Génération de labyrinthes PACMAN

TER S5, M1 Informatique

Lesage Arno, Razafindrabe Keryann, Viale Jean-Jacques

EUR DS4H - Université Côte d'Azur

github.com/KeryannR/TER_S1_F

Sommaire

,

1. Algorithmes de génération

Qu'est ce qu'un labyrinthe PACMAN ?

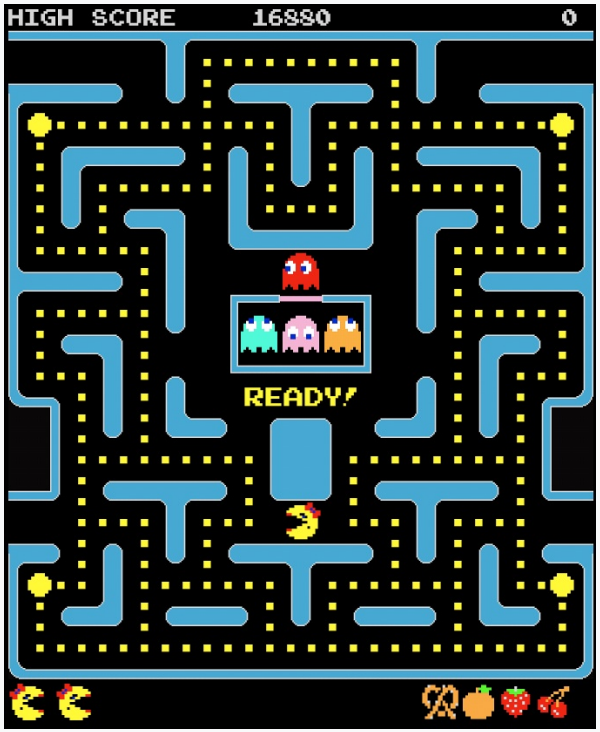


Figure: Un labyrinthe PACMAN

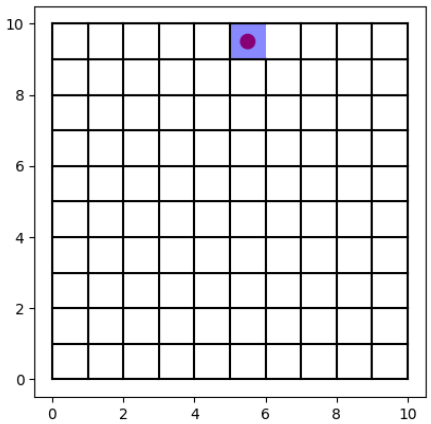
Un labyrinthe PACMAN est généralement caractérisé par :

- La présence de boucles,
- La présence de zones inaccessibles,
- Une 4-connexité,
- Une presque symmétrie.

Un premier algo : *Hunt & Kill*

Algorithm 1 *Hunt & Kill*

1: $\forall (x, y) \in [\![1, n]\!] \times [\![1, m]\!] : c_{x,y} \leftarrow 0$
2: Choisir une cellule c aléatoirement
3: **while** $\exists (x, y) : c_{x,y} = 0$ **do**
4: $\quad c_{x,y} \leftarrow 1$
5: $\quad$ Choisir un voisin $c' : c'_{x,y} = 0$
6: $\quad c \leftarrow c'; c_{x,y} \leftarrow 1$
7: $\quad$ Chercher $c : \exists (x, y)\ c'_{x,y} = 0$



Problèmes :

- Labyrinthes parfaits,
- Possible biai sur les bords,
- Ne ressemble pas vraiment à PACMAN…

Une deuxième idée...

Tetris

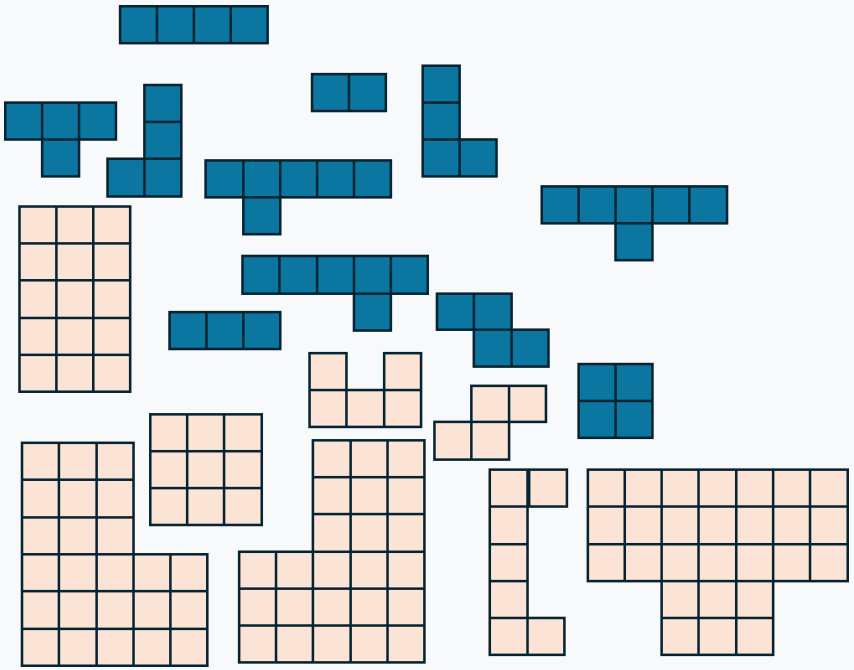


Figure: Quelques Polyominos

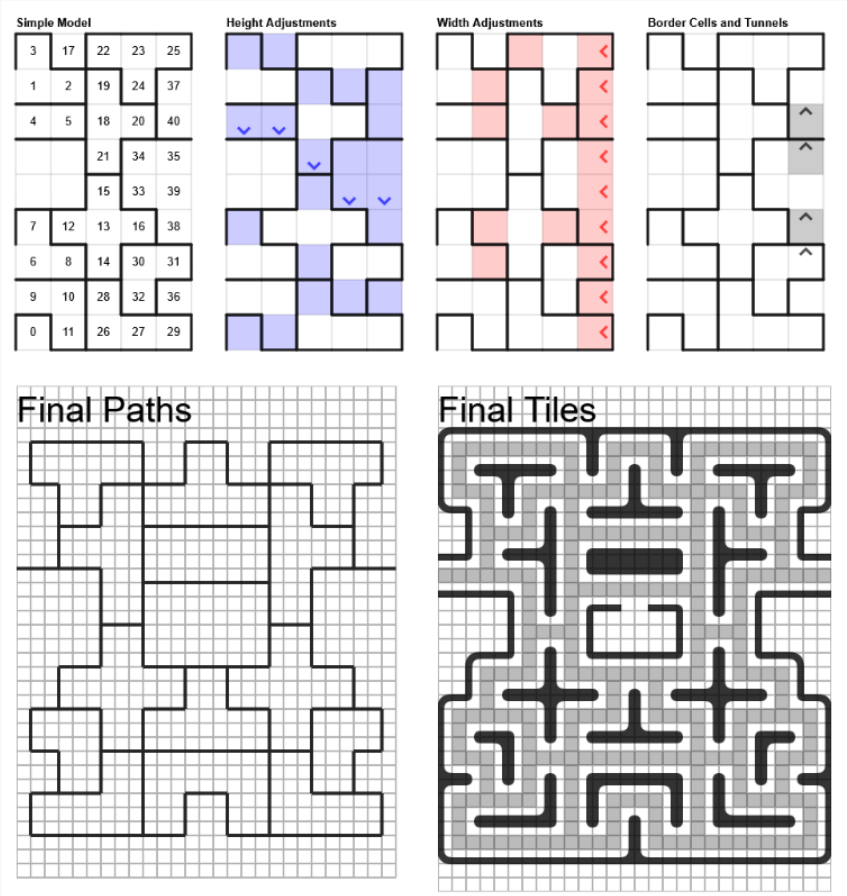


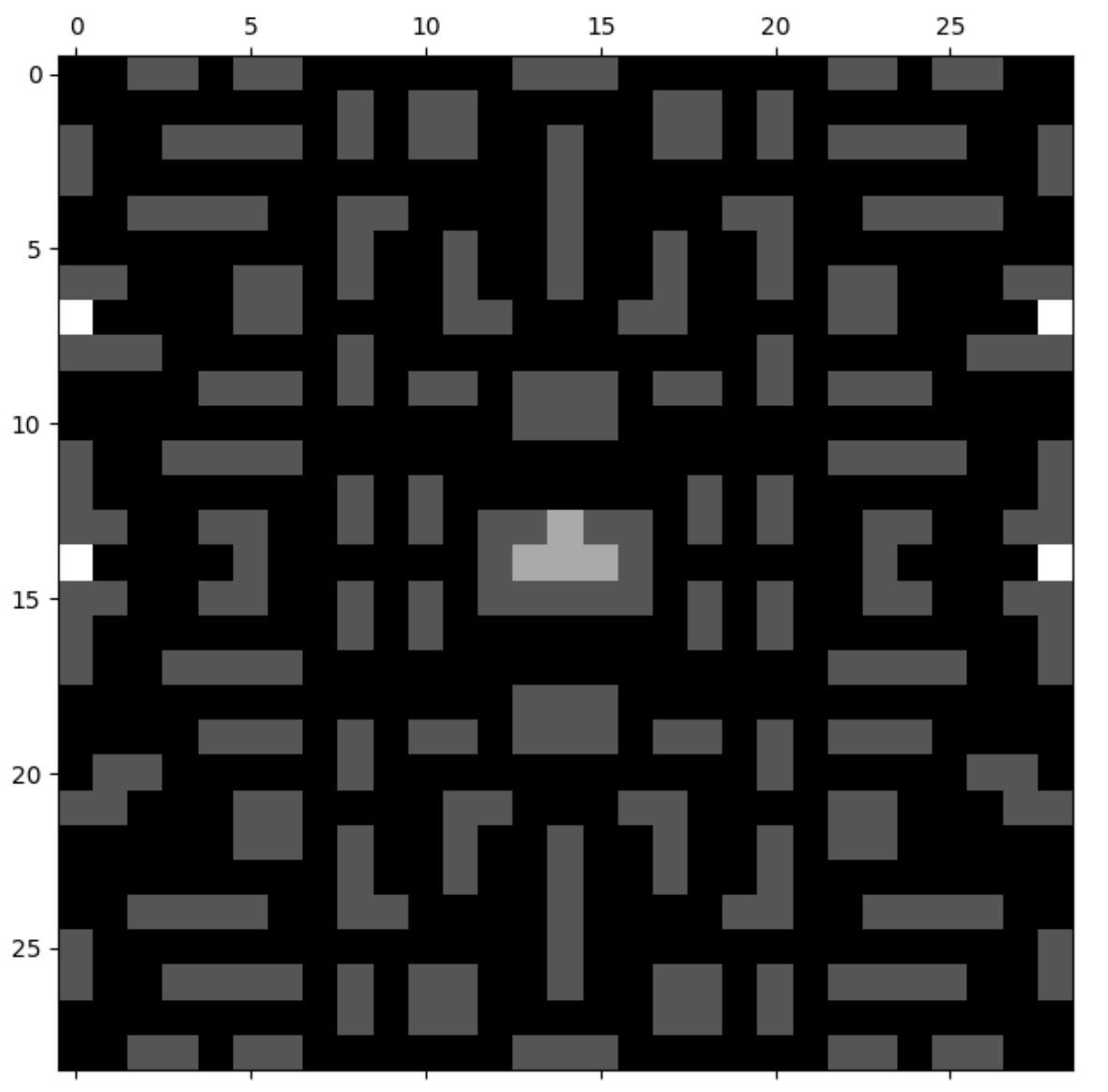
Figure: Résultat ?

Modified Tetris : *Algorithme*

Algorithm 2 *Modified Tetris*

Soit $\mathbb{P}$ l'ensemble des polyominos disponibles
for $i = 0,\ i < n,\ (n \in \mathbb{N})$ **do**
    Choisi $p \in \mathbb{P}$
    Faire $m \in \mathbb{N}$ rotations à p
    Choisir une zone (x, x', y, y') de forme p
    if $\exists(c_{x,y}) \in (x, x', y, y') : c_{x,y} = 1$ **then**
        continue
    Placer p dans (x, x', y, y')
Placer la cage aux fantômes
Retirer les "spikes" aux bords
Retirer la 8-connexité
Placer les portails

Modified Tetris : *Résultat*



2. Évaluation et métriques

3. API

Génération de labyrinthes via l'API

L'API permet de générer dynamiquement des labyrinthes PAC-MAN en utilisant des paramètres personnalisés. **Points clés :**

- **Endpoint :** `/generate` (GET)
- Paramètres :
  - `xSize`, `ySize` : dimensions du labyrinthe
  - `seed` : graine aléatoire pour la reproductibilité
  - `nStep` : nombre d'itérations pour la génération
  - `maxBorderSpikeSize` : taille maximale des prolongements de bord
  - `includeTile` : inclusion de tuiles "spéciales"
- Retour : JSON contenant le labyrinthe, les métriques et les options de génération

Exemple d'utilisation : `/generate?xSize=15&ySize=15&nStep=200`

4. Tests

5. PACMAN

6. GitHub et collaboration

Merci de votre attention !