

HAARU



乾 謙一

ハムレット——と言っても、シェイクスピアとは、何の関係もありません。今月、TBNのお送りするのは、おなじみ乾謙一さんから送られて来ました、NAKAMOZU TINY BASICによる、2次元4目並べ(略して、2D-4M)ゲームの名前です。2次元は2次元でも、このゲームで用いる平面は中と高さからなる、壁のような直立した平面です。従って、石は、盤の下から順に積み重ねて行かねばなりません。コンピュータ相手にひとりでもやるもよし、コンピュータにはジャッジをお願いして友達と2人でやるもよし。但し、コンピュータ同志で対戦をやっているのをひとりで見ていると、ホレイシオのような厭世的な気分になって来てしまいます。それで、ハムレットなどという、大時代的な名前が付いたのでは……?

このプログラムは、互換性を十分に考慮して書かれていますので、5月号のNAKAMOZU TINY BASICのマニュアルを参照して戴ければ、フローチャート等を参考にして全てのシステムで走らせることが可能です。NTBのシステムをお持ちでない方も、ぜひこの機会に移植のテクニックをマスターして下さい。

1. ゲームのルール

このゲームは、6×6の盤で行なう4目並べです。但し、石は任意の位置に打てる訳ではなく、必ず下から順番に積み上げてゆかねばなりません。連珠等とは違って、5個以上並んだり、同時に2方向以上に並んだ場合でも勝ちとし、禁じ手は特にありません。もちろん、パスはできません。36個の石を全て打ち終わっても勝負がつかなかった場合には、ここでは、後手勝ちとしています。

2. ゲームの遊び方

このプログラムは、NAKAMOZU TINY BASICだけで書いてあり、マシン語のサブルーチン是用いていません。これは、BASICで書いた方が、他方面への応用が容易であろうと考えたためです。

また、画面表示は、カーソル移動とプリント文のみで行なっていますので、システムへの依存性はありません。必要とするワーキングエリアは、\$2000からの100バイト程度で十分です。

プログラムを走らせると、まず、「何手読む?」と聞いてきますので、3以上の数を入力

して下さい。これによって、コンピュータの強さが変わりますが、インタープリタでは時間的に3~4が実用的でしょう。

続いて、「何人でやるか?」と聞いてきますので、0~2を入力して下さい。0はコンピュータ同志の対戦、1は人間対コンピュータの対戦、2は人間同志の対戦です。1を入力した場合には更に、人間側が先手を取るか聞いて来ますので、YかNを入力すればゲーム開始です。

石を落とす位置は、1~6で指定するだけです。キーを押すと、画面上部の1~6の数字の下から石の落ちてゆく様子が、アニメーション表示されます。

3. MIN-MAX法

コンピュータの思考部分には、MIN-MAXと呼ばれる方法を用いています。探索の原理を図1に示します。私は、以前にこの方法を用いてオセロのプログラムを製作して、好結果を得ています。ここでは、同時に必勝手順の探索も可能としています。

評価関数は、盤全体から算出するようなものが望ましいのですが、ここでは簡単のために、先読みの処理で最後に置かれた石の周囲から評価値を計算しています。

評価値は1バイト、即ち0~255の値をとりませんが、自分の石が4つ以上並んだ場合は200、相手の石が並んだら1となって、その枝の探索を打ち切って別の枝へと進みます。

コンピュータ・ルーチンのフローチャートを図2に示します。図中で、Sは読みの深さを示し、iは現在処理している深さ、またE(i)は評価値バッファを示しています。スタックとバッファの構成を図3に示します。

この方法では、やはり木探索を用いているために、読みの深さを増すと、処理時間が指数関数的に増大してしまいます。しかし、評価関数が単純な割には強くて、5手読み位になると、簡単には勝てないようです。

ぜひ一度、試して下さい。

なお、参考のために、表1と表2にプログラム説明と変数の割り当ての一覧表を掲げます。

最後に、ハードコピーを提供して下さい。山下春生氏に感謝致します。

参考文献

Byte October 1978
"A Computer Chess Tutorial"

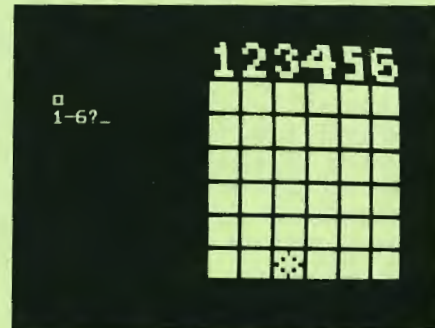


写真1: コンピュータ「3」でゲーム開始。

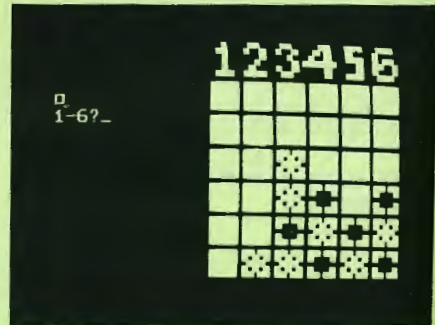


写真2: 折角の斜め3を止められたところ。

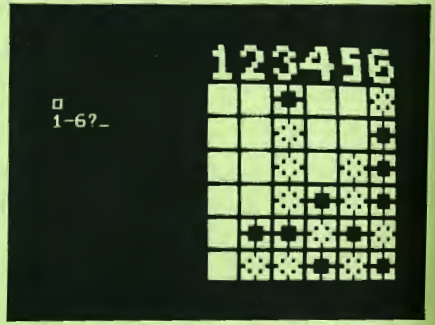


写真3: 中盤過ぎ、コンピュータやや優勢。

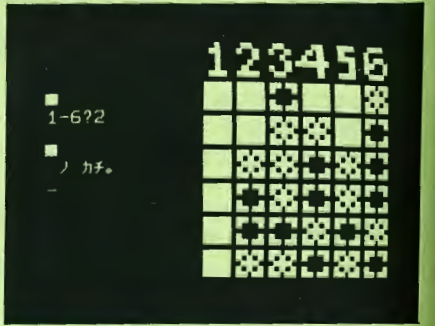


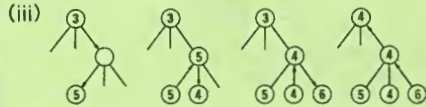
写真4: くっく! 斜め4で人間様の負け。



4, 3, 7の最小値をとる。



ここで、 $3 > 2$ であるから、他の枝の評価が2以上でも、2未満でも、この枝が選ばれる可能性はなく、 $\alpha - \beta$ 刈りが行なわれる。



5, 4, 6の最小値4と以前の評価値3との最大値4をとる。

木探索の完成図

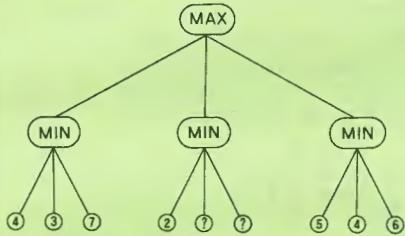


図1: MIN-MAX評価と $\alpha - \beta$ 枝刈り

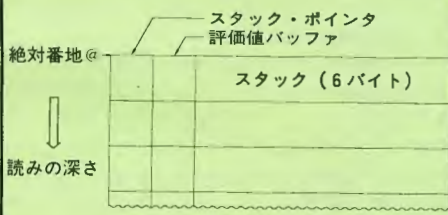


図3: スタックとバッファの構成

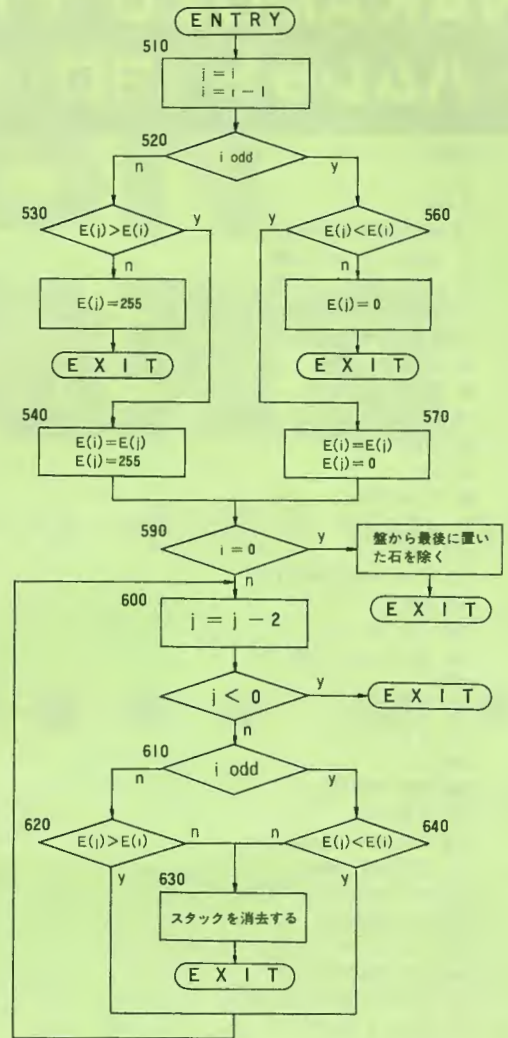
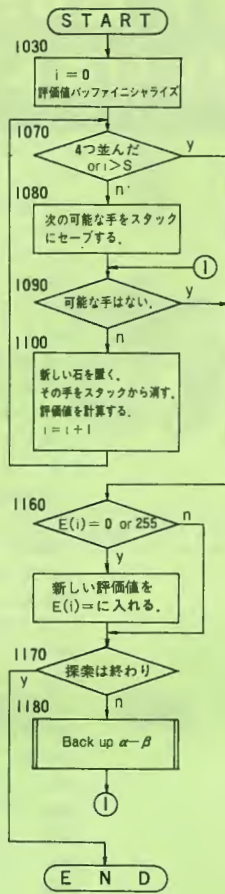


図2: フローチャート

60 ~ 180	LIST	可能な次の手の座標をスタックにセーブする。
200 ~ 390	VALUE	盤面から評価値を計算する。
500 ~ 650	BACK UP	バックアップ および $\alpha - \beta$ 枝刈り
1000 ~ 1200	MIN-MAX	コンピュータの思考ルーチン
2000 ~ 2270	MAIN	メインプログラム
3000 ~ 3070	DISPLAY	ディスプレイ
3200 ~ 3280	DISPLAY-SUB	ディスプレイ

可能な次の手の座標をスタックにセーブする。
盤面から評価値を計算する。
バックアップ および $\alpha - \beta$ 枝刈り
コンピュータの思考ルーチン
メインプログラム
ディスプレイ
ディスプレイ

表1: プログラムの説明

- @: スタックの先頭番地
- A: 石の種類
- B: 手数
- C: 人数
- D: Aの一時記憶
- E: カウンタ 他
- F: 座標の一時記憶
- G: 座標の一時記憶
- H: カウンタ 他
- I: [処理の深さ] × 8
- J: Iの一時記憶
- K: ループ・カウンタ

- L: ループ・カウンタ
- M: ベクトル
- N: ベクトル
- O: 空白の数
- P: 座標の一時記憶
- Q: 座標の一時記憶
- R: 評価値
- S: [読みの深さ] × 8
- T: 枝刈りの可否
- V: ベクトルデータの先頭番地
- X: 座標
- Z: 盤の先頭番地

表2: 変数の割り当て



