

程设大作业

概述

小明在学习了《程序设计基础》后，想要设计一款游戏。这款游戏以小明未来的房间为背景，**房间**中有许多积木和两个传送带，如图1所示。每块**积木**上有一个数字。一个传送带（被称为**输入传送带**）会传送进一串积木，另一个传送带（被称为**输出传送带**）会将积木传送走。房间里有一些带序号的**空地**（**序号从0开始**），每个空地可以储存一个积木。

一个**机器人**在房间中运行，机器人根据用户输入的**机器人程序**来运行。机器人程序由一串**指令**组成（**序号从1开始**）。机器人会按照机器人程序中的指令，从前往后依次执行（在没有跳转指令的情况下）。机器人每次只能搬运一个积木，机器人手中的积木被称为**当前积木**。初始情况下，机器人手中没有当前积木。为了保证机器人能正常运行，小明需要模拟机器人的运行结果。

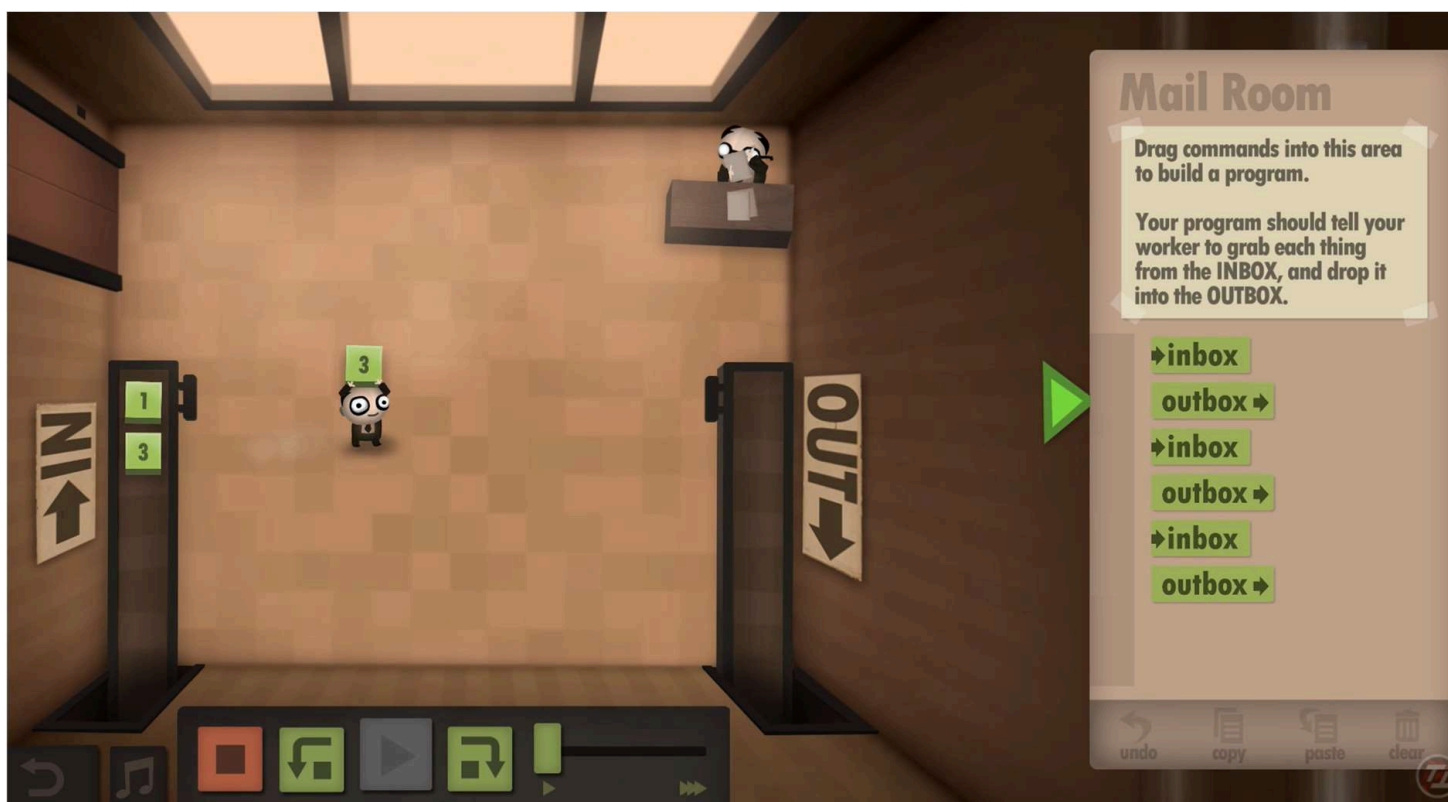


图1 小明设想的游戏场景示意图

功能

基础功能

1. 选择关卡

用户打开游戏后，首先进入**选择关卡页面**（使用命令行制作界面即可，不要求使用图形界面）。各个关卡应显示是否已经通过，且重启程序后仍保留之前的过关情况。关卡间有先后关系，只有通过前置关卡后才能进入新的关卡。

- 作为游戏，界面应有必要的提示来提醒玩家进行关卡选择，并能够处理非法的用户输入。
- 用户完成一局游戏后，返回到选择关卡页面。

2. 游戏界面

(1) 关卡信息：进入某一关卡后，**显示游戏的各项说明和设置**，例如输入队列的积木数和积木内容、空地数、可用指令集、输出序列的要求（即需要完成的任务）。

(2) 读取机器人程序：游戏需要支持从键盘或文件读取用户输入的机器人程序，模拟机器人的运行，并按照格式要求将结果输出到屏幕。

a. 键盘输入：输入方式合理即可，不做统一要求。一种可行的指令输入方式是，先要求用户输入程序的指令数量 M ，在读取 M 条指令后开始游戏运行。

b. 文件输入：询问用户机器人程序文件的路径。输入方式合理即可，不做统一要求。一种可行的指令输入方式是，输入文件由 $M+1$ 行组成，第一行为指令数量 M 。第2至 $M+1$ 行，每行包含一条指令。

(3) 关卡运行：游戏需要**逐步模拟**机器人的运行，并在命令行输出每次运行后的状态。如图2为一个输出样例。

- 游戏界面应**清晰美观地显示出游戏的状态**，建议实现与上图类似的界面。输出形式不严格要求相同，但须包含上图红字标注出的各个游戏元素。

(4) 关卡结束：当满足下述两条件之一，关卡结束，并判断输出是否符合要求

(a) 指令执行完成

(b) 执行inbox指令时，输入传送带上没有积木

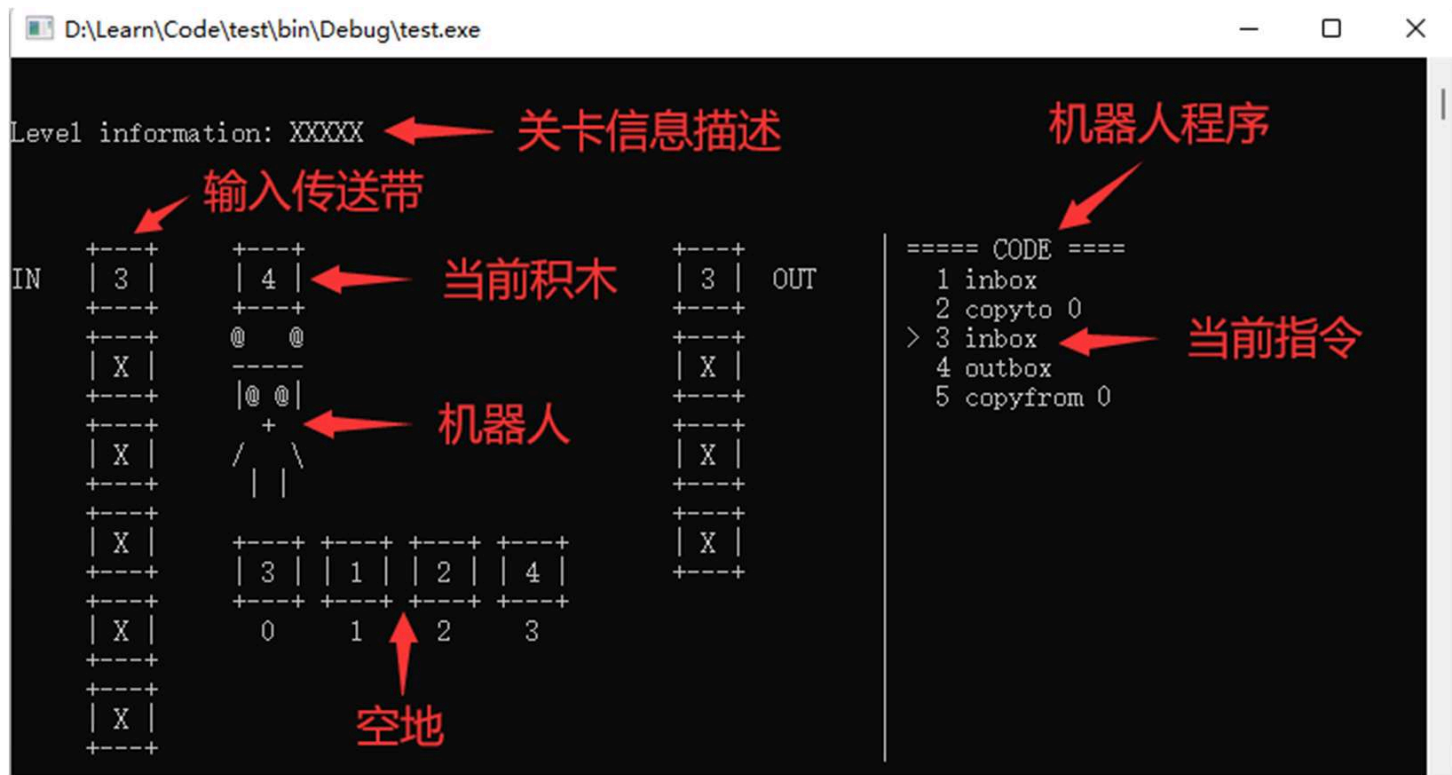


图2 关卡运行输出样例

3. 机器人的指令集

注：带操作数的指令，格式为指令与操作数用空格分开，如 `add 0`。

指令名 操作数（均为整数） 解释 错误情况

指令名	操作数 (均为整数)	解释	错误情况
inbox		机器人拿起输入传送带的第一个积木（成为当前积木）。原有当前积木被舍弃。当输入传送带上没有积木时，游戏结束。	
outbox		将当前积木摆在输出传送带上（机器人不再拥有当前积木）。	1. 没有当前积木
add	X	将当前机器人手中积木的数字加上 X 号空地上的积木。X 号空地上的积木保持不变，机器人手中积木的数字替换成加完的结果。	1. 没有当前积木 2. X 号空地不存在或空地中没有积木

指令名	操作数 (均为整数)	解释	错误情况
sub	X	将当前积木的数字减去 X 号空地上的积木。X 号空地上的积木保持不变，机器人手中积木的数字替换成减完的结果。	1. 没有当前积木 2. X 号空地不存在或空地中没有积木
copyto	X	复制当前积木到 X 号空地。若 X 空地上有积木则舍弃原有积木。	1. 没有当前积木 2. X 号空地不存在
copyfrom	X	复制 X 号空地上的积木到当前积木，即把机器人手中积木的数字替换成 X 号空地上的积木的数字。若存在当前积木则舍弃原有当前积木。	1. X 号空地不存在或其中没有积木
jump	X	改变机器人接下来执行的指令，跳转至第X条指令并从该指令开始继续向后依次执行每条指令（这里的第 X 条是指原始输入程序中的第 X 条，不会受到程序执行的影响）。	1. 不存在第 X 条指令
jumpifzero	X	如果当前积木为 0，则改变机器人接下来执行的指令，跳转至第X条指令并从该指令开始继续向后依次执行每条指令；如果当前积木不为 0，则不做任何操作。	1. 不存在第 X 条指令 2. 没有当前积木

4. 异常情况处理

异常情况包括不在指令表中的未定义指令，不属于当前关卡固定的可用指令集，不符合指令表规定的指令使用（如操作数非整数、指令特定的错误情况、指令后面的操作数数量与要求不符）。在执行一条指令时，如果遇到了异常情况，则输出“Error on instruction X”（其中X[编号从1开始]表示指令位于机器人原始输入程序中的第X条。比如假设第一条被跳转执行过10次，它依旧是第一条指令），并结束当前游戏。对于未运行到的指令不做异常情况判断。

5. 正确性检测

如果输出符合游戏关卡要求，则输出“Success”，否则输出“Fail”。输出这次游戏的统计信息，如总共执行的指令数。

6. OJ自动化测试

为了对机器人运行模拟部分的正确性和鲁棒性进行检测，该部分代码需要提交至OJ进行测试。提交至OJ的代码，需要首先读取一个整数，代表关卡编号（与“关卡任务”中的关卡编号一致，仅测试非自定义关卡），然后按照上文“文件输入”所述的格式，读取用户输入的机器人程序。详细信息请参考OJ说明。

- 提示：为了方便将机器人模拟部分提交至OJ，请保持合理的工程结构，能方便地将这一模块以**单个文件**的形式提交至OJ。
- OJ测试的程序**输出仅有一行**，为“Success”、“Fail”或“Error on instruction X”（其中X表示在第X条指令出错）三者之一。

关卡任务

需要实现的关卡及输入输出序列如下所示。

1. 让机器人取出输入序列上的每个积木放入输出序列中
 - a. 关卡的输入输出
 - i. 输入序列：1, 2
 - ii. 目标输出序列：1, 2
 - b. 可用空地数：0
 - c. 可用指令集：inbox, outbox
2. 对于输入序列中的每两个东西，先把第1个减去第2个，并把结果放在输出序列中，然后把第2个减去第1个，再把结果放在输出序列中，重复。
 - a. 关卡的输入输出
 - i. 输入序列：3, 9, 5, 1, -2, -2, 9, -9
 - ii. 目标输出序列：-6, 6, 4, -4, 0, 0, 18, -18
 - b. 可用空地数：3
 - c. 可用指令集：inbox, outbox, copyfrom, copyto, add, sub, jump, jumpifzero
3. 从输入序列中依次取2个数字，如果相等则将其中一个输出，否则扔掉。重复这个过程直到输入传送带为空。
 - a. 关卡的输入输出
 - i. 输入序列：6, 2, 7, 7, -9, 3, -3, -3
 - ii. 目标输出序列：7, -3
 - b. 可用空地数：3
 - c. 可用指令集：inbox, outbox, copyfrom, copyto, add, sub, jump, jumpifzero
4. 一个自由创新关卡

扩展功能

扩展功能不是必须实现的部分，有较大的难度和灵活性，请同学们合理安排时间。

1. 关卡配置：在不修改游戏源程序的前提下，通过关卡数据文件来自定义增加关卡。
2. 图形界面：支持单步模拟、连续模拟。
 - a. 图形界面可以基于其他框架，但编程语言必须用C++。
 - b. 完成GUI可以不做CLI，功能上要求一致。
 - c. OJ自动测试不支持GUI框架，要保证向OJ上传的代码能通过测试。

评分标准

提交格式要求

提交文件应满足如下格式：

```
第X组_组长姓名+组员姓名_程设大作业.zip
├─ report.pdf（报告文件）
├─ video
│   └─ demo1.mp4（录屏文件，尽量为mp4格式）
│       └─ ...（该文件夹下可包含多个录屏文件，也可只有一个）
├─ code
│   └─ README.txt（编译和运行说明，为可选文件，txt、md或pdf格式均可，若在报告中已有说明则可不提供）
│   └─ main.cpp（提交至OJ的代码文件，包含main函数）
│   └─ src（完整程序源代码文件夹）
│       └─ ...（源代码文件）
│       └─ ...（该文件夹下可包含多个源代码文件夹代表不同版本/模块，但需在README中说明）
└─ ...（根目录下可包含其他用于补充说明的文件或文件夹）
```

需注意：最终提交的压缩包大小（包括视频）应不超过128M。

基础成绩

- 50%的成绩是自动化测试的结果。
 - 你可以在OJ，找到【大作业检验】题，检查程序正确性。注意，当前检验仅为小数据，通过该检验，**并不说明当前程序完全正确！**
 - 在大作业截止日期后，会统一进行更大数据量的测试。**OJ评测以各小组组长的最后一次提交代码为准。**
 - 可以使用以下方法进一步检验程序正确性
 - 人工检验：采用不同测例，手动验证游戏的正确性和鲁棒性。
 - 自动检验：自动生成数据进行上述测试。

- 对比检验：参考Human Resource Machine等类似游戏的关卡，或其他小组的关卡，设计相同的机器人程序，并对比输出结果。
- 50%的成绩是报告和手动测试，请给出项目的**OJ提交编号、设计思路、工程结构、整体游戏界面的设计、游戏测试、一个自由创新关卡和小组分工**等信息（**这些信息均要出现在报告中！**）。
 - 请给出详细的编译、运行说明，方便助教手动测试。
 - 程序要求进行录屏演示所有功能和关卡并进行介绍（在视频中以文字或声音的形式均可）。
 - 最终压缩文件由组长提交网络学堂，一人提交即可。
- 扩展功能：对于扩展功能实现效果突出的作业，将获得课程组颁发的优秀大作业奖状。

得分系数

- 若提交压缩包命名格式/文件组成不符合要求，得分系数将会酌情降低。
- 得分系数与提交时间相关，具体规则如下，可从图3中直观理解：
 - **基准提交时间：第16周周末**，得分系数为 100%。
 - **提前提交**：按整周计算（周内任意一天均按该周的整周处理）。若早于第16周，**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%；
 - **最晚提交时间：第20周周末**，逾期不候，得分系数为 0%。

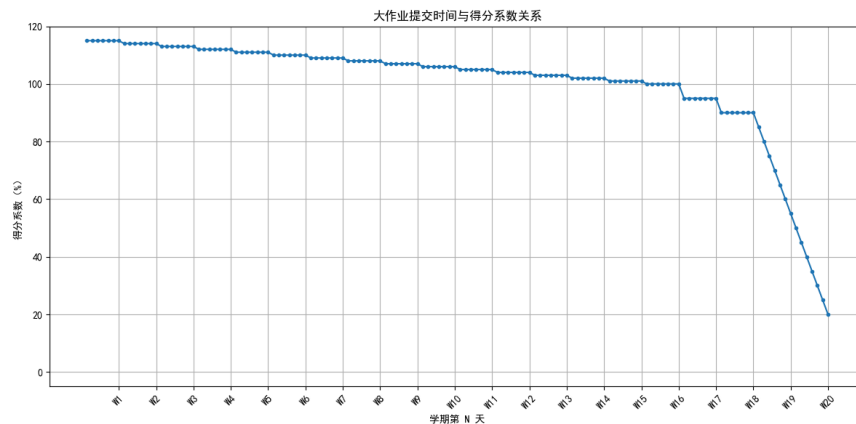


图3 得分系数与提交时间的关系

- 最终成绩计算方式（按照百分制计算）：
 - **最终成绩 = min(基础成绩 × 得分系数, 100)**

提交方式

大作业需**组长**提交，提交内容包括以下三个部分：

- **OJ**：代码文件（main.cpp）提交至OJ进行自动化测试。
- **网络学堂**：压缩包提交至网络学堂。
- **提交时间问卷**：在**确保**前两项均已提交的前提下，填写提交时间问卷，用以确定小组的得分系数。
问卷链接：<https://wj.qq.com/s2/23827675/2788/>。
 - 提交问卷后助教随时可能批阅大作业，批阅后成绩将确定并无法更改，请**慎重提交**；若在网络学堂中未发现提交文件，则视为未提交，得0分。
 - 每组**仅有一次**填写机会，请务必在提交压缩包并确认最终版本后再填写问卷。
 - 批阅时会问卷中填写的大作业提交时间（并非问卷自身的提交时间）与网络学堂中的提交时间进行核对，如发现不符，将以较晚的提交时间为准。

补交政策

- 假设a日23:59是在网络学堂上的截止时间，那么补交**必须向一位助教发邮件告知**，并尽可能通过微信等方式联系助教**确认收到邮件**。
- **邮件主题**请写为“程设大作业补交+组长姓名+组长学号+组员姓名+组员学号+补交时间”，如“程设大作业补交+张三+1234567890+李四+0987654321+1月20日”。
- 邮件正文请写明补交的原因和补交的时间。邮件请发送至任意一位助教的邮箱。助教邮箱查询方式为：网络学堂 → 程序设计基础(30240233-1) → 课程邮件 → 教师通讯录
- 补交的代码和报告请作为邮件附件发送，**附件命名格式与正式提交时相同**。若附件过大无法发送，请使用网盘等方式，并在邮件中附上下载链接。
- 请务必确保助教能下载到附件。补交成功后，助教会通过邮件或微信等方式回复确认收到。若未收到回复，请通过微信等方式联系助教确认收到邮件。
- **补交时间**是助教收到邮件的时间。

学术规范

由于作业有一定难度，同学之间相互学习和指导是提倡的。对于其他同学的代码，可以参考，但禁止直接拷贝。如有代码交给其他同学参考，必须在报告中声明，告知给哪些同学拷贝过代码（包括可能通过间接渠道传播给其他同学）。请所有同学不要将自己的代码托管至任何公开的仓库上，托管至私有仓库的请不要给其他同学任何访问权限。

我们将会**对所有同学的代码作相似度检查**，如发现有代码雷同的情形，拷贝者和被拷贝者将会得到同样的处罚，除非被拷贝的同学提交时已做过声明。代码雷同情节严重的，课程组有权上报至院系和学校，并按照相关规定严肃处理。

参考资料

- Human resource machine：一款手机游戏（似乎也有电脑版），可以作为完成本作业时的参考。本作业文档与游戏不一致之处，以本作业文档为准。
 - 游戏通关视频参考链接：<https://www.bilibili.com/video/BV1HW411N7XK>