

Explications sur les benchmarks

Nous avons réalisé des benchmarks sur les **opérations les plus importantes** de la classe **ConvexPolygon** :

- addVertex
- contain
- convexHull
- intersects
- isConvex
- sortVertices

Pour chaque exécution des benchmarks, on génère un fichier texte qui recense les temps d'exécution des opérations, et uniquement leurs temps. C'est-à-dire qu'on prend le temps juste avant l'opération et juste après (on ne prend pas en compte dans les résultats les temps de génération de points, d'ajout à un polygone, etc.).

Afin de pouvoir générer des polygones convexes avec un très grand nombre de sommets, nous avons écrit la méthode **generatePoints**, celle-ci prend en paramètre un nombre entier. Elle retourne le nombre de points demandés dans un tableau. Elle est elle-même basée sur la méthode **getNPointsOnCircle** de la classe **PointDouble**. Celle-ci retourne le nombre de points désirés appartenant à un même cercle, ce qui nous permet de générer de très grands nombres de points qui pourront tous être ajoutés à un polygone convexe sans le rendre concave.

Opération addVertex :

Pour cette opération, on crée un polygone convexe avec le nombre de sommets correspondant au paramètre reçu en entrée, et on mesure le temps nécessaire pour faire l'insertion d'un sommet supplémentaire à ce polygone.

Opération contain :

Pour cette opération, on crée un polygone convexe avec le nombre de sommets correspondant au paramètre reçu en entrée, et on mesure le temps nécessaire pour vérifier si ce polygone contient un point qu'on génère de façon aléatoire.

Opération convexHull

Pour cette opération, on génère le nombre de points correspondant au paramètre reçu en entrée, et on mesure le temps nécessaire pour calculer l'enveloppe convexe de cet ensemble de points.

Opération intersects

Pour cette opération, on crée deux polygones convexes avec le nombre de sommets correspondant au paramètre reçu en entrée, et on mesure le temps nécessaire pour vérifier s'il y a une collision entre les deux polygones.

Opération isConvex

Pour cette opération, on crée un polygone convexe avec le nombre de sommets correspondant au paramètre reçu en entrée, et on mesure le temps nécessaire pour vérifier si ce polygone est convexe. Ici on teste le pire cas, c'est-à-dire que le polygone est convexe, et donc le temps d'exécution est le plus long (si un polygone est désigné comme convexe, c'est qu'on aura testé tout avant de retourner vrai).

Opération sortVertices

Pour cette opération, on génère un tableau de points (de taille correspondante au paramètre reçu en entrée) qu'on mélange aléatoirement. On crée ensuite un polygone à partir de ce tableau, et on mesure le temps nécessaire pour trier les sommets du polygone.