

本文档版本。1.1。

1 工作的组织

适用场合。以下文字只是关于本课程期末考试的“预案”，只适用于以下情形：由于疫情，学生无法返校参加闭卷考试。如果能较早预知到学生能够返校并参加传统的闭卷考试，将不采用本预案，而是采用传统的闭卷考试。解释：本预案的方式不如传统的闭卷考试公平，无法精确地评估每个学生的真实水平。

1. 最迟于 5 月 25 日公布一个大作业题目。自题目公布之日起，此项工作就会被启动。期末考试的形式就固定为这种形式，即使在题目公布之后学生们能够返校。解释：6 月 15 日-28 日为考试周，将提交截止日期设置为 6 月 28 日，留给大家 1 个月的时间，再加上组队时间，也就是最迟 5 月 25 日前公布。一旦公布，学生们就已经开始工作，如果再切换到传统的闭卷考试，会浪费大家的精力。

2. 每 4-8 个学生组成一个小组。形成小组的原则是各个小组成员的总体编程能力相当。由课程助教（而不是班长）负责完成分组。分成两个阶段：

A. 自由组队，但是要求一个小组中所有成员的在课程“高级语言程序设计 2-1（也就是 C 课程）”的平均分应该大约等于该课程所有学生的平均分。2019 年级的是 80.6 分，各小组的平均分应该和这个平均分相差一般不超过 5 分。解释：为了使得各个小组的总体能力大致相当，避免排名靠后的同学不被任何一个组接纳的情形。

B. 对于没有纳入任何一个小组的同学（如有），由助教负责组队。必要时，可以插入到其他已经形成的小组中。

3. 每个小组在期末考试周结束之前，提交以下材料给老师：

a. 项目的源程序。不要包括和构建相关的文件，比如 Visual Studio 二进制项目文件或者 CMake 文件。这是由于每个小组所使用的操作系统、集成环境、构建工具都会不同。老师不会尝试构建并运行提交的程序，而只会查看源程序。

b. 一份 4 页 ~8 页的 PDF 格式的报告。报告中的主要内容：

- * 小组成员以及每个成员承担的工作；
- * 使用的操作系统、集成环境、构建工具；
- * 使用了哪些 C++ 语言特征来实现预期的功能；使用了哪些第三方库以及标准库中的特征来完成预期的功能；

- * 系统中有哪些 C++ 类，各自的功能以及如何协作来完成预期的功能的；

- * 对于“俄罗斯方块”项目，显式标注出所能消解的方块的数目，作为该程序性能的指标。

c. 对于“俄罗斯方块”项目，需要录制整个程序从开始运行一直到游戏结束的整个视频，以证明该程序的能力。视频中要明确地显示出当前时间、已经被消解的方块的个数。视频文件会被较大，需要上传到某个网盘，然后由助教下载、观看、验证 PDF 格式报告中游戏成绩的真实性。

4. 评估标准。参“俄罗斯方块”项目部分。

5. 成绩。

* 一般情况下，以组为单位评估，而不是以个人。一个小组的得分就是其中每个成员的得分。如果一个小组中有贡献特别大或者特别小的情形，在报告中指明。将依据具体情形对涉及到的成员予以调整。

* 每个学生该课程的最终成绩是此作业的成绩（占 40%）和上机环节的成绩（占 60%）的加权和。对于少数由于各种原因没有参与上机环节的学生，以此作业的成绩为最终成绩。解释：之所以上机环节的比重大，是由于该环节的执行时间较长，能够更加准确地反映各个学生的能力。

6. 鼓励组内成员通过网络沟通协作，禁止抄袭其他小组的源代码、报告。对于组间抄袭的，分为以下两种情况处理：

a. 被抄袭方有意将本小组的作业结果呈现给抄袭者，相关两个下组的此项工作的成绩都将被记为 0 分。

b. 被抄袭方无意间被抄袭，抄袭者此项工作的成绩被记为 0 分，被抄袭者成绩将被减去此项工作的成绩的 20%。

备注：由于组的数量不多，老师容易通过阅读报告、源代码审查而检测到抄袭。

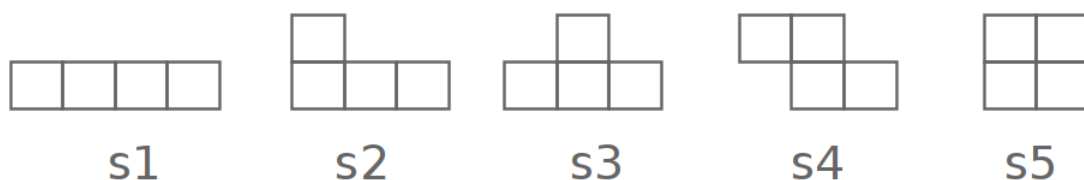
2 俄罗斯方块

2.1 题目描述

该作业要求每个小组开发一个能够玩儿俄罗斯方块游戏的程序，关于该游戏，可以参考 <https://chvin.github.io/react-tetris/>，我们将该网页中的那个游戏程序称为“参考程序”。要求程序给出对每个方块的处理方法，而不是由人类玩家给出。程序的输入为老师给定的方块序列，其中包含哪些时间出现什么方块的信息，要求程序模仿人类玩家，给出对这些方块的处理方法。方块只能在一个方框中移动或者旋转，该方框被称为“游戏区域”。我们约定该游戏区域和上述网页中的一致，为一个 20 行、10 列的矩形方框，其中每个小格子被称为一个“栅格”。当游戏区域的一个行被方块完全填满时，那一行会被“消解”，也就是该行之上所有的被方块占领的栅格会下移一行。程序的得分是所消解的栅格的总数。

以下是该游戏的具体要求。如果以下要求和参考程序中的冲突，以本文档的为准。对于本文档没有提及的部分，以参考程序为准。

方块的形状。如果排除掉旋转后相同的，只有以下几个拓扑上独立的：



玩家所能执行的操作。玩家只能对出现时间最早的那个方块进行操作，可以执行“左移”，“右移”，“下移”，“旋转”，“掉落”的操作。各个操作的含义参上述网页游戏。

游戏难度级别。我们将一个方块从一行自然下落到下一行的时间间隔称为“下降周期”。游戏有三个难度级别：“容易”，“适中”以及“挑战”。各个级别的区别在于下降周期以及新方块的生成频率的不同。越容易的级别，下降周期越长，新方块的生成频率越低。具体地，“容易”级别，下降周期为 500 毫秒，新方块的生成的时间间隔为 5 秒，“适中”级别，下降周期为 300 毫秒，新方块的生成的时间间隔为 3 秒。“挑战”级别，二者分别为 200 毫秒以及 1 秒。

测试集。老师给定的一个测试集包含三个文本文件，对应着上述三个难度级别，分别为 easy.txt, normal.txt 以及 challenge.txt。该文件每一行的内容为：【方块产生时间】【方块编号】，时间的格式为“分钟: 秒”，比如 58:3。方块编号为“s1”到“s5”。

程序性能评估标注。程序分别对上述三个难度级别的文件进行处理，将所能消解的方块的数目进行加和，做为评估该程序智能程度的测度。

2.2 程序的功能

该程序绘制游戏画面，包括能够绘制出方块落下的过程，底部没有被消除的方块的样子。当一行的方块被消除时，应该在画面上有些视觉效果，表明该行的方块被消除。

读入存放方块序列的文件，依据当前时刻以及之前出现的方块的情形，计算对正在下落的、出现的 earliest 的那个方块的处理方法。对于当前时刻新出现的那个方块，应该在游戏区域的最顶那行显示其全部（对于方块 s2）或者部分形状（对于其他方块），而且，在水平方向上该方块应该出现在尽量居中的位置。此处需要说明的是：

1. 正在下落的可能有多个方块，但是我们假定程序只能操作那个出现最早的方块，不能处理它出现之后的方块。

2. 整个程序虽然能够读入很长时间内的方块序列，但是当前时刻只能分析处理该时刻之前的方块的信息，不能预先分析整个序列，也就是不能“偷看”当前时刻之后的方块信息。

游戏画面中应该显示一个“游戏时间”，这个游戏时间的流逝速度应该物理时间的一致。比如，方块序列文件中如果指定在 1 分 20 秒出现某个形状的方块，那在游戏画面中，就应该在这个游戏时间中看到有上述指定形状的方块出现。

游戏画面中还应该显示出当前出现的方块的总个数，这个数将用来评价该程序的智能程度。

除了上述界面元素要求之外，界面中无需显示其他元素。界面是否美观不作为评估标准。

2.3 技术方面的要求

整个项目全部使用 C++ 语言开发，不能有其他语言。

在程序界面部分，可以借鉴其他第三方开源软件，但是需要在报告中引用。

程序智能部分（也就是计算玩家动作的部分）不可以参考任何第三方软件，需要独立完成。

2.4 测试

方块的出现是完全随机的，也就是说，每当游戏中需要新增一个方块时，形状 s1 到 s5 的方块各自的出现概率为 1/5。

在开发阶段每个小组会收到若干测试用的方块序列文件，用于开发、测试程序。在最终评估各个小组的程序性能时，会在约定的时间公布一个新的测试序列。要求所有小组在规定的时间内（比如 2 小时）内，完成程序运行、屏幕录像、上传到网盘、通知助教的操作。

2.5 评估标准

设整个课程大作业的分数为 100 分，以下是评估子项以及各项的总分。

1. 10 分。小组内成员的分工是否合理，团队是否能协同工作。解释：此项工作的目的不但是考察学生的技术能力，也锻炼大家的团队协作能力。要求每个小组应该合理分工，一方面避免能力强的人承担太多的工作，另一方面避免有的成员不做工作而蹭分。为此，小组分工也将作为一个检查指标。在每个小组的报告中，需要详细列出各个成员的工作。老师会使用在线通信工具抽查部分小组，以核查这个分工是否属实。
2. 40 分。对标准库的相关模块以及第三方库的选择与使用是否合理。对 C++ 语言特征是否能够正确、适当的使用。代码格式、用法是否规范。
3. 40 分。程序的功能与智能。具体地，能够显示方块的移动、旋转，游戏区域方块的堆积，一个完整行的消除，可得 20 分；表现出初步智能，加 10 分，表示出较高智能，加 20 分。
4. 10 分。文档的撰写是否规范。

2.6 相关知识参考

本节内容仅供各小组参考，各小组可以完全不使用本节提及的技术。

界面部分。Qt (<https://www.qt.io/>) 以及 wxWidget (<https://www.wxwidgets.org/>)。

智能部分。alpha-beta pruning (https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning)。