**项目开发文档**

**项目名称：**简易贪吃蛇游戏

**项目功能介绍：**本项目是一个基于16位汇编语言，在DOSBOX实模式下编写的经典贪吃蛇游戏。该游戏在文本模式下运行，使用BIOS中断服务实现图形绘制、键盘输入、延时等功能。

游戏的主要功能包括：

地图绘制：绘制了贪吃蛇游戏的边框，定义游戏区域

蛇的初始化与移动：游戏开始时蛇会显示在屏幕中央，之后朝向默认方向，按照设定的速度自动向前移动；玩家可以通过键盘的上下左右箭头键改变蛇的方向。

食物生成：利用系统时间戳作为随机种子，在随机位置生成食物块，当蛇吃到食物后长度增加，并且得分加一。

碰撞检测：实现了对墙壁和自身身体的碰撞检测，一旦发生碰撞则游戏结束，在屏幕上显示“Game Over”和最终得分信息。

分数显示：实时更新并显示当前得分。

**项目开发流程：**

数据段：

定义了所有需要的数据结构，包括蛇的位置、方向键检测、颜色、提示信息、食物的位置、得分信息等。在数据段中，特别处理了蛇身的数据结构，每个节点记录前一个节点在内存中的相对偏移量，该节点在屏幕上的位置和下一个节点在内存中的相对偏移量，通过这种方式，可以高效地管理蛇的移动和增长。主要的数据量有：

UP、DOWN、LEFT、RIGHT：存放键盘方向键的 ASCII 值，用于控制蛇的运动。

DIRECTION：存放当前蛇头的运动方向（上、下、左、右）。

SNAKE：存储蛇各个节点，包括上一个结点位置，下一个结点位置以及本节点在屏幕上的显示位置。

SNAKE\_HEAD ：存放蛇头内存偏移量

SNAKE\_TAIL：存放蛇尾内存偏移量

ALLOC\_BLOCK：存放新分配结点内存偏移量

FOOD：存储食物的坐标。

SCORE：存储玩家的当前得分。

IS\_DEAD：存储游戏的状态，以此判断游戏是否继续进行

代码段：

初始化：设置了视频模式为80x25字符模式，清屏并初始化地图，在屏幕上的初始位置绘制蛇

主循环：整合了所有子程序，形成了完整的逻辑框架，包括读取用户输入、更新游戏状态、渲染画面等步骤。

主要子程序：

CLEAR\_SCREEN：调用了BIOS中断10H的功能06H来清除屏幕，并设置背景色和前景色。

DRAW\_MAP：通过直接操作显存地址，绘制游戏边界。

DRAW\_SNAKE：根据预先设定的坐标，在屏幕上绘制初始状态下的蛇。

KEYBOARD\_INPUT：获取键盘输入，判断是否为方向键，并根据方向键的类型改变蛇的行进方向。

DELAY：实现了一个简单的延时函数。

MOVE\_SNAKE：负责将蛇根据移动方向移动到新的位置。

DRAW\_NEW\_SNAKE：负责蛇的重新绘制，具体为擦除掉蛇尾，并且绘制新的蛇头。

GENERATE\_FOOD：利用系统时间戳作为随机数生成器，随机生成食物的位置。

JUDGE\_GAME：判断游戏状态的函数，检查蛇是否撞到了边界或自己的身体，以及有没有碰到食物。如果碰到了边界或自己的身体，则判定游戏结束；如果碰到食物，则“吃掉”食物增加自己身体的长度，并且增加得分。吃掉食物后，重新生成食物。

DEAD：处理游戏结束后的逻辑，输出游戏结束提示消息以及对应的得分。

EAT\_FOOD：当蛇吃掉食物时，增加蛇的长度，同时更新得分。

PRINT\_SCORE：将当前得分转换为字符串格式，并在屏幕上显示出来。

**开发心得：**

通过本次项目，我对汇编语言有了更加深入的理解，也对汇编语言如何对计算机系统进行更改有了更加清晰的认识。

在实现游戏逻辑时，特别是在处理键盘输入、屏幕绘制和延迟控制等方面，均使用了BIOS中断机制。中断不仅是操作系统与硬件之间的桥梁，也是汇编程序中高效实现输入输出操作的核心。通过调用 INT 10H 实现屏幕绘制和 INT 16H 读取键盘输入，我更直观地了解了计算机是如何响应外部事件的。

在实现蛇的移动和增长的过程中，我体会到了良好的算法设计和合适的数据结构对提高程序效率的重要性。为了管理蛇身的各个节点，我设计了一个双向链表结构，通过指针连接每个节点。在蛇的状态变化时，需要手动管理每个节点的位置，并更新指针。对于移动蛇，程序的逻辑是将蛇尾结点擦除，并在新结点位置画上新的蛇头结点；这个操作需要很小心地调整内存中的数值，以达到正确的结果。

在本程序中，也有一些需要改进的地方。本程序中每次循环都依赖键盘输入；通过定时器中断，可以控制输入处理的频率，避免每次循环都处理键盘输入，减轻 CPU 的负担。可以专门在中断服务程序中处理键盘输入，避免每次主循环都扫描键盘；需要在系统初始化时，配置定时器来产生中断；定时器中断触发时，读取键盘输入，更新游戏状态

总的来说，这次实验让我更全面地理解了汇编语言的应用与优势，尤其是在需要精确控制计算机硬件和优化性能的场合。通过编写贪吃蛇游戏，不仅加深了我对低级语言的理解，还在实践中锻炼了我的问题解决能力。