

上海交通大学软件学院本科二年级

2011 年《暑期软件开发技术大型课程设计》题目

目录

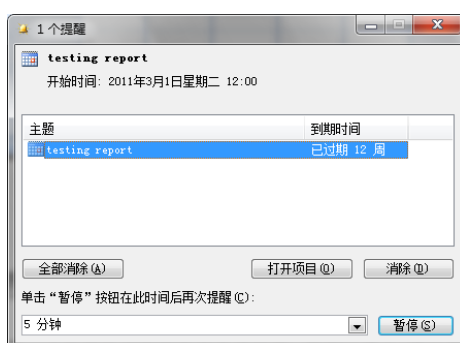
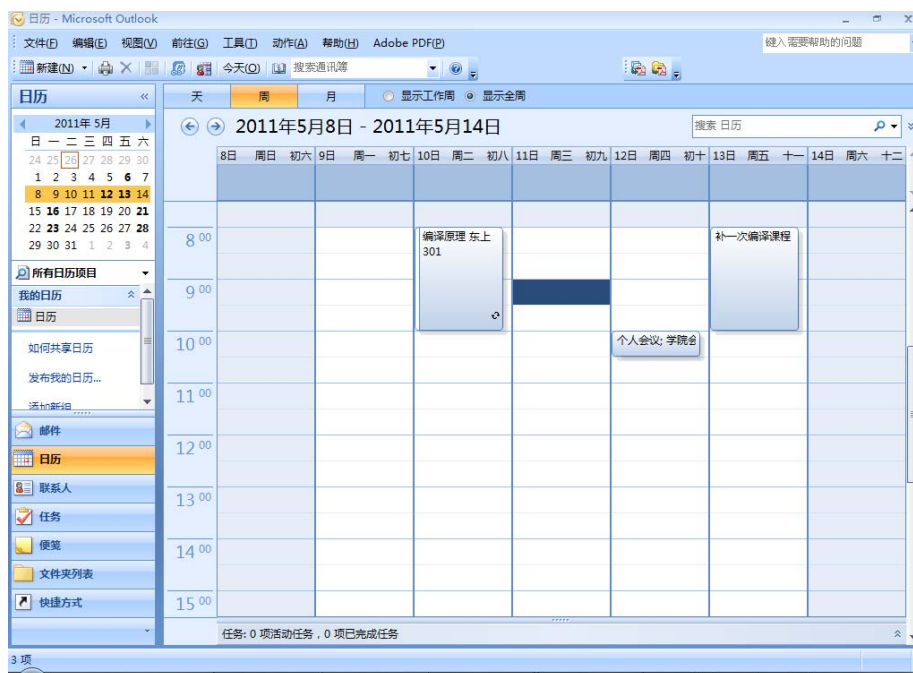
上海交通大学软件学院本科二年级	1
2011 年《暑期软件开发技术大型课程设计》题目	1
1. 软件学院电子日历（应用题）	3
基本功能（难度值=3）:	4
进阶功能（难度值=2）:	4
2. 铁路换乘查询系统（应用题）	6
基本要求（难度值=3）:	6
进阶要求（难度值=2）:	6
3. 考试排位系统（应用题）	8
基本要求（难度值=2）:	8
进阶要求（难度值=2）:	8
4. 医院排班系统（应用题）	9
基本要求:（难度值=1）	9
进阶要求:（难度值=2）	10
5. 婚恋信息聚合（应用题）	11
基本要求（难度值=3）	11
进阶要求（难度值=2）:	11
6. 物流装箱（研究与科学计算题）	12
基本要求（难度值=2）:	12
进阶要求（难度值=2）:	12
7. 简单的程序调试器（研究及科学计算题）	13
基本要求（难度值=3）:	13
进阶功能（难度值=2）:	13
8. 球队列的概率计算（研究及科学计算题）	14
基本要求（难度值=1）:	14
进阶要求（难度值=2）:	14
9. 社会网络图（研究及科学计算题）	15
基本要求（难度值=2）:	15
进阶要求（难度值=2）:	16
10. 愤怒的小鸟（游戏题）	17
基本要求（难度值=3）:	17

	进阶要求（难度值=2）：	17
11.	蚂蚁找食物的动画模拟（游戏题）	19
	基本要求（难度值=3）：	20
	进阶要求（难度值=2）：	20
12.	太阳系系统（游戏题）	21
	基本要求（难度值=3）	22
	进阶要求（难度值=2）	22
13.	抓老鼠 (Catcha Mouse)（游戏题）	23
	基本要求（难度值=2）：	23
	进阶要求（难度值=2）：	23
14.	自设题目	24

1. 软件学院电子日历（应用题）

很多人用过微软的 Outlook 软件，其中有一个日历功能。Microsoft Office Outlook 2007 日历是 Office Outlook 2007 的日历和计划组件，并与电子邮件、联系人和其他功能完全集成。使用日历可以：

- 创建约会和事件
 - 就像在日记本中书写一样，您可以单击 Outlook 日历中的任何时间段并开始键入。新增的渐变颜色更方便了快速查看当前日期和时间。当前时间仅在“日”和“工作周”视图中以彩色突出显示。可以选择使用声音或消息提醒您的约会、会议和事件，并且可以给项目上颜色以便快速识别。
- 查看小组日程
 - 您可以创建这样的日历，其中同时显示一组人员或资源的日程。例如，您可以查看本部门中的所有人员或本建筑中的所有资源（如会议室）的日程。这样有助于快速安排会议。
- 并排查看日历
 - 可以并排查看自己创建的多个日历以及由其他 Outlook 用户共享的日历。例如，可以为个人约会创建单独的日历，然后并排同时查看工作和个人日历。
 - 还可以在所显示的日历间复制或移动约会。使用导航窗格可快速共享自己的日历和打开其他共享日历。根据日历所有者授予的权限，您可以创建或修改共享日历上的约会。
- 以重叠视图中在一个日历之上查看另一个日历
 - 可以使用重叠视图显示自己创建的多个日历以及由其他 Outlook 用户共享的日历。例如，可以为个人约会创建单独的日历，并重叠工作和个人日历以快速查看有冲突或有空闲时间的地方。
- 其他功能
 - 略



基本功能（难度值=3）：

- 本项目中，你需要实现一个实用化的软院版的电子日历，其可以导入/导出学院课表（一般为 Excel 方式，需要自己定义合理的数据结构）、设置年级课程，并进行定时提醒（参考 Outlook 的日历的定时提醒功能）。
- 设置课程表还需要考虑假期及调课等情况，部分课程需要考虑单双周。
- 完全复制 Outlook 功能过于庞大，您的工具需要包括一些核心功能，请自己定义这些核心功能，并与老师或者助教讨论获得。

进阶功能（难度值=2）：

- 实现多种查看视图，如教务老师视图、单个教师视图、学生视图。
 - 针对教务老师，其可以选择不同年级的课程，并且查看相关课表
 - 针对单个教师，其可以选择单个教师需要教的课程，并且查看相关课表
 - 针对单个学生，其除了可以查看公选课，还可以定制个人的课表

- 针对不同用户视图，均可以增加个性化的事件，并且完成一个学期课程情况的汇总
- 针对不同的群体，你可以实现不同的提醒功能。
- 针对每一个可能用户，需要提供电子日历的个性化打印功能，即对于教务、教师、学生的课表，可以分别打印其本周/本学期的课程。

2. 铁路换乘查询系统（应用题）

从上海乘火车要去乌鲁木齐，有很多种选择，有人会选择使用最快的方式（当然，这种方式极有可能是最费钱的），有人会选择换乘最少的方式，有人会选择始发列车，有人还会选择从某日某时刻出发，或者选择至少在某日某时刻到达。因此，本软件的主要目的是实现：给定多种选择模式下，选择最合理的几条路径供使用者选择（在这里，合理的意思是，理论上来说这种选择是无限的，但大部分的选择是不合理的，比如，从上海到乌鲁木齐途径南京，某人可以选择下车，等坐第二天同样的一班列车出发；再如，从上海坐一班火车先到南京，再从南京坐从上海始发到乌鲁木齐的列车，因此，提供的选择中要尽量避免这些情况）。同时，本软件也可以提供简单的查询内容，比如，查询某车次的时间和停靠车站，查询从 A 地到 B 地所有的直达列车等。并且，为换乘两列车之间预留指定的时间（比如，某列车是到达上海虹桥火车站，下一班换乘火车是从上海火车站出发）。

基本要求（难度值=3）：

1. 建立合适的关系数据库，来存储火车车次和车站信息，并且提供从铁路系统信息表到该关系数据库的转换程序。
2. 提供多种换乘选择方式：最快、换乘最少、始发车站、指定出发时刻、指定到达时刻，这些选择方式可以复选。并且提供在该种选择方式下最合理的 3 到 5 条线路。并且根据用户的要求，提供更多的选择。
3. 提供简单的查询，按照车次查询，以及查询直达列车。
4. 提供火车换乘时，更加灵活的时间选择。如：可以让用户输入换乘火车之间的间隔时间。默认为 1 小时。

本项目的详细需求要求向老师及助教进行需求调研。

进阶要求（难度值=2）：

6. 可以显示该站在 google map 上的信息。

（附图是日本雅虎的交通查询系统，其中包括了普通火车、新干线、地铁、甚至还有“徒步”的内容）

経路1

11:06出発

13:11到着

時間:2時間5分(乗車1時間11分、ほか54分)

運賃:片道6,130円(乗車券3,010円 特別料金3,120円)

定期代:1か月138,930円 3か月395,950円

距離:171.6km

乗り換え:5回

この経路をケータイに送る

この経路をメールで送る

カレンダーに追加

印刷

● 京都大学		
11:06~11:22	徒歩16分 地図でルートを表示	
<input type="checkbox"/> 出町柳 出口:7	[駅情報]	地図 宿泊施設 グルメ
11:24~11:27	1駅 京阪鴨東線特急・淀屋橋行	
<input type="radio"/> 三条(京都府)	[駅情報]	地図 宿泊施設 グルメ 260円
11:29~11:35	4駅 京阪本線・淀屋橋行	
<input type="radio"/> 東福寺	[駅情報]	地図 宿泊施設 グルメ
11:37~11:39	1駅 JR奈良線	
<input type="radio"/> 京都 [出口案内]	[駅情報]	地図 宿泊施設 グルメ 2,520円
11:52~12:29	1駅 JR新幹線のぞみ18号	
<input type="radio"/> 名古屋 [出口案内]	[駅情報]	地図 宿泊施設 グルメ
12:39~12:51	4駅 JR中央本線・多治見行	
<input type="radio"/> 大曽根 [出口案内]	[駅情報]	地図 宿泊施設 グルメ 230円
13:00~13:11	6駅 名古屋市営名城線右回り・本山・八事方面	
<input type="checkbox"/> 名古屋大学 [出口案内]	[駅情報]	地図 宿泊施設 グルメ

3. 考试排位系统（应用题）

学校每学期期中和期末都要进行一件非常费时、费力的事情，就是为考试安排座位。要求开发一个考试排位系统，帮助院系的教务老师提高排位的工作效率、准确性和合理性。

考试排位要综合考虑以下因素：

1. 教室信息，包括教室的编号（如东上院 301 等），座位布局。
2. 教务处下达的考试排教室信息，包括课程编号、课程名称、考试时间、教室的编号、考生名单等。
3. 排位要符合随机性。

基本要求（难度值=2）：

- 教室座位布局，对每个教室的座位布局进行方便录入、删除、修改、查询。至少录入东上、东中、东下院教室信息并予以保存。
- 基本信息导入。
 - a) 以 excel 格式导入某个教室的考试学生信息（姓名）。自动排位。对学生座位进行随机排序，生成和打印考试座位表。
 - b) 录入考试时间、地点
 - 支持图形界面方式浏览每门考试排位情况。对于不满意的排位情况，可以再次自动排位或者手工拖拽调整。
 - 座位查询。可按课程、教室、考试时间、考生姓名等方式来查询排位情况。
 - 单机版。

进阶要求（难度值=2）：

- 提供打印功能，将考试座位可以打印于 A3 纸上。
- 要求软件质量高、实用性强，能实际投入使用。

本项目的详细需求要求向雷磊老师及助教进行需求调研。

4. 医院排班系统（应用题）

某医院排班，一周为一周期，固定输入部分班后，剩下的班可以自动随机产生，排班有如下要求：

1. 周日必需一个人值班（轮流，机会均等），其余人都休息；
2. 周一到周六，每天必需一个人值班，必需 2 个人连班（不能多，不能少）；
3. 值班紧跟后面是夜休；
4. 要求上午大于等于 4 人；
5. 下午最好大于等于 4 人但必需至少 3 人，周五下午可以大于等于 3 人；
6. 每周每个人最好有三天休息（即休+休+夜休=3）；

以上的要求 1 的重要性大于 2,在 1 实现的情况下考虑 2,2 实现再考虑 3,依次类推,能达到第几层就第几层。

“班”描述特征：

1. 专/日，上午专家门诊（不在），下午日班（在）；
2. 日/专，上午日班（在），下午专家门诊（不在）；
3. 值班，上午+下午+夜班；
4. 夜休：值班后紧跟第二天休息（全天不在）；
5. 休：全天休息（全天不在）；
6. 日：上午日班，下午日班（全天在）
7. 连：上午日班+中午+下午日班（全天在）

只有 1 和 3，2 和 3 有可能重复，其余各班之间每个人当天均不能重复，例如：如果 1 和 3 重复时，班的特征等于该天有值班（满足条件 1），但上午缺 1 人（可能不满足条件 4）；同理，如果 2 和 3 重复时，班的特征是改天有值班（满足条件 1），但下午缺 1 人（可能不满足条件 5）

设计程序时，值班和夜休是一并产生的。一个排班的例子如下：

	周 1	周 2	周 3	周 4	周 5	周 6	周 7
苏	连	专/日	休	日	休	值	夜休
薛	日/专（值）	夜休	日/专	休	连	日	休
邵	日/专	连	连	值	夜休	休	休
沈	休	日/转	值	夜休	连	连	休
刘	连	连	休	休	日/专	休	值
张	专/日	休	连	连	值	夜休	休
杜	夜休	值	夜休	连	专/日	连	休

基本要求：（难度值=1）

- 用 C++或 Java 实现排班系统；

- 提供友好的 GUI 用户界面，操作简单、直观；
- 如果完全不能满足的话，给予一个最接近的解法，并显示哪些规则无法满足。

进阶要求：（难度值=2）

- 定义规则的输入语法，使得用户可以增加指定新的排班规则。

5. 百科应用（应用题）

在 Android 手机上构建一个百科应用，用户可以选择不同的百科数据源进行搜索。数据源为：百度百科(baike.baidu.com)。一个常见的应用可能包括下列界面：

- 首页面：可参考 UCWEB，一个输入框，数据源选择框，一个搜索按钮。
- 搜索结果列表页面：显示搜索结果列表。
- 详情页面：显示该结果的详情，需要解析数据源返回的 HTML，简化之以适应手机狭窄的屏幕。
- 错误信息 popup：在各种错误（尤其是网络错误）发生时 popup 以提醒用户，给用户一个良好的体验。

基本要求（难度值=3）

1. 建立一个手机应用，使得用户可以进行搜索，具体需求请咨询教师获得

进阶要求（难度值=2）：

2. 增加数据源为 wikipedia(<http://zh.wikipedia.org>)

6. 物流装箱（研究与科学计算题）

在现实生活中，有很多装箱问题。例如搬家的时候，需要将家里物品打包然后放入卡车中。货运公司为客户提供集装箱、货柜、或车皮等标准化的运输工具，客户根据需要来订购并装入货物。在这些问题当中，如何尽可能大的利用装载的空间并同时满足各种约束条件是一个挑战。

基本要求（难度值=2）：

- 假定物品有三种类型：球体、圆柱体、以及纸箱。用户可以自由定义三种物品的规格以及数量。
- 提供两种标准化的运输箱体：集装箱和其它任意一种箱体。并支持用户自定义箱体大小。
- 用 C++ 或 Java 实现装箱问题，计算出物品放在箱体中的位置以及装箱顺序。
- 提供友好的 GUI 用户界面，操作简单、直观，能够三维显示最后结果，允许用户从各个角度和位置观看最后的装箱结果。
- 分析算法的复杂度，对算法结果做出评估。

进阶要求（难度值=2）：

- 物品可以定义更多的属性，包括重量以及包装的最大的载重能力。在装箱的过程中，还必须考虑任意一个物品的负重不能超过它的载重能力。计算在这种条件下如何装箱已经装箱的最后结果。直观显示物体当前的负重。
- 物品在遭到垂直方向的挤压而导致水平方向膨胀，允许客户设置水平方向（长度和宽度方向）的膨胀率，动态地对纸箱实际摆放时的长度和宽度进行膨胀。

7. 简单的程序调试器（研究及科学计算题）

在这个题目中，我们来做一个程序调试器，这个程序调试器应该能被集成到某个程序开发工具（如 Eclipse）中。这个程序调试器以目标程序行号为执行任务，按照行号完成程序执行。

例如，我们有一个 Java 或者 C++ 程序，程序员可以自己输入语句行号（例如：1=>2=>3=>4=>6=>...），使得程序执行按照给定的行号单步/一次性执行，这个新的执行序列命令使得程序执行略过了第 5 句的执行。当然，这个被输入的语句序列可以被组成一个完整的程序。

一个程序员可以指定某一个程序，输入多个执行序列，使程序按照序列多次执行程序的不同语句。

基本要求（难度值=3）：

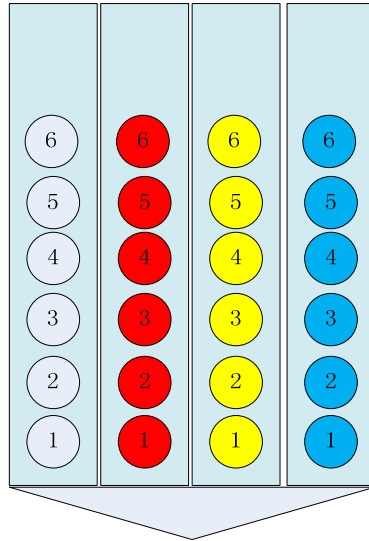
- 根据上面要求，完成一个关于程序指定语句的程序调试器，并能够被集成到某一个开发工具中。
- 增加输入窗口，允许程序员输入语句序列。
- 对于输入的语句行号，允许增加一些合理的规则，如 1=>(3-7)=>9 代表 1=>3=>4=>5=>6=>7=>9。你可以自行设计这些规则。
- 准确的需求需要向教师或者助教咨询后获得。

进阶功能（难度值=2）：

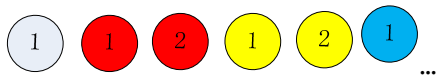
- 对于多线程程序的执行，可以通过本方法达到多线程程序执行的确定性。指定多线程程序的执行语句（及相应的线程），使得可以控制多线程程序确定地执行。注意，这里输入的语句行号格式上需要被改变，例如，可以采用下列表格：

线程	行号
main	1-7
T1	15
T2	20
T1	16-17
T2	21
main	22-24

8. 球队列的概率计算（研究及科学计算题）



上面是一个容器，该容器有若干个管道，每个管道里有一组标了序号的球（1-n 个，每个管道里的球数不定），容器底部有一个喷嘴，每次可以掉下来一个球，那么最后掉下来的球可以排成一个队列。如：



基本要求（难度值=1）：

- 用图形化界面显示如何在各管道里设置若干颜色的球，管道总数不定；
- 计算某一个给定队列的发生概率。需要考虑如何描述这种特定的队列。
- 准确的需求需要向教师或者助教咨询后获得。

进阶要求（难度值=2）：

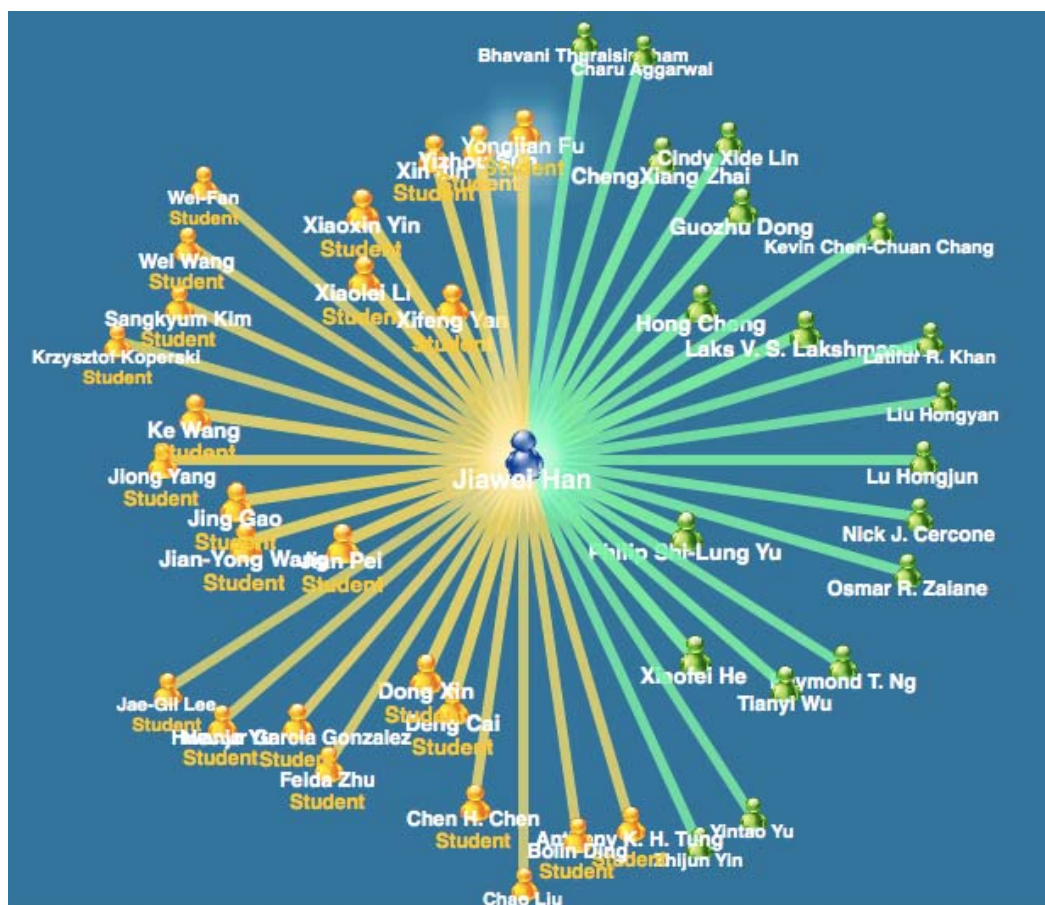
- 对于所有可能产生的队列，可能设置有一些约束条件（如所有队列必须以蓝色球 1-3 号为队列的开头，如写成 $B1 \Rightarrow B2 \Rightarrow B3 \Rightarrow *$ ；红色球 4、5 号必须连续出现，可以写成 $* \Rightarrow R4 \Rightarrow R5 \Rightarrow *$ ；这里“*”表示 0-n 个连续的球，“？”可以用来表示某 1 个球），在你的工具中可以编辑并且显示这些约束。
- 如果有 n 个满足这些约束条件的队列，则其中某一个给定队列的概率如何？

9. 社会网络图（研究及科学计算题）

社交网络（例如 Facebook、Flickr、人人网、开心网等）是当前一个非常流行的应用，有着广泛的使用人群。当前社交网络中，往往都是采用列表的方式来显示好友之间的关系；而在这个项目中，你将采用图形化的方式来展现这些关系，提供比列表更友好的用户体验。

基本要求（难度值=2）：

- 由于社交网络中人与人的关系数据并没有公开，所以我们在这个项目中采用公开的数据集：DBLP 提供的计算机文章信息 <http://dblp.uni-trier.de/xml/>，如果两个研究者是同一篇文章的作者，那么就认为互相是对方的好友。
- 本项目的结果是一个 GUI 工具，它可以根据人名来搜索并显示好友信息，并用图形化的方式展现出来。例如 Jiawei Han 的网络可能如下图：



在图中，我们可能对其中某个人感兴趣。在双击此人以后，将这个人朋友也显示在图中。考虑到图的布局，请尽量减少交叉边的出现。

- 给点任意两个人名，找出他们是否认识、或通过他们的哪些朋友介绍可以相互认识。例如 A 认识 B，B 认识 C，C 认识 D，那么我们可以知道 A 通过 B、然后 C 可以认识 D。
- 就你采用的数据结构和算法，分析他们的时间和空间复杂度。

进阶要求（难度值=2）：

- 在生成的图中，考虑如何评价并找出某位学者是重要的、关键性的专家。
- 如果考虑发表文章的领域，如何评价并找出领域内重要的专家。例如考虑文章引用次数，有一个 H-index，可以作为这样的指标。

10. 愤怒的小鸟（游戏题）

当小鸟们回到家里，发现巢里的蛋都没有了。四处寻找，发现是小猪偷走了所有的蛋。为了发泄愤怒的情绪，小鸟们不惜以自身作为肉弹，对小猪们发起攻击。小鸟的口号是：杀死所有小猪！

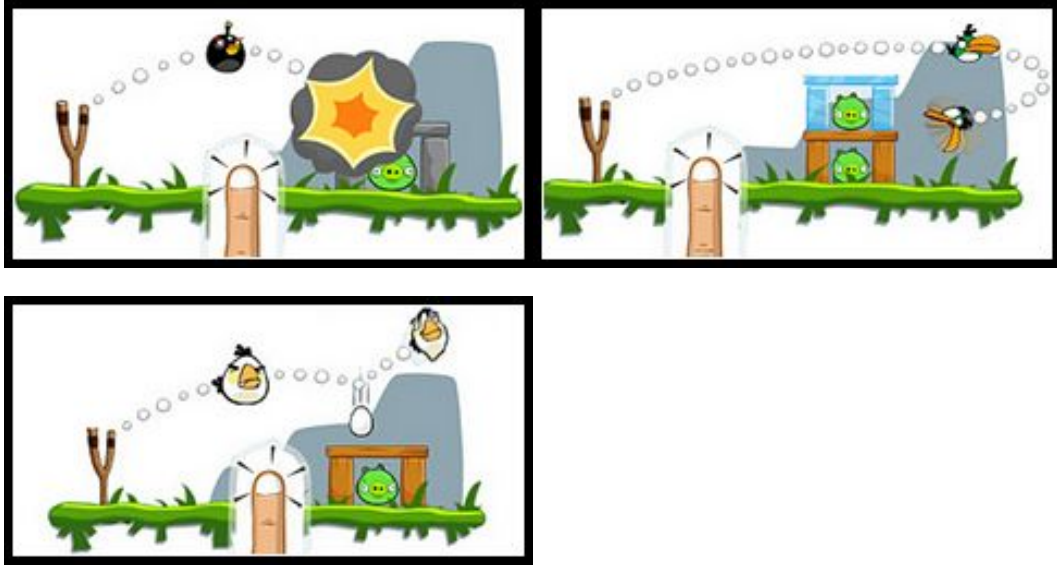
基本要求（难度值=3）：

- 用 C++或 Java 实现图形化的游戏。
- 小鸟们通过弹弓弹射出去，弹弓拉的力量和角度可能不同，因此初始速度和角度是可以不一样的。小鸟在飞行的过程中，需要考虑重力的影响，因此轨迹应该是抛物线。在发射小鸟的时候，是看不见小猪的，只有在发射前移动场景才能看见小猪和各种障碍物。
- 小猪可能有多个，放在不同的障碍物之间或之中。需要实现三种障碍类型：玻璃、木头、和石头。小鸟撞击到玻璃和木头可以造成障碍物的破碎，而石头则无法毁坏。小鸟只有以一定速度撞击到小猪上，才能杀死对方；或小猪一定速度撞击障碍物，也会被杀死。每关通过必须将所有小猪都杀死。
- 设计游戏的积分方法，并根据完成情况给分。设计出三个以上的场景，包括无阻挡障碍物和有阻挡障碍物的场景。

进阶要求（难度值=2）：

- 添加不同的小鸟类型：加速、分裂、炸弹、迂回和下蛋型，至少实现这 5 种类型中的 3 种。





- 针对你实现的每一类新小鸟类型，分别设计至少两个新场景。

11. 蚂蚁找食物的动画模拟（游戏题）

蚂蚁找食物时，他们知道很小范围内的眼前信息，而且根据这些局部信息利用几条简单的规则进行决策。但是，当一群蚂蚁同时遵守这样的几条简单规则，它们就可以很快地找到食物并通知其它蚂蚁去搬运食物。这些简单规则是：

1、范围：蚂蚁观察到的范围是一个方格世界，蚂蚁有一个参数为速度半径（一般是3），那么它能观察到的范围就是 3×3 个方格世界，并且能移动的距离也在这个范围之内。

2、环境：蚂蚁所在的环境是一个虚拟的世界，其中有障碍物，有别的蚂蚁，还有信息素，信息素有两种，一种是找到食物的蚂蚁洒下的食物信息素，一种是找到窝的蚂蚁洒下的窝的信息素。每个蚂蚁都仅仅能感知它范围内的环境信息。环境以一定的速率让信息素消失。

3、觅食规则：在每只蚂蚁能感知的范围内寻找是否有食物，如果有就直接过去。否则看是否有信息素，并且比较在能感知的范围内哪一点的信息素最多，这样，它就朝信息素多的地方走，并且每只蚂蚁多会以小概率犯错误，从而并不是往信息素最多的点移动。蚂蚁找窝的规则和上面一样，只不过它对窝的信息素做出反应，而对食物信息素没反应。

4、移动规则：每只蚂蚁都朝向信息素最多的方向移。当周围没有信息素指引的时候，蚂蚁会按照自己原来运动的方向惯性的运动下去，并且在运动的方向有一个随机的小的扰动。为了防止蚂蚁原地转圈，它会记住最近刚走过了哪些点，如果发现要走的下一点已经在最近走过了，它就会尽量避开。

5、避障规则：如果蚂蚁要移动的方向有障碍物挡住，它会随机的选择另一个方向，并且有信息素指引的话，它会按照觅食的规则行为。

6、播撒信息素规则：每只蚂蚁在刚找到食物或者窝的时候散发的信息素最多，并随着它走远的距离，播撒的信息素越来越少。

根据这几条规则，蚂蚁之间并没有直接的关系，但是每只蚂蚁都和环境发生交互，而通过信息素这个纽带，实际上把各个蚂蚁之间关联起来了。比如，当一只蚂蚁找到了食物，它并没有直接告诉其它蚂蚁这儿有食物，而是向环境播撒信息素，当其它的蚂蚁经过它附近的时候，就会感觉到信息素的存在，进而根据信息素的指引找到了食物。

蚂蚁究竟是怎么找到食物的呢？在没有蚂蚁找到食物时，环境没有有用的信息素，那么蚂蚁为什么会相对有效的找到食物呢？这要归功于蚂蚁的移动规则，尤其是在没有信息素时的移动规则。首先，它要能尽量保持某种惯性，这样使得蚂蚁尽量向前方移动（开始时，这个前方是随机固定的一个方向），而不是原地无谓的打转或者震动；其次，蚂蚁要有一定的随机性，虽然有了固定的方向，但它也不能像粒子一样直线运动下去，而是有一个随机的干扰。这样就使得蚂蚁运动起来具有了一定的目的性，尽量保持原来的方向，但又有新的试探，尤其当碰到障碍物时它会立即改变方向，这可以看成一种选择的过程，也就是环境的障碍物让蚂蚁的某个方向正确，而其他方向则不对。这就解释了为什么单个蚂蚁在复杂的诸如迷宫的地图中仍然能找到隐蔽得很好的食物。当然，在有一只蚂蚁找到了食物时，其他蚂蚁会沿着信息素很快找到食物的。

蚂蚁如何找到最短路径的？这要归功于信息素，另外要归功于环境。信息素多的地方显然经过这里的蚂蚁会多，因而会有更多的蚂蚁聚集过来。假设有两条路从窝通向食

物，开始的时候，走这两条路的蚂蚁数量同样多（或者较长的路上蚂蚁多，这也无关紧要）。当蚂蚁沿着一条路到达终点以后会马上返回来，这样，短的路蚂蚁来回一次的时间就短，这也意味着重复的频率就快，因而在单位时间里走过的蚂蚁数目就多，洒下的信息素自然也会多，自然会有更多的蚂蚁被吸引过来，从而洒下更多的信息素.....；而长的路正相反，因此，越来越多地蚂蚁聚集到较短的路径上来，最短的路径就近似找到了。也许有人会问局部最短路径和全局最短路的问题，实际上蚂蚁逐渐接近全局最短路的，为什么呢？这源于蚂蚁会犯错误，也就是它会按照一定的概率不往信息素高的地方走而另辟蹊径，这可以理解为一种创新，这种创新如果能缩短路途，那么根据刚才叙述的原理，更多的蚂蚁会被吸引过来。

基本要求（难度值=3）：

1. C++或 Java 实现一个蚂蚁觅食的动画过程。
2. 蚂蚁种群的数量，蚁窝的位置，食物位置可以通过 GUI 自由定义。
3. 发现各个适用的参数，例如信息素多少、消散速度、蚂蚁的移动速度、蚂蚁的记忆大小等。

进阶要求（难度值=2）：

1. 可以自由定义地图上障碍物（可能为不规则形状）。
2. 自由定义多个食物并模拟觅食过程。

12. 太阳系系统（游戏题）

编程模拟太阳系行星卫星的运行，可以计算什么时候出现日全食，什么时候出现“行星连珠”。



九星连珠

最近一次“行星连珠”发生在 2000 年 5 月 20 日，当然这是个渐近的过程，从 5 月 5 日就开始了。到 5 月 20 日这天，除天王星和海王星外，太阳系的其余七大行星——水星、金星、地球、火星、木星、土星、冥王星（已降为二级行星），将排列在一定的方向上，但不是如望文生义那样，像糖葫芦串成一条线，而是分散在一个有限的范围内。太阳系内九大行星实际上对黄道面（包括太阳在内的地球公转轨道的平面）各自略有倾斜，也就是说，就算“行星连珠”这九大行星也不会排列在一条直线上，而是散落参差，所谓“行星连珠”只存在于人们心目中。从这个意义上说，“行星连珠”与其说是天文学的研究对象，不如说是人们感兴趣的“视觉现象”。



九星连珠

科学家们现在根据下列四个前提来确定“行星连珠”：首先，行星的位置取为在黄道面上的投影位置；其次，在黄道面上，把行星聚集在太阳与地球连线（1）的附近，视为“行星连珠”，不考虑不包括太阳的“行星连珠”；第三，把地球与其他行星的连线与太阳与地球的连线构成的夹角（ θ ），作为“行星连珠”的量化“指标”。这个夹角取小于 90 度的锐角；第四，求出同一时刻各行星的 θ 角，取其构成的最大夹角，把 θ 角的最大值变为最小值的时刻视为“行星连珠”。这里，考虑的行星数目从 6 个到 9 个，并研究所有太阳系行星的组合。地球必须包括在内。

简而言之，其他行星来到地球与太阳连线附近时，将会发生“行星连珠”的现象。不过，这是在一个扇形的范围内发生的“行星连珠”，所以远离地球的其他行星距直线 1 也相当遥远。确定上述条件的理由是，在这个扇形范围内的行星作用于地球的引力方向大致相同。

科学家根据计算结果，选出了近 300 年间（1850——2150 年）7 个以上行星的“行星连珠”， θ 角的最大值，把角度小于 13 度的列入“行星连珠”，这种天象共有 17 次，距现在最近的一次“行星连珠”发生在 2000 年 5 月 20 日零时， θ 角 12.6 度。此时，水星、金星、地球、火星、木星、冥王星，这七大行星排列在 12.6 度的范围内，上一次“七星连珠”发生在 1965 年 3 月 6 日 9 时，水星、金星、地球、火星、土星、天王星、冥王星排列在 9.3 度的范围内。2149 年 12 月 6 日 4 时发生的将是“八星连珠”，其余 16 次都是“七星连珠”。2000 年 5 月 20 日这样的“行星连珠”为 30 年一遇，就人的一生来说是少见的，但从时间的大尺度来看是频繁发生的，并不罕见。

基本要求（难度值=3）

模拟日全食过程，仅有太阳、八大行星和月亮，需要使用 3D 技术，选择适合的角度（可能需要多视角）和观察视野。

进阶要求（难度值=2）

模拟行星连珠过程，追寻阿哥穿越的时间。

13. 抓老鼠 (Catcha Mouse) (游戏题)

在一个可以由多个六边形组成的地图中，有一只老鼠，可以向六个方向（上、下，左上，左下，右上，右下）移动，同时，某个或者多个六边形被定义为地洞。游戏开始，人可以在地图上任何一个六边形中放置捕鼠夹；然后，老鼠会以最短路径方式向最近地洞逃逸，每次移动一格；老鼠移动之后，人可以放置下一个捕鼠夹，由此人机次第进行游戏。老鼠每一次移动时，会重新计算最短路径（因为捕鼠夹相当于障碍，会使最短路径改变）。直至：

- 老鼠没有任何机会进入地洞，则人获胜；
- 老鼠最终进入地洞，则电脑获胜。

基本要求（难度值=2）：

- 用 C++或 Java 实现图形化的游戏。
- 布置 5 个以上场景，要求每个场景都会有解决方案是人可以获胜。
- 设置积分。

进阶要求（难度值=2）：

- 设置更好的老鼠逃逸的策略，增加游戏的难度。
- 3D 效果。



14. 自设题目

对于自设题目，有这样的原则性要求：

1. 题目要有一定规模和难度
2. GUI 人机交互、可视化
3. 具有较复杂的算法、数据结构和数据量

同学们在课程设计开始之前，提交详细的自找课题说明，本课程指导小组对其是否符合要求做出客观公正的判定。团队开发模式及从其它同学处的借鉴将不被允许。