

程序设计基础：大作业 - SnakeFoP

开始时间：2023.10.25 22:00

截止时间：2023.12.24 23:59

引言

欢迎各位同学来到程序设计基础课程的大作业部分！

在大作业中，各位同学需要二人一组，共同完成一个具有一定工程量的项目。在这个过程中，大家需要综合运用课上讲述的和此前作业中的知识，尝试并学习如何在实际工程中进行程序设计。在评分中，我们一方面关注系统功能的正确性、完备性和鲁棒性，另一方面也会对项目附属的文档、视频以及二人的合作进行评价。我们期待大作业能够锻炼各位同学落地实践、精益求精、团队协作和精准表达的能力。

大作业文档的解释权归属于教学团队，如果你对文档有疑问，或者找到了文档中的错误，请与助教联系！期待大家的大作业成果！

1 概述

在本次大作业中，各位同学将设计并完成一款经典的街机游戏——**贪吃蛇**。此类游戏发源于1976年，在上世纪90年代伴随移动电话的兴起而流行起来。如果你们曾使用过功能机时代的手机，或许能在其中发现它的身影。

贪吃蛇游戏的主体是一条细长的直线（即“蛇”），它会不停前进，玩家只能操控其头部方向。场地中有**食物**的存在，蛇路过时会吃掉食物，其身体因此会变长一些，让游戏难度逐渐变大。在操控中要避免蛇碰到自身或者障碍物，这两种情况都会导致游戏结束。

在大作业中，你们需要编写一个**完整的命令行贪吃蛇游戏**（Snake - Fundamentals of Programming, SnakeFoP），包含了基础游戏环境和其他周边功能。

2 基础游戏环境

首先，你们需要将上述的游戏逻辑和过程表现为命令行的程序，并实现恰当易用的界面交互。

2.1 主界面

用户打开程序后，应当看到一个主界面，并能够通过不同的输入进行不同功能选择。用户输入“g”表示开始游戏；用户输入“q”则表示退出游戏。当用户完成一局游戏后，应当回到主界面。程序需要给出必要的提示，并能够妥善处理用户的非法输入。

2.2 游戏界面绘制

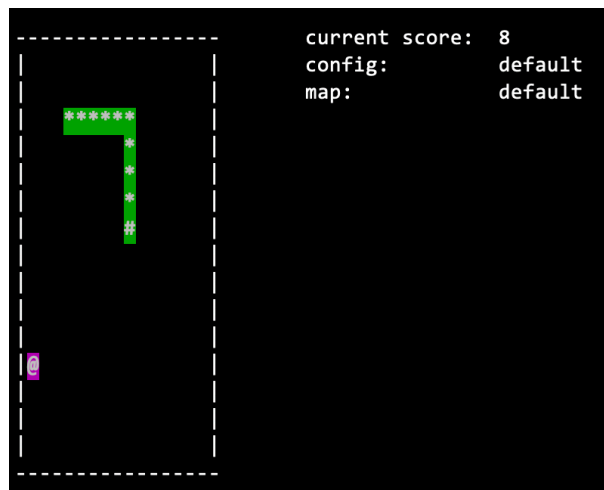
游戏地图的大小为 15×15 ，四周的墙壁均为实墙（即撞上会结束游戏），墙壁用“|”和“-”在命令行中表示。

蛇的初始长度为 4，蛇头在地图的正中央，面向右侧。蛇头用“#”表示，蛇身用“*”表示。蛇的背景色为绿色，当游戏结束后，背景色则变为红色。

地图中随机生成食物，其不会生成在蛇已经占据的格子中。每刻场上只有一个食物，用“@”表示。食物有不同的分值：有 60% 的概率为 1 分食物，表示为蓝色的符号；有 30% 的概率为 2 分食物，表示为紫色的符号；有 10% 的概率为 3 分食物，表示为黄色的符号。

同时，游戏界面中应当展示出当前的分数、游戏配置名称（当完成**自定义配置功能**一节后）和地图名称（当完成**地图配置功能**一节后）。

一个最基础的实现如下：（你们的程序满足以上要求即可，不需要和这个案例完全一致，我们欢迎更好看的界面绘制!）



2.3 蛇的移动

蛇每 1 秒钟朝其面向的方向移动 1 格，每 1 秒钟需要清空命令行并重新绘制游戏界面。用户可以通过键盘的 w、a、s、d 键分别控制蛇头朝向上、左、下、右方向。

在游戏的过程中，用户点击空格可以暂停游戏，再次点击空格可以继续游戏。在暂停状态下点击“q”可以直接退出游戏，返回主界面。

蛇每吃到一个食物，便会变长一格并获得食物对应的积分。如果撞墙或撞到自身则结束游戏。程序需要在游戏结束后给出适当的提示语，并指引用户通过点击任意键返回主界面。

在游戏过程中，程序需要正确地处理用户的异常输入（比如忽略上述功能以外的任何输入，忽略和当前蛇头方向相同或相反的输入等）。如果 1 秒内用户输入了多个方向键，则应选取最后一个合法的输入。

3 其他周边功能

在完成上述功能后，你们的贪吃蛇程序应该已经可以正常进行游戏了。不过既然要实现完整的命令行游戏，为了用户更好的游戏体验，你们还需要提供一些周边的功能。

3.1 自定义配置功能

一款好的游戏应当提供足够大的自由度，供用户根据自己的喜好进行定制。因此，希望你们加入贪吃蛇游戏配置文件的创建、保存和载入功能。

可以自定义的配置包括：

- **游戏难度**：一个 1~10 的正整数，即蛇的移动速度。蛇每移动一格的时间是难度的倒数。即，假设游戏难度为 s ，则每隔 $t = \frac{1}{s}$ 秒蛇即会向前移动一格。
- **随机种子**：一个无符号整数，用于生成随机数。如果设定为 -1 即表示不固定随机种子，使用运行时的时间（即 `time` 函数的返回值）作为随机种子。
- **食物数量**：一个 1~5 的正整数，表示同一时刻场地中的食物数量。
- **食物生成概率**：三个 0~1 的浮点数，其和为 1，分别表示 1 分、2 分和 3 分食物的生成概率。

在主界面中，用户可以输入“i”进行自定义配置的新建。然后，程序要求用户需要输入配置文件的名称。名称只能由英文字母、数字和下划线组成，且如果用户输入的是单独的“q”，则立即结束新建配置的过程。

输入合法名称后，程序要求用户依次输入以上配置的值，能够验证输入的合法性并恰当处理用户的错误输入。输入完毕后，程序随即将配置文件保存下来，文件名为 **XXX.config**，保存在游戏根目录下的 **config** 文件夹中。

例如对于第 2 节描述的配置（即默认配置），应当表示为：

```
1 // In "config/default.config":
2 1
3 -1
4 1
5 0.6 0.3 0.1
```

在主界面中，用户可以输入“u”进行配置载入。然后，程序需要告知当前的配置文件名称，并要求用户输入新的配置文件名称。如果存在对应的配置文件，应当载入并应用于此后的游戏中；如果不存在配置文件，则需要提示用户并要求再次输入。用户可以通过输入“q”退出配置载入过程。

游戏每次启动时，应当载入用户上次使用的配置。

3.2 地图配置功能

地图的设计也应当是游戏定制化的一部分。你们需要用命令行中的可视化方法帮助用户更直观地进行地图的创建和保存，并正确实现地图的载入。

地图信息包含：

- **地图大小**：两个 8~20 的正整数，分别表示横向和纵向的格子数。
- **四面墙的虚实**：四个 0 或 1 的值，分别表示上、下、左、右四面墙的虚实设定。1 表示实墙，0 表示虚墙。撞上实墙会直接结束游戏，而碰到虚墙则会穿墙到达同一行或列另一端的格子位置（即可以理解为地图形成了一个环）。
- **障碍物的数量**：一个正整数，表示障碍物的数量，地图最多可以有 15 个障碍物。蛇撞到障碍物同样会结束游戏。

- **障碍物的坐标**：多行数据，每行表示一个障碍物的位置，有两个正整数，分别表示其所在的行或列坐标，取值为 0 到 $n - 1$ （若 n 为地图行或列的总格子数）。

例如对第 2 节描述的地图（即默认配置），应当表示为：

```
1 // In "maps/default.map":
2 15 15
3 1 1 1 1
4 0
```

另一种合法的地图文件为：

```
1 // In "maps/example.map":
2 20 12
3 1 1 0 0
4 3
5 12 8
6 3 10
7 0 6
```

在主界面中，用户可以输入“n”进行地图的新建。然后，程序要求用户需要输入地图文件的名称。名称只能由英文字母、数字和下划线组成，且如果用户输入的是单独的“q”，则立即结束新建地图的过程。

输入合法名称后，程序要求用户输入地图的大小。确认输入合法后，程序需要清空命令行并用“|”和“-”绘制出对应行数和列数的表格。表格的四个外边框用红色描绘，表示默认的四面实墙。然后，用户可以输入指令进行地图自定义。合法的指令包括以下几类：

指令头	参数	解释	错误情况
o	X Y	X 和 Y 均为正整数。表示新建一个障碍物,其所在的行和列的坐标分别为 X 和 Y。	X 或 Y 超出地图大小范围。
p	X Y	X 和 Y 均为正整数。表示删除一个障碍物,其所在的行和列的坐标分别为 X 和 Y。	X 或 Y 超出地图大小范围/所在坐标没有障碍物。
w/a/s/d	V	V 为 0 或 1。分别表示将上 (w)、左 (a)、下 (s)、右 (d) 墙设置为虚墙 (0) 或实墙 (1)。	
f		表示结束地图输入，并保存地图文件。	
q		表示直接退出地图新建，返回主界面。	

用户每输入一句指令后，程序需要验证其合法性，并做出对应的反应。如果一个格子有障碍物，则将对应的格子用蓝色的任意字符填充，如果一个墙变为了虚墙，则将表示该墙的“|”或“-”由红色变为绿色。程序也需要恰当处理用户的非法输入。

例如对上文提到的“example.map”，用户输入 20、12 后，可以通过下列指令完成地图创建：

```
1 o 12 8
2 o 3 10
3 o 6 9
4 a 0
5 s 0
6 d 0
7 p 6 9
```

```
8 | s 1
9 | o 0 6
10| f
```

完成地图编辑后，程序随即将地图文件保存下来，文件名为 **XXX.map**，保存在游戏根目录下的 **maps** 文件夹中。

在主界面中，用户可以输入“m”进行地图载入。然后，程序需要告知当前的地图文件名称，并要求用户输入新的地图文件名称。如果存在对应的地图文件，应当载入并应用于此后的游戏中；如果不存在地图文件，则需要提示用户并要求再次输入。用户可以通过输入“q”退出配置载入过程。

游戏每次启动时，应当载入用户上次使用的地图。

3.3 游戏记录回放功能

如果玩家完成了一局精彩的游戏，能够记录下来供以后欣赏的话该有多好！所以尝试在你们的贪吃蛇游戏中加入记录和回放功能吧！

具体而言，在玩家进行游戏的时候，让程序记录下游戏的各种事件。在一局游戏结束后，用户可以点击“b”键保存本局游戏的记录（点击除 b 以外的键仍然是返回主界面）。然后，程序要求用户输入记录文件的名称。名称只能由英文字母、数字和下划线组成，且如果用户输入的是单独的“q”，则立即结束保存记录的过程。输入合法名称后，程序随即将记录文件保存下来，文件名为 **XXX.rec**，保存在游戏根目录下的 **record** 文件夹中。

对于记录文件的格式，你们可以自行设计，一种参考的方法如下：

文件前两行为两个字符串，分别表示配置文件和地图文件的名称。然后是由许多行字符串构成的记录，每行的开始为一个正整数 T，表示事件发生的时间，蛇每动 1 次增加 1。此后的部分表示具体的事件，包括以下几类：

事件表示	解释
F X Y V	X、Y、V 均为正整数。表示生成了一个分值为 V 的食物，其所在的行和列坐标分别为 X 和 Y。
W	用户点击 w 键使得蛇头朝上
A	用户点击 a 键使得蛇头朝左
S	用户点击 s 键使得蛇头朝下
D	用户点击 d 键使得蛇头朝右

在主界面中，用户可以输入“r”进行记录重放。然后，程序要求用户输入记录文件名称。如果存在对应的记录文件，则应载入其记录的配置文件和地图文件并绘制游戏界面，随后根据记录的内容自动复现出游戏的过程。在重放的过程中用户可以通过键入“q”回到主界面，其余的所有输入均应被忽略。

在重放时，程序也需要恰当处理错误的情况。例如当记录中的配置文件或地图文件不存在时、记录中有错误的条目时或者重放的中途便结束游戏时。

4 扩展功能

做到这里，恭喜你们已经成功搭建出了一个十分完整且好用的贪吃蛇游戏，完成了大作业全部的功能要求！如果你们在其他课程上面临着很大的压力，我们强烈建议你们跳过这一节，直接查看

大作业的提交要求，并将时间精力用于其他课程的学习。

而如果你们仍有余力，且愿意探索更多有趣的功能和内容，可以尝试一下本节所述的扩展功能。再次提醒：**扩展功能不是必须实现的部分，有较大的难度、灵活性和工作量，请合理安排时间！**

4.1 贪吃蛇 AI 的编写

游戏 AI 一直是人工智能中一个很重要的领域。在很多游戏中，计算机科学家们都在努力建造出达到甚至超过人类水平的 AI，并已经取得了许多成功（例如深蓝攻克了国际象棋、AlphaGo 攻克了围棋等）。相比于这些对抗性的棋类，贪吃蛇自然没有那么复杂，但编写一个贪吃蛇的 AI 仍需要一定的思考和创造力。

一个很容易想到的方法是，通过搜索找到蛇头到食物的最短安全路径，并不断调用这个搜索算法，实现一个贪心的策略。这个方法比较简单、易于实现，且在很多情况下颇为有用，是编写贪吃蛇 AI 一个很好的起步。

但当你实现后会发现一个问题：当蛇比较长的时候，有时并不存在一个安全的路径，遑论最短的安全路径了（例如四面墙都是实墙，蛇身将场地完全分隔为两半，而蛇头与食物在两个半区中）。这是因为贪心的策略往往导致短视，为了一时的最短牺牲了未来的利益。那么如何才能让 AI 具有长远的目光呢？提示，或许让 AI 不总是走最短路径，而是适当的“绕远”，能够缓解这个问题。

当然，贪吃蛇 AI 也不必非要基于搜索。如果你对图论有所了解，或许知道 Hamilton 回路的概念。这个概念或许能完全解决贪吃蛇的问题，真正做到满屏。但相对的，这个方法失去了一些乐趣，因为它完全不需要考虑食物的位置，且吃一个食物可能需要走很远的路程。那么，有没有可能在 Hamilton 回路的基础上，走一些捷径从而更快吃到食物呢？

我们也鼓励你们探索其他可能的算法，来构建更强大的贪吃蛇 AI。如果你们决定编写贪吃蛇 AI 的话，请注意：

- 请在主界面增加一个接口。用户可以输入“v”进入自动游戏模式，此时需要适当调快蛇前进的速度，并调用你们编写的 AI 完成贪吃蛇游戏，且这期间用户的所有输入均应被忽略。如果你们编写了多个 AI，请增加让用户选择 AI 的接口。
- 为了简化问题，在四面实墙且没有障碍物的地图上运行你们的 AI 即可。如果你们的 AI 能在更复杂的地图上工作，会带来一定的加分。
- 若你们实现了两个或以上的 AI 算法，你们可以固定地图和随机种子，用**平均分数 (Score)** 和**获得每分的平均步数 (Steps per score)** 两个指标比较并分析一下不同 AI 的性能，总结它们各自的特点。

4.2 图形界面绘制

此前的大作业都是通过命令行和用户进行交互的。然而在如今，命令行的游戏已经很难见到了。图形化是当代游戏必不可少的一步。你们可以尝试将贪吃蛇游戏界面转换为图形用户界面 (Graphic User Interface, GUI)。

如果你们决定走上这条复杂但有趣的道路，请注意：

- GUI 界面可以基于其他框架完成，但编程语言必须使用 C++。
- GUI 界面需要至少实现上文描述的功能，如果缺少则会扣除对应功能的分数。

- 如果你们完整完成了 GUI 界面绘制，则可以无需完成命令行的界面编写，且不会损失分数。

Hint: 一个可以使用的框架为 Qt，这是一个 C++ 的 GUI 编程框架，相对比较简单易用。不过 Qt 编程需要一些面向对象（OOP）的编程知识。你们也可以尝试搜索其他 C++ 的 GUI 框架。

4.3 其他扩展功能

你们也可以创新并完成本文没有提到的其他扩展功能。如果你们实现了这些扩展功能，请在大作业报告中阐释功能的意义、具体内容和相应的技术实现。我们将会根据功能的创造性、工作量和完成度给予评价。

5 提交要求和评分细则

5.1 组队要求

本次大作业**两人一组**，助教会在退课阶段结束后统计分组信息，提交组员信息后不能再调整组队。除教学团队允许的特殊情况外，大作业不接受单人或超过两人一组的提交。在提交时，每组**有且仅有 1 人**提交要求的材料即可，请不要同时提交，因为这会为评分过程带来不必要的额外工作量。

5.2 提交要求

请在网络学堂上提交一个 **ZIP 压缩包**。文件名为【**组员一学号_组员一姓名_组员二学号_组员二姓名.zip**】。压缩包内应包括但不限于：

1. 程序源文件，放置在 **src** 文件夹下，可以有多级目录，应包括全部工程编译所需要的文件及必要的资源文件。
2. 大作业报告，为一个 **PDF 文档**，放置在压缩包根目录下，内容应包括但不限于：
 - 小组人员信息（姓名、学号、班级）
 - 项目运行环境（系统类型、系统版本、编译器版本、所需的最低 C++ 标准等）
 - 功能实现逻辑（如代码框架、核心函数的功能等）
 - 扩展功能说明（如有）
 - 小组分工情况（请尽可能详细，这部分将作为组内同学评分的依据）
3. 演示视频，放置在压缩包根目录下，内容应包括对全部功能（包括可能的扩展功能）的展示。要求长度不超过 3 分钟，视频大小尽量不超过 60MB，推荐提交 mp4 格式。

如果上述文件过大无法上传至网络学堂，请在网络学堂提交一个指向大作业提交内容的清华网盘链接。如你选择这种方式提交，请不要在截止日期后更改清华网盘链接指向的内容，否则将按照迟交处理。

5.3 评分细则

大作业的满分为 100 分，具体分配如下：

- 功能分数（80%）：主要考察功能的正确性、完整性和鲁棒性。

- 基础游戏环境 (40%): 包括主界面、游戏界面、游戏逻辑。
- 其他周边功能 (40%): 包括自定义配置、地图配置和游戏记录回放功能。
- 文档分数 (20%)
 - 代码风格 (5%): 变量函数命名规范一致, 正确适当使用空格和缩进, 代码可读性强。
 - 报告视频 (10%): 报告和视频简洁明了, 有效说明了功能实现的思路 and 效果。
 - 分工合作 (5%): 两人分工合理, 组员对小组做出了足够的贡献。

扩展功能 如果你们完成了扩展功能, 教学团队将会根据工作量和完成度进行评分。扩展功能的分数可以补充大作业的扣分, 上限为大作业获得满分, 不可补充其他教学环节的分数。

大作业展示 如果你们被邀请并最终完成了课堂的大作业展示, 则可以获得文档分数部分的 5 分加分, 上限为文档分数部分获得满分。

5.4 补交政策

如你们错过了提交截止时间, 请通过给助教发送邮件 (wt22@mails.tsinghua.edu.cn) 进行补交。请在邮件中写明组员的姓名和学号, 并在附件中提交上述压缩文件。

补交的时间是助教收到邮件的时间。如果在截止日期后第 k 天的 23:59 前提交, 则会将最终的得分乘以 $(1 - \frac{k}{15})$ 的折扣系数。

截止日期第 14 天 (即 2024 年 1 月 7 日, 第 16 周周日) 23:59 后不接受补交, 大作业得 0 分。

6 学术规范提示

大作业具有一定的难度和代码量, 我们欢迎同学之间的相互学习和指导。但请注意不要违反学术规范, 即: 可以参考其他同学的思路或代码, 但不要直接复制粘贴!

如果你曾在大作业中参考了其他同学的工作, 请进行恰当的引用, 即在大作业报告末尾进行**声明**。如果你曾将你的代码交给其他同学参考, 也请在大作业报告末尾进行**声明**, 告知参考了这份代码的同学 (包括可能通过间接渠道传播到的其他同学)。

我们欢迎使用 GitHub 等代码托管平台进行更好的项目管理和团队合作, 但请所有同学不要把自己的代码托管至任何公开的仓库中, 托管至私有仓库后也请不要给予其他同学访问权限。

我们会**对所有组的代码进行查重**。如发现代码雷同的情况, 涉及雷同的各组 (不论拷贝者和被拷贝者) 均会得到同样的处罚, 除非各组均对雷同片段进行过声明。情节严重者, 教学团队有权上报至院系和学校, 按照相关规定严肃处理。

7 写在最后

本次大作业是教学团队首次命题, 我们希望尽可能做到知识性和趣味性的平衡, 让大家收获更丰富的知识和探索的能力。但初次尝试难免可能有不详尽之处, 如果同学们对大作业的安排或内容有想法或建议, 欢迎向助教团队反馈, 帮助我们将课程建设地越来越好!