



Affichage de formes Géométriques

Préparation

Le code fourni est une base permettant de dessiner les formes géométriques sur un canvas de pixels.

La classe `Paint` contient les données de chaque pixel, avec les méthodes `set_pixel` pour modifier un pixel, et `show` pour afficher le résultat.

À partir de cela, votre tâche est d'implémenter des *classes* représentant des formes géométriques, en vous aidant des classes prédéfinies:

- `Point` est une classe représentant un point à partir de ses coordonnées
- `Ligne` est une classe représentant la ligne rejoignant 2 points, permettant de les dessiner

1 - Parallélogramme

1. Créer une classe `Parallelogramme`, son constructeur prends *3 points* en paramètres
2. Dans la fonction `__init__`, calculer les coordonnées du 4e point:
 - Sa coordonnée D_x est définie par: $D_x = C_x + (B_x - A_x)$
 - Sa coordonnée D_y est définie par: $D_y = C_y + (B_y - A_y)$
3. Créer les 4 objets `Ligne` correspondant aux lignes entre chacun de ces points
4. Créer une méthode `draw` qui appelle la méthode `draw` de chacune des lignes
5. Tester et valider le fonctionnement

2 - Rectangle

1. Créer une classe `Rectangle` qui hérite de la classe `Parallelogramme`, son constructeur prends 3 paramètres:
 - 2 points A et B
 - La longueur séparant A et C , notée AC ici

2. Dans la fonction `__init__`, calculer les coordonnées du 3e point C :

Calculer la distance AB séparant le point A au point B

$$C_x = A_x + AC \times \frac{B_y - A_y}{AB}$$

$$C_y = A_y - AC \times \frac{B_x - A_x}{AB}$$

3. Enfin, initialiser la classe parente `Parallelogramme` en lui passant en paramètre les 3 points A , B et C .
4. Tester et valider le fonctionnement

3 - Carré

1. En suivant le même principe, créer une classe `Carre`

Note: Un carré est un rectangle dont chaque côté est égal

2. Tester et valider le fonctionnement

4 - Triangle quelconque

1. Créer une classe `Triangle`, dont le constructeur attends 3 paramètres de type `Point`, et créé des `Ligne` entre ces points.
2. Créer sa fonction `draw` affichant chacune de ces lignes
3. Tester et valider le fonctionnement

5 - Triangle rectangle

1. Créer une classe `TriangleRectangle` héritant de la classe `Triangle`

Son constructeur prends en paramètre 2 points A et B , et un angle θ en degrés

Dans le constructeur, nous allons calculer les coordonnées du point C tel que ABC forme un triangle rectangle

2. Convertir l'angle θ en radians

Calculer la distance entre le point B et C en utilisant la formule $BC = \frac{AB}{\cos(\theta)}$

Calculer la distance entre le point A et le point C en utilisant la formule $AC = BC \times \sin(\theta)$

On peut alors calculer les coordonnées du point C :

$$\begin{aligned} C_x &= A_x + AC \times \frac{B_y - A_y}{AB} \\ C_y &= A_y + AC \times \frac{B_x - A_x}{AB} \end{aligned}$$

3. Initialiser alors la classe parente `Triangle` à partir des 3 points.

4. Tester et valider le fonctionnement de cette classe

6 - Triangle isocèle

1. Créer une classe `TriangleIsocele` héritant de la classe `Triangle`

Son constructeur prends en paramètre 2 points A et B formant la base, et un angle θ en degrés correspondant aux angles aux extrémités

2. Calculer les coordonnées du point au milieu du segment AB , que l'on appellera M

Créer un `TriangleRectangle` sur les points M , B , et avec l'angle θ

Ce triangle rectangle correspond à "la moitié" de notre triangle isocèle

3. Initialiser la classe parente `Triangle` avec les points A , B et le point C

qui a été calculé lors de l'initialisation du triangle rectangle

7 - Triangle rectangle isocèle

En réutilisant les composants précédents, créer une classe `TriangleRectangleIsocele`

Un triangle rectangle isocèle est un triangle isocèle dont les angles aux extrémités ont une valeur de 45°

8 - Triangle équilatéral

De même, créer une classe `TriangleEquilateral`, qui est un triangle isocèle dont les angles aux extrémités ont une valeur de 60°

9 - Calcul de périmètre

Pour chacune des classes parentes, créer une méthode `perimeter` qui retourne la somme de la longueur de chaque ligne composant le polygône

Valider le fonctionnement avec des cas simples (carré, triangle équilatéral, etc...)

10 - Calcul d'aire

Pour chacune des classes parentes, créer une méthode `area` qui retourne l'aire du polygône créé

Aire d'un triangle

Calculer un nombre $S = \frac{AB + BC + AC}{2}$

L'aire du triangle ABC est alors $\sqrt{S \times (S - AB) \times (S - BC) \times (S - AC)}$

Note Cette méthode alternative est appelée **formule de Héron**

Aire d'un parallélogramme

Un parallélogramme ABDC peut être divisé en deux triangles ABC et BDC égaux.

L'aire du parallélogramme est alors 2 fois l'aire du triangle ABC