

2017 软件学院 C++ 期末项目说明

2017 软件学院 C++ 期末项目说明

基本要求

评分标准

选择参考选题进行开发的项目

自拟内容进行开发的项目

参考选题

Gameboy 模拟器

泡泡堂

Agar.io

3D 魔方

俄罗斯方块

2048

加分项

版本控制与团队协作

代码质量与安全

功能和架构

其他

助教和联系方式

附录

钦定的 git 托管网站

必须使用的 C++ 特性

必须达成的代码规范

参考资料

游戏引擎

网络库

Git 教程

Git 图形化客户端

基本要求

1. 在给定的几个参考选题中选择一个完成，或自行拟定内容，但必须是游戏类型。对于自行拟定的题目，需在 4 月 16 日前联系助教以评估难度、工作量和可行性（联系方式见"助教和联系方式"）；
2. 以个人或团队身份完成，可以跨班组队。若从参考选题里选题，则团队人数限制见选题的说明；若自选题目，则团队人数不超过 4 人；
3. 必须使用 git 进行版本控制，且公开发布指定的 git 托管网站之一（见"附录"）；
4. 实现了基本功能、可编译为可执行文件、可正常运行不崩溃；
5. 不得抄袭任何代码，包括开源项目中的代码，但允许使用开源库；若以源码方式引用开源库，则开源项目的源码或头文件需放入和你编写的源代码不一样的目录中；
6. 有基本的图形化界面（如使用 Qt、Cocos 等框架，或 OpenGL、DirectX 等原生 API），不得只包含控制台界面；本项目偏向于考察和锻炼代码和架构设计，请勿将重点错误地放在了精美的界面上；
7. 必须贯彻且合理地使用了指定 3 条以上的 C++ 功能（见"附录"）；
8. 必须达成所有指定的代码规范要求（见"附录"）；
9. 需在 4 月 23 日前将分组信息（包含学号和姓名）、组长（仅限团队项目）、选题（或自拟课题的详细描述）、项目 git 地址发送给助教（两位助教都需要发送，联系方式见"助教和联系方式"）。
10. 期末需进行答辩（答辩时间另行通知）；
11. 期末需递交项目源码、可执行程序 and 文档（文档需涵盖的内容另行公布）。

评分标准

以团队为单位进行评分，同一个团队下的成员一般会获得比较相近的期末项目分数，但贡献非常少的团队成员可能被扣除不超过 40% 的期末项目分数。贡献程度由 git 递交历史评定。

选择参考选题进行开发的项目

评分项	权重
基础功能的实现程度 [*]	40% × 得分权值
基础功能以外新开发的功能的创新性	5%
基础功能以外新开发的功能的技术难度	15%
基础功能以外新开发的功能的工作量	15%
C++ 功能和特性的使用程度	10%
答辩	10%
文档	5%
以个人身份完成项目	额外 5%
加分项	不超过额外 30%

[*] 选题的得分权值（本文档给出，见"参考选题"）将影响你在该项目的最终得分。例如在得分权值为 0.6 的情况下，你在该项目中的最终得分不会超过 $40\% \times 0.6 = 24\%$ ，因此你的期末项目总分至多为 84 分（100 分计）。得分权值大于 1 时，多出来的分数作为额外加分项（不与评分项中的加分项冲突）。

自拟内容进行开发的项目

评分项	权重
所完成内容的创新性	10%
所完成内容的技术难度	30%
所完成内容的工作量	30%
C++ 功能和特性的使用程度	10%

评分项	权重
答辩	15%
文档	5%
以个人身份完成项目	额外 5%
加分项	不超过额外 30%

参考选题

每个选题都给定了一个得分权值，代表项目的难度和工作量，也决定了你期末项目分数的上限。得分权值越高，完成该项目的难度和所需的预计工作量越大、完成相同数量的基础功能情况下所得的分数也会越多。请根据自己的水平选择合适的项目，避免由于选择了高难度项目毫无进展而得零分，或由于选择了太低难度项目无法得到全部分数。

Gameboy 模拟器

得分权值：1.5

团队人数限制：≤ 4 人

描述：实现一个 Gameboy Classic 模拟器。

需要达成的基础功能：

- 实现 Gameboy Z80 CPU 模拟（支持所有或大部分指令集）
- 实现时钟模拟
- 实现内存模拟
- 支持基本图形操作
- 支持对游戏进行交互操作（即输入）
- 支持载入 ROM
- 可以基本玩一款 GB 游戏

基础功能以外的可选功能（欢迎自创）：

- Background Window Graphic
- Sprite
- 音频输出
- 金手指

注：该选题不是要求实现 GBA。

可供参考的资料：

- [Gameboy CPU Manual](#)（非常重要）
- [Gameboy Emulation Tutorial \(JavaScript\)](#)
- [NO\\$GMB \(Gameboy Emulator / Debugger\)](#)
- [Start Gameboy Programming](#)

泡泡堂

得分权值：1.2

团队人数限制：<= 4 人

描述：实现一个类似于[泡泡堂](#)的游戏。

需要达成的基础功能：

- 支持地图绘制、人物绘制、水泡效果绘制等
- 支持鼠标和键盘操作交互
- 支持障碍物
- 支持泡泡的放置与爆炸
- 支持三种基本增强型道具（鞋子，泡泡，药水）
- 实现服务端：支持局域网联机对战（自由对抗模式），且支持多人在同一个地图游戏
- 支持动画效果

基础功能以外的可选功能（欢迎自创）：

- 支持多个角色（角色造型和属性不同）
- 支持聊天（组队聊天和所有人聊天）
- 支持房间列表
- 支持 >= 2 张地图
- 支持 >= 2 种游戏模式（如团队模式、大乱斗、抢包山、刺猬模式等）
- 支持使用道具（如香蕉皮，飞镖）
- 支持骑宠（如乌龟，猫头鹰，飞碟，恶魔等）

Agar.io

得分权值：1.0

团队人数限制：<= 4 人

描述：实现一个类似于[agar.io](#) 或球球大作战的游戏。

需要达成的基础功能：

- 支持图形显示
- 支持鼠标和键盘操作
- 支持 agar.io 中的吞噬操作
- 支持 agar.io 中的分裂操作
- 支持动画
- 实现服务端：支持局域网联机对战（自由模式），支持多人在同一个地图上游戏

基础功能以外的可选功能（欢迎自创）：

- 支持病毒

- 支持聊天
- 支持团队协作模式
- 服务端能抵御作弊，即客户端不能任意传送自己的位置或宣称吞噬了某个细胞等，必须有真实操作

3D 魔方

得分权值：0.9

团队人数限制：≤ 2 人

描述：实现一个 3D 魔方游戏。

需要达成的基础功能：

- 3D 展示魔方
- 视角可 360 度旋转
- 支持鼠标和键盘方式操作
- 支持 3 ~ 20 阶
- 旋转时有动画
- 排行榜

俄罗斯方块

得分权值：单机版 ≤ 0.7，多人版 ≤ 0.9

团队人数限制：单机版 ≤ 2 人，多人版 ≤ 3 人

描述：实现一个俄罗斯方块游戏。

需要达成的基础功能：

- 随机方块、方块下落、碰撞检测、鼠标或键盘控制
- 排行榜
- 可以修改难度（难度对应方块下落速度）
- 支持动画

多人版需额外达成的基础功能：

- 实现双人对抗，对抗时同步地实时显示其他人的屏幕

基础功能以外的可选功能（欢迎自创）：

- 自定义开局
- 多人对抗时己方的快速操作能给对方施加难度

2048

得分权值： ≤ 0.5

团队人数限制： ≤ 1 人

描述：实现一个 2048 游戏

需要达成的基础功能：

- 本地排行榜
- 支持检测游戏终止
- 开始游戏前可修改阶数
- 支持移动动画和融合动画

基础功能以外的可选功能（欢迎自创）：

- 下落模式

加分项

加分项可叠加；请在答辩前告知助教核查项目的加分项。

版本控制与团队协作

- 符合一些 git 最佳实践（如 commit 历史干净规范、commit 描述规范、commit 包含且仅包含一项功能、使用 Pull Request 功能或分支进行团队协作等）；
- 团队成员分工平等、合理；

代码质量与安全

- 对于复杂逻辑使用单元测试验证正确性；
- 在单元测试基础上使用持续集成平台（如 Travis CI、AppVeyor）；
- 使用断言验证程序性质；

功能和架构

- 界面精致；
- 跨桌面平台（Windows、Linux、MacOS 中至少两个）；
- **Super Bonus: 能移植或直接运行在 Android 或 iOS 平台并参加竞赛；**
- 使用现代化构建工具（如 cmake、ninja）；
- 项目目录结构良好、清晰；
- 恰当合理地使用了 boost 库；

其他

- 没有内存泄漏；
- （助教自行通过界面玩耍程序时）程序很少崩溃；
- 贯彻且合理地使用了下列 3 条以上的 C++11 或更高的 C++ 新特性：
 - 初始化列表
 - 类型推断 (`auto` / `decltype`)
 - 基于范围的 `for` 循环
 - 智能指针
 - 常量表达式 (`constexpr`)
 - Lambda 表达式

- 右值引用
- 字符串字面量
- 其他 C++11 或更高的新特性
- 其他你认为可以作为加分项的亮点。

助教和联系方式

助教1:

负责加分项认定、选题难度评估

QQ: 490241327 (wish)

Email: QQ 对应邮箱

助教2:

负责其他所有事宜

QQ: 505968815 (Sweet)

Email: QQ 对应邮箱

附录

钦定的 git 托管网站

你的项目必须使用 git 进行版本控制（git 图形化客户端见"参考资料"），且公开发布以下 git 托管网站之一，否则扣除不超过总分 100% 的分数：

- GitHub
- GitLab
- Bitbucket
- Coding.net（若上述三个网站访问困难可考虑使用这个）

必须使用的 C++ 特性

你的项目必须贯彻且合理地使用了下列 3 条以上的 C++ 功能，否则扣除不超过总分 100% 的分数：

- STL 容器，如 `std::vector`，`std::map` 等；
- 迭代器；
- 类和多态；
- 模板；
- 异常；
- 函数重载；
- 操作符重载；
- C++11 或以上功能。

必须达成的代码规范

你的项目必须达成以下所有代码规范要求，否则扣除不超过总分 50% 的分数：

- 代码在缩进、命名等方面基本遵循了统一和一致的风格（如 Google C++ Style）；
- 正确、合理地使用了 C++ 风格类型转换（如 `static_cast`、`dynamic_cast`）且没有使用 C 风格的强制转换；
- 尽可能地使用了 `const` 和引用；
- 类的设计合理规范。

参考资料

游戏引擎

使用游戏引擎可加速游戏开发并能比较方便的实现跨平台。完整的游戏引擎列表：

https://en.wikipedia.org/wiki/List_of_game_engines

请自行搜寻资料并选择合适的游戏引擎。一般需要考虑的因素有资料丰富程度、功能丰富度、开发复杂度等。

网络库

一些游戏引擎自带网络库（但一般不支持服务端），你也可以使用其他专业的网络库，例如：

- [Boost asio](#)
- [Libuv](#)
- [Libevent](#)

Git 教程

- [官方文档](#)
- [GitHub 和 SourceTree 入门教程](#)

Git 图形化客户端

对于新手，推荐使用图形化界面进行 git 版本控制，例如：

- [SourceTree](#)（跨平台）
- [SmartGit](#)（跨平台）
- [GitHub Desktop](#)（跨平台）