

Algorithme de triangulation en Python

Auteur : Eng. Sergio Adrián Martin

Introduction :

Le processus de triangulation est bien connu des ingénieurs et des architectes et est utilisé pour mesurer la superficie de polygones irréguliers. Il implique de connaître le côté de chaque segment droit du polygone et l'angle extérieur entre chaque côté et le suivant.

Base du processus.

En théorie, si un polygone a n côtés, il peut être divisé en $(n-2)$ triangles. Si vous pouvez calculer la hauteur et la base de chaque triangle, vous pouvez calculer l'aire de chaque triangle, puis additionner toutes les aires des triangles pour obtenir l'aire finale.

Cependant, le calcul de chaque base et de chaque hauteur peut être compliqué, à moins d'utiliser des méthodes graphiques.

Changement de coordonnées et utilisation de vecteurs.

Si nous avons dans deux listes ou vecteurs les valeurs de la longueur de chaque côté, nous pouvons supposer comme référence que le premier côté est sur l'axe x en coordonnées cartésiennes, le premier sommet du polygone serait dans les coordonnées $(0,0)$ et le deuxième sommet dans $(côté1,0)$.

En supposant que chaque côté agit comme un vecteur d'une magnitude égale à la longueur du côté, son orientation serait donnée par la somme de tous les angles externes avant ce lac.

Voir l'équation

$$A(i) = \text{somme}(o(i))$$

Ensuite, pour calculer les positions de chaque sommet du polygone, nous devrions considérer les éléments suivants

$$X(i) = x(i-1) + l(i) \cos(a(i))$$

$$Y(i) = y(i-1) + l(i) \sin(a(i))$$

Une fois les coordonnées de tous les sommets calculées, le produit vectoriel peut être utilisé pour calculer l'aire de tous les triangles dans lesquels le polygone est divisé.

Pour ce faire, la formule suivante est appliquée.

$$S = \left(\frac{1}{2} \right) \sum_{i=0}^{n-1} x_i y_{i+1} - x_{i+1} y_i$$

Où s est la surface totale du polygone et le sommet n doit être $(0,0)$ pour fermer le polygone.

Le traitement de ces grandeurs vectorielles peut être accompli en utilisant des listes ou des tableaux en python.

Le code source correspondant est présenté ci-dessous.