

タイトル：限界集落

学校名：豊田工業高等専門学校

1. システム概要

与えられた盤面から最終盤面へとそろえる操作を探すソルバープログラムと、盤面の表示や盤面情報の取得と操作の送信を行うビジュアライザを作成した。

2. 解法

盤面や寄せの操作で用いるデータ構造と、最終盤面へとそろえるアルゴリズムを説明する。

2.1 データ構造

盤面の各ピースは高々4色であるため、2bitのデータに収めることができる。そのため64bit 整数に最大 32 個の盤面情報を入れたデータ構造にすることにより、メモリの削減を行った。これにより、各ピースを愚直に持つものに比べて 16 倍ほど高速化することができた。

盤面情報は列ごとに上記で述べたデータ構造を保持することで管理した。上下寄せを多用する解法を使うため、上下寄せのシミュレーションにおいてビットシフトを使用し高速化を行った。

2.2 アルゴリズム

定型抜き型のみを用いて、現在の盤面の最下行に、最終盤面の行と同じピースを集めていくことを繰り返し、最終盤面へと持っていく。以下では、なるべく少ない手数で最下行にピースを集める方法を説明する。

最下行に必要なピースがある程度固まって盤面に存在していた場合、それらをまとめて最下行へ移す。その際、盤面の中で、最下行に必要なピース群のうち最も効率が良いものを探索する。具体的には、ピース群のうち、その長さを最下行へ移動させるのに必要な手数で割った値が最も大きいものを選ぶ。基本、最下の左側へピースを集めていく。必要なピース群は、(最下行に集めたいピース列)\$(現在の盤面の1行目)%(2行目)%... というような文字列を作

り、Z-Algorithm 用いることで高速に見つける。

また、左右寄せ操作を1回行うとき、行の特定の区間が入れ替わる特徴と Z-Algorithm を利用して、必要なピースが長く連結するようなものを探し、効率的に最下行へ移す。

そして、現在の盤面から最下行に寄せられるピース群の効率的な操作の候補をいくつか見つける。それを始点として、最下行に集まったピース数とそれにかかった手数を評価値として最下行をそろえていくビームサーチを行い、より効率的に操作するものを見つける。

盤面を反転・回転させた状態を入れた8つの盤面についてこの解法を使い、操作回数が最小なものを最終的な回答として採用する。

3. ビジュアライザ

盤面の描画や対戦に必要な処理を行うプログラムについて説明する。

3.1 生成プログラムとの分離

回答データを生成する実行ファイルとビジュアライザは分離させ、それぞれ非同期で実行させる。ビジュアライザが実行ファイルに盤面の情報を入力し、操作情報の出力を逐一受け取るようにした。また、描画する盤面もその度に更新して、操作手順を可視化した。

3.2 データの送受信

データの送受信もビジュアライザで行う。ボタンを押すとサーバーにリクエストを送り、必要なデータのみを取り出す。

3.3 盤面の描画

OpenSiv3D には図形の描画がサポートされている。しかし、ピース一つ一つを図形で描画すると、256×256 個の処理が必要となり、非常に負荷がかかってしまう。そこで、ピース1つの情報を 1px の色で表現し、画像として描画している。