**保卫萝卜项目文档**

**一、项目名称**

保卫萝卜CarrotDefense

**二、项目实现功能**

**基础功能**

* 支持至少3种防御塔和防御塔的删除
* 需要每种防御塔的攻击特效，至少要完成发射物弹道
* 怪物、萝卜生命值显示
* 支持资源功能，资源可用于购买防御塔，资源可通过击杀怪物获得
* 支持每种防御塔的升级，至少可升级2次
* 支持萝⼘的升级，至少可升级2次
* 支持至少3种怪物
* 支持至少2张地图
* 支持背景音乐
* 需要关卡选择界面和保存通关进度记录功能（即已完成哪些关卡，可进⾏哪些关卡，哪些关卡还需解锁）
* 暂停游戏功能
* 支持中途退出时记录当前状态，下次进⼊同一关卡继续上一次游戏进程的功能

**加分项**

* 版本控制和团队协作

合理使用 Git 控制版本，将项目开源至 GitHub；团队成员分工合理平等。

本项目依据团队每个成员所擅长的领域，进行了合理与平等的分工。

* 代码质量和安全

使用单元测试保证代码质量。

单元测试是软件开发中的一种测试方法，用于验证代码中最小的可测试单元（如函数、方法、类等）的行为是否符合预期。它的目标是对每个独立的代码单元进行测试，确保其在各种情况下都能按照预期工作。

* 共享数据类SharedData类

首先，它能实现数据的集中管理，所有共享数据都在一个地方维护，这样可以避免数据分散，确保数据的一致性，减少出错的可能性。比如游戏中的分数、玩家信息等重要数据都可以统一在这里管理。

其次，从开发角度看，这种方式让代码结构更清晰，易于维护。当其他类需要访问共享数据时，有了统一的访问接口，不仅提高了代码的复用性，也让调试变得更加方便。

最后，这种设计在性能上也有优势。它避免了数据的重复存储，优化了内存使用，同时由于数据集中管理，也提高了数据访问的效率。不过要注意的是，在实现时需要考虑线程安全等问题，确保数据访问的安全性。

* 界面精美，UI设计尽量还原《保卫萝卜》游戏本身

**三、成员分工**

任泓臻 2351458 （组长）

项目后端工作统筹，项目任务分工，代码审查，项目框架搭建，网络环境搭建 写地图类（GameScene）、防御塔类及附属子类（DefenseTower）、共享资源类（SharedData）、目标类（Target）运行调试，做项目文档（50%）。

宋文轩 2351872

项目前端工作统筹，写怪物类（Monster）及附属子类、场景选择类（SelectionScene）、目标类（Target），音频引擎，背景音乐与音效，运行调试，做答辩汇报PPT（50%）。

**四、项目技术细节文档**

整个游戏程序有三个主要的类：游戏场景类、防御塔类、怪物类。

各个类的主要成员函数和数据成员如下：

防御塔类（DefenseTower）：

成员函数：

* 产生：点击场景中的可使用位置，弹出建造防御塔的按钮，点击按钮后在该位置产生防御塔。
* 升级：点击防御塔，弹出升级或移除的按钮，点击升级按钮后将原位置的防御塔升级。
* 移除：点击防御塔，弹出升级或移除的按钮，点击移除按钮后将原位置的防御塔移除。
* 攻击怪物：防御塔自产生后，便一直搜寻在攻击范围内且生成序号最小的怪物并发射子弹，如攻击范围内无怪物，则不攻击。

数据成员：

* 攻击半径（Radius）：表示防御塔的攻击范围，到防御塔的距离小于攻击半径（Radius）的怪物将会被攻击。
* 攻击频率（Frequency）：表示防御塔连续两次发射子弹的最小时间间隔。
* 攻击力度（Damage）：表示单次攻击对怪物造成的伤害。

怪物类（Monster）：

成员函数：

* 产生：在场景的指定位置产生。
* 被攻击：接收来自防御塔的攻击并减少血量，当血量减为0时消失。
* 移动：按照场景中的路径移动，当移动到终点时攻击萝卜，同时怪物自己消失。

数据成员：

血量（HP）：大于0且不超过20的整数，怪物产生时血量初始值为20（不同怪物血量的初始值不同），每次被防御塔攻击则血量减少相应值，当血量减为0时怪物消失。

移动速度（Speed）：以这个值在场景中的路径上匀速移动。

场景类（GameScene）：

调用相关函数，实现游戏功能，并为整个游戏的提供背景图、背景音乐等。

**五、必须满足的要求**

* 使用STL容器

使用了vector、string等STL容器。

* 类和多态

游戏中大量使用了继承和多态，以降低开发技术量。

* C++11或以上功能

auto变量、基于范围的for循环、lambda表达式等的使用。