

TP sur les SDF et les surfaces de b茅zier

Florent DIET

November 2022

1 Introduction

2 Primitive

2.1 Sph猫re

La sph猫re est cr茅e a partir d'un centre et d'un rayon.

$$Sphere = Norm(p - centre) - rayon \quad (1)$$

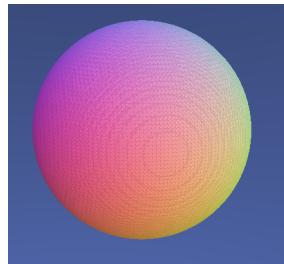


Figure 1: SDF de sph猫re.

2.2 Capsule

La capsule a besoin de deux points A et B pour connaitre sa direction et sa taille et un rayon pour son volume. Ici h represente la projet茅 du point p sur la droite AB sachant que cette valeur est born茅 entre 0 et 1. Soit 0 quand il a d茅pass茅 A et 1 quand il a d茅pass茅 B sinon sa projet茅 se trouve sur le segment. Pour cette primitive je me suis inspir茅 du travail de inigo quillez.

$$h = dotProduct(PA, BA) / dotProduct(BA, BA) \quad (2)$$

$$Capsule = Norm(PA - BA * h) - rayon \quad (3)$$

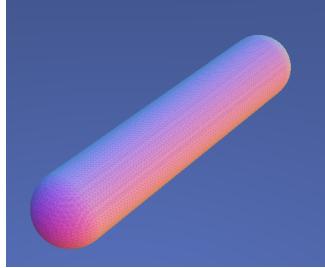


Figure 2: SDF de capsule.

2.3 Cylindre infini

Le cylindre infini a juste besoin d'un vecteur directeur qui va donner l'orientation du cylindre et d'un rayon pour son volume.

$$Norm(Vector(p[0] - A[0], 0, p[2] - A[2])) - rayon; \quad (4)$$

2.4 Cylindre fini

Le cylindre fini utilise la formule de la capsule et renvoie 0 dès que le point projeté sur la droite AB est en dehors du segment AB.

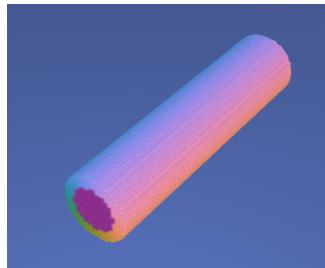


Figure 3: SDF de cylindre fini.

2.5 Cone

Pour le cone je me suis rendu compte que c'était ni plus ni moins qu'une capsule avec un rayon variable donc tout simplement j'ai multiplié mon rayon par h. Aussi puisque sa me faisait une glace du au fait que arrivé au point B, h restait à 1 donc il me reproduisait le bout d'une capsule j'ai forcé h à être à 0 quand il se retrouve sur B.

$$Norm((p - A) - (B - A) * h) - rayon * h; \quad (5)$$

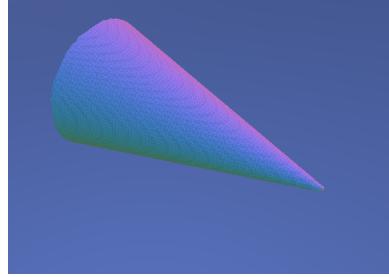


Figure 4: SDF de cone.

2.6 Graine

Pour Créer un oeil j'ai voulu refaire la forme de graine de son contour. J'ai d'abord commencé par faire l'intersection de 4 sphère cependant le résultat ne me satisfaisait pas. Apres quelques essaie j'ai vu que le mélange d'un cylindre infini et d'une sphère faisait l'affaire.

$$cylindreInfini \circ sph\acute{e}re \quad (6)$$

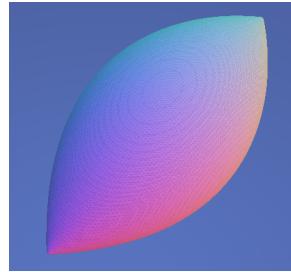


Figure 5: SDF de graine.

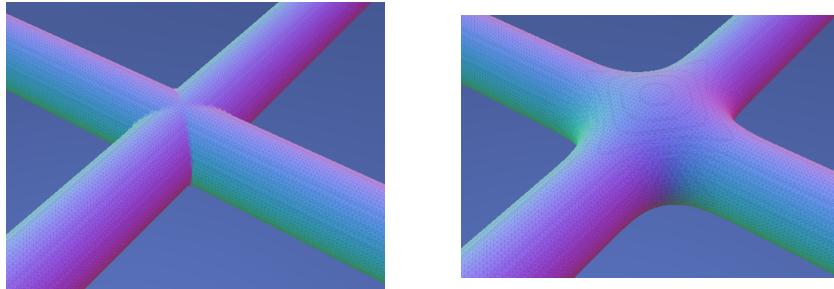
3 Operation

3.1 Union / smooth union

Pour toute les fonctions "smooth" j'ai repris la formule du cours qui invitait à soustraire un offset. r représente l'intensité du smooth que l'on veut appliquer.

$$\min(NodeA, NodeB) \quad (7)$$

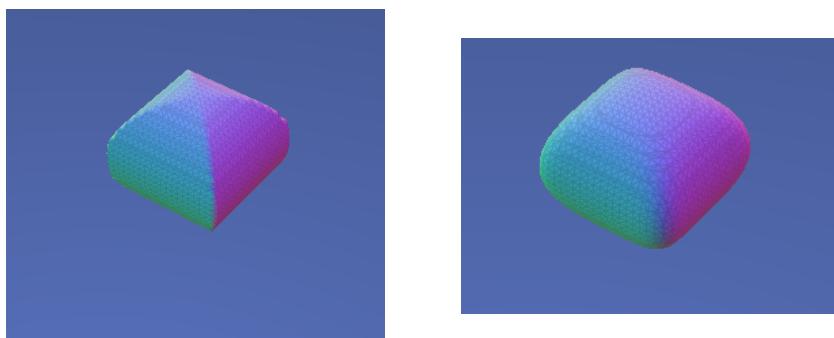
$$offset = \frac{r * (\frac{\max(r - |a - b|, 0)}{r})^3}{6} \quad (8)$$



3.2 intersection / smooth intersection

J'ai ajouté l'offset pour faire le smooth intersecion.

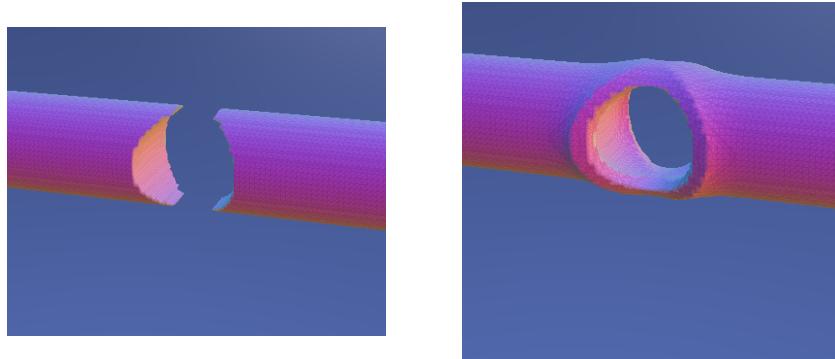
$$\max(\text{NodeA}, \text{NodeB}) \quad (9)$$



3.3 Difference / smooth difference

J'ai soustrait l'offset pour faire le smooth intersecion.

$$\max(\text{NodeA}, -\text{NodeB}) \quad (10)$$



3.4 Mélange

$$NodeA + NodeB \quad (11)$$

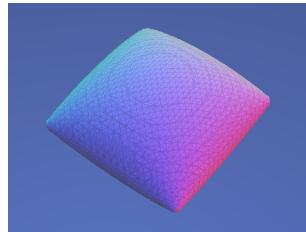


Figure 6: Opération de mélange.

3.5 Deformation

Pour déformer mon node j'ai du implémenter des matrices j'ai mis un exemple dans la partie "des yeux sur une sphère".

$$NodeA -> value(ApplyMatrice(P)) \quad (12)$$

3.6 Twist

Pour le twist j'ai repris la formule proposé par inigo quillez.

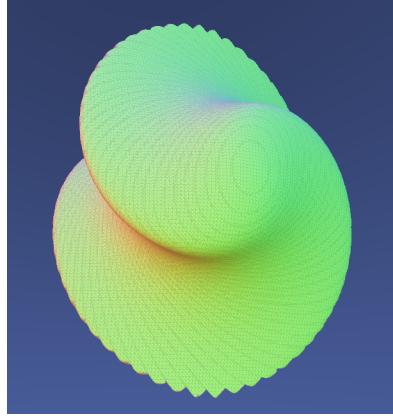


Figure 7: le twist d'une graine.

4 Autre

4.1 Matrice de translation / rotation / scale

J'ai repris les formules du cours et les cours de jean claude iehl pour pouvoir créer ma classe de matrice.

4.2 Ray marching / Sphère tracing

Pour le ray marching l'utilisateur doit donner un node et 2 points afin de pouvoir créer la droite qui va nous donner le point de contact. J'avance sur cette droite d'une distance fixe.

Pour le sphere tracing j'avance sur cette droite d'une distance variable en fonction de où se trouve le points que je suis en train d'analyser par rapport à mon node. Plus le point est loin de mon node plus la distance parcouru sur la droite sera élevée. On peut noter une nette amélioration en terme d'itération et de temps comparé au ray marching.

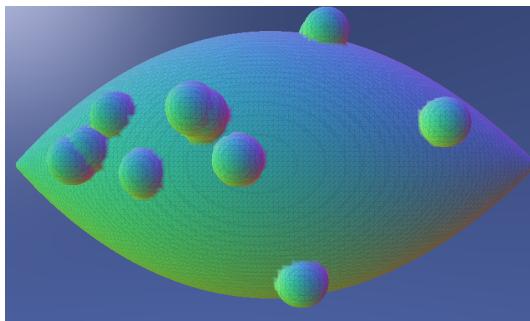


Figure 8: Exemple de ray marching / sphere tracing d'ajout de sphère.

5 Joli création

5.1 Minion

Pour me familiariser avec les surfaces implicites j'ai commencé par faire un "minion", créature que l'on retrouve dans le film "moi moche et méchant". Je suis donc resté avec des formes très simples comme des cônes, des capsules et des sphères que j'ai ensuite collé ensemble à l'aide d'opérations.

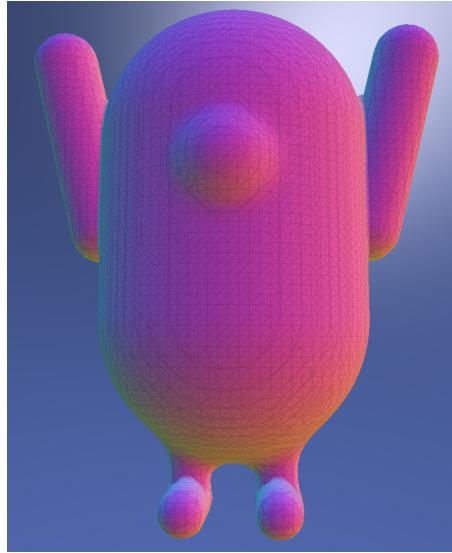


Figure 9: Minion créée grâce à des surfaces implicites.

5.2 Oeil

Ensuite j'ai voulu faire un œil et c'est pour cela que j'ai créé la classe graine qui m'a permis de faire le contour de mon œil. Puis j'ai appliqué une différence entre le cône et cette graine pour faire un creux afin de placer l'œil qui est une sphère.

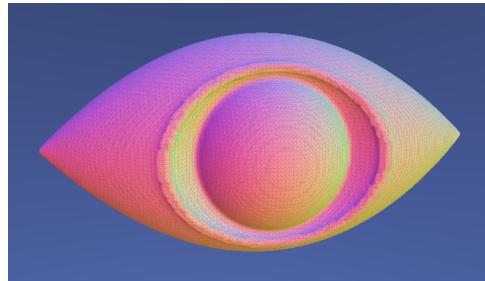


Figure 10: oeil.

5.3 Des yeux sur une sphère

Apres avoir fait un oeil j'ai voulu pouvoir le répercuter autour d'une sphère et c'est a partir de la ou j'ai utilisé mes matrices. J'ai d'abord réduit la taille de l'oeil avec le scale puis en translatant cette oeil et en le tournant selon le nombre d'oeil que je voulais insérer dans un anneau. Apres avoir fait 1 seul anneau j'ai pu le duppliquer sur lui même en appliquant une rotation a chaque fois a celui ci ce qui m'a donné le résultat escompté.

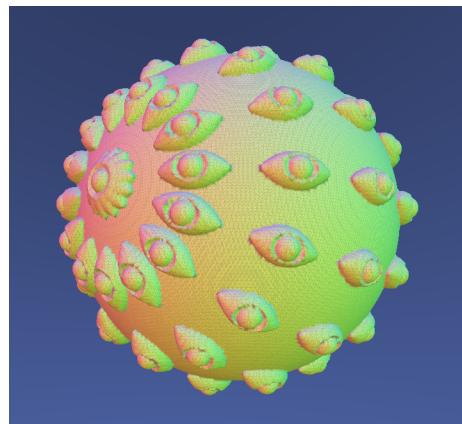


Figure 11: multitude d'oeil mis sur une sphère.

6 TP2

J'ai appliqué l'algorithme de castlejau afin de faire ma surface de bézier.

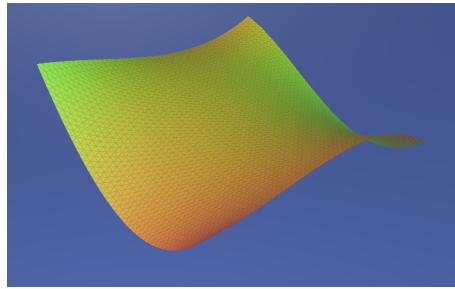


Figure 12: une surface de bázier.

Je lui ai appliqué le twist en utilisant la même formule que j'ai pu utiliser pour mes nodes.

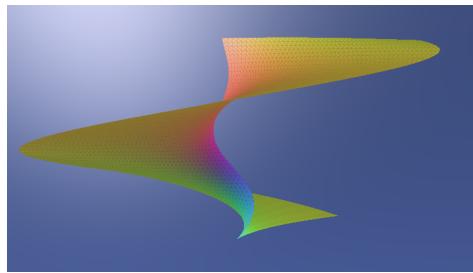


Figure 13: une surface de bázier twisté.