科廷学院信息技术学院

改编自科廷大学 计算学科

Unix 和 C 编程(UCP1000)

2024 年第 1 学期

作业一

重量:单位的15%

1 简介

本次作业的任务是设计、编码和测试一款受经典谜题启发的游戏
使用 C 编程语言（C89 标准)的游戏“激光坦克”。总之,您的程序将:

- ·从命令行读取参数并将其用于游戏配置。
- ·创建动态分配的2D 字符数组来制作简单的基于ASCII 的游戏。
- ·接收用户输入以控制游戏流程。
- ·编写合适的makefile。如果没有 makefile,我们将假设它无法编译。

2 代码设计

您必须使用 C 注释 (/* ... */) 彻底记录您的代码。对于您定义的每个函数和声明的每个数据类型（例如,使用 struct 或 typedef）,在其正上方放置注释,解释其用途、工作原理以及与其他函数和类型的关系。总的来说,这些评论应该解释您的设计。

您的代码的结构应该使每个文件都有明确的范围和目标。例如，“main.c”应仅包含主函数，“game.c”应包含处理和控制游戏功能的函数,等等。函数应该位于合理适当的文件中,并且您将使用 makefile 链接它们。不要将所有内容放在一个源代码文件中。确保正确使用头文件和头防护。切勿直接包含 .c 文件。

确保释放 malloc() 函数分配的所有内存。使用 valgrind 检测任何内存泄漏并修复它们。内存泄漏可能不会破坏您的程序,但如果有的话,仍然会受到惩罚。如果不使用malloc,就会丢分。

请记住,如果一个完整工作的程序编写混乱、结构较差、包含内存泄漏或违反许多编码标准,则可能会丢失重要的分数。

3 学术诚信

这是一项可评估的任务,因此有严格的规则。您不得在完成任务时寻求或接受任何其他人的帮助。您在任何时候都不得向本单元注册的其他学生展示您的作品,否则他们可能会从中获得不公平的优势。这些事情都被认为是抄袭或串通。为了安全起见,永远不要向公众发布您的代码。

不要从互联网资源或人工智能代码生成器复制 C 源代码。
从互联网上复制代码而不引用将被视为抄袭。如果您包含来自其他来源且具有有效引用的代码,则该部分代码将不会被标记,因为它不是您的作品。

工作人员可以帮助您了解一般的单元材料,但不允许任何人帮助您
解决本规范中提出的具体问题。作业的目的是让你自己解决它们,以便你从中学习。请参阅科廷学院的

有关学术不端行为信息的学术诚信政策（包括
剽窃和串通）。

您将需要参加快速面试以回答有关您的计划的问题。为了对您的作品进行评分,您需要解释其工作原理。您无法向导师充分解释的任何代码都不会获得分数,并且可能会在学术不端行为调查中作为证据。

4 任务详情

4.1 快速预览

您将实现一个受经典益智游戏“激光坦克”启发的简单游戏。
有关规范的更多详细信息将在后续部分中进行解释。

4.2 命令行参数

请确保可执行文件名为“laserTank”。您的可执行文件应该接受八个
(8) 命令行参数/参数：

LaserTank <行大小> <列大小> <玩家行> <玩家列>
<玩家方向> <敌人行> <敌人列> <敌人方向>

- <row_size> 和<col_size> 确定可播放地图的大小。 <row_size> 和 <col_size> 的最小值和最大值分别为 5 和 25。您需要这些数字来动态分配 2D 字符数组的内存。 （使用 malloc() 函数）

- <player_row> 和<player_col> 是玩家坦克的坐标。您可以使用这些数字作为地图的数组索引。坐标<0,0>位于地图的左上角。

- <player_direction> 是玩家开始游戏时所面对的方向。有 4 个选择：n、s、e、w（小写字母），源自单词“北”、“南”、“东”和“西”。

- <enemy_row>、<enemy_col> 和 <enemy_direction> 与前面的含义相同

解释。然而,这些参数配置了敌方坦克。

这是一个示例：./laserTank 10 20 1 1 w 5 10 n

> 注意:请记住,可执行文件的名称存储在变量 argv[0] 中。

2.1地图

该映射可以从二维字符数组派生,该数组已根据命令行参数动态分配。

2.2主界面

运行该程序后,它应该清除终端屏幕并打印地图以及说明：

这是用户玩游戏的界面。用户可以在句子“action:”之后键入命令。每次用户输入命令时,程序都应相应更新地图并刷新屏幕。 （清除屏幕并重新打印所有内容）

2.3用户输入

用户只需为每个命令键入 1 个字符（小写）。以下是可能的命令的列表：

- w 将玩家移动到上方一个方块。如果玩家坦克没有面朝上,此命令只会使坦克面朝上。 将玩家移动到下方一格。
  如果玩家坦克不是面朝下,此命令只会使坦克面朝下。“a”将玩家向左移动一个格。如果玩家坦克没有面向左侧,此命令只会导致坦克面向左侧。

- d 将玩家向右移动一格。如果玩家坦克没有面向右侧,此命令只会使坦克面向右侧。

- f 使玩家坦克向它所面对的方向发射激光。有关更多详细信息,请参阅下一节“激光动画”。另外,请观看“作业演示”的补充视频。

·任何其他字符都将被忽略。程序还需要刷新界面才能

输入新命令。如果需要,您可以添加警告消息。

> 注意:如果玩家尝试移动到与地图边界或敌方坦克相同的位置,则不应执行任何操作。 （玩家位置保持不变）

2.4激光动画

当用户提供命令“f”时,地图上会出现一个代表激光的附加字符。如果水箱面向水平方向,则可以使用字符“-”,如果水箱面向水平方向,则可以使用字符“|”如果水箱面向垂直方向。当激光发射时,程序应该触发激光从玩家坦克前方移动的简单动画。激光将继续前进,直到击中边界或敌方坦克。当激光击中敌方坦克时,其字符应替换为“X”。

为了制作简单的动画,可以使用Moodle中提供的UCPSleep函数。
这个想法是一次将激光的位置移动一个格,中间有一个非常短暂的休眠期。我们建议睡眠时间少于 0.5 秒,以达到合理的速度。当动画仍在运行时,程序不应显示游戏说明,也不应提示用户输入任何命令。

应使用默认颜色以外的字体颜色将激光打印在地图上。

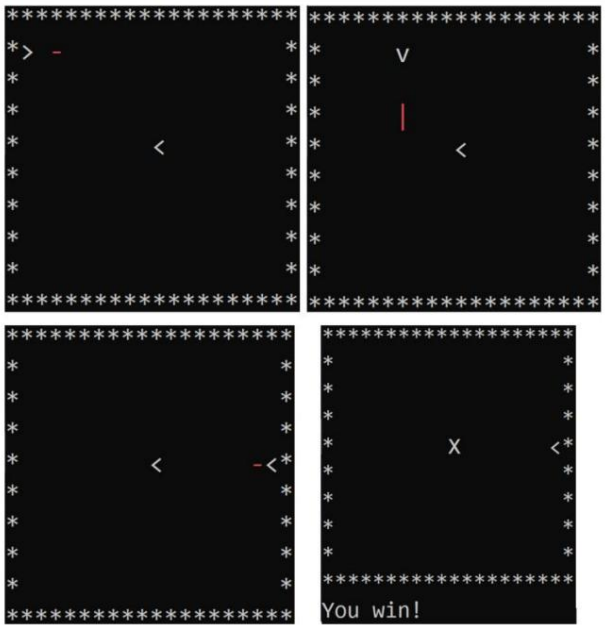
> 注意:如果敌方坦克正好位于玩家前方（没有距离）,则无需显示激光,敌方坦克应立即被摧毁,游戏应结束。

2.5获胜/失败条件

游戏将继续进行,直到玩家获胜或失败。这是每个结果的条件:

- 获胜:玩家坦克设法用激光击中敌方坦克。只有当玩家坦克面向敌方坦克发射激光时,无论它们之间的距离如何,才会发生这种情况。敌方坦克的角色将变成角色“X”（大写X）。

- 失败:玩家移动到敌方坦克前面（面向玩家）。敌方坦克会立即向玩家发射激光（带有激光动画）。动画发生时玩家无法移动。玩家的角色将变成角色“X”。



发生这种情况时,请不要忘记在程序结束之前释放通过 malloc() 函数分配的所有内存。

2.6生成文件

你应该按照讲座和实践中解释的格式编写makefile。它应该包括 Make 变量、适当的标志、适当的目标和干净的规则。我们将通过 makefile 编译您的程序。如果缺少 makefile,我们将认为该程序无法编译,并且您将失去分数。

2.7假设

出于简化目的,您可以在此作业中使用以下假设:

·玩家坦克和敌方坦克的坐标不会与提供的命令行参数重叠。但是,您仍然需要检查坐标是否在地图边界内。

·坦克的坐标也不会与地图的边界重叠。

·开始游戏后,玩家坦克不会立即出现在敌方坦克的前面。
游戏。 （没有即时损失）

·任何命令或参数始终采用正确的数据类型并采用小写形式（对于字母）。

3 评分标准

这是供您指导的标记分布:

代码结构: ·包含任何非

C89 语法。使用 -ansi 和 -pedantic 标志进行编译,以防止丢失标记。 （5分）

·结构正确的makefile （10 分）

·程序可以用makefile 编译并成功执行,无需立即执行。
吃得很饱（5分）

·使用足够的代码内注释（5 分）

·整个程序可读性强,结构合理,使用多个文件（5
分数）。

·避免不良的编码标准。 （10 分）

一些最常见的错误是：

–使用全局变量

–调用exit()函数突然结束程序

–在 switchcase 语句中使用“break”

–使用“继续”

·无内存泄漏（10 分）

·请使用valgrind 检查是否有泄漏。如果你的程序非常稀疏或者不使用
任何 malloc(),你在这个类别上都会得到零分。

功能：

–能够正确读取和解释命令行参数。这包括澄清论点的数量和价值。如果出现任何问题,则会在终端上打印错误
消息并结束程序。否则,应该相应地分配地图数组。 （10 分）

–地图及其组件的正确字符使用。 （5分）

-玩家坦克根据玩家命令正确移动和改变方向。 （10
分数）

–使用 UCPSleep 功能,激光在地图上正确显示动画。 （5分）

– （奖励标记)激光每移动一格就会改变颜色。 （提示:使用静态变量来跟踪。） （5 分） （总分仍上限为
100 分。）

–能够使用更新的地图/说明可视化来刷新屏幕
用户输入命令的时间。 （5分）

–正确执行获胜/失败条件。 （确保释放内存
结束程序之前的内容） （10 分）

4 简短报告

如果您的计划不完整/不完美,请写一份简短的报告,解释您在作业中所做的事情。这份报告没有评分,但它将对我们评分您的作业有很大帮助。请确保您的报告正确反映您的工作（不夸张）。不诚实的报告将导致学术不端行为。

5 最终检查和提交

完成作业后,请确保它可以在基于 Linux 的操作系统上编译和运行。如果您在其他环境（例如在 Windows 操作系统上)上完成作业,那么您有责任确保它在 Linux 环境中运行。 Ubuntu 将用于测试。如果不兼容,它将被视为“无法工作”,并会受到一些处罚。您必须提交包含以下内容的 tarball：

·您的基本作业文件 提交所有.c 和.h 文件以及您的makefile。请

不要提交可执行文件和对象 (.o) 文件,因为我们无论如何都会重新编译它们。

·简要报告（如果适用）- 如果您想撰写有关作业完成情况的简要报告,请将其另存为 PDF 或 TXT 文件。

tarball 的名称应采用以下格式：

<学生 ID>_<全名>_Assignment1.tar.gz 示例:12345678_Antoni-Liang_Assignment1.tar.gz

请确保您提交的内容完整且未损坏。您可以重新下载提交并检查是否可以再次编译并运行程序。损坏的提交将立即归零。您可以多次提交,但只会标记您最新提交的内容。

作业结束