

评分

本题的评分采用线下 check 的方式。本次 check 的评分整体分为三个部分组成，其中基础功能和扩展功能为加分项，而代码规范部分为扣分项。在实现代码之前，可以仔细思考代码规范中提到的设计思路，来预先对整个游戏工程的代码做好设计。

基础功能和扩展功能中的各项不再设部分分，**如果有 bug 则视为未实现**，所以请务必测试你实现的功能。

功能实现 (100%)

以下的功能为加分项，功能应该正确实现没有 bug，并且拥有合适的贴图与动画显示。

1. (10%) 背景能动起来，而且看起来和样例大体一致
2. (10%) 玩家能原地跑起来
3. (10%) 玩家能正确的跳起来，落地位置和原位置一致
4. (10%) 哥布林能正确产生，并血量归零后消失
5. (10%) 哥布林会对玩家近身造成伤害，玩家血量与伤害数值设置合理，玩家血量归零后结束游戏
6. (10%) 玩家可以开枪，子弹能正确射出
7. (10%) 射出的子弹将会对第一个碰到的哥布林造成伤害，碰撞后子弹将消失。
8. (10%) 哥布林能扔出斧子，哥布林扔斧子动画正确
9. (10%) 哥布林的斧子能正确造成伤害，碰撞后斧子会消失
10. (10%) 玩家起跳的数值足够规避斧子的攻击

代码规范 (20%)

此部分为扣分项，扣满 20% 为止。

1. **(20%)** 编译没有来自于你自己的代码的 warning。
2. (5%) 代码中不应手动管理内存，而是应该正确使用智能指针和标准库容器。
 - “正确使用智能指针”还包括**正确使用** `std::enable_shared_from_this` **相关设施**，例如，应该只在必要时调用 `shared_from_this()`，而不是粗暴地将所有 `this` 都替换为 `shared_from_this()`。
3. (5%) 将代码按类别分在不同的文件里。按类别把文件分在不同的子目录里，并为每个子目录都创建 `CMakeLists.txt`，以 `add_subdirectory` 的方式添加进项目。
4. (5%) 对于相似类的类似功能，不应复制代码，而是应该在基类里统一实现。
5. (5%) 正确使用枚举 (enum) 类型，并且应该使用限定作用域的枚举类型 (即 `enum class`)，除非有特殊情况 (请解释)。不应使用神秘数字 (magic numbers) 或字符串来完成本该由枚举类型完成的功能。
6. (5%) 正确使用 `utils.hpp` 中定义的 `const` 变量，而一些类内常量比如重力加速度，则可以在类内用 `static const int GRAVITY = 2;` 的 static 成员变量，而不是使用神秘数字。
7. (5%) 通常情况下，所有数据成员几乎都属于“实现细节”，应当设置为 `private`。注意：
 - 一些需要被子类访问的数据成员应该设置为 `protected`。但是你不能滥用 `protected`，因为 `protected` 成员的封装性实际上和 `public` 是一样弱的，详见《Effective C++》条款 22。

- 的确存在一些足够简单的类，比如 `std::pair`，它的数据成员（`first` 和 `second`）就是它的接口，此时直接将它们设为 `public` 即可，[无需编写那些 trivial 的 getters/setters](#)。如果你的代码中的确存在这种情况，你需要给出解释。
8. (5%) 正确使用 `static` 成员函数、`const` 成员函数和虚函数。
- 与 `this` 无关的成员函数应当是 `static` 的。
 - 逻辑上不修改当前对象状态的 `non-static` 成员函数应当是 `const` 的。
 - 一些函数需要是虚函数，但不应将所有函数都设为虚函数。
9. (5%) 合理设计各个类之间的继承关系。
10. (5%) 尽可能使用构造函数初始值列表。
11. (5%) 非必要不使用显式类型转换（用来构造临时对象的表达式 `Type(args...)` 或 `Type{args...}` 可以被允许）。如若需要，不使用 C 风格的类型转换运算符 `(Type)expr`，而是使用 C++ 的有名字的类型转换运算符 `what_cast<Type>(expr)`。
12. (5%) 变量应在即将使用时才被声明，特别是不应随意使用全局变量（常量除外）。

提交

将 `src`、`include`（如果有的话）和 `CMakeLists.txt` 打包为一个 `.zip` 文件提交到 OJ。如果你自己增加了一些贴图，可以在 `check` 的时候单独发给你的 TA。

我们会在 OJ 上设置一些编译时检测，以 `error` 或 `warning` 的方式给出反馈。这些反馈仅供参考，可能不准确，最终分数仍以 `check` 为准。

`check` 是助教自己 `check`，无需到场。

查重

特别说明：本题将查重。和信院其它代码课类似，抄袭被发现的后果**非常严重**，并且抄袭者与被抄袭者是同等处罚的，请不要抱有侥幸心理。

抄袭来自网络或者 AI 工具的代码也是同等处理的。像“Copilot 莫名其妙帮我补出了和别人一样的代码”这种解释我们不会接受。请合理使用 AI 工具，特别是不要直接使用 AI 生成的代码。