华南理工大学

C++实训作业报告

设计题目:俄罗斯方块

姓名:莫源欢 学号:202230273210

班级:22 软件工商双学位班

指导教师:朱延钊

设计报告

【问题描述】

一、 基本要求

1. 用类实现，使用继承和多态的概念；
2. 合理设计构造函数、复制构造函数和析构函数，不能出现内存泄漏的情况
3. 良好的编程网络（有语义的变量名、代码对齐、注释清晰、各种数据权限配置合理并检查合法性，并作出异常情况的处理等）；

二、 任务一

1. 友好的界面；
2. 随机产生各种方块，并提前预告给玩家；
3. 方块能自动下降；
4. 能通过键盘控制方块的移动和旋转并摆放在合适的位置；
5. 当底部有被方块填满的行后，要能够及时消去对应的方块；
6. 设计消去方块计分算法；
7. 当游戏已经无法继续进行时，则需要结束游戏，合理判断结束游戏的。时机。

三、 任务二

1. 利用文件方式，支持保存游戏进度，并从上次存档中恢复游戏；
2. 临时暂停和恢复游戏；
3. 保存玩家信息并支持多玩家存档和读取。

四、 任务三

1. 支持设置游戏参数，如方块下落速度、棋盘大小等。
2. 能够用合理的方式计算分数、出具游戏排行榜。

五、 任务四

1. 更友好的用户界面：可以选择方块样式和切换背景；可以显示游戏规则简介；其他你能想到的方面也可继续增加。
2. 可播放与切换背景音乐。
3. 其他更完善的功能，想到可继续增增加，如：支持按键设置（如把上下左右改为 wasd）；快捷键设置（如你原来将暂停按键设置为 p，也可以支持让用户自己修改）；恢复默认设置；复制与删除存档；……

【系统设计说明】

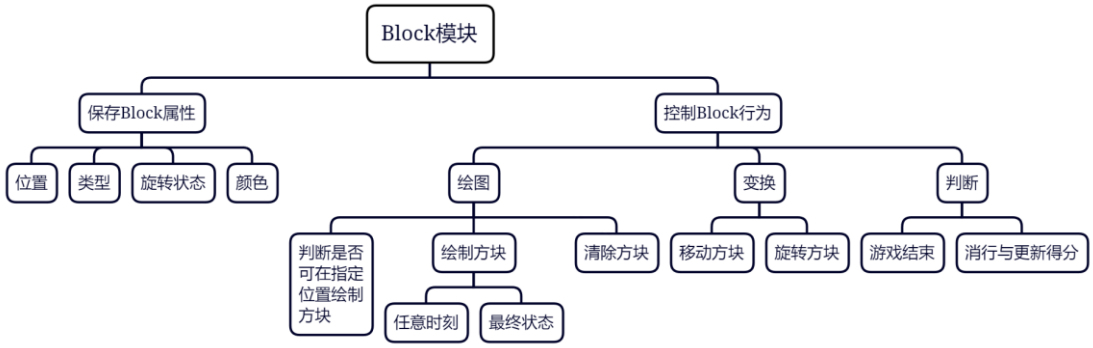
1. 数据结构说明

程序在出具游戏排行榜时，创建了 UserRecord 类对象数组临时保存各用户的 ID 和得分，从而通过排序算法对各用户的游戏得分进行排序并出具排行榜。

2. 程序功能说明

（1） 各模块功能

①



②

③

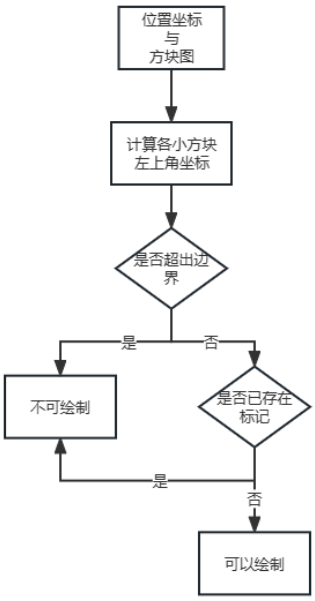
（2） 算法描述

①判断是否可在指定位置绘制方块

使用 ifEmpty 函数。

每一种类型的方块都由四个小方块构成，绘制一个大方块就是同时绘制四个小方块的过程。每一个大方块的位置由包围该大方块的最小矩形区域的左上角的坐标(x, y)确定，而每一个小方块的左上角坐标都可以通过(x, y)计算得到。

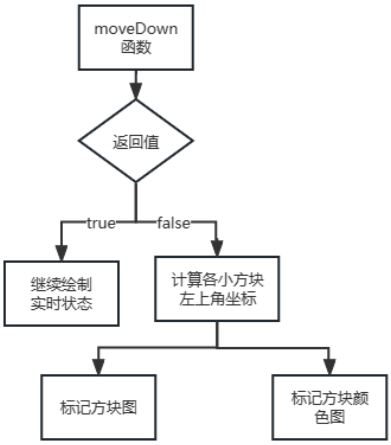
获取指定位置坐标(_x, _y)以及方块图（标记了页面每一个位置是否已被方块占据）。通过(_x, _y)计算得到四个小方块左上角在方块图中对应点的横纵坐标值，并用两个数组分别保存各点的横坐标和纵坐标值。逐个判断对应点是否超出游戏区域以及在方块图中是否已存在标记。



②绘制方块最终状态

使用 drawFinal 函数。

最终状态即为方块不可再下落的时候，可以由函数 moveDown 确定，而所谓的绘制最终状态实际上是在方块图和方块颜色图的对应位置进行相应的标注。与算法 1 类似，先通过(_x, _y)计算得到四个小方块左上角在方块图中对应点的横纵坐标值，并用两个数组分别保存各点的横坐标和纵坐标值，再根据坐标将方块图上对应点标记为 1，方块颜色图对应点标记为当前方块的颜色。由于每次在绘制新的方块前都先根据已更新的方块图和方块颜色图在标记位置绘制小方块，故在视图上的效果为原有的方块被固定。



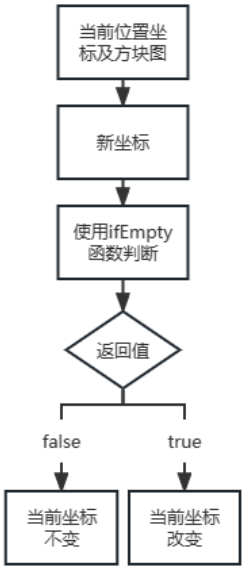
③方块变换

移动：使用 moveLeft, moveRight, moveDown 函数

移动的本质是改变当前位置坐标，在新位置重新绘制指定方块。

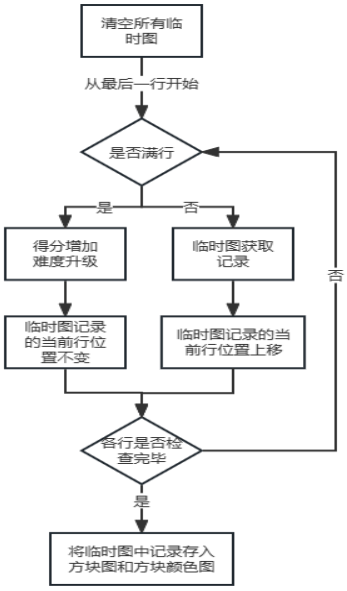
左右移动：左移和右移类似，以左移为例。获取当前位置坐标(now_x, now_y)以及方块图，由于方块每次移动距离为一个小方块的边长，将坐标(now_x-size, now_y)以及当前方块图传入 ifEmpty 函数判断是否可以绘制方块，若可以绘制则将当前位置坐标的横坐标改为 now_x-size。由于 getNextBlock 的状态仍为 0，故当前方块的各项属性只有横坐标发生改变，在以新坐标重新绘制方块时，在视图上的效果为原有的方块左移 1 个小方块的距离。

向下移动：由于能否向下移动同时反映了当前方块最终状态是否确定，故 moveDown 函数需要返回能否下移的结果，不能下移时则应调用 drawFinal 函数同时检查是否消行和结束游戏。moveDown 函数获取当前位置坐标(now_x, now_y)以及方块图，将坐标(now_x, now_y+size)以及当前方块图传入 ifEmpty 函数判断是否可以绘制方块，若可以绘制则将当前位置坐标的纵坐标改为 now_y+size，同时返回 1，若不可绘制则返回 0，并检查是否消行和结束游戏。



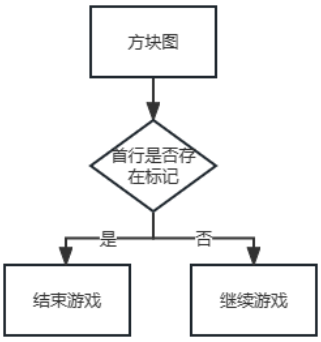
④消行与计分

首先清空临时方块图和临时方块颜色图，从最后一行开始，检查每一行是否满行，若满行则增加得分和计数器（用于实现得分每增加 10 次，难度等级增加 1 级，即方块下落速度增加一个单位），而临时图的记录位置不变，否则将方块图和方块颜色图各点的值存入临时方块图和临时颜色图，同时临时图的记录位置上移一行，当每一行都检查并完成相应记录后，则将临时图中的所有数据存入方块图和方块颜色图。



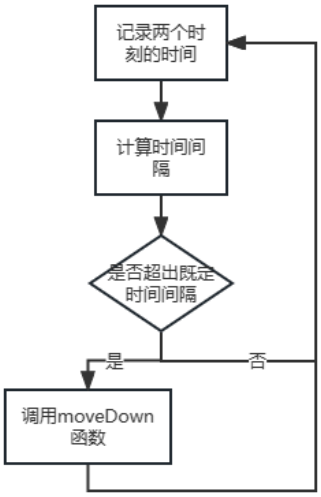
⑤判断游戏是否结束

通过判断方块图的首行是否存在标记为 1 的点即可判断游戏是否结束。



⑥方块自动下降

设置变量 t 并初始化为 0，通过 GetTickCount 函数获取当前已运行时间并存入变量 newtime，则时间间隔为 newtime-t，当时间间隔超出规定时间时则进行方块下移，具体操作见上文。



（3）接口说明

无。

（4）关键变量说明

①InitUser 类对象 init：保存了用户 ID 以及用户对窗口大小、游戏背景、背景音乐的偏好，同时控制背景音乐的播放。

②GameData 类对象 data：保存了游戏的核心数据，包括窗口的长、窗口的宽、游戏区域的长、预览框的大小、预览框左上角横坐标、预览框左上角纵坐标、得分区域的大小、得分区域左上角横坐标、得分区域左上角纵坐标、难度等级区域的大小、难度等级区域的横坐标、难度等级区域的纵坐标、方块边长大小。

③RunningState 类对象 run：保存了游戏在运行过程使用的各种需要经常改变取值的变量，包括当前方块左上角横坐标、当前方块左上角纵坐标、当前方块类型、当前方块旋转状态、当前方块颜色、下一方块类型、下一方块旋转状态、下一方块颜色、当前自设备启动后

的毫秒数、下落时间间隔、开始下落下一方块标志、游戏结束标志、当前游戏分数、当前难度等级、计数器、方块图、临时方块图、方块颜色图、临时方块颜色图、调色板。

④GameBG 类对象 gameBG：控制游戏背景的绘制

⑤Control 类对象 controlBG：控制预览区域、得分区域、难度等级区域的背景绘制，控制游戏的暂停和结束窗口的弹出，控制游戏排行榜的输出

⑥Block 类双指针 p1、p2：分别代表下一方块和当前方块，能通过多态实现不同类型方块的绘制、移动、旋转。

⑦UserRecord 类对象 u：保存了用户 ID 和有效得分。

【实现环境】

（1） 设计平台要求

本程序采用 Embarcadero Dev-C++ 6.3 编写。

（2） 运行环境要求

配置有 EasyX 图形库

【设计实现及分析】

本次程序设计过程中出现的问题是多方面，主要包括算法层面、运行结果等。

（1）算法层面：重点在于消行与积分模块，经过思考，最终考虑创建与方块图和方块颜色图相同的临时图解决，即先对方块图和方块颜色图进行处理，并将所需的结果记录到临时图中，再临时图的数据复制到方块图和方块颜色图。

（2）运行结果：部分功能在实现的过程中，运行结果有时与想法相去甚远。此时主要采用输出运行过程中多个关键变量的结果，从输出结果中分析得出设计的缺陷。

【讨论】

1.程序结构：本程序分为三大模块，Block 模块、Game 模块、主函数模块，不同模块负责了不同的功能，程序结构清晰，方便维护。

2.时空效率：统计多次编译运行的结果可知，编译时间约为 0.16s,输出大小约为 4.47MiB。

【系统使用说明】

本程序在使用时需要先由用户输入其 ID 及偏好，按任意键开始游戏后，通过方向键控制方块的变换，其中上键控制方块的旋转。在游戏过程中，可通过空格键暂停游戏，此时会弹出交互弹窗，如选择确定则继续游戏，否则退出游戏且当前游戏得分无法保存。当游戏失败时，再次弹出交互弹窗，如选择确定则以当前用户 ID 再来一局，且上一局的游戏得分无法保存，否则保存本次游戏得分和用户 ID 并出具游戏排行榜，按任意键退出游戏。

工作小结

本次实训提高了我对C++面向对象思想的认识以及编写代码的能力，也锻炼了我的专注度和耐心。

（1）保持专注：编写代码的过程并非易事，需要全身关注，精力基中，只有这样才能保证编写过程中思路清晰，减少不必要的细节错误。

（2）自顶向下，面向对象：在程序设计时，应先进行整体设计，再逐步精细化，这样才能做到全盘考虑，不至于顾此失彼，头重脚轻，而在功能的实现上，面向对象的思想既有利于问题的分析简化，也有助于系统的维护。

（3）保持良好的编程心态：不要被代码的长度吓到，更不要厌烦，遇到问题，沉着冷静耐心查找，也许最终会发现错误并非很难找。