```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <windows.h>

extern "C"
{
#include "Game15Entities.h"
#include "Game15GUI.h"
#include "Game15Util.h"
}

int main()
{
Game15Field field = create_field();
int end = 0;

display_logo();
system("pause");

while(!(end = is_end_game(field)))
{
system("cls");
display_game_field(field);

char move = (char)_getch();

switch (move)
{
case 'w': make_turn_game(&field, UP); break;
case 's': make_turn_game(&field, DOWN); break;
case 'a': make_turn_game(&field, LEFT); break;
case 'd': make_turn_game(&field, RIGHT); break;
case 'q': goto exit; break;
case 'r': shuffle_field(&field);
}
}
if(end)
{
display_win();
}
exit: system("pause");
return 0;
}
```

```
#ifndef GAME15_GAME15_H
#define GAME15_GAME15_H

#endif //GAME15_GAME15_H
```

```
 #ifndef GAME15_GAME15ENTITIES_H
 #define GAME15_GAME15ENTITIES_H
 #define FIELD_WIDTH 4
 #define FIELD_HEIGHT 4
 typedef struct Game15Field
 {
 int cells[FIELD_WIDTH][FIELD_HEIGHT];
 } Game15Field;
 typedef enum Game15Direction
 {
 UP, DOWN, LEFT, RIGHT
 } Game15Direction;
#endif //GAME15_GAME15ENTITIES_H
```

```
#include "Game15GUI.h"
void display_game_field(Game15Field field)
{
```

```
printf("X-------------------X\n");
printf("| 15 GAME |\n");
printf("X---- ---- ---- ----X\n");
for(int i=0;i<FIELD_WIDTH;i++)
{
for(int j=0;j<FIELD_HEIGHT;j++)
{
if(field.cells[i][j] != 0)
{
printf("| %2d ", field.cells[i][j]);
}
else
{
printf("| ");
}
}
printf("|\n");
printf("X---- ---- ---- ----X\n");
}
printf("| Manual : |\n");
printf("|-------------------|\n");
printf("| w - UP |\n");
printf("| s - DOWN |\n");
printf("| a - LEFT |\n");
printf("| d - RIGHT |\n");
printf("| q - EXIT |\n");
printf("| r - RESTART |\n");
printf("X---- ---- ---- ----X\n");
}

void display_logo()
{
printf("X----------------------------------------------------------------------------------X\n");
printf("X X\n");
printf("X ** ***** ******** * * * ***** X\n");
printf("X * * * * * * * * * * * X\n");
printf("X * * ***** * ***** * * * * * * ***** X\n");
printf("X * * * * ******* * * * * * X\n");
printf("X **** ***** ******** * * * * * ***** X\n");
printf("X X\n");
printf("X----------------------------------------------------------------------------X\n");
}

void display_win()
{
printf("X----------------------------------------------------------------------------X\n");
printf("X X\n");
printf("X * * ***** * * X\n");
printf("X * * * * * * * X\n");
printf("X * * * * * * * * X\n");
printf("X * * * * * * * * X\n");
printf("X * * ***** * * X\n");
printf("X X\n");
printf("X----------------------------------------------------------------------------X\n");
}


#ifndef GAME15_GAME15GUI_H
#define GAME15_GAME15GUI_H

#include "stdio.h"
#include "Game15Entities.h"

void display_game_field(Game15Field field);

void display_logo();

void display_win();

#endif //GAME15_GAME15GUI_H


#include "Game15Util.h"
```

```
void swap_values(int* a, int* b)
{
int buffer = *a;
*a = *b;
*b = buffer;
}

void shuffle_field(Game15Field* field)
{
for (int i = 0; i < FIELD_HEIGHT; ++i)
{
for (int j = 0; j < FIELD_WIDTH; ++j)
{
swap_values(&field->cells[i][j],
&field->cells[rand() % FIELD_HEIGHT][rand() % FIELD_WIDTH]);
}
}
}

Game15Field create_field()
{
Game15Field field;
int counter = 0;

for (int i=0;i<FIELD_HEIGHT;i++)
{
for(int j=0;j<FIELD_WIDTH;j++)
{
field.cells[i][j] = counter++;
}
}

shuffle_field(&field);

return field;
}

void make_turn_game(Game15Field* field, Game15Direction direction)
```

```
{
int emptyCellI = -1;
int emptyCellJ = -1;

for (int i=0;i<FIELD_HEIGHT;i++)
{
for(int j=0;j<FIELD_WIDTH;j++)
{
if(field->cells[i][j] == 0)
{
emptyCellI = i;
emptyCellJ = j;
break;
}
}
}

if(emptyCellI == -1 || emptyCellJ == -1)
return;

switch (direction)
{
case UP: {
if(emptyCellI < FIELD_HEIGHT - 1)
{
swap_values(&field->cells[emptyCellI][emptyCellJ],
&field->cells[emptyCellI+1][emptyCellJ]);
}
break;
}
case DOWN: {
if(emptyCellI > 0)
{
swap_values(&field->cells[emptyCellI][emptyCellJ],
&field->cells[emptyCellI-1][emptyCellJ]);
}
```

```
break;
}
case LEFT: {
if(emptyCellJ < FIELD_WIDTH - 1)
{
swap_values(&field->cells[emptyCellI][emptyCellJ],
&field->cells[emptyCellI][emptyCellJ+1]);
}
break;
}
case RIGHT: {
if(emptyCellJ > 0)
{
swap_values(&field->cells[emptyCellI][emptyCellJ],
&field->cells[emptyCellI][emptyCellJ-1]);
}
break;
}
}
}

int is_end_game(Game15Field field)
{
int counter = 1;

if(field.cells[FIELD_HEIGHT-1][FIELD_WIDTH-1] != 0)
return 0;

for(int i=0;i<FIELD_HEIGHT;i++)
{
for(int j=0;j<FIELD_WIDTH;j++)
{
if(i != FIELD_HEIGHT - 1 || j != FIELD_WIDTH - 1)
{
if(counter++ != field.cells[i][j])
return 0;
}
}
}
return 1;
}


#ifndef GAME15_GAME15UTIL_H
#define GAME15_GAME15UTIL_H

#include "Game15Entities.h"
#include <stdlib.h>
#include <time.h>

void swap_values(int* a, int* b);

void shuffle_field(Game15Field* field);

Game15Field create_field();

void make_turn_game(Game15Field* field, Game15Direction direction);

int is_end_game(Game15Field field);

#endif //GAME15_GAME15UTIL_H
```