

2025 年春程序设计思想与方法 (C++) 大作业

一、作业目标

本项目要求同学们以 3-4 人小组合作的形式，综合运用 C++ 语言的面向对象编程、图形编程，完成一个具有良好用户体验的**图形化贪吃蛇游戏**。本作业目标如下：

- 熟悉 C++ 的基本语法、类与对象、STL 容器等基础；
- 掌握游戏循环、键盘事件、碰撞检测等基础游戏机制；
- 初步掌握图形库使用（如 Qt）；
- 训练模块化设计、程序调试与文档撰写能力。

二、核心功能要求（必做）

1. 图形用户界面 (GUI)

- 实现图形化界面：
 - 建议使用 Qt。（参考 https://blog.csdn.net/qq_41622214/article/details/124776481）
- 游戏窗口显示贪吃蛇、食物、得分信息；
- 实现基本的图形绘制（蛇身、食物、背景）；
- 实现键盘控制（↑↓←→）来操控蛇的移动；
- 蛇头朝向应随方向改变而变化（如可用颜色/形状区分）；

2. 基本游戏机制

- 贪吃蛇自动匀速移动，方向由玩家控制；
- 蛇吃到食物后，蛇身变长，得分增加；
- 食物出现在地图内的随机位置，不与蛇身重叠；
- 撞墙或咬到自己时游戏结束，并显示最终得分；

3. 计分与重开

- 显示当前得分；
- 游戏结束后提供“重新开始”按钮或快捷键（如 R）；
- 可选实现简单菜单（开始游戏、退出游戏）；

三、附加功能要求（加分项）

以下功能为加分项，供有余力的同学挑战：

功能	加分说明
多种难度选择（速度快慢）	+5 分
计时器（游戏用时显示）	+3 分
音效/背景音乐	+5 分
自定义皮肤（颜色/形状）	+5 分
增加障碍物地图	+5 分
存档与读取历史最高分（本地）	+5 分
多人对战（双蛇对战）	+10 分（挑战）

四、技术要求与规范

1. 推荐类设计结构

- class SnakeGame; // 游戏主控类，管理整体状态、循环

- class Snake; // 贪吃蛇逻辑，包含坐标队列、移动逻辑
- class Food; // 食物类，生成与重置逻辑
- class GameRenderer; // 图形绘制模块（封装 SFML/SDL/Qt）

2. 构建与依赖管理

- 建议使用 **CMake** 管理项目；
- 保持项目结构清晰、模块独立、注释规范。

五、提交要求

- 源代码及完整工程文件
- 说明文档
 - 项目概述、运行说明
 - 构建环境与依赖说明
 - 实现功能概述与演示截图
- 可执行程序（建议 Windows）

六、评分标准（总分 100）

项目	分值	说明
基础游戏功能完整性	30 分	蛇移动、吃食物、变长、撞墙判断等
图形界面与交互体验	25 分	图形绘制清晰、键盘响应顺畅、信息展示直观
代码结构与规范	15 分	类设计合理、注释清晰、模块分离良好
计分系统与游戏循环	10 分	实现得分统计、重新开始逻辑
提交文档与材料完整性	20 分	提交说明清晰、可正常运行
附加功能实现（如音效/难度）	最高 15 分	每项附加功能相应加分(加分后总分不超过 100)

七、注意事项

1. 请完全以面向对象的思想实现，即通过封装类、创建类的对象并调用对象的功能的形式完成本项目，尽量不要使用结构体。
2. 请充分利用大模型协助完成大作业。包括使用大模型学习图形化界面开发，帮助生成项目框架和简单的代码、协助 Debug、甚至可以通过不断的提示词交互，生成一个完整的项目。我们不会因为代码是由模型生成的而扣分，反而鼓励积极使用大模型协助开发和撰写项目报告。我们不限使用哪个大模型，除了上课介绍的通义灵码外，也可以使用 DeepSeek、通义千问、文心一言、豆包等其他大语言模型工具。
3. 提交的可执行文件请务必在小组多个成员的电脑上测试并确保可运行，尤其是 Mac 系统的同学，请在 Windows 系统上生成 exe 文件，而后再提交。同时，我们还会亲自检查并运行你们的代码，如果 exe 文件不可运行同时源码也无法编译，那么我们将不能测试程序的其他功能，大作业分数将不会超过 70。
4. 请将图形化界面设计用到的库所对应的 dll 文件一并复制到 exe 所在的文件夹内，以确保在没有图形化编程功能的电脑上仍然能运行你的程序（可以在小组内部成员间的电脑上互相测试）。
5. 请确保程序对异常输入具有一定的鲁棒性。这并不是要求你们的程序对所有种类的异常输入均要有响应，但请尽量不要出现无法输入、闪退、死循环等情况。根据以往经验，请额外注意指针对内存的访问是否规范、数组是否越界以及异常类型的输入的问题。
6. 报告建议采用 Word 或 LaTeX 撰写，最后导出 pdf 提交。不建议使用 Markdown。

请注意 字号、字体、行间距等格式前后统一，标题请分级。我们对格式没有具体的要求，但务必简洁美观。同时，请不要只重视代码实现而忽视报告撰写。我们通常会依据报告来对照评估代码功能，报告中缺少必要的说明可能会使我们忽视程序中的某些潜在功能进而影响大作业的最终成绩。

7. 请注意代码撰写规范，包括但不限于：
 - a) 变量类型正确、命名规范且易于理解；
 - b) 缩进、空格合理且美观；
 - c) 适当的注释；
 - d) 尽量少的全局变量；
 - e) new 与 delete 匹配；
 - f) const 和 static 的正确使用；
 - g) 组合、继承和运算符重载的灵活设计；
 - h) private、protected 和 public 权限的合理设置。

建议好好利用 CLion 的代码规范检查功能。例如，当 CLion 的代码界面右上角有黄色的感叹号时，请仔细分析是否存在代码不规范的情况。

8. 对大作业有任何问题，可以随时在群里或跟私下和助教联系。
-