

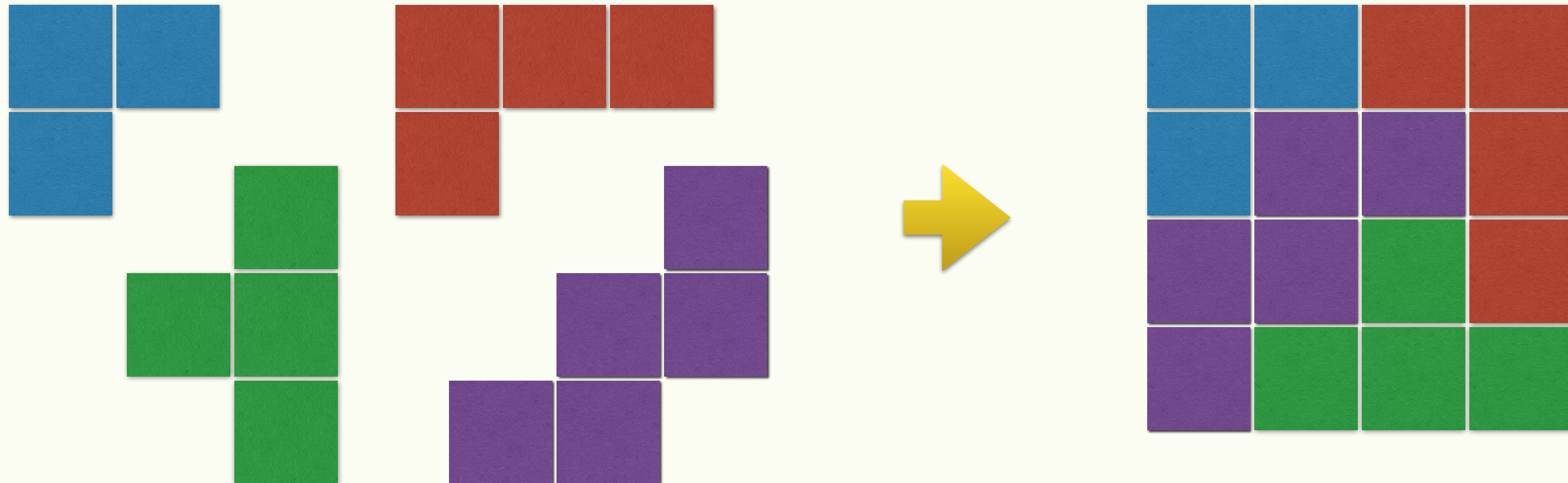
Polyomino Solver

Fixstars Amplify ハッカソン 説明スライド

GitHub: [kaz130](#)

ポリオミノパズル（箱詰めパズル）

- 複数のポリオミノを決められた形に隙間なく並べるパズル
- ポリオミノ：1つ以上の正方形を組み合わせで作られる多角形



開発背景

最適化問題としてポリオミノパズルはちょうど良さそう

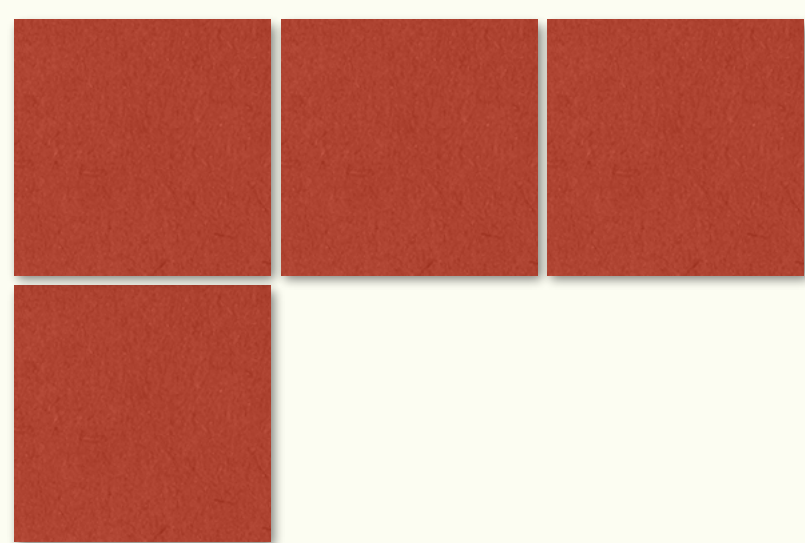
- ルールが非常にシンプルであり、問題として分かりやすい
- ピース数を n として、愚直に解くと計算量は $O(n!)$ 以上に
- 詰め方のルールを制約条件に落とし込むことで、**組合せ最適化**により効率的に解けそう

実装

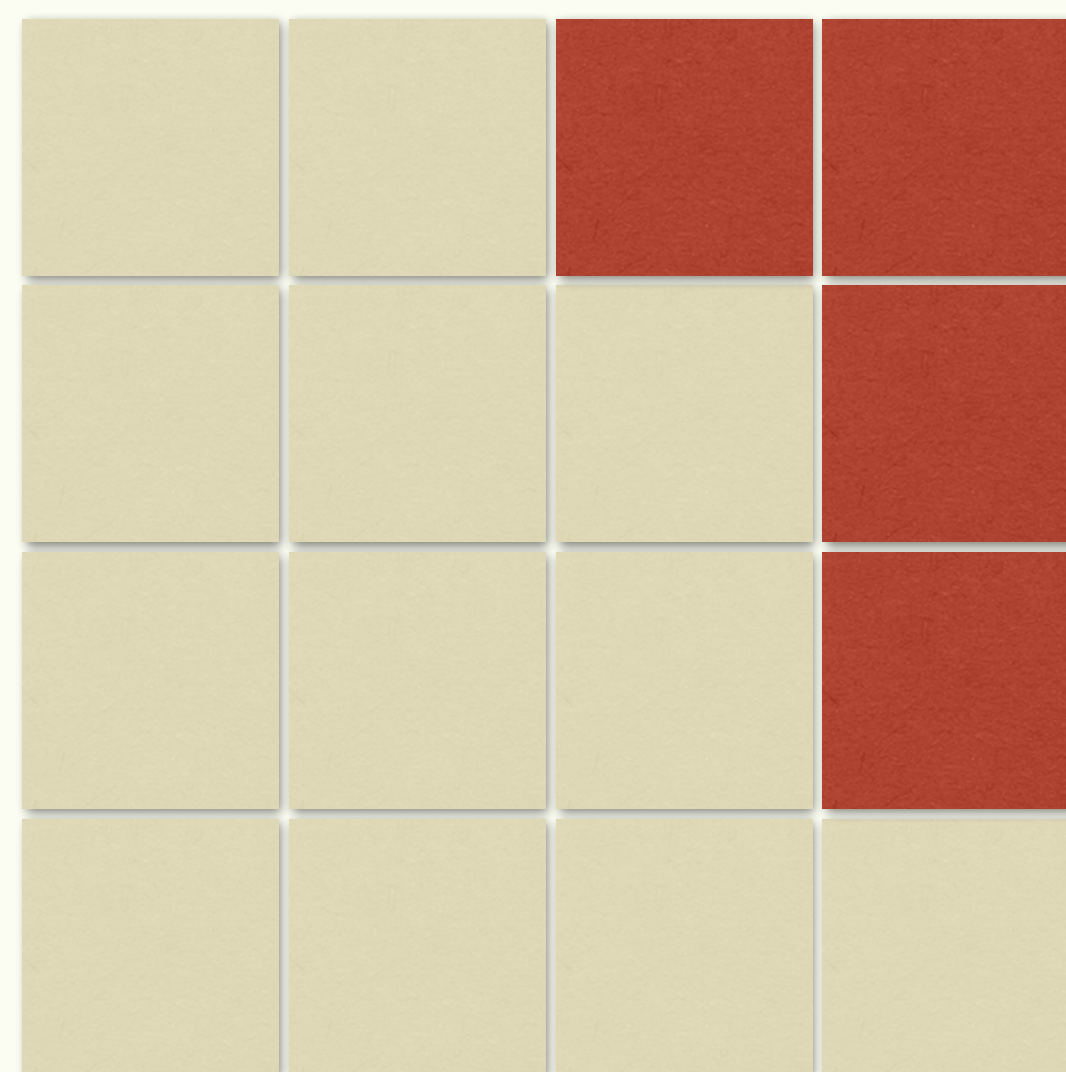
ピースの種類、位置、置き方をバイナリ変数で表現する

- $q_{y,x,i,p} = 1 \rightarrow (x, y)$ の位置に i 番目のピースを置き方 p で置く

$i = 2$



$q_{1,3,2,2} = 1$



$p = 1$: そのまま

$p = 2$: 90°回転

$p = 3$: 180°回転

⋮

$p = 8$: 裏返して270°回転

実装

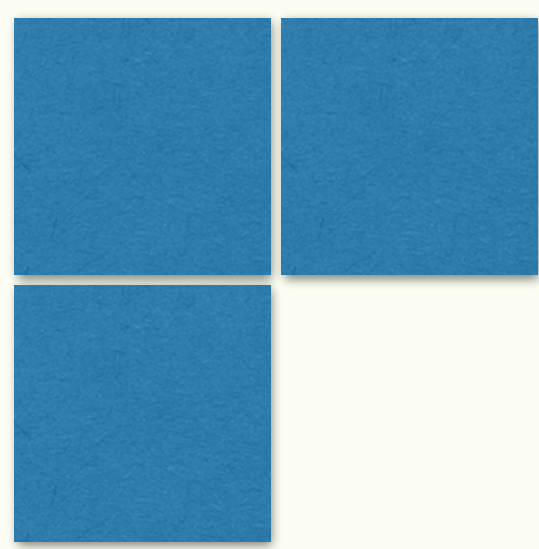
パズルを解くための制約条件を考える

A. 重複する置き方を除外する

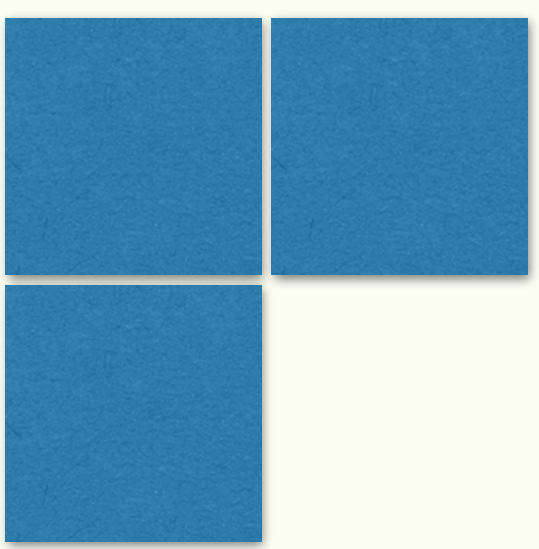
B. 枠から外に出る置き方を除外

ex) $i = 1$ のピース

$p = 1$



$p = 5$

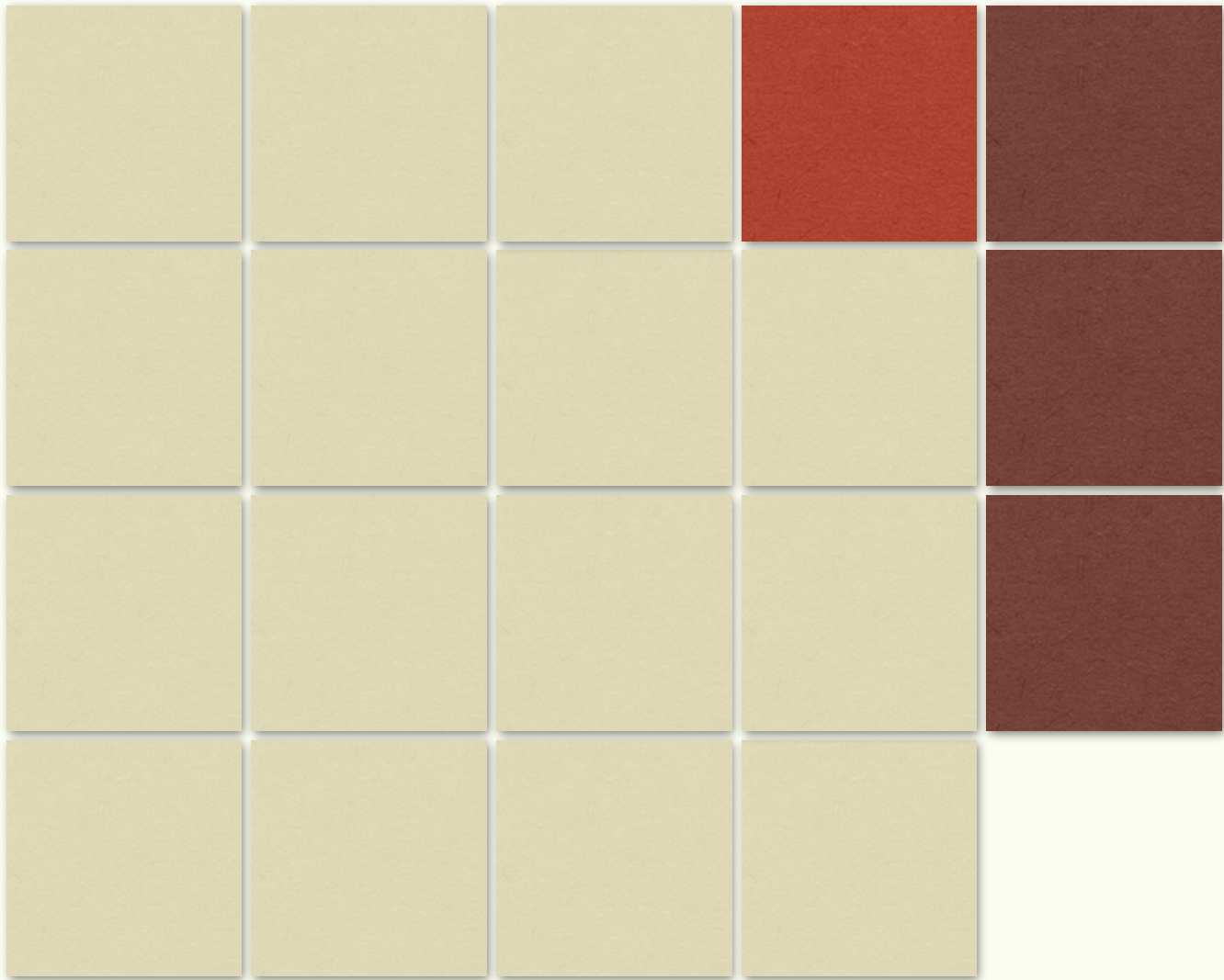


同じ形になる
置き方を除外

$$q_{y,x,1,5} = 0$$

ex)

$$q_{1,4,2,2} = 0$$



実装

パズルを解くための制約条件を考える

C. ピース同士が重なる置き方を除外

$$\sum_{i,p \in (x,y) \text{ と重なる}} q_{y,x,i,p} = 1$$

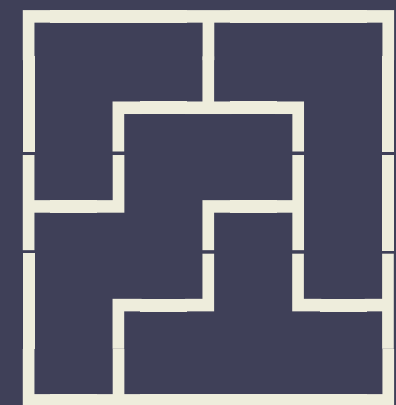
D. 全てのピースが一度ずつ使われる

$$\sum_y \sum_x \sum_p q_{y,x,i,p} = 1$$

実行

- 入力データとして、ピースとボードを文字列で設定
- 回転・反転の有無も設定可能

```
$ python -m psolver sample1.toml
```



sample1.toml

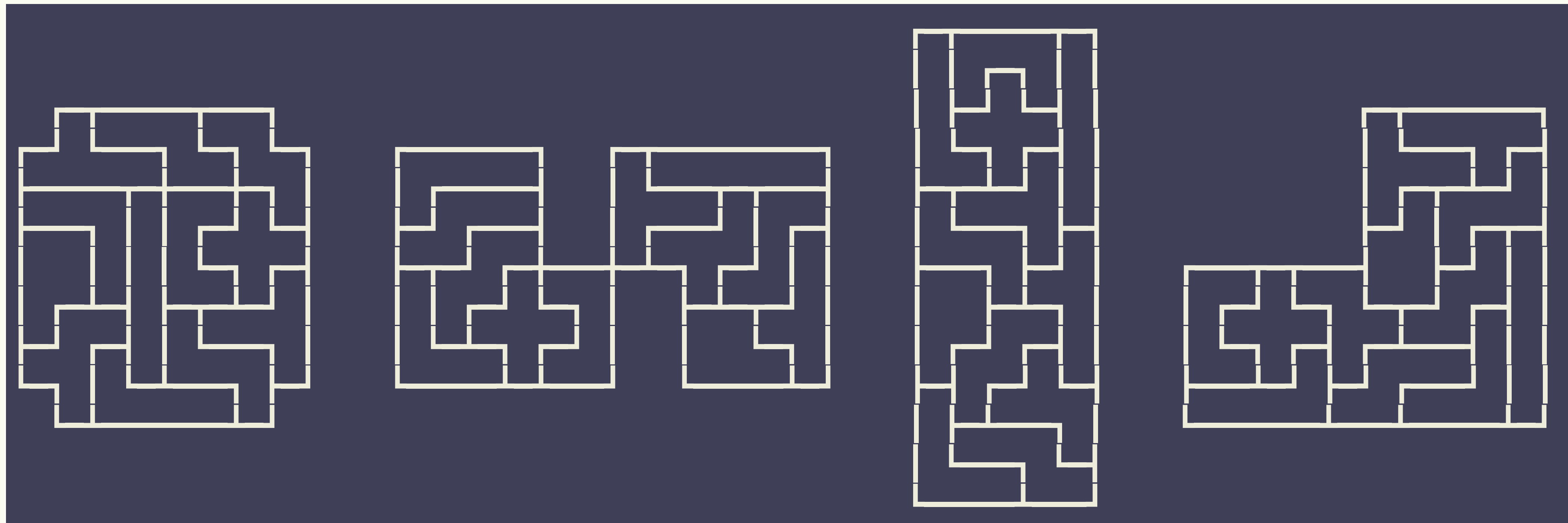
```
pieces = '''  
##  
#  
  
###  
#  
  
#  
##  
#  
  
#  
##  
##  
##  
'''
```

```
board = '''  
####  
####  
####  
####  
'''  
  
can_rotate = true  
can_reverse = true
```

実行

様々な入力パターンに対応

- 入力に5ブロックのポリオミノ（ペントミノ）を使用



おわりに

まとめ

- Fixstars Amplify を用いて、ポリオミノパズルを解いた
- テキストファイルを入力データとし、様々なパターンで結果を確認

今後の課題

- GUIアプリケーションの開発
- 3次元箱詰めパズルへの拡張