

Lot H

Tom Gilgenkrantz

1^{er} juin 2024

Résumé

Ce document est un rapport expliquant mes idées et mon implémentation du lot H, le lot libre d'informatique.

1 Introduction

Mon idée principale pour le lot H a été d'étendre le jeu initial à un jeu en 3 dimensions. Cela permet de complexifier le jeu et de rendre les parties plus intéressantes. D'autres modifications plus ou moins importantes ont été apportées pour améliorer le jeu et seront aussi détaillées.

2 Le jeu

J'ai fais le choix pour une question d'interface de verouiller la taille du sac de pièces à 3. Comme le jeu est en 3 dimensions, on a donc possibilité de faire tourner les pièces du sac pour les voir sous toutes leurs coutures :

Pièce 1 (3 points)

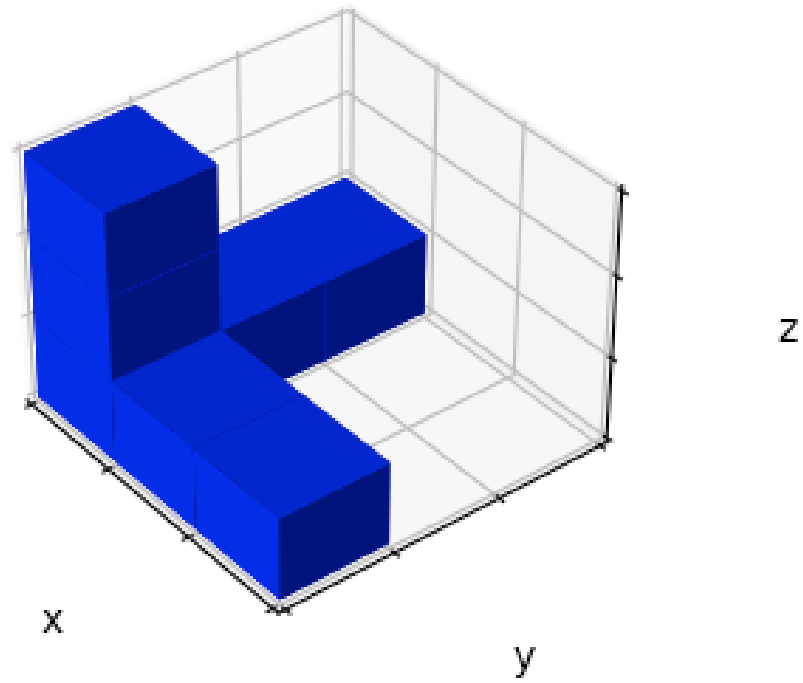


FIGURE 1 – Exemple 1

Pièce 3 (1 points)

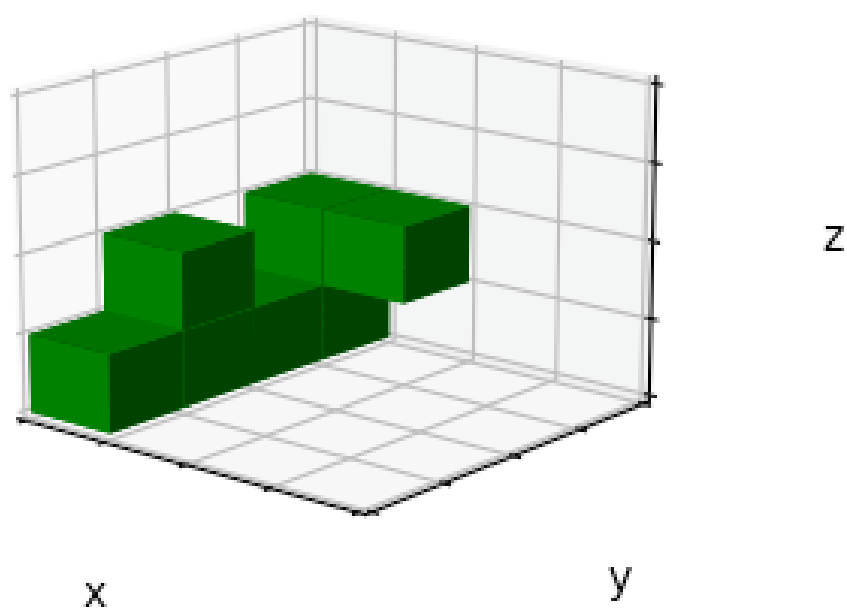


FIGURE 2 – Exemple 2

Pièce 2 (5 points)

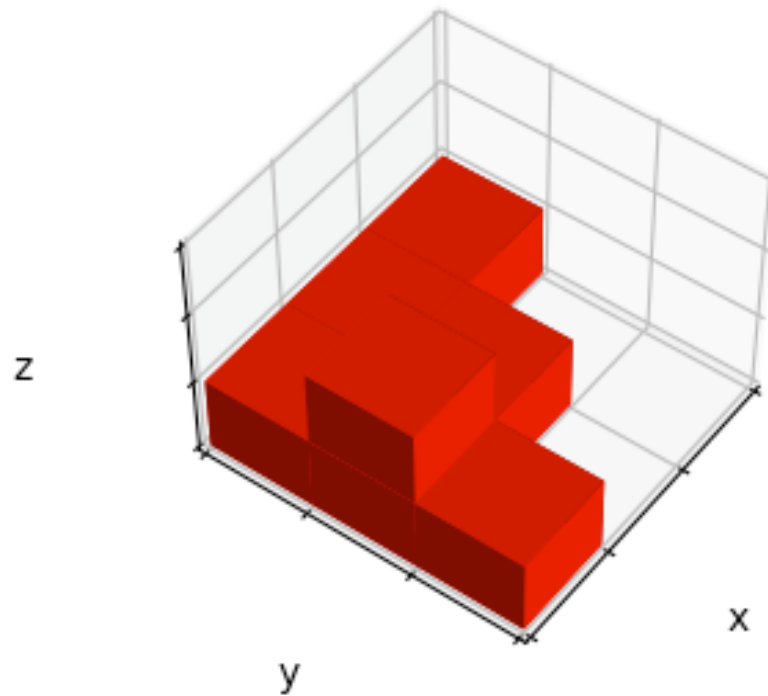


FIGURE 3 – Exemple 3

Le terrain est alors une grille en 3 dimensions constitués de pièces (on peut ici aussi faire tourner l'espace). On affiche le score en permanence sur l'interface :

Score : 9

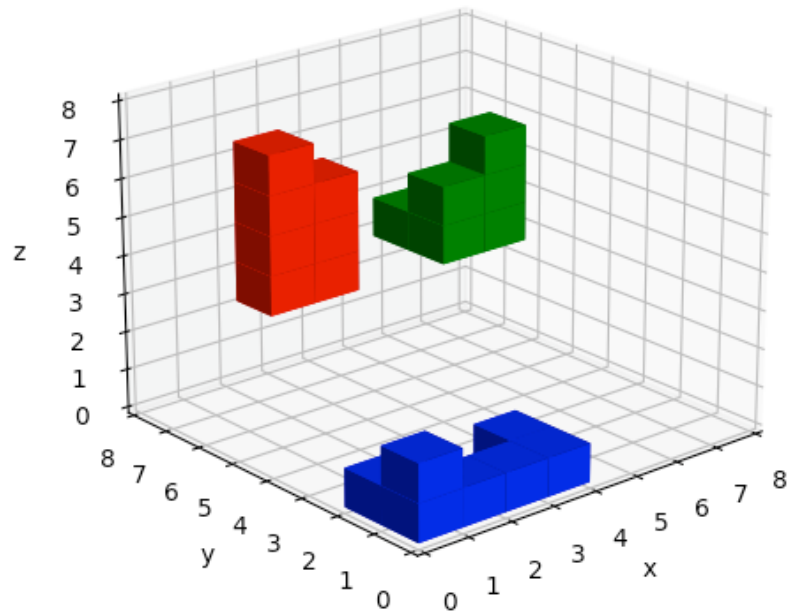


FIGURE 4 – Exemple 4

Ce qui donne l'interface entièrement fonctionnelle suivante :

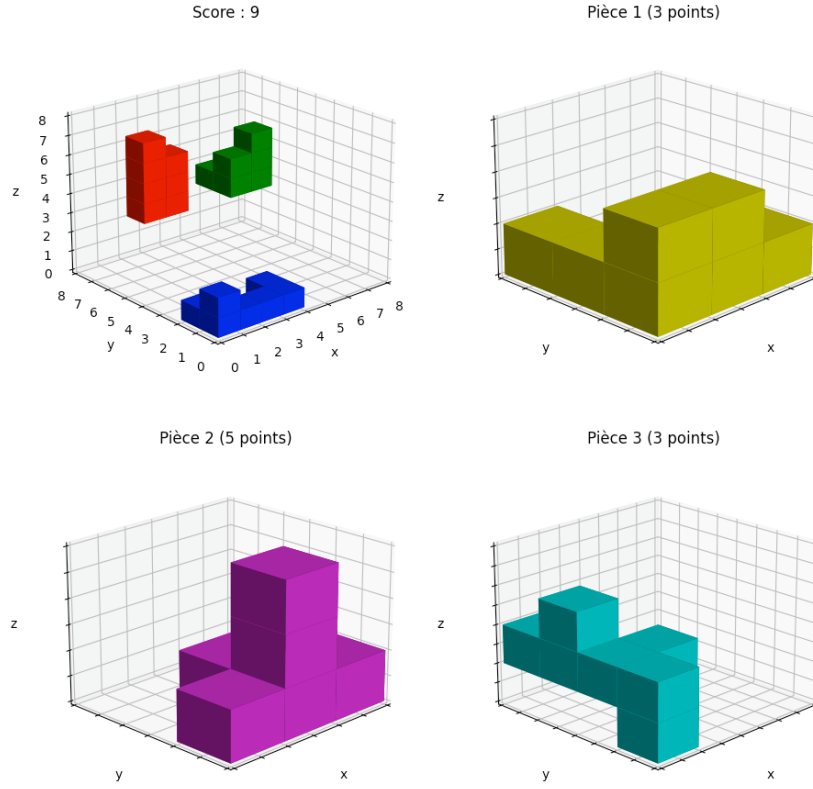


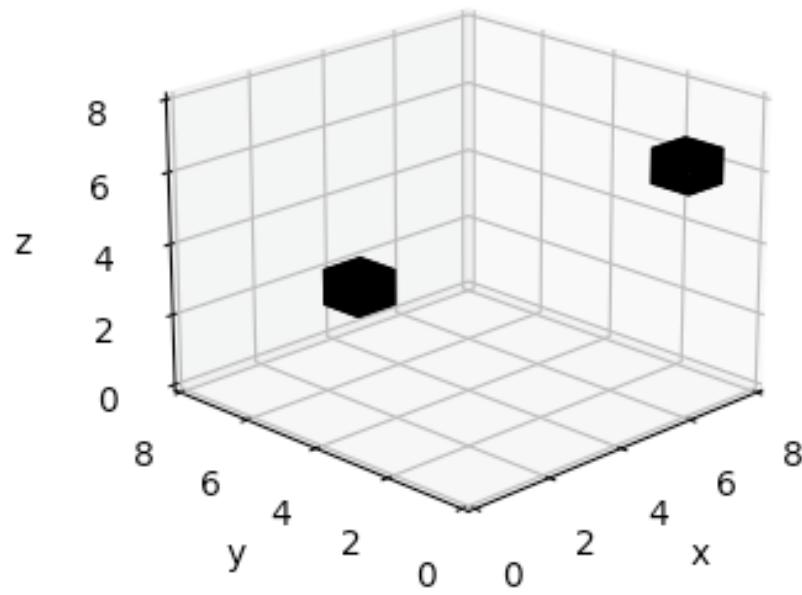
FIGURE 5 – Exemple 5

3 Fonctionnalités supplémentaires

3.1 Obstacles

Le premier ajout secondaire auquel j'ai pensé est l'ajout d'obstacles. Ces obstacles sont des pièces qui ne peuvent pas être déplacées et qui bloquent le joueur. Ils sont générés aléatoirement sur le terrain et le joueur doit les contourner pour continuer à jouer. On peut faire varier la difficulté du jeu en faisant varier la densité d'obstacles sur le terrain :

Score : 0



Pièce 2 (3 points)

FIGURE 6 – Densité de 1%

Score : 0

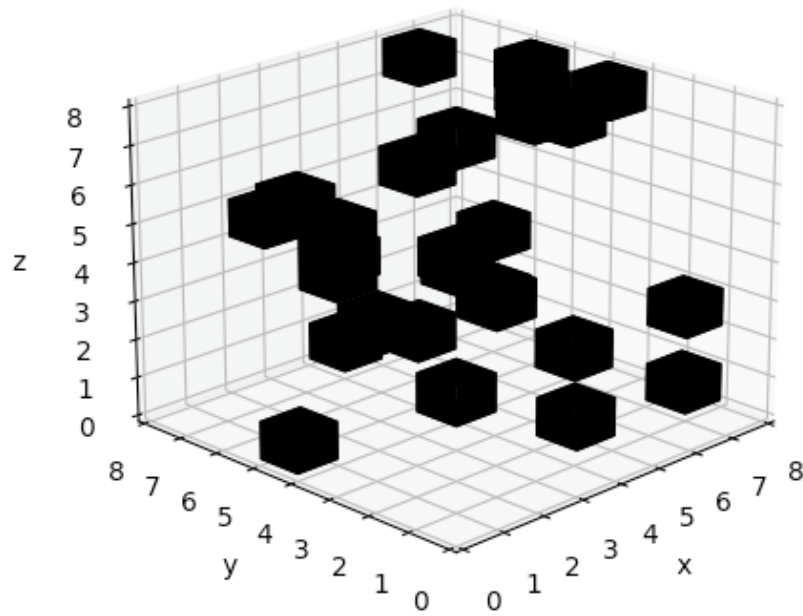


FIGURE 7 – Densité de 5%

3.2 Génération de pièces

Le deuxième ajout secondaire est la génération aléatoires des pièces. En effet, au moment de lancer une partie le joueur peut choisir une taille de pièce $n \in \mathbb{N}$ et le jeu génère alors des pièces de taille n aléatoirement. Cela permet de varier les parties et de rendre le jeu plus intéressant. Plus les pièces sont grandes, plus le jeu est difficile :

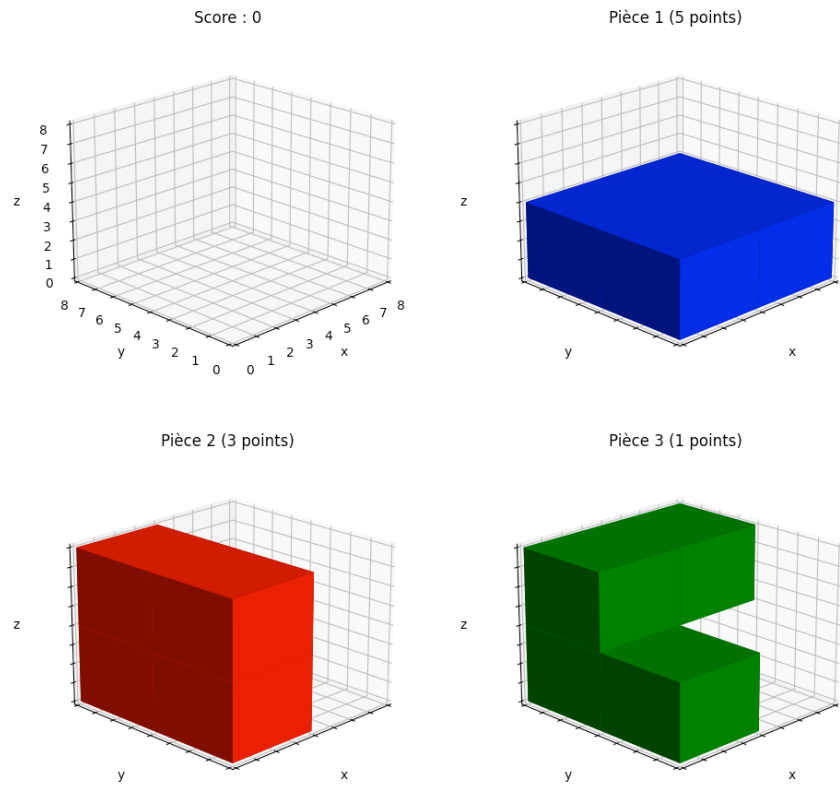


FIGURE 8 – Pièces de taille 4

On remarque que pour $n = 4$ le jeu revient à celui des tetrominos classiques mais en 3 dimensions.

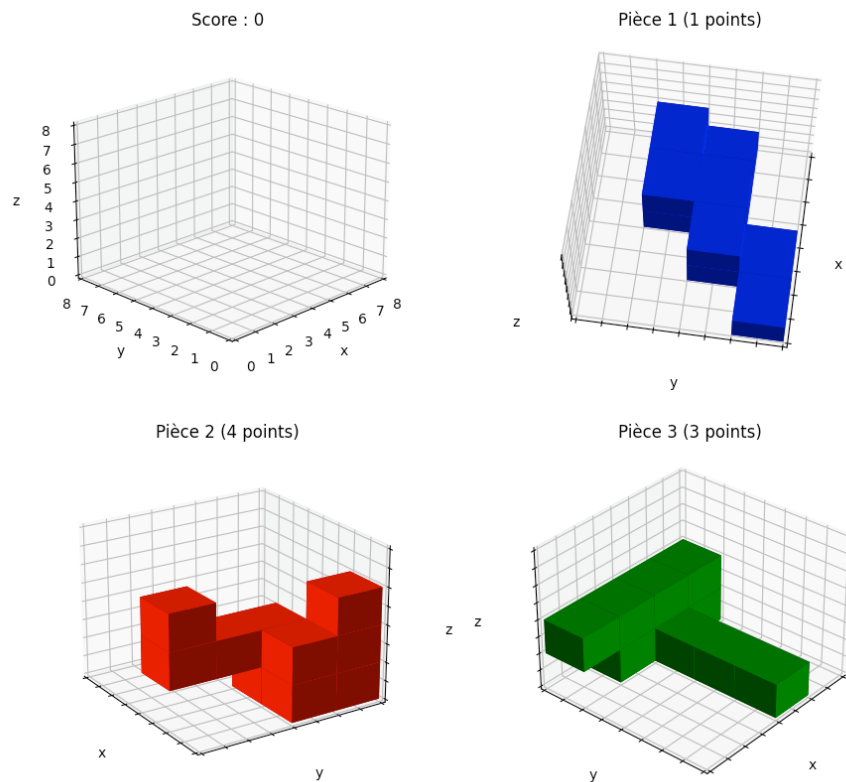


FIGURE 9 – Pièces de taille 10

J'ai aussi permis à l'utilisateur de choisir la gamme de score que peuvent prendre les pièces générées.

3.3 Taille du terrain

Comme pour le projet initial, la taille du terrain est variable, mais en 3 dimensions. On remarque d'ailleurs que en prenant une des dimensions égale à 1, on retrouve le jeu initial.

4 Utilisation du programme

Pour finir voilà comment paramétrer chacune des options qui ont été ajoutées :

```
# Paramètres du jeu

# - taille_x : taille du terrain en x
# - taille_y : taille du terrain en y
# - taille_z : taille du terrain en z
# - taille_pièce : taille des pièces
# - gamme_score : liste des scores possibles pour une pièce
# - proportion_obstacle : proportion d'obstacles sur le terrain

main(taille_x,taille_y,taille_z,taille_pièce,game_score,proportion_obstacle)
```

FIGURE 10 – Paramètres