**完成过程与设计思路**

**完成过程：**

1. **选题**

选择贪吃蛇作为项目的最主要原因其实是因为在十个题目中我对贪吃蛇最熟悉，毕竟谁小时候没在诺基亚手机上玩过贪吃蛇呢？

1. **查找资料，学习相关知识**

在知乎，B站，CSDN等网站上查询了大量有关贪吃蛇游戏的教程，了解了一些以前从来没用过的头文件的作用，学习了很多课外知识，比如利用Windows API中的函数控制光标位置，设置文字颜色；通过虚拟键码识别按键，等等。

1. **开始编写**

题目要求很多，所以采用模块化编程是最好的选择。我把题目中的各个要求写成一个个函数，比方说食物吃完要刷新这个要求，我就编写了两个函数来满足，一个用来随机生成食物，另一个用来确保食物没有刷新在蛇身里；再比如蛇要动起来，这个该怎么用函数实现呢？我在蛇头插入一个新节点，再把尾部节点删除，这样就能实现蛇移动的效果了，诸如此类。通过不断地编写函数满足题目要求，我一点点地写完了这个程序（当然，在编写过程中有些部分借鉴了一下AI）。

1. **Debug**

罗马不是一天建成的，程序也不是一次就能写好的。简单列举一下我印象深刻的一些bug:蛇身加速移动时直接断裂，原因在于Sleep参数计算错误导致刷新过快，调整公式为Sleep(max(50,350-speed))后解决；整个界面闪烁严重，原因在于直接刷新了整个界面，修改为局部刷新，即只刷新蛇身后解决。

**设计思路：**

**1.核心设计目标**

**实时交互：**通过键盘控制蛇的移动方向，实现流畅的响应。

**动态更新：**蛇身随移动自动增长，界面实时刷新。

**规则约束：**碰撞检测（墙、自身）触发游戏结束。

**数据持久化：**保存历史分数并支持排名展示。

**2. 模块化设计框架**

| **模块** | **功能** |
| --- | --- |
| **图形界面** | 使用控制台字符（■ ★ ●）绘制地图、蛇身和食物，通过gotoxy精准定位坐标 |
| **输入处理** | 监听方向键（↑↓←→）和功能键（F1/F2/空格），实现非阻塞响应 |
| **蛇身管理** | 动态链表存储蛇节点，头部插入新坐标实现移动，尾部删除维持长度 |
| **碰撞系统** | 坐标比对检测蛇头与墙、自身或食物的碰撞 |
| **分数系统** | 文件读写（scores.txt）保存得分，冒泡排序实现排行榜 |