

北京理工大学

汇编原理与接口技术

组队上机作业报告

| | |
|---------|----------------|
| 学 院： | 计算机学院 |
| 专 业： | 计算机科学与技术 |
| 组 长： | 1120192092-曾群鸿 |
| 组 员： | 1120192080-胡书杰 |
| 组 员： | 1120180677-龚肖尹 |
| 组 员： | 1120192286-陈彦廷 |
| 指导教师： | 张全新 |

2022 年 6 月 12 日

组队上机作业报告

摘 要

本次作业我们基于 masm32 及 Windows API，实现了一个高性能的图形界面贪吃蛇游戏，并对游戏进行各种功能扩展，如实现了不同的食物（普通食物，爱心，炸弹，星星等），并将静态食物进一步扩展为动态的幽灵蛇，实现了不同档次的分数梯度排名给予玩家目标等。

在本报告中，我们按小组分工，首先介绍贪吃蛇游戏的基本框架，然后介绍基于框架实现的分数梯度系统，以及幽灵蛇系统，最后介绍。

关键词：贪吃蛇；图形界面；汇编语言

Abstract

In this work, based on masm32 and Windows API, we have implemented a high-performance graphic interface Snake game, and extended various functions of the game, such as implementing different foods (common food, love, bombs, stars, etc.), And further expand the static food into a dynamic ghost snake, and achieve different grades of score gradient ranking to give players goals and so on.

In this report, we work in groups, first introduce the basic framework of the Snake game, and then introduce the fractional gradient system implemented based on the framework, as well as the ghost snake system

Key Words:Grady Snake; Graphic Interface; Assembly language

目 录

| | |
|-----------------------|----|
| 摘 要 | I |
| Abstract | II |
| 第 1 章 基础架构及食物系统 | 1 |
| 1.1 设计目的 | 1 |
| 1.2 详细设计 | 1 |
| 1.2.1 游戏界面 | 1 |
| 1.2.2 贪吃蛇移动控制 | 1 |
| 1.2.3 食物生成 | 2 |
| 1.2.4 冲突检测 | 2 |
| 1.2.5 丰富的食物系统 | 2 |
| 第 2 章 分数排名功能 | 4 |
| 2.1 设计目的 | 4 |
| 2.2 详细设计 | 4 |
| 2.2.1 文件处理 | 4 |
| 2.2.2 读取一行 | 5 |
| 2.2.3 数字与字符串转化 | 5 |
| 2.2.4 比较函数 | 5 |
| 第 3 章 幽灵蛇功能 | 6 |
| 3.1 设计目的 | 6 |
| 3.2 详细设计 | 6 |
| 3.2.1 数据部分 | 6 |
| 3.2.2 函数实现 | 6 |
| 总 结 | 7 |
| 人员分工及贡献度 | 8 |

第 1 章 基础架构及食物系统

作者：1120192092-曾群鸿

1.1 设计目的

实现贪吃蛇游戏从 0 到 1 的过程，并力求设计一个模块性较好，可扩展性强的基本框架，方便小组成员进行各种功能扩展。

在完成基础架构的编写后，我对游戏的食物系统进行较为方便的扩展，设计不同的食物道具，会有不同的特效，如蛇身变长缩短。后续组员进行的各种功能扩展也证实了本游戏的基础架构较为可行。

1.2 详细设计

1.2.1 游戏界面

(1) 通过 Windows 3D 画图绘制游戏元素位图，如贪吃蛇的蛇身像素，食物，爱心等元素。

(2) 通过 CreateWindowEx 创建窗口，在 WM_CREATE 时加载各个位图，保存其 handle，在 WM_DESTROY 时关闭创建时的各个 handle。

(3) 在 WM_PAINT 时绘制游戏的一帧，按顺序检查当前的各种道具是否存在，如果存在则调用 BitBlt 对其在游戏界面上的位置进行更新。贪吃蛇的坐标用两个 XY 坐标数组保存，依次遍历数组中的每个元素，对蛇进行绘制。游戏的分数文本框也同样在此时进行绘制，通过 sprintf 函数将当前的分数按一定格式输出多缓冲区后，使用 DrawText API 绘制当前游戏分数。

1.2.2 贪吃蛇移动控制

(1) 设置全局变量 snakeDir 来表示当前贪吃蛇的方向，在获取键盘输入时根据键盘输入来改变蛇的方向。需要注意的是这里需要判断蛇的长度，当蛇只有一个长度时可以任意改变方向，而当蛇的长度大于一就不能朝当前方向的反方向移动，如当前朝上移动，不能立即向下移动。

(2) 通过 CreateEvent 创建一个进程用于定时对游戏元素数值进行更新，如蛇数组位置的变化，食物位置变化，当前分数等。

实验报告

(3) 当蛇身需要移动时, 首先将蛇数组的每个坐标向后移动一个位置, 如当前坐标为 $[(x_1, y_1), (x_2, y_2), (x_3, y_3)]$, 移动后变成 $[(x_1, y_1), (x_1, y_1), (x_2, y_2)]$, 最后根据当前蛇的全局方向更新蛇头坐标, 如当前为向上移动, 那么更新完后变成 $[(x_1, y_1 - 1), (x_2, y_2), (x_3, y_3)]$

(4) 在更新完蛇的数组后, 则会在 WM_PAINT 时对蛇的图形界面进行重新绘制, 实现蛇的移动控制和更新。

1.2.3 食物生成

(1) 食物生成通过 random 随机数来实现, 但需要保证生成出来的坐标 (x, y) 不能与蛇数组中的任意个坐标冲突。

(2) 因此我实现了一个函数来实现食物的生成, 生成方法如下:

(3) 确定当前游戏界面的块数, 如右下角坐标为 (n, m) , 则游戏有 $n \times m$ 块, 然后从 0 开始遍历到 $n \times m$, 对于当前遍历变量 i , 如何判断 i 是否可以用于食物坐标呢? 很简单, 再遍历一下蛇的坐标数组, 当二维坐标转成一维坐标, 判断是否与 i 相等, 如果每个坐标的一维形式都不与 i 相等, 那么说明 i 是一个合法的坐标。

(4) 将所有合法的坐标存储下来, 并记录其个数, 然后通过随机数随机选择一个坐标作为食物的坐标。这里需要将一维坐标转为二维坐标, 通过对 i 除 n 得到商和余数从而得到食物的二维坐标。

(5) 最后通过上述 WM_PAINT 时绘制食物。

1.2.4 冲突检测

(1) 上述所提到的进程用于定时更新游戏元素属性, 同时也进行冲突检测。冲突检测主要包含两种, 一是蛇与游戏边界的冲突, 二是蛇与自身的冲突。

(2) 与游戏边界的冲突检测比较简单, 判断蛇头的坐标是否合法即可。

(3) 与自身的冲突检测则需要从第 2 个位置开始遍历蛇数组, 检查是否有与蛇头坐标相同的坐标产生。

(4) 当发生冲突时, 游戏结束, 调用一定函数对游戏分数等进行组织然后输出。通过这种方法能实现比较好的扩展性, 如后续加入更多的游戏地形, 也是用同样的方法实现。

1.2.5 丰富的食物系统

(1) 在上述食物绘制的过程中, 又添加了各种不同功能的道具:

实验报告

(2) 爱心：蛇减少一个长度；炸弹：蛇增加 5 个长度并减少 50 得分；星星：增加 50 个得分，长度不变；普通食物：增加 10 得分并增加一个长度。

(3) 道具的实现与普通食物相同，通过遍历所有的位置检查是否与蛇以及其他道具发生冲突，保存所有可用位置，然后随机选择一个进行绘制。

第 2 章 分数排名功能

作者：1120192080-胡书杰

2.1 设计目的

对于等级排名存储在同级的 rank.txt 文件内，采用顺序的方式记录分数，分数由数字字符串组成，由高到低，会在分数栏一块同时显示 1 级需要的分数，分为 6 级 1000、600、400、200、100、50，得到对应级别会得到对应的级别提示。

2.2 详细设计

为实现分数排名功能，具体需要实现 Fileprocess 文本处理，其中需要使用 compare 函数，dtoc 函数以及 getline 函数。

2.2.1 文件处理

先对主体的 Fileprocess 进行讲述，其使用 compare、dtoc、GetLine 三个函数来帮助实现。首先是打开文件 invoke CreateFile, fpath1, GENERIC_READ, FILE_SHARE_READ, NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL 共有 L0: L1: L2: ENDFUNC: 四个地方进行跳转，L0 完成通过 GetLine 完成读取文件一行内容的功能，完成后到达 L1，L1 完成文件的判断，如果为空则结束，再到 L2，L2 完成三种功能，worktype==0 时就记录第一行把最高级别的分数传输给 maxone，如果 worktype==2，就进行等级判断，这里需要用到 compare、dtoc 两个函数，完成等级判断，具体逻辑就是先把 Score(得分) 转化为字符串，再使用 compare 进行比较，再使用 invoke sprintf, addr buffer_ranke, offset ranktell, index_line 来记录当前的等级。还有第三种工作模式，即记录各个等级需要的得分，通过 invoke sprintf, addr buffer_ranke, offset rankContent, index_line, offset buffer1 invoke strcat, offset rank, addr buffer_ranke 完成各个等级的记录。在窗口初始化前 invoke Fileprocess, offset file1_path, 0 获得 maxone 在窗口初始化时 invoke crt_sprintf, addr text, addr scoreFormat, score, offset maxone;hsj 在分数栏位置加入 maxone。在游戏判断这一轮结束时 invoke Fileprocess, offset file1_path, 1 invoke MessageBoxA, NULL, offset rank, offset MessengerankTitle, MB_OK 弹出窗口提醒各个等级得分 invoke Fileprocess, offset

实验报告

file1_path, 2 invoke MessageBoxA, NULL, offset tell, offset MessengerankTitle, MB_OK
弹出窗口提醒当前等级

2.2.2 读取一行

读取 rank 文件的每一个行, 文件内已排好序, 函数需要用到 HANDLE, buffer2:ptr
byte, worktype:DWORD 三个参数。有 L0、L1 跳转, 在 L0 通过 invoke ReadFile, fp,
addr char, 1, addr len, NULL 读取文件内容。L1 需要判断工作类型, 为 0 或者 2 时需
要把字符串的末尾即 0Ah 去除, 方便后续的处理和展示的美观。

2.2.3 数字与字符串转化

这个函数完成数字到字符串的实现, 也是借鉴了教程的内容, 因为得分是 dd 类
型所以使用的寄存器的类型也有所不同 mov eax,score

```
xor edx,edx
```

```
xor ecx,ecx
```

```
mov ebx, 10 ; 设置除数
```

rem: 求余 eax 存储商 edx 存储余数, 同 push 方式保存每一位余数, 直到余数为
0 copy: 把栈中的数据复制到 string 中通过 pop eax, add al,'0'; 把对应的数字转换成
ASCII 码, mov [edi],al 循环的方式得到对应的字符串。

2.2.4 比较函数

这个函数会通过 eax 返回三种 0, -1, 1, 因之前设置, 这里只比较同长度的字
符串, mov esi, str1, mov edi, str2 地位两字符串有 L0 一个跳转点, 通过 mov al,byte
ptr[esi], mov dl,byte ptr[edi] 保存两个字符串当前的值如果 str1 当前的值大于 str2 当
前值 mov eax,1 ret 如果小于则 mov eax,-1 ret 然后循环到 str1 当前的值为空, 如果为
空还未判断完成, 则 mov eax,0 ret

第3章 幽灵蛇功能

1120192286-陈彦廷

3.1 设计目的

游戏开始和每当蛇吃了一个苹果后会在地图中随机生成一条幽灵蛇，幽灵蛇长度为3，方向随机，只会走直线。如果玩家蛇成功吃到幽灵蛇（为提升难度，仅幽灵蛇头部有效）则游戏胜利。由于幽灵蛇与玩家蛇速度一致，想要吃到幽灵蛇较为困难，并且每次吃苹果才会刷新幽灵蛇，一直吃不到话蛇长度增加，吃到幽灵蛇变得更加困难。

3.2 详细设计

3.2.1 数据部分

- 1.GSnake equ 106 ; 代表幽灵蛇位图 ID
- 2.gsDir dd ? ; 幽灵蛇的方向
- 3.gsSize dd ? ; 幽灵蛇的长度
- 4.gsX dd BLOCK_NUM1 dup(?) ;
; 幽灵蛇在图上的 x 坐标, -1 表示不存在
- 5.gsY dd BLOCK_NUM1 dup(?) ;
; 幽灵蛇在图上的 y 坐标, -1 表示不存在
- 6.gsHandle dd ? ; 幽灵蛇位图 handle
- 7.MessageBoxFormat1 byte "Yon win!", 0

3.2.2 函数实现

在基础框架上进行修改，当创建窗口是创建幽灵蛇位图，当关闭窗口时关闭其 handle。在绘制游戏一帧时遍历幽灵蛇的数组，逐个元素进行绘制。

由于幽灵蛇的方向不会改变，与前文中所提到的蛇的移动方式相同，先想幽灵蛇向前移动一个位置，然后通过蛇的放心更新蛇头产生新的坐标。

在初始化时，将蛇头的坐标设置为-1，其长度固定为3；在生成一个食物的同时调用生成幽灵蛇函数对幽灵蛇的属性进行初始化。

总 结

本次作业我们实现了一个功能丰富，可玩性强的贪吃蛇游戏。除了实现传统贪吃蛇该有的功能，还添加了如排名系统，幽灵蛇得趣味性更强的游戏元素。所谓知行合一，通过本次作业，极大加深了我们对 masm32 汇编和 Windows API 的理解。

从一开始觉得使用汇编写一个 GUI 游戏是一件不可思议的事，都逐渐完成个人作业体验 Windows 的汇编编程，再到逐步实现一个可以运行的贪吃蛇框架，最后在这个框架上一个个添加各种有趣的功能。整个过程让我们对汇编更加熟悉，也不再觉得汇编是一件遥不可及的事，事实上哪怕不写汇编，了解汇编对我们日常使用高级语言也有极大的好处。很多高级语言可以转成汇编码，通过了解其汇编码让我们对高级语言有了更细致入微的理解。

人员分工及贡献度

曾群鸿：贡献度，实现本次作业的基础框架，包括游戏界面，游戏控制，碰撞监测，弹窗提示等游戏逻辑，并实现了不同的食物系统，当吃到不同食物会有不同的事件发生。

胡书杰：贡献度，基于初始框架实现等级排名系统，将游戏得分分为六个等级，在游戏结束后提示本次游戏等级。具体实现了文件处理，读取文件行等功能。

陈彦廷：贡献度，基于初始框架实现了幽灵蛇模式，当玩家吃到一个苹果后会出现一条幽灵蛇，当玩家吃到幽灵蛇的头部获得胜利，并随着游戏时间增加，难度上升。