q_1}$).

2. $f(x_2, y_2) < 0$ et $-f(x_2, y_2)$ croît si $\boldsymbol{p_2}$ s'éloigne de $\boldsymbol{q_2}$ tout en restant sur la demi-droite $[\boldsymbol{C_0}, \boldsymbol{q_2})$.

De manière générale, pour un point $\boldsymbol{p}(x,y)$, $|f(x,y)|$ représente une quantité proportionnelle à la distance entre $\boldsymbol{p}$ et le point d'intersection de E et de la droite $(\boldsymbol{C_0},\boldsymbol{p})$ (cf. puissance d'un point par rapport à une courbe plane).

☞ Idée : pour un pixel $\boldsymbol{p}(x,y)$ situé sur un anneau, la quantité $|f(x,y)|$ peut être utilisée comme dans l'exercice 12 (chap. 6), pour obtenir la nuance de couleur à inscrire dans $\boldsymbol{p}$.

☞ Vous devez écrire, en langage C, la fonction qui calcule $|f(x,y)|$ en fonction de x et y.

4. Construction de l'ombre de Saturne sur ses anneaux

La projection perspective d'une sphère sur un plan est toujours une ellipse (car cette projection correspond à l'intersection d'un plan et d'un cône). L'ombre d'une sphère sur un plan est donc toujours une ellipse. Par conséquent, la partie des anneaux qui est à l'ombre de Saturne est définie par l'intersection de plusieurs ellipses. Les ellipses permettant de délimiter les anneaux ont été discutées au paragraphe 2, il reste donc à définir l'ellipse S représentant l'ombre de Saturne.
Dans le cas général, S est une ellipse dont les axes principaux ne correspondent plus aux axes du repère, elle est donc « inclinée » par rapport au repère de l'image (cf. fig. 4).

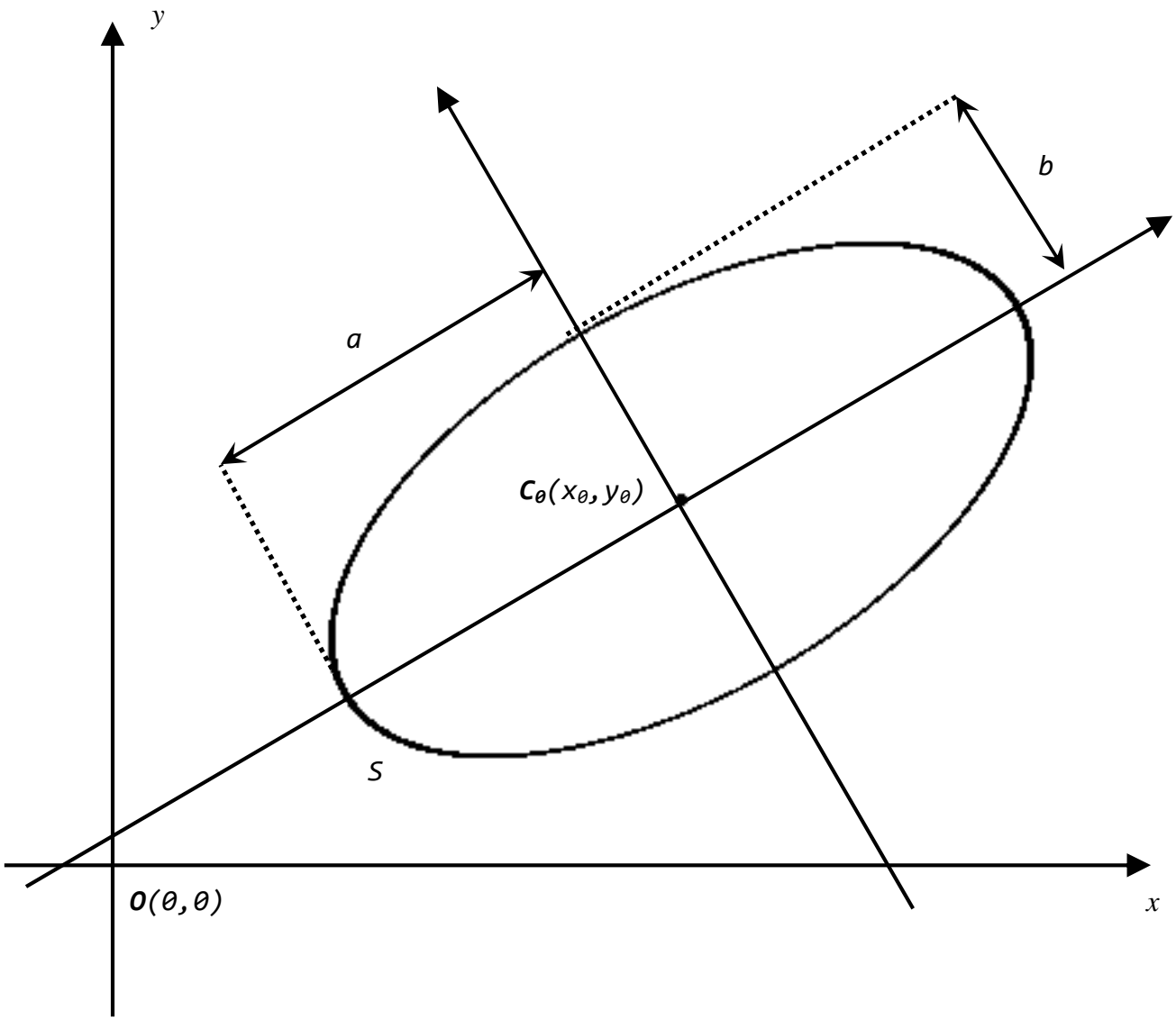


fig. 4. Dans le cas général, l'ombre de Saturne est une ellipse « inclinée » *S*.

L'équation de S, dans le cas général est de la forme :

$$F(x,y) = 0$$

avec

$$F(x,y) = \left(\frac{u}{a}\right)^2 + \left(\frac{v}{b}\right)^2 - 1 \qquad (4)$$

où $u = p\ d_x + q\ d_y$
et
$v = -q\ d_x + p\ d_y$

avec $d_x = x\text{-}x_0$ et $d_y = y\text{-}y_0$
S est donc paramétrée par x_0, y_0, a, b, p et q tels que :

- x_0 et y_0 définissent le centre de l'ombre.
- a et b définissent la forme de l'ellipse.
- p et q définissent l'inclinaison de l'ellipse.

Ces six paramètres peuvent être déterminés expérimentalement.

☞ Vous devez écrire, en langage C, la fonction qui calcule $F(x,y)$ en fonction de x et y.

Notez que pour assombrir une couleur, il suffit de la mélanger avec du noir grâce à la fonction écrite dans l'exercice 11 du chapitre 6 du cours.

5. Assemblage des anneaux et de Saturne

L'image à construire peut être décomposée en quatre parties :

1. le fond,
2. le demi-anneau elliptique supérieur situé en arrière plan,
3. le disque central situé sur l'arrière plan et enfin,
4. le demi-anneau elliptique inférieur situé au premier plan.

Algorithme du peintre

Le remplissage de l'image est décomposé en trois étapes, chaque étape recouvrant partiellement les pixels remplis par l'étape précédente. L'algorithme procède ainsi de la même manière qu'un peintre qui peint un tableau en commençant par l'arrière-plan et qui finit par le premier plan.

Etape 1 : on dessine le demi-anneau elliptique supérieur situé en arrière-plan et le fond.
Etape 2 : on dessine la planète (et on recouvre partiellement le demi-anneau dessiné à l'étape 1).
Etape 3 : on dessine le demi-anneau elliptique inférieur situé au premier plan.

Le rayon de la planète et les rayons a_1, b_1 et a_2, b_2 peuvent être ajustés expérimentalement.

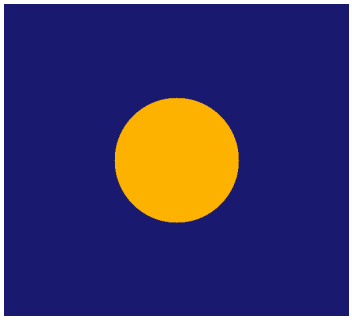
6. Collage de l'image produite sur un fond de ciel étoilé

L'image finale est obtenue par alpha-blending avec une image de ciel étoilé. On peut procéder comme pour l'exercice de l'image de la lune (cf. exercices 9 et 10, chap. 6).

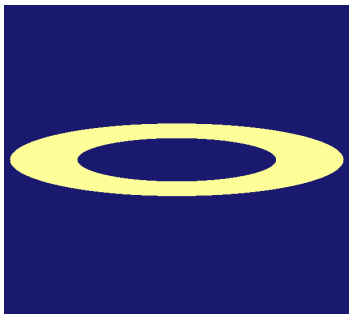
7. Conseils

Pour construire l'image, procédez étape par étape.

7.1 Obtenez d'abord l'image du disque de couleur unie de la planète (facile) :

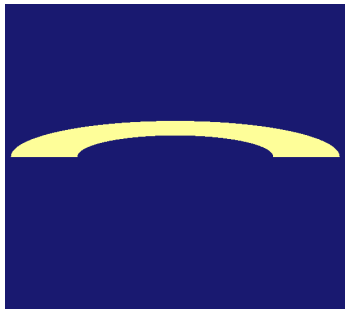


7.2 Construisez ensuite l'image d'un anneau unique de couleur unie (facile).

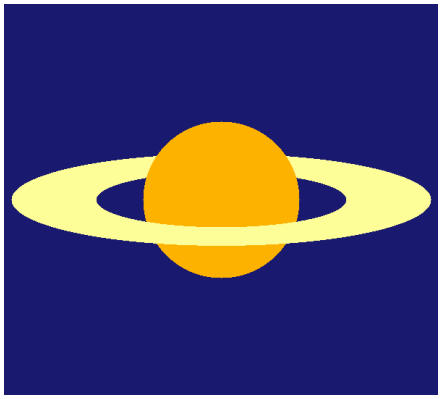
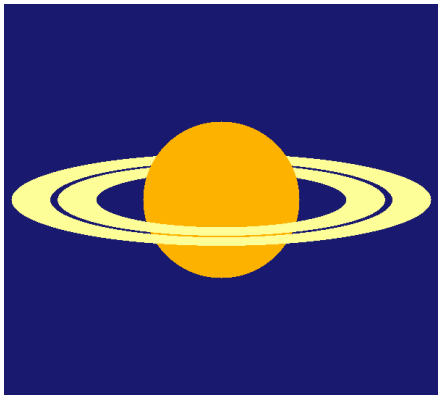


Pour s'exercer, on peut s'entraîner, dans l'ordre, à obtenir d'abord l'image d'une ellipse, puis celle d'un anneau elliptique

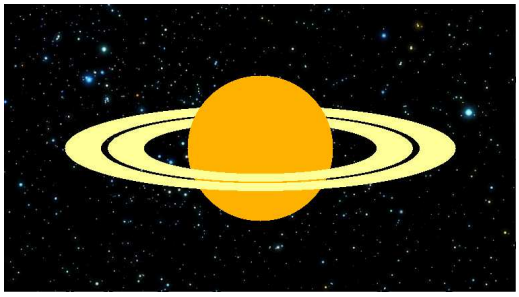
7.3 Puis construisez les images des deux demi-anneaux




7.4 Assemblez enfin la planète et ses anneaux

d'abord  puis 

7.5 Ajoutez le fond étoilé



L'image du fond étoilé est fournie avec le sujet.

7.6 Ajoutez enfin les dégradés de couleurs et l'ombre de la planète sur les anneaux

☞Conservez les différentes versions ! Ecrivez à chaque fois un nouveau programme. Chaque étape réussie apporte des points.

☞Un dernier conseil : faire sur papier un plan détaillé de l'image à construire avant de commencer à programmer !

8. Qualité du code source du programme C écrit

Les paragraphes précédents vous imposent d'écrire les fonctions suivantes :
paragraphe 1 : fonction *distance*
paragraphe 2 : fonctions (1), (2), (3) et (4)
paragraphe 3 : fonction $f(x,y)$
paragraphe 4 : fonction $F(x,y)$

Vous devez aussi vous efforcer d'écrire des fonctions qui s'utilisent les unes les autres.

Ces contraintes ont deux objectifs principaux :

☞Simplifier l'algorithme final

☞Eviter les copiés-collés de code source qui représentent un désastre pour la maintenance du programme.

De manière générale, dans une application informatique, il convient de décomposer les calculs. Par conséquent, il est recommandé d'écrire plusieurs fonctions qui effectuent les calculs intermédiaires.

Par ailleurs, l'image finale contient 760 x 428 = 325 280 pixels. Par conséquent, les instructions contenues dans la boucle de remplissage du tableau de pixels sont répétées 325 280 fois !

☞Afin d'optimiser le temps d'exécution du programme, il convient donc de sortir de la boucle tous les calculs qui peuvent n'être effectués qu'une seule fois, en particulier les calculs de racines carrées, de fonctions trigonométriques, logarithmes et exponentielles qui sont très coûteux en temps.

☞ La notation tiendra compte du strict respect de ces consignes.

9. Langage de développement

Le langage de développement utilisé est C++. Le compilateur et l'éditeur de texte sont libres : MinGW, Visual, Code-Blocks, CLion, Sublime Text, Linux, etc. Les fichiers *make* sont encouragés.
Pour le remplissage d'une image, il est obligatoire, comme vu en cours, de créer un tableau de pixels puis de se servir de la fonction OutilsCreationImage::*creeImage*(...) pour créer le fichier image sur le disque.

10. Extension – Anneaux inclinés

Créer une image où les anneaux de Saturne sont inclinés.

Si cette extension est réalisée en respectant les conditions énoncées dans le paragraphe 8, elle rapporte des points supplémentaires.

11. Organisation du travail

Les étudiants se groupent par équipes de 2 pour réaliser le travail.

12. Documents à rendre

Exécutable de l'application et programmes sources en C++.

13. Date de remise

Tous les documents concernant le projet sont à rendre au plus tard le dimanche 17 décembre 2023 à minuit à l'enseignant que vous avez en cours (D. Michel ou A. Zidna). N'oubliez pas d'indiquer : noms et prénoms des membres de l'équipe, nom de la filière et nom du projet. Le projet doit être rendu par courrier électronique à l'adresse suivante :

Pour M. Michel : domic62@hotmail.com.

Pour M. Zidna : ahmed.zidna@univ-lorraine.fr

14. Soutenance

Les soutenances seront organisées à partir du lundi 18 décembre 2023. Elles se dérouleront à l'UFR MIM. Pour les groupes ayant en cours M. Michel, les lieux de soutenance seront communiqués dans la semaine du 27 novembre 2023. Toujours concernant les groupes dirigés par M. Michel, les créneaux disponibles pour placer ces soutenances seront diffusés à partir du 27 novembre 2023 sur un fichier excel partagé. Les étudiants sont invités à s'y inscrire eux-mêmes pour prendre RDV. Les deux membres de l'équipe doivent être présents. La soutenance dure environ 30 minutes.

15. Notation

La notation prendra en compte :

- La qualité du code source et la rigueur de programmation
- Le fonctionnement de l'application
- Les réponses que vous apporterez sur les questions concernant l'application présentée

N'hésitez pas à nous poser des questions ou à solliciter des RDVs pour discuter de points de programmation.

Bon travail