

C 言語：スライド・パズル (15Puzzle)

富山工業高校 機械工学科 2 年

2023 年 10 月 11 日

1 ゲームの概要

4 × 4 に区切った盤面上の各タイルに、0～15 の番号が割り振られている。0 が割り振られたタイルは空欄になっていて、他のタイルはその空欄にスライドさせて移動することができる。最初は不規則に並べられている盤面ですが、空欄の上下、あるいは空欄の左右のタイルを選んで、空欄の方向にスライドさせる事によって、最終的に 1 から 15 まで規則正しく並んだ盤面状態を目指すゲームである。

```
% ./slidetile
```

```
[11][ 3][ 6][10]
[ 9][ 1][ 8][14]
[ 2][ 4][ 7][12]
[13][15][ ][ 5]
```

```
Select number:6
```

```
[11][ 3][ ][10]
[ 9][ 1][ 6][14]
[ 2][ 4][ 8][12]
[13][15][ 7][ 5]
```

```
Select number:11
```

```
[ ][11][ 3][10]
[ 9][ 1][ 6][14]
[ 2][ 4][ 8][12]
[13][15][ 7][ 5]
```

```
Select number:11
```

```
[11][ ][ 3][10]
[ 9][ 1][ 6][14]
[ 2][ 4][ 8][12]
[13][15][ 7][ 5]
```

```
Select number:
```

2 主処理 (main)

```
1  int main(void) {
2      int sel;
3      init();
4      do {
5          disp();
6          do {
7              printf("\nSelect number:");
8              scanf("%d", &sel);
9          } while( sel>ROW*COLUMN );
10         moveTile(sel);
11     } while( check() );
12
13     return 0;
14 }
```

3 盤面の初期化 (init)

乱数を使ってタイルをシャフルするので、時刻を乱数の種に指定することによって、プログラムを起動する度に、異なるタイル配置になる様になっている

```
1  void init(){
2      int i, j, n;
3      srand((unsigned)time(NULL));
4      for (i = 0; i < ROW; i++){
5          for (j = 0; j < COLUMN; j++){
6              n = i * COLUMN + j;
7              tiles[n].num = n;
8              tiles[n].row = i;
9              tiles[n].clm = j;
10         }
11     }
12     shuffle(1000);
13 }
```

4 シャッフル (shuffle)

乱数を使って、タイルを不規則に選択し移動させている

```
1  void shuffle(int n){
2      int sel, min = 0, max = ROW*COLUMN;
3      do{
4          sel = (rand() % (max - min + 1)) + min;
5          moveTile(sel);
6          n--;
7      } while (0 < n);
8  }
```

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 (tiles[n].num == 0){
8                 printf("[ ]");
9             } else {
10                printf("[%2d]", tiles[i * COLUMN + j].num);
11            }
12        }
13        printf("\n");
14    }
15 }
```

6 タイルのスライド (moveTile)

```
1 int findTileNum(int num){
2     int i, j;
3     for (i = 0; i < ROW * COLUMN; i++){
4         if (tiles[i].num == num){
5             return i;
6         }
7     }
8     return -1;
9 }
10 void swapTile(int n1, int n2){
11     int tmp = tiles[n1].num;
12     tiles[n1].num = tiles[n2].num;
13     tiles[n2].num = tmp;
14 }
15 void swapTileR(int c, int r1, int r2){
16     int n1 = r1 * COLUMN + c;
17     int n2 = r2 * COLUMN + c;
18     swapTile(n1, n2);
19 }
20 void swapTileC(int r, int c1, int c2){
21     int n1 = r * COLUMN + c1;
22     int n2 = r * COLUMN + c2;
23     swapTile(n1, n2);
24 }
25 void moveTile(int sel){
26     int i, j;
27     int s = findTileNum(sel);
28     int sr = tiles[s].row;
29     int sc = tiles[s].clm;
30     int z = findTileNum(0);
31     int zr = tiles[z].row;
```

```

32     int zc = tiles[z].clm;
33     if (sr == zr){
34         if (sc < zc){
35             for (j = zc; j > sc; j--){
36                 if (0 <= j - 1){
37                     swapTileC(sr, j, j - 1);
38                 }
39             }
40         } else if (zc < sc) {
41             for (j = zc; j < sc; j++){
42                 if (j + 1 < COLUMN){
43                     swapTileC(sr, j, j + 1);
44                 }
45             }
46         }
47     }
48     if (sc == zc){
49         if (sr < zr){
50             for (j = zr; j > sr; j--){
51                 if (0 <= j - 1){
52                     swapTileR(sc, j, j - 1);
53                 }
54             }
55         } else if (zr < sr){
56             for (j = zr; j < sr; j++){
57                 if (j + 1 < ROW){
58                     swapTileR(sc, j, j + 1);
59                 }
60             }
61         }
62     }
63 }

```

7 完成チェック (check)

```

1     enum BOOLEAN check(){
2         int i, j, n;
3         for (i = 0; i < ROW; i++){
4             for (j = 0; j < COLUMN; j++){
5                 n = i * COLUMN + j;
6                 if (tiles[n].num != n){
7                     return true;
8                 }
9             }
10        }
11        return false;
12    }

```

8 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
5      // 定数
6      #define ROW 4
7      #define COLUMN 4
8      // 構造体
9      struct Tile{
10         int num;
11         int row;
12         int clm;
13     };
14     enum BOOLEAN{
15         false, /* false = 0, true = 1 */
16         true
17     };
18     // function prototypes
19     /*
20     int findTileNum(int);
21     void swapTile(int,int);
22     void swapTileR(int,int,int);
23     void swapTileC(int,int,int);
24     void moveTile(int);
25     void shuffle(int);
26     enum BOOLEAN check();
27     void disp();
28     void init();
29     */
30
31     static struct Tile tiles[16];    /* 16 = ROW * COLUMN */
```

9 プログラムの全体

ソースコード 1 スライド・パズル

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  // 定数
6  #define ROW 4
7  #define COLUMN 4
8  // 構造体
9  struct Tile{
10     int num;
11     int row;
12     int clm;
13 };
14 enum BOOLEAN{
15     false, /* false = 0, true = 1 */
16     true
17 };
18 // function prototypes
19 /*
20 int findTileNum(int);
21 void swapTile(int,int);
22 void swapTileR(int,int,int);
23 void swapTileC(int,int,int);
24 void moveTile(int);
25 void shuffle(int);
26 enum BOOLEAN check();
27 void disp();
28 void init();
29 */
30
31 static struct Tile tiles[16]; /* 16 = ROW * COLUMN */
32
33 // サブプログラム
34 int findTileNum(int num){
35     int i;
36     for (i = 0; i < ROW * COLUMN; i++){
37         if (tiles[i].num == num){
38             return i;
39         }
40     }
41     return -1;
42 }
43 /* ===== */
44 void swapTile(int n1, int n2){
45     int tmp = tiles[n1].num;
46     tiles[n1].num = tiles[n2].num;
47     tiles[n2].num = tmp;
48 }
49 /* ===== */
50 void swapTileR(int c, int r1, int r2){
51     int n1 = r1 * COLUMN + c;
52     int n2 = r2 * COLUMN + c;
```

```

53     swapTile(n1, n2);
54 }
55 /* ===== */
56 void swapTileC(int r, int c1, int c2){
57     int n1 = r * COLUMN + c1;
58     int n2 = r * COLUMN + c2;
59     swapTile(n1, n2);
60 }
61 /* ===== */
62 void moveTile(int sel){
63     int j;
64     int s = findTileNum(sel);
65     int sr = tiles[s].row;
66     int sc = tiles[s].clm;
67     int z = findTileNum(0);
68     int zr = tiles[z].row;
69     int zc = tiles[z].clm;
70     if (sr == zr){
71         if (sc < zc){
72             for (j = zc; j > sc; j--){
73                 if (0 <= j - 1){
74                     swapTileC(sr, j, j - 1);
75                 }
76             }
77         } else if (zc < sc) {
78             for (j = zc; j < sc; j++){
79                 if (j + 1 < COLUMN){
80                     swapTileC(sr, j, j + 1);
81                 }
82             }
83         }
84     }
85     if (sc == zc){
86         if (sr < zr){
87             for (j = zr; j > sr; j--){
88                 if (0 <= j - 1){
89                     swapTileR(sc, j, j - 1);
90                 }
91             }
92         } else if (zr < sr){
93             for (j = zr; j < sr; j++){
94                 if (j + 1 < ROW){
95                     swapTileR(sc, j, j + 1);
96                 }
97             }
98         }
99     }
100 }
101 /* ===== */
102 void shuffle(int n){
103     int sel, min = 0, max = ROW*COLUMN;
104     do{
105         sel = (rand() % (max - min + 1)) + min;
106         moveTile(sel);
107         n--;
108     } while (0 < n);
109 }
110 /* ===== */

```



```

111 enum BOOLEAN check(){
112     int i, j, n;
113     for (i = 0; i < ROW; i++){
114         for (j = 0; j < COLUMN; j++){
115             n = i * COLUMN + j;
116             if (tiles[n].num != n){
117                 return true;
118             }
119         }
120     }
121     return false;
122 }
123 /* ===== */
124 void disp(){
125     int i, j, n;
126     printf("\n");
127     for (i = 0; i < ROW; i++){
128         for (j = 0; j < COLUMN; j++){
129             n = i * COLUMN + j;
130             if (tiles[n].num == 0){
131                 printf("[  ]");
132             } else {
133                 printf("[%2d]", tiles[i * COLUMN + j].num);
134             }
135         }
136         printf("\n");
137     }
138 }
139 /* ===== */
140 void init(){
141     srand((unsigned)time(NULL));
142     int i, j, n;
143     for (i = 0; i < ROW; i++){
144         for (j = 0; j < COLUMN; j++){
145             n = i * COLUMN + j;
146             tiles[n].num = n;
147             tiles[n].row = i;
148             tiles[n].clm = j;
149         }
150     }
151     shuffle(1000);
152 }
153 // メイン
154 int main(void) {
155     int sel;
156     init();
157     do {
158         disp();
159         do {
160             printf("\nSelect number:");
161             scanf("%d", &sel);
162         } while( sel>ROW*COLUMN );
163         moveTile(sel);
164     } while( check() );
165
166     return 0;
167 }

```