

卒業研究報告

題目

「Visual C++」によるカードゲーム作成

指導教員

原 央 教授

報告者

山科 高志

平成 13年 2月 9日

高知工科大学 電子・光システム工学科

目次

1 . 概要	2
2 . Visual C++ について	3
3 . 「Black Jack」 について	4
3 - 1 . 「Black Jack」 のルール	4
4 . プログラム内容の解説	6
4 - 1 . プログラムを組むにあたって	6
4 - 2 . 基本概念	6
4 - 3 . 基本的な流れ	7
4 - 4 . プログラム内容の解説	8
4 - 4 - 1 . 起動直後に行われる処理	8
4 - 4 - 2 . 「ヒット」 時の処理	18
4 - 4 - 3 . 「スタンド」 時の処理	19
5 . 参考文献	21
6 . 付録 1 各フローチャート	22
7 . 付録 2 プログラムリスト	27

1. 概要

この卒業研究のは、Windows上で動作するプログラムを作成する事を目的とする。

この目的を達するために「VisualC++」を選んだ。これは身の回りに在るオペレーティング・システムの大半を占めるWindows上で動作する事プログラムを組むことが出来、動作確認がしやすいと言う理由からである。

またプログラム内容はゲームとし、カードゲームの一種である「BlackJack」を選んだ。

2. VisualC++について

「VisualC++」はWindows上のアプリケーションが開発できるプログラミングツールである。Windowsが高度な機能を持つようになり、それとともなってその実力を十分に引き出せる高度なアプリケーションが必要になってきた。Windows上の開発ツールとしては「VisualBasic」が広く普及しているが、それと並んで「VisualC++」も重要な開発ツールである。双方ともWindows用プログラムを開発するための十分な機能を持ち合わせているが、比較すると「VisualBasic」の方は基本機能の習得が比較的容易で、小規模のプログラムならすぐに作成することができるようになっている。それに対して「VisualC++」はより複雑なプログラムを作る能力を持っているが、反面大変回りくどい作業を必要とする。

上記のように性格の異なる両者はそれぞれに十分な機能を備えていて、個々のプログラマーの好み・用途に応じて使い分けられている重要な開発ツールである。

3. 「Black Jack」について

「Black Jack」は親と子に分かれ、手札が21点を超えないように21点に近づけるのが目的のカードゲームである。カジノ等でも遊ばれており、比較的有名で大体のルールはよく知られている。

3-1. 「Black Jack」のルール

形式は親1人对子1～5人の対戦形式で、子がチップを賭け、親との勝負を楽しむ形となる。基本的にカードを卓の上に公開するので親と子の位置関係は余り決まてはいない。

まず子がチップを賭ける。次に適度に繰られたカードを山札とし、親はプレイヤーに表向きで1枚ずつ配り、自分には裏向き（子に数字が見えない様）に1枚配る。次にもう一巡、親も含め全てのカードを表向きで配り、この2枚が基本の手札となる。子は親の手札（1枚は見えないが）と自らの手札を考慮して2つの選択肢から次の行動を選ぶ事が出来る。一つは「ヒット」と言って親からカードを1枚貰う。この時、手札の合計が21を超えたら「バースト」となり、チップは親に没収される。もう一つは「スタンド（ステイ）」と言い手札を確定する。つまりもう「ヒット」をしないという宣言で、子の手札によっては一度も「ヒット」せずに「スタンド」が宣言されることもある。そして1人目の子が「スタンド」すると、次の子に移り「ヒット」と「スタンド」を行う。子の手札がすべて確定したら、親は手札の1枚目を公開し、手札が21に近づく様にカードを取る。親が「バースト」したら、「バースト」していない子に賭けたチップと同額のチップを払う。そうでなければ、最も（21以下で）21に近い人が勝ちとなる。

手札の点数の計算は、「A」が11または1、数字札はその数字と同じ、「J」「Q」「K」が10となる（表1）。また、「バースト」時、手札に「A」あった場合は「A」を11から1にして再計算できる。つまり手札が「A」「A」だった時、点数は22から12に下げる事が出来るのである。

A	11、または1
2~10	その数字と同じ
J、Q、K	10

----表1----

子が勝てば賭けたチップと同額のチップが親から払われ、親が勝てば子のチップは全て没収となる。

以上が基本的な流れであるが、手札を作った時点で親がブラックジャック、つまり「A」と絵札（「J」「Q」「K」）だった場合は親の勝ちとなる。ただし子がそれよりも強いブラックジャックだった場合は、その子の勝ちとなる。

「Black Jack」では、子が勝てば賭けた額と同額のチップが支払われる（つまり場に賭けたチップが、2倍になって返ってくる）。しかし、もしそのとき子の手札の内容が「役」になっていれば、チップの増加は2倍に留まらず2.5~1.5倍になる。ブラックジャックの強さとは、増加チップの多さで決まる。「Black Jack」の役は以下の6種類である（表2）。

表ブラックジャック（チップ×1.5）	「スペードのA」と「スペードの絵札」
裏ブラックジャック（×5）	同じスートの「A」と「絵札」
ブラックジャック（×2.5）	違うスートの「A」と「絵札」
セブンカード（×1.0）	手札が7枚になっても21を超えない時
シックスカード（×5）	手札が6枚になっても21を超えない時
スリーセブン（×1.0）	「7」が3枚揃った時

----表2----

つまり3種のブラックジャックの内、「表ブラックジャック」が最も強い。

4 . プログラム内容の解説

4 - 1 . プログラムを組むにあたって

基本的に本に載っている「Black Jack」のルールを採用した。しかし、カジノ等で行われている本格的なルールは省いてある。例えば、「スプリット」というルールは最初の手札2枚が同じ数字であった場合、手札を2組に出来るというものである。これらは少々馴染みがないので採用しないこととした。また、表示スペースとカードの大きさから、手札の増えすぎを抑えるために「セブンカード」に達すれば勝ちとした。「セブンカード」とは、手札が7枚以上になっても21を超えていない場合に成り立つ役で、本来はこの役で勝つとチップが10倍されるだけなのだが、本ゲームでは手札の最大を7枚に限定するために無条件で勝ちとした。

4 - 2 . 基本概念

本プログラム内ではカードは全てカード番号で処理されている。そのカード番号は「スペードのA」が1、「ハートのA」が2、「ダイヤのA」が3、「クラブのA」4、以下「クラブのK」まで52の番号がふってある。そしてそれぞれ山札に相当する配列や手札に相当する配列に順次代入していくことにより、カードの動きを実現している。基本的な流れは、山札に相当する配列にランダムで1から52までの数字を重複がないように代入する。この時、手札の上限は7枚なので単純計算で14枚の山札があればよい。次に「Black Jack」のルール通りに手札を配る。これは山札に代入されたカード番号を、山札の1番目からそれぞれ順番に子と親の手札に相当する配列に代入しただけである。そして手札が配られたら、カード番号からそれぞれのカードのスイートと数字を算出し、それをもとにカードを表示させる。カードの表示には18ファイルのBITMAPを使用した。内訳はスイート4ファイル+数字13ファイル+カードの裏1枚である。スイート画像はカードのフルサイズと同じサイズで上中下と3パートに分けることができる。上と下にはスイートが描いてあり、中は空白である。数字画像の横幅はカードのサイズと同じであるが、縦幅はスイート画像の上下に描いてあるスイートを隠さないように狭めてある。表示の際にスイートを表示して、スイート画像中部の空白に数字画像を上書き表示させる事によってカードを表現している。

4 - 3 . 基本的な流れ

プログラムの基本的な流れとしては、大まかに3つのパートに分けることができる。

- ・ 起動直後に行われる処理
- ・ プレイヤーが「ヒット」したときに行われる処理
- ・ プレイヤーが「スタンド」したときに行われる処理

以上の3つである。「ヒット」と「スタンド」はそれに相当するボタンをクリックすることにより処理が開始される。

プログラムの起動直後に行われる処理は、

- ・ 配列の初期化、山札の作成、カード配り
- ・ ゲーム台と手札の表示

以上の2つである。内容は「ヒット」が子の処理、「スタンド」が親の処理を行う。「ヒット」はヒットボタンがクリックされると、山札から新たなカード番号を子の手札に加えカード画像を表示し、合計を再計算して必要なら「A」を11から1になおし、「バースト」判定をする。「スタンド」も基本的には「ヒット」処理と同じだが、唯一違うのは親の思考が入っていることである。スタンドボタンを押せば親の思考があり、カードを加えるならヒットボタンと同じく山札から親の手札へカードを配り、それに対応したカード画像を表示する。そして手札合計の再計算・「A」の処理・バースト判定の後、再び親の思考に戻り、以下親が「スタンド」するまで繰り返される。

4 - 4 . プログラムの内容解説

プログラムの内容に入る前に、プログラムの基礎的な用語解説をしておく。

関数：

命令文の塊のこと。

C++では命令文は関数の中に所属する。

変数：

プログラム内で処理されるデータを記憶しておくところ。

配列：

複数の変数を成分を持った、変数の塊のこと。

コメント：

プログラマーが記述の内容を、解るように書き加えるメモ書きのこと。

コメントにしておくとその部分の記述はプログラムに影響しない。

4 - 4 - 1 . 起動直後の処理

まず、プログラム起動後に最初の処理として以下 (Pro.1) の16行の記述がある。その内7・9行目は区切りとして書かれたコメントである。

```
////////////////////////////////////
```

```
1      void syokika(),yamaran(),kubari(),padown(),cadown();
2
3      syokika();
4      yamaran();
5      kubari();          //子関数:innumb(),insuit(),pmaxsan(),cmaxsan()
6      padown();cadown();      // 処理》手札が「A」「A」の時の 11  1
7      //-----
8      Invalidate();      //OnPaint に飛ぶ
9      //-----
10/*   sprintf(ccfirst,"スーツは = %d、数字は = %d",csuit[1],cnumb[1]);*/
11     printf(ccfirst,"ディーラー「さあ、どうぞ」");
```

```

12     m_ed2.SetWindowText(ccfirst);
13     sprintf(mmaaxx,"プレイヤーの合計点は = %d",pnumb[0]);
14     m_ed1.SetWindowText(mmaaxx);
15     sprintf(cmmaaxx,"ディーラーの合計点は = ??");
16     m_ed3.SetWindowText(cmmaaxx);

```

```

////////////////////////////////////

```

```

-----Pro.1-----

```

Pro.1 の内容を上から解説していくと、1 行目は関数呼び出しに必要な宣言である。

3 行目が山札・手札等に相当する配列を初期化する関数 syokika (Pro.2) を実行させる命令であり、内容である初期化関数は以下 Pro.2 のようになっている。内容としては、全ての配列に「0」を代入させている。繰り返し回数を制御する変数 yamamax と havemax はプログラムの冒頭でそれぞれ 20 と 10 が代入してある

```

////////////////////////////////////

```

```

void syokika(){
    int co01;
    for(co01 = 0;co01 < yamamax;++co01){
        yama[co01] = 0;
    }
    for(co01 = 0;co01 < havemax;++co01){
        pcard[co01] = 0;
        ccard[co01] = 0;
        psuit[co01] = 0;
        csuit[co01] = 0;
        pnumb[co01] = 0;
        cnumb[co01] = 0;
    }
}

```

```

////////////////////////////////////

```

```

-----Pro.2-----

```

次に Pro.1 で 4 行目に呼び出されている関数は以下 Pro.3 に示した山札を作る関数 yamaran (Pro.3) である。乱数を利用してカード番号を代入している。変数 yamamax には 20 がすでに代入されており、山札は 19 枚しか作られない。

繰り返し回数が - 1 されているのは、繰り返しを終了させるキーワードに値「0」を使う場合があるからである。つまり 19 回しかカード番号を代入しないことによって、配列の最後である 20 番目の変数に「0」が残してある。また変数 che01 により重複チェックがされており、重複を防いでいる。

```
////////////////////////////////////
```

```
void yamaran(int ranmax,int ranmin){
    int rand();
    int che01 = 0;
    int co01, co02,co03 = 1;

    srand((unsigned int)time(0));

    rand();
    for(co01 = 1;co01 < yamamax - 1;){
        ran01 = ranmin +(int)(rand() * (ranmax - ranmin + 1) / (1 + RAND_MAX));
        che01 = 0;
        for(co02 = 1;yama[co02] != 0;++co02){
            if(ran01 == yama[co02]){
                che01 = 10;break;
            }
        }
        if(che01 == 0){
            yama[co03] = ran01;++co03;++co01;
        }
    }
}
```

```
////////////////////////////////////
```

```
-----Pro.3-----
```

5行目は最初のカード配りを行う関数 kubar i (Pro.4) を呼び出している。この関数は山札から子、親それぞれの手札にカード番号を渡しカード配りを再現している。実際のカード配りと同様に、山札の1番目から子の1番目・親の1番目・子の2番目・親の2番目とそれぞれの配列に代入していく。またこの関数内に出てくる変数 yamano・pcardno・ccardno は、それぞれ山札と手札の処理を制御する為のもので、変数 yamano は山札配列からカード番号を参照する時に次に参照する配列の番号を現している。変数 pcardno はプレイヤーの手札の配列番号を指定するもので、カード画像を表示する時の表示位置の決定時にも参照される。変数 ccardno も同様にコンピューターの手札の配列番号を管理している。

```
////////////////////////////////////
```

```
void kubar i(){
    void insuit(),innumb(),pmaxsan(),cmaxsan();

    pcard[1] = yama[1];
    ccard[1] = yama[2];
    pcard[2] = yama[3];
    ccard[2] = yama[4];
    yamano = 5;                //次に引くカードは「山札の5枚目」
    pcardno = 3;ccardno = 3;   //次に持つ位置は「p c共に3枚目」
    psuit[0] = 2;csuit[0] = 2; //手札が2枚ある」と言う意味

    insuit();innumb();
    pmaxsan();cmaxsan();
}
```

```
////////////////////////////////////
```

```
----Pro.4----
```

上記の関数 kubar i 内では、さらに4つの関数 insuit・innumb・pmaxsan・cmaxsan が呼び出されている。

この内、関数 insuit (Pro.5)・innumb (Pro.6) は親と子双方の手札配列に代入されているカード番号からスートと数字を算出し、それを記憶しておく別の配列に代入する役割を持っている。この両関数は手札が増えたときなどの再計算にも呼び出されている。両関数を以下に示す。Pro.5中に出てくる配列 psuit・csuit に手札のスートが記憶され、Pro.6

の配列 pnumb・cnumb が数字を記憶する役割を持っている。

```
////////////////////////////////////
```

```
void insuit(){
    int co01;          //1:スゝ-ト 2:ハ-ト 3:ダイヤ 0:クワ-ハ-

    for(co01 = 1;pcard[co01] != 0;++co01){
        psuit[co01] = pcard[co01] % 4;
    }
    for(co01 = 1;ccard[co01] != 0;++co01){
        csuit[co01] = ccard[co01] % 4;
    }
}
```

```
////////////////////////////////////
```

-----Pro.5-----

```
////////////////////////////////////
```

```
void innumb(){
    int co01;

    for(co01 = 1;pcard[co01] != 0;++co01){
        pnumb[co01] = ((pcard[co01] - 1) / 4) + 1;
    }
    for(co01 = 1;ccard[co01] != 0;++co01){
        cnumb[co01] = ((ccard[co01] - 1) / 4) + 1;
    }
}
```

```
////////////////////////////////////
```

-----Pro.6-----

また関数 kubar i 中の他方の関数 pmaxsan (Pro.7)・ cmaxsan (Pro.8) は、それぞれプレイヤーとコンピューターの手札の合計を計算し、配列 pnumb・cnumb の1番目の成分に記憶します。配列 pnumb についての詳細は1番目の成分である pnumb[0]に手札の合計が代入され、次の成分 pnumb[1]に手札の1枚目の数字が、pnumb[2]に手札の2枚目の数字がと言う風に手札の数字が代入されるようにしました。

```
////////////////////////////////////
```

```
void pmaxsan(){
    int co01;

    pnumb[0] = 0;
    for(co01 = 1;pnumb[co01] != 0;++co01){
        if((pnumb[co01] >= 1) && (pnumb[co01] <= 14)){
            if(pnumb[co01] == 1){
                pnumb[0] += 11;}
            if((pnumb[co01] >= 2) && (pnumb[co01] <= 10)){
                pnumb[0] += pnumb[co01];}
            if((pnumb[co01] >= 11) && (pnumb[co01] <= 13)){
                pnumb[0] += 10;}
            if(pnumb[co01] == 14){
                pnumb[0] += 1;}
        }
        else{
            AfxMessageBox("!!警告!! 関数 pmaxsan に異常");
        }
    }
}
```

```
////////////////////////////////////
```

```
----Pro.7----
```

```
////////////////////////////////////
```

```
void cmaxsan(){
    int co01;
```

```

cnumb[0] = 0;
for(co01 = 1;cnumb[co01] != 0;++co01){
    if((cnumb[co01] >= 1) && (cnumb[co01] <= 14)){
        if(cnumb[co01] == 1){
            cnumb[0] += 11;}
        if((cnumb[co01] >= 2) && (cnumb[co01] <= 10)){
            cnumb[0] += cnumb[co01];}
        if((cnumb[co01] >= 11) && (cnumb[co01] <= 13)){
            cnumb[0] += 10;}
        if(cnumb[co01] == 14){
            cnumb[0] += 1;}
    }
    else{
        AfxMessageBox("!!警告!! 関数 cmaxsan に異常");
    }
}
}

```

```

////////////////////////////////////

```

-----Pro.8-----

次に Pro.1 の 6 行目の関数 padown (Pro.9)・ cadown (Pro.10) の役割は、それぞれプレイヤーとコンピューターの手札中の「A」を 1 1 から 1 に変化させることである。プレイヤーを担当する方の関数 padown で見ると、まずプレイヤーの手札が 2 1 を超えているか調べ超えていれば次に手札中の「A」の有無を調べる。そして「A」が在れば 1 1 を 1 に下げて、すでに解説した関数 pmaxsan を呼び出し合計を再計算する。一応、再計算後も 2 1 を越えていたらその作業を手札の終わりまで続けるようになっている。内容はコンピューターの「A」処理を担当する関数 cadown でも同じである。

```

////////////////////////////////////

```

```

void padown(){
    void pmaxsan();
    int co01;

```

```

        for(co01 = 1;pnumb[0] > 21;++co01){
            if(pnumb[co01] == 0){
                break;}
            if(pnumb[co01] == 1){
                pnumb[co01] = 14;}
            pmaxsan();
        }
    }
}

```

```

////////////////////////////////////

```

```

-----Pro.9-----

```

```

////////////////////////////////////

```

```

void cadown(){
    void cmaxsan();
    int co01;

    for(co01 = 1;cnumb[0] > 21;++co01){
        if(cnumb[co01] == 0){
            break;}
        if(cnumb[co01] == 1){
            cnumb[co01] = 14;}
        cmaxsan();
    }
}

```

```

////////////////////////////////////

```

```

-----Pro.10-----

```

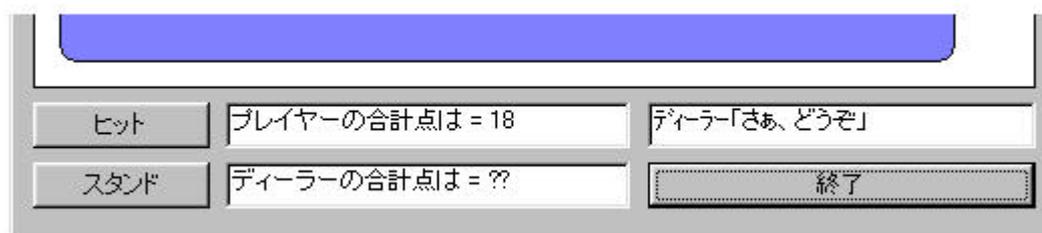
8行目の関数 Invalidate は、初期の表示のために呼び出される関数です。この関数については、後で説明する。

10 ~ 16行目 (Pro.11) は、プログラム実行画面の下側にある3つのメッセージボックス (図1) にテキストを表示させる部分である。その内、1行目デバッグ用の表示を行うようになっていて、隠されている親の手札の1枚目をプログラマーに伝える役割を持っている。通常のゲーム時には、プレイヤーに見せないために1行目はコメントとし、2

行目のセリフで埋めておく。 1行目を有効にするためには2行目をコメントにしておく必要がある。

```
////////////////////////////////////  
  
1/*    sprintf(ccfirst,"スートは = %d、数字は = %d",csuit[1],cnumb[1]);*/  
2    printf(ccfirst,"ディーラー「さあ、どうぞ」);  
3    m_ed2.SetWindowText(ccfirst);  
4    sprintf(mmaaxx,"プレイヤーの合計点は = %d",pnumb[0]);  
5    m_ed1.SetWindowText(mmaaxx);  
6    sprintf(cmmaaxx,"ディーラーの合計点は = ??");  
7    m_ed3.SetWindowText(cmmaaxx);
```

```
////////////////////////////////////  
-----Pro.11-----
```



-----図 1-----

最後に残っている8行目の関数 `Invalidate` だが、これは別の場所に書いてある最初の表示を行わせる命令 (Pro.12) である。 表示内容はゲーム台の表示と最初に配られた2枚の手札である。 8・9行目はコメントである。

まず1・2行目で表示に必要な宣言をする。 1行目の `m_land1` は表示スペースを指している。 3行目はゲーム台の画像を展開する変数を確保している。 4行目で画像ファイル `IDB_LAND1` を3行目で確保した変数内に展開し、5行目でコンピューターが表示のために持っている変数を使えるようにして6行目で画像を展開した変数を表示のために選択している。 そして最後に7行目で選択した画像を場所を `X,Y` 座標で指定しながら表示の命令をしている。 9行目以降はカードを表示するための記述だが方法はゲーム台を表示したときと同じなので、ここでは割愛する。

```
////////////////////////////////////
```

```
1      CDC* pDC=m_land1.GetDC();
2      CDC myDC;
3      CBitmap myBMP;
4      myBMP.LoadBitmap(IDB_LAND1);
5      myDC.CreateCompatibleDC(pDC);
6      myDC.SelectObject(myBMP);
7      pDC->BitBlt(0,0,499,398,&myDC,0,0,SRCCOPY);
8      //-----
9      // 手札 1 枚目スート表示
10     CBitmap myPSUIT01;
11     int iti01 = 1;
12     if(psuit[iti01] == 1){
13         myPSUIT01.LoadBitmap(IDB_SSUITS);myDC.SelectObject(myPSUIT01);
14         pDC->BitBlt(22 + (55 * (iti01 - 1)),205,48,68,&myDC,0,0,SRCCOPY);
```

~ (中略) ~

```
15     // 手札 1 枚目数字表示
16     CBitmap myPNUMB01;
17     if(pnumb[iti01] == 1){
18         myPNUMB01.LoadBitmap(IDB_SNUMBA);myDC.SelectObject(myPNUMB01);
19         pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);
```

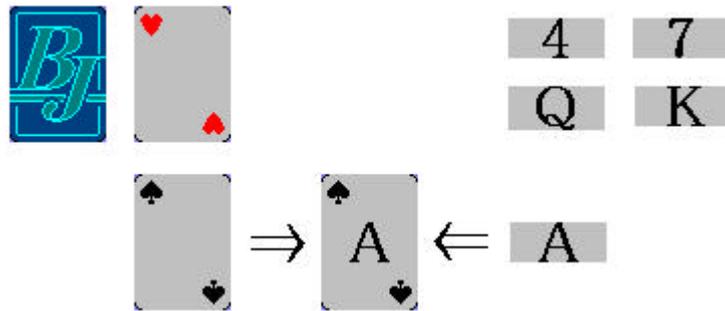
~ (中略) ~

```
////////////////////////////////////
```

----Pro.12----

カードの表示はスート画像と数字画像の2種類の画像によって表示されている。例えば「スペードのA」を表示したいとき、「スペード」のスート画像と「A」の数字画像を重ねて表示することによって1枚のカードのように表示することが出来る(図2)。この方法によって52枚のトランプカードを17の画像(スート4、数字13)で全て表現することが可能となっている。

(カードの画像としてはもう1枚、カードの裏画像もある(図2左上))



-----図 2-----

以上が起動直後に行われる処理である。

4 - 4 - 2 .「ヒット」時の処理

プレイヤーが「ヒット」ボタンを押したときに処理される内容は、

- ・ 山札から新しいカード番号をとって、手札に入れる
- ・ 新しいカードのスイートと数字を算出する
- ・ 「バースト」していないかを調べ、「A」処理を行う
- ・ 手札の合計を算出する
- ・ 手札が21を超えていたら「バースト」でプレイヤーの負けを表示する
- ・ 手札が7枚に達していたら「セブンカード」でプレイヤーの勝ちを表示する

以上の処理が必要となる。しかし、処理内容は起動直後の処理と同じ事が多く、つまり同じ関数がまた呼び出されているだけなので詳しい説明は割愛する。

4 - 4 - 3 . 「スタンド」時の処理

スタンド時の処理として必要なのは、

- ・ 親、すなわちコンピューターの思考
- ・ カードの表示
- ・ 結果の判定

以上の処理が必要となる。

この内、親の思考は以下 (Pro.13) のようになっている。

(21 ページに、フローチャートを付属)

////////////////////////////////////

```
if(cnumb[0] > pnumb[0]){cisi = 0;}
if(cnumb[0] < pnumb[0]){cisi = 1;}
if(cnumb[0] == pnumb[0]){
    if(sa == 1){                //2 の位置 が、引く確立
        if(ckettei[ccardno - 3] <= 2){cisi = 1;}}
    if(sa == 2){
        if(ckettei[ccardno - 3] <= 3){cisi = 1;}}
    if(sa == 3){
        if(ckettei[ccardno - 3] <= 5){cisi = 1;}}
    if(sa == 4){
        if(ckettei[ccardno - 3] <= 10){cisi = 1;}}
    if(sa == 5){
        if(ckettei[ccardno - 3] <= 20){cisi = 1;}}
    if(sa == 6){
        if(ckettei[ccardno - 3] <= 50){cisi = 1;}}
    if(sa == 7){
        if(ckettei[ccardno - 3] <= 70){cisi = 1;}}
    if(sa == 8){
        if(ckettei[ccardno - 3] <= 85){cisi = 1;}}
    if(sa == 9){
```

```

        if(ckettei[ccardno - 3] <= 93){cisi = 1;}
    if(sa == 10){
        if(ckettei[ccardno - 3] <= 98){cisi = 1;}
    if(sa >= 11){
        cisi = 1;}
    }

```

////////////////////////////////////

-----Pro.13-----

カードの表示はすでに説明してあるので割愛する。
 結果の判定は以下 (Pro.14) のように、非常に簡単な物となっている

////////////////////////////////////

```

    if(pnumb[0] > cnumb[0]){
        AfxMessageBox("貴方の勝ちです");}
    if(pnumb[0] == cnumb[0]){
        AfxMessageBox("引き分けです");}
    if(pnumb[0] < cnumb[0]){
        AfxMessageBox("貴方の負けです");}

```

////////////////////////////////////

-----Pro.14-----

5. 参考文献

今回の卒業研究を行うにあたって、下記の文献を参考にした。

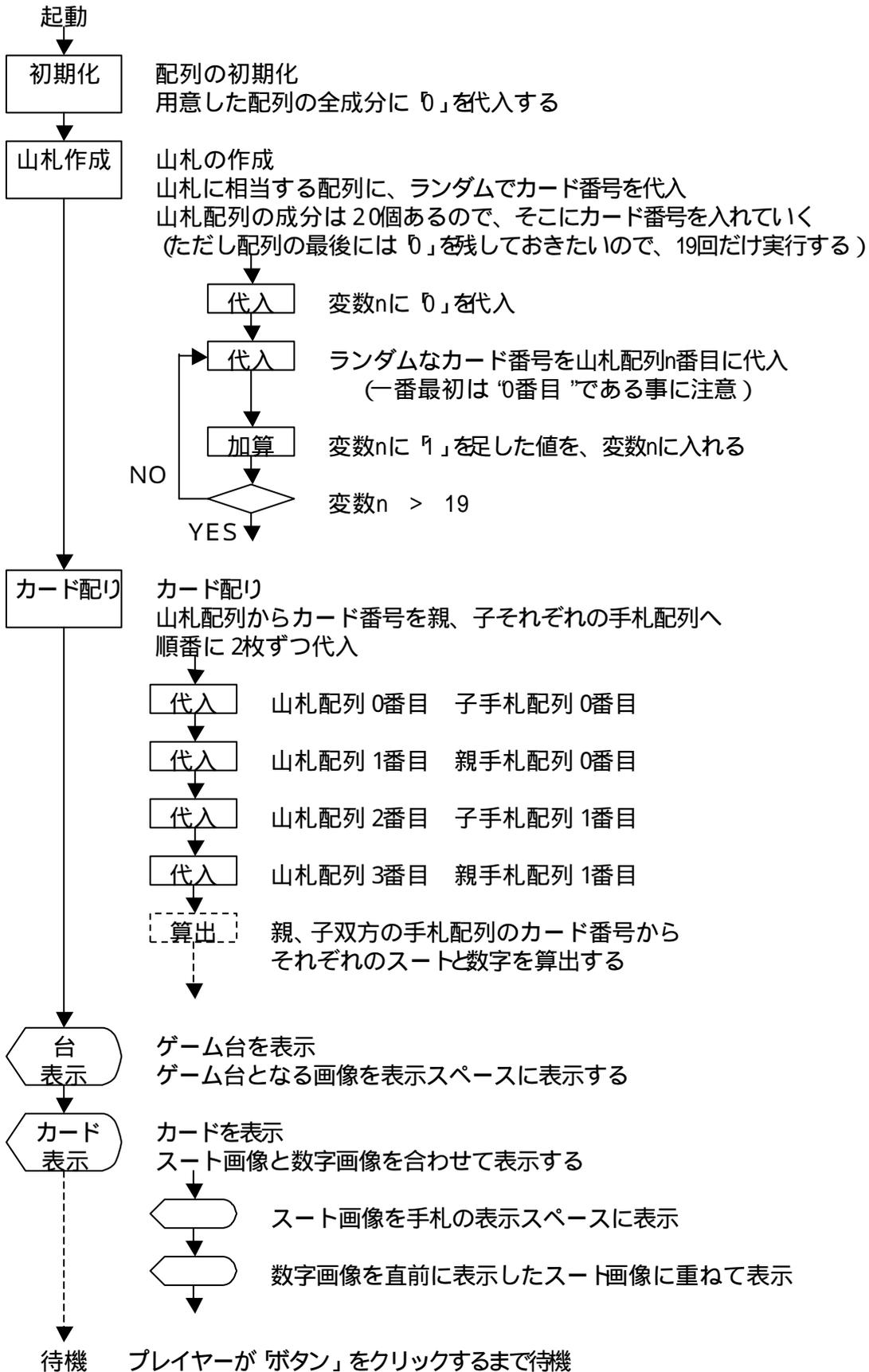
- ・新Visual C++ 6.0 入門～スーパービギナー編～
著者：林 晴比古 (SOFTBANK)
- ・新Visual C++ 6.0 入門～ビギナー編～
著者：林 晴比古 (SOFTBANK)
- ・初めてのVisual C++ 6.0
著者：山地 秀美 (技術評論社)

6 . 付録 1 各フローチャート

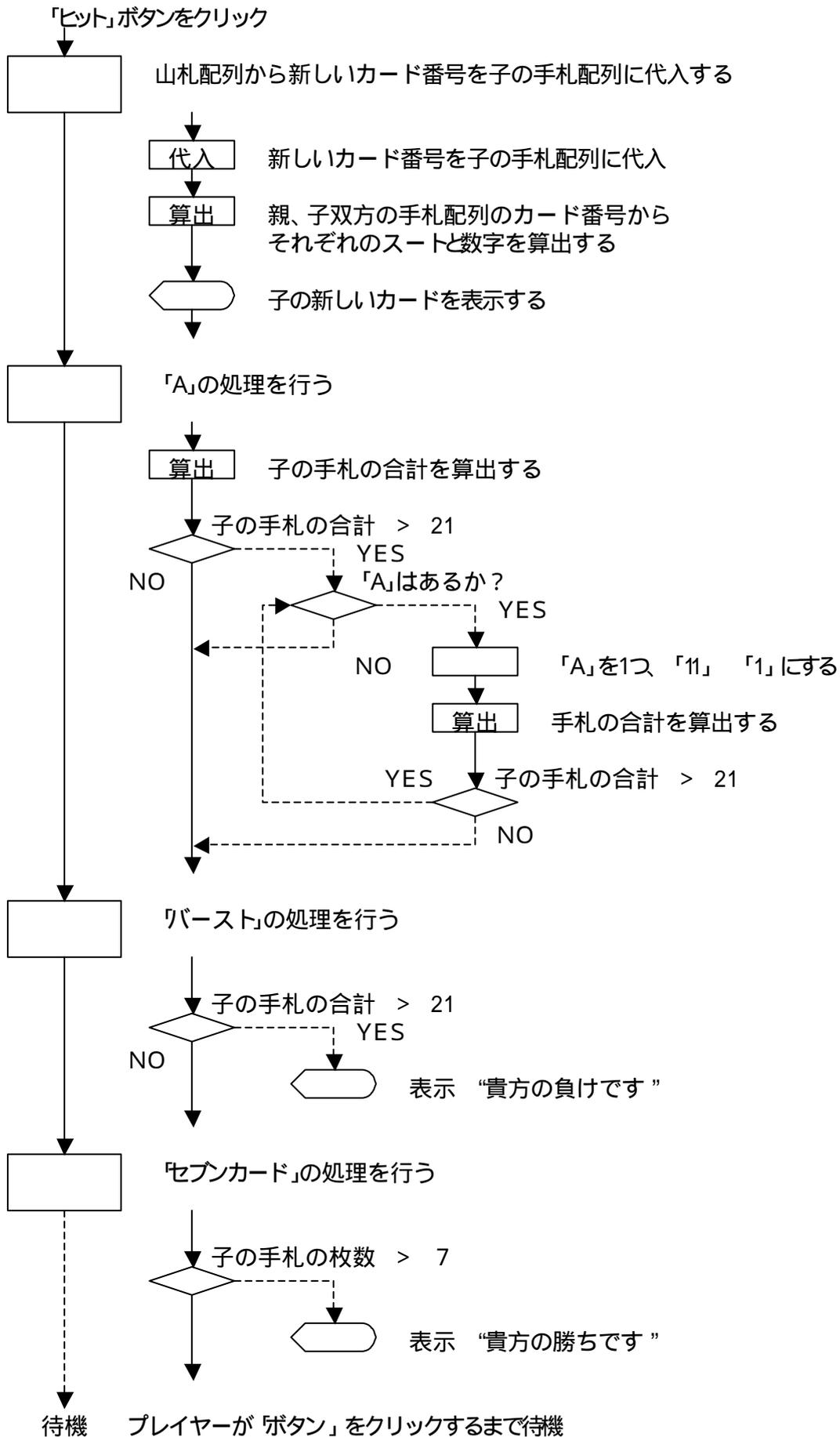
- ・ 起動直後に行われる処理
- ・ 「ヒット」時の処理
- ・ 「スタンド」時の処理
- ・ 親の思考

上記の簡単な流れを、簡単なフローチャートとして表す。

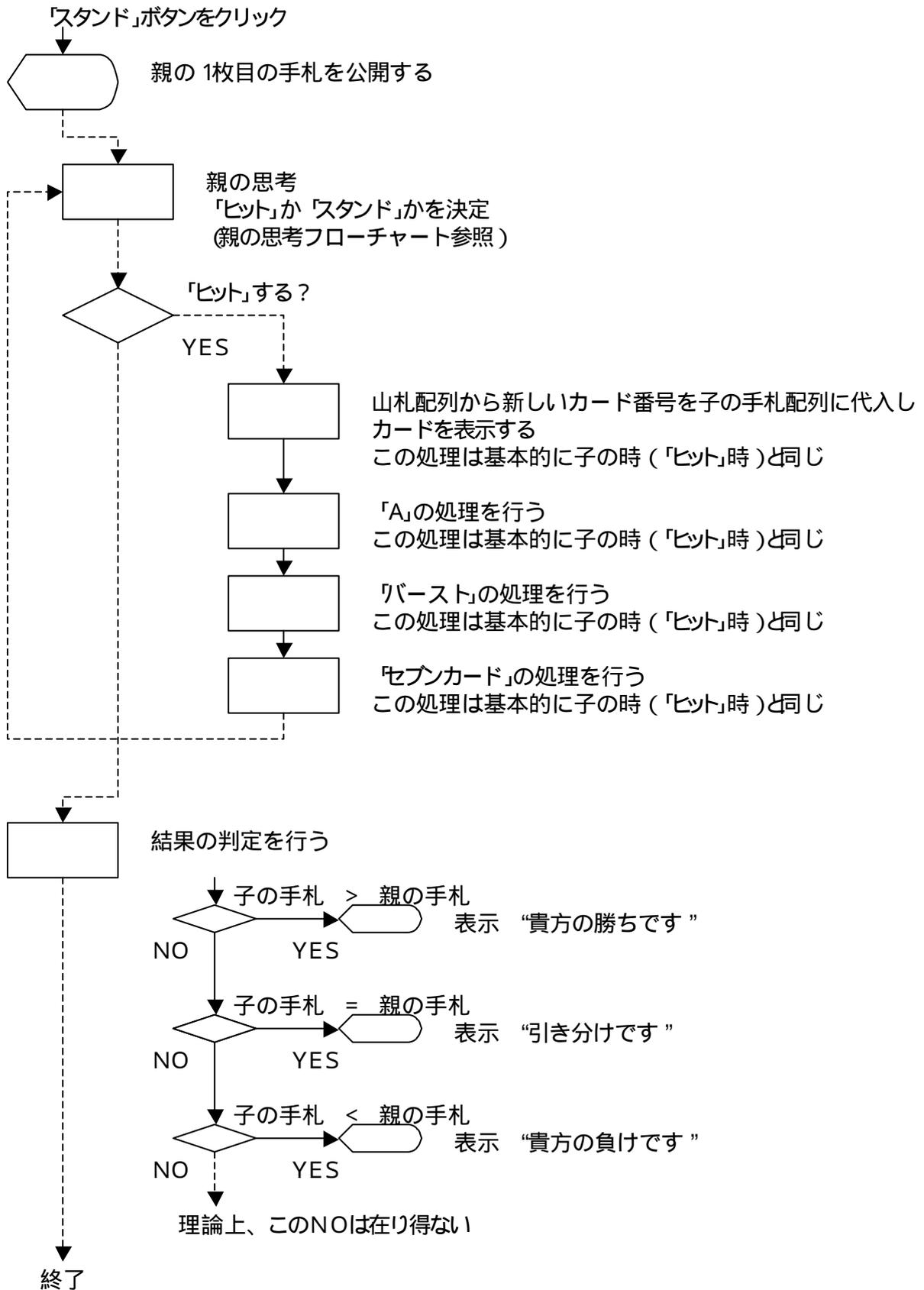
起動直後に行われる処理



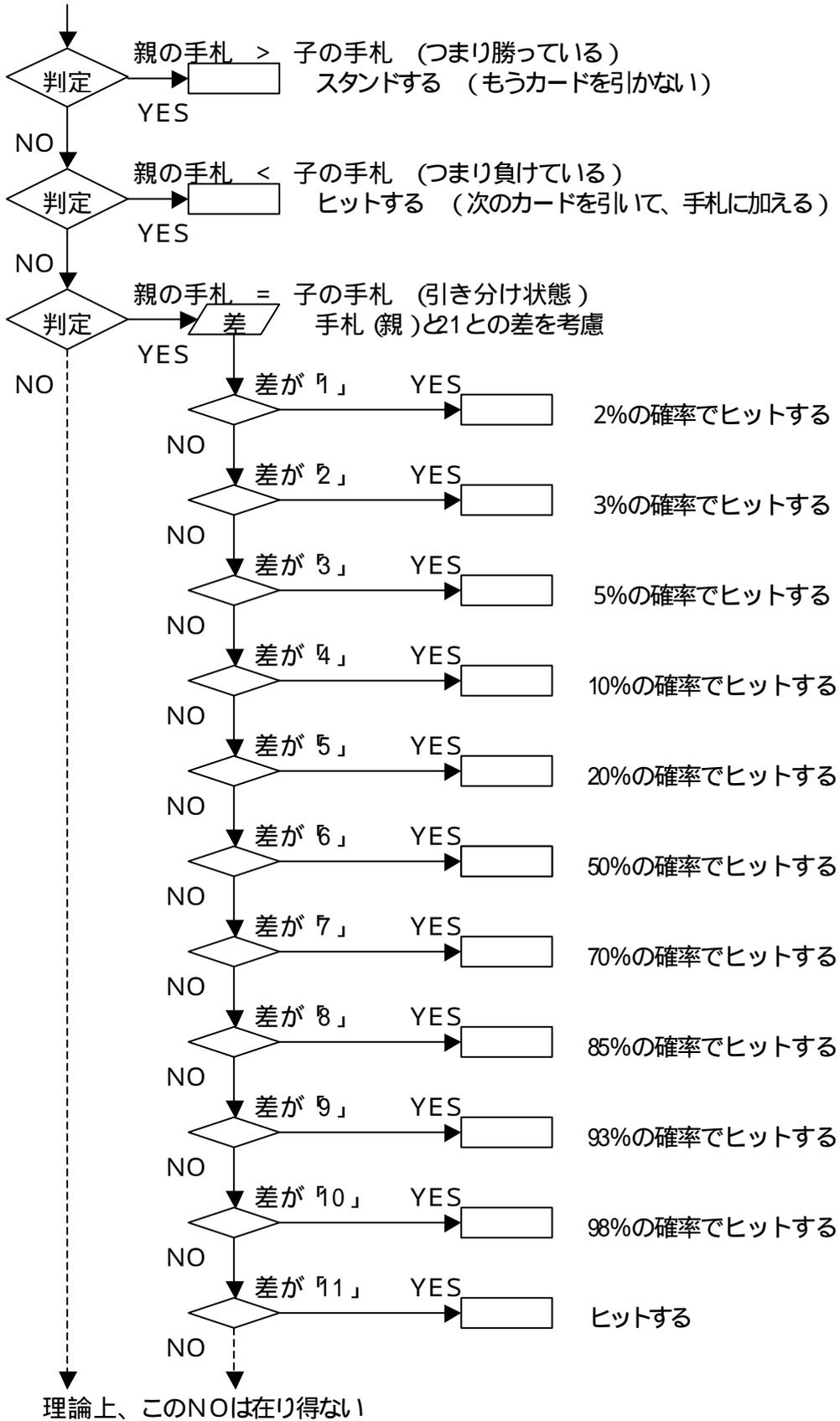
プレイヤーが「ヒット」したときに行われる処理



プレイヤーが「スタンド」したときに行われる処理



親の思考



7 . 付録 2 プログラムリスト BJver100Dlg.cpp

今回作成したプログラムのリストを付録としてつけます。 論文中では書けなかったそれぞれの関数の位置などを、このリストによって補完できればと思います。

```
// BJver100Dlg.cpp : インプリメンテーション ファイル
//
```

```
#include "stdafx.h"
#include "BJver100.h"
#include "BJver100Dlg.h"
```

```
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

```
//=====
int ran01,yamano,pcardno,ccardno,cisi;
char ccfirst[50],mmaaxx[50],cmmaaxx[50];
const int yamamax = 20,havemax = 10;
int yama[yamamax];
int pcard[havemax];
int ccard[havemax];
int psuit[havemax];
int csuit[havemax];
int pnumb[havemax]; // [0]:手札合計点 [1] ~ [havemax]:手札の数字
int cnumb[havemax]; // [0]:手札合計点 [1] ~ [havemax]:手札の数字
int ckettei[havemax]; // 親の判断
//=====
```

```
////////////////////////////////////
// アプリケーションのバージョン情報で使われている CAboutDlg ダイアログ
```

```
class CAboutDlg : public CDialog
{
public:
    CAboutDlg();
```

```
// ダイアログ データ
//{{AFX_DATA(CAboutDlg)
enum { IDD = IDD_ABOUTBOX };
//}}AFX_DATA
```

```
// ClassWizard は仮想関数のオーバーライドを生成します
```

```
//{{AFX_VIRTUAL(CAboutDlg)
```

```
protected:
```

```
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV のサポート
//}}AFX_VIRTUAL
```

```
// インプリメンテーション
```

```
protected:
```

```
//{{AFX_MSG(CAboutDlg)
```

```
//}}AFX_MSG
```

```
DECLARE_MESSAGE_MAP()
```

```
};
```

```
CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
```

```
{
```

```
//{{AFX_DATA_INIT(CAboutDlg)
```

```
//}}AFX_DATA_INIT
```

```
}
```

```

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CAboutDlg)
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    //{{AFX_MSG_MAP(CAboutDlg)
    // メッセージ ハンドラがありません。
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CBJver100Dlg ダイアログ

CBJver100Dlg::CBJver100Dlg(CWnd* pParent /*=NULL*/)
    : CDialog(CBJver100Dlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CBJver100Dlg)
    // メモ: この位置に ClassWizard によってメンバの初期化が追加されます。
    //}}AFX_DATA_INIT
    // メモ: LoadIcon は Win32 の DestroyIcon のサブシーケンスを要求しません。
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CBJver100Dlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CBJver100Dlg)
    DDX_Control(pDX, IDC_EDIT3, m_ed3);
    DDX_Control(pDX, IDC_EDIT2, m_ed2);
    DDX_Control(pDX, IDC_EDIT1, m_ed1);
    DDX_Control(pDX, IDC_LAND1, m_land1);
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CBJver100Dlg, CDialog)
    //{{AFX_MSG_MAP(CBJver100Dlg)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BHIT, OnBhit)
    ON_BN_CLICKED(IDC_BSTAND, OnBstand)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CBJver100Dlg メッセージ ハンドラ

BOOL CBJver100Dlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // "バージョン情報..." メニュー項目をシステム メニューへ追加します。

    // IDM_ABOUTBOX はコマンド メニューの範囲でなければなりません。
    ASSERT((IDM_ABOUTBOX & 0xFFFF) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);
}

```

```

CMenu* pSysMenu = GetSystemMenu(FALSE);
if (pSysMenu != NULL)
{
    CString strAboutMenu;
    strAboutMenu.LoadString(IDS_ABOUTBOX);
    if (!strAboutMenu.IsEmpty())
    {
        pSysMenu->AppendMenu(MF_SEPARATOR);
        pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
    }
}

// このダイアログ用のアイコンを設定します。フレームワークはアプリケーションのメイン
// ウィンドウがダイアログでない時は自動的に設定しません。
SetIcon(m_hIcon, TRUE); // 大きいアイコンを設定
SetIcon(m_hIcon, FALSE); // 小さいアイコンを設定

// TODO: 特別な初期化を行う時はこの場所に追加してください。
//=====
void syokika(), yamaran(), kubar i (), padown(), cadown();

syokika();
yamaran();
kubar i (); //子関数: innumb(), insuit(), pmaxsan(), cmaxsan()
padown(); cadown(); // 《処理》手札が「A」「A」の時の11 1
// 《処理》親手札のB J判定
// 《処理》子手札のB J判定 (親がB Jなら)
//-----
Invalidate(); //OnPaint に飛ぶ
//-----
/* sprintf(ccfirst, "スートは = %d、数字は = %d", csuit[1], cnumb[1]); */ //
C sprintf(ccfirst, "ディーラー「さあ、どうぞ」"); //
C m_ed2.SetWindowText(ccfirst); //
C

sprintf(mmaaxx, "プレイヤーの合計点は = %d", pnumb[0]); //c
m_ed1.SetWindowText(mmaaxx); //c

sprintf(cmmaaxx, "ディーラーの合計点は = ??"); //c
m_ed3.SetWindowText(cmmaaxx); //c
//=====

return TRUE; // TRUE を返すとコントロールに設定したフォーカスは失われませ
ん。
}

void CBJver100Dlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFFF) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {

```

```

        CDialog::OnSysCommand(nID, IParam);
    }
}

// もしダイアログボックスに最小化ボタンを追加するならば、アイコンを描画する
// コードを以下に記述する必要があります。MFC アプリケーションは document/view
// モデルを使っているなので、この処理はフレームワークにより自動的に処理されます
。

void CBJver100Dlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // 描画用のデバイス コンテキスト

        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

        // クライアントの矩形領域内の中央
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // アイコンを描画します。
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
        //=====
        //台表示
        CDC* pDC=m_land1.GetDC();

        CDC myDC;
        CBitmap myBMP;

        myBMP.LoadBitmap(IDB_LAND1);
        myDC.CreateCompatibleDC(pDC);

        myDC.SelectObject(myBMP);
        pDC->BitBlt(0,0,499,398,&myDC,0,0,SRCCOPY);
        //-----
        //p手札1枚目スート表示
        CBitmap myPSUIT01;
        int iti01 = 1;
        if(psuit[iti01] == 1){
            myPSUIT01.LoadBitmap(IDB_SSUITS);myDC.SelectObject(myPSUIT01);
            pDC->BitBlt(22 + (55 * (iti01 - 1)),205,48,68,&myDC,0,0,SRCCOPY);}
        if(psuit[iti01] == 2){
            myPSUIT01.LoadBitmap(IDB_SSUITH);myDC.SelectObject(myPSUIT01);
            pDC->BitBlt(22 + (55 * (iti01 - 1)),205,48,68,&myDC,0,0,SRCCOPY);}
        if(psuit[iti01] == 3){
            myPSUIT01.LoadBitmap(IDB_SSUITD);myDC.SelectObject(myPSUIT01);
            pDC->BitBlt(22 + (55 * (iti01 - 1)),205,48,68,&myDC,0,0,SRCCOPY);}
        if(psuit[iti01] == 0){
            myPSUIT01.LoadBitmap(IDB_SSUITC);myDC.SelectObject(myPSUIT01);
            pDC->BitBlt(22 + (55 * (iti01 - 1)),205,48,68,&myDC,0,0,SRCCOPY);}
    }
}

```

```

    if(psuit[iti01] != 1 && psuit[iti01] != 2 && psuit[iti01] != 3 && psuit[iti01] != 0){
        myPSUIT01.LoadBitmap(IDB_RCARD);myDC.SelectObject(myPSUIT01);
        pDC->BitBlt(22 + (55 * (iti01 - 1)),205,48,68,&myDC,0,0,SRCCOPY);}
// p 手札 1 枚目数字表示
CBitmap myPNUMB01;
if(pnumb[iti01] == 1){
    myPNUMB01.LoadBitmap(IDB_SNUMBA);myDC.SelectObject(myPNUMB01);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 2){
    myPNUMB01.LoadBitmap(IDB_SNUMB2);myDC.SelectObject(myPNUMB01);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 3){
    myPNUMB01.LoadBitmap(IDB_SNUMB3);myDC.SelectObject(myPNUMB01);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 4){
    myPNUMB01.LoadBitmap(IDB_SNUMB4);myDC.SelectObject(myPNUMB01);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 5){
    myPNUMB01.LoadBitmap(IDB_SNUMB5);myDC.SelectObject(myPNUMB01);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 6){
    myPNUMB01.LoadBitmap(IDB_SNUMB6);myDC.SelectObject(myPNUMB01);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 7){
    myPNUMB01.LoadBitmap(IDB_SNUMB7);myDC.SelectObject(myPNUMB01);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 8){
    myPNUMB01.LoadBitmap(IDB_SNUMB8);myDC.SelectObject(myPNUMB01);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 9){
    myPNUMB01.LoadBitmap(IDB_SNUMB9);myDC.SelectObject(myPNUMB01);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 10){
    myPNUMB01.LoadBitmap(IDB_SNUMB10);myDC.SelectObject(myPNUMB01);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 11){
    myPNUMB01.LoadBitmap(IDB_SNUMBJ);myDC.SelectObject(myPNUMB01);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 12){
    myPNUMB01.LoadBitmap(IDB_SNUMBQ);myDC.SelectObject(myPNUMB01);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 13){
    myPNUMB01.LoadBitmap(IDB_SNUMBK);myDC.SelectObject(myPNUMB01);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
// c 手札 1 枚目表示 (裏面)
CBitmap myCSUIT01;
iti01 = 1;
myCSUIT01.LoadBitmap(IDB_RCARD);myDC.SelectObject(myCSUIT01);
pDC->BitBlt(90 + (55 * (iti01 - 1)),55,48,68,&myDC,0,0,SRCCOPY);
// p 手札 2 枚目スート表示
CBitmap myPSUIT02;
iti01 = 2;
if(psuit[iti01] == 1){
    myPSUIT02.LoadBitmap(IDB_SSUITS);myDC.SelectObject(myPSUIT02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),205,48,68,&myDC,0,0,SRCCOPY);}
if(psuit[iti01] == 2){
    myPSUIT02.LoadBitmap(IDB_SSUITH);myDC.SelectObject(myPSUIT02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),205,48,68,&myDC,0,0,SRCCOPY);}

```

```

if(psuit[iti01] == 3){
    myPSUIT02.LoadBitmap(IDB_SSUITD);myDC.SelectObject(myPSUIT02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),205,48,68,&myDC,0,0,SRCCOPY);}
if(psuit[iti01] == 0){
    myPSUIT02.LoadBitmap(IDB_SSUITC);myDC.SelectObject(myPSUIT02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),205,48,68,&myDC,0,0,SRCCOPY);}
if(psuit[iti01] != 1 && psuit[iti01] != 2 && psuit[iti01] != 3 && psuit[iti01] != 0){
    myPSUIT02.LoadBitmap(IDB_RCARD);myDC.SelectObject(myPSUIT02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),205,48,68,&myDC,0,0,SRCCOPY);}
// p 手札 2 枚目数字表示
CBitmap myPNUMB02;
if(pnumb[iti01] == 1){
    myPNUMB02.LoadBitmap(IDB_SNUMBA);myDC.SelectObject(myPNUMB02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 2){
    myPNUMB02.LoadBitmap(IDB_SNUMB2);myDC.SelectObject(myPNUMB02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 3){
    myPNUMB02.LoadBitmap(IDB_SNUMB3);myDC.SelectObject(myPNUMB02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 4){
    myPNUMB02.LoadBitmap(IDB_SNUMB4);myDC.SelectObject(myPNUMB02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 5){
    myPNUMB02.LoadBitmap(IDB_SNUMB5);myDC.SelectObject(myPNUMB02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 6){
    myPNUMB02.LoadBitmap(IDB_SNUMB6);myDC.SelectObject(myPNUMB02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 7){
    myPNUMB02.LoadBitmap(IDB_SNUMB7);myDC.SelectObject(myPNUMB02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 8){
    myPNUMB02.LoadBitmap(IDB_SNUMB8);myDC.SelectObject(myPNUMB02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 9){
    myPNUMB02.LoadBitmap(IDB_SNUMB9);myDC.SelectObject(myPNUMB02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 10){
    myPNUMB02.LoadBitmap(IDB_SNUMB10);myDC.SelectObject(myPNUMB02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 11){
    myPNUMB02.LoadBitmap(IDB_SNUMBJ);myDC.SelectObject(myPNUMB02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 12){
    myPNUMB02.LoadBitmap(IDB_SNUMBQ);myDC.SelectObject(myPNUMB02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[iti01] == 13){
    myPNUMB02.LoadBitmap(IDB_SNUMBK);myDC.SelectObject(myPNUMB02);
    pDC->BitBlt(22 + (55 * (iti01 - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
// c 手札 2 枚目スート表示
CBitmap myCSUIT02;
iti01 = 2;
if(csuit[iti01] == 1){
    myCSUIT02.LoadBitmap(IDB_SSUITS);myDC.SelectObject(myCSUIT02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),55,48,68,&myDC,0,0,SRCCOPY);}
if(csuit[iti01] == 2){
    myCSUIT02.LoadBitmap(IDB_SSUITH);myDC.SelectObject(myCSUIT02);

```

```

        pDC->BitBlt(90 + (55 * (iti01 - 1)),55,48,68,&myDC,0,0,SRCCOPY);}
if(csuit[iti01] == 3){
    myCSUIT02.LoadBitmap(IDB_SSUITD);myDC.SelectObject(myCSUIT02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),55,48,68,&myDC,0,0,SRCCOPY);}
if(csuit[iti01] == 0){
    myCSUIT02.LoadBitmap(IDB_SSUITC);myDC.SelectObject(myCSUIT02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),55,48,68,&myDC,0,0,SRCCOPY);}
if(csuit[iti01] != 1 && csuit[iti01] != 2 && csuit[iti01] != 3 && csui
t[iti01] != 0){
    myCSUIT02.LoadBitmap(IDB_RCARD);myDC.SelectObject(myCSUIT02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),55,48,68,&myDC,0,0,SRCCOPY);}
// c 手札 2 枚目数字表示
CBitmap myCNUMB02;
if(cnumb[iti01] == 1){
    myCNUMB02.LoadBitmap(IDB_SNUMBA);myDC.SelectObject(myCNUMB02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),78,48,23,&myDC,0,0,SRCCOPY);}
if(cnumb[iti01] == 2){
    myCNUMB02.LoadBitmap(IDB_SNUMB2);myDC.SelectObject(myCNUMB02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),78,48,23,&myDC,0,0,SRCCOPY);}
if(cnumb[iti01] == 3){
    myCNUMB02.LoadBitmap(IDB_SNUMB3);myDC.SelectObject(myCNUMB02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),78,48,23,&myDC,0,0,SRCCOPY);}
if(cnumb[iti01] == 4){
    myCNUMB02.LoadBitmap(IDB_SNUMB4);myDC.SelectObject(myCNUMB02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),78,48,23,&myDC,0,0,SRCCOPY);}
if(cnumb[iti01] == 5){
    myCNUMB02.LoadBitmap(IDB_SNUMB5);myDC.SelectObject(myCNUMB02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),78,48,23,&myDC,0,0,SRCCOPY);}
if(cnumb[iti01] == 6){
    myCNUMB02.LoadBitmap(IDB_SNUMB6);myDC.SelectObject(myCNUMB02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),78,48,23,&myDC,0,0,SRCCOPY);}
if(cnumb[iti01] == 7){
    myCNUMB02.LoadBitmap(IDB_SNUMB7);myDC.SelectObject(myCNUMB02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),78,48,23,&myDC,0,0,SRCCOPY);}
if(cnumb[iti01] == 8){
    myCNUMB02.LoadBitmap(IDB_SNUMB8);myDC.SelectObject(myCNUMB02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),78,48,23,&myDC,0,0,SRCCOPY);}
if(cnumb[iti01] == 9){
    myCNUMB02.LoadBitmap(IDB_SNUMB9);myDC.SelectObject(myCNUMB02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),78,48,23,&myDC,0,0,SRCCOPY);}
if(cnumb[iti01] == 10){
    myCNUMB02.LoadBitmap(IDB_SNUMB10);myDC.SelectObject(myCNUMB02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),78,48,23,&myDC,0,0,SRCCOPY);}
if(cnumb[iti01] == 11){
    myCNUMB02.LoadBitmap(IDB_SNUMBJ);myDC.SelectObject(myCNUMB02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),78,48,23,&myDC,0,0,SRCCOPY);}
if(cnumb[iti01] == 12){
    myCNUMB02.LoadBitmap(IDB_SNUMBQ);myDC.SelectObject(myCNUMB02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),78,48,23,&myDC,0,0,SRCCOPY);}
if(cnumb[iti01] == 13){
    myCNUMB02.LoadBitmap(IDB_SNUMBK);myDC.SelectObject(myCNUMB02);
    pDC->BitBlt(90 + (55 * (iti01 - 1)),78,48,23,&myDC,0,0,SRCCOPY);}
//=====
}
}

```

```

// システムは、ユーザーが最小化ウィンドウをドラッグしている間、
// カーソルを表示するためにここを呼び出します。
HCURSOR CBJver100Dlg::OnQueryDragIcon()

```

```

{
    return (HCURSOR) m_hIcon;
}

void syokika(){ //=====
    int co01;
    for(co01 = 0;co01 < yamamax;++co01){
        yama[co01] = 0;
    }
    for(co01 = 0;co01 < havemax;++co01){
        pcard[co01] = 0;
        ccard[co01] = 0;
        psuit[co01] = 0;
        csuit[co01] = 0;
        pnumb[co01] = 0;
        cnumb[co01] = 0;
        ckettei[co01] = 0;
    }
}

void yamaran(){ //=====
    int rand();
    int ranmax = 52,ranmin = 1;
    int che01 = 0;
    int co01,co02,co03 = 1;

    srand((unsigned int)time(0));

    rand();
    for(co01 = 1;co01 < yamamax - 1;){
        ran01 = ranmin + (int)(rand() * (ranmax - ranmin + 1) / (1 + RAND_MAX)
);
        che01 = 0;
        for(co02 = 1;yama[co02] != 0;++co02){
            if(ran01 == yama[co02]){
                che01 = 10;break;
            }
        }
        if(che01 == 0){
            yama[co03] = ran01;++co03;++co01;
        }
    }
}

void kubari(){ //=====
    void insuit(),innumb(),pmaxsan(),cmaxsan();

    pcard[1] = yama[1];
    ccard[1] = yama[2];
    pcard[2] = yama[3];
    ccard[2] = yama[4];
    yamano = 5; //次に引くカードは「山札の5枚目」
    pcardno = 3;ccardno = 3; //次に持つ位置は「p、c共に3枚目」
    psuit[0] = 2;csuit[0] = 2; //「手札が2枚ある」という意味

    insuit();innumb();
    pmaxsan();cmaxsan();
}

```

```

void insuit(){          //=====
    int co01;          //1:スゝ-ト 2:ハ-ト 3:タ イ 0:ク口-ハ -

    for(co01 = 1;pcard[co01] != 0;++co01){
        psuit[co01] = pcard[co01] % 4;
    }
    for(co01 = 1;ccard[co01] != 0;++co01){
        csuit[co01] = ccard[co01] % 4;
    }
}

void innumb(){         //=====
    int co01;

    for(co01 = 1;pcard[co01] != 0;++co01){
        pnumb[co01] = ((pcard[co01] - 1) / 4) + 1;
    }
    for(co01 = 1;ccard[co01] != 0;++co01){
        cnumb[co01] = ((ccard[co01] - 1) / 4) + 1;
    }
}

void pmaxsan(){       //=====
    int co01;

    pnumb[0] = 0;
    for(co01 = 1;pnumb[co01] != 0;++co01){
        if((pnumb[co01] >= 1) && (pnumb[co01] <= 14)){
            if(pnumb[co01] == 1){
                pnumb[0] += 11;}
            if((pnumb[co01] >= 2) && (pnumb[co01] <= 10)){
                pnumb[0] += pnumb[co01];}
            if((pnumb[co01] >= 11) && (pnumb[co01] <= 13)){
                pnumb[0] += 10;}
            if(pnumb[co01] == 14){
                pnumb[0] += 1;}
        }
        else{
            AfxMessageBox("!!警告!! 関数pmaxsanに異常");
        }
    }
}

void cmaxsan(){       //=====
    int co01;

    cnumb[0] = 0;
    for(co01 = 1;cnumb[co01] != 0;++co01){
        if((cnumb[co01] >= 1) && (cnumb[co01] <= 14)){
            if(cnumb[co01] == 1){
                cnumb[0] += 11;}
            if((cnumb[co01] >= 2) && (cnumb[co01] <= 10)){
                cnumb[0] += cnumb[co01];}
            if((cnumb[co01] >= 11) && (cnumb[co01] <= 13)){
                cnumb[0] += 10;}
            if(cnumb[co01] == 14){
                cnumb[0] += 1;}
        }
        else{

```

```

        AfxMessageBox("!!警告!! 関数cmaxsanに異常");
    }
}

void CBJver100Dlg::OnBhit()
{
    // TODO: この位置にコントロール通知ハンドラ用のコードを追加してください
    //=====
    void insuit(), innumb(), padown();

    pcard[pcardno] = yama[yamano];
    insuit(); innumb();

    // p手札3枚目以降スーツ表示
    CDC* pDC=m_land1.GetDC();
    CDC myDC;
    myDC.CreateCompatibleDC(pDC);
    CBitmap myPSUIT03;
    if(psuit[pcardno] == 1){
        myPSUIT03.LoadBitmap(IDB_SSUITS); myDC.SelectObject(myPSUIT03);
        pDC->BitBlt(22 + (55 * (pcardno - 1)), 205, 48, 68, &myDC, 0, 0, SRCCOPY);}
    if(psuit[pcardno] == 2){
        myPSUIT03.LoadBitmap(IDB_SSUITH); myDC.SelectObject(myPSUIT03);
        pDC->BitBlt(22 + (55 * (pcardno - 1)), 205, 48, 68, &myDC, 0, 0, SRCCOPY);}
    if(psuit[pcardno] == 3){
        myPSUIT03.LoadBitmap(IDB_SSUITD); myDC.SelectObject(myPSUIT03);
        pDC->BitBlt(22 + (55 * (pcardno - 1)), 205, 48, 68, &myDC, 0, 0, SRCCOPY);}
    if(psuit[pcardno] == 0){
        myPSUIT03.LoadBitmap(IDB_SSUITC); myDC.SelectObject(myPSUIT03);
        pDC->BitBlt(22 + (55 * (pcardno - 1)), 205, 48, 68, &myDC, 0, 0, SRCCOPY);}
    if(psuit[pcardno] != 1 && psuit[pcardno] != 2 && psuit[pcardno] != 3 && ps
uit[pcardno] != 0){
        myPSUIT03.LoadBitmap(IDB_RCARD); myDC.SelectObject(myPSUIT03);
        pDC->BitBlt(22 + (55 * (pcardno - 1)), 205, 48, 68, &myDC, 0, 0, SRCCOPY);}
    // p手札3枚目以降数字表示
    CBitmap myPNUMB03;
    if(pnumb[pcardno] == 1){
        myPNUMB03.LoadBitmap(IDB_SNUMBA); myDC.SelectObject(myPNUMB03);
        pDC->BitBlt(22 + (55 * (pcardno - 1)), 228, 48, 23, &myDC, 0, 0, SRCCOPY);}
    if(pnumb[pcardno] == 2){
        myPNUMB03.LoadBitmap(IDB_SNUMB2); myDC.SelectObject(myPNUMB03);
        pDC->BitBlt(22 + (55 * (pcardno - 1)), 228, 48, 23, &myDC, 0, 0, SRCCOPY);}
    if(pnumb[pcardno] == 3){
        myPNUMB03.LoadBitmap(IDB_SNUMB3); myDC.SelectObject(myPNUMB03);
        pDC->BitBlt(22 + (55 * (pcardno - 1)), 228, 48, 23, &myDC, 0, 0, SRCCOPY);}
    if(pnumb[pcardno] == 4){
        myPNUMB03.LoadBitmap(IDB_SNUMB4); myDC.SelectObject(myPNUMB03);
        pDC->BitBlt(22 + (55 * (pcardno - 1)), 228, 48, 23, &myDC, 0, 0, SRCCOPY);}
    if(pnumb[pcardno] == 5){
        myPNUMB03.LoadBitmap(IDB_SNUMB5); myDC.SelectObject(myPNUMB03);
        pDC->BitBlt(22 + (55 * (pcardno - 1)), 228, 48, 23, &myDC, 0, 0, SRCCOPY);}
    if(pnumb[pcardno] == 6){
        myPNUMB03.LoadBitmap(IDB_SNUMB6); myDC.SelectObject(myPNUMB03);
        pDC->BitBlt(22 + (55 * (pcardno - 1)), 228, 48, 23, &myDC, 0, 0, SRCCOPY);}
    if(pnumb[pcardno] == 7){
        myPNUMB03.LoadBitmap(IDB_SNUMB7); myDC.SelectObject(myPNUMB03);
        pDC->BitBlt(22 + (55 * (pcardno - 1)), 228, 48, 23, &myDC, 0, 0, SRCCOPY);}
    if(pnumb[pcardno] == 8){

```

```

        myPNUMB03.LoadBitmap(IDB_SNUMB8);myDC.SelectObject(myPNUMB03);
        pDC->BitBlt(22 + (55 * (pcardno - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[pcardno] == 9){
    myPNUMB03.LoadBitmap(IDB_SNUMB9);myDC.SelectObject(myPNUMB03);
    pDC->BitBlt(22 + (55 * (pcardno - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[pcardno] == 10){
    myPNUMB03.LoadBitmap(IDB_SNUMB10);myDC.SelectObject(myPNUMB03);
    pDC->BitBlt(22 + (55 * (pcardno - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[pcardno] == 11){
    myPNUMB03.LoadBitmap(IDB_SNUMBJ);myDC.SelectObject(myPNUMB03);
    pDC->BitBlt(22 + (55 * (pcardno - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[pcardno] == 12){
    myPNUMB03.LoadBitmap(IDB_SNUMBQ);myDC.SelectObject(myPNUMB03);
    pDC->BitBlt(22 + (55 * (pcardno - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
if(pnumb[pcardno] == 13){
    myPNUMB03.LoadBitmap(IDB_SNUMBK);myDC.SelectObject(myPNUMB03);
    pDC->BitBlt(22 + (55 * (pcardno - 1)),228,48,23,&myDC,0,0,SRCCOPY);}
++pcardno; ++yamano;
//-----
pmaxsan();
if(pnumb[0] > 21){pdown();} //子関数pmaxsan、バースト処理

char mmaaxx[50]; //c
sprintf(mmaaxx,"プレイヤーの合計点は = %d",pnumb[0]); //c
m_ed1.SetWindowText(mmaaxx); //c

if(pnumb[0] >= 22){
    AfxMessageBox("バースト、貴方の負けです");
    CDialog::OnOK();
}
//=====
}

void pdown(){ //=====
void pmaxsan();
int co01;

for(co01 = 1;pnumb[0] > 21;++co01){
    if(pnumb[co01] == 0){
        break;}
    if(pnumb[co01] == 1){
        pnumb[co01] = 14;}
    pmaxsan();
}
}

void cadown(){ //=====
void cmaxsan();
int co01;

for(co01 = 1;cnumb[0] > 21;++co01){
    if(cnumb[co01] == 0){
        break;}
    if(cnumb[co01] == 1){
        cnumb[co01] = 14;}
    cmaxsan();
}
}
}

```

```

void CBJver100Dlg::OnBstand()
{
    // TODO: この位置にコントロール通知ハンドラ用のコードを追加してください
    //=====
    void thinkran(),cthink(),insuit(),innumb(),cadown();

    //-----
    // c 1 枚目公開
    CDC* pDC=m_land1.GetDC();
    CDC myDC;
    myDC.CreateCompatibleDC(pDC);
    // c 1 枚目スート表示
    CBitmap myCSUIT01;
    if(csuit[1] == 1){
        myCSUIT01.LoadBitmap(IDB_SSUITS);myDC.SelectObject(myCSUIT01);
        pDC->BitBlt(90,55,48,68,&myDC,0,0,SRCCOPY);}
    if(csuit[1] == 2){
        myCSUIT01.LoadBitmap(IDB_SSUITH);myDC.SelectObject(myCSUIT01);
        pDC->BitBlt(90,55,48,68,&myDC,0,0,SRCCOPY);}
    if(csuit[1] == 3){
        myCSUIT01.LoadBitmap(IDB_SSUITD);myDC.SelectObject(myCSUIT01);
        pDC->BitBlt(90,55,48,68,&myDC,0,0,SRCCOPY);}
    if(csuit[1] == 0){
        myCSUIT01.LoadBitmap(IDB_SSUITC);myDC.SelectObject(myCSUIT01);
        pDC->BitBlt(90,55,48,68,&myDC,0,0,SRCCOPY);}
    // c 1 枚目数字表示
    CBitmap myCNUMB01;
    if(cnumb[1] == 1){
        myCNUMB01.LoadBitmap(IDB_SNUMBA);myDC.SelectObject(myCNUMB01);
        pDC->BitBlt(90,78,48,23,&myDC,0,0,SRCCOPY);}
    if(cnumb[1] == 2){
        myCNUMB01.LoadBitmap(IDB_SNUMB2);myDC.SelectObject(myCNUMB01);
        pDC->BitBlt(90,78,48,23,&myDC,0,0,SRCCOPY);}
    if(cnumb[1] == 3){
        myCNUMB01.LoadBitmap(IDB_SNUMB3);myDC.SelectObject(myCNUMB01);
        pDC->BitBlt(90,78,48,23,&myDC,0,0,SRCCOPY);}
    if(cnumb[1] == 4){
        myCNUMB01.LoadBitmap(IDB_SNUMB4);myDC.SelectObject(myCNUMB01);
        pDC->BitBlt(90,78,48,23,&myDC,0,0,SRCCOPY);}
    if(cnumb[1] == 5){
        myCNUMB01.LoadBitmap(IDB_SNUMB5);myDC.SelectObject(myCNUMB01);
        pDC->BitBlt(90,78,48,23,&myDC,0,0,SRCCOPY);}
    if(cnumb[1] == 6){
        myCNUMB01.LoadBitmap(IDB_SNUMB6);myDC.SelectObject(myCNUMB01);
        pDC->BitBlt(90,78,48,23,&myDC,0,0,SRCCOPY);}
    if(cnumb[1] == 7){
        myCNUMB01.LoadBitmap(IDB_SNUMB7);myDC.SelectObject(myCNUMB01);
        pDC->BitBlt(90,78,48,23,&myDC,0,0,SRCCOPY);}
    if(cnumb[1] == 8){
        myCNUMB01.LoadBitmap(IDB_SNUMB8);myDC.SelectObject(myCNUMB01);
        pDC->BitBlt(90,78,48,23,&myDC,0,0,SRCCOPY);}
    if(cnumb[1] == 9){
        myCNUMB01.LoadBitmap(IDB_SNUMB9);myDC.SelectObject(myCNUMB01);
        pDC->BitBlt(90,78,48,23,&myDC,0,0,SRCCOPY);}
    if(cnumb[1] == 10){
        myCNUMB01.LoadBitmap(IDB_SNUMB10);myDC.SelectObject(myCNUMB01);
        pDC->BitBlt(90,78,48,23,&myDC,0,0,SRCCOPY);}
    if(cnumb[1] == 11){
        myCNUMB01.LoadBitmap(IDB_SNUMBJ);myDC.SelectObject(myCNUMB01);

```

```

    pDC->BitBlt(90,78,48,23,&myDC,0,0,SRCCOPY);}
if(cnumb[1] == 12){
    myCNUMB01.LoadBitmap(IDB_SNUMBQ);myDC.SelectObject(myCNUMB01);
    pDC->BitBlt(90,78,48,23,&myDC,0,0,SRCCOPY);}
if(cnumb[1] == 13){
    myCNUMB01.LoadBitmap(IDB_SNUMBK);myDC.SelectObject(myCNUMB01);
    pDC->BitBlt(90,78,48,23,&myDC,0,0,SRCCOPY);}
//-----
cisi = 1;
for(;cisi == 1;){
    //《処理》親思考
    thinkran();
    cthink();          //子関数pmaxsan,cmaxsan
    //-----
    //《処理》思考結果がカード引き可ならカード引き
    if(cisi == 1){
        ccard[ccardno] = yama[yamano];
        insuit();innumb();

        // c 手札 3 枚目以降スート表示
        CDC* pDC=m_land1.GetDC();
        CDC myDC;
        myDC.CreateCompatibleDC(pDC);
        CBitmap myCSUIT03;
        if(psuit[ccardno] == 1){
            myCSUIT03.LoadBitmap(IDB_SSUITS);myDC.SelectObject(myCSUIT03);
            pDC->BitBlt(90 + (55 * (ccardno - 1)),55,48,68,&myDC,0,0,SRCCO
PY);}
        if(psuit[ccardno] == 2){
            myCSUIT03.LoadBitmap(IDB_SSUITH);myDC.SelectObject(myCSUIT03);
            pDC->BitBlt(90 + (55 * (ccardno - 1)),55,48,68,&myDC,0,0,SRCCO
PY);}
        if(psuit[ccardno] == 3){
            myCSUIT03.LoadBitmap(IDB_SSUITD);myDC.SelectObject(myCSUIT03);
            pDC->BitBlt(90 + (55 * (ccardno - 1)),55,48,68,&myDC,0,0,SRCCO
PY);}
        if(psuit[ccardno] == 0){
            myCSUIT03.LoadBitmap(IDB_SSUITC);myDC.SelectObject(myCSUIT03);
            pDC->BitBlt(90 + (55 * (ccardno - 1)),55,48,68,&myDC,0,0,SRCCO
PY);}
        if(psuit[ccardno] != 1 && psuit[ccardno] != 2 && psuit[ccardno] !=
3 && psuit[ccardno] != 0){
            myCSUIT03.LoadBitmap(IDB_RCARD);myDC.SelectObject(myCSUIT03);
            pDC->BitBlt(90 + (55 * (ccardno - 1)),55,48,68,&myDC,0,0,SRCCO
PY);}

        // c 手札 3 枚目以降数字表示
        CBitmap myCNUMB03;
        if(cnumb[ccardno] == 1){
            myCNUMB03.LoadBitmap(IDB_SNUMBA);myDC.SelectObject(myCNUMB03);
            pDC->BitBlt(90 + (55 * (ccardno - 1)),78,48,23,&myDC,0,0,SRCCO
PY);}
        if(cnumb[ccardno] == 2){
            myCNUMB03.LoadBitmap(IDB_SNUMB2);myDC.SelectObject(myCNUMB03);
            pDC->BitBlt(90 + (55 * (ccardno - 1)),78,48,23,&myDC,0,0,SRCCO
PY);}
        if(cnumb[ccardno] == 3){
            myCNUMB03.LoadBitmap(IDB_SNUMB3);myDC.SelectObject(myCNUMB03);
            pDC->BitBlt(90 + (55 * (ccardno - 1)),78,48,23,&myDC,0,0,SRCCO
PY);}

```

```

    if(cnumb[ccardno] == 4){
        myCNUMB03.LoadBitmap(IDB_SNUMB4);myDC.SelectObject(myCNUMB03);
        pDC->BitBlt(90 + (55 * (ccardno - 1)),78,48,23,&myDC,0,0,SRCCO
PY);}
    if(cnumb[ccardno] == 5){
        myCNUMB03.LoadBitmap(IDB_SNUMB5);myDC.SelectObject(myCNUMB03);
        pDC->BitBlt(90 + (55 * (ccardno - 1)),78,48,23,&myDC,0,0,SRCCO
PY);}
    if(cnumb[ccardno] == 6){
        myCNUMB03.LoadBitmap(IDB_SNUMB6);myDC.SelectObject(myCNUMB03);
        pDC->BitBlt(90 + (55 * (ccardno - 1)),78,48,23,&myDC,0,0,SRCCO
PY);}
    if(cnumb[ccardno] == 7){
        myCNUMB03.LoadBitmap(IDB_SNUMB7);myDC.SelectObject(myCNUMB03);
        pDC->BitBlt(90 + (55 * (ccardno - 1)),78,48,23,&myDC,0,0,SRCCO
PY);}
    if(cnumb[ccardno] == 8){
        myCNUMB03.LoadBitmap(IDB_SNUMB8);myDC.SelectObject(myCNUMB03);
        pDC->BitBlt(90 + (55 * (ccardno - 1)),78,48,23,&myDC,0,0,SRCCO
PY);}
    if(cnumb[ccardno] == 9){
        myCNUMB03.LoadBitmap(IDB_SNUMB9);myDC.SelectObject(myCNUMB03);
        pDC->BitBlt(90 + (55 * (ccardno - 1)),78,48,23,&myDC,0,0,SRCCO
PY);}
    if(cnumb[ccardno] == 10){
        myCNUMB03.LoadBitmap(IDB_SNUMB10);myDC.SelectObject(myCNUMB03)
;
        pDC->BitBlt(90 + (55 * (ccardno - 1)),78,48,23,&myDC,0,0,SRCCO
PY);}
    if(cnumb[ccardno] == 11){
        myCNUMB03.LoadBitmap(IDB_SNUMBJ);myDC.SelectObject(myCNUMB03);
        pDC->BitBlt(90 + (55 * (ccardno - 1)),78,48,23,&myDC,0,0,SRCCO
PY);}
    if(cnumb[ccardno] == 12){
        myCNUMB03.LoadBitmap(IDB_SNUMBQ);myDC.SelectObject(myCNUMB03);
        pDC->BitBlt(90 + (55 * (ccardno - 1)),78,48,23,&myDC,0,0,SRCCO
PY);}
    if(cnumb[ccardno] == 13){
        myCNUMB03.LoadBitmap(IDB_SNUMBK);myDC.SelectObject(myCNUMB03);
        pDC->BitBlt(90 + (55 * (ccardno - 1)),78,48,23,&myDC,0,0,SRCCO
PY);}
    ++ccardno; ++yamano;
}
//-----
cmaxsan();
if(cnumb[0] > 21){cadown();} //子関数cmaxsan、バースト処理

char cmmaaxx[50]; //c
sprintf(cmmaaxx,"ディーラーの合計点は = %d",cnumb[0]); //c
m_ed3.SetWindowText(cmmaaxx); //c

//《処理》バースト判定(バーストなら終わり)
if(cnumb[0] >= 22){
    AfxMessageBox("バースト、貴方の勝ちです");
    cisi = 2;
    CDialog::OnOK();
}
}
//=====

```

```

// 《処理》判定
pmaxsan();cmaxsan();
if(cisi == 0){
    if(pnumb[0] > cnumb[0]){
        AfxMessageBox("貴方の勝ちです");}
    if(pnumb[0] == cnumb[0]){
        AfxMessageBox("引き分けです");}
    if(pnumb[0] < cnumb[0]){
        AfxMessageBox("貴方の負けです");}
}
CDialog::OnOK();
}

void cthink(){ //=====
void pmaxsan(),cmaxsan();
int sa,coa;

cisi = 0;
pmaxsan();cmaxsan();
sa = 21 - cnumb[0];
//-----
// 《処理》親の「A」の枚数カウント（思考に影響）
coa = 0;
//-----
if((ckettei[ccardno - 3] <= 0) || (ckettei[ccardno - 3] >= 101)){
    AfxMessageBox("!!警告!! 親思考確立数値異常");
}
if(cnumb[0] > pnumb[0]){cisi = 0;}
if(cnumb[0] < pnumb[0]){cisi = 1;}
if(cnumb[0] == pnumb[0]){
    if(sa == 1){ //2の位置 が、引く確立
        if(ckettei[ccardno - 3] <= 2){cisi = 1;}}
    if(sa == 2){
        if(ckettei[ccardno - 3] <= 3){cisi = 1;}}
    if(sa == 3){
        if(ckettei[ccardno - 3] <= 5){cisi = 1;}}
    if(sa == 4){
        if(ckettei[ccardno - 3] <= 10){cisi = 1;}}
    if(sa == 5){
        if(ckettei[ccardno - 3] <= 20){cisi = 1;}}
    if(sa == 6){
        if(ckettei[ccardno - 3] <= 50){cisi = 1;}}
    if(sa == 7){
        if(ckettei[ccardno - 3] <= 70){cisi = 1;}}
    if(sa == 8){
        if(ckettei[ccardno - 3] <= 85){cisi = 1;}}
    if(sa == 9){
        if(ckettei[ccardno - 3] <= 93){cisi = 1;}}
    if(sa == 10){
        if(ckettei[ccardno - 3] <= 98){cisi = 1;}}
    if(sa >= 11){
        cisi = 1;}
}
}

void thinkran(){ //=====
int rand();
int ranmax = 100,ranmin = 1;
int co01;

```

```
srand((unsigned int)time(0));  
rand();  
for(co01 = 0;co01 < (havemax - 1);++co01){  
    ckettei[co01] = ranmin + (int)(rand() * (ranmax - ranmin + 1) / (1 + R  
AND_MAX));  
}  
}
```