**深 圳 大 学 实 验 报 告**

**课程名称： 计算机图形学**

**实验项目名称： 期中作业 俄罗斯方块**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师： 徐鹏飞**

**报告人： 叶翔宇 学号： 2021150153 班级： 国际班**

**实验时间： 2023年 9月27日 -- 2023年10月 29日**

**实验报告提交时间： 2023年 10月 25日**

**教务部制**

|  |
| --- |
| **实验目的：**  设计一个简化版的俄罗斯方块游戏 。游戏窗口大小是20×10的方形棋盘格阵列（下图左），游戏中出现的方块共有7种（下图右），每种方块的旋转中心由黑点标出。    **实验要求：**   1. 方块/棋盘格的渲染和方块向下移动   创建OpenGL绘制窗口，然后绘制网格线来完成对棋盘格的渲染。随机选择方块并赋上颜色，从窗口最上方中间开始往下自动移动，每次移动一个格子。初始的方块类型和方向也必须随机选择，另外可以通过键盘控制方块向下移动的速度，在方块移动到窗口底部的时候，新的方块出现并重复上述移动过程。   1. 方块叠加   不断下落的方块需要能够相互叠加在一起，即不同的方块之间不能相互碰撞和叠加。另外，所有方块移动不能超出窗口的边界。   1. 键盘控制方块的移动   通过方向键（上/下/左/右）来控制方块的移动。按“上”键使方块以旋转中心顺（逆）时针旋转，每次旋转90°，按“左”和“右”键分别将方块向左/右方向移动一格，按“下”键加速方块移动。   1. 游戏逻辑   当游戏窗口中的任意一行被方块占满，该行即被消除，所有上面的方块向下移动一格子。当整个窗口被占满而不能再出现新的方块时，游戏结束。通过按下“q”键结束游戏，和按下“r”键重新开始游戏。**完成代码后同时在main.cpp开头的注释中总结所实现的所有功能。**   1. 其他扩展   在以上基本内容的基础上，可以增加更多丰富游戏性的功能，如通过空格键使方块快速下落等。  **实验步骤：**  **功能补全：**  1、随机生成方块并赋上不同的颜色：  7种类型方块的定义：    在newtile()函数中设置随机生成不同种类的方块：    不同颜色的定义：    随机给方块上不同的颜色：    **2、方块之间、方块与边界之间的碰撞检测**  在checkvalid()函数原有条件（对边界的碰撞检测）外新增方块之间的碰撞检测：    **3、基于难度等级定义的方块的自动向下移动**  Init()函数中记录游戏窗口开始时间：    在main函数循环中新增方块自动下落的功能：    循环中不断截取当前的窗口时间nowtime，与开始时间starttime作差，达到对应难度的时间差值就执行一次movetile。  **4、棋盘格中每一行填充满之后自动消除**  补全checkfullrow()函数：    这里写的是单行检测第row行是否为满，首先遍历row行，检测board状态，若全为true，说明这行已满，进入到消除和下移操作。下移为遍历row行以上的所有行，将有方块的格子往下移一行。  Checkfullrow的调用：  我们知道可能出现一行满之前的最后一个时刻是方块下落到某个位置并固定的时候，即执行settile()的过程，因此调用checkfullrow应在方块放置后调用。    调用方式为从上到下遍历每一行，检测并消除满行。  **额外内容：**  **5、计分系统：每有一行消除增加50分**  **6、难度系统：难度分为5级，5级之前每得到600分就增加一级难度，5级之后不再增加，难度越高，方块下落速度越快**  **7、bonus系统：消除2行及以上会有bonus加成，具体体现在得分增加和技能槽蓄力速度增加，得分每多一行消除就额外增加25分，蓄力槽每有多一行消除就额外增加5%**  **8、技能与技能槽：每成功消除一次技能槽增加10%，每累计50%可以按'b'释放技能，技能可以连续释放，技能效果为消除最底层两行所有方块**  上述功能的相关定义如下：    计分函数calPoints()：    在前面的checkfullrow中，将其返回值改为了bool，如果为true，则记录bonus，  然后进入加分计算阶段，根据bonus数值的大小，对应计算加分的数值、技能槽应加数值。根据加分后的总分判断是否增加难度。  技能函数bomb()：    直接消除最底层两行，消除流程和checkfullrow中的消除一样。  **9、新增的键盘回调映射：**  **空格：使方块马上下落到顶部**    **'r': 使游戏重新开始**    **'p'： 暂停/恢复**    由于新增了暂停功能，为了保证重启游戏响应能够正常接收，将按键r的回调设置在switch结构外面，其余事件仍保留在switch结构内，使得在暂停的状态下也能够正常重启游戏。  空格键使方块立即下落到底部的函数：movetiletothebottom()    直接在循环中重复调用movetile直到方块落到底部即可实现效果。  重启游戏：restart()    重置分数、技能槽、难度，将游戏界面初始化。  暂停功能的实现：  全局变量增加一个暂停判断isPaused    主函数循环中检测是否暂停：    使用glfwWaitEvents()进入等待，实现暂停效果。  P按键映射部分：若游戏正在运行，则将暂停标志置为true，并记录暂停开始时间戳，否则将暂停标志置为false，记录暂停结束时间戳，并计算暂停的时间，以便于在实现方块自动下落的时候除去暂停的时间带来的影响。  **10、游戏结束的判断**  在settile()中判断方块位置的时候检查方块位置是否位于最上行，如果是，将testGameOver标志位设为true。    在最后若testGameOver为true，执行游戏结束函数    Game\_over()函数：    在gameover为true的时候，键盘响应基本都用不了了，因此重启游戏的键盘响应应放在这个判断外面。    **11、控制台打印菜单**    **实验结果：**  这里采用gif展示各个功能：  自动下落：    满行消除以及分数计算：    一次性消除多行以及bonus的计算：    难度增加：    技能释放以及连续释放技能：    高难度方块下落速度演示以及游戏暂停演示：    可以看到下落速度比起难度一明显提高。  游戏结束判定以及游戏重启：    **实验心得：**  本次实验我实现了方块形状的绘制。根据方块的类型选择不同的几何图形进行绘制。  此外，通过设计游戏循环控制逻辑。包括方块移动、下落、消行等游戏过程控制。利用定时器实现动画效果。  另一个值得重视的点是处理键盘事件。识别不同按键来控制方块移动和旋转等操作。使用GLUT检测键盘输入。  我还实现了检测方块是否触底或相互碰撞的功能。这是核心游戏规则实现。通过对board是否为真检验。  同时，我也实现分数和水平显示。使用OpenGL文本绘制函数输出游戏信息。  在设计的过程中，我认识到调试是必须的。OpenGL错误很难定位,需要分析每一步是否执行正确。 |

深圳大学学生实验报告用纸

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字： 2022年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。