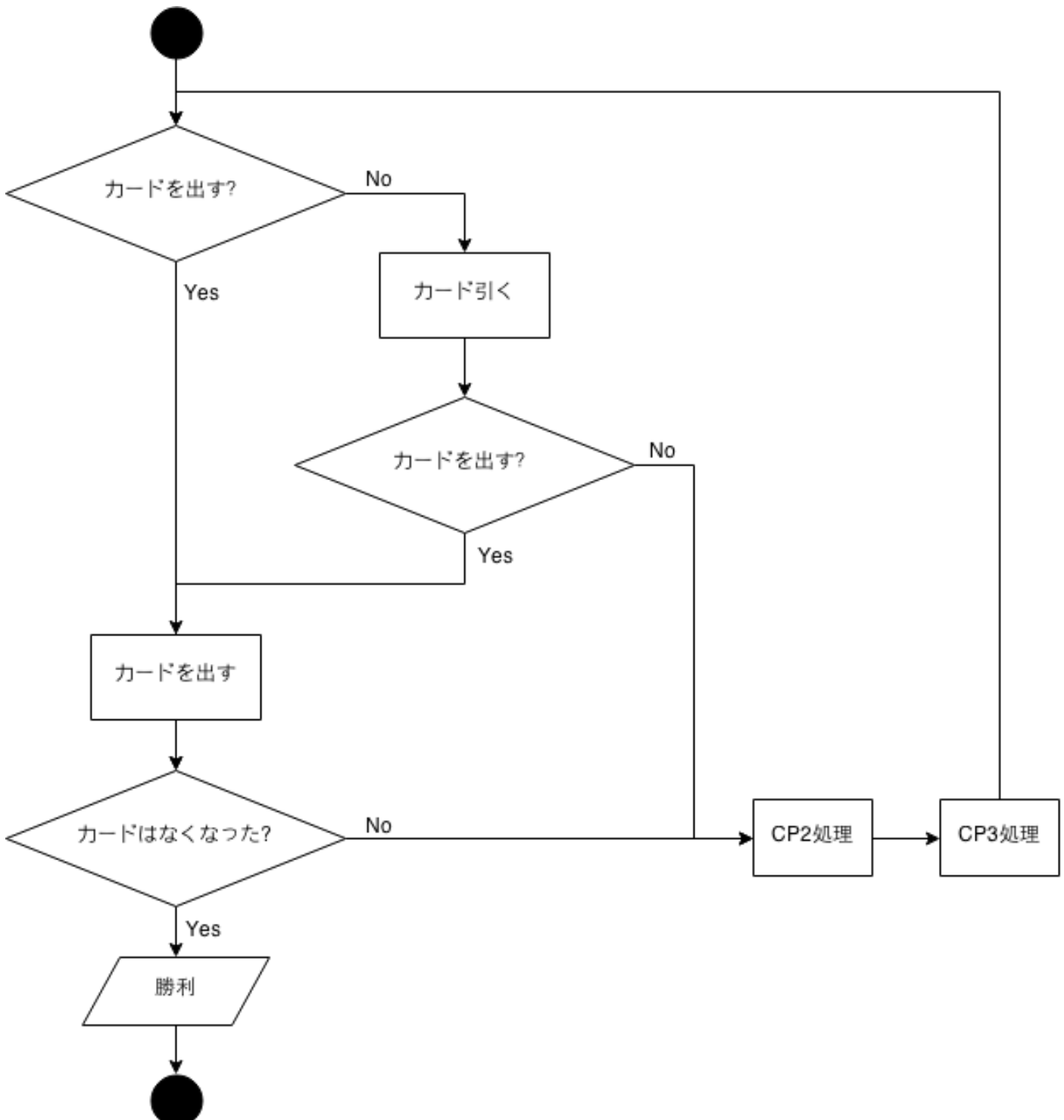


アメリカンページワン

【ゲーム説明】(Player vs Computers(コンピュータは自動))

UNO の元となったゲームである。各プレイヤーに 5 枚ずつカードを配る。全員に配り終わったら山札から一枚フィールドに置く(フィールドカード)。プレイヤーはフィールドカードと同じ数字、または同じスート(記号:ハートやクロバーなど)のカードを手札から出すか、カードを一枚引く。一枚引いた場合、再度カードを出す場合はカードをだし自分のターンを終了する。手札をなくした人がその回の勝者となる。勝者には、敗者の手札の点数を合計した点数が与えられる。敗者には、手札の点数分(カードの点数で計算)(マイナス)が与えられる。最後に点数が高かった人が勝者となる。

[勝利条件] (1 ゲーム)最初に手札を 0 にしたものが勝利, (総合勝利):点数が多い人の勝利



[役判定]

名称	カード	効果
ツー	2	次のプレイヤーは、2を出せない場合、カードデッキから2枚引かなくてはならない。 2を出した場合、次のプレイヤーのターンになる。 以降のプレイヤーは2が重なった枚数×2枚のカードを引かなくてはならない。
スリー	3	次のプレイヤーは、3を出せない場合、カードデッキから3枚引かなくてはならない。 3を出した場合、次のプレイヤーのターンになる。 以降のプレイヤーは3が重なった枚数×3枚のカードを引かなくてはならない。
エイト	8	フィールドカードの記号がどんなカードでもフィールドに出すことができる。 このカードを出した場合フィールドカードのスイートを変更することができる。
リバース	9	ゲーム進行の順番が逆になる
スキップ	A	Playerを1つ飛ばす

[カードの点数]

カード	点数(点)
A, 2, 3, 8, 9	20
10, J, Q, K	10
4, 5, 6, 7	数字の通り

【使用方法】

ゲームを何回やるか指定

①手札からカード出すか出さないかを入力(出す場合：手札の何枚目,出さない場合:0)

出すことができない場合、「そのカードは出すことができません」と出力され、再度入力画面に移行する。

0を入力した場合、カードを手札に1枚加え、手札からカードを出すか出さないかの入力画面に移行する。(出す場合：手札の何枚目,出さない場合:0)

そのプレイヤーのターンが終了。

次のプレイヤーになり、上記と同じ事を繰り返す。

手札がなくなれば勝者

指定した回数ゲームを行っていない場合①に戻る

指定した回数ゲームを行った場合得点を採点し、一番得点が高い人が優勝

[役がでた場合]

ツー、スリー：次の Player は出すことのできるカードが2のみになる。出せない場合はツーがでた回数*2枚引く。

スリー：次の Player は出すことのできるカードが3のみになる。出せない場合はスリーがでた回数*3枚引く。

エイト：スイート入力画面に移り、自分が変えたいスイートに変えることができる。(SPADE:0, HEART:1, DIA:2, CLUB:3)に対応している。

リバース:Player1→2→3 という順番が逆になる。再度出された場合は元に戻る。

スキップ：スキップを出した Player の次の Player が飛ばされ、飛ばされた次のプレイヤーの番になる。

[変数]

グローバル変数

int idx player 判定

reversecard リーバス判定

color 記号変えカードの記号を格納

drawcard ドローカードのどの種類が出ているか判定

recount 何回ドローカードがでたか判定

drawjudge ドローカードがでたか判定

Player players[4]; Player1〜3

Player field; //フィールドカード

ローカル(main)変数

vector<int> carddeck; 山札格納

vector<int> carddeck2 出したカード格納

cardidx 出すカードのインデックス格納

nots ドローカードの枚数を計算

tehuda 手札のサイズ格納

playcount :プレイする回数格納

idxkeep idx 格納

carddecksize カードデッキのサイズ格納

[関数の説明]

1. void create_carddeck(vector<int> &carddeck);

山札を作成する関数である。1〜56の数で14で割った余りが0以外の数字をcarddeck(山札)として代入する。14で割った解を数字に、余りをスートに割り当てるためである。

2. unsigned int Random (unsigned int max);

乱数を発生させる関数。maxまでの値を返す。

3. void draw(Player &players, vector<int> &carddeck);

carddeckからplayersにカードを一枚与える。与えたカードをカードデッキから削除

void print_struct(Player, const string [], const string []);

Playrのカードを数字と記号に変えて出力

void print_fieldtehuda(Player &, Player &, const string [], const string []);

手札とフィールドカードとplayer番号を出力

int action_judge(Player);

カードを出すか引くかを判定。カードを出す場合はカードのインデックス、ドローする場合は0を入力。入力された値を返す

int put_judge(Player, Player, int, int);

カードが出せるのか出せないのかを判定。出せる場合は1をそれ以外は0を返す

void put_action(Player [], Player &, vector<int> &, vector<int> &, const string [], const string [], int);

出したカードが8の場合、idx==0(自分)の時にどの記号に変えるか入力、それ以外はランダムで記号を入力、そして出したカードを入力された記号に変えフィールドカードに格納し、手札からそのカードを消す。8以外の場合は出したカードをフィールドにだし、手札から消す

void card_judge(Player &, Player &, int);

カードの役判定。出したカードが9の時reversecardに1を代入し逆回りにする。2の場合はdrawbardに2を代入し2を出したということを記憶させる。何回出したかという判定のためにrecount+=1をする。出したカードが3の場合は2の処理を3に変えたものである。

void r_judge();

Player更新関数である。リバーズではない時idxを1増やす。それ以外は-1をして逆回りにする。

void message(Player, Player, const string [], const string [], int);

処理後のメッセージを出力する関数である。

それぞれの処理後に出すメッセージが書かれている。

bool tfsort(const Card &left, const Card &right);

手札のソート関数である。数字の降順に並び替え、数字が同じ時はスートの降順に並べる。

int pscore(Player, int);

手札のスコア換算関数である。10,J,Q,Kが10、2,3,8,9,Aが20、それ以外は数字通りの数を返す。

void midle_message(Player[], int, int, int);

中間結果を出力する関数である。勝者の手札に敗者の手札の点数を合計した点数が与えられ、敗者は自分の手札のマイナスの点数が与えられる。そしてplayerの手札をクリアし初期化する。最後にPlayerごとに点数を出力する。

void last_message(Player[], int, int);

最終結果を出力する関数である。Player 事の手札を換算し、各プレイヤーの点数を出力する。最も点数が高かつ

た **player** を求めてそのプレイヤーが優勝者と出力する。