

easyx库 include < graphics.h> include < conio.h>
 initgraph(x, y) 创建窗体图形, 设置大小 closegraph()
 setcolor(YELLOW) 设置颜色
 setfillcolor(GREEN) fillcircle(x, y, r)
 sleep(t)

BeginBatchDraw 开始批量绘图 但不输出到屏幕上

EndBatchDraw 结束批量绘图

FlushBatchDraw 一批次同次输出到屏幕

_getch() / getch() 任意字符读取

~~setline~~ 画线 lin(x0, y0, x1, y1); setlinestyle(PS_SOLID, 2);

SYSTEMTIME ti; 获取当前时间

GetLocalTime(&ti); ti.wSecond 为秒数

Outtextxy(x, y, char); 在屏幕输出字符串内容

system("cls"); 清屏

char input;

if (kbhit()) // 当按任意键时执行

input = getch();

MOUSEMSG m;

if (MouseHit())

m = GetMouseMsg();

// 函数外全局变量定义

int main()

{

startup(); // 游戏初始化

while(1)

{

show(); 显示画面

updateWithoutInput(); // 与用户输入无关的更新

updateWithInput(); // 与用户输入有关的更新

return 0;

}

```
Void gotoxy(intx, inty) // 光标移动到 (x, y) 位置.
```

```
{
```

```
    HANDLE handle = GetStdHandle(STD_OUTPUT_HANDLE);
```

```
    COORD pos;
```

```
    pos.X = x;
```

```
    pos.Y = y;
```

```
    SetConsoleCursorPosition(handle, pos);
```

```
}
```

```
Void HideCursor() // 隐藏光标
```

```
{
```

```
    CONSOLE_CURSOR_INFO cursor_info = {1, 0};
```

```
    // 第二值为0表示隐藏光标
```

```
    SetConsoleCursorInfo(GetStdHandle(STD_OUTPUT_HANDLE),  
                          &cursor_info);
```

```
}
```

```
MOUSEMSG m;
```

```
if (MouseHit())
```

```
    m = GetMessage();    m.x m.y 为当前光标位置
```

```
    if (m.uMsg == WM_MOUSEMOVE) //判断鼠标是否移动
```

```
    if (m.uMsg == WM_LBUTTONDOWN) //判断左键是否按下.
```

```
        m.uMsg == WM_RBUTTONDOWN.. 判断右键按下又弹起
```

画点
putpixel(x, y, WHITE);

图像文件

```
IMAGE img_bk; //定义 IMAGE 对象
```

```
loadimage (&img_bk, "E:\\background.jpg");
```

```
putimage (0, 0, &img_bk); //在 (0,0) 位置显示 IMAGE 对象
```

```
putimage(x, y, &img, NOTSRCPERASE); 遮罩
```

SRCINVERT 图片

声音文件导入

播放MP3, 使用API函数mciSendString

使用前 #pragma comment(lib, "Winmm.lib")

```
mciSendString("open E:\\background.mp3 alias bkmusic", NULL, 0, NULL);
```

//背景音乐

```
mciSendString("play bkmusic repeat.", NULL, 0, NULL); //循环播放
```

```
mciSendString("open E:\\Jump.mp3 alias jpmusic.", NULL, 0, NULL);
```

```
mciSendString("Play jpmusic", NULL, 0, NULL); //仅播放一次
```

多次播放

```
mciSendString("Close jpmusic", NULL, 0, NULL); //关掉先前音乐
```

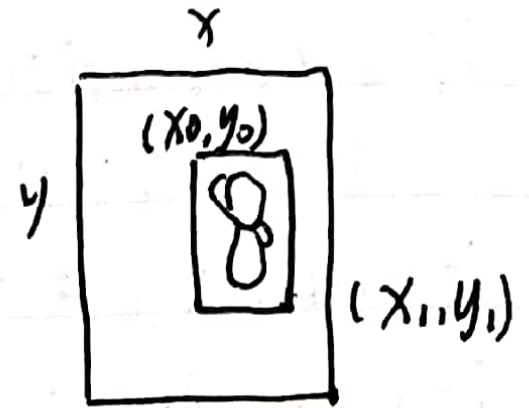
```
mciSendString("open E:\\Jump.mp3 alias jpmusic", NULL, 0, NULL); 打开
```

```
mciSendString("play jpmusic", NULL, 0, NULL); //仅播放一次
```


线型设置 `setlinestyle (style, size);`

`putimage (x, y, width, &img_bk, x0, y0);`

图片



`clearrectangle (x0, y0, x1, y1);` 清除矩形区域图形

整 \rightarrow 字符

`itoa (int number, char *str, 10);` 10表示10进制

`atoi (char *str);` 返回整型

异步输入

```
#include <window.h>
```

```
if ((GetAsyncKeyState(0x41) & 0x8000)) // a键
```

~~0x41~~

(0x44)

// d键

(VK_LEFT)/(VK_RIGHT) 向左键/向右键

0x57 W键 0x53 S键

VK_UP/VK_DOWN 上下键