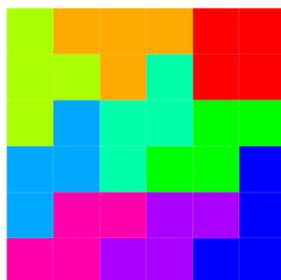


Projet Tétris



INTRODUCTION

Le célèbre jeu Tétris utilise des polygones particuliers formés de petits cubes. Ces polygones sont appelés *polyominoes* et peuvent être étudiés d'un point de vue mathématique. C'est le but de ce projet.

Mots clés : pavages, combinatoire, géométrie.

Le principe du projet est d'étudier une question mathématique complexe de façon autonome et originale en utilisant l'outil informatique. **L'ensemble du projet est à faire par groupe de 3 étudiants.** Le projet est divisé en 2 sections : théorique et pratique. La première partie est **obligatoire**, la seconde partie correspond à des **pistes de travail**.

Rendu du projet :

- Un **rapport de mi projet** répondant aux questions de la Section 1.
- Un **exposé** présentant votre travail et vos résultats sur la Section 2 en fin de projet.

Le Rapport. .

Date de rendu : 29 mars à minuit.

Que faut-il faire ? Répondre aux questions de la Section 1.

Quel format ? Format PDF obligatoire. Si vous souhaitez ajouter des dessins, vous pouvez les scanner et les ajouter à votre document.

Nom du fichier : tetris_nom1_nom2_nom3.pdf

Qui réalise le rapport ? Un rapport par groupe de 3 étudiants.

A qui le rendre ? Envoyez par mail à viviane.pons@u-psud.fr, objet du mail "Rapport Projet tetris" et noms des étudiants dans le corps du mail.

L'exposé. .

Quand ? Les exposés auront lieu en fin de semestre, la date sera précisée ultérieurement.

Que doit-on présenter ? Vous devez présenter le travail effectué sur la Section 2 du projet, en particulier, les résultats que vous avez obtenus, les algorithmes que vous avez utilisés, les images ou vidéos produites, etc.

En combien de temps ? Vous aurez 10 minutes de présentation, puis 5 minutes de questions.

Sur quel support ? Vous aurez un vidéo projecteur et un ordinateur à disposition (ou le votre si vous le souhaitez). Vous pourrez donc présenter votre exposé sous forme d'un powerpoint ou pdf. Vous pouvez aussi montrer des images, vidéos, démos de code.

Qui parle ? Tout le monde ! Les 3 étudiants doivent participer.

Doit-on présenter notre code ? Vous pouvez utiliser un notebook SageMathCloud pour présenter des démos de code, cependant, nous ne notons pas la programmation mais bien les résultats obtenus !

Doit-on répondre aux questions ? Les questions de la Section 2 ne sont pas obligatoires, ce sont des pistes de travail, vous pouvez les suivre, ou pas...

1. ÉTUDE THÉORIQUE DU PROBLÈME

1.1. **Définition.** Un *polyomino* est une figure plane composée de carrés unitaires ayant au moins un côté commun. La *taille* d'un polyomino est donnée par son nombre de carrés, voir les exemples Figure 1.

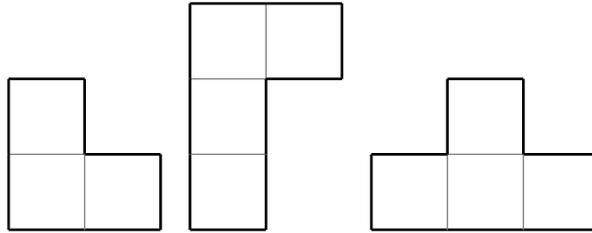


FIGURE 1. Trois polyominos de tailles respectives 3, 4 et 4

Il y a plusieurs conventions pour décider si deux polyominos sont égaux. Nous travaillons avec la convention la plus simple celle des *polyominos à forme fixée*. Dans ce cas, deux polyominos sont considérés égaux s'ils ont la même forme et tels que les rotations et retournements ne sont pas permis. La Figure 2 donne un exemple.

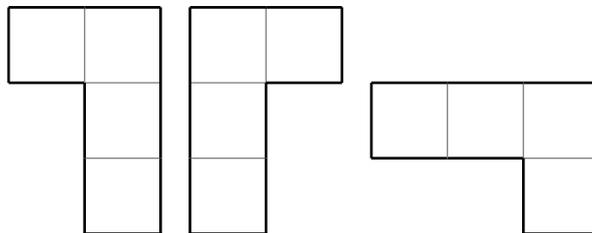


FIGURE 2. Pour la convention à *forme fixée*, les trois polyominos ci-dessus sont considérés comme différents.

La *classe de rotation* d'un polyomino P est l'ensemble des polyominos que l'on obtient par rotations successives de 45 degrés de P . Sa *classe de symétrie* est l'ensemble des polyominos que l'on obtient par symétries successives de P . Enfin, on appellera l'ensemble des polyominos obtenus par rotations ou symétries successive de P la *classe complète* de P . Des exemples sont donnés en Figures 3, 4 et 5.

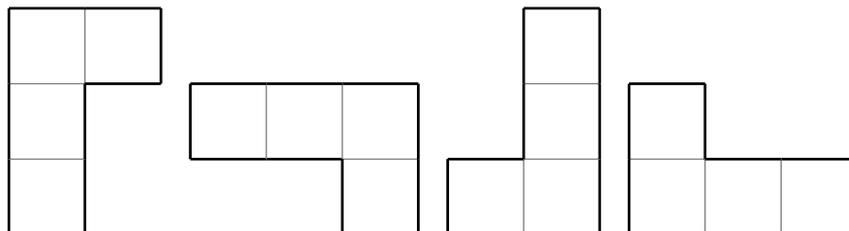


FIGURE 3. Classe de rotation d'un polyomino

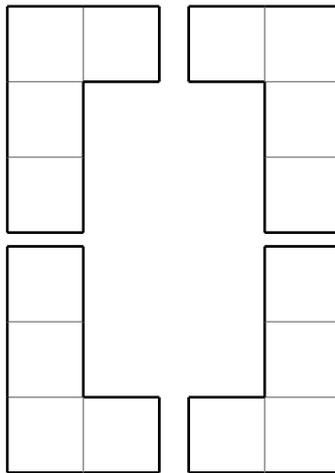


FIGURE 4. Classe de symétrie d'un polyomino

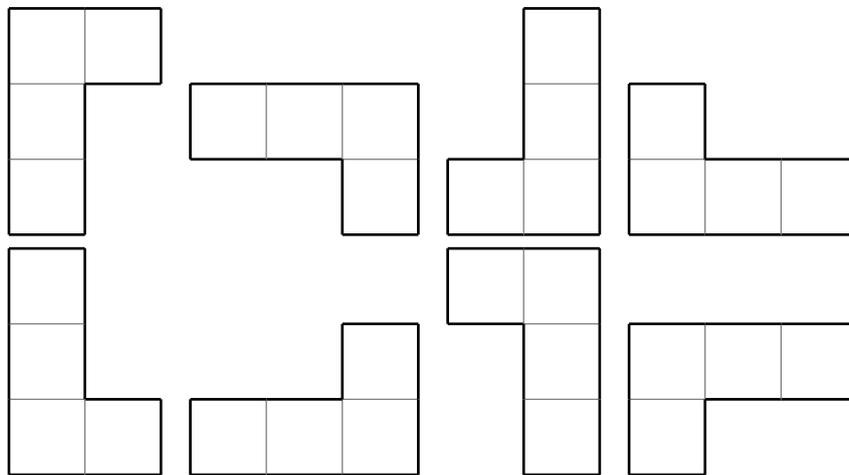
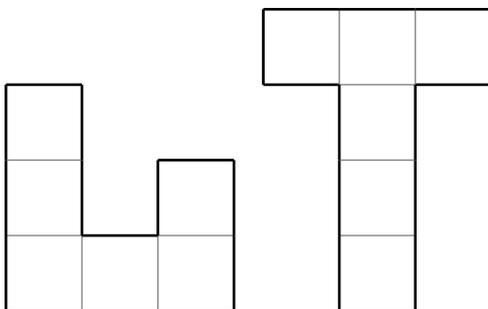


FIGURE 5. Classe complète d'un polyomino

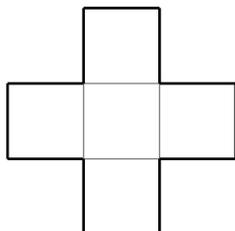
1.2. Questions.

- (1) Donnez les classes de rotation, de symétrie et classes complètes des deux polyominos suivants.



- (2) Combien d'éléments une classe de rotation peut-elle contenir au maximum ? Justifiez la réponse et donnez un exemple où le maximum n'est pas atteint.
- (3) Même question pour une classe complète.
- (4) Donnez tous les polyominos de tailles 4 en distinguant les différentes classes complètes et classes de rotation. Vous devez trouver 19 polyominos, 7 classes de rotations et 5 classes complètes.

- (5) Donnez les 12 classes complètes de polyominos de taille 5 (pour chaque classe, vous pouvez ne dessiner qu'un seul polyomino).
- (6) La question la plus classique sur les polyominos est celle du *pavage* : recouvrir complètement une zone du plan (ou le plan entier) par des polyominos. Un exemple de pavage est donné en Figure 6. Donnez un autre pavage du carré 6×6 par des polyominos de taille 4.
- (7) Prouvez qu'il n'est pas possible de paver un rectangle avec le polyomino suivant. Cependant, expliquez comment vous pourriez paver un plan infini.



- (8) Prouvez qu'il est possible de paver n'importe quel rectangle $2n \times 4m$ avec les polyominos de la classe représentée en Figure 5.

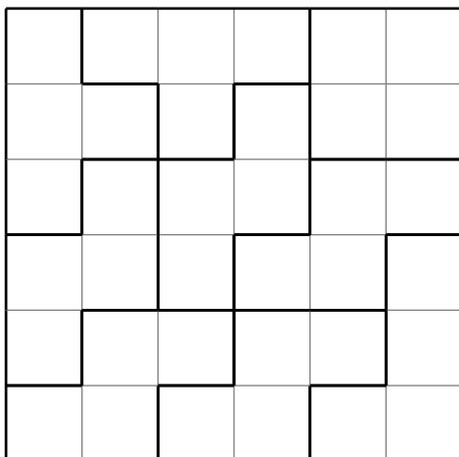


FIGURE 6. Pavage d'un carré 6×6 par des polyominos de taille 4

2. MODÉLISATION ET EXPLORATION

Le but de cette partie est d'explorer les questions relatives aux polyominos avec l'aide de l'outil informatique. Nous vous proposons ici des pistes, à vous de les explorer ou d'en proposer de nouvelles.

2.1. Modélisation et visualisation. Le but de cette section est de représenter et traiter les polyominos en tant qu'objet informatique.

- (1) Proposez une ou des solutions pour représenter les polyominos en tant qu'objet informatique.
- (2) Comment obtenir les classes de rotation et de symétrie d'un polyomino donné ?
- (3) Comment tester que deux polyominos appartiennent à la même classe complète ?
- (4) Comment représenter le pavage d'une zone donnée ?
- (5) Proposez des fonctions d'affichage (avec `plot`) pour représenter les polyominos et les pavages.

2.2. Pavages. Le but de cette section est d'explorer automatiquement les pavages de rectangles par un ensemble de polyominos donnés.

- (1) Proposez un algorithme permettant de tester si un rectangle donné peut être pavé par un ensemble de polyominos donnés.
- (2) Testez votre algorithme sur différents rectangles et ensemble de polyominos. Le but est comprendre quels rectangle peuvent être pavés par un ensemble donné. Exemple d'ensembles : tous les polyominos de taille 4, une classe complète pour un polyomino donné etc.
- (3) Essayez de prouver certains des résultats que vous obtenez de façon expérimentale.
- (4) Adaptez votre algorithme à des formes différentes (pas des rectangles).
- (5) Comment faire pour rechercher des pavages du plan entier ?

2.3. Génération des polyominos. Comment engendrer l'ensemble des polyominos ?

- (1) Commencez par générer les polyominos à forme fixée. Vous pouvez soit définir votre propre algorithme, soit implantez des algorithmes existants décrits en particulier sur wikipedia.
- (2) Comment adapter cet algorithme pour la génération des classes rotations ou classes complètes ?

2.4. Spécial Tétris. On se place dans la situation suivante : on a un rectangle et un ensemble de polyominos donnés. A chaque étape, on reçoit un polyomino au hasard qui ne peut être placé "qu'en bas" du rectangle. Si une ligne complète est formée, elle disparaît et toutes les autres lignes descendent d'un cran. Le but n'est pas de jouer mais de chercher un algorithme optimal !

- (1) On part du principe que le polyomino envoyé est à forme fixe (on effectue pas de rotation), proposez un algorithme pour le placer de façon optimale et effectuez des tests sur divers rectangles et ensemble de polyominos.
- (2) Même question avec rotation autorisée.
- (3) Même question avec symétrie et rotation.