/\*------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Dialog.h\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------\*/

#include <windows.h>

//------------------------函数申明----------------------------

BOOL CALLBACK DlgSpeed (HWND, UINT, WPARAM, LPARAM) ;//速度设置对话框

BOOL CALLBACK DlgTime (HWND, UINT, WPARAM, LPARAM) ;//时间设置对话框

/\*-----------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Dialog.cpp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-----------------------------------------------------------------------------\*/

#include "resource.h"

#include "Dialog.h"

extern int iSpeed, iTime, count\_time ;

//------------------------速度设置对话框------------------------

BOOL CALLBACK DlgSpeed (HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)

{

int temp ;

switch (message)

{

case WM\_INITDIALOG :

return TRUE ;

case WM\_COMMAND:

switch (LOWORD (wParam))

{

case IDOK :

temp = GetDlgItemInt (hDlg, IDC\_SPEED, NULL, TRUE) ;

if (temp==0)

{

MessageBox (NULL, TEXT("对不起，速度不能为0 "), TEXT("提示"),MB\_ICONINFORMATION);

return TRUE;

}

if (temp>200)

{

MessageBox (NULL, TEXT("对不起，速度不能为大于200 "), TEXT("提示"), MB\_ICONINFORMATION) ;

return TRUE;

}

iSpeed = temp ;

EndDialog (hDlg, TRUE) ;

return TRUE ;

case IDCANCEL :

EndDialog (hDlg, FALSE) ;

return TRUE ;

}

break ;

}

return FALSE ;

}

//------------------------时间设置对话框-----------------------

BOOL CALLBACK DlgTime (HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)

{

int temp ;

char str[30] ;

switch (message)

{

case WM\_INITDIALOG :

return TRUE ;

case WM\_COMMAND:

switch (LOWORD (wParam))

{

case IDOK :

temp = GetDlgItemInt (hDlg, IDC\_TIME, NULL, TRUE) ;

if (temp==0)

{

MessageBox (NULL, TEXT("对不起，游戏时间不能为0 "), TEXT("提示"), MB\_ICONINFORMATION) ;

return TRUE;

}

if (temp<=count\_time/6000)

{

wsprintf (str, "对不起,现在已经超过%d分钟了", count\_time/6000) ;

MessageBox (NULL, str, TEXT("提示"), MB\_ICONINFORMATION) ;

return TRUE;

}

iTime = temp ;

EndDialog (hDlg, TRUE) ;

return TRUE;

case IDCANCEL :

EndDialog (hDlg, FALSE) ;

return TRUE ;

}

break;

}

return FALSE ;

}

/\*------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*function.h\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------\*/

#include <windows.h>

#define ID\_EDIT 1

#define BEGIN 2

#define END 3

#define CLOSE 4

#define RUN 5

#define PAUSE 6

#define STOP 7

#define TIMER\_FALL 1

#define TIMER\_TIME 2

#define TIME\_TIME 10

#define LETTER\_NUM 5

#define TEXT\_LEFT 10

#define TEXT\_TOP (height-80)

typedef struct{

TCHAR iLetter ;

POINT pos ;

}LETTER ;

//------------------------函数申明----------------------------

void CreatButton (HWND, HINSTANCE) ;

void DrawBack (HDC) ;

void InitLetter () ;

void CleanGame (HWND) ;

void FallLetter (HWND) ;

void DrawTime (HWND) ;

void DrawLetter (HDC) ;

void ClickLetter (HWND, TCHAR) ;

void DrawNumber (HWND) ;

/\*-----------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*function.cpp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-----------------------------------------------------------------------------\*/

#include <stdio.h>

#include "function.h"

//------------------------------全局变量-----------------------------------

LETTER NowLetter[LETTER\_NUM] ;

int num\_letter, num\_correct, count\_time = 0 ;

extern HWND hwndButton\_Begin, hwndButton\_Stop, hwndButton\_Close, hwndEdit ;

extern int iGameStatus, width, height, iSpeed, iTime ;

/\*----------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*函数群\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-----------------------------------------------------------------------------\*/

//----------------------创建按钮----------------------------

void CreatButton(HWND hwnd, HINSTANCE hInstance)

{

hwndButton\_Begin = CreateWindow (TEXT("button"), TEXT("开始"),

WS\_CHILD | WS\_VISIBLE | BS\_DEFPUSHBUTTON,

0, 0, 0, 0, hwnd, (HMENU) BEGIN, hInstance, NULL) ;

hwndButton\_Stop = CreateWindow (TEXT("button"), TEXT("停止"),

WS\_CHILD | WS\_VISIBLE | BS\_DEFPUSHBUTTON,

0, 0, 0, 0, hwnd, (HMENU) END, hInstance, NULL) ;

hwndButton\_Close = CreateWindow (TEXT("button"), TEXT("关闭"),

WS\_CHILD | WS\_VISIBLE | BS\_DEFPUSHBUTTON,

0, 0, 0, 0, hwnd, (HMENU) CLOSE, hInstance, NULL) ;

return ;

}

//----------------------划分界线----------------------------

void DrawBack (HDC hdc)

{

char text[10] ;

SelectObject (hdc, CreatePen (PS\_SOLID, 2, RGB (0, 0, 0))) ;

MoveToEx (hdc, 0, height-100, NULL) ;

LineTo (hdc, width, height-100) ;

sprintf (text, "正确数:") ;

TextOut (hdc, TEXT\_LEFT, TEXT\_TOP, text, strlen (text)) ;

sprintf (text, "总 数:") ;

TextOut (hdc, TEXT\_LEFT, TEXT\_TOP+30, text, strlen (text)) ;

DeleteObject (SelectObject (hdc, GetStockObject (BLACK\_PEN))) ;

return;

}

//---------------------初始化字母------------------------------

void InitLetter()

{

int i ;//为rand()得到随机数作准备;

srand ((int) GetCurrentTime ());

for (i=0; i<LETTER\_NUM; i++)

{

NowLetter[i].iLetter = rand ()%26 ;

NowLetter[i].iLetter += 65 ;

NowLetter[i].pos.x = 100+i\*100 ;

NowLetter[i].pos.y = -20 ;

}

//初始化统计数字

num\_correct = 0 ;

//游戏刚开始就有5个

num\_letter = 5 ;

return ;

}

//---------------------清理游戏区------------------------------

void CleanGame(HWND hwnd)

{

HDC hdc ;

HBRUSH hBrush ;

//

hdc = GetDC (hwnd) ;

SelectObject (hdc, GetStockObject (WHITE\_PEN)) ;

hBrush = (HBRUSH) SelectObject (hdc, CreateHatchBrush (BS\_DIBPATTERNPT, RGB (255, 255, 255))) ;

Rectangle (hdc, 0, 0, width, height-103) ;

DeleteObject (SelectObject (hdc, GetStockObject (BLACK\_PEN))) ;

DeleteObject (SelectObject (hdc, hBrush)) ;

ReleaseDC (hwnd, hdc) ;

return ;

}

//---------------------落字母------------------------------

void FallLetter(HWND hwnd)

{

int i ;

HDC hdc ;

hdc = GetDC (hwnd) ;

SetBkMode (hdc, TRANSPARENT) ;//透明文本

SelectObject (hdc, CreatePen (PS\_SOLID, 3, RGB (255, 255, 255))) ;

for (i=0; i<LETTER\_NUM; i++)

{

Ellipse (hdc, NowLetter[i].pos.x, NowLetter[i].pos.y,

NowLetter[i].pos.x+30, NowLetter[i].pos.y+30) ;

NowLetter[i].pos.y += 5 ;

if(NowLetter[i].pos.y>height-130)

{

//没打住的也统计在内

num\_letter ++ ;

DrawNumber (hwnd) ;

NowLetter[i].pos.y = -20 ;

}

}

DeleteObject (SelectObject (hdc, GetStockObject (BLACK\_PEN))) ;

DrawLetter (hdc) ;

ReleaseDC (hwnd, hdc) ;

return ;

}

//---------------------画正在落的字母----------------------

void DrawLetter(HDC hdc)

{

int i ;

char letter[2] ;

HBRUSH hBrush ;

//

hBrush = (HBRUSH) SelectObject (hdc, CreateHatchBrush (BS\_DIBPATTERNPT, RGB (0, 255, 0))) ;

for (i=0; i<LETTER\_NUM; i++)

{

Ellipse (hdc, NowLetter[i].pos.x, NowLetter[i].pos.y,

NowLetter[i].pos.x+30, NowLetter[i].pos.y+30) ;

sprintf (letter, "%c", NowLetter[i].iLetter) ;

TextOut (hdc, NowLetter[i].pos.x+10, NowLetter[i].pos.y+8, letter, strlen(letter)) ;

}

DeleteObject (SelectObject (hdc, hBrush)) ;

return ;

}

//---------------------画正在落的字母----------------------

void ClickLetter(HWND hwnd, TCHAR letter)

{

int i ;

HDC hdc ;

for (i=0; i<LETTER\_NUM; i++)

if (letter==NowLetter[i].iLetter+32||letter==NowLetter[i].iLetter)

{

PlaySound(TEXT("Sound\\Directdown.wav"),NULL,SND\_FILENAME|SND\_ASYNC);

hdc = GetDC (hwnd) ;

SelectObject (hdc, GetStockObject (WHITE\_PEN)) ;

Ellipse (hdc, NowLetter[i].pos.x, NowLetter[i].pos.y,

NowLetter[i].pos.x+30, NowLetter[i].pos.y+30) ;

//统计字母数量

//新的字母要出现,统计

num\_letter ++ ;

//正确数加1

num\_correct ++ ;

DrawNumber (hwnd) ;

//将黑色笔恢复为设备描述表项

DeleteObject (SelectObject (hdc, GetStockObject (BLACK\_PEN))) ;

ReleaseDC (hwnd, hdc) ;

//产生新的字母

NowLetter[i].iLetter = rand ()%26 ;

NowLetter[i].iLetter += 65 ;

NowLetter[i].pos.y = -20 ;

break ;

}

return ;

}

//---------------------显示时间----------------------

void DrawTime(HWND hwnd)

{

char time[8] ;

HDC hdc ;

count\_time ++ ;

hdc = GetDC (hwnd) ;

//用白笔白画刷擦去原来的时间

SelectObject (hdc, GetStockObject (WHITE\_PEN)) ;

Rectangle (hdc, width-50, 30, width, 50) ;

DeleteObject (SelectObject (hdc, CreatePen (PS\_SOLID, 3, RGB (255, 0, 0)))) ;

sprintf (time, "%d:%d.%d",count\_time/6000,(count\_time%6000)/100,count\_time%100) ;

TextOut (hdc, width-50, 30, time, strlen (time)) ;

DeleteObject (SelectObject (hdc, GetStockObject (BLACK\_PEN))) ;

ReleaseDC (hwnd, hdc) ;

//判断是不是已经时间了

if (count\_time/6000>=iTime)

{

//游戏停止

SendMessage (hwnd, WM\_COMMAND, STOP, NULL) ;

//准备下一次游戏的开始

count\_time = 0 ;

}

return ;

}

//---------------------显示统计数字----------------------

void DrawNumber(HWND hwnd)

{

char num[5] ;

HDC hdc ;

hdc = GetDC (hwnd) ;

SelectObject (hdc, GetStockObject (WHITE\_PEN)) ;

//擦去原来的统计数目

Rectangle (hdc, TEXT\_LEFT+60, TEXT\_TOP, TEXT\_LEFT+90, TEXT\_TOP+60) ;

//用红色显示正确的数目

SetTextColor (hdc, RGB (255, 0, 0)) ;

//改写正确的数目

sprintf (num, "%d", num\_correct) ;

TextOut (hdc, TEXT\_LEFT+60, TEXT\_TOP, num, strlen (num)) ;

//改写总数

sprintf (num, "%d", num\_letter) ;

TextOut (hdc, TEXT\_LEFT+60, TEXT\_TOP+30, num, strlen (num)) ;

//将黑色笔恢复为设备描述表项

DeleteObject (SelectObject (hdc, GetStockObject (BLACK\_PEN))) ;

ReleaseDC (hwnd, hdc) ;

return ;

}

/\*-----------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*打字.cpp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-----------------------------------------------------------------------------\*/

#include <windows.h>

#include "resource.h"

#include "function.h"

#include "Dialog.h"

//------------------------------全局变量-----------------------------------

HWND hwndButton\_Begin, hwndButton\_Stop, hwndButton\_Close ;

HMENU hMenu;

int iGameStatus, width, height, iSpeed, iTime ;

extern LETTER iNowLetter[LETTER\_NUM] ;

extern int count\_time ;

//-----------------------------函数申明----------------------------------

LRESULT CALLBACK WndProc(HWND,UINT,WPARAM,LPARAM);

/\*-------------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*函数入口\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-------------------------------------------------------------------------------\*/

int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,

PSTR szCmdLine,int iCmdShow)

{

static TCHAR szAppName[]=TEXT("打字");

WNDCLASS wndclass;

HWND hwnd;

MSG msg;

wndclass.style =NULL;

wndclass.lpfnWndProc =WndProc;

wndclass.cbClsExtra =0;

wndclass.cbWndExtra =0;

wndclass.hInstance =hInstance;

wndclass.hIcon =LoadIcon (hInstance, MAKEINTRESOURCE(IDI\_ICON1)) ;

wndclass.hCursor =LoadCursor(NULL, IDC\_ARROW);

wndclass.hbrBackground =(HBRUSH) GetStockObject (WHITE\_BRUSH) ;

wndclass.lpszMenuName =NULL;

wndclass.lpszClassName =szAppName;

if(!RegisterClass(&wndclass))

{

MessageBox(NULL,TEXT("Program requires Windows NT!"),

szAppName,MB\_ICONERROR);

return 0;

}

hMenu=LoadMenu(hInstance,MAKEINTRESOURCE(IDR\_MENU1));

hwnd=CreateWindow(szAppName, //window class name

szAppName, //window caption

WS\_OVERLAPPEDWINDOW&~WS\_SIZEBOX&~WS\_MAXIMIZEBOX,//window style

350, //initial x position

150, //initial y position

650, //initial x size

500, //initial y size

NULL, //parent window handle

hMenu, //window minu handle

hInstance, //program instance handle

NULL); //creation parameters

ShowWindow(hwnd,iCmdShow);

UpdateWindow(hwnd);

while(GetMessage(&msg,NULL,0,0))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

return msg.wParam;

}

/\*-------------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*窗口过程函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-------------------------------------------------------------------------------\*/

LRESULT CALLBACK WndProc(HWND hwnd,UINT message,WPARAM wParam,LPARAM lParam)

{

static HINSTANCE hInstance ;

PAINTSTRUCT ps ;

HDC hdc ;

switch(message)

{

//-------WM\_CREATE--------------------------------------------------------

case WM\_CREATE :

hInstance = ((LPCREATESTRUCT)lParam)->hInstance ;

//创建各种按钮

CreatButton (hwnd, hInstance) ;

//停止按钮不可用

EnableWindow (hwndButton\_Stop, FALSE);

//游戏结束

iGameStatus = STOP ;

//初始化时间设置

iSpeed = 100 ;

iTime = 5 ;//5分钟

return 0 ;

//-------WM\_COMMAND-------------------------------------------------------

case WM\_COMMAND :

hMenu = GetMenu(hwnd);

switch (wParam)

{

case SPEED :

if (DialogBox (hInstance, TEXT("Speed"), hwnd, DlgSpeed)==TRUE)

if (iGameStatus==RUN)

{

//重新创建字母下落的计时器

SetTimer (hwnd, TIMER\_FALL, iSpeed, NULL) ;

}

break ;

case TIME :

DialogBox (hInstance, TEXT("Time"), hwnd, DlgTime) ;

break ;

case BEGIN :

//清理界面

CleanGame (hwnd) ;

//游戏开始

iGameStatus = RUN ;

//初始化字母

InitLetter () ;

DrawNumber (hwnd) ;

//开始按钮不可用

EnableWindow (hwndButton\_Begin, FALSE) ;

//停止按钮可用

EnableWindow (hwndButton\_Stop, TRUE) ;

//将输入焦点设置为编辑框

SetFocus (hwnd) ;

//创建字母下落的计时器

SetTimer (hwnd, TIMER\_FALL, iSpeed, NULL) ;

//创建时间的计时器

SetTimer (hwnd, TIMER\_TIME, TIME\_TIME, NULL) ;

break ;

case END :

//游戏结束

iGameStatus = STOP ;

count\_time = 0 ;

//开始按钮可用

EnableWindow (hwndButton\_Begin, TRUE) ;

//停止按钮不可用

EnableWindow (hwndButton\_Stop, FALSE) ;

//停止两个计时器

KillTimer (hwnd, TIMER\_FALL) ;

KillTimer (hwnd, TIMER\_TIME) ;

break ;

case CLOSE :

//停止两个计时器

KillTimer (hwnd, TIMER\_FALL) ;

KillTimer (hwnd, TIMER\_TIME) ;

PostQuitMessage (0) ;

break ;

}

return 0 ;

//-------WM\_SIZE----------------------------------------------------------

case WM\_SIZE :

width = LOWORD (lParam) ;

height = HIWORD (lParam) ;

MoveWindow (hwndButton\_Begin, width-300, height-80, 70, 40, TRUE) ;

MoveWindow (hwndButton\_Stop, width-200, height-80, 70, 40, TRUE) ;

MoveWindow (hwndButton\_Close, width-100, height-80, 70, 40, TRUE) ;

return 0 ;

//-------WM\_TIMER----------------------------------------------------------

case WM\_TIMER :

switch(wParam)

{

case TIMER\_FALL :

if(iGameStatus==RUN)

FallLetter (hwnd) ;

break ;

case TIMER\_TIME :

if(iGameStatus==RUN)

DrawTime (hwnd) ;

break ;

}

return 0 ;

//-------WM\_TIMER----------------------------------------------------------

case WM\_CHAR :

if (iGameStatus==RUN)

ClickLetter (hwnd, (TCHAR) wParam) ;

return 0 ;

//-------WM\_INITMENUPOPUP---------------------------------------------------

case WM\_INITMENUPOPUP ://弹出菜单暂停

//-------WM\_KILLFOCUS-------------------------------------------------------

case WM\_KILLFOCUS://失去输入焦点

if (iGameStatus==RUN)

iGameStatus = PAUSE ;

return 0 ;

//-------WM\_EXITMENULOOP---------------------------------------------------

case WM\_EXITMENULOOP ://菜单收回

//-------WM\_SETFOCUS-------------------------------------------------------

case WM\_SETFOCUS ://得到输入焦点

if (iGameStatus==PAUSE)

iGameStatus = RUN ;

return 0 ;

//-------WM\_PAINT-----------------------------------------------------------

case WM\_PAINT :

hdc=BeginPaint (hwnd, &ps);

DrawBack (hdc) ;

if(iGameStatus!=STOP)

{

DrawLetter (hdc) ;

DrawNumber (hwnd) ;

}

EndPaint (hwnd, &ps) ;

return 0 ;

//------WM\_DESTROY-----------------------------------------------------------

case WM\_DESTROY :

//停止计时器

KillTimer (hwnd, TIMER\_FALL) ;

KillTimer (hwnd, TIMER\_TIME) ;

PostQuitMessage (0) ;

return 0 ;

}

return DefWindowProc( hwnd, message, wParam, lParam) ;

}