

# C 言語：神経衰弱（Flip Cards）

○×工業高校 機械工学科 2 年

2023 年 10 月 11 日

## 1 ゲームの概要

カードを 2 枚開いて、一致すれば得点となり、不一致なら再度伏せて相手の手番になる。

```
% ./flipcard
```

```
[ a ][ b ][ c ][ d ][ e ]  
[ f ][ g ][ h ][ i ][ j ]  
[ k ][ l ][ m ][ n ][ o ]  
[ p ][ q ][ r ][ s ][ t ]
```

A さん：0 点 B さん：0 点 = A さんの番です =

Select 1st card : a

```
[ 4 ][ b ][ c ][ d ][ e ]  
[ f ][ g ][ h ][ i ][ j ]  
[ k ][ l ][ m ][ n ][ o ]  
[ p ][ q ][ r ][ s ][ t ]
```

Select 2nd card : p

```
[ 4 ][ b ][ c ][ d ][ e ]  
[ f ][ g ][ h ][ i ][ j ]  
[ k ][ l ][ m ][ n ][ o ]  
[ 5 ][ q ][ r ][ s ][ t ]
```

ハズレ

```
[ a ][ b ][ c ][ d ][ e ]  
[ f ][ g ][ h ][ i ][ j ]  
[ k ][ l ][ m ][ n ][ o ]  
[ p ][ q ][ r ][ s ][ t ]
```

A さん：0 点 B さん：0 点 = B さんの番です =

Select 1st card : e

```
[ a ][ b ][ c ][ d ][ 4 ]
```

[ f ][ g ][ h ][ i ][ j ]  
[ k ][ l ][ m ][ n ][ o ]  
[ p ][ q ][ r ][ s ][ t ]

Select 2nd card : a

[ 4 ][ b ][ c ][ d ][ 4 ]  
[ f ][ g ][ h ][ i ][ j ]  
[ k ][ l ][ m ][ n ][ o ]  
[ p ][ q ][ r ][ s ][ t ]

当たり

[ 4 ][ b ][ c ][ d ][ 4 ]  
[ f ][ g ][ h ][ i ][ j ]  
[ k ][ l ][ m ][ n ][ o ]  
[ p ][ q ][ r ][ s ][ t ]

A さん : 0 点 B さん : 1 点 = B さんの番です =  
Select 1st card : t

[ 4 ][ b ][ c ][ d ][ 4 ]  
[ f ][ g ][ h ][ i ][ j ]  
[ k ][ l ][ m ][ n ][ o ]  
[ p ][ q ][ r ][ s ][ 0 ]

Select 2nd card : q

[ 4 ][ b ][ c ][ d ][ 4 ]  
[ f ][ g ][ h ][ i ][ j ]  
[ k ][ l ][ m ][ n ][ o ]  
[ p ][ 9 ][ r ][ s ][ 0 ]

ハズレ

[ 4 ][ b ][ c ][ d ][ 4 ]  
[ f ][ g ][ h ][ i ][ j ]  
[ k ][ l ][ m ][ n ][ o ]  
[ p ][ q ][ r ][ s ][ t ]

A さん : 0 点 B さん : 1 点 = A さんの番です =  
Select 1st card :

## 2 主処理 (main)

```
1  int main( void ){
2      int cnum1, cnum2, Closed=ROW*COLUMN;
3      int pointA=0, pointB=0;
4      init();
5      disp();
6      do{
7          printf(" Aさん : %d点 \t Bさん : %d点 \n", pointA, pointB);
8          if (Turn)
9              printf("= Aさんの番です \n");
10         else
11             printf("= Bさんの番です \n");
12         cnum1 = getNum("1st");
13         cnum2 = getNum("2nd");
14         if ( match(cnum1, cnum2) ){
15             printf("当たり\n");
16             Closed -= 2;
17             if (Turn)
18                 pointA++;
19             else
20                 pointB++;
21         } else {
22             printf("ハズレ\n");
23             closeCard(cnum1);
24             closeCard(cnum2);
25             Turn = !Turn; /* 手番の交代 */
26         }
27         sleep(4); /* 開いたカードを見せておく時間 */
28         disp();
29     } while ( 0 < Closed );
30
31     return 0;
32 }
```

## 3 盤面の初期化 (init)

プログラム中で乱数を使うので、時刻を乱数の種に指定して、プログラムを起動するたびに異なるカード配置になるようにしている

```
1  void init(){
2      srand((unsigned)time(NULL));
3      int i, j, num;
4      for (i = 0; i < ROW/2; i++){
5          for (j = 0; j < COLUMN; j++){
6              num = i * COLUMN + j;
7              cards[num].num = cards[ROW * COLUMN - num].num = num;
8              cards[num].row = cards[ROW * COLUMN - num].row = i;
9              cards[num].clm = cards[ROW * COLUMN - num].clm = j;
10             cards[num].opn = cards[ROW * COLUMN - num].opn = false;
11         }
12     }
```

```

12     }
13     shuffle();
14 }

```

## 4 カードのシャフル (shuffle, swapCards)

乱数を使ってカード配置を入れ替えている

```

1 void swapCards(int n1, int n2){
2     struct Card temp;
3     temp = cards[n2];
4     cards[n2] = cards[n1];
5     cards[n1] = temp;
6 }
7 void shuffle(){
8     int sel, min = 0, max = ROW*COLUMN;
9     while (0 < max){
10         sel = (rand() % (max - min + 1)) + min;
11         swapCards(sel, max--);
12     }
13 }

```

## 5 盤面の表示 (disp)

```

1 void disp(){
2     int i, j, n;
3     printf("\n");
4     for (i = 0; i < ROW; i++){
5         for (j = 0; j < COLUMN; j++){
6             n = i * COLUMN + j;
7             if ( !cards[n].opn )
8                 printf("[_%c_]", 'a' + i * COLUMN + j );
9             else
10                 printf("[_%d_]", cards[i * COLUMN + j].num );
11         }
12         printf("\n");
13     }
14     printf("\n");
15 }

```

## 6 入力 (getNum)

```

1 int getNum(char* s){
2     int cnum;
3     char str[24], work[24];
4     do {
5         sprintf(work, "%s%s%s", "\tSelect_", s, "_card_:");
6         printf("%s", work);

```

```

7         scanf("%s", str);
8         cnum = str[0] - 'a';
9     } while (cnum > ROW * COLUMN);
10    openCard(cnum);
11    disp();
12    return cnum;
13 }

```

## 7 カードの開閉 (openCard, closeCard)

```

1 void openCard(int n){
2     cards[n].opn = true;
3 }
4 void closeCard(int n){
5     cards[n].opn = false;
6 }

```

## 8 一致不一致の判定 (match)

```

1 enum BOOLEAN match(int n1, int n2){
2     if ( cards[n1].num == cards[n2].num )
3         return true;
4     return false;
5 }

```

## 9 各種宣言など

これは冒頭に記述する

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  #include <unistd.h>
5  // 定数
6  #define ROW 4
7  #define COLUMN 5
8  // 構造体
9  enum BOOLEAN {
10     false, /* false = 0, true = 1 */
11     true
12 };
13 struct Card {
14     int num;
15     int row;
16     int clm;
17     enum BOOLEAN opn;
18 };
19 // function prototypes

```

```
20      /*
21      void openCard(int);
22      void closeCard(int);
23      void switchTurn(enum BOOLEAN);
24      void disp();
25      int getNum(char*);
26      enum BOOLEAN match(int, int);
27      void wasteTime(int);
28      void swapCards(int, int);
29      void shuffle();
30      void init();
31      */
32      static struct Card cards[20]; /* 20 = ROW * COLUMN */
33      enum BOOLEAN Turn = true;
```

## 10 プログラムの全体

ソースコード 1 神経衰弱

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  #include <unistd.h>
5
6  // 定数
7  #define ROW 4
8  #define COLUMN 5
9  // 構造体
10 enum BOOLEAN {
11     false, /* false = 0, true = 1 */
12     true
13 };
14 struct Card {
15     int num;
16     int row;
17     int clm;
18     enum BOOLEAN opn;
19 };
20 // function prototypes
21 /*
22 void openCard(int);
23 void closeCard(int);
24 void switchTurn(enum BOOLEAN);
25 void disp();
26 int getNum(char*);
27 enum BOOLEAN match(int, int);
28 void wasteTime(int);
29 void swapCards(int, int);
30 void shuffle();
31 void init();
32 */
33
34 static struct Card cards[20]; /* 20 = ROW * COLUMN */
35
36 enum BOOLEAN Turn = true;
37
38 // サブプログラム
39 void openCard(int n){
40     cards[n].opn = true;
41 }
42 /* ===== */
43 void closeCard(int n){
44     cards[n].opn = false;
45 }
46 /* ===== */
47 void disp(){
48     int i, j;
49     printf("\n");
50     for (i = 0; i < ROW; i++){
51         for (j = 0; j < COLUMN; j++){
52             int n = i * COLUMN + j;
```

```

53         if ( !cards[n].opn )
54             printf("[_c_]", 'a' + i * COLUMN + j );
55         else
56             printf("[_d_]", cards[i * COLUMN + j].num );
57     }
58     printf("\n");
59 }
60 printf("\n");
61 }
62 /* ===== */
63 int getNum(char* s){
64     int cnum;
65     char str[24], work[24];
66     do {
67         sprintf(work, "%s%s", "\tSelect_", s, "_card_:");
68         printf("%s", work);
69         scanf("%s", str);
70         cnum = str[0] - 'a';
71     } while (cnum > ROW * COLUMN);
72     openCard(cnum);
73     disp();
74     return cnum;
75 }
76 /* ===== */
77 enum BOOLEAN match(int n1, int n2){
78     if ( cards[n1].num == cards[n2].num )
79         return true;
80     return false;
81 }
82 /* ===== */
83 void swapCards(int n1, int n2){
84     struct Card temp;
85     temp = cards[n2];
86     cards[n2] = cards[n1];
87     cards[n1] = temp;
88 }
89 /* ===== */
90 void shuffle(){
91     int min = 0;
92     int max = ROW * COLUMN;
93     while (0 < max){
94         int sel = (rand() % (max - min + 1)) + min;
95         swapCards(sel, max--);
96     }
97 }
98 /* ===== */
99 void init(){
100     srand((unsigned)time(NULL));
101     int i, j;
102     for (i = 0; i < ROW/2; i++){
103         for (j = 0; j < COLUMN; j++){
104             int num = i * COLUMN + j;
105             cards[num].num = cards[ROW * COLUMN - num].num = num;
106             cards[num].row = cards[ROW * COLUMN - num].row = i;
107             cards[num].clm = cards[ROW * COLUMN - num].clm = j;
108             cards[num].opn = cards[ROW * COLUMN - num].opn = false;
109         }
110     }

```



```

111     shuffle();
112 }
113 // メイン
114 int main( void ){
115     int cnum1, cnum2, Closed=ROW*COLUMN;
116     int pointA=0, pointB=0;
117     init();
118     disp();
119     do{
120         printf("Aさん：%d点\tBさん：%d点\n", pointA, pointB);
121         if (Turn)
122             printf("= Aさんの番です\n");
123         else
124             printf("= Bさんの番です\n");
125         cnum1 = getNum("1st");
126         cnum2 = getNum("2nd");
127         if ( match(cnum1, cnum2) ){
128             printf("当たり\n");
129             Closed -= 2;
130             if (Turn)
131                 pointA++;
132             else
133                 pointB++;
134         } else {
135             printf("ハズレ\n");
136             closeCard(cnum1);
137             closeCard(cnum2);
138             Turn = !Turn;
139         }
140         sleep(4);
141         disp();
142     } while ( 0 < Closed );
143     return 0;
144 }

```