目录

[一、 设计题目 - 1 -](#_Toc343725199)

[二、 题目介绍 - 1 -](#_Toc343725200)

[三、 系统功能模块划分 - 1 -](#_Toc343725201)

[四、 数据结构介绍 - 2 -](#_Toc343725202)

[五、 各模块具体设计 - 2 -](#_Toc343725203)

[六、 程序设计环境 - 2 -](#_Toc343725204)

[七、 试验结果 - 2 -](#_Toc343725205)

[八、 体会 - 6 -](#_Toc343725206)

[九、 参考文献 - 6 -](#_Toc343725207)

[十、 源代码 - 6 -](#_Toc343725208)

# 设计题目

Windows下C语言控制台打字游戏。

# 题目介绍

使用C语言实现一个控制台的打字游戏，使用了windows的API对控制台的输出进行控制;使用定时器实现单词的下降;

游戏形式：单词从窗口上方以一定的速度匀速下落，使用键盘输入字符，正确的字符将使单词变短，直到消失。每隔一定的时间有新的单词产生，单词到达最下一行，游戏失败，回到主界面。中间可以输入快捷键退出到主界面。

游戏特色：可以设置游戏的显示颜色，单词的下落速度，单词的来源文件，等。通过修改单词文件，不仅可以让我们练习打字速度，同时也可以用来记忆单词。

# 系统功能模块划分

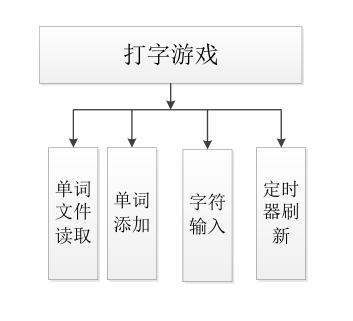


图 三‑1 系统框图

程序使用了定时器，每隔一个时间周期会添加一个单词，同时刷新屏幕内容；当前屏幕的单词保存在一个结构体的数组中，每次刷新屏幕的时间会读取单词的位置和修改单词的位置属性；

当前输入的单词用一个结构体变量specailOne保存；输入字符c的时候查找当前的单词变量specailOne是否为空，如果为空，查找单词数组中以c为开头的单词，如果找到，则加入specailOne；如果specailOne不为空，则比对c和specailOne中单词的首字符，如果相同，则删除specailOne中单词的首字符。

# 数据结构介绍

1. 单词库数组

`file.c:

char word[MAX\_WORD\_COUNT][MAX\_WORD\_LENGHT]；

保存着当前的单词库，通过读取文件，或者默认的方式初始化。每次添加单词的时候，从这个单词数组中获取单词。

1. 当前屏幕单词数组

typedef struct \_words

{

char str[MAX\_WORD\_LENGHT];

COORD pos;

}words;

words head[MAX\_WORD\_COUNT];

int total = 0;

结构体words保存单词的值str和坐标位置pos; head是一个一维的数组，total记录当前屏幕上的单词数目，单词顺序保存在数组head中，每当删除一个单词的时候，后面的单词向前移，这样保证前面的单词连续，且按时间顺序排列。

# 各模块具体设计

1. 主函数



图 五‑1 主函数

主函数在一个循环中，调用初始化程序Init(),然后运行游戏Run(),从游戏退出后，调用收发函数Quit()；循环进入下一次游戏循环。游戏通过Run()中调用exit(0)，函数退出。

1. 初始化



图 五‑2 Init函数

1. 游戏函数Run



图 五‑3 run函数

1. 退出函数



图 五‑4 Quit函数

1. 定时函数



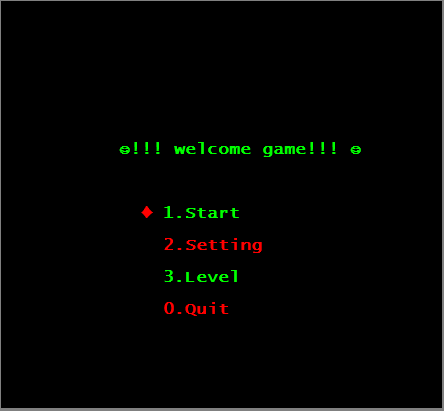
图 五‑5 refresh 设定定时器

# 程序设计环境

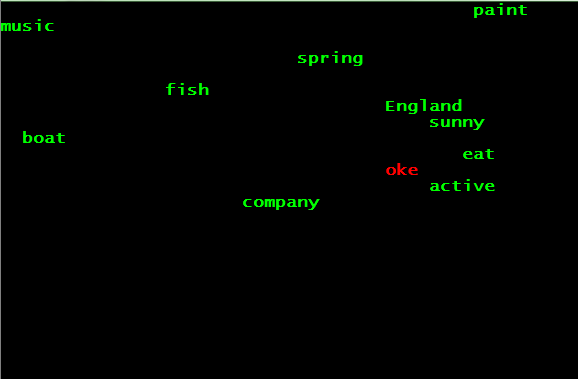
1. 系统： Windows 7 / Windows XP
2. 编译环境：VC++6.0

# 试验结果

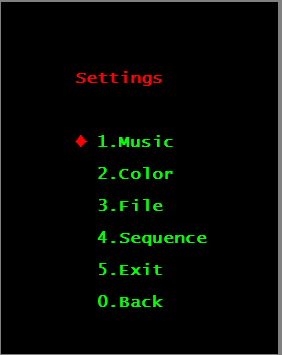
1. 主菜单



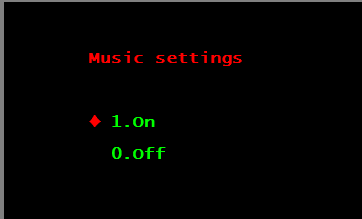
1. 游戏界面



1. 设置界面



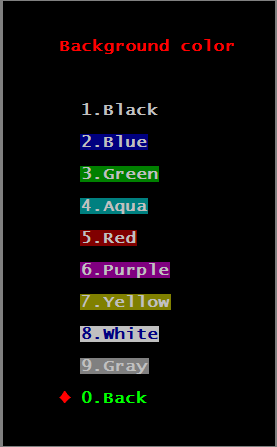
1. 声音设置界面



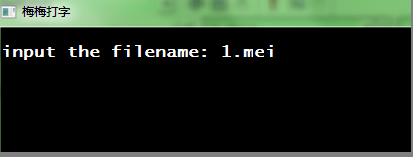
1. 前景色设置界面



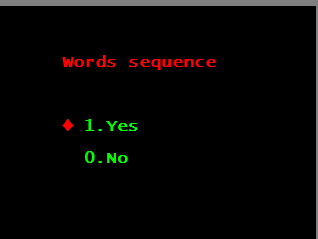
1. 背景色设置界面



1. 文件名输入界面



1. 单词产生顺序设置界面：有序，还是无序



# 体会

# 参考文献

1. 控制台窗口界面编程控制: http://www.yesky.com/20020913/1630252.shtml
2. MSDN: <http://msdn.microsoft.com/en-us/library/windows/desktop/ms682073(v=vs.85).aspx>
3. VC 帮助文档。

# 源代码

1. 主函数

#include "dazi.h"

int main(int argc, char \*argv[])

{

while (1) {

Init();

Run();

Quit();

}

return 0;

}

1. File 文件

//file.h

#ifndef FILE\_H

#define FILE\_H

#include "words.h"

#define MAX\_PATH\_LENGHT 124

//读取文件初始化单词表

int setFile();

//设置单词文件的名字

int setFilename(char \*path);

extern char word[MAX\_WORD\_COUNT][MAX\_WORD\_LENGHT];

extern int total\_word;

#endif

//file.c

#include <stdio.h>

#include <string.h>

#include "file.h"

char filename[MAX\_PATH\_LENGHT] = "S:\\sources\\meixian\\1.mei";

char word[MAX\_WORD\_COUNT][MAX\_WORD\_LENGHT]={

"pen", "pencil", "pencil", "ruler", "eraser", "crayon", "book", "bag", "school",

"sharpener", "hello", "hi", "my", "name", "goodbay", "bye", "what",

"is", "your", "you", "head", "face", "nose", "mouth", "eye",

"ear", "arm", "hand", "finger", "leg", "foot", "food", "body",

"good", "morning", "this", "nice", "meet", "go", "to", "afternoon",

"too", "red", "yellow", "green", "blue", "purple", "white", "black",

"orange", "pink", "brown", "how", "are", "fine", "thank", "thanks",

"paint", "great", "cat", "dog", "monkey", "panda", "rabbit", "duck",

"pig", "birdb", "bear", "mouse", "elephant", "squirrel", "have", "super",

"zoo", "really", "may", "sure", "like", "it", "cake", "bread",

"hot", "hamburger", "chicken", "French", "French", "Coke", "cock", "juice",

"milk", "water", "tea", "coffee", "some", "and", "no", "can",

"one", "two", "three", "four", "five", "six", "seven", "eight",

"nine", "ten", "doll", "boat", "ship", "ball", "kite", "balloon",

"car", "plane", "friend", "for", "happy", "birthday", "how", "eat",

"how", "how", "gift", "boy", "girl", "teacher", "student", "this",

"my", "friend", "nice", "meet", "welcome", "back", "new", "where",

"you", "your", "America", "see", "say", "again", "class", "we",

"today", "watchout", "sorry", "ant", "apple", "Come", "egg", "father",

"dad", "mother", "mom", "man", "woman", "grandmother", "grandma", "grandfather",

"grandpa", "sister", "brother", "great", "really", "and", "how", "family",

"come", "who", "that", "she", "he", "fish", "funny", "fun",

"watch", "TV", "watch", "big", "ice", "ice", "eleven", "twelve",

"thirteen", "fourteen", "fifteen", "sixteen", "seventeen", "eighteen", "nineteen", "twenty",

"can", "look", "fly", "draw", "picture", "beautiful", "jeep", "jump",

"kangaroo", "key", "guess", "open", "right", "lion", "lock", "night",

"nest", "peach", "orange", "apple", "banana", "strawberry", "grape", "like",

"hungry", "please", "them", "very", "taste", "queen", "quiet", "fruit",

"certainly", "rainbow", "rain", "snake", "tiger", "taxi", "bus", "bike",

"desk", "chair", "walkman", "lamp", "play", "game", "here", "toy",

"box", "Under", "On", "umbrella", "look", "violin", "vest", "window",

"wind", "excuse", "no", "here", "zebra", "use", "us", "yo",

"small", "big", "long", "short", "tall", "giraffe", "deer", "dear",

"cute", "lovely", "at", "feed", "animal", "children", "so", "has",

"tail", "CAN", "PRC", "UK", "USA", "M", "P", "board",

"light", "picture", "door", "floor", "classroom", "many", "our", "classmate",

"have", "seat", "near", "new", "where", "computer", "teacher", "wall",

"fan", "clean", "good", "all", "good", "Chinese", "English", "math",

"schoolbag", "story", "notebook", "colour", "fat", "may", "sure", "here",

"twenty", "thirty", "thirty", "forty", "forty", "fifty", "too", "heavy",

"sorry", "long", "short", "thin", "strong", "quiet", "friend", "Chinese",

"photo", "music", "science", "sports", "painting", "computer", "her", "right",

"study", "bathroom", "bedroom", "living", "kitchen", "fish", "room", "open",

"close", "phone", "sofa", "shelf", "fridge", "rice", "noodle", "soup",

"vegetable", "beef", "dinner", "wait", "hungry", "knife", "chopstick", "spoon",

"plate", "plane", "fork", "pass", "ready", "try", "helpyourself", "show",

"yummy", "use", "family", "parents", "uncle", "aunt", "baby", "people",

"member", "only", "puppy", "baseball", "driver", "doctor", "farm", "farmer",

"nurse", "garden", "teacher", "library", "canteen", "first", "visitor", "there",

"lunch", "this", "that", "art", "computer", "washroom", "music", "gym",

"TV", "second", "cool", "English", "music", "breakfast", "P", "over",

"go", "just", "go", "kid", "run", "get", "go", "go",

"ready", "hurry", "math", "Chinese", "china", "England", "English", "P",

"class", "music", "for", "jacket", "shirt", "T", "skirt", "dress",

"sweater", "whose", "no", "not", "jeans", "pants", "parents", "socks",

"shoes", "shorts", "these", "those", "what", "neighbour", "colour", "warm",

"cold", "cool", "hot", "weather", "weather", "wear", "football", "put",

"wear", "rain", "rainy", "now", "snow", "snowy", "wind", "windy",

"cloud", "cloudy", "sun", "sunny", "not", "New", "matter", "have",

"colourful", "pretty", "cheap", "expensive", "sixty", "seventy", "hundred", "how",

"ninety", "sneakers", "slippers", "sandals", "boots", "want", "pair", "a",

"son", "sun", "size", "take", "them", "sheep", "horse", "hen",

"lamb", "goat", "cow", "donkey", "key", "rabbit", "pig", "duck",

"tomoto", "cucumber", "potato", "onion", "carrot", "fresh", "there", "young",

"funny", "tall", "strong", "kind", "old", "who", "Mr", "from",

"Canada", "what", "like", "know", "now", "no", "principal", "university",

"strict", "Miss", "smart", "active", "Lady", "so", "fun", "quiet",

"she", "he", "her", "his", "very", "but", "Monday", "Tuesday",

"Wednesday", "Thursday", "Friday", "Moral", "day", "Social", "wait", "waiter",

"tomorrow", "have", "on", "do", "watch", "read", "Saturday", "Sunday",

"sometimes", "usually", "love", "what", "yeah", "play", "too", "to",

"do", "cabbage", "pork", "mutton", "eggplant", "egg", "plant", "tofu",

"fish", "green", "potato", "tomato", "for", "lunch", "we", "menu",

"sound", "healthy", "tasty", "sweet", "adj", "sour", "fresh", "salty",

"they", "favourite", "fruit", "grape", "have", "eat", "empty", "cook",

"water", "sweep", "clean", "ill", "helpful", "at", "wash", "just",

"do", "make", "robot", "set", "wash", "do", "put", "I",

"would", "have", "play", "use", "air", "curtain", "trash", "closet",

"close", "mirror", "end", "look", "own", "flat", "third", "bedroom",

"kitchen", "bathroom", "living", "very", "in", "on", "under", "near",

"behind", "over", "in", "clothes", "work", "tell", "cloud", "mountain",

"river", "flower", "grass", "lake", "forest", "path", "park", "nature",

"farm", "farmer", "holiday", "picture", "village", "city", "house", "bridge",

"tree", "road", "building", "clean", "air", "run", "any", "do",

"eat", "have", "play", "eat", "when", "about", "policeman", "work",

"evening", "get", "at", "usually", "noon", "tell", "climbing", "go",

"play", "visit", "go", "weekend", "week", "end", "often", "sometimes",

"rain", "either", "next", "spring", "summer", "fall", "winter", "season",

"which", "best", "always", "play", "snow", "leaf", "up", "north",

"Halloween", "Thanksgiving", "Swim", "Fly", "skate", "make", "plant", "why",

"because", "sleep", "January", "February", "March", "April", "May", "June",

"July", "August", "September", "October", "November", "December", "chart", "birthday",

"uncle", "first", "second", "third", "fourth", "fifth", "eighth", "ninth",

"twelfth", "twentieth", "send", "e", "her", "able", "everyone", "then",

"date", "draw", "cook", "read", "answer", "talk", "mom", "Children",

"see", "listen", "clean", "write", "write", "grandpa", "speak", "study",

"fly", "jump", "walk", "run", "swim", "kangaroo", "truck", "sleep",

"climb", "fight", "swing", "drink", "climber", "take", "watch", "pick",

"do", "ant", "catch", "woods", "interesting", "have", "honey", "thing",

"count", "collect", "write", "play", "have", "him", "leave", "us",

"over", "by", "fifth", "right", "china", "English", "England", "Australia",

"left", "hospital", "cinema", "bookstore", "science", "museum", "excuse", "where",

"please", "next", "far", "supermarket", "bank", "after", "Want", "a",

"shoe", "get", "minute", "north", "south", "east", "west", "turn",

"right", "left", "straight", "then", "twelfth", "party", "tell", "start",

"take", "look", "next", "this", "This", "This", "Tonight", "tomorrow",

"take", "read", "Go", "theme", "the", "busy", "together", "comic",

"Post", "newspaper", "magazine", "dictionary", "shoe", "buy", "fruit", "pet",

"need", "plant", "else", "shop", "hobby", "ride", "riding", "dive",

"diving", "play", "make", "making", "collect", "collecting", "Show", "pen",

"dear", "twin", "look", "something", "must", "fun", "with", "TV",

"live", "lives", "lives", "teach", "teaches", "go", "goes", "watch",

"watches", "read", "reades", "does", "doesn", "different", "week", "say",

"soon", "excited", "sing", "singer", "Write", "Writer", "Work", "Worker",

"farm", "farmer", "teach", "teacher", "actor", "actress", "artist", "TV",

"Show", "n", "Hong", "engineer", "accountant", "police", "policeman", "policewoman",

"salesperson", "clean", "cleaner", "company", "where", "factory", "design", "tip",

"help", "money", "enjoy", "tourist", "way", "motor", "rain", "cloud",

"vapour", "sun", "stream", "come", "shine", "become", "little", "drop",

"wake", "feel", "think", "meet", "high", "other", "fall", "dwon",

"into", "come", "again", "seed", "Soil", "Sprout", "Plant", "Should",

"then", "garden", "easy", "put", "several", "day", "see", "pot",

"lovely", "make", "get", "get", "month", "old", "still", "hardly",

"come", "taller", "shorter", "stronger", "older", "younger", "cm", "than",

"little", "tail", "think", "fun", "funny", "funnier", "kg"

};

int total\_word = MAX\_WORD\_COUNT;

//读取文件初始化单词表

int setFile()

{

FILE \*fp;

int i;

char str[MAX\_WORD\_LENGHT];

fp = fopen(filename, "r");

if (fp == NULL)

{

return -1;

}

for(i=0;(fscanf(fp, "%s", str)) != EOF;i++) {

if (i>=MAX\_WORD\_COUNT)

{

break;

}

strcpy(word[i], str);

}

total\_word = i;

fclose(fp);

return 0;

}

//设置单词文件的名字

int setFilename(char \*path)

{

strcpy(filename, path);

return setFile();

}

1. Words

//words.h

#ifndef WORDS\_H

#define WORDS\_H

#include <windows.h>

#define MAX\_WORD\_LENGHT 20

#define MAX\_WORD\_COUNT 1000

typedef struct \_words

{

char str[MAX\_WORD\_LENGHT];

COORD pos;

}words;

//添加一个单词到单词链

int addWord();

//删除一个字符

int deleteChar(char c);

//判断是否有这个单词

int hasWord(char str);

//现有的单词数

int wordLength();

//所有单词下落一行

void downWords();

//设置窗口缓冲区

void setWindowsBuffer(char \*buffer);

//显示现有单词，调试用

void showWords();

//设置取词是否有序

void setSeqence(int flag);

//现有单词总数

extern int total;

#endif

//words.c

#include <stdlib.h>

#include <string.h>

#include "words.h"

#include "dazi.h"

#include "file.h"

//单词头

words head[MAX\_WORD\_COUNT];

int total = 0;

//单词取得是否有序

int sequenceFlag = 0;

//设置取词是否有序

void setSeqence(int flag)

{

sequenceFlag = flag;

}

//设置str的值

void setString(char \*str)

{

static int num;

if (sequenceFlag)

{

num = (num+1) % total\_word;

} else {

num = rand() % total\_word;

}

strcpy(str, word[num]);

}

//生成一个新的单词结点

void setWord(words \*word)

{

words \*temp = word;

setString(temp->str);

temp->pos.Y = 0;

temp->pos.X = (rand() % (WIDTH-strlen(temp->str)));

temp->pos.Y -= 1;

}

//添加一个单词到单词链

int addWord()

{

if (total >= MAX\_WORD\_COUNT)

{

return -1;

}

setWord(&head[total]);

total++;

return 0;

}

//删除一个单词

int deleteWord(char c)

{

int i;

for (i=0;i<total;i++)

{

if (c == head[i].str[0])

break;

}

if (i>=total) //没有

{

return -1;

}

//放到当前 的输入单词中

strcpy(specailOne.str, head[i].str + 1);

specailOne.pos = head[i].pos;

specailOne.pos.X += 1;

for (;i<total-1;i++)

{

head[i] =head[i+1];

}

total--;

return 0;

}

//判断是否有这个字符开头的单词，返回位置

int hasWord(char str)

{

int i;

for (i=0;i<total;i++)

{

if (head[i].str[0]==str)

break;

}

if (i < total)

return i;

return -1;

}

//删除字符串第一个字符

void removeHead(char \*str)

{

int i;

for (i=0; str[i]; i++)

str[i] = str[i+1];

}

//删除一个字符

int deleteChar(char c)

{

if (specailOne.str[0] == '\0') //

{

return deleteWord(c);

} else {

if (c == specailOne.str[0]){

removeHead(specailOne.str);

specailOne.pos.X += 1;

return 0;

}

}

return -1;

}

//现有的单词数

int wordLength()

{

return total;

}

//所有单词下落一行

void downWords()

{

int i;

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

head[i].pos.Y += 1;

if (head[i].pos.Y > HEIGHT){

head[i].pos.Y = HEIGHT;

fail();

total = 0;

}

}

if (specailOne.str[0] != '\0')

{

specailOne.pos.Y += 1;

if (specailOne.pos.Y > HEIGHT)

{

specailOne.str[0] = '\0';

fail();

}

}

}

void mycopy(char \*taget, char \*src)

{

int i =0;

while(src[i])

taget[i]=src[i++];

}

//设置窗口缓冲区

void setWindowsBuffer(char \*buffer)

{

int i;

memset(buffer, ' ', HEIGHT\*WIDTH);

for (i=0; i<total; i++)

mycopy(buffer+head[i].pos.X+head[i].pos.Y\*WIDTH, head[i].str);

}

//显示现有单词，调试用

void showWords()

{

int i;

for (i=0; i<total; i++)

printf("%s(%d,%d) ", head[i].str,head[i].pos.X, head[i].pos.Y);

//printf("%s ", p->str);

printf("\n");

}

1. Dazi

//dazi.h

#ifndef DAZI\_H

#define DAZI\_H

#include "words.h"

#include <stdio.h>

#include <conio.h>

#include <time.h>

#include <windows.h>

#pragma comment(lib,"Winmm.lib")

#define WIDTH 80

#define HEIGHT 35

//背景颜色

#define F\_BLUE 0x1

#define F\_GREEN 0x2

#define F\_AQUA 0x3

#define F\_RED 0x4

#define F\_PURPLE 0x5

#define F\_YELLOW 0x6

#define F\_WHITE 0x7

#define F\_Gray 0x8

#define F\_Light\_Blue 0x9

#define F\_Light\_Green 0xA

#define F\_Light\_Aqua 0xB

#define F\_Light\_Red 0xC

#define F\_Light\_Purple 0xD

#define F\_Light\_Yellow 0xE

#define F\_Brigt\_white 0xF

#define B\_BLACK 0x00

#define B\_BLUE 0x10

#define B\_GREEN 0x20

#define B\_AQUA 0x30

#define B\_RED 0x40

#define B\_PURPLE 0x50

#define B\_YELLOW 0x60

#define B\_WHITE 0x70

#define B\_Gray 0x80

//控制上下左右

#define KEY\_UP 72

#define KEY\_DOWN 80

#define KEY\_LEFT 75

#define KEY\_RIGHT 77

//选定键：enter

#define KEY\_OK 13

//退出键:ctrl+X

#define KEY\_QUIT 24

//设置键：Esc

#define KEY\_ESC 27

//定时器时间

#define TIME\_DELAY 2000

//级数

#define LEVEL\_NUM 3

//定时器精度

#define RESOLUTION 1

extern HANDLE hOut;

extern CONSOLE\_SCREEN\_BUFFER\_INFO bBackInfo; //窗口缓冲区信息

extern words specailOne;

int Init();

int Quit();

int Run();

//失败

void fail();

#endif

//dazi.c

#include "dazi.h"

#include "words.h"

#include "file.h"

char title[] = "梅梅打字";

HANDLE hOut;

CONSOLE\_SCREEN\_BUFFER\_INFO bBackInfo; //窗口缓冲区信息

CONSOLE\_CURSOR\_INFO cursorInfo; //鼠标的信息

UINT cp;

words specailOne; //正在处理的单词

char screenBuffer[WIDTH\*HEIGHT]; //窗口的缓冲区内容

UINT timerID; //定时器的ID

int delay = TIME\_DELAY;

WORD attribute = F\_Light\_Green ; //屏幕显示属性

WORD specailAttri = F\_Light\_Red; //当前单词显示属性输出样式

int musicFlag = 1; //是否有声音

int failFlag = 0; //是否失败

//清屏幕

void clearScreen();

void PASCAL refresh(UINT wTimerID, UINT msg,DWORD dwUser,DWORD dwl,DWORD dw2) ;

//初始化的时候设置窗口

int setWindow()

{

SMALL\_RECT rect;

COORD size;

COORD home= {0,0};

DWORD result;

rect.Left = rect.Top = 0;

rect.Right = WIDTH -1;

rect.Bottom = HEIGHT -1;

SetConsoleWindowInfo(hOut, TRUE, &rect);

size.X = WIDTH;

size.Y = HEIGHT;

SetConsoleScreenBufferSize(hOut, size);

FillConsoleOutputAttribute(hOut, attribute, WIDTH\*HEIGHT, home,&result);

return 0;

}

//初始化

int Init()

{

CONSOLE\_SCREEN\_BUFFER\_INFO bBackInfo; //窗口缓冲区信息

hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

cp = GetConsoleOutputCP(); //代码页

SetConsoleOutputCP(437);

GetConsoleScreenBufferInfo(hOut, &bBackInfo); //保存窗口信息

GetConsoleCursorInfo(hOut, &cursorInfo);

cursorInfo.bVisible = FALSE;

SetConsoleCursorInfo(hOut, &cursorInfo);

SetConsoleTitle(title);

total = 0; //现有单词总数为0

specailOne.str[0] = '\0'; //空

if (setWindow())

{

printf("set window error!\n");

return -1;

}

srand( (unsigned)time( NULL ) );//设置随机种子

if (setFile())

{

// printf("file open error!\n");

return -1;

}

return 0;

}

//退出时恢复

int Quit()

{

SetConsoleScreenBufferSize(hOut, bBackInfo.dwSize);

SetConsoleTextAttribute(hOut, bBackInfo.wAttributes);

SetConsoleWindowInfo(hOut, FALSE, &(bBackInfo.srWindow));

cursorInfo.bVisible = TRUE;

SetConsoleCursorInfo(hOut, &cursorInfo);

SetConsoleOutputCP(cp);

return 0;

}

//显示当前的单词

void showSpecailWord()

{

static int len;

static COORD pos;

DWORD result;

FillConsoleOutputAttribute(hOut, attribute,len, pos, &result); //恢复显示属性

len = strlen(specailOne.str);

pos = specailOne.pos;

if (specailOne.str[0] == '\0')

return ;

FillConsoleOutputAttribute(hOut, specailAttri,strlen(specailOne.str), pos, &result);

WriteConsoleOutputCharacter(hOut, specailOne.str, strlen(specailOne.str),pos, &result);

}

void showScreenBuffer()

{

COORD home = {0,0};

DWORD result;

setWindowsBuffer(screenBuffer);

WriteConsoleOutputCharacter(hOut, screenBuffer,WIDTH\*HEIGHT,home,&result);

showSpecailWord();

}

void resumeTimer()

{

timerID = timeSetEvent(10, RESOLUTION, refresh, 0, TIME\_ONESHOT);//快速重启

if (timerID == 0)

{

printf("timer error!\n");

}

}

void startTimer()

{

if (failFlag)

return ;

timerID = timeSetEvent(delay, RESOLUTION, refresh, 0, TIME\_ONESHOT);

if (timerID == 0)

{

printf("timer error!\n");

}

}

void stopTimer()

{

timeKillEvent(timerID);

}

//清屏幕

void clearScreen()

{

COORD home = {0,0};

DWORD result;

memset(screenBuffer, ' ', HEIGHT\*WIDTH);

WriteConsoleOutputCharacter(hOut, screenBuffer,WIDTH\*HEIGHT,home,&result);

}

void PASCAL refresh(UINT wTimerID, UINT msg,DWORD dwUser,DWORD dwl,DWORD dw2)

{

addWord();

downWords();

showScreenBuffer();

//showWords();

startTimer();

}

//显示字符串

void showString(char \*str, COORD pos, WORD attri)

{

DWORD result;

FillConsoleOutputAttribute(hOut, attri,strlen(str),pos, &result);

WriteConsoleOutputCharacter(hOut, str, strlen(str),pos, &result);

}

//设置关卡

void showLevel()

{

COORD pos= {WIDTH/2 - 10, HEIGHT/2 - 6};

int choose = 3;

int totalChoose = 4;

int quit = 0;

int ypos;

char c;

clearScreen();

showString("Speed level", pos, specailAttri);

pos.Y += 4;

pos.X += 2;

showString("1.Slow", pos, attribute);

pos.Y += 2;

showString("2.Faster", pos, attribute);

pos.Y += 2;

showString("3.Fastest", pos, specailAttri);

pos.Y += 2;

showString("0.Back", pos, attribute);

pos.X -= 2;

pos.Y -= (totalChoose-1)\*2;

ypos = pos.Y;

while (quit != 1) {

//显示光标

showString(" ", pos, specailAttri);

pos.Y = ypos + choose\*2;

showString("\4", pos, specailAttri);

c = getch();

switch(c) {

case KEY\_OK:

quit = 1;

break;

default:

if (c < 0)

{

c = getch();

if (c == KEY\_UP || c == KEY\_LEFT) //上一个

{

choose = (choose + totalChoose -1) % totalChoose;

} else if (c == KEY\_RIGHT || KEY\_DOWN) //下一个

{

choose = (choose + 1) % totalChoose;

}

}

if (c>='0' && c < ('0'+totalChoose))

{

choose = c - '0';

choose = (choose + totalChoose -1) % totalChoose;

}

break;

}

}

fflush(stdin);

if (choose>=0 && choose <= 2)

{

//0: 1500

//1: 1000

//2: 500

delay = TIME\_DELAY - (choose \* TIME\_DELAY) / LEVEL\_NUM;

}

}

void showMusicSettings()

{

COORD pos= {WIDTH/2 - 10, HEIGHT/2 - 6};

int choose = 0;

int totalChoose = 2;

int quit = 0;

int ypos;

char c;

clearScreen();

showString("Music settings", pos, specailAttri);

pos.Y += 4;

pos.X += 2;

showString("1.On", pos, attribute);

pos.Y += 2;

showString("0.Off", pos, attribute);

pos.X -= 2;

pos.Y -= (totalChoose-1)\*2;

ypos = pos.Y;

while (quit != 1) {

//显示光标

showString(" ", pos, specailAttri);

pos.Y = ypos + choose\*2;

showString("\4", pos, specailAttri);

c = getch();

switch(c) {

case KEY\_OK:

quit = 1;

break;

default:

if (c < 0)

{

c = getch();

if (c == KEY\_UP || c == KEY\_LEFT) //上一个

{

choose = (choose + totalChoose -1) % totalChoose;

} else if (c == KEY\_RIGHT || KEY\_DOWN) //下一个

{

choose = (choose + 1) % totalChoose;

}

}

if (c>='0' && c < ('0'+totalChoose))

{

choose = c - '0';

choose = (choose + totalChoose -1) % totalChoose;

}

break;

}

}

fflush(stdin);

if (choose == 0) //on

{

musicFlag = 1;

} else {

musicFlag = 0;

}

}

//设置前景色

void setForeColorSettings()

{

COORD pos= {WIDTH/2 - 10, HEIGHT/2 - 10};

int choose = 15;

int totalChoose = 16;

int quit = 0;

int ypos;

char c;

clearScreen();

showString("fore ground color", pos, specailAttri);

pos.Y += 4;

pos.X += 1;

showString("1. Blue", pos, F\_BLUE);

pos.Y += 1;

showString("2. Green", pos, F\_GREEN);

pos.Y += 1;

showString("3. Aqua", pos, F\_AQUA);

pos.Y += 1;

showString("4. Red", pos, F\_RED);

pos.Y += 1;

showString("5. Purple", pos, F\_PURPLE);

pos.Y += 1;

showString("6. Yellow", pos, F\_YELLOW);

pos.Y += 1;

showString("7. White", pos, F\_WHITE);

pos.Y += 1;

showString("8. Gray", pos, F\_Gray);

pos.Y += 1;

showString("9. Light Blue", pos, F\_Light\_Blue);

pos.Y += 1;

showString("10.Light Green", pos, F\_Light\_Green);

pos.Y += 1;

showString("11.Light Aqua", pos, F\_Light\_Aqua );

pos.Y += 1;

showString("12.Light Red", pos, F\_Light\_Red);

pos.Y += 1;

showString("13.Light Purple", pos, F\_Light\_Purple);

pos.Y += 1;

showString("14.Light Yellow", pos, F\_Light\_Yellow);

pos.Y += 1;

showString("15.Light White", pos, F\_Brigt\_white);

pos.Y += 1;

showString("0.Back", pos, attribute);

pos.X -= 2;

pos.Y -= (totalChoose-1)\*1;

ypos = pos.Y;

while (quit != 1) {

//显示光标

showString(" ", pos, specailAttri);

pos.Y = ypos + choose\*1;

showString("\4", pos, specailAttri);

c = getch();

switch(c) {

case KEY\_OK:

quit = 1;

break;

default:

if (c < 0)

{

c = getch();

if (c == KEY\_UP || c == KEY\_LEFT) //上一个

{

choose = (choose + totalChoose -1) % totalChoose;

} else if (c == KEY\_RIGHT || KEY\_DOWN) //下一个

{

choose = (choose + 1) % totalChoose;

}

}

if (c>='0' && c < ('0'+totalChoose))

{

choose = c - '0';

choose = (choose + totalChoose -1) % totalChoose;

}

break;

}

}

fflush(stdin);

if (choose<15 && choose >=0)

{

attribute &= 0xf0;

attribute |= choose+1;

}

}

//设置背景色

void setBackColorSettings()

{

COORD pos= {WIDTH/2 - 10, HEIGHT/2 - 10};

int choose = 9;

int totalChoose = 10;

int quit = 0;

int ypos;

char c;

clearScreen();

showString("Background color", pos, specailAttri);

pos.Y += 4;

pos.X += 2;

showString("1.Black", pos, F\_WHITE|B\_BLACK);

pos.Y += 2;

showString("2.Blue", pos, F\_WHITE|B\_BLUE);

pos.Y += 2;

showString("3.Green", pos, F\_WHITE|B\_GREEN);

pos.Y += 2;

showString("4.Aqua", pos, F\_WHITE|B\_AQUA);

pos.Y += 2;

showString("5.Red", pos, F\_WHITE|B\_RED);

pos.Y += 2;

showString("6.Purple", pos, F\_WHITE|B\_PURPLE);

pos.Y += 2;

showString("7.Yellow", pos, F\_WHITE|B\_YELLOW);

pos.Y += 2;

showString("8.White", pos, F\_BLUE|B\_WHITE);

pos.Y += 2;

showString("9.Gray", pos, F\_WHITE|B\_Gray);

pos.Y += 2;

showString("0.Back", pos, attribute);

pos.X -= 2;

pos.Y -= (totalChoose-1)\*2;

ypos = pos.Y;

while (quit != 1) {

//显示光标

showString(" ", pos, specailAttri);

pos.Y = ypos + choose\*2;

showString("\4", pos, specailAttri);

c = getch();

switch(c) {

case KEY\_OK:

quit = 1;

break;

default:

if (c < 0)

{

c = getch();

if (c == KEY\_UP || c == KEY\_LEFT) //上一个

{

choose = (choose + totalChoose -1) % totalChoose;

} else if (c == KEY\_RIGHT || KEY\_DOWN) //下一个

{

choose = (choose + 1) % totalChoose;

}

}

if (c>='0' && c < ('0'+totalChoose))

{

choose = c - '0';

choose = (choose + totalChoose -1) % totalChoose;

}

break;

}

}

fflush(stdin);

if (choose<9 && choose >=0)

{

attribute &= 0xf;

attribute |= choose \* 0x10;

specailAttri &= 0xf;

specailAttri |= choose \* 0x10;

}

}

//设置颜色

void showColorSettings()

{

setForeColorSettings();

setBackColorSettings();

setWindow();

}

//设置单词文件

int showFileSettings()

{

char path[MAX\_PATH\_LENGHT];

clearScreen();

printf("\ninput the filename: ");

if(scanf("%s", path)==EOF)

return -1;

fflush(stdin);

setFilename(path);

setWindow();

return 0;

}

//设置取词顺序

void showSeqenceSettings()

{

COORD pos= {WIDTH/2 - 10, HEIGHT/2 - 6};

int choose = 0;

int totalChoose = 2;

int quit = 0;

int ypos;

char c;

clearScreen();

showString("Words sequence", pos, specailAttri);

pos.Y += 4;

pos.X += 2;

showString("1.Yes", pos, attribute);

pos.Y += 2;

showString("0.No", pos, attribute);

pos.X -= 2;

pos.Y -= (totalChoose-1)\*2;

ypos = pos.Y;

while (quit != 1) {

//显示光标

showString(" ", pos, specailAttri);

pos.Y = ypos + choose\*2;

showString("\4", pos, specailAttri);

c = getch();

switch(c) {

case KEY\_OK:

quit = 1;

break;

default:

if (c < 0)

{

c = getch();

if (c == KEY\_UP || c == KEY\_LEFT) //上一个

{

choose = (choose + totalChoose -1) % totalChoose;

} else if (c == KEY\_RIGHT || KEY\_DOWN) //下一个

{

choose = (choose + 1) % totalChoose;

}

}

if (c>='0' && c < ('0'+totalChoose))

{

choose = c - '0';

choose = (choose + totalChoose -1) % totalChoose;

}

break;

}

}

fflush(stdin);

if (choose == 0) //on

{

setSeqence(1);

} else {

setSeqence(0);

}

}

//设置界面

void showSettings()

{

COORD pos= {WIDTH/2 - 10, HEIGHT/2 - 6};

int choose = 0;

int totalChoose = 6;

int quit = 0;

int ypos;

char c;

clearScreen();

showString("Settings", pos, specailAttri);

pos.Y += 4;

pos.X += 2;

showString("1.Music", pos, attribute);

pos.Y += 2;

showString("2.Color", pos, attribute);

pos.Y += 2;

showString("3.File", pos, attribute);

pos.Y += 2;

showString("4.Sequence", pos, attribute);

pos.Y += 2;

showString("5.Exit", pos, attribute);

pos.Y += 2;

showString("0.Back", pos, attribute);

pos.X -= 2;

pos.Y -= (totalChoose-1)\*2;

ypos = pos.Y;

while (quit != 1) {

//显示光标

showString(" ", pos, specailAttri);

pos.Y = ypos + choose\*2;

showString("\4", pos, specailAttri);

c = getch();

switch(c) {

case KEY\_OK:

quit = 1;

break;

default:

if (c < 0)

{

c = getch();

if (c == KEY\_UP || c == KEY\_LEFT) //上一个

{

choose = (choose + totalChoose -1) % totalChoose;

} else if (c == KEY\_RIGHT || KEY\_DOWN) //下一个

{

choose = (choose + 1) % totalChoose;

}

}

if (c>='0' && c < ('0'+totalChoose))

{

choose = c - '0';

choose = (choose + totalChoose -1) % totalChoose;

}

break;

}

}

fflush(stdin);

switch (choose) {

case 0: //music

showMusicSettings();

break;

case 1: //color

showColorSettings();

break;

case 2: // file

showFileSettings();

break;

case 3: // sequence

showSeqenceSettings();

break;

case 4: //Exit

fail();

break;

default: //back

break;

}

setWindow(); //恢复图形显示

}

//退出界面

void showQuit()

{

COORD pos= {WIDTH/2 - 10, HEIGHT/2 - 6};

int choose = 0;

int totalChoose = 2;

int quit = 0;

int ypos;

char c;

clearScreen();

showString("Do you want quit?", pos, specailAttri);

pos.Y += 4;

pos.X += 2;

showString("1.Yes", pos, attribute);

pos.Y += 2;

showString("0.No", pos, attribute);

pos.X -= 2;

pos.Y -= (totalChoose-1)\*2;

ypos = pos.Y;

while (quit != 1) {

//显示光标

showString(" ", pos, specailAttri);

pos.Y = ypos + choose\*2;

showString("\4", pos, specailAttri);

c = getch();

switch(c) {

case KEY\_OK:

quit = 1;

break;

default:

if (c < 0)

{

c = getch();

if (c == KEY\_UP || c == KEY\_LEFT) //上一个

{

choose = (choose + totalChoose -1) % totalChoose;

} else if (c == KEY\_RIGHT || KEY\_DOWN) //下一个

{

choose = (choose + 1) % totalChoose;

}

}

if (c>='0' && c < ('0'+totalChoose))

{

choose = c - '0';

choose = (choose + totalChoose -1) % totalChoose;

}

break;

}

}

fflush(stdin);

if (choose == 0)

{

Quit();

exit(0);

}

}

//开始界面

void showWelcome()

{

COORD pos= {WIDTH/2 - 10, HEIGHT/2 - 6};

int choose = 0;

int totalChoose = 4;

int quit = 0;

int ypos;

char c;

clearScreen();

showString("\1!!! welcome game!!! \1", pos, attribute);

pos.Y += 4;

pos.X += 4;

showString("1.Start", pos, attribute);

pos.Y += 2;

showString("2.Setting", pos, specailAttri);

pos.Y += 2;

showString("3.Level", pos, attribute);

pos.Y += 2;

showString("0.Quit", pos, specailAttri);

pos.X -= 2;

pos.Y -= (totalChoose-1)\*2;

ypos = pos.Y;

while (quit != 1) {

//显示光标

showString(" ", pos, specailAttri);

pos.Y = ypos + choose\*2;

showString("\4", pos, specailAttri);

c = getch();

switch(c) {

case KEY\_OK:

quit = 1;

break;

case KEY\_QUIT:

showQuit();

break;

default:

if (c < 0)

{

c = getch();

if (c == KEY\_UP || c == KEY\_LEFT) //上一个

{

choose = (choose + totalChoose -1) % totalChoose;

} else if (c == KEY\_RIGHT || KEY\_DOWN) //下一个

{

choose = (choose + 1) % totalChoose;

}

}

if (c>='0' && c < ('0'+totalChoose))

{

choose = c - '0';

choose = (choose + totalChoose -1) % totalChoose;

}

break;

}

}

switch(choose) {

case 0: //start

failFlag = 0;

break;

case 1: //settings

showSettings();

//如果没有退出

showWelcome();

break;

case 2: //level

showLevel();

//如果没有退出

showWelcome();

break;

case 3: //quit

showQuit();

//如果没有退出

showWelcome();

break;

default: //error

break;

}

setWindow(); //重新设置窗口显示信息

}

int Run()

{

char c;

showWelcome();

resumeTimer();

while (1)

{

while(!kbhit()) //有键输入

if (failFlag)

return 0;

c = getch();

if (c == KEY\_QUIT) //ctrl+x,退出

{

fflush(stdin);

stopTimer();

showQuit();

resumeTimer();

}

if (c == KEY\_ESC) //settings

{

fflush(stdin);

stopTimer();

showSettings();

resumeTimer();

}

if(deleteChar(c) == 0)

{

//成功

if (musicFlag)

MessageBeep(MB\_ICONASTERISK);

} else {

if (musicFlag)

MessageBeep(MB\_ICONHAND);

}

showScreenBuffer();

//showWords();

}

showQuit();

stopTimer();

return 0;

}

//失败

void fail()

{

stopTimer();

failFlag = 1;

}