

# 塔防游戏

October 21, 2022

## Abstract

塔防游戏是一种很有趣的游戏。在过去的十多年间, 该类型的游戏一直流行着, 如《植物大战僵尸》、《王国保卫战》、《保卫萝卜》和《明日方舟》等。所以, 让我们写一个塔防游戏吧!

## 1 开始之前

WARNING: All of the code for this project must be your own. You may not copy source code from other students or other sources that you find on the web. Plagiarism will not be tolerated.

- 同项目 1 一样, 抄袭是被严厉禁止的。游戏必须是自己从第一行代码开始写成的, 符合面向对象和泛型程序设计范式的 C++ 程序。推荐使用第三方的 C++ 图形库来完成图形的绘制, 键盘鼠标事件的捕获和处理等等内容, 这样可以集中精力设计和实现游戏的架构。
- 共分四个阶段, 一二阶段不要求有图形界面, 第三阶段才要求完成图形界面。第四阶段则是完成前面阶段的拓展功能, 或者你当然也可以自己提出并实现新的拓展功能, 我们会根据难度进行评分。每阶段依然是两周, 我们会在第三阶段验收前面所有基础功能, 在第四阶段验收拓展功能。
- GUI 可以用 Qt 之类第三方库实现的图形窗口, 也可以用 ncurses 之类库实现的终端 (Terminal) 窗口, 简而言之就是任意发挥, 界面是否漂亮/酷炫不算在基础功能中。但这里我们需要强调, 你实现的项目应该有 C/C++ 代码文件, 使用完全不涉及 C/C++ 的引擎并不符合本课程要求。
- Question 内容是我们对你们可能问题的预演和回答, 并有对本阶段内容的总结。Note 内容是对我们认为可以自由发挥部分的进一步说明, 如 Note2.1, 仅供参考。Extension 内容是第四阶段才要求的拓展功能。

### Question 1.1: 分数构成是什么样的?

各阶段基础功能: 70% (阶段一、二 + 三中对前面阶段 GUI 的实现占 65%, 阶段三中的攻击和受伤效果仅占 5%, 所以各位可以自行斟酌工作量分配。)

各阶段选做内容: 20%, 每个选做内容的难度不同, 会根据实际情况给相应的分数。

代码风格: 10%。



敌方的单位将从路径的起始格子出现, 按照序列, 移动到路径的结束格子。特别地, 不同路径可以共用格子, 包括起始格子和结束格子。

#### Note 2.2

地图的数量和具体图形在阶段一中不做要求, 但请在自己的项目中留好接口, 方便完成潜在的后续要求。

#### Note 2.3

引入路径的概念让地图的设计变得灵活: 如果把所有的路径都设置成从右向左依次排开, 就成为经典的《植物大战僵尸》; 如果把路径设置成有很多转折, 就有一点《王国保卫战》的感觉。

## 2.3 单位

每个阵营都可以产生若干种具有一定功能的单位。不同阵营的单位用各自的方式互相攻击, 为对应的阵营争取胜利。每个单位都有一定的生命值, 如果生命值过低, 单位将被破坏。

我方可以在地图上合适的位置部署单位, 当然, 实现上, 需要游戏为玩家提供一套界面, 让玩家能够选择要在何处部署何种单位。本游戏(包括之后阶段)只要求实现两种我方单位, 即近战塔和远程塔。近战塔被部署在路径上, 攻击范围为 1 (即只攻击相邻格子的敌人), 能阻拦敌方单位的前进; 而远程塔则可以被部署在路径之外, 且与路径相邻的格子上, 默认攻击范围自行设定。此外, 近战塔和远程塔都不具备范围伤害效果, 一次只能攻击一个敌人。Figure 2给出了远程塔和敌人的示例。



Figure 2: 图中只有远程塔和敌人, 近战塔需要布置在路径上。

#### Note 2.4

纯键盘操作当然是可行的。为了更好的游戏体验, 也可以尝试加入对鼠标的支持。

敌方的单位, 当然, 不受我方控制。它们将在路径的起始点随机生成, 按照一定的策略向路径的结束点移动, 并攻击沿途的处在攻击范围内的我方单位。

#### Note 2.5

在阶段一中, 你只需实现默认的近战敌人, 但请在自己的项目中留好接口, 方便完成潜在的后续要求。

#### Extension 2.1: 增加资源设定

你可以设定部署单位要消耗资源, 资源随时间流逝而自动增加。我方也可以手动撤退已经部署的单位。

这里的设定完全可以随各位同学开心, 不过想要强调游戏平衡性的同学可以在这里稍微多思考一点。比如, 可以设置一些特殊的可以产出资源的单位, 也可以设定成在手动撤退单位的时候返还若干资源。

## 2.4 Q&A

### Question 2.1: 我必须在阶段一实现一个 GUI 吗?

并非如此。我们只要求项目在最终有一个可用的 GUI, 并不强求各位同学在阶段一就实现它的部分或者全部。

不过为了避免不必要的重构, 思考如何设计程序的核心逻辑使其能在各种 UI(包括 Graphical User Interface 和 Command Line Interface) 上展示是有必要的。

当然, 我们认为一边实现核心逻辑, 一边实现对应 GUI 功能是最稳妥的办法。

### Question 2.2: 如果我在该阶段没有实现 GUI 的话, 那么我该如何显示呢?

直接在控制台显示日志就可以。

### Question 2.3: 我在阶段一必须要实现出什么?

1. 一张地图, 包含起点终点、路径、以及其他格子。注意我方远程塔对格子的布置要求。
  2. 我方的近战塔和远程塔, 以及部署操作。
  3. 敌方的近战地面单位。
  4. 设定血量、伤害, 不要求显示这些, 但要有基础的敌我双方对战过程。
- 显示可以用日志的格式显示也可以直接使用图形界面。

### Question 2.4: 我可以使⤵用 EasyX、Unreal Engine、NoesisGUI 之类的库吗?

可以, 只要它们为程序的开发者提供的 API 是 C++ 的即可。过于复杂和高级的库可能并不容易学习掌握, 打包和发布程序或许也会相对困难; 过于简单的库则缺乏一些必要的基础设施, 会不必要地加大任务量。

一般来说, 我们会推荐初学者使用一些成熟的、有良好文档的、容易上手的库, 这样的库不必是 Qt, 或许某些 (不为我们所熟悉的) 专为游戏开发设计的库是最适合这个项目的。如果有同学对此有心得, 欢迎联系我们。

## 3 技术指标: 阶段二

我们已经完成了一个塔防游戏! 这听上去棒极了! 所以, 让我们为这个游戏添加更多的内容吧! 游戏应当基于阶段一中已经完成的部分, 只在确实有必要的情况下重构已有的代码。

### 3.1 地图的导入与导出

在阶段一中, 我们可能只硬编码 ( Hard Code ) 了一张地图。现在, 我们要能让游戏支持地图的导入和导出。你只需为自己设计一套格式用来表达地图, 包括尺寸、路径、起点终点位置等, 演示时修改、加载地图即可。

#### Note 3.1: 被导出和等待导入的地图

被导出和等待导入的地图应当是具有特定语法的文本文件 (可以理解成一种配置文件)。这样, 玩家可以:

1. 首先导出默认的地图;
2. 然后可以参考其格式, 对其进行修改后成为新的地图;
3. 最后将新的地图导入到游戏中并正常游玩。

各位同学可以自由设计自己的地图的语法。在这个语法下, 一个合法的文本文件就唯一对应于一张地图。

### 3.2 词缀

在阶段一中, 双方单位的实现略为单调, 现在我们引入词缀的设定。

对于我方, 杀死敌方单位能掉落词缀, 拾取的词缀保存在词缀库中。打开词缀库可以选中词缀对我方塔进行安装/卸下。无论是近战塔还是远程塔都有两个词缀槽, 一个槽可以安装一类词缀, 而安装后塔的属性 (伤害、攻击范围、攻击间隔、其他机制) 将发生改变。

#### Note 3.2: 词缀库选择

一种简单的方式是在右端添加词缀库的 UI, 与部署单位的选择放在一起。

- 对于近战塔, 我们给出三种基本的词缀:
  - 狂暴的: 附加狂暴属性, 装备后伤害和攻击间隔都大幅增加 (比如简单设定为都乘以 2)。
  - 冰系的: 附加冰冻效果, 装备后攻击能使得敌方单位停止攻击、移动一段时间。
  - 群伤的: 附加范围伤害, 装备后能同时攻击多个敌人。
- 对于远程塔, 我们同样给出一种基本的词缀要求实现:
  - 放血的: 装备后对敌人的攻击附加流血效果, 使得敌人在一定时间内持续扣血。

同样, 对于敌方, 我们也要求有实现不同的词缀。敌方单位的词缀可以为空, 也可以自由组合, 为了方便考虑, 我们规定一个敌方单位只能拥有不超过两类词缀。我们这里再给出两类基本的词缀:

- 闪现的: 该单位能发动闪现, 越过我方近战塔的阻挡前进, 但需有冷却时间。

- 神速的：该单位的移动速度要超过默认单位。

#### Note 3.3: 单位的区分

到了这个阶段，如果你有实现相应的图形界面，就需要对不同词缀的单位进行区分了。一个简单的办法是直接在单位头上用文字标出如“群伤的”这类词缀。

#### Extension 3.1: 更多的词缀

在拓展功能中，我们鼓励你们实现其他的词缀。例如，对于我方，你们可以实现能自带回血效果的词缀/能削减敌人伤害的词缀（注意不要让词缀机制互相冲突）；对于敌方，你们也可以尝试实现能为自己加攻速的词缀。

### 3.3 Q&A

#### Question 3.1: GUI 问题?

同阶段一。

#### Question 3.2: 我需要把所有的可选内容都实现吗?

我们鼓励这样做，但是完成一定量（不是凑代码行数）的工作就可以得到相应分数，拓展满分并不需要完成所有可选内容。评分标准视具体情况而定。

#### Question 3.3: 我在阶段二必须要实现出什么?

1. 不同类型的格子和地图的导入导出。
  2. 基础的我方词缀。
  3. 基础的敌方词缀。
- 显示可以用日志的格式显示也可以直接使用图形界面。

## 4 技术指标：阶段三

我们已经完成了一个比较复杂的塔防游戏，如果你前面没有实现 GUI，那么现在是时候加入 GUI 了。此外，在本阶段，还有一些（容易使用 GUI 实现的）实用功能需要完成。

### 4.1 展示单位的攻击和受伤效果

在阶段二中，我们实现了游戏的核心逻辑。但是，如果远程单位在攻击时一声不响，近战单位在攻击时不展示任何动作，单位在受到攻击时也不会有任何变化，那么双方单位的血量就悄悄地减少着。这就会损害游玩的体验。因此，在阶段三中，我们需要在游戏的界面中展示双方单位的攻击/受伤和异常效果：

- 单位发起、受到攻击时，需要绘制出相应的效果。
  - 不必播放一整段的动画，可以简单地把表示这个单位的图形闪烁一下，

- 或者可以在它的头上飘一个数字，表示这个单位受到了多少伤害，
- 又或者在头上绘制血条显示。
- 非常欢迎其他的想法。总之，它只要动了就可以。

#### Extension 4.1: 远程攻击显示

对于远程单位，还需要表示出它在攻击哪个或者哪些敌方单位。可以发射一些飞向敌方单位的飞行物，那么这种攻击不是立即生效的；也可以绘制一道光束之类，这种则属于立即生效的。

#### Extension 4.2: 异常状态显示

对受到的某些异常状态，如流血、冰冻，尝试体现在画面上。可以用头顶飘字的方式，也可以简单在敌方单位图像上添加颜色。实现一个即有分。

## 4.2 Q&A

#### Question 4.1: 我必须在阶段三实现一个 GUI 吗？

是的。但如果很喜欢终端窗口的话，也可以用 ncurses 之类的库。如果不使用相关的库的话，可能需要自己重新造轮子来完成规定动作。

当然，也可以选择未完成某些在已有的程序架构下难以完成的规定动作，但这可能会影响项目的分数。

#### Question 4.2: 我在阶段三必须要实现出什么？

1. 图形界面显示前面阶段的各种单位的部署以及单位的移动。
2. 展示单位的攻击/受伤效果。

## 5 技术指标：阶段四

我们已经完成了一个“有趣”的塔防游戏啦！为了让它更加的有趣，此阶段是扩展阶段，你可以单纯按照前面的 Extension 内容来补充，也可以再加上任何你觉得很酷的想法。

比如，你可以为游戏添加适当的音效。从功能上讲，可以设计酷炫的动画来提升用户体验。任何你觉得有趣的想法都可以加上来！

## 6 如何提问

### 6.1 提出问题的错误方式

(不相干的问题) 电脑出现了故障应该如何处理。

请联系有能力处理这些故障的人，而不是你的程序设计课的助教。

当然，如果是和编程强相关的、软件方面的故障，而你自己毫无头绪，你当然可以问一问。



(可以在搜索引擎里非常轻松地找到答案的问题) 编译器抱怨发现了一个 error 并停止编译

编译器讲的是人话 (虽然是英语), 真的。我们不想成为复读机, 尤其是当问题的答案就在问题的描述里的时候。

当然, 如果你相信问题不出在你的代码上, 互联网上也没有相关的讨论的时候, 欢迎和我们一起探讨。

我的程序又 bug 了, 我怎么办?

原则上, 不应该让别人帮你 debug, 如果实在是无法解决 (搜索了全网都没办法解决, 折腾了好久), 可以寻求 TA 帮助。不应该一遇到 error 就找 TA 寻求帮助。在寻求帮助的时候, 最好需要给出你的运行环境, 你的错误是什么, 你尝试了哪些解决方法。

## 7 参考资料

### 7.1 图形库推荐

如果对 C++ 的图形库并不熟悉, 这里我们从简单到复杂给出三个库供参考:

FLTK 是一个相当轻量级的 C++ 图形库。

wxWidgets 是一个相对轻量级的 C++ 图形库。

Qt 是非常流行的 C++ 图形应用开发框架。

其中的每一个都有足够多的文档和示例程序, 互联网上也有足够多的讨论内容, 可供学习使用。

### 7.2 图形库的安装

对于 FLTK, Windows 用户可能需要安装 MSYS2 才能从中得到预编译的库; 大部分的 Linux 发行版通过自己的包管理系统提供打包好的库和头文件; macOS 用户可以使用 Homebrew 作为自己的包管理器。当然, 总是可以直接下载它的源代码然后自行编译- 这其实不是一件困难的事情, 但是想要完全理解整个过程, 需要综合运用各位同学在计算机系的专业课 (主要是计算机系统基础) 上学到的知识。对 wxWidgets 和 Qt 而言, 在主流的操作系统下都有打包好的安装程序。

### 7.3 其他有用的工具

Address Sanitizer 是一个开箱即用的内存安全错误检查工具, 它被集成在包括 MSVC 在内的所有主流编译器中。它对于找出 C++ 程序中的内存错误很有帮助。

clang-format 是一个代码排版工具。可以使用它把 C++ 代码排版成自己喜欢的样式, 很多主流编译器比如 CLion 和 VS 都支持。

clang-tidy 是一个代码风格检查工具。它可以就代码的不良风格给出警告。

-Wall 是 gcc/clang 编译器的编译选项。开启它将得到更多来自编译器的警告, 这有助于在编写代码的过程中尽早发现问题。类似的选项还有很多, 可以参考[这里](#)或者查询编译器的在线手册 (manual)。



## 7.4 Qt 教程

<https://www.bilibili.com/video/BV1g4411H78N>

## 7.5 一些开源的实现

<https://github.com/exmex/KingdomRushFrontiers>

<https://github.com/marblexu/PythonPlantsVsZombies>

<https://github.com/ErLinErYi/PlantsVsZombies>