数据结构课程设计自评表(2023)

组号：4

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **验收内容** | **系统架构（采用的编程语言、架构采用BS/CS /Android）** | **课程内日程管理** | | **课外活动日程管理** | | **临时事务日程管理** | | **日程导航** | | **用户数和日程时长、模拟系统时钟** | **日志功能** | **选做功能（功能简介、核心算法描述）** |
| **完成的基本功能描述** | **核心算法描述、算法优缺点及性能** | **完成的基本功能描述** | **核心算法描述、算法优缺点及性能** | **完成的基本功能描述** | **核心算法描述、算法优缺点及性能** | **完成的基本功能描述** | **核心算法描述、算法优缺点及性能** |
| **完成情况** | **使用Gamemaker游戏引擎作为集成开发环境，GML作为脚本语言。GML在使用上类似于基于C语言的其他编译性编程语言。架构采用CS。** | **管理员可以设置课程信息，可以给学生分配课程，若冲突会有冲突提示。学生可以查看课程信息，可以对课程进行“模糊查询”。** **提前提醒：每天0:00会提醒当天的所有课程。临近提醒：每门课程上课前会提醒。** | **使用公共课程信息表，每个课程、学生有唯一id，课程具有学生列表，学生具有选课列表，列表都用数组实现。更新课程信息时分别更新课程学生列表和学生选课列表；删除时不移动课程信息，以便课程信息的随机存取；添加课程时在课程信息表中寻找空位；冲突检测时查找与之冲突所有课程，遍历冲突课程的学生列表是否有该课学生。优点是数组信息读写较快，删除课程时可避免数据的大量移动，缺点是添加信息不够高效，修改信息时要维护各个学生的选课表。** | **学生可以设置活动信息，自行添加个人或集体活动，若发生冲突则提供三个备选时间。学生可以对活动进行“模糊查询”且可以对查询结果进行排序。学生可以设置活动闹钟。提前提醒功能（同课程）。** | **添加活动、更新信息和删除大致算法与课程管理相同。冲突检测可以通过课表O(1)实现，若有冲突根据当天最小冲突次数的时间点生成提示信息；加载时将所有课程、活动、临时事务名称和唯一id映射用哈希表存储以便查询，使用快速排序整理查询结果；通过课程表读取下一时间段的日程id实现提醒功能。** | **学生可以设置临时事务信息，自行添加临时事务，若冲突会有提示。学生可以对临时事务进行“模糊查询”且可以对查询结果进行排序。学生可以设置临时事务闹钟。临时事务完成后自动删除。** | **临时事务信息存储在一块连续的内存区域中，加载时根据每个事务所在时间段生成链表，链表依次存储该时段所有临时事务在内存区域中的位置，链表头节点用哈希表访问。优点是可通过事务在内存中的位置为其分配id，查询时根据id直接读取信息，缺点是临时事务较少时占用内存较多。** | **学生可以随时设定当前所在位置，然后通过日程信息进行导航，地图上会显示导航信息。对于课程或活动，导航显示从当前点到目标点的最短距离。对于同一时段的临时事务，导航显示途径所有临时事务的最短路径。** | **导航课程和活动时使用Dijkstra最短路算法得到两点间最短路径，导航多点临时事务时首先用最短路算出各目标点距离，对这些目标点进行用估值函数优先队列加剪枝的广搜，确定途径顺序后还原出完整路径。优点是堆优化最短路算法性能较好，缺点是途径点数很多时会出现卡顿。** | **用户数可达100个，日程时长16周。模拟系统时钟实现了可以随时暂停，随时设置时间流逝速度，以及随时跳跃至未来的某一天。** | **管理员和学生的几乎所有操作和系统通知都会被存储到日志中，日志可以通过log文件查看，也可以直接在程序内查看。在程序内可以快速查询任意时间段的日志。** | **1.实现了导航功能的图形界面展示：可以显示地图和导航路径，地图可放大缩小，且可以交互（可在地图上选择当前所在位置）。核心算法是按比例绘制地图信息和地图点存储绘制导航路径。**  **2.实现了课表图形界面。可以通过课表查看日程，课表可交互（点击课表可以显示日程的详细信息）。核心算法课表存储对应时段日程id，数组维护课表信息。** |