《汇编语言》期末项目说明文档

项目名称：简易绘图板

- 2254269 吴昊泽
- 指导教师：王冬青

一、项目概述

这是一个基于汇编语言编写的简单绘图板程序，它允许用户在320 x 200的VGA模式下，通过鼠标在屏幕上绘制图形。程序支持基本的鼠标操作来绘制像素点，并提供了颜色切换和橡皮擦功能。

虽然现代编程语言和开发环境提供了丰富的图形库和工具，但理解底层绘图原理和硬件交互仍然是计算机科学的重要组成部分。因此，开发一个简单的绘图板程序，不仅可以帮助学习汇编语言的基本语法，还能深入理解计算机图形学和硬件操作。

二、项目功能点

1. 初始设置

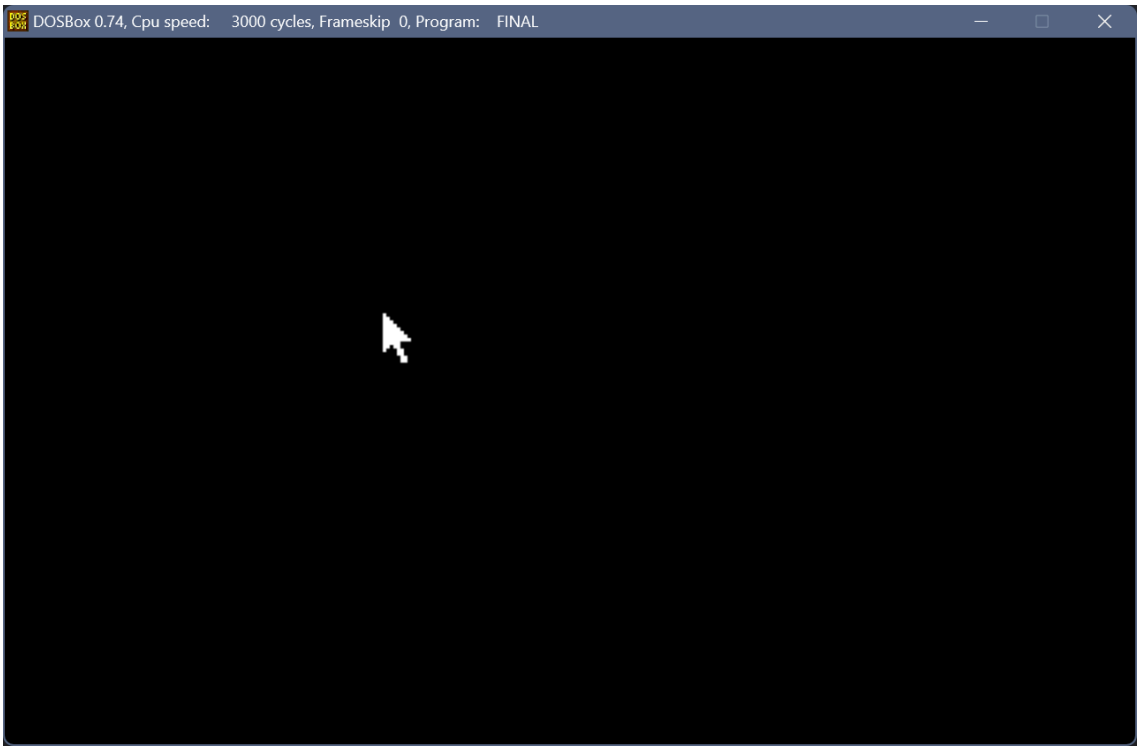
- **VGA模式设置**：使用`MOV AH, 0`和`MOV AL, 13h`设置视频模式为320x200的VGA模式，并通过`INT 10H`中断调用BIOS视频服务。

2. 鼠标初始化和显示

- **鼠标初始化**：通过`int 33h`中断调用初始化鼠标，检查鼠标是否可用。如果鼠标不可用（`or ax, ax`结果为0），则跳转到退出程序。
- **设置鼠标热点**：使用`mov ax, 4`和`int 33h`设置鼠标的热点位置为(0,0)。
- **显示鼠标光标**：通过`mov ax, 1`和`int 33h`显示鼠标光标。
- ```
  ; 初始化鼠标
  mov ax, 0
  int 33h
  or ax, ax
  jz near ptr exit

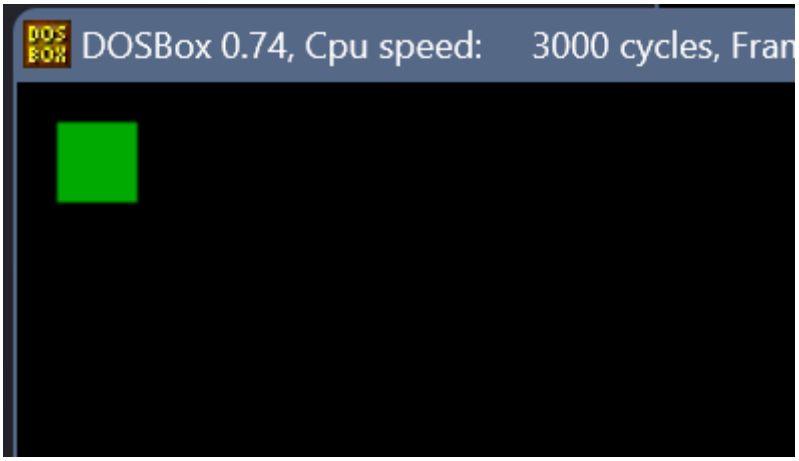
  ; 设置鼠标热点位置
  mov ax, 4
  mov cx, 0
  mov dx, 0
  int 33h

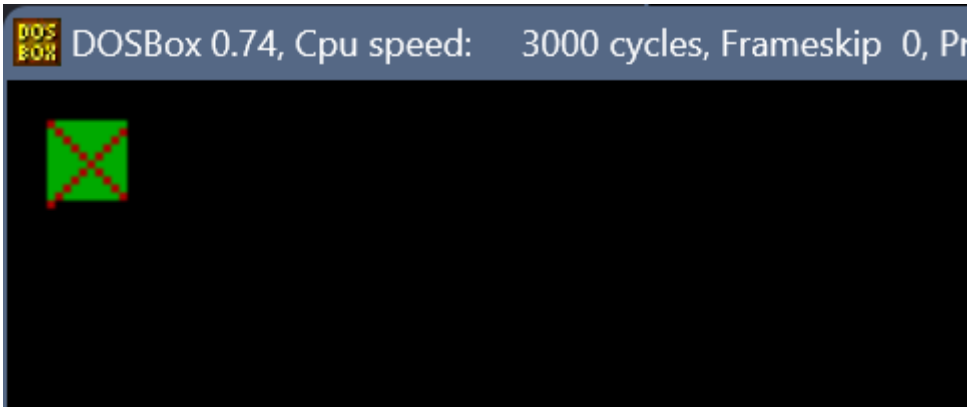
  ; 显示鼠标光标
  mov ax, 1
  int 33h
  ```
  ```



3. 颜色指示器绘制

- **绘制颜色指示器**：调用`draw_indicator`子程序，在屏幕上绘制一个颜色指示器，显示当前的绘图颜色。如果处于橡皮擦模式，则绘制一个红色的X形状。





4. 鼠标绘图操作

- **获取鼠标状态**：在`mouse_loop`循环中，通过`mov ax, 3`和`int 33h`获取鼠标状态。
- **检查左键按下**：通过`test bx, 1`检查鼠标左键是否被按下。如果没有按下，则跳转到`check_exit`。

- **坐标转换和绘制**：如果按下左键，程序会保存原始坐标，然后根据是否处于橡皮擦模式（检查`eraser_mode`），决定是绘制像素点还是擦除区域。

```
mouse_loop:
    ; 获取鼠标状态
    mov ax, 3
    int 33h

    ; 检查是否按下左键
    test bx, 1
    jz near ptr check_exit

    ; 保存原始坐标
    push cx
    push dx

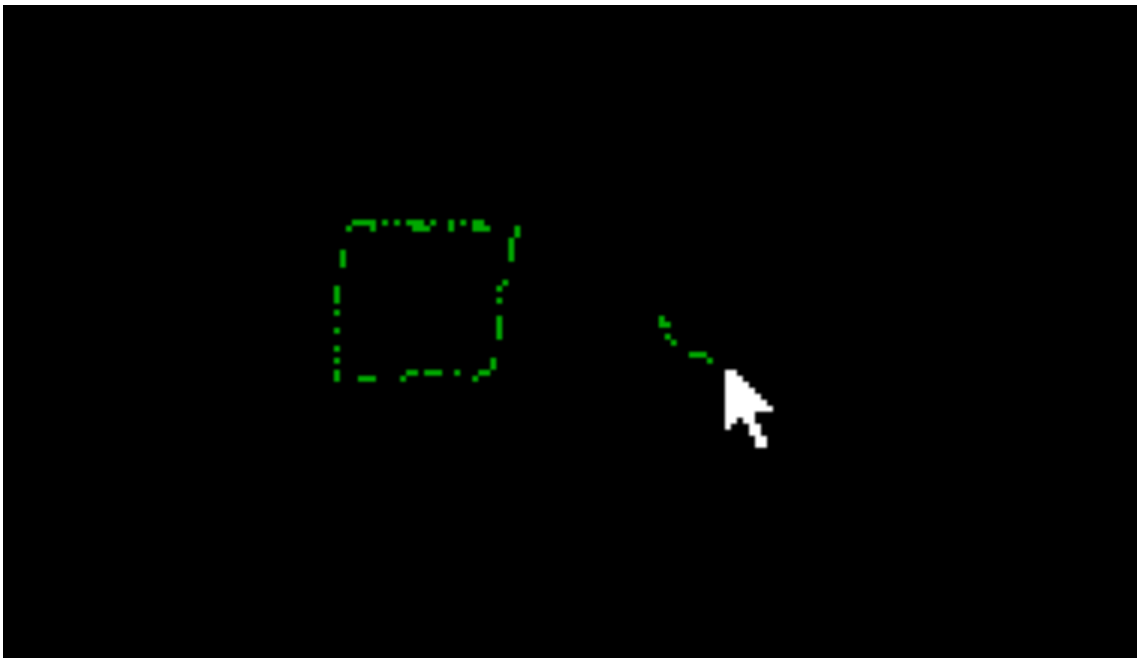
    ; 转换鼠标坐标，似乎DOSBOX中的鼠标点击位置和实际位置不一致
    shr cx, 1

    ; 检查是否在橡皮擦模式
    cmp [eraser_mode], 1
    je erase_area

    ; 正常绘制模式
    mov ah, 0Ch
    mov al, [pixel_color]
    mov bh, 0
    int 10h
    jmp drawing_done
```

5. 橡皮擦功能

- **擦除区域**：在`erase_area`标签下，程序绘制一个3x3的区域，使用黑色（`mov al, 0`）来擦除像素。



6. 键盘交互

- **退出程序**：通过检查键盘输入（`mov ah, 1`和`int 16h`），如果用户按下'q'键，则程序会退出。
- **清屏功能**：如果用户按下'c'键，程序会隐藏鼠标光标，清空屏幕，并重新显示鼠标光标和颜色指示器。
- **切换橡皮擦模式**：如果用户按下'm'键，程序会切换橡皮擦模式，并重新绘制颜色指示器。
- **颜色切换**：如果用户按下'n'键，程序会增加`pixel_color`的值，并更新颜色指示器。

三、项目开发过程及心得

在开发过程中，首先通过BIOS中断`INT 10H`设置视频模式，将显示模式更改为320x200的VGA图形模式。这一步骤是程序能够进行图形操作的基础。紧接着，通过`INT 33H`中断初始化鼠标，设置鼠标的热点位置，并显示鼠标光标。这些操作为后续的鼠标交互提供了必要的前提。

程序的主体是一个循环，不断地通过`INT 33H`获取鼠标状态，判断鼠标左键是否被按下，并据此执行绘图逻辑。在这一过程中，对鼠标坐标的处理尤为关键，需要将鼠标的相对坐标转换为屏幕坐标，并根据当前的颜色值绘制像素点。这一步骤涉及到对`CX`和`DX`寄存器值的操作，以及对颜色寄存器`AL`的设置。

对于橡皮擦模式的实现，程序检查一个特定的标志位，如果处于橡皮擦模式，则将绘制颜色设置为黑色，并对鼠标周围的3x3区域进行像素清除。这一部分的代码需要精确控制像素的绘制位置，以确保擦除效果的正确性。

颜色切换功能通过监听键盘输入实现，当用户按下特定键时，程序会更新颜色寄存器`AL`的值，并重新绘制颜色指示器。这一功能要求程序能够处理键盘中断，并根据用户输入更新程序状态。

对于本项目，目前遇到的难题在于如何优化程序性能，目前当用户使用鼠标在画板上绘制时，普遍非常卡顿，线条不流畅。未来的优化方向包括但不限于：优化绘图性能、增加更多绘图工具，添加图形保存功能，支持更多颜色。

通过这个项目，我对汇编语言的掌握程度得到了显著提升，尤其是在指令集的应用、寄存器管理以及硬件交互方面。在整个编码过程中，对寄存器的操作、内存访问和中断调用的精确控制是实现程序功能的关键。每一个汇编指令的选择都必须考虑到程序的整体架构和性能要求。对中断调用的处理让我深刻理解了系统级编程的复杂性。中断服务例程的编写和调试，尤其是在处理鼠标和键盘输入时，需要对程序流程有非常清晰的认识。这种对程序流程的控制和对硬件中断的响应，是对我编程能力的一种考验。