

Projets INFO31 en L2, année 2017-2018

Dominique Michelucci, Université de Dijon

14 septembre 2017

1 Arithmétique sur de grands nombres, $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$

Par binôme.

Le but de ce projet est de réaliser une arithmétique sur de grands entiers.

Un grand entier sera représenté par un tableau de chiffres. $E[i]$ contient le chiffre de B^i . Utiliser une base B qui est une puissance de 10 est commode : pas besoin de conversion pour afficher les nombres.

Test : calculer factorielle de 1000, en prenant soin de multiplier des nombres de tailles presque égales. Puis diviser le résultat par 2, 3, etc 1000, pour vérifier.

Etape 1 : \mathbb{N} . Planter $+$, $-$, \times , \div , mod et comparaison sur de grands entiers (naturels). Pour la soustraction, on peut calculer $|a - b|$. Utiliser l'algorithme naïf et Karatsuba (expliqué en CM) pour \times . Pour la division, utiliser la méthode russe é : pour calculer a/b , enlever de a la plus grande $2^k b$ possible et incrémenter le quotient de 2^k , et itérer ou récuser sur $a - 2^k b$ divisé par b .

Etape 2. \mathbb{Z} . Gérer les signes ; le plus simple est de représenter un entier relatif par une paire : partie positive, et partie négative. Les deux parties sont des entiers naturels. Ensuite, planter $+$, $-$, \times , \div , mod et comparaison sur des entiers relatifs. Planter le PGCD.

Etape 3. Planter \mathbb{Q} , ou l'algorithme Rho de Pollard (factorisation d'entiers jusqu'à 10^{36}), ou le Log discret : résoudre $a^x = b$ modulo P un premier (pas trop grand).

2 Mini lancer de rayons

Le but est de réaliser un programme de lancer de rayons, par groupe de 3 étudiants.

Etape 1. Définir une classe Rayon avec pour champs : x_0, y_0, z_0 l'origine du rayon, et a, b, c le vecteur de direction du rayon. Normer (a, b, c) . Définir une interface Intersection, qui contient une méthode intersection(Rayon). Définir une classe Cube, ou Sphere, qui implémente Intersection. Au début, Intersection peut ne calculer que le premier point d'intersection, et dans un second temps, vous calculerez une liste d'intervalles le long du rayon (c'est nécessaire pour les opérations booléennes). Le plus simple est sans doute la sphère. On peut se contenter de la sphère de centre $(0, 0, 0)$ et de rayon 1 ; elle a pour équation $x^2 + y^2 + z^2 - 1 = 0$. Le point d'intersection avec le rayon d'équation $x = x_0 + at, y = y_0 + bt, z = z_0 + ct$ nécessite la résolution d'une équation du second degré en t . Il faut aussi une interface Couleur, contenant une méthode couleur en un point x, y, z ; une couleur peut être encodée par un entier $256^2 R +$

$256V + B$ (Rouge, Vert, Bleu sont entre 0 et 255). Au début, chaque objet aura une couleur constante, ensuite vous pourrez la pondérer par l'angle (ou son produit scalaire) avec la direction du soleil. Définir une classe Camera, qui contient un oeil (xO, yO, zO) et trois axes orthogonaux : un axe G (à gauche, (xG, yG, zG)), un axe R (direction du regard (xR, yR, zR)), un axe V (vertical), ainsi qu'un angle ; il y a aussi un nombre de pixels. La classe Camera contient une méthode pour générer le rayon passant par un pixel l, c (ligne, colonne) donné. Vous supposerez pour commencer que tous ses rayons sont parallèles (au vecteur regard). Plus tard, vous pourrez gérer des perspectives centrales (les rayons sont issus de l'oeil). Sauver l'image au format ppm ("<http://netpbm.sourceforge.net/doc/ppm.html>").

Etape 2. Définir une classe Union, qui contient deux objets, et qui implémente l'interface Intersection. Définir une classe Translation (champs Tx, Ty, Tz), et Rotation (axe et angle). Un Objet peut être une Union, une Translation, une Rotation. Il faut écrire la méthode intersection pour ces classes. L'idée est d'appliquer la transformation inverse sur le rayon.

Etape 3. Définir de nouveaux objets : cube unité, cylindre (axe Oz, rayon 1, entre $z = 0$ et $z = 1$), et leur méthode d'intersection avec un objet.

Etape 4. Définir un type Expression : c'est un Nombre, un X, un Y, un Z, un T, une opération Plus, Mult, Moins. L'idée est de représenter certains objets (quadriques, tores, surfaces de Steiner) par leurs équations. Il faut une méthode pour remplacer X, Y, Z par une Expression en T (l'abscisse le long d'un rayon, par exemple remplacer X par $x_0 + aT$), et une méthode pour convertir une expression en T en un polynôme en T : il faut écrire la somme de deux polynômes en T, et le produit de deux polynômes en T. Le polynôme sera représenté par un tableau : $p[i]$ représente le coefficient de T^i . Comment calculer les racines du polynôme sera vu en CM. A l'issue de cette étape, vous pouvez visualiser des tores, des cyclides et d'autres surfaces implicites (chercher sur internet l'équation des surfaces de Steiner, des cyclides, etc).

Etape 5. Faire des animations.

Etape 6. Réaliser les opérations booléennes entre objets. Il faut créer des classes Inter et Differ pour l'intersection et la différence entre deux objets. L'intersection entre un rayon et un objet devient une liste d'intervalles, et il faut écrire des méthodes pour calculer la réunion, la différence et l'intersection entre deux listes d'intervalles ; ça ressemble à la méthode de fusion de deux listes triées dans le tri par fusion.

Etape 7, pour les plus courageux : réaliser des textures 3D (chercher Perlin noise sur internet).