5601

マッチ 3 ゲームの総当たりの解探索の評価

Evaluation of brute force search in three match games

CS01　雨宮隆行
指導教員　大島真樹

1. 研究目的

本研究ではマッチ 3 ゲームの解探索のプログラムを作成し，実行した結果スコアと秒数がどのように関係してくるか明らかにする．

2. マッチ 3 ゲームのシステム

今回使うマッチ 3 ゲームは 6×5 のマスに 6 種類のドロップを配置する．ドロップを動かし縦か横に 3 つ以上同じドロップを並べることでドロップを消すことができ，消すことで得点が加算される．消した場所にはまた新しいドロップが追加される．この得点をかせいでいくゲームである．また従来のマッチ 3 ゲームと違い，1 つのドロップを選択して隣接する場所に連続してドロップを動かすことができる．

3. 開発環境

作成言語：Java 言語 version7
作成環境：Eclipse ver4.2
OS:Microsoft Windows 7 Professional
CPU:Intel(R)Core(TM)i3-220 CPU @3.30GHz
メモリ:2.00GB

4. 研究方法

このマッチ 3 ゲームを作成し，総当たりの解探索(実験 1)，総当たりに経路の制限を加えた解探索(実験 2)の 2 種類の実験を行う．

結果を固定させるためにドロップの配置を行っている乱数の seed 値を固定して実行した．評価するための得点の計算式を下記に示す．

S[0]＝(1000+(n-3)*250)
SCORE=S*{1+((combo-1)*0.25)}

図 1. スコアの算出方法

※S=その列のスコア，n=ドロップの列の長さ，SCORE＝最終的な得点，combo=何列ドロップを消せたか

―実験 1 のアルゴリズム―

ドラッグ処理をして，それが終了したらフィールドのドロップを確認する．縦か横に同じ種類のドロップが 3 つ以上並んでいたらそれを消滅させ，空いた場所にドロップを落とす．足りないドロップは上から落として追加する．

―実験 2 のアルゴリズム―

先ほどのプログラムに「1121」や「1211」という並びがあった時に 11 と 2 つ並んでいる場所を移動不可能にし，2 の部分は 1 のほうに向かい，同じドロップを 3 つ並ばせるパターンを追加した．これを i パターンと呼ぶ．

4. 研究結果

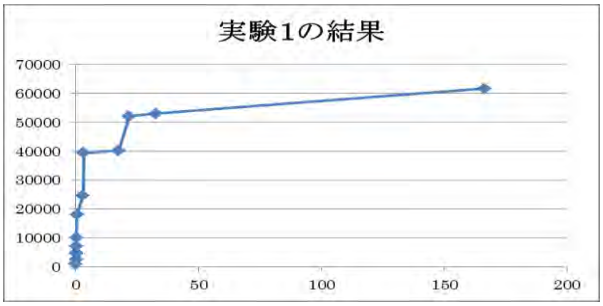

図 2. 実験 1 の結果

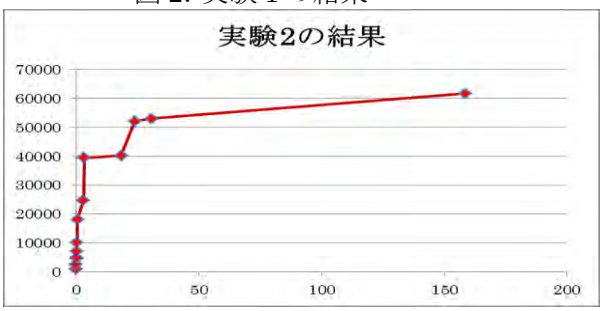

図 3. 実験 2 の結果

どちらも，ハイスコアが更新されたらスコアを表示するようにしている．

5. 考察

それぞれ同じ乱数の系列を使用しているため，スコアの表れ方は一緒だが，出力する秒数の速さが違う．今回の結果では経路の制限を追加したプログラムが最終的な秒数を 10 秒近く改善した．

実験 1 では実行した 20 秒当たりまでは順調にハイスコアが更新されたが，それ以降はあまり更新されなかった．これは意味のない経路まで大量に探索したせいでハイスコア更新に時間がかかったためである．

実験 2 では最終的な秒数は更新したものの，中間地点では総当たりのほうが速い場合もあった．これはルート制限を加えることで経路は少なくできるが，制限のせいでハイスコアが狙えるルートが閉じしまったことが原因と思われる．

6. 今後の課題

別のパターンを追加してみて探索時間の減少に努める．しかしただ追加するだけではハイスコアのルートをつぶしたり．移動不可能区間に挟まれて移動できなくなるエラーが発生してしまうため．3 つ同じドロップを並べたら移動させるなどエラーを避ける工夫が必要となる．

7. 参考文献

[1]河西朝雄,"Java による初めてのアルゴリズム入門"，技術評論社(2001)