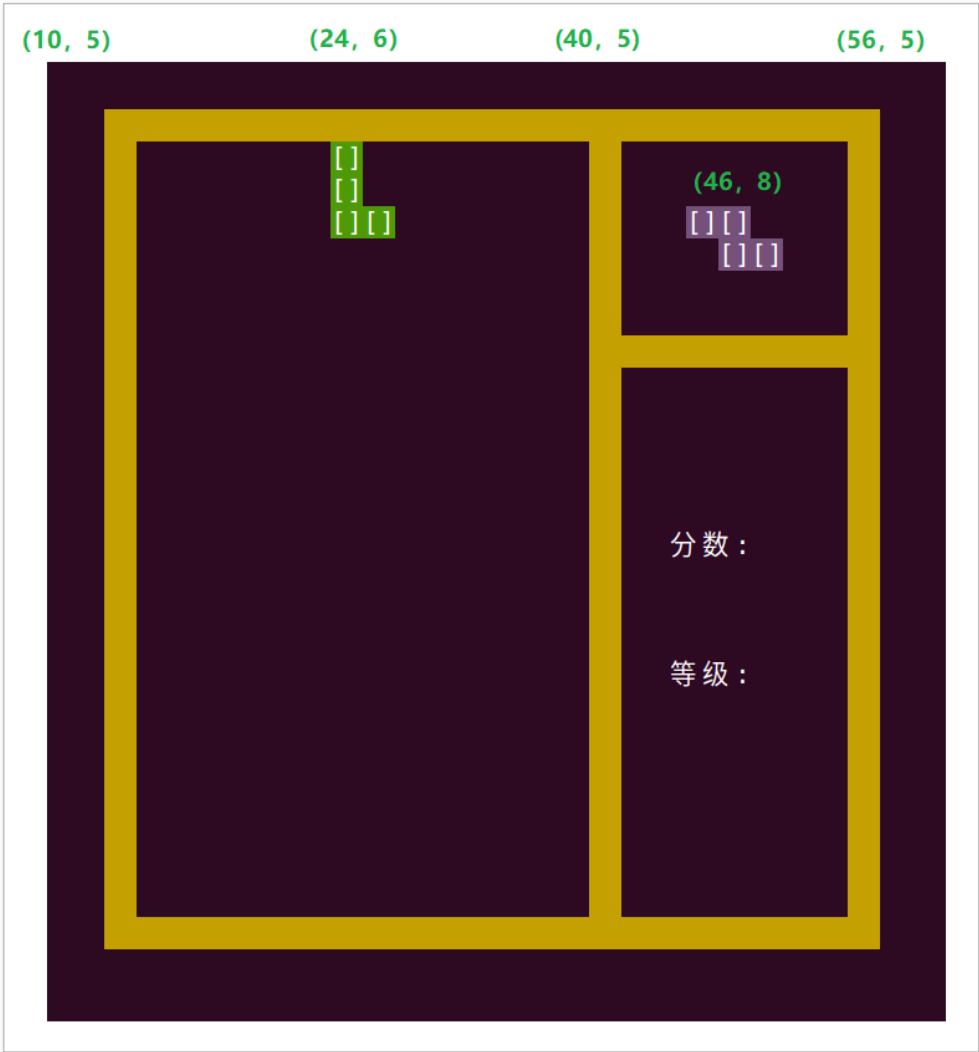


# 3.2 图形存储及输出设计\_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 3.2 图形存储及输出设计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。

```
int shape[7][4][18];

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},
    }
};
```



user\_global.c

```
#include "user_print.h"

int next_num = 0;
int next_mode = 0;
int next_color = 0;

int init_x = 24;
int init_y = 6;

int next_x = 46;
int next_y = 8;

int dynamic_x = 0;
int dynamic_y = 0;

int dynamic_num = 0;
int dynamic_mode = 0;
int dynamic_color = 0;

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}},
};

```

user\_print.h

```

#ifndef _USER_PRINT_H_
#define _USER_PRINT_H_

extern int next_num;
extern int next_mode;
extern int next_color;

extern int next_x;
extern int next_y;

extern int init_x;
extern int init_y;

extern int dynamic_x;
extern int dynamic_y;

extern int dynamic_num;
extern int dynamic_mode;
extern int dynamic_color;

extern int shape[7][4][18];

extern void print_mode_shape(int n,int m,int x,int y,int c);
extern void print_next_shape();
extern void erase_last_shape(int n,int m,int a,int b);
#endif

```

user\_print.c

```

#include <stdio.h>
#include <sys/time.h>
#include <stdlib.h>
#include <signal.h>
#include "user_print.h"

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 = 0; 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 erase_last_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);
}

```

```

void print_next_shape()
{
    erase_last_shape(next_num, next_mode, next_x, next_y);

    next_num = random()%7;
    next_mode = random()%4;
    next_color = random()%7 + 40;

    print_mode_shape(next_num, next_mode, next_x, next_y, next_color);

    fflush(NULL);
}

```

main.c

```

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

int score_x = 45;
int score_y = 18;
int level_x = 45;
int level_y = 22;

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_start_ui()
{

    printf("\33[2J");
    int i;

    for(i = 0; i < 47; i++){
        printf("\33[%d;%dH\33[43m \33[0m",5,i+10);
        printf("\33[%d;%dH\33[43m \33[0m",30,i+10);
    }

    for(i = 0; i < 26; i++){
        printf("\33[%d;%dH\33[43m \33[0m",i+5,10);
        printf("\33[%d;%dH\33[43m \33[0m",i+5,40);
        printf("\33[%d;%dH\33[43m \33[0m",
                i+5,56);
    }

    for(i=0; i < 17; i++){
        printf("\33[%d;%dH\33[43m \33[0m",12,40+i);
    }


    printf("\33[%d;%dH分数:\33[0m",score_y,score_x);

    printf("\33[%d;%dH等级:\33[0m",level_y,level_x);

    fflush(NULL);
}

void init_game_ui()
{

    print_start_ui();

    getch();


    srand(time(NULL));

    dynamic_num = random()%7;
    dynamic_mode = random()%4;
    dynamic_color = random()%7+40;

    dynamic_x = init_x;
    dynamic_y = init_y;


    print_mode_shape(dynamic_num,dynamic_mode,dynamic_x,dynamic_y,dynamic_color);

    print_next_shape();
    printf("\33[?25l");
}

int main()
{
    init_game_ui();
    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

