

# Projet Fenêtrage et Remplissage - Mathématiques infographie

LIRZIN Léo et TALVET Ivo

December 16, 2025

## 1 Introduction

Ce programme en Python permet de créer et manipuler des polygones dans un plan interactif à l'aide de l'interface Tkinter. L'utilisateur peut tracer des polygones et des fenêtres polygonales, appliquer un fenêtrage polygonal, et remplir des polygones en utilisant différentes règles de remplissage.

Les principaux algorithmes implémentés sont :

- **Marche de Jarvis** pour l'enveloppe convexe d'un nuage de points
- **Ear Cutting** pour la triangulation de polygones concaves
- **Fenêtrage Sutherland-Hodgman** pour découper des polygones par une fenêtre convexe
- **Remplissage LCA** (scanline) avec la règle de parité ou de l'enroulement

Cette documentation présente :

1. Les interactions utilisateur et l'utilisation du programme
2. Les algorithmes géométriques
3. Les classes `Polygon` et `App`
4. La fonction `main`

## 2 Interactions Utilisateur

### 2.1 Tracé de polygones et de fenêtres

- **Clic gauche** : ajoute un sommet au polygone ou à la fenêtre en cours de création
- **Double clic gauche** : termine le polygone ou la fenêtre en cours

## 2.2 Menu contextuel

- **Clic droit** : ouvre un menu contextuel permettant :
  - de choisir la couleur du polygone, de la fenêtre ou du remplissage
  - de changer la règle de remplissage (parité ou enroulement)
  - d’activer ou désactiver le fenêtrage et le remplissage
  - de basculer entre le tracé de polygone et le tracé de fenêtre
  - de réinitialiser tous les polygones et fenêtres

## 2.3 Zoom et Pan

- **Molette de la souris** : zoom avant ou arrière centré sur le curseur.
- **Clic molette + déplacement** : déplace le plan (pan).

## 2.4 Affichage

Les polygones tracés sont affichés avec leur couleur choisie. Les polygones découpés par les fenêtres apparaissent en rouge si le fenêtrage est activé, et peuvent être remplis selon la règle choisie.

# 3 Algorithmes Implémentés

## 3.1 Marche de Jarvis (Jarvis March)

**Objectif** : calculer l’enveloppe convexe d’un ensemble de points

**Principe** :

1. Sélectionner le point le plus à gauche.
2. Depuis ce point, choisir le point le plus à gauche par rapport à la ligne actuelle.
3. Ajouter ce point à l’enveloppe.
4. Répéter jusqu’à revenir au point initial.

## 3.2 Triangulation Ear Cutting

**Objectif** : découper un polygone concave en triangles pour faciliter le fenêtrage ou le remplissage

**Principe** :

1. Identifier un “ear” (triangle convexe dont aucun autre point du polygone n’est à l’intérieur).
2. Supprimer ce triangle et le stocker.
3. Répéter jusqu’à ce qu’il ne reste qu’un triangle.

### 3.3 Fenêtrage Sutherland-Hodgman

**Objectif :** découper un polygone par une fenêtre polygonale convexe

**Principe :**

1. Traiter chaque côté de la fenêtre polygonale
2. Pour chaque segment du polygone, calculer l'intersection avec le côté de la fenêtre
3. Conserver uniquement les points situés à l'intérieur de la fenêtre
4. Répéter pour tous les côtés de la fenêtre

### 3.4 Remplissage LCA (Scanline)

**Objectif :** remplir un polygone avec une couleur selon la règle de parité ou de l'enroulement

**Principe :**

1. Construire la Structure d'Intersection (SI) pour toutes les arêtes non horizontale
2. Pour chaque ligne de balayage (scanline) :
  - Mettre à jour la Liste des Arêtes Actives (LCA).
  - Trier les intersections en  $x$  croissant
  - Remplir entre intersections selon la règle choisie
3. Mettre à jour les coordonnées  $x$  pour la ligne suivante

## 4 Classes et Structures

### 4.1 Classe Polygon

Représente un polygone avec :

- `points` : liste de tuples  $(x, y)$
- `color` : couleur du polygone

### 4.2 Classe App

Gère l'interface graphique, les interactions et le traitement des polygones

- **Attributs principaux :**
  - `canvas` : zone de dessin Tkinter
  - `polygons`, `windows` : listes des polygones et fenêtres
  - Couleurs, zoom, pan et mode de remplissage

- **Méthodes principales :**

- Gestion des événements souris : ajout de points, fin de polygone, menu, zoom, pan
- `redraw` : efface et redessine tous les polygones
- `process_clip_fill` : applique fenêtrage et remplissage
- Méthodes pour choisir les couleurs et basculer les modes

## 5 Fonction main

```
1 if __name__ == "__main__":  
2     root = tk.Tk()  
3     root.title("Fen trage et remplissage")  
4     app = App(root)  
5     root.mainloop()
```

La fonction `main` initialise Tkinter, crée une instance de l'application et lance la boucle principale.