

# 2019 软件学院 C++ 期末项目说明

## 基本要求

---

1. 在给定的几个参考选题中选择一个完成；
2. 以个人或团队身份完成，允许跨班组队。团队人数限制见选题的说明。
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21日 前将分组信息（包含学号和姓名）、组长（仅限团队项目）、选题、项目 git 地址发送给助教（三位助教都需要发送，联系方式见"助教和联系方式"）。
10. 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），提交截止时间为 16周 周日24点 之前。
11. 期末需进行答辩，答辩时间初步安排在 17周 课上答辩；

## 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"）将影响你在该项的最终得分。例如在得分权值为 0.6 的情况下，你在该项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）。

## 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](#)

## 王者荣耀（2D）

得分权值：1.3

团队人数限制：<= 4

实现一个类似于王者荣耀的在线MOBA游戏。以下功能为王者荣耀为例，你可以使用其他游戏题材，但必须是在线MOBA类型的，并且工作量与下列要求相符。需要达成的基础功能：

- 实现服务端：支持联机对战、电脑AI加入
- 1v1地图
- 支持至少三种英雄类型（亚瑟、后羿、妲己）
- 支持三种基础小兵（近战兵、远程兵、炮车）
- 支持金钱的各种获取方式（补兵、防御塔、击败敌方英雄、时间）
- 支持经验、等级系统
- 支持装备系统
- 支持战绩、装备页面查看
- 支持小兵、防御塔AI
- 英雄操作方式可以参考英雄联盟orDota

如果有余力，可以考虑的进阶功能（欢迎自创）：

- 5v5地图
- 支持更多的英雄类型
- 支持三种野怪（红buff、蓝buff、暴君）
- 支持聊天

## 弓箭手大作战

得分权值：1.1

团队人数限制：<= 4

实现一个类似于 [arrow.io](#) 或弓箭手大作战的游戏。需要达成的基础功能：

- 支持鼠标和键盘操作
- 支持道具拾取功能
- 支持动画

- 支持多种不同类型的武器
- 实现服务端：支持联机对战、电脑AI加入，多人在同一个地图上游戏

如果有余力，可以考虑的进阶功能（欢迎自创）：

- 支持聊天
- 支持团队协作模式
- 更多的武器类型
- 更多的动画

## 象棋

得分权值：0.9

团队人数限制：<= 2人

描述：实现一个在线象棋对战的游戏。

需要完成的基本功能：

- 移动棋子、吃棋子动画
- “将军”提示
- 判断下子方是否已经胜利（对方已经死棋）

基础功能以外的可选功能（欢迎自创）：

- 支持多个房间
- 支持聊天
- 支持残局对战

## 打砖块

得分权值：单机版<=0.7，多人版<=1.0

团队人数限制：单机版<=2人，多人版<=3人

描述：实现一个打砖块游戏。

需要达成的基础功能：

- 碰撞检测、鼠标或键盘控制
- 打碎砖块掉落多种道具
- 支持动画
- 排行榜
- 随着分数增高，小球移动速度变快
- 设置不同关卡（砖块位置、数量，初始速度，初始板的长度）

多人版需要额外完成的基本功能：

- 实现双人对抗，对抗时同步地实时显示其他人的屏幕

其他基础功能以外的可选功能（欢迎自创）：

- 多人对抗时，己方短时间消除多个方块，给对方增加难度
- 支持聊天

- 多种材质的砖块
- 可以从配置文件读取游戏配置，加载不同关卡（砖块位置、数量，初始速度，初始板的长度）

## 贪吃蛇

得分权值： $\leq 0.6$

团队人数限制： $\leq 1$  人

描述：实现一个贪吃蛇游戏

需要达成的基础功能：

- 本地排行榜
- 碰撞检测、鼠标或键盘控制
- 支持吃食物的动画
- 可以穿越地图边界（例如，穿过右边边界，会到达左边边界）
- 随着分数增高，蛇的移动速度变快
- 设置不同关卡（障碍物位置，初始速度等）

其他基础功能以外的可选功能（欢迎自创）：

- 可以吐出舌头吃食物
- 可以从配置文件读取游戏配置，加载不同关卡（障碍物位置，初始速度等）

## 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；

- 恰当合理的使用了 **设计模式**

## 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

---

- 钱煜 QQ: 348368689、Email: QQ 对应邮箱
- 张尹嘉 QQ: 1051867531、Email: QQ 对应邮箱
- 刘子赫 QQ: 272910663、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](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][<http://think-async.com/>]
- [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）

### 历年优秀项目

- [GameBoy 模拟器](#)

- 泡泡堂
- 泡泡堂
- 泡泡堂