ペントミノ

1　取り組み

- ペントミノ用ヘッダファイル
- 解の表示
- 実行時間

2　ペントミノ用ヘッダファイル

盤面（Board）とペントミノ（Pentominos）を表現したヘッダファイルを作成した．実行には，Eigen という線形代数ライブラリが必要である．詳しい説明は，ヘッダファイル内に記述している．簡単な説明は，以下の通りである．（引数は省略している．）

2.1　board.h

1. void setColor()　　盤面に色を塗る．
2. bool check()　　盤面に色が塗れるか判定する．
3. bool next()　　マス目の移動と移動ができるか判定する．
4. void print()　　盤面の表示する．

2.2　pentomino.h

1. void setSP()　　特殊なピースを区別する変数を初期化する．
2. void setCoords()　　ピースの座標（相対座標）をセットする．
3. void setPattern()　　ピースのパターン数をセットする．
4. void setColor()　　ピースの色をセットする．
5. vector pop()　　ピースの相対座標を返す．（引数によってパターン（回転，対称移動）を変えてくれる）

2.3　サンプル

サンプルプログラムとして sample.cpp を用意した．ピースを表示するプログラムとなっている．

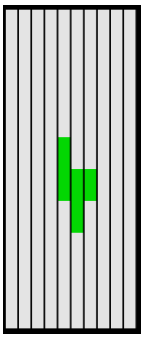
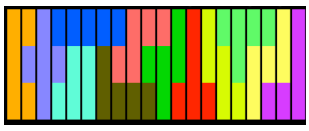
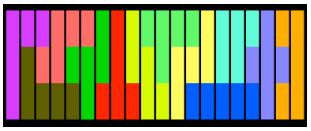


図 1　表示したピースの 1 つ

3　解の表示

解の表示は，ANSI Escape を使用した．





全ての解を表示は数が多いため参考として盤面のサイズが 3×20 の場合を載せた．

4　実行時間

各盤面に対して全ての解を求めるまでの実行時間は，以下の通りである．

1. 3×20　265ms
2. 4×15　4546ms
3. 5×12　19624ms
4. 6×10　59299ms

実行環境は，以下の通り．
プロセッサ：2.3 GHz デュアルコア Intel Core i5
メモリ：8 GB 2133 MHz LPDDR3
グラフィクス：Intel Iris Plus Graphics 640 1536 MB

5　まとめ

今回は，ペントミノに関数するヘッダファイルを作成した．解を求める際は，再起関数を用いて問題を解いた．また実行する際は，枝刈りしやすいように盤面の高さが横幅より短くしている．