

**课程作品报告（理工类）**

**2020-2021 学年第一学期**

**课程名称： 微机系统与接口技术**

**作品主题： 钢琴块**

**提交时间： 第17周**

**学院名称： 信息工程学院**

**指导教师： 舒红波**

目录

[一、程序总体介绍 3](#_Toc62297217)

[二、大体功能模块 3](#_Toc62297218)

[按键说明 3](#_Toc62297219)

[游戏说明 3](#_Toc62297220)

[三、运用到所学的知识 4](#_Toc62297221)

[1. 多进程 4](#_Toc62297222)

[2. 键盘中断 4](#_Toc62297223)

[3. 汉字点阵 4](#_Toc62297224)

[4. 音乐播放 4](#_Toc62297225)

[四、操作实现流程图 4](#_Toc62297226)

[1. 初始外部框架 4](#_Toc62297227)

[2. 当三个框到了终点线（倒数第二根红线）的时候，分别按下 a，s，d，他们会改变颜色，同时发出声音 17](#_Toc62297228)

[3. 定时器根据键盘中断改变的值去实现重新绘制 20](#_Toc62297229)

[4. 进度条和国旗显示 24](#_Toc62297230)

[5. 暂停与继续 28](#_Toc62297231)

[总体流程 33](#_Toc62297232)

[五、心得体会 34](#_Toc62297233)

[六、附件程序代码 35](#_Toc62297234)

# 一、程序总体介绍

为了将所学的知识（多进程，定时器中断，键盘中断，音乐等）都用到，

所以我选择音乐类的游戏：钢琴块（别踩白块），它是一款经典的音乐游戏，

它的游戏机制就是：点击方块，改变颜色、发出声音。以及添加了一些游戏界面。

# 二、大体功能模块

按键说明： a（弹起） 为第一个钢琴块

s（弹起） 为第二个钢琴块

d（弹起） 为第三个钢琴块

z 暂停游戏，x 继续游戏

游戏说明：

只有钢琴块到了倒数第二根线，然后按键也符合，该钢琴块变为黑色，同时发出一个音节。 按键不符合，或者钢琴块没有达到终点线，钢琴块颜色不变。

左边的为进度条，记录你点击的钢琴块数目，一共一百条。

右边会慢慢显示一个红旗。

歌曲合起来就是小星星（一首歌）。

# 三、运用到所学的知识

1. 多进程（定时器中断）

时间片轮转（顺序）的方式，让三个不同颜色方框分开往下，还有音乐进程和停止进程。

1. 键盘中断

对键盘缓存进行读取，如果为a，s，d，z，x，则对其进行相应的函数操作

1. 汉字点阵

在画图模式中，用点阵的方式输出汉字。

1. 音乐播放

开关扬声器，播放七大音节，形成一首歌曲

# 四、操作实现流程图

1. 初始外部框架（静态的，初始化就固定了的）

通过start去调用outframe行数，去执行一些画图函数

这些画图函数是通过传递参数进行处理的。而且在outframe进行压栈、出栈操作，就不会导致参数变形问题。（备注：画图函数的功能都在最后一行显示）

外部框函数：

outframe:

     push ax

     push bx

     push cx

     push si

     mov ax,0

     mov bx,0

     mov cx,320

     mov si,1

     call DrawLineOfH

     mov ax,0

     mov bx,0

     mov cx,200

     mov si,1

     call DrawLineOfS

     mov ax,0

     mov bx,319

     mov cx,200

     mov si,1

     call DrawLineOfS

     mov ax,0

     mov bx,303

     mov cx,16

     mov si,1

     call DrawLineOfS

     mov ax,0

     mov bx,287

     mov cx,16

     mov si,1

     call DrawLineOfS

     mov ax,0

     mov bx,271

     mov cx,16

     mov si,1

     call DrawLineOfS

     mov ax,16

     mov bx,0

     mov cx,320

     mov si,1

     call DrawLineOfH

     mov ax,199

     mov bx,0

     mov cx,320

     mov si,1

     call DrawLineOfH

     mov ax,16

     mov bx,307

     mov cx,184

     mov si,1

     call DrawLineOfS

     mov ax,8

     mov bx,311

     mov cx,5

     mov si,3

     call DrawLineOfx

     ;关闭按键

     mov ax,8

     mov bx,275

     mov cx,8

     mov si,3

     call DrawLineOfH

     mov ax,179

     mov bx,0

     mov cx,320

     mov si,1

     call DrawLineOfH

     mov ax,30

     mov bx,15

     mov cx,25

     mov si,5

     call DrawLineOfH

     mov ax,130

     mov bx,15

     mov cx,25

     mov si,5

     call DrawLineOfH

     mov ax,30

     mov bx,15

     mov cx,100

     mov si,5

     call DrawLineOfS

     mov ax,30

     mov bx,40

     mov cx,100

     mov si,5

     call DrawLineOfS

     ;上面四个积分框

     pop ax

     pop bx

     pop cx

     pop si

ret

;外部框绘制

clearfrist:

       push ax

       push di

       mov al,4

       mov di,35

CF:

        mov byte [es:bx],al

        inc bx

        dec di

        cmp di,0

        ja  CF

      pop ax

      pop di

ret

;清空前一行

DrawPianeframe:

    push di

    push si

    mov di,35    ;宽

    mov si,55    ;长

DPFsi:

       mov bp,di

DPFdi:

        cmp bx,5120

        jb  DPF\_head

DPF\_head\_back:

        cmp bx,57280

        ja  DPF\_fin

        mov byte [es:bx],al

        inc bx

        dec di

        cmp di,0

        ja  DPFdi

     mov di,bp

     sub bx,di

     add bx,320

     dec si

     cmp si,0

     ja  DPFsi

     jmp DPF\_fin

DPF\_head:

     add bx,320

     dec si

     jmp DPFdi

DPF\_fin:

    pop di

    pop si

ret

;钢琴块

DrawLineOfx:                        ; si: 颜色 ax:行数 bx:列数 cx:大小

     push bp

     mov bp,colornumber

     add bp,si

     mov dx,320

     mul dx

     add bx,ax

     mov ax,cx

     mov dx,320

     mul dx

     mov si,ax

     mov al,[bp]

     mov byte [es:bx],al

     mov di,bx

DXe: mov bx,di

     add bx,si

     add bx,cx

     mov byte [es:bx],al

     mov bx,di

     add bx,si

     sub bx,cx

     mov byte [es:bx],al

     mov bx,di

     sub bx,si

     sub bx,cx

     mov byte [es:bx],al

     mov bx,di

     sub bx,si

     add bx,cx

     mov byte [es:bx],al

     dec cx

     sub si,320

     cmp cx,0

     ja  DXe

     pop bp

ret

;画叉

DrawLineOfFso:                        ;si:color  ax bx   cx:长   di:宽    空心框

    push bp

    push si

    push ax

    push bx

    push cx

    push di

    mov  di,1

    mov bp,FSO

Dfso:

     mov si,[bp]

     mov ax,[bp+2]

     mov bx,[bp+4]

     mov cx,[bp+6]

     call DrawLineOfH

     mov si,[bp]

     mov ax,[bp+2]

     mov bx,[bp+4]

     mov cx,[bp+8]

     call DrawLineOfS

     mov si,[bp]

     mov ax,[bp+2]

     mov bx,[bp+4]

     mov cx,[bp+6]

     add bx,cx

     mov cx,[bp+8]

     call DrawLineOfS

     mov si,[bp]

     mov ax,[bp+2]

     mov bx,[bp+4]

     mov cx,[bp+8]

     add ax,cx

     mov cx,[bp+6]

     call DrawLineOfH

     dec di

     add bp,10

     cmp di,0

     ja  Dfso

    pop bp

    pop si

    pop ax

    pop bx

    pop cx

    pop di

ret

DrawLineOfH:       ;ax 为x坐标 ，cx为y坐标

     push bp

     mov bp,colornumber

     add bp,si

     mov dx,320

     mul dx

     add bx,ax

     mov al,[bp]

DHe:   mov byte [es:bx],al

     inc bx

     dec cx

     cmp cx,0

     ja  DHe

     pop bp

ret

;画横线

DrawLineOfS:

     push bp

     mov bp,colornumber

     add bp,si

     mov dx,320

     mul dx

     add bx,ax

     mov al,[bp]

DSe: mov byte [es:bx],al

     add bx,320

     dec cx

     cmp cx,0

     ja  DSe

     pop bp

ret

;画竖线

backcolorwrite:

    push bx

    push ax

    mov bx,0

    mov al,0

    mov byte [es:bx],al

    pop bx

    pop ax

ret

;背景颜色

Drawnamefun:

    push di       ;要写的名字

    push ax       ;al 颜色色号

    push bx       ;起始坐标

    push bp       ;行数

    push si       ;列数

    mov di,wjm

    mov al,1

    mov bx,57600

    mov bp,0        ;行数

    mov si,0        ;列数

Dback:

    cmp si,20

    jae Drawnamefun\_close   ;多于列数，结束函数

    cmp bp,48

    jae change\_si   ;多余行数，换行

    jmp draw       ;两者都满足，绘图

draw\_back:

    add bp,1

    jmp Dback        ;循环

change\_si:

    add si,1

    sub bp,48

    add bx,272

    jmp Dback

draw:

d1: mov cx,0

    cmp [di],cx

    je draw\_close

    mov byte[es:bx],al

draw\_close:

    add di,2        ;dw+2

    add bx,1

    jmp draw\_back

Drawnamefun\_close:

    pop di       ;要写的名字

    pop ax       ;al 颜色色号

    pop bx       ;起始坐标

    pop bp       ;行数

    pop si       ;列数

ret

;写名字

Drawwjxfun:

    push di       ;要写的名字

    push ax       ;al 颜色色号

    push bx       ;起始坐标

    push bp       ;行数

    push si       ;列数

    mov di,wjx

    mov al,4

    mov bx,16200

    mov bp,0        ;行数

    mov si,0        ;列数

Dwjxback:

    cmp si,20

    jae Drawwjxfun\_close   ;多于列数，结束函数

    cmp bp,48

    jae changewjx\_si   ;多余行数，换行

    jmp drawwjx       ;两者都满足，绘图

drawwjx\_back:

    add bp,1

    jmp Dback        ;循环

changewjx\_si:

    add si,1

    sub bp,48

    add bx,272

    jmp Dback

drawwjx:

dwjx1:  mov cx,0

    cmp [di],cx

    je draw\_close

    mov byte[es:bx],al

drawwjx\_close:

    add di,2        ;dw+2

    add bx,1

    jmp drawwjx\_back

Drawwjxfun\_close:

    pop di       ;要写的名字

    pop ax       ;al 颜色色号

    pop bx       ;起始坐标

    pop bp       ;行数

    pop si       ;列数

ret

;画五角星

colorchange:

    push dx

    push ax

    push di

    push bp

    mov di,color

    mov bp,0

cc1:

    mov dx,0x3c8

    mov al,[di+bp]

    out dx,al

    mov dx,0x3c9

    mov al,[di+1+bp]

    out dx,al

    mov al,[di+2+bp]

    out dx,al

    mov al,[di+3+bp]

    out dx,al

    add bp,4

    cmp bp,24

    jb cc1

    pop dx

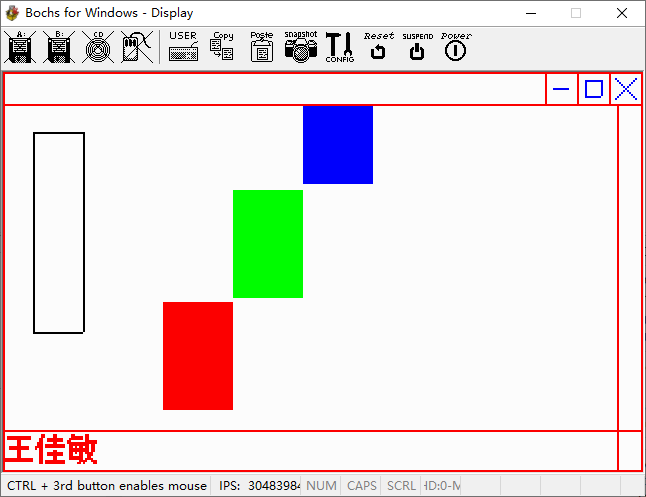
    pop ax

    pop bp

    pop di

ret

;颜色改变



1. 当三个框到了终点线（倒数第二根红线）的时候，分别按下 a，s，d，他们会改变颜色，同时发出声音。

键盘中断：

int\_key:

    push dx

    push ax

    push cx

    push bx

    mov dx,0x60

    in al,dx                  ;读取键盘缓冲区

    cmp al,0x9E

    je  change\_paino1

    cmp al,0x9F

    je  change\_paino2

    cmp al,0xA0

    je  change\_paino3

    cmp al,0xAC

    je change\_stop

    cmp al,0xAD

    je change\_continue

change\_back:

    jmp key\_iret

change\_paino3:

    mov bx,[ds:Paino+4]

    cmp bx,39680

    ja  change\_paino\_color3

    jmp change\_back

change\_paino\_color3:

     mov cx,5

     mov [ds:Painocolor+4],cx

     mov cx,1

     mov [ds:musicflag],cx

     jmp change\_back

change\_paino2:

    mov bx,[ds:Paino+2]

    cmp bx,39680

    ja  change\_paino\_color2

    jmp change\_back

change\_paino\_color2:

     mov cx,5

     mov [ds:Painocolor+2],cx

     mov cx,1

     mov [ds:musicflag],cx

     jmp change\_back

change\_paino1:

    mov bx,[ds:Paino]

    cmp bx,39680

    ja  change\_paino\_color1

    jmp change\_back

change\_paino\_color1:

     mov cx,5

     mov [ds:Painocolor],cx

     mov cx,1

     mov [ds:musicflag],cx

     jmp change\_back

change\_stop:

     sti

     mov word [ds:0x20],int\_timer5

     mov word [ds:0x22],0

     jmp key\_iret

change\_continue:

     sti

     mov word [ds:0x20],int\_timer1

     mov word [ds:0x22],0

     jmp key\_iret

key\_iret:

    mov dx,0x20               ;键盘中断结束

    mov al,0x61

    out dx,al                 ;中断结束

    pop dx

    pop ax

    pop cx

    pop bx

iret

1. 定时器根据键盘中断改变的值去实现重新绘制。

int\_timer0:

    push bx

    push ax

    push si

    push bp

    push cx

    mov bx,[ds:Paino]

    mov al,[ds:Painocolor]

    mov si,bx

    push si

    call DrawPianeframe

    pop si

    mov bx,si

    push si

    call clearfrist

    ;绘制钢琴块

    pop si

timer0\_quiter:

    add si,320

    cmp si,57280

    ja  IT0\_si

IT0\_back:

    mov [ds:Paino],si

    mov al,0x20

    mov dx,0x20

    out dx,al

    mov bp,[ds:count]

    inc bp

    mov [ds:count],bp

    cmp bp,57

    ja  start\_IT1

start\_IT1\_back:

    jmp IT0\_fin

IT0\_si:

    mov cx,1

    mov [ds:Painocolor],cx

    mov si,400

    jmp IT0\_back

start\_IT1:

     sti

     mov word [ds:0x20],int\_timer1

     mov word [ds:0x22],0

     jmp start\_IT1\_back

IT0\_fin:

    call back\_s

    pop bx

    pop ax

    pop si

    pop bp

    pop cx

iret

;第一个块移动

int\_timer1:

    push bx

    push ax

    push si

    push bp

    push cx

    mov bx,[ds:Paino+2]

    mov al,[ds:Painocolor+2]

    mov si,bx

    push si

    call DrawPianeframe

    pop si

    mov bx,si

    push si

    call clearfrist

    ;绘制钢琴块

    pop si

timer1\_quiter:

    add si,320

    cmp si,57280

    ja  IT1\_si

IT1\_back:

    mov [ds:Paino+2],si

    mov al,0x20

    mov dx,0x20

    out dx,al

    mov bp,[ds:count+2]

    inc bp

    mov [ds:count+2],bp

    cmp bp,57

    ja  start\_IT2

    cmp bp,55

    jbe start\_IT\_0

start\_IT2\_back:

    jmp IT1\_fin

IT1\_si:

    mov cx,2

    mov [ds:Painocolor+2],cx

    mov si,435

    jmp IT1\_back

start\_IT2:

     sti

     mov word [ds:0x20],int\_timer2

     mov word [ds:0x22],0

     jmp start\_IT2\_back

start\_IT\_0:

     sti

     mov word [ds:0x20],int\_timer0

     mov word [ds:0x22],0

     jmp start\_IT2\_back

IT1\_fin:

    call back\_s

    pop bx

    pop ax

    pop si

    pop bp

    pop cx

iret

;第二个块移动

int\_timer2:

    push bx

    push ax

    push si

    push bp

    push cx

    mov bx,[ds:Paino+4]

    mov al,[ds:Painocolor+4]

    mov si,bx

    push si

    call DrawPianeframe

    pop si

    mov bx,si

    push si

    call clearfrist

    pop si

timer2\_quiter:

    add si,320

    cmp si,57280

    ja  IT2\_si

IT2\_back:

    mov [ds:Paino+4],si

    mov al,0x20

    mov dx,0x20

    out dx,al

    jmp  start\_IT0

start\_IT0\_back:

    jmp IT2\_fin

IT2\_si:

    mov cx,3

    mov [ds:Painocolor+4],cx

    mov si,470

    jmp IT2\_back

start\_IT0:

     sti

     mov word [ds:0x20],int\_timer4

     mov word [ds:0x22],0

     jmp start\_IT0\_back

IT2\_fin:

    call back\_s

    pop bx

    pop ax

    pop si

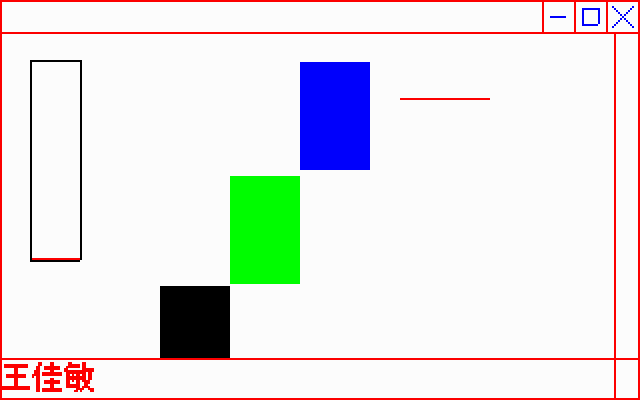
    pop bp

    pop cx

iret

;第三个块移动

贴图：



1. 进度条和国旗显示

原理上是差不多的，差别在于国旗中有图片显示。

wjx\_add:

pusha

    mov cx,[ds:wjx\_site]

    cmp cx,25

    ja  wjx\_add\_stop

    mov ax,49

    add ax,cx

    mov bx,200

    push cx

    mov cx,45

    mov si,1

    call DrawLineOfH

    pop  cx

    inc cx

    mov [ds:wjx\_site],cx

wjx\_add\_stop:

popa

pusha

call Drawwjxfun

popa

ret

score\_add:

pusha

    mov cx,[ds:score]

    cmp cx,100

    ja  score\_add\_stop

    mov ax,129

    sub ax,cx

    mov bx,16

    push cx

    mov cx,24

    mov si,1

    call DrawLineOfH

    pop  cx

    inc cx

    mov [ds:score],cx

score\_add\_stop:

popa

ret

Drawwjxfun:

    push di       ;要写的

    push ax       ;al 颜色色号

    push bx       ;起始坐标

    push bp       ;行数

    push si       ;列数

    mov di,wjx

    mov al,4

    mov bx,16200

    mov bp,0        ;行数

    mov si,0        ;列数

Dwjxback:

    cmp si,20

    jae Drawwjxfun\_close   ;多于列数，结束函数

    cmp bp,48

    jae changewjx\_si   ;多余行数，换行

    jmp drawwjx       ;两者都满足，绘图

drawwjx\_back:

    add bp,1

    jmp Dback        ;循环

changewjx\_si:

    add si,1

    sub bp,48

    add bx,272

    jmp Dback

drawwjx:

dwjx1:  mov cx,0

    cmp [di],cx

    je draw\_close

    mov byte[es:bx],al

drawwjx\_close:

    add di,2        ;dw+2

    add bx,1

    jmp drawwjx\_back

Drawwjxfun\_close:

    pop di       ;要写的

    pop ax       ;al 颜色色号

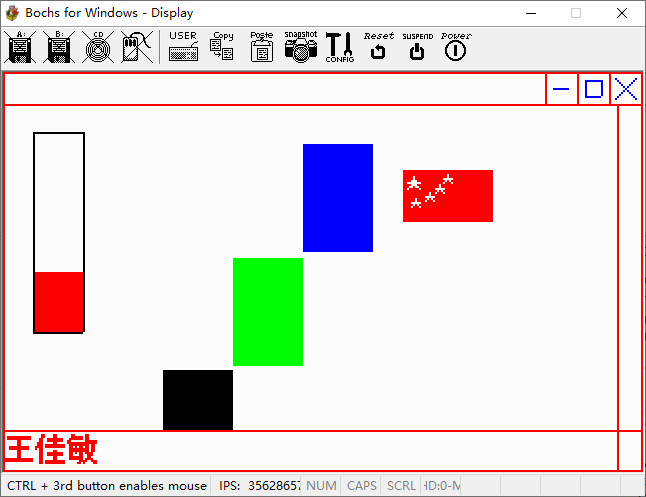
    pop bx       ;起始坐标

    pop bp       ;行数

    pop si       ;列数

ret

;画五角星



1. 暂停与继续

按z停止，按x继续（靠按键转换）

空进程停止

change\_stop:

     sti

     mov word [ds:0x20],int\_timer5

     mov word [ds:0x22],0

     jmp key\_iret

change\_continue:

     sti

     mov word [ds:0x20],int\_timer1

     mov word [ds:0x22],0

     jmp key\_iret

int\_timer5:

   push ax

   push dx

   push di

    mov al,0x20

    mov dx,0x20

    out dx,al

   pop ax

   pop dx

   pop di

iret

6．音乐播放

每次按对一个键就会响一个音节，然后连起来就是一首歌曲。

littlestar dw 1,1,5,5,6,6,5,5,4,4,3,3,2,2,1,1,5,5,4,4,3,3,2,2,5,5,4,4,3,3,2,2,1,1,5,5,6,6,5,5,4,4,3,3,2,2,1,1

musiccount dw 0

musicflag  dw 0

music\_part:

    scale1:

        push bx

        push ax

        mov bx , [ds:table]

        call tosound

        pop  bx

        pop  ax

        jmp music\_change

    scale2:

        push bx

        push ax

        mov bx , [ds:table+2]

        call tosound

        pop  bx

        pop  ax

        jmp music\_change

    scale3:

        push bx

        push ax

        mov bx , [ds:table+4]

        call tosound

        pop  bx

        pop  ax

        jmp music\_change

    scale4:

        push bx

        push ax

        mov bx , [ds:table+6]

        call tosound

        pop  bx

        pop  ax

        jmp music\_change

    scale5:

        push bx

        push ax

        mov bx , [ds:table+8]

        call tosound

        pop  bx

        pop  ax

        jmp music\_change

    scale6:

        push bx

        push ax

        mov bx , [ds:table+10]

        call tosound

        pop  bx

        pop  ax

        jmp music\_change

    scale7:

        push bx

        push ax

        mov bx , [ds:table+12]

        call tosound

        pop  bx

        pop  ax

        jmp music\_change

tosound:

    mov [ds:nowUseHz] , bx

    call initSound

    call delay1

    call TurnOnSoundPort

    call delay2

    call closeSoundPort

ret

initSound:

    mov  al, 10110110B

    out  43H, al

    call TurnHzToHex

    mov  ax, [ds:changeHz]

    out  42H, al

    mov  al, ah

    out  42H, al

ret

; 打开扬声器

TurnOnSoundPort:

    push ax

    in al , 61H

    or al , 00000011B

    out 61H , al

    pop ax

ret

;写一个关闭扬声器

closeSoundPort:

    push ax

    in   al,61H

    and  al,11111100B

    out  61H,al

    pop ax

ret

TurnHzToHex:

    ; ;设置频率

    push dx

    push ax

    push di

    mov dx,12H

    mov ax,34DEH

    mov di,[ds:nowUseHz]

    div di

    mov [ds:changeHz] , ax

    pop dx

    pop ax

    pop di

ret

delay1:

    push cx

    push ax

    push dx

    mov  cx, 00H

    mov  dx, 4240H

    mov  ah, 86H

    int  15H

    pop dx

    pop ax

    pop cx

ret

delay2:

    push cx

    push ax

    push dx

    mov  cx, 0FH

    mov  dx, 4240H

    mov  ah, 86H

    int  15H

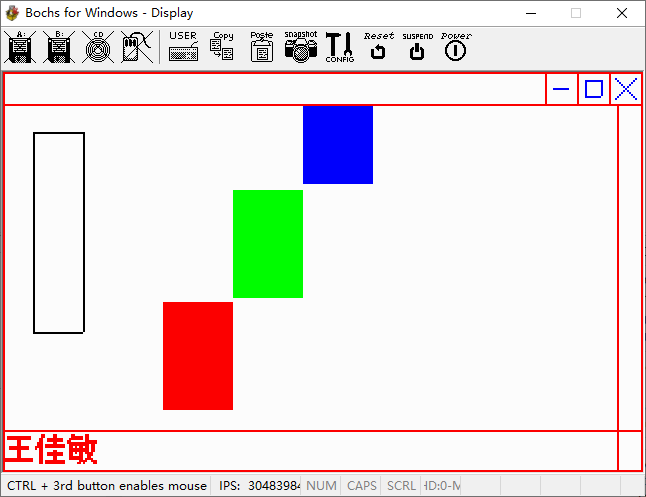
    pop dx

    pop ax

    pop cx

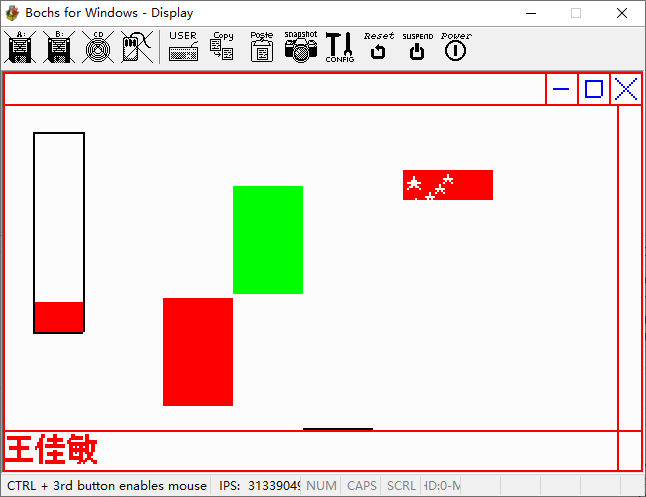
ret

总体流程：

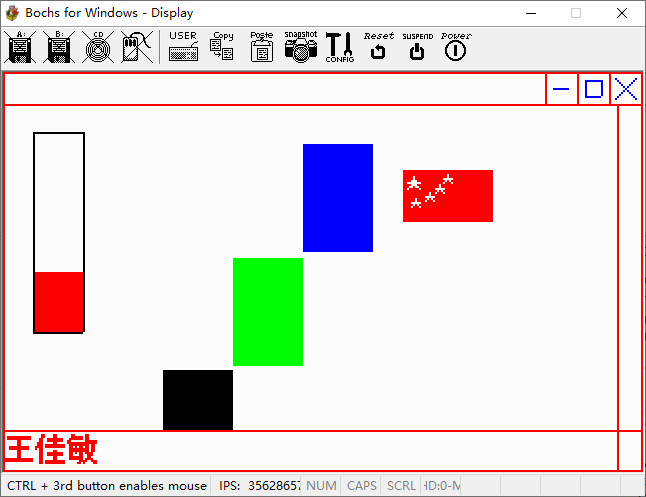


按a按键，第一块钢琴块变黑，进度条加1.

钢琴块进度：右边逐渐显示出国旗



最终显示：



# 五、心得体会

看起来比较简单的功能，真正做起来还是麻烦的。微机运用到的都是底层的东西，与其他的课程课设不同，别的都已经分装的很好，很简便。但是别的课程也会出现一个问题，就是在哪有，怎么用，感觉就是不断地在套层。虽然说操作上简便了，但是在原理和专利性的角度来说，是有极大劣势的。因此，我觉得微机这门课很大程度的让我了解了一些计算机内部的原理和机械，存储器，寄存器，cpu、扬声器等等，这对未来写代码有很大的好处。

老师也问过我，对课程的评价。我以为老师的第一二节都是在交历史课，没太多意思。但是直到后来，老师讲述了经历才明白，这就是为了培养对计算机专业的乐趣。还有就是由于性格原因，虽然我能很自信的与人聊天，但是我不会去主动去提问。这是我的一个缺陷吧，

我的资源或者提问都是向一些比较熟悉的人，以后要加以改正。

不管如何，在此谢谢老师。

# 六、附件程序代码

org 0x8400

jmp start    ;跳转

color  db 1,255,0,0        ;红色

       db 2,0,255,0        ;绿色

       db 3,0,0,255        ;蓝色

       db 4,255,255,255        ;白色

       db 5,0,0,0          ;黑色

       db 0,255,255,255    ;背景设置为白色

;颜色

color\_2 db 10,255,255,0

;渐变颜色

colornumber db 0,1,2,3,4,5

FSO     dw 3,4,291,8,8   ;缩小框

Paino   dw 400,435,470

;Paino   dw -2800,-2765,-2730

Painocolor dw 1,2,3,4

wjm dw  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,1,1,0,0,0,0,0

    dw  0,0,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,1,1,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,0,1,1,0,0,0,0,0

    dw  0,0,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,1,1,0,0,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,0,1,1,0,0,0,0,0

    dw  0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,1,1,0,0,1,1,1,1,1,1,1,1,0,0,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,0

    dw  0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,1,1,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0

    dw  0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,0,1,1,0,0

    dw  0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,0,1,1,0,1,0,1,1,0,1,1,0,1,1,0,0

    dw  0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,1,1,0,0

    dw  0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,1,1,0,0

    dw  0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,1,1,0,0,1,1,1,1,1,1,1,1,0,0,0,1,1,0,1,0,1,1,0,1,1,1,1,0,0,0

    dw  0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,1,1,0,0,1,1,1,1,1,1,1,1,0,0,0,1,1,0,0,0,1,1,0,0,1,1,1,0,0,0

    dw  0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,1,1,0,0,0,0

    dw  0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,1,1,1,1,1,0,1,1,1,1,0,0,0

    dw  0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,0,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,1,1,1,1,0,0,1,1,1,0

    dw  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,1,1,0,1,0,0,0,0,1,1,0

    dw  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

;名字 20

wjx dw  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,1,1,0,1,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dw  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

count dw 0,0

nowUseHz dw 900

changeHz dw 0

table   dw  262         ;k1 发 do    的音，261.6HZ

        dw  294         ;k2 发 re    的音，293.6HZ

        dw  330         ;k3 发 mi    的音，329.6HZ

        dw  349         ;k4 发 fa    的音，349.2HZ

        dw  392         ;k5 发 sol   的音，392HZ

        dw  440         ;k6 发 la    的音，440HZ

        dw  494         ;k7 发 si    的音，493.8HZ

; ------------------------------------------------

littlestar dw 1,1,5,5,6,6,5,5,4,4,3,3,2,2,1,1,5,5,4,4,3,3,2,2,5,5,4,4,3,3,2,2,1,1,5,5,6,6,5,5,4,4,3,3,2,2,1,1

musiccount dw 0

musicflag  dw 0

score      dw 0

wjx\_site   dw 0

start:

    mov ah,00h

    mov al,13h

    int 10h

    mov ax,0x0a000

    mov es,ax

    ;初始化

    call colorchange

    ;自定义颜色

    call backcolorwrite

    ;设置背景颜色

    call Drawnamefun

    ;写名字

    call DrawLineOfFso

    ;绘空心框

    call outframe

    ;绘制外部框架

     sti

     mov word [ds:0x20],int\_timer0

     mov word [ds:0x22],0

     sti

     mov word [ds:0x24],int\_key

     mov word [ds:0x26],0

exit:

    jmp exit

    ;退出

int\_key:

    push dx

    push ax

    push cx

    push bx

    mov dx,0x60

    in al,dx                  ;读取键盘缓冲区

    cmp al,0x9E

    je  change\_paino1

    cmp al,0x9F

    je  change\_paino2

    cmp al,0xA0

    je  change\_paino3

    cmp al,0xAC

    je change\_stop

    cmp al,0xAD

    je change\_continue

change\_back:

    jmp key\_iret

change\_paino3:

    mov bx,[ds:Paino+4]

    cmp bx,39680

    ja  change\_paino\_color3

    jmp change\_back

change\_paino\_color3:

     mov cx,5

     mov [ds:Painocolor+4],cx

     mov cx,1

     mov [ds:musicflag],cx

     jmp change\_back

change\_paino2:

    mov bx,[ds:Paino+2]

    cmp bx,39680

    ja  change\_paino\_color2

    jmp change\_back

change\_paino\_color2:

     mov cx,5

     mov [ds:Painocolor+2],cx

     mov cx,1

     mov [ds:musicflag],cx

     jmp change\_back

change\_paino1:

    mov bx,[ds:Paino]

    cmp bx,39680

    ja  change\_paino\_color1

    jmp change\_back

change\_paino\_color1:

     mov cx,5

     mov [ds:Painocolor],cx

     mov cx,1

     mov [ds:musicflag],cx

     jmp change\_back

change\_stop:

     sti

     mov word [ds:0x20],int\_timer5

     mov word [ds:0x22],0

     jmp key\_iret

change\_continue:

     sti

     mov word [ds:0x20],int\_timer1

     mov word [ds:0x22],0

     jmp key\_iret

key\_iret:

    mov dx,0x20               ;键盘中断结束

    mov al,0x61

    out dx,al                 ;中断结束

    pop dx

    pop ax

    pop cx

    pop bx

iret

music:

   push cx                   ;是否播放音乐

   push bx                   ;获取音乐位数

   push ax                   ;获取音符

   push dx

   push si

   mov cx,[ds:musicflag]

   cmp cx,0

   je  music\_ret

   mov bx,[ds:musiccount]

   mov si,bx

   mov ax,[ds:littlestar+bx]

   call score\_add

   call wjx\_add

music\_play:

    cmp ax,1

    je  scale1

    cmp ax,2

    je  scale2

    cmp ax,3

    je  scale3

    cmp ax,4

    je  scale4

    cmp ax,5

    je  scale5

    cmp ax,6

    je  scale6

    cmp ax,7

    je  scale7

music\_change:

   mov cx,0

   mov [ds:musicflag],cx

   add si,2

   cmp si,96

   ja  change\_music\_si

change\_music\_si\_back:

   mov [ds:musiccount],si

   jmp music\_ret

change\_music\_si:

  mov si,0

  jmp change\_music\_si\_back

 music\_ret:

   pop cx

   pop bx

   pop ax

   pop dx

   pop si

ret

wjx\_add:

pusha

    mov cx,[ds:wjx\_site]

    cmp cx,25

    ja  wjx\_add\_stop

    mov ax,49

    add ax,cx

    mov bx,200

    push cx

    mov cx,45

    mov si,1

    call DrawLineOfH

    pop  cx

    inc cx

    mov [ds:wjx\_site],cx

wjx\_add\_stop:

popa

pusha

call Drawwjxfun

popa

ret

score\_add:

pusha

    mov cx,[ds:score]

    cmp cx,100

    ja  score\_add\_stop

    mov ax,129

    sub ax,cx

    mov bx,16

    push cx

    mov cx,24

    mov si,1

    call DrawLineOfH

    pop  cx

    inc cx

    mov [ds:score],cx

score\_add\_stop:

popa

ret

outframe:

     push ax

     push bx

     push cx

     push si

     mov ax,0

     mov bx,0

     mov cx,320

     mov si,1

     call DrawLineOfH

     mov ax,0

     mov bx,0

     mov cx,200

     mov si,1

     call DrawLineOfS

     mov ax,0

     mov bx,319

     mov cx,200

     mov si,1

     call DrawLineOfS

     mov ax,0

     mov bx,303

     mov cx,16

     mov si,1

     call DrawLineOfS

     mov ax,0

     mov bx,287

     mov cx,16

     mov si,1

     call DrawLineOfS

     mov ax,0

     mov bx,271

     mov cx,16

     mov si,1

     call DrawLineOfS

     mov ax,16

     mov bx,0

     mov cx,320

     mov si,1

     call DrawLineOfH

     mov ax,199

     mov bx,0

     mov cx,320

     mov si,1

     call DrawLineOfH

     mov ax,16

     mov bx,307

     mov cx,184

     mov si,1

     call DrawLineOfS

     mov ax,8

     mov bx,311

     mov cx,5

     mov si,3

     call DrawLineOfx

     ;关闭按键

     mov ax,8

     mov bx,275

     mov cx,8

     mov si,3

     call DrawLineOfH

     mov ax,179

     mov bx,0

     mov cx,320

     mov si,1

     call DrawLineOfH

     mov ax,30

     mov bx,15

     mov cx,25

     mov si,5

     call DrawLineOfH

     mov ax,130

     mov bx,15

     mov cx,25

     mov si,5

     call DrawLineOfH

     mov ax,30

     mov bx,15

     mov cx,100

     mov si,5

     call DrawLineOfS

     mov ax,30

     mov bx,40

     mov cx,100

     mov si,5

     call DrawLineOfS

     ;上面四个积分框

     pop ax

     pop bx

     pop cx

     pop si

ret

;外部框绘制

back\_s:

     pusha

     mov ax,16

     mov bx,0

     mov cx,320

     mov si,1

     call DrawLineOfH

     popa

ret

int\_timer0:

    push bx

    push ax

    push si

    push bp

    push cx

    mov bx,[ds:Paino]

    mov al,[ds:Painocolor]

    mov si,bx

    push si

    call DrawPianeframe

    pop si

    mov bx,si

    push si

    call clearfrist

    ;绘制钢琴块

    pop si

timer0\_quiter:

    add si,320

    cmp si,57280

    ja  IT0\_si

IT0\_back:

    mov [ds:Paino],si

    mov al,0x20

    mov dx,0x20

    out dx,al

    mov bp,[ds:count]

    inc bp

    mov [ds:count],bp

    cmp bp,57

    ja  start\_IT1

start\_IT1\_back:

    jmp IT0\_fin

IT0\_si:

    mov cx,1

    mov [ds:Painocolor],cx

    mov si,400

    jmp IT0\_back

start\_IT1:

     sti

     mov word [ds:0x20],int\_timer1

     mov word [ds:0x22],0

     jmp start\_IT1\_back

IT0\_fin:

    call back\_s

    pop bx

    pop ax

    pop si

    pop bp

    pop cx

iret

;第一个块移动

int\_timer1:

    push bx

    push ax

    push si

    push bp

    push cx

    mov bx,[ds:Paino+2]

    mov al,[ds:Painocolor+2]

    mov si,bx

    push si

    call DrawPianeframe

    pop si

    mov bx,si

    push si

    call clearfrist

    ;绘制钢琴块

    pop si

timer1\_quiter:

    add si,320

    cmp si,57280

    ja  IT1\_si

IT1\_back:

    mov [ds:Paino+2],si

    mov al,0x20

    mov dx,0x20

    out dx,al

    mov bp,[ds:count+2]

    inc bp

    mov [ds:count+2],bp

    cmp bp,57

    ja  start\_IT2

    cmp bp,55

    jbe start\_IT\_0

start\_IT2\_back:

    jmp IT1\_fin

IT1\_si:

    mov cx,2

    mov [ds:Painocolor+2],cx

    mov si,435

    jmp IT1\_back

start\_IT2:

     sti

     mov word [ds:0x20],int\_timer2

     mov word [ds:0x22],0

     jmp start\_IT2\_back

start\_IT\_0:

     sti

     mov word [ds:0x20],int\_timer0

     mov word [ds:0x22],0

     jmp start\_IT2\_back

IT1\_fin:

    call back\_s

    pop bx

    pop ax

    pop si

    pop bp

    pop cx

iret

;第二个块移动

int\_timer2:

    push bx

    push ax

    push si

    push bp

    push cx

    mov bx,[ds:Paino+4]

    mov al,[ds:Painocolor+4]

    mov si,bx

    push si

    call DrawPianeframe

    pop si

    mov bx,si

    push si

    call clearfrist

    pop si

timer2\_quiter:

    add si,320

    cmp si,57280

    ja  IT2\_si

IT2\_back:

    mov [ds:Paino+4],si

    mov al,0x20

    mov dx,0x20

    out dx,al

    jmp  start\_IT0

start\_IT0\_back:

    jmp IT2\_fin

IT2\_si:

    mov cx,3

    mov [ds:Painocolor+4],cx

    mov si,470

    jmp IT2\_back

start\_IT0:

     sti

     mov word [ds:0x20],int\_timer4

     mov word [ds:0x22],0

     jmp start\_IT0\_back

IT2\_fin:

    call back\_s

    pop bx

    pop ax

    pop si

    pop bp

    pop cx

iret

;第三个块移动

int\_timer4:

   push ax

   push dx

   push di

     call music

     sti

     mov word [ds:0x20],int\_timer0

     mov word [ds:0x22],0

    mov al,0x20

    mov dx,0x20

    out dx,al

   pop ax

   pop dx

   pop di

iret

int\_timer5:

   push ax

   push dx

   push di

    mov al,0x20

    mov dx,0x20

    out dx,al

   pop ax

   pop dx

   pop di

iret

clearfrist:

       push ax

       push di

       mov al,4

       mov di,35

CF:

        mov byte [es:bx],al

        inc bx

        dec di

        cmp di,0

        ja  CF

      pop ax

      pop di

ret

;清空前一行

DrawPianeframe:

    push di

    push si

    mov di,35    ;宽

    mov si,55    ;长

DPFsi:

       mov bp,di

DPFdi:

        cmp bx,5120

        jb  DPF\_head

DPF\_head\_back:

        cmp bx,57280

        ja  DPF\_fin

        mov byte [es:bx],al

        inc bx

        dec di

        cmp di,0

        ja  DPFdi

     mov di,bp

     sub bx,di

     add bx,320

     dec si

     cmp si,0

     ja  DPFsi

     jmp DPF\_fin

DPF\_head:

     add bx,320

     dec si

     jmp DPFdi

DPF\_fin:

    pop di

    pop si

ret

;钢琴块

DrawLineOfx:                        ; si: 颜色 ax:行数 bx:列数 cx:大小

     push bp

     mov bp,colornumber

     add bp,si

     mov dx,320

     mul dx

     add bx,ax

     mov ax,cx

     mov dx,320

     mul dx

     mov si,ax

     mov al,[bp]

     mov byte [es:bx],al

     mov di,bx

DXe: mov bx,di

     add bx,si

     add bx,cx

     mov byte [es:bx],al

     mov bx,di

     add bx,si

     sub bx,cx

     mov byte [es:bx],al

     mov bx,di

     sub bx,si

     sub bx,cx

     mov byte [es:bx],al

     mov bx,di

     sub bx,si

     add bx,cx

     mov byte [es:bx],al

     dec cx

     sub si,320

     cmp cx,0

     ja  DXe

     pop bp

ret

;画叉

DrawLineOfFso:                        ;si:color  ax bx   cx:长   di:宽    空心框

    push bp

    push si

    push ax

    push bx

    push cx

    push di

    mov  di,1

    mov bp,FSO

Dfso:

     mov si,[bp]

     mov ax,[bp+2]

     mov bx,[bp+4]

     mov cx,[bp+6]

     call DrawLineOfH

     mov si,[bp]

     mov ax,[bp+2]

     mov bx,[bp+4]

     mov cx,[bp+8]

     call DrawLineOfS

     mov si,[bp]

     mov ax,[bp+2]

     mov bx,[bp+4]

     mov cx,[bp+6]

     add bx,cx

     mov cx,[bp+8]

     call DrawLineOfS

     mov si,[bp]

     mov ax,[bp+2]

     mov bx,[bp+4]

     mov cx,[bp+8]

     add ax,cx

     mov cx,[bp+6]

     call DrawLineOfH

     dec di

     add bp,10

     cmp di,0

     ja  Dfso

    pop bp

    pop si

    pop ax

    pop bx

    pop cx

    pop di

ret

DrawLineOfH:       ;ax 为x坐标 ，cx为y坐标

     push bp

     mov bp,colornumber

     add bp,si

     mov dx,320

     mul dx

     add bx,ax

     mov al,[bp]

DHe:   mov byte [es:bx],al

     inc bx

     dec cx

     cmp cx,0

     ja  DHe

     pop bp

ret

;画横线

DrawLineOfS:

     push bp

     mov bp,colornumber

     add bp,si

     mov dx,320

     mul dx

     add bx,ax

     mov al,[bp]

DSe: mov byte [es:bx],al

     add bx,320

     dec cx

     cmp cx,0

     ja  DSe

     pop bp

ret

;画竖线

backcolorwrite:

    push bx

    push ax

    mov bx,0

    mov al,0

    mov byte [es:bx],al

    pop bx

    pop ax

ret

;背景颜色

Drawnamefun:

    push di       ;要写的名字

    push ax       ;al 颜色色号

    push bx       ;起始坐标

    push bp       ;行数

    push si       ;列数

    mov di,wjm

    mov al,1

    mov bx,57600

    mov bp,0        ;行数

    mov si,0        ;列数

Dback:

    cmp si,20

    jae Drawnamefun\_close   ;多于列数，结束函数

    cmp bp,48

    jae change\_si   ;多余行数，换行

    jmp draw       ;两者都满足，绘图

draw\_back:

    add bp,1

    jmp Dback        ;循环

change\_si:

    add si,1

    sub bp,48

    add bx,272

    jmp Dback

draw:

d1: mov cx,0

    cmp [di],cx

    je draw\_close

    mov byte[es:bx],al

draw\_close:

    add di,2        ;dw+2

    add bx,1

    jmp draw\_back

Drawnamefun\_close:

    pop di       ;要写的名字

    pop ax       ;al 颜色色号

    pop bx       ;起始坐标

    pop bp       ;行数

    pop si       ;列数

ret

;写名字

Drawwjxfun:

    push di       ;要写的名字

    push ax       ;al 颜色色号

    push bx       ;起始坐标

    push bp       ;行数

    push si       ;列数

    mov di,wjx

    mov al,4

    mov bx,16200

    mov bp,0        ;行数

    mov si,0        ;列数

Dwjxback:

    cmp si,20

    jae Drawwjxfun\_close   ;多于列数，结束函数

    cmp bp,48

    jae changewjx\_si   ;多余行数，换行

    jmp drawwjx       ;两者都满足，绘图

drawwjx\_back:

    add bp,1

    jmp Dback        ;循环

changewjx\_si:

    add si,1

    sub bp,48

    add bx,272

    jmp Dback

drawwjx:

dwjx1:  mov cx,0

    cmp [di],cx

    je draw\_close

    mov byte[es:bx],al

drawwjx\_close:

    add di,2        ;dw+2

    add bx,1

    jmp drawwjx\_back

Drawwjxfun\_close:

    pop di       ;要写的名字

    pop ax       ;al 颜色色号

    pop bx       ;起始坐标

    pop bp       ;行数

    pop si       ;列数

ret

;画五角星

colorchange:

    push dx

    push ax

    push di

    push bp

    mov di,color

    mov bp,0

cc1:

    mov dx,0x3c8

    mov al,[di+bp]

    out dx,al

    mov dx,0x3c9

    mov al,[di+1+bp]

    out dx,al

    mov al,[di+2+bp]

    out dx,al

    mov al,[di+3+bp]

    out dx,al

    add bp,4

    cmp bp,24

    jb cc1

    pop dx

    pop ax

    pop bp

    pop di

ret

;颜色改变

music\_part:

    scale1:

        push bx

        push ax

        mov bx , [ds:table]

        call tosound

        pop  bx

        pop  ax

        jmp music\_change

    scale2:

        push bx

        push ax

        mov bx , [ds:table+2]

        call tosound

        pop  bx

        pop  ax

        jmp music\_change

    scale3:

        push bx

        push ax

        mov bx , [ds:table+4]

        call tosound

        pop  bx

        pop  ax

        jmp music\_change

    scale4:

        push bx

        push ax

        mov bx , [ds:table+6]

        call tosound

        pop  bx

        pop  ax

        jmp music\_change

    scale5:

        push bx

        push ax

        mov bx , [ds:table+8]

        call tosound

        pop  bx

        pop  ax

        jmp music\_change

    scale6:

        push bx

        push ax

        mov bx , [ds:table+10]

        call tosound

        pop  bx

        pop  ax

        jmp music\_change

    scale7:

        push bx

        push ax

        mov bx , [ds:table+12]

        call tosound

        pop  bx

        pop  ax

        jmp music\_change

tosound:

    mov [ds:nowUseHz] , bx

    call initSound

    call delay1

    call TurnOnSoundPort

    call delay2

    call closeSoundPort

ret

initSound:

    mov  al, 10110110B

    out  43H, al

    call TurnHzToHex

    mov  ax, [ds:changeHz]

    out  42H, al

    mov  al, ah

    out  42H, al

ret

; 打开扬声器

TurnOnSoundPort:

    push ax

    in al , 61H

    or al , 00000011B

    out 61H , al

    pop ax

ret

;写一个关闭扬声器

closeSoundPort:

    push ax

    in   al,61H

    and  al,11111100B

    out  61H,al

    pop ax

ret

TurnHzToHex:

    ; ;设置频率

    push dx

    push ax

    push di

    mov dx,12H

    mov ax,34DEH

    mov di,[ds:nowUseHz]

    div di

    mov [ds:changeHz] , ax

    pop dx

    pop ax

    pop di

ret

delay1:

    push cx

    push ax

    push dx

    mov  cx, 00H

    mov  dx, 4240H

    mov  ah, 86H

    int  15H

    pop dx

    pop ax

    pop cx

ret

delay2:

    push cx

    push ax

    push dx

    mov  cx, 0FH

    mov  dx, 4240H

    mov  ah, 86H

    int  15H

    pop dx

    pop ax

    pop cx

ret