

2025 年秋季学期《程序设计基础》课程设计

兴军亮教师课程团队

清华大学计算机系

jlxing@tsinghua.edu.cn

摘 要：课程设计是《程序设计基础》课程的重要教学和考核环节，旨在帮助同学们巩固和综合运用课堂上所学的编程知识。在此次大作业中，同学们需要自行进行项目规划、需求分析、程序设计与实现等多个环节。通过完成大作业，同学们将提高编程技能，培养解决实际问题的能力，并为后续课程奠定扎实的基础。

关键词：程序设计基础；课程设计；课程大作业

1 概述

本学期《程序设计基础》课程设计需要**双人合作**¹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。

2 基础游戏环境

打砖块是一种电子游戏。屏幕上部有若干层砖块，一个球在屏幕上方的砖块和墙壁、屏幕下方的移动短板和两侧墙壁之间来回弹，当球碰到砖块时，球会反弹，而砖块会消失。

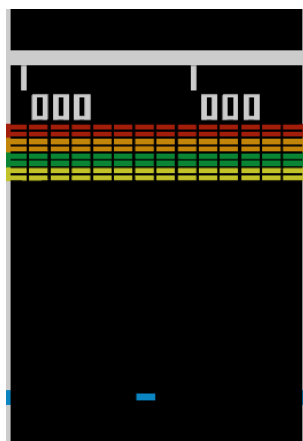


图 1: 游戏“打砖块”的经典界面

在个人电脑刚刚诞生的时代，经典的打砖块游戏于 1976 年由美国雅达利公司发行。游戏设计正是创立苹果电脑的史蒂夫·乔布斯与斯蒂夫·沃兹尼亚克两人。在打砖块游戏中，玩家要

¹个别小组三人，我们会综合考核项目整体工作量和完成人数进行项目打分

控制屏幕下方的板子，让“球”通过撞击消去所有的“砖块”，球碰到屏幕底边就会消失，所有的球消失则游戏失败。把砖块全部消去就可以破关。打砖块游戏作为计算机历史上的经典游戏，其规则简单，玩法有趣，甚至是检验人工智能算法的经典环境。打砖块游戏最终被来自英国的 DeepMind 团队研发的人工智能程序攻克，其背后的强化学习算法更是对后续的 AlphaGo, ChatGPT(RLHF) 等革命性产品产生了深远的影响。

在本次作业中，你们需要将经典的**打砖块**（Breakout）游戏实现为一个可在个人电脑上执行的程序，并提供友好的用户交互。程序需包含主界面、游戏界面绘制、球与挡板控制、碰撞与反弹、砖块消除与计分、关卡与难度等基本要素。

2.1 主界面

用户打开程序后，应看到主界面：输入 g 开始游戏，q 退出程序。一局结束后应回到主界面。主界面可展示当前配置与关卡入口（如有）。

2.2 游戏界面绘制

默认地图大小为 9×18 （不含左右与底部边界），墙体采用“|”与“-”表示。挡板使用连续字符（如“=====”）表示，球使用两个英文括号“()”表示，砖块使用不同字符区分类型（如耐久砖“#”、普通砖“@”、不可摧毁砖“*”）。界面每个时间步重绘，并在上方显示分数、关卡与剩余生命。

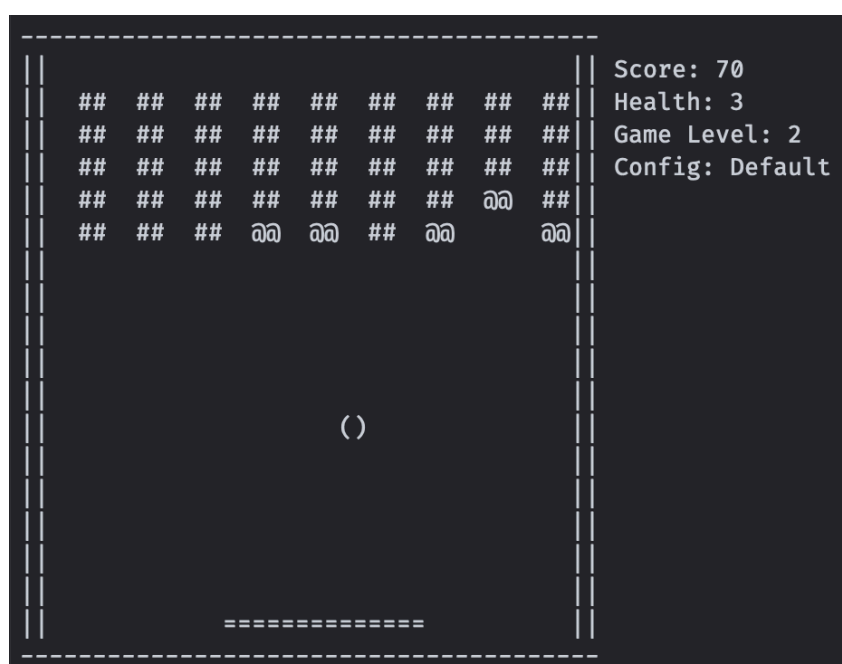


图 2: 一个基于命令行进行游戏界面绘制的程序案例

一个基于命令行进行游戏界面绘制的程序如图 2 所示。值得注意的是，在本文档提供的示例命令行版本中，球、砖块、墙壁均占两个字符宽度，因此地图字符宽度应该为理论地图宽度的两倍。当然，你们的程序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 球与挡板的移动与控制

玩家可以通过键盘控制挡板水平移动（如 a/d），使用空格键发球或再发球，p 暂停，r 重新开始本关。挡板不可越界。发球前，球可与挡板保持贴合并随之水平移动。

2.4 碰撞与反弹规则

球与边界和砖块发生碰撞时需要进行反弹计算。基本规则为：命中左右墙反转水平速度，命中上边界反转垂直速度；命中砖块则先结算砖块伤害/消除与得分，再按碰撞面翻转对应速度分量。

为了增加游戏趣味性，球与挡板的反弹需体现命中点的影响：设命中点相对挡板中心的归一化水平位移为 $dx \in [-1, 1]$ ，则出射角 $\theta = \theta_0 + k \cdot dx$ （ θ_0 为竖直向上角度， k 控制最大偏转）**出射角函数可以自定义，符合逻辑与直觉即可**。保持球速模长不变。若球触底（越过挡板以下边界），判定为掉球并扣减一条生命。

2.5 砖块与计分

普通砖被击中即消除并获得基础分；耐久砖需要多次命中才会消除；不可摧毁砖仅用于改变反弹路径且不计分。可按需要设计连击或时间奖励。计分应在界面中实时显示。

2.6 关卡与难度

每一关载入不同的砖块布局，通关条件为清除本关可消除砖块。为提升难度，可随关卡提升逐步增加球速、缩短挡板宽度，或引入更多耐久砖/不可摧毁砖。通关后展示过关信息并进入下一关或返回主界面。

2.7 游戏结束

默认生命值为 3。球触底扣减生命；生命为 0 时判定游戏结束，展示本局成绩并返回主界面。

3 其他周边功能

在完成上述功能后，你们的**打砖块**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，希望你们加入打砖块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：

- **基础球速**：一个 1 到 10 的正整数，表示球运动的基础速度。每次位置更新的时间间隔与速度成反比。即，假设速度为 s ，则每隔 $t = \frac{1}{s}$ 秒球的位置进行一次更新。

- **随机种子**：一个整数，用于生成关卡砖块布局（以及球初始方向等）的随机序列。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。
- **初始关卡**：一个正整数，表示游戏开始直接从某个关卡开始。关卡越高，球速越快、布局越复杂，难度越大。

在主界面中，用户可以输入 `i` 新建自定义配置。然后，程序要求用户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`（其中 `XXX` 是一个用户可自行指定的字符串），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
```

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。

3.2 残局挑战功能

3.2.1 手动残局录入

为了提升游戏的可玩性和挑战性，打砖块的残局挑战也是一项有趣的功能。你们需要在命令行中提供一种方式，帮助用户创建和保存自定义的残局，并正确实现残局的载入和游戏。

残局信息至少包含：

- **地图大小**：两个 8 到 20 的正整数，分别表示横向和纵向的格子数，定义了游戏区域的大小（不含左右与底部墙体）。
- **初始砖块布局**：用户可以在地图上预先放置不同类型的砖块，形成特定的初始局面，增加游戏的挑战性（如普通砖、耐久砖、不可摧毁砖等）。
- **初始关卡**：一个正整数，表示游戏从某个关卡开始。关卡越高，球速越快，难度越大。

在主界面中，用户可以输入 `n` 进行残局的创建。然后，程序要求用户输入残局文件的名称。名称只能由英文字母、数字和下划线组成，如果用户输入的是单独的 `q`，则立即结束残局创建的过程。

输入合法名称后，程序要求用户输入地图的大小和初始关卡数。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空白地图。然后，用户可以通过输入指令在地图上放置砖块。合法的指令包括以下几类：

- P X Y Type: 在位置 (X,Y) 处放置一个砖块, *Type* 为砖块的符号表示 (例如耐久砖用 #, 普通砖用 @, 不可摧毁砖用 *)。X 和 Y 为正整数, 表示地图中的行和列。
- D X Y: 删除位置 (X,Y) 处的砖块。
- f: 表示结束残局创建, 并保存残局文件。
- q: 表示直接退出残局创建, 返回主界面。

用户每输入一条指令后, 程序需要验证其合法性, 并更新地图的显示。例如放置的位置超出了地图边界或者与已有砖块重叠, 程序需要提示错误并要求重新输入。你需要合理分析其他边界情况, 妥善处理用户的非法输入或不合理的残局地图。

完成残局编辑后, 程序随即将残局文件保存下来, 文件名为 XXX.end, 保存在游戏根目录下的 endgames 文件夹中。残局文件的格式可以参考以下示例:

```
1 // In endgames/example.end:
2 9 18 // 地图大小: 横向9(不含墙面), 纵向18(不含墙面)
3 3 // 初始关卡数
4 P 3 4 #
5 P 3 5 #
6 P 3 6 @
7 P 4 4 #
8 P 4 5 *
9 f
```

在主界面中, 用户可以输入 m 进行残局载入。然后, 程序需要告知当前的残局文件名称, 并要求用户输入新的残局文件名称。如果存在对应的残局文件, 应当载入并应用于此后的游戏中; 如果不存在残局文件, 则需要提示用户并要求再次输入。用户可以通过输入 q 退出残局载入过程。

游戏每次启动时, 应当载入默认的空残局 (即无初始砖块的地图), 初始关卡为 1。开始游戏后, 程序会根据载入的残局文件, 在地图上预先放置指定的砖块, 并从指定的关卡开始, 玩家需要在此基础上继续游戏。游戏的其他规则保持不变 (例如球触底扣生命、清空可摧毁砖即通关等)。

3.2.2 暂停残局生成

除了手动残局录入, 在游戏过程中, 玩家可能会遇到一些精彩的局面, 或者希望将当前的游戏状态直接保存下来供以后挑战。为此, 你们需要在游戏中加入**暂停-残局生成**的功能, 使得玩家可以在游戏进行中随时暂停并将当前的游戏状态保存为一个残局文件。通过实现暂停残局生成功能, 玩家可以随时保存当前的游戏状态, 日后重新挑战或与他人分享, 提升游戏的可玩性和趣味性。

暂停残局生成的保存文件应与手动残局录入的格式兼容; 如需记录挡板与球的即时状态, 可在文件末尾追加可选字段 (例如挡板中心位置、球的位置与速度等), 并在载入时进行兼容解析。

4 其他扩展功能

做到这里，恭喜你们已经成功搭建出了一个十分完整的打砖块游戏，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，并将精力用于其他课程的学习。

4.1 图形界面绘制

大作业的基础要求为命令行界面。然而，图形化是当代游戏必不可少的一步。你们可以尝试将打砖块的游戏界面转换为图形用户界面 (Graphic User Interface, GUI)，这将极大提升游戏的用户体验。

如果你们决定实现图形界面，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**所有功能** (包括基础游戏环境和其他周边功能)，如果缺少功能，则会扣除相应分数。
- 如果你们完整实现了 GUI 界面，则无需完成命令行的界面编写，且不会损失分数。

提示：你们可以使用 Qt 作为 C++ 的 GUI 编程框架。Qt 相对比较简单易用，但需要一定的面向对象编程知识。你们也可以选择其他 C++ 的 GUI 框架，例如更加快捷轻量化的 EasyX 开发框架，这需要你具备一定的自学能力。

4.2 更多扩展功能

如果你们仍有余力，你们也可以创新并实现本文没有提到的其他扩展功能。扩展功能的准则是**合理、切题**，例如“游戏记录回放功能”、“连续命中分数成倍增加”、“随机掉落道具（分裂球、穿透球）”是一些好的扩展功能，而“增加了贪吃蛇等其他小游戏”不是一个好的扩展功能。

如果你们成功实现了这些扩展功能，请在大作业报告中详细说明这些功能的意义、具体实现和相应的技术细节。我们将根据功能的创造性、工作量和完成度给予额外的评分。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！**

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（个别小组允许三人），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三人一组的提交。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

项目的截止时间为 2026 年 1 月 4 日 23:59 (十六周周日)。

请在网络学堂上提交一个 ZIP 压缩包。文件名为【**组员 1 学号_组员 1 姓名_组员 2 学号_组员 2 姓名.zip**】(三人组为【**组员 1 学号_组员 1 姓名_组员 2 学号_组员 2 姓名_组员 3 学号_组员 3 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 60MB²，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云盘链接(谨慎使用)。如你选择这种方式提交，请不要在截止日期后更改清华云盘链接指向的内容，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。
 - 基础游戏功能(40%)：包括自定义配置和残局挑战功能。
- 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³，组员对小组做出了足够的贡献。
- 扩展功能分数
 - 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，上限为大作业满分，不可补充其他教学环节的分数。

5.4 提前提交与补交政策

我们鼓励高效、高质量的课程设计提交，惩罚超时的课程设计提交。本课程作业提交政策如下：

²超过 60MB 请使用视频压缩软件等降低视频分辨率

³如在合作中遇到困难，请及时与助教进行沟通，具体评分课程团队会根据实际情况一事一议。

- **提前提交（早于第 16 周）**：按整周计算（周内任意一天均按该周的整周处理）。每提前一周，得分系数增加 1%。即：
 - 在第 15 周提交 → 101%；
 - 在第 14 周提交 → 102%；
 - 以此类推。
- **延迟提交（第 17–18 周）**：在基准时间之后，每晚一周，得分系数降低 5%。
 - 第 17 周提交 → 95%；
 - 第 18 周提交 → 90%。
- **重度延迟提交（晚于第 18 周）**：在第 18 周的基础上，每晚一天，得分系数再降低 5%。即：
 - 第 19 周第 1 天 → $90\% - 1 \times 5\% = 85\%$ ；
 - 第 19 周第 2 天 → 80%；
 - 以此类推

请在保证质量的前提下选择提前提交，**提前提交后不支持任何提交材料的修改。**

6 学术规范提示

本次大作业具有一定的难度和代码量，我们鼓励同学们之间相互学习和讨论。但请务必遵守学术规范，即：可以参考其他同学的思路或代码，但不要直接复制粘贴！如果你在大作业中参考了其他同学的工作，请在大作业报告的末尾进行恰当的引用声明。若你的代码曾被其他同学参考，也请在大作业报告末尾声明，注明参考了你的代码的同学（包括可能通过间接渠道传播到的其他同学）。

我们支持使用 GitHub 等代码托管平台以便更好地管理项目和协作，但请确保不要将代码托管至任何公开的仓库。若使用私有仓库，也请确保不对其他同学开放访问权限。

我们将对所有提交的代码进行查重。如果发现代码雷同的情况，所有涉及的组（无论是复制者还是被复制者）均将受到相同的处罚，除非相关组别已对雷同部分作出了声明。对于情节严重的违规行为，教学团队将上报至院系和学校，并按照相关规定进行严肃处理。

7 写在最后

本次大作业旨在知识性与趣味性之间找到一个良好的平衡，既帮助大家在编程知识方面有所收获，又激发大家探索与创造的能力。如果大家对大作业的安排或内容有任何建议或想法，欢迎随时向助教团队反馈，我们将不断努力改进课程内容，提升教学质量，力求为大家提供更好的学习体验。希望你们在完成这次课程设计的过程中，能够收获知识、激发灵感，享受编程开发的乐趣。