

2-5 俄罗斯方块之按键控制方块_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 2-5 俄罗斯方块之按键控制方块涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

```
#include <time.h>
time_t time(time_t *tloc);
功能：获得从1970年到现在所经过的秒数。
```

一般用法：
time_t t = time(NULL);

```
void srandom (unsigned int seed);
功能：srandom设置产生一系列伪随机数发生器的起始点，一般常用时间作为起始点。
生成随机数种子。
参数：
@seed 随机数产生器的初始值(种子值)。
```

```
long int random();
功能：会随机生成一个位于 0 ~ RAND_MAX 之间的整数。
RAND_MAX 是 <stdlib.h> 头文件中的一个宏，它用来指明 rand() 所能返回的随机数的最大值。
不同的系统RAND_MAX的值不同。linux中它的值非常的大，大于是 16 * ((2^31) -1)。
```

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int i = 0, k = 0;

    srandom((unsigned int)time(NULL));

    for(i = 0; i < 10; i++)
    {
        k = random() % 200;
        printf("k = %d\n", k);
    }

    return 0;
}

int n_x = 6;
int n_y = 6;
int n_num;
int n_mode;
int n_color;

void init_shape()
{
    srandom(time(NULL));
    n_num = random() % 7;
    n_mode = random() % 4;
    n_color = random() % 7 + 40;

    print_mode_shape(n_num, n_mode, n_x, n_y, n_color);

    fflush(NULL);
}
```

```

void eraser_shape(int n,int m,int a,int b)
{
    int i = 0;
    int xx = a;
    int yy = b;

    for(i = 0;i < 16;i++){

        if(i != 0 && i%4 == 0){
            yy++;
            xx = a;
        }
        if(shape[n][m][i] == 1){
            printf("\33[%d;%dH  \33[0m",yy,xx);
        }

        xx += 2;
    }
    fflush(NULL);
}

```

```

int change_shape()
{
    int m = (n_mode+1)%4;

    eraser_shape(n_num,n_mode,n_x,n_y);
    n_mode = m;
    print_mode_shape(n_num,n_mode,n_x,n_y,n_color);
}

```

```

int move_left(int n,int m)
{
    eraser_shape(n,m,n_x,n_y);
    n_x -= 2;
    print_mode_shape(n,m,n_x,n_y,n_color);

    return 0;
}

```

```

int move_right(int n,int m)
{
    eraser_shape(n,m,n_x,n_y);
    n_x += 2;
    print_mode_shape(n,m,n_x,n_y,n_color);

    return 0;
}

```

```

int move_down(int n,int m)
{
    eraser_shape(n,m,n_x,n_y);
    n_y++;
    print_mode_shape(n,m,n_x,n_y,n_color);
    return 0;
}

```

```

#include <stdio.h>
#include <sys/time.h>
#include <stdlib.h>
#include <signal.h>
#include <termios.h>
#include <unistd.h>
#include <time.h>

```

```

int shape[7][4][18] =
{
    {
        {1,1,0,0, 1,1,0,0, 0,0,0,0, 0,0,0,0, 2,2},
        {1,1,0,0, 1,1,0,0, 0,0,0,0, 0,0,0,0, 2,2},
        {1,1,0,0, 1,1,0,0, 0,0,0,0, 0,0,0,0, 2,2},
        {1,1,0,0, 1,1,0,0, 0,0,0,0, 0,0,0,0, 2,2},
    },
    {
        {1,0,0,0, 1,0,0,0, 1,0,0,0, 1,0,0,0, 3,0},

```

```

        {1,1,1,1, 0,0,0,0, 0,0,0,0, 0,0,0,0, 0,3},
        {1,0,0,0, 1,0,0,0, 1,0,0,0, 1,0,0,0, 3,0},
        {1,1,1,1, 0,0,0,0, 0,0,0,0, 0,0,0,0, 0,3},
    },
    {
        {0,1,0,0, 1,1,1,0, 0,0,0,0, 0,0,0,0, 1,2},
        {1,0,0,0, 1,1,0,0, 1,0,0,0, 0,0,0,0, 2,1},
        {1,1,1,0, 0,1,0,0, 0,0,0,0, 0,0,0,0, 1,2},
        {0,1,0,0, 1,1,0,0, 0,1,0,0, 0,0,0,0, 2,1}
    },
    {
        {1,1,0,0, 0,1,1,0, 0,0,0,0, 0,0,0,0, 1,2},
        {0,1,0,0, 1,1,0,0, 1,0,0,0, 0,0,0,0, 2,1},
        {1,1,0,0, 0,1,1,0, 0,0,0,0, 0,0,0,0, 1,2},
        {0,1,0,0, 1,1,0,0, 1,0,0,0, 0,0,0,0, 2,1},
    },
    {
        {0,1,1,0, 1,1,0,0, 0,0,0,0, 0,0,0,0, 1,2},
        {1,0,0,0, 1,1,0,0, 0,1,0,0, 0,0,0,0, 2,1},
        {0,1,1,0, 1,1,0,0, 0,0,0,0, 0,0,0,0, 1,2},
        {1,0,0,0, 1,1,0,0, 0,1,0,0, 0,0,0,0, 2,1},
    },
    {
        {0,0,1,0, 1,1,1,0, 0,0,0,0, 0,0,0,0, 1,2},
        {1,0,0,0, 1,0,0,0, 1,1,0,0, 0,0,0,0, 2,1},
        {1,1,1,0, 1,0,0,0, 0,0,0,0, 0,0,0,0, 1,2},
        {1,1,0,0, 0,1,0,0, 0,1,0,0, 0,0,0,0, 2,1}
    },
    {
        {1,0,0,0, 1,1,1,0, 0,0,0,0, 0,0,0,0, 1,2},
        {1,1,0,0, 1,0,0,0, 1,0,0,0, 0,0,0,0, 2,1},
        {1,1,1,0, 0,0,1,0, 0,0,0,0, 0,0,0,0, 1,2},
        {0,1,0,0, 0,1,0,0, 1,1,0,0, 0,0,0,0, 2,1}},
};

int n_x = 6;
int n_y = 6;
int n_num,n_mode,n_color;

int getch()
{
    struct termios tm,tm_old;

    tcgetattr(0,&tm_old);

    cfmakeraw(&tm);

    tcsetattr(0,0,&tm);

    int ch = getchar();

    tcsetattr(0,0,&tm_old);

    return ch;
}

void print_mode_shape(int n,int m,int x,int y,int c)
{
    int i = 0;
    int xx = x;
    int yy = y;
    for(;i < 16;i++)
    {
        if(i != 0 && i%4 == 0)
        {
            yy += 1;
            xx = x;
        }

        if(shape[n][m][i] == 1){
            printf("\033[%d;%dH\033[%dm[]\033[0m",yy,xx,c);
        }
        xx += 2;
    }
    fflush(NULL);
}

void init_shape()
{

```

```

srand(time(NULL));
n_num = random()%7;
n_mode = random()%4;
n_color = random()%7+40;

print_mode_shape(n_num,n_mode,n_x,n_y,n_color);

fflush(NULL);
}

void eraser_shape(int n,int m,int a,int b)
{
    int i = 0;
    int xx = a;
    int yy = b;

    for(i = 0;i < 16;i++){
        if(i != 0 && i%4 == 0){
            yy++;
            xx = a;
        }
        if(shape[n][m][i] == 1){
            printf("\033[%d;%dH \033[0m",yy,xx);
        }
        xx += 2;
    }
    fflush(NULL);
}

int change_shape()
{
    int m = (n_mode+1)%4;

    eraser_shape(n_num,n_mode,n_x,n_y);
    n_mode = m;
    print_mode_shape(n_num,n_mode,n_x,n_y,n_color);
}

int move_left(int n,int m)
{
    eraser_shape(n,m,n_x,n_y);
    n_x -= 2;
    print_mode_shape(n,m,n_x,n_y,n_color);

    return 0;
}

int move_right(int n,int m)
{
    eraser_shape(n,m,n_x,n_y);
    n_x += 2;
    print_mode_shape(n,m,n_x,n_y,n_color);

    return 0;
}

int move_down(int n,int m)
{
    eraser_shape(n,m,n_x,n_y);
    n_y++;
    print_mode_shape(n,m,n_x,n_y,n_color);
    return 0;
}

void key_control()
{
    int ch;
    while(1){
        ch = getch();
        if(ch == 'q' || ch == 'Q'){
            break;
        }else if(ch == '\r'){
            printf("down\n");
        }else if(ch == '\033'){
            ch = getch();
            if(ch == '['){
                ch = getch();
            }
        }
    }
}

```

```
        switch(ch){
            case 'A':
                change_shape();
                break;
            case 'B':
                move_down(n_num,n_mode);
                break;
            case 'C':
                move_right(n_num,n_mode);
                break;
            case 'D':
                move_left(n_num,n_mode);
                break;
            default:
                break;
        }
    }
}

int main()
{
    printf("\033[2J");
    printf("\033[?25l");
    init_shape();
    key_control();
    printf("\033[?25h");
    return 0;
}
```

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，[点击查看详细说明](#)

