**C语言项目报告**

**项目名称：课程主题大富翁——谁是肝帝**

**项目成员：夏启东 张茗雨**

**填写日期：2020.9.4**

1. **摘要**

我们项目的内容是设计一款以唐班化学方向的课程为主题的类大富翁棋类游戏。不同于传统的大富翁花钱买地盖楼的经典模式，我们还引入了玩家的精力属性，学习课程将消耗精力，且同一课程允许不同玩家学习，低课程等级的玩家将缴纳给高课程等级玩家相应的金钱，精力殆尽和金钱不足均导致游戏失败，同时我们还引入了玩家的道具栏和状态栏、丰富的随机事件以及新的棋盘样式，这大大增强了游戏的可玩性、策略性和趣味性。结合传统大富翁的思路和我们创新的规则和棋盘，我们通过C语言和easyX图形库实现了这款类大富翁棋类游戏。

1. **问题描述**

**2.1 背景介绍**

大富翁作为一款经典的棋类游戏，是许多90后乃至00后童年的回忆。至今市面上的大富翁已有大量的变式，基本上可分为两类：一类只是简单地替换主题，不修改和优化游戏的规则和棋盘；另一类是以庞大的棋盘、复杂的规则重新改良大富翁游戏的玩法。前者虽然缺乏创意，但保留了经典大富翁的原汁原味，后者有所创新，但是作为一副实体棋，过于复杂的新棋盘和新规则辅之以各种眼花缭乱的小配件将给玩家带来许多操作上的不便，且一些新棋盘和新规则也未能很好地解决传统大富翁游戏无效时长过长（指胜方已基本确定但是游戏还将进行很久）带来的枯燥。除此之外，传统大富翁还有操作单一、玩家间互动性不强、游戏策略简单等问题。我们考虑到电脑游戏没有实体棋因为复杂化导致的布置操作上的不便，设计以唐班化学方向的课程为主题的类大富翁棋类游戏企图找到一个平衡点，既尽可能地保留传统大富翁游戏的原汁原味，又针对性地通过一些规则和地图上的适当改良，优化解决传统大富翁中存在的弊端。

**2.2 须解决的问题**

**2.2.1** 独立的玩家结构体和课程结构体，各玩家的课程等级由课程结构体记录，这样如何简单地实现特定玩家到达特定位置改变课程结构体的属性？

**2.2.2** 玩家信息和课程信息如何及时更新并显示？传统大富翁通过地皮卡玩家可查看相应地皮的信息，通过何种手段来代替地皮卡在有限的区域内使玩家可以查看完整的课程信息？

**2.2.3** 如何表示玩家的位置？如何通过掷骰子（产生随机数）来改变玩家的位置？

**2.2.4** 玩家的一系列持续性状态如何在函数中记录经过的回合数？

**2.2.5** 如何解决键盘和鼠标交互的健壮性问题？如何处理玩家的非法操作？

**2.2.6** 如何美化图形界面？如何设计状态栏和物品栏？如何添加背景音乐以及掷骰子音效使游戏更带感？

**2.2.7** 如何在图形界面中实现通过类似控制台输出，适当提示玩家相应的操作和状态物品等的效果，从而使游戏更容易上手？

**2.2.8** 如何使判负的玩家不继续干扰游戏？

**2.2.9** 如何实现自动存档以及读档？

**2.2.10** 如何解决游戏画面闪烁问题？

1. **组内分工**

|  |  |  |
| --- | --- | --- |
| 姓名 | 主要任务 | 工作量 |
| 夏启东 | 游戏规则和棋盘的设计与程序逻辑的实现；图形界面框架的基本实现；程序健壮性的优化；Debug和游戏性的更新。 | 60% |
| 张茗雨 | 贴图的美工设计和图形界面交互的实现；自动存档和读档功能的实现；音乐素材的选取以及音效的实现。 | 40% |

1. **分析**

在已设计好游戏棋盘和规则的基础上，我们的项目实现主要经过以下四个阶段完成：

**第一阶段**，游戏规则（包括道具、状态等）的控制台实现和调试以及游戏性的优化更新，我们先通过最熟悉最简单的printf语句，此时不考虑用户的非法操作，也不考虑棋盘的显示（调试时自行参照棋盘图片），对游戏性的漏洞和平衡问题进行小心的排查和修复，先确保程序逻辑的正确性。

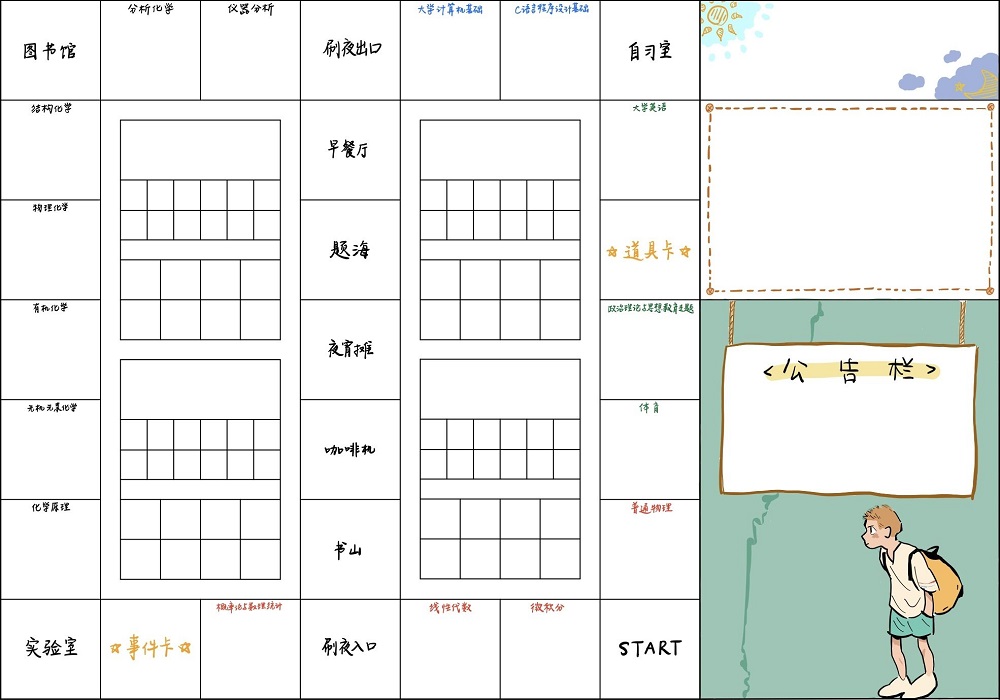
**第二阶段**，程序健壮性的完善和图形界面的初步加入，实现半控制台半easyX图形界面，通过while（true）死循环处理用户的非法操作，基本实现棋子在棋盘上的移动。

**第三阶段**，控制台printf提示信息逐步移植入图形界面，丰富鼠标交互的功能，删除一些冗余代码，优化代码结构，调试图形界面程序，通过BeginBatchDraw（）和EndBatchDraw（）函数通过批量画图基本解决了画面闪烁问题。

**第四阶段**，图形界面的美化和音效的添加，同时利用简单的文件处理实现了自动存档以及读档功能。

1. **设计**

**5.1 棋盘和规则的设计**

****

棋盘如上图所示。四名玩家的信息将更新于棋盘中部的四个大矩形框中，第一栏用于显示玩家的基本信息（名称、金钱、精力），第二栏的12个空格用于响应12种不同的状态，第三栏留空，第四栏的8个空格前7个用于响应7种不同的物品，第8个空格用于标记当前回合。最右侧是原控制台提示信息的移植，第一栏用于显示当前的回合数以及最终的胜负情况，第二栏用于显示非操作相关的提示语，第三栏用于显示操作相关的提示语。每门课程下方用于显示玩家的课程等级以及课程的基本信息（最高等级、精力消耗、金钱收益）。

我们的游戏与传统大富翁一样，是标准的四人游戏。每名玩家起始时拥有10点精力和500金钱。玩家起始位置于START处沿顺时针方向行进。每名玩家在回合内可以选择掷骰子行进或者使用道具，四名玩家回合结束计回合数+1且自动存档。每4个回合玩家精力会得到一定的恢复，恢复值与回合数正相关且有上限（初始为2点，每经过4个回合恢复值+1，上限为6点）。玩家到达相应课程位置时，若此时没有玩家达到该门课程的该等级，玩家可自行决定是否进行学习，学习将只消耗精力，反之玩家将自动学习，学习消耗精力的同时将缴纳给高课程等级玩家一定的金钱。不同的课程分为专业课程（黑色）、数理课程（红色）、计算机课程（蓝色）和其他课程（绿色），在棋盘中用不同的颜色标明，当玩家学习了所有的数理课程后，专业课程收益+60；当玩家学习了所有的计算机课程后，专业课收益+40。除其他课程外，所有课程均具有遗忘机制，当这些课程修满最高等级后玩家再次到达课程位置需消耗3点精力进行复习，否则课程等级-1。不同课程有着不同的最高等级、精力消耗和收益，详见下表：

|  |  |  |
| --- | --- | --- |
| **专业课程** | **精力消耗** | **收益** |
| 化学原理 | 1/3/5 | 60/120/180 |
| 无机元素化学 | 1/3/5 | 60/120/180 |
| 有机化学 | 1/2/3/4/5 | 20/40/80/160/320 |
| 物理化学 | 2/4/6/8 | 80/160/240/360 |
| 结构化学 | 2/4/6 | 80/160/240 |
| 分析化学 | 1/2/4 | 60/100/140 |
| 仪器分析 | 1/3/5 | 80/120/160 |
| **数理课程** | **精力消耗** | **收益** |
| 微积分 | 1/2/3/4/5 | 40/80/120/160/200 |
| 线性代数 | 1/3/5 | 60/120/180 |
| 概率论与数理统计 | 1/3/5 | 70/140/210 |
| 普通物理 | 1/2/3/4/5 | 40/80/120/160/200 |
| **计算机课程** | **精力消耗** | **收益** |
| 大学计算机基础 | 1/3/5 | 20/60/100 |
| C语言程序设计基础 | 2/4/6 | 60/120/240 |
| **其他课程** | **精力消耗** | **收益** |
| 大学英语 | 1/2/3/4/5/6/7 | 40/60/80/100/120/200 |
| 政治理论和思想教育专题 | 1/2/3/4/5/6/7 | 20/40/60/80/100/130/170 |
| 体育 | 1/2/3/4/5/6/7 | 20/40/60/80/100/130/170 |

玩家到达事件卡处将触发一次随机事件，具体事件如下表所示（红色事件为小概率事件）：

|  |  |
| --- | --- |
| **事件** | **效果** |
| 迟到 | 点亮状态——休息：跳过一回合 |
| 浪脱了整个假期 | 点亮状态——生疏：下次学习课程精力消耗+3 |
| 熬夜打游戏 | 精力-2 |
| 氪金 | 金钱-100 |
| 大病一场 | 精力-5 |
| 庆祝生日 | 精力+7，金钱-300 |
| 周末兼职 | 精力-3，金钱+200 |
| 月底生活费 | 金钱+150 |
| ddl终结 | 精力+6 |
| 奖学金 | 金钱+300 |
| 储蓄 | 金钱-200，接下来的5个回合每回合结束金钱+60 |
| 校园贷 | 金钱+400，接下来的5个回合每回合结束金钱-100 |
| 一夜暴富 | 当前金钱翻倍 |
| 时空裂隙 | 传送至非刷夜区的随机位置 |
| 洗劫一空 | 失去当前所有道具 |
| 噩梦 | 精力减半（至少剩余1点精力）并休息一回合 |
| 名师的指点 | 同时点亮状态——轻松学习、登峰造极、实践与理论相结合（具体效果见后） |

到达道具卡处将随机获得两件消耗性道具，具体道具如下表所示：

|  |  |
| --- | --- |
| **道具** | **效果** |
| 刷夜传送门 | 随机传送到刷夜区的某个位置并触发位置响应函数。 |
| 能量饮料 | 点亮状态——精力恢复：接下来的5个回合每回合结束恢复1点精力（精力<10时每回合恢复3点精力）。 |
| 移形换位 | 选定一名玩家与之交换位置 |
| 学习资料 | 点亮状态——轻松学习：接下来的5个回合学习课程精力消耗减半（至少消耗1点精力） |
| 交易许可证 | 向商店购买一件道具或以半价向一名玩家购买一件道具 |
| 百变骰子 | 可选择掷骰子获得的随机点数范围（1-3,4-6,7-9,10-12） |

玩家到达刷夜入口时可以选择消耗4点精力进入中间的刷夜区，也可选择不进入，若玩家选择进入刷夜区且掷骰子点数较大可直接穿越刷夜区，则不会消耗4点精力。刷夜区的各个位置（除刷夜出入口外）均会触发特殊效果：

|  |  |
| --- | --- |
| **位置** | **效果** |
| 书山 | 获得道具进阶读物（每获得一本进阶读物，专业课收益+20） |
| 咖啡机 | 点亮状态——提神醒脑：精力+15，接下来的10个回合每回合结束精力-1 |
| 夜宵摊 | 精力+7，金钱-200 |
| 题海 | 点亮状态——登峰造极：下次课程学习不需要消耗精力和缴纳金钱 |
| 早餐厅 | 精力+5 |

非刷夜区也有位置会触发特殊效果：

|  |  |
| --- | --- |
| **位置** | **效果** |
| 实验室 | 点亮状态——实践与理论相结合，可额外学习专业课程一次 |
| 图书馆 | 点亮状态——借阅：学习课程缴纳金钱减半，直至再次到达图书馆，若经十回合未再次到达图书馆，状态解除且金钱-250 |
| 自习室 | 点亮状态——自习中：跳过一回合，不会获得收益也不会受到道具影响 |
| 起始位置 | 金钱+400 |

当玩家的精力殆尽（精力<=0）或金钱不足（金钱<0）时，玩家判负。三名玩家判负后游戏结束。

**5.2 程序设计**

**5.2.1** **玩家结构体和课程结构体**

由游戏规则易知，玩家结构体（struct player）应当具有精力（int energy）、金钱（int gold）、物品栏（int item[7]）、状态栏（bool state[12]）、位置（int position）这五项基本属性，其中道具栏可用int数组实现，标记道具的数量，状态栏可用bool数组实现，标记状态的有无，但是通过角标引用相应的道具和状态是并不直观，故我们采用宏定义的方式将角标数与道具和状态的名称相对应，增强了代码的可读性。同理，位置这一变量也可用int来描述，只需通过宏定义将一系列连续的整数与棋盘中的各个格子的名称相对应即可（非刷夜区自START开始顺时针一圈分别宏定义为0~23，刷夜区自上而下分别宏定义为-1~-5）。除此之外，许多函数都需要针对某一特定玩家进行，故必须引入一项属性名称（char name）来代表玩家的身份，初始化四名玩家的名称分别初始化为字符A、B、C、D。之后通过switch case（playerX->name）语句即可判断当前操作的玩家。游戏结束时需要结算各个玩家的排名，故玩家还需要一项排名属性（int rank），初始化为1。可通过初始化全局变量playernumber为4，每当有一名玩家判负时，将playernumber赋值给他的rank，再使playernumber自减1，从而实现最终的排名。

课程结构体（struct course）应当具有课程最高等级（int max\_level）、精力消耗（int energy\_cost[7]）、金钱收益(int gold\_gain[7])、课程类型（int category）、遗忘机制（ bool forget）以及各名玩家的课程等级（int playerA\_level，playerB\_level，playerC\_level，playerD\_level）等属性。

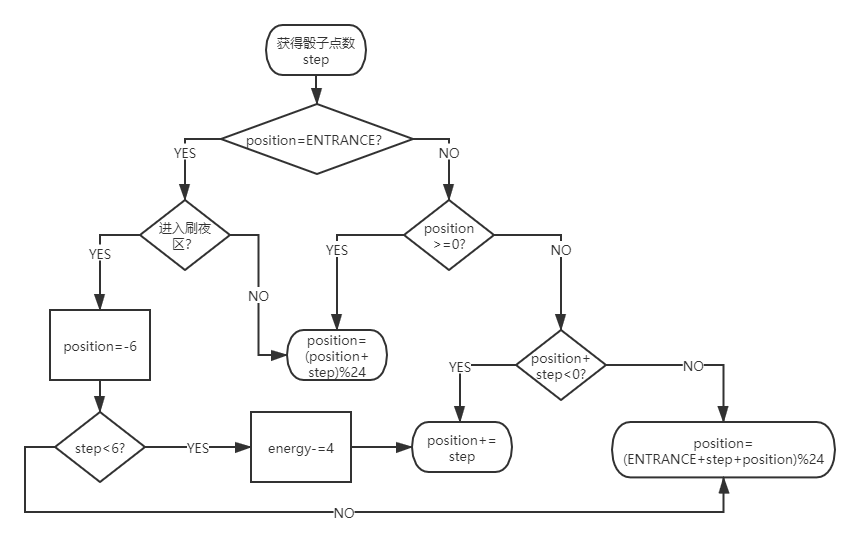
此处可以看到，当玩家的position属性取特定值即到达某课程位置时，并不能直接通过玩家的属性引用到所在课程的课程等级，故玩家还需要一个指针属性（int\* level），每次玩家到达某课程位置，各个玩家的level指针就分别指向该课程的玩家课程等级属性，从而实现引用。

**5.2.2 课程信息与课程玩家等级的查看**

考虑到经典的大富翁游戏棋盘中并没有空间在相应的地皮处显示地皮的所有信息，而是将其置于地皮卡中由玩家自行查看。我们设计的游戏也只允许玩家在回合开始前通过鼠标操作对课程的信息进行查看。具体为默认显示课程等级，回合开始前按下鼠标右键可显示课程信息，松开即又显示课程等级，这样的操作可以无限进行（通过while（true）死循环），直至玩家按下鼠标左键开始回合。但在此处我们注意到可能鼠标交互的缓冲区和键盘交互的缓冲区是独立的，此时玩家误操作键盘的指令会在后续的键盘输入中自动读入，故在此时一旦检测到键盘信息，都需要用\_getch（）函数捕获之。为了避免类似的问题，在无效的鼠标操作后，也立即使用FlushMouseMsgBuffer（）函数清空鼠标的缓冲区。

**5.2.3 玩家回合与道具使用**

回合开始时，先判定玩家是否有跳过回合的状态（休息中、自习中和失败），有即直接跳过回合。之后玩家可以选择掷骰子或使用道具。若玩家拥有道具百变骰子，在选择掷骰子后将提示是否使用百变骰子，普通骰子可以获得1-6之间的随机数，而百变骰子可以自选随机数的范围（1-3、4-6、7-9、10-12），此后获得的随机数将与玩家的position属性进行运算，获得下一个position，如前所述，非刷夜区自START开始顺时针一圈分别宏定义为0~23，刷夜区自上而下分别宏定义为-1~-5，考虑刷夜入口的特殊性，其具体算法可通过如下流程图表示：



玩家到达特定位置后将触发位置响应函数（详见**位置响应**）。玩家也可使用其他物品。

规则中所涉及改变状态的道具，均可通过调整玩家相应的state属性为true来完成，后用独立的持续性状态函数（详见**持续性状态**）实现具体的效果；涉及操作其他玩家的道具，此处需要小心考虑程序的健壮性问题，例如玩家不能与自己交易或交换，不能与判负或自习中的玩家交易或交换，金钱不足理应交易失败而不是直接判负，交易或交换失败后理应回到回合内重新选择操作而不是跳过回合以及处理玩家的其他无效操作等。

**5.2.4 位置响应**

玩家到达特定位置后将触发位置响应函数，相应课程位置依据规则学习课程（详见**课程学习**）。

玩家到达事件卡位置时将触发一项随机事件，有一部分小概率事件，其触发概率比其他事件低得多，可通过随机数落在的区间的大小控制概率的大小。

玩家到达道具卡位置时将随机获得两件消耗性道具（可能重复）。

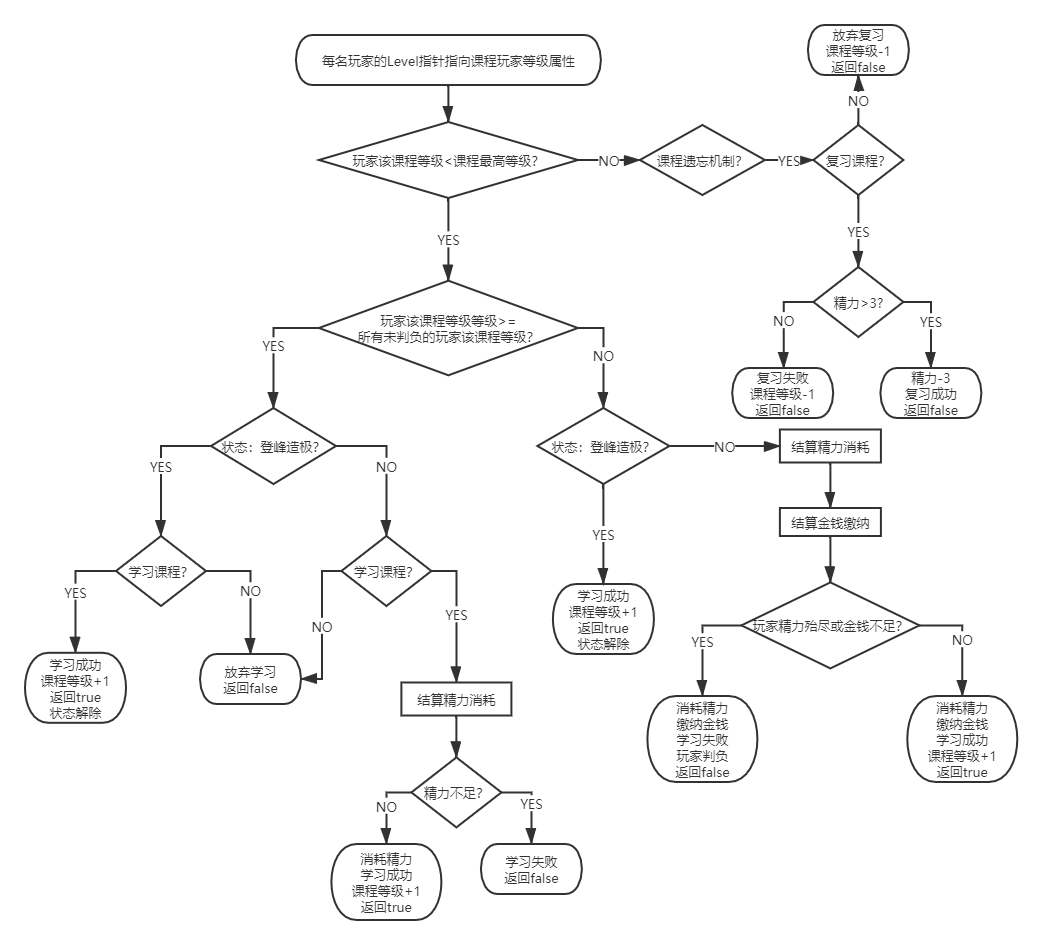
玩家到达刷夜区和非刷夜区特殊位置将触发相应的效果，基本上是通过读入玩家结构体指针改变玩家结构体的属性（包括点亮状态）。

**5.2.5 持续性状态**

我们先通过一个枚举类型的变量存储各持续性状态的持续回合数以方便修改，每个持续性状态函数均会读入玩家结构体指针，则为每个玩家定义一个专属的静态整型变量（static int），若玩家拥有响应状态，该静态变量将从其相应的持续回合数自减并伴随触发相应效果，直至该静态变量递减至0，之后将玩家相应状态属性调为false，同时将通过赋值使静态变量复原。

**5.2.6 课程学习**

玩家到达相应课程位置时将执行课程学习函数，函数读入课程与相应玩家结构体的指针，首先使四名玩家的level指针指向当前课程的四项玩家等级属性，接着根据用户的操作和状态来结算金钱的缴纳和精力的消耗，同时考虑实践与理论相结合这一状态的健壮性，我们为课程学习函数设置了bool型的返回值，若初次未学习或学习失败就不再询问是否继续学习，其具体流程如下流程图所示：



此处注意到，若课程已达到最高等级（可能是经初次学习达到最高等级，课程学习函数返回true），实践与理论相结合状态也没有必要触发，故触发实践与理论相结合函数还需判断课程等级是否已达到最高等级。除此之外，上述结算精力消耗中需判断学习玩家的状态（生疏、轻松学习），金钱消耗函数需考虑低课程等级玩家的状态（借阅）、高课程等级玩家的状态（失败、自习中）以及高课程等级玩家的专业课收益加成情况。

**5.2.7 自动存档和读档**

考虑到传统的大富翁游戏可能需要继续上次未完成的游戏，但是往往不必要继续更早的游戏。同时我们也不希望在每个回合结束都询问玩家是否存档，造成不必要的繁琐，因此我们设计了最简单的自动存档。每个回合结束（指四个玩家依次完成一轮回合后），程序将把当前的回合数、各玩家和课程结构体的所有属性写入documents.text文件中保存，游戏结束时自动清空文件。如图，游戏开始界面中，玩家可以选择开始新游戏或读取存档。为体现按钮的效果，鼠标移至相应位置处我们将使用红色矩形框以示强调。



1. **实施**

**6.1 主函数——程序主体框架**

/\*主函数\*/

int main(void)

{

/\*播放背景音乐\*/

mciSendString(\_T("open D:\\C\_Project\_Resource\\BGM.mp3 alias bkmusic"), NULL, 0, NULL);

mciSendString(\_T("play bkmusic repeat"), NULL, 0, NULL);

/\*选择新游戏或读取存档\*/

initgraph(1000, 700);

IMAGE cover;

loadimage(&cover, \_T("D:\\C\_Project\_Resource\\cover.jpg"));

putimage(0, 0, &cover);

while (true)

{

if (MouseHit())

{

MOUSEMSG m = GetMouseMsg();

if (m.x > 340 && m.x < 660 && m.y > 400 && m.y < 500)

{

setlinecolor(RED);

setlinestyle(PS\_SOLID, 4);

rectangle(335, 395, 665, 505);

}

else if (m.x > 340 && m.x < 660 && m.y > 520 && m.y < 620)

{

setlinecolor(RED);

setlinestyle(PS\_SOLID, 4);

rectangle(335, 515, 665, 625);

}

else

{

setlinecolor(WHITE);

setlinestyle(PS\_SOLID, 4);

rectangle(335, 395, 665, 505);

rectangle(335, 515, 665, 625);

}

if (m.uMsg == WM\_LBUTTONUP && m.x > 340 && m.x < 660 && m.y > 400 && m.y < 500)

{

/\*清除存档\*/

FILE\* clear;

clear = fopen("D:\\C\_Project\_Resource\\documents.txt", "w");

fclose(clear);

clear = fopen("D:\\C\_Project\_Resource\\continous\_state\\Borrowing.txt", "w");

fclose(clear);

clear = fopen("D:\\C\_Project\_Resource\\continous\_state\\Saving.txt", "w");

fclose(clear);

clear = fopen("D:\\C\_Project\_Resource\\continous\_state\\Loan.txt", "w");

fclose(clear);

clear = fopen("D:\\C\_Project\_Resource\\continous\_state\\Recovering.txt", "w");

fclose(clear);

clear = fopen("D:\\C\_Project\_Resource\\continous\_state\\Easier.txt", "w");

fclose(clear);

clear = fopen("D:\\C\_Project\_Resource\\continous\_state\\Fresh.txt", "w");

fclose(clear);

break;

}

if (m.uMsg == WM\_LBUTTONUP && m.x > 340 && m.x < 660 && m.y > 520 && m.y < 620)

{

load\_game();

load\_continous\_state = true;

break;

}

}

if (\_kbhit())

\_getch();

}

/\*游戏初始化\*/

IMAGE map;

loadimage(&map, \_T("D:\\C\_Project\_Resource\\map.jpg"));

putimage(0, 0, &map);

setbkmode(TRANSPARENT);

srand(time(NULL));

course\_start\_up();

TCHAR s[50];

RECT r;

BeginBatchDraw();

settextcolor(BLACK);

show\_course\_level();

setlinecolor(RED);

if (playerA.state[FAILURE])

delete\_course\_level(&playerA);

if (playerB.state[FAILURE])

delete\_course\_level(&playerB);

if (playerC.state[FAILURE])

delete\_course\_level(&playerC);

if (playerD.state[FAILURE])

delete\_course\_level(&playerD);

show\_item();

show\_player(&playerA, false);

show\_state(&playerA);

item\_number(&playerA);

show\_player(&playerB, false);

show\_state(&playerB);

item\_number(&playerB);

show\_player(&playerC, false);

show\_state(&playerC);

item\_number(&playerC);

show\_player(&playerD, false);

show\_state(&playerD);

item\_number(&playerD);

EndBatchDraw();

/\*开始游戏\*/

while (true)

{

\_stprintf\_s(s, \_T("第%d回合"), Round);

r = { 700,25,1000,75 };

settextcolor(BLACK);

settextstyle(25, 0, \_T("宋体"));

drawtext(s, &r, DT\_CENTER | DT\_WORDBREAK);

turn(&playerA);

turn(&playerB);

turn(&playerC);

turn(&playerD);

settextcolor(WHITE);

settextstyle(25, 0, \_T("宋体"));

drawtext(s, &r, DT\_CENTER | DT\_WORDBREAK);

Round += 1;

int energy\_plus;

if (Round % 4 == 0) //每四个回合所有玩家恢复2-6点精力（与回合数相关）

{

struct player copyA = playerA;

struct player copyB = playerB;

struct player copyC = playerC;

struct player copyD = playerD;

energy\_plus = 2 + Round / 8;

if (energy\_plus > 6)

energy\_plus = 6;

\_stprintf\_s(s, \_T("新的一天开始了\n所有玩家精力+%d"), energy\_plus);

settextcolor(BLACK);

settextstyle(25, 0, \_T("宋体"));

drawtext(s, &r, DT\_CENTER | DT\_WORDBREAK);

Sleep(2500);

settextcolor(WHITE);

settextstyle(25, 0, \_T("宋体"));

drawtext(s, &r, DT\_CENTER | DT\_WORDBREAK);

if (!playerA.state[FAILURE])

playerA.energy += energy\_plus;

if (!playerB.state[FAILURE])

playerB.energy += energy\_plus;

if (!playerC.state[FAILURE])

playerC.energy += energy\_plus;

if (!playerD.state[FAILURE])

playerD.energy += energy\_plus;

BeginBatchDraw();

show\_player(&copyA, true);

show\_player(&copyB, true);

show\_player(&copyC, true);

show\_player(&copyD, true);

show\_player(&playerA, false);

show\_player(&playerB, false);

show\_player(&playerC, false);

show\_player(&playerD, false);

EndBatchDraw();

}

if (playernumber <= 1)

{

\_stprintf\_s(s, \_T("游戏结束\n排名：\n玩家A：第%d名 玩家B：第%d名\n玩家C：第%d名 玩家D：第%d名\n按任意键退出游戏"), playerA.rank, playerB.rank, playerC.rank, playerD.rank);

r = { 700,0,1000,100 };

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(s, &r, DT\_CENTER | DT\_WORDBREAK);

/\*清除存档\*/

FILE\* clear;

clear = fopen("D:\\C\_Project\_Resource\\documents.txt", "w");

fclose(clear);

clear = fopen("D:\\C\_Project\_Resource\\continous\_state\\Borrowing.txt", "w");

fclose(clear);

clear = fopen("D:\\C\_Project\_Resource\\continous\_state\\Saving.txt", "w");

fclose(clear);

clear = fopen("D:\\C\_Project\_Resource\\continous\_state\\Loan.txt", "w");

fclose(clear);

clear = fopen("D:\\C\_Project\_Resource\\continous\_state\\Recovering.txt", "w");

fclose(clear);

clear = fopen("D:\\C\_Project\_Resource\\continous\_state\\Easier.txt", "w");

fclose(clear);

clear = fopen("D:\\C\_Project\_Resource\\continous\_state\\Fresh.txt", "w");

fclose(clear);

\_getch();

closegraph();

break;

}

else

{

\_stprintf\_s(s, \_T("游戏已自动保存"));

settextcolor(BLACK);

settextstyle(25, 0, \_T("宋体"));

drawtext(s, &r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1500);

settextcolor(WHITE);

settextstyle(25, 0, \_T("宋体"));

drawtext(s, &r, DT\_CENTER | DT\_WORDBREAK);

save\_game(); //回合结束，游戏自动保存

}

}

}

**6.2 玩家回合函数——玩家主要操作**

此处我们使用了批量画图BeginBatchDraw（）与EndBatchDraw（）函数避免了画面的闪烁，使用回合开始前事先复制的结构体在回合结束时对玩家信息进行清除

/\*玩家回合函数\*/

void turn(struct player\* playerX)

{

char input;

int step;

/\*更新玩家物品和位置\*/

BeginBatchDraw();

show\_item();

item\_number(&playerA);

item\_number(&playerB);

item\_number(&playerC);

item\_number(&playerD);

signal(playerX, false);

if (!playerA.state[FAILURE])

select\_position(&playerA, false);

if (!playerB.state[FAILURE])

select\_position(&playerB, false);

if (!playerC.state[FAILURE])

select\_position(&playerC, false);

if (!playerD.state[FAILURE])

select\_position(&playerD, false);

EndBatchDraw();

struct player copyA = playerA;

struct player copyB = playerB;

struct player copyC = playerC;

struct player copyD = playerD;

/\*回合开始前查看课程信息\*/

view\_course();

if (playerX->state[REST] || playerX->state[STUDYING] || playerX->state[FAILURE]) //判断是否跳过回合

{

playerX->state[REST] = false;

playerX->state[STUDYING] = false;

if (playerX->state[FAILURE])

{

\_stprintf\_s(guide\_text, \_T("玩家%c已判负"), playerX->name);

settextcolor(RED);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

}

else

{

\_stprintf\_s(guide\_text, \_T("玩家%c跳过回合"), playerX->name);

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

}

}

else

{

while (true)

{

\_stprintf\_s(guide\_text, \_T("空格:掷骰子，A:使用刷夜传送门\nS:使用移形换位，D:使用能量饮料\nF:使用学习资料，G:使用交易许可证"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

input = \_getch();

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

if (input == ' ') //按下空格键掷骰子

{

if (playerX->item[MAGIC\_DICE] > 0)

{

\_stprintf\_s(guide\_text, \_T("Q：使用百变骰子（其他键跳过使用）"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

input = \_getch();

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

if (input == 'q' || input == 'Q') //按下“Q”键使用百变骰子

{

playerX->item[MAGIC\_DICE] -= 1;

step = magic\_dice();

}

else

step = roll();

}

else

step = roll();

mciSendString(\_T("close rollmusic"), NULL, 0, NULL);

mciSendString(\_T("open D:\\C\_Project\_Resource\\dice\_music.mp3 alias rollmusic"), NULL, 0, NULL);

mciSendString(\_T("play rollmusic"), NULL, 0, NULL);

if (playerX->position == ENTRANCE) //刷夜区入口处按下空格键不进入刷夜区，其他键进入刷夜区并消耗4点精力，穿越刷夜区不消耗精力

{

\_stprintf\_s(guide\_text, \_T("骰子点数：%d\n空格：不进入刷夜区（其他键进入刷夜区并消耗4点精力，穿越刷夜区不消耗精力)\n"), step);

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

input = \_getch();

if (input == ' ')

{

select\_position(playerX, true);

playerX->position = (playerX->position + step) % 24;

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

position(playerX);

break;

}

else

{

select\_position(playerX, true);

playerX->position = -6;

if (step < 6)

playerX->energy -= 4;

}

}

else

{

\_stprintf\_s(guide\_text, \_T("骰子点数：%d"), step);

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(2000);

}

if (playerX->position >= 0 && playerX->position != ENTRANCE) //非刷夜区行走

{

select\_position(playerX, true);

playerX->position = (playerX->position + step) % 24;

}

if (playerX->position < 0) //刷夜区行走

{

select\_position(playerX, true);

if (playerX->position + step < 0)

playerX->position += step;

else

playerX->position = (EXIT + step + playerX->position) % 24;

}

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

position(playerX);

break;

}

else if ((input == 'a' || input == 'A') && playerX->item[STAYING\_UP\_DOOR] > 0)

{

\_stprintf\_s(guide\_text, \_T("传送至刷夜区内的随机位置"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(2000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

select\_position(playerX, true);

playerX->position = -rand() % 5 - 1;

playerX->item[STAYING\_UP\_DOOR] -= 1;

position(playerX);

break;

}

if ((input == 's' || input == 'S') && playerX->item[SWAP] > 0)

{

if (swap(playerX))

{

playerX->item[SWAP] -= 1;

break;

}

}

if ((input == 'd' || input == 'D') && playerX->item[ENERGY\_DRINK] > 0)

{

\_stprintf\_s(guide\_text, \_T("点亮状态——精力恢复：接下来的五个回合每回合结束精力+1（精力<10时每回合结束精力+3）"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(2000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

playerX->state[RECOVERING] = true;

playerX->item[ENERGY\_DRINK] -= 1;

break;

}

if ((input == 'f' || input == 'F') && playerX->item[LEARNING\_MATERIALS] > 0)

{

\_stprintf\_s(guide\_text, \_T("点亮状态——轻松学习：接下来的五个回合学习课程精力消耗减半(至少消耗1点精力)"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(2000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

playerX->state[EASIER] = true;

playerX->item[LEARNING\_MATERIALS] -= 1;

break;

}

if ((input == 'g' || input == 'G') && playerX->item[TRADING\_LICENSE] > 0)

{

if (trading\_license(playerX))

{

\_stprintf\_s(guide\_text, \_T("交易成功"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

playerX->item[TRADING\_LICENSE] -= 1;

break;

}

else

{

\_stprintf\_s(guide\_text, \_T("交易无效"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

}

}

}

}

/\*判定持续性状态\*/

easier(playerX);

recovering(playerX);

borrowing(playerX);

saving(playerX);

loan(playerX);

fresh(playerX);

load\_continous\_state = false;

/\*更新玩家信息\*/

BeginBatchDraw();

signal(playerX, true);

show\_player(&copyA, true);

show\_player(&copyB, true);

show\_player(&copyC, true);

show\_player(&copyD, true);

show\_player(&playerA, false);

show\_state(&playerA);

show\_player(&playerB, false);

show\_state(&playerB);

show\_player(&playerC, false);

show\_state(&playerC);

show\_player(&playerD, false);

show\_state(&playerD);

EndBatchDraw();

}

**6.3** **课程学习——复杂的程序逻辑**

/\*收益加成函数\*/

int extra(struct player\* playerX, struct course\* courseX)

{

int extra\_gold = 0;

if (courseX->category == MAJOR) //专业课收益加成：数理课程+60，计算机课程+40，每本进阶读物+20

{

switch (playerX->name)

{

case 'A':

if (cal.playerA\_level > 0 && l\_a.playerA\_level > 0 && p\_a\_s.playerA\_level > 0 && g\_p.playerA\_level > 0)

{

\_stprintf\_s(guide\_text, \_T("玩家A已学习所有数理通识课程，专业课程收益+60"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

extra\_gold += 60;

}

if (b\_o\_c\_e.playerA\_level > 0 && c\_p.playerA\_level > 0)

{

\_stprintf\_s(guide\_text, \_T("玩家A已学习所有计算机通识课程，专业课程收益+40"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

extra\_gold += 40;

}

extra\_gold += playerA.item[ADVANCED\_READINGS] \* 30;

break;

case 'B':

if (cal.playerB\_level > 0 && l\_a.playerB\_level > 0 && p\_a\_s.playerB\_level > 0 && g\_p.playerB\_level > 0)

{

\_stprintf\_s(guide\_text, \_T("玩家B已学习所有数理通识课程，专业课程收益+60"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

extra\_gold += 60;

}

if (b\_o\_c\_e.playerB\_level > 0 && c\_p.playerB\_level > 0)

{

\_stprintf\_s(guide\_text, \_T("玩家B已学习所有计算机通识课程，专业课程收益+40"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

extra\_gold += 40;

}

extra\_gold += playerB.item[ADVANCED\_READINGS] \* 30;

break;

case 'C':

if (cal.playerC\_level > 0 && l\_a.playerC\_level > 0 && p\_a\_s.playerC\_level > 0 && g\_p.playerC\_level > 0)

{

\_stprintf\_s(guide\_text, \_T("玩家C已学习所有数理通识课程，专业课程收益+60"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

extra\_gold += 60;

}

if (b\_o\_c\_e.playerC\_level > 0 && c\_p.playerC\_level > 0)

{

\_stprintf\_s(guide\_text, \_T("玩家C已学习所有计算机通识课程，专业课程收益+40"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

extra\_gold += 40;

}

extra\_gold += playerC.item[ADVANCED\_READINGS] \* 30;

break;

case 'D':

if (cal.playerD\_level > 0 && l\_a.playerD\_level > 0 && p\_a\_s.playerD\_level > 0 && g\_p.playerD\_level > 0)

{

\_stprintf\_s(guide\_text, \_T("玩家D已学习所有数理通识课程，专业课程收益+60"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

extra\_gold += 60;

}

if (b\_o\_c\_e.playerD\_level > 0 && c\_p.playerD\_level > 0)

{

\_stprintf\_s(guide\_text, \_T("玩家D已学习所有计算机通识课程，专业课程收益+40"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

extra\_gold += 40;

}

extra\_gold += playerD.item[ADVANCED\_READINGS] \* 30;

break;

}

}

return extra\_gold;

}

/\*金钱缴纳函数\*/

void pay(struct player\* playerX, struct player\* playerY, struct course\* courseX)

{

if (\*playerY->level > \* playerX->level && !playerY->state[FAILURE])

{

if (playerY->state[STUDYING])

{

\_stprintf\_s(guide\_text, \_T("玩家%c自习中，不会获得收益"), playerX->name);

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(2000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

return;

}

else

{

int gold\_gain;

if (playerX->state[BORROWING])

gold\_gain = (courseX->gold\_gain[\*playerY->level - 1] + extra(playerY, courseX)) / 2;

else

gold\_gain = courseX->gold\_gain[\*playerY->level - 1] + extra(playerY, courseX);

\_stprintf\_s(guide\_text, \_T("玩家%c缴纳给玩家%c %d金钱"), playerX->name, playerY->name, gold\_gain);

switch (playerY->name)

{

case 'A':

settextcolor(LIGHTRED);

break;

case 'B':

settextcolor(BLUE);

break;

case 'C':

settextcolor(MAGENTA);

break;

case 'D':

settextcolor(GREEN);

break;

}

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1500);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

playerX->gold -= gold\_gain;

playerY->gold += gold\_gain;

}

}

}

/\*金钱与精力消耗结算函数\*/

bool cost(struct player\* playerX, struct course\* courseX)

{

bool learned = false;

char input;

if (\*playerX->level < courseX->max\_level)

{

if ((\*playerA.level <= \*playerX->level || playerA.state[FAILURE]) && (\*playerB.level <= \*playerX->level || playerB.state[FAILURE]) && (\*playerC.level <= \*playerX->level || playerC.state[FAILURE]) && (\*playerD.level <= \*playerX->level || playerD.state[FAILURE])) //优先学习者无需缴纳金钱

{

if (playerX->state[SKILLFUL])

{

\_stprintf\_s(guide\_text, \_T("状态：登峰造极\nL：学习课程（其他键放弃学习）"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

input = \_getch();

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

if (input == 'l' || input == 'L') //按下“L”键进行学习，其他键放弃学习

{

\_stprintf\_s(guide\_text, \_T("学习成功"));

settextcolor(RED);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

BeginBatchDraw();

settextcolor(WHITE);

show\_course\_level();

setlinecolor(WHITE);

if (playerA.state[FAILURE])

delete\_course\_level(&playerA);

if (playerB.state[FAILURE])

delete\_course\_level(&playerB);

if (playerC.state[FAILURE])

delete\_course\_level(&playerC);

if (playerD.state[FAILURE])

delete\_course\_level(&playerD);

\*playerX->level += 1;

settextcolor(BLACK);

show\_course\_level();

setlinecolor(RED);

if (playerA.state[FAILURE])

delete\_course\_level(&playerA);

if (playerB.state[FAILURE])

delete\_course\_level(&playerB);

if (playerC.state[FAILURE])

delete\_course\_level(&playerC);

if (playerD.state[FAILURE])

delete\_course\_level(&playerD);

EndBatchDraw();

learned = true;

playerX->state[SKILLFUL] = false;

}

}

else

{

\_stprintf\_s(guide\_text, \_T("L：学习课程（其他键放弃学习）"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

input = \_getch();

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

if (input == 'l' || input == 'L') //按下“L”键进行学习，其他键放弃学习

{

int energy\_cost = courseX->energy\_cost[\*playerX->level];

if (playerX->state[EASIER])

{

\_stprintf\_s(guide\_text, \_T("状态：轻松学习"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

if (energy\_cost > 1)

energy\_cost /= 2;

}

if (playerX->state[UNFAMILIAR])

{

\_stprintf\_s(guide\_text, \_T("状态：生疏"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

energy\_cost += 3;

}

if (playerX->energy <= energy\_cost)

{

\_stprintf\_s(guide\_text, \_T("精力不足，学习失败"));

settextcolor(RED);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

return false;

}

else

{

\_stprintf\_s(guide\_text, \_T("学习成功"));

settextcolor(RED);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

playerX->energy -= energy\_cost;

BeginBatchDraw();

settextcolor(WHITE);

show\_course\_level();

setlinecolor(WHITE);

if (playerA.state[FAILURE])

delete\_course\_level(&playerA);

if (playerB.state[FAILURE])

delete\_course\_level(&playerB);

if (playerC.state[FAILURE])

delete\_course\_level(&playerC);

if (playerD.state[FAILURE])

delete\_course\_level(&playerD);

\*playerX->level += 1;

settextcolor(BLACK);

show\_course\_level();

setlinecolor(RED);

if (playerA.state[FAILURE])

delete\_course\_level(&playerA);

if (playerB.state[FAILURE])

delete\_course\_level(&playerB);

if (playerC.state[FAILURE])

delete\_course\_level(&playerC);

if (playerD.state[FAILURE])

delete\_course\_level(&playerD);

EndBatchDraw();

learned = true;

playerX->state[UNFAMILIAR] = false;

}

}

}

}

else

{

if (playerX->state[SKILLFUL])

{

\_stprintf\_s(guide\_text, \_T("状态：登峰造极\n学习成功"));

settextcolor(RED);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

BeginBatchDraw();

settextcolor(WHITE);

show\_course\_level();

setlinecolor(WHITE);

if (playerA.state[FAILURE])

delete\_course\_level(&playerA);

if (playerB.state[FAILURE])

delete\_course\_level(&playerB);

if (playerC.state[FAILURE])

delete\_course\_level(&playerC);

if (playerD.state[FAILURE])

delete\_course\_level(&playerD);

\*playerX->level += 1;

settextcolor(BLACK);

show\_course\_level();

setlinecolor(RED);

if (playerA.state[FAILURE])

delete\_course\_level(&playerA);

if (playerB.state[FAILURE])

delete\_course\_level(&playerB);

if (playerC.state[FAILURE])

delete\_course\_level(&playerC);

if (playerD.state[FAILURE])

delete\_course\_level(&playerD);

EndBatchDraw();

learned = true;

playerX->state[SKILLFUL] = false;

}

else

{

int energy\_cost = courseX->energy\_cost[\*playerX->level];

if (playerX->state[EASIER])

{

\_stprintf\_s(guide\_text, \_T("状态：轻松学习"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

if (energy\_cost > 1)

energy\_cost /= 2;

}

if (playerX->state[UNFAMILIAR])

{

\_stprintf\_s(guide\_text, \_T("状态：生疏"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

energy\_cost += 3;

}

playerX->energy -= energy\_cost;

pay(playerX, &playerA, courseX);

pay(playerX, &playerB, courseX);

pay(playerX, &playerC, courseX);

pay(playerX, &playerD, courseX);

failure(playerX);

if (playerX->state[FAILURE])

return false;

\_stprintf\_s(guide\_text, \_T("学习成功"));

settextcolor(RED);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

BeginBatchDraw();

settextcolor(WHITE);

show\_course\_level();

setlinecolor(WHITE);

if (playerA.state[FAILURE])

delete\_course\_level(&playerA);

if (playerB.state[FAILURE])

delete\_course\_level(&playerB);

if (playerC.state[FAILURE])

delete\_course\_level(&playerC);

if (playerD.state[FAILURE])

delete\_course\_level(&playerD);

\*playerX->level += 1;

settextcolor(BLACK);

show\_course\_level();

setlinecolor(RED);

if (playerA.state[FAILURE])

delete\_course\_level(&playerA);

if (playerB.state[FAILURE])

delete\_course\_level(&playerB);

if (playerC.state[FAILURE])

delete\_course\_level(&playerC);

if (playerD.state[FAILURE])

delete\_course\_level(&playerD);

EndBatchDraw();

learned = true;

playerX->state[UNFAMILIAR] = false;

}

}

}

else

{

if (courseX->forget)

{

\_stprintf\_s(guide\_text, \_T("该课程已达到最高等级，具有遗忘机制\nL：复习（消耗3点精力，其他键放弃复习，课程等级-1）"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

input = \_getch();

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

if ((input == 'l' || input == 'L') && playerX->energy > 3) //按下“L”键进行复习，其他键放弃复习

{

\_stprintf\_s(guide\_text, \_T("复习成功"));

settextcolor(RED);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

playerX->energy -= 3;

}

else

{

\_stprintf\_s(guide\_text, \_T("未复习"));

settextcolor(RED);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

BeginBatchDraw();

settextcolor(WHITE);

show\_course\_level();

setlinecolor(WHITE);

if (playerA.state[FAILURE])

delete\_course\_level(&playerA);

if (playerB.state[FAILURE])

delete\_course\_level(&playerB);

if (playerC.state[FAILURE])

delete\_course\_level(&playerC);

if (playerD.state[FAILURE])

delete\_course\_level(&playerD);

\*playerX->level -= 1;

settextcolor(BLACK);

show\_course\_level();

setlinecolor(RED);

if (playerA.state[FAILURE])

delete\_course\_level(&playerA);

if (playerB.state[FAILURE])

delete\_course\_level(&playerB);

if (playerC.state[FAILURE])

delete\_course\_level(&playerC);

if (playerD.state[FAILURE])

delete\_course\_level(&playerD);

EndBatchDraw();

}

}

}

return learned;

}

/\*课程学习函数\*/

bool learn(struct player\* playerX, struct course\* courseX)

{

playerA.level = &courseX->playerA\_level;

playerB.level = &courseX->playerB\_level;

playerC.level = &courseX->playerC\_level;

playerD.level = &courseX->playerD\_level;

switch (playerX->name)

{

case 'A':

return cost(&playerA, courseX);

break;

case 'B':

return cost(&playerB, courseX);

break;

case 'C':

return cost(&playerC, courseX);

break;

case 'D':

return cost(&playerD, courseX);

break;

}

}

**6.4 课程查看函数——鼠标与键盘交互的健壮性**

/\*课程查看函数\*/

void view\_course(void)

{

BeginBatchDraw();

setlinecolor(RED);

if (playerA.state[FAILURE])

delete\_course\_level(&playerA);

if (playerB.state[FAILURE])

delete\_course\_level(&playerB);

if (playerC.state[FAILURE])

delete\_course\_level(&playerC);

if (playerD.state[FAILURE])

delete\_course\_level(&playerD);

EndBatchDraw();

\_stprintf\_s(guide\_text, \_T("鼠标右键查看课程信息\n鼠标左键跳过查看"));

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

while (true)

{

if (MouseHit())

{

MOUSEMSG m = GetMouseMsg();

if (m.uMsg == WM\_RBUTTONDOWN) //按下鼠标右键查看课程信息

{

BeginBatchDraw();

settextcolor(WHITE);

show\_course\_level();

setlinecolor(WHITE);

if (playerA.state[FAILURE])

delete\_course\_level(&playerA);

if (playerB.state[FAILURE])

delete\_course\_level(&playerB);

if (playerC.state[FAILURE])

delete\_course\_level(&playerC);

if (playerD.state[FAILURE])

delete\_course\_level(&playerD);

settextcolor(BLACK);

show\_course\_information();

EndBatchDraw();

}

else if (m.uMsg == WM\_RBUTTONUP) //松开鼠标右键显示课程等级

{

BeginBatchDraw();

settextcolor(WHITE);

show\_course\_information();

settextcolor(BLACK);

show\_course\_level();

setlinecolor(RED);

if (playerA.state[FAILURE])

delete\_course\_level(&playerA);

if (playerB.state[FAILURE])

delete\_course\_level(&playerB);

if (playerC.state[FAILURE])

delete\_course\_level(&playerC);

if (playerD.state[FAILURE])

delete\_course\_level(&playerD);

EndBatchDraw();

}

else if (m.uMsg == WM\_LBUTTONUP) //鼠标左键跳过查看

{

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK); //清空操作指导区域

break;

}

else

FlushMouseMsgBuffer();

}

if (\_kbhit())

\_getch();

}

}

**6.5 状态持续函数——静态变量的应用**

/\*借阅函数\*/

void borrowing(struct player\* playerX)

{

static int a = borrowing\_round;

static int b = borrowing\_round;

static int c = borrowing\_round;

static int d = borrowing\_round;

if (load\_continous\_state)

{

FILE\* Borrowing = fopen("D:\\C\_Project\_Resource\\continous\_state\\Borrowing.txt", "r");

fread(&a, sizeof(int), 1, Borrowing);

fread(&b, sizeof(int), 1, Borrowing);

fread(&c, sizeof(int), 1, Borrowing);

fread(&d, sizeof(int), 1, Borrowing);

fclose(Borrowing);

}

if (playerX->state[BORROWING])

{

switch (playerX->name)

{

case 'A':

\_stprintf\_s(guide\_text, \_T("状态：借阅中（还剩%d个回合）"), a);

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

a -= 1;

if (a == -1 || playerX->position == LIBRARY && a != borrowing\_round - 1)

{

a = borrowing\_round;

playerX->state[BORROWING] = false;

if (playerX->position != LIBRARY)

playerX->gold -= 250;

failure(playerX);

}

break;

case 'B':

\_stprintf\_s(guide\_text,\_T( "状态：借阅中（还剩%d个回合）"), b);

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

b -= 1;

if (b == -1 || playerX->position == LIBRARY && b != borrowing\_round - 1)

{

b = borrowing\_round;

playerX->state[BORROWING] = false;

if (playerX->position != LIBRARY)

playerX->gold -= 250;

failure(playerX);

}

break;

case 'C':

\_stprintf\_s(guide\_text, \_T("状态：借阅中（还剩%d个回合）"), c);

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

c -= 1;

if (c == -1 || playerX->position == LIBRARY && c != borrowing\_round - 1)

{

c = borrowing\_round;

playerX->state[BORROWING] = false;

if (playerX->position != LIBRARY)

playerX->gold -= 250;

failure(playerX);

}

break;

case 'D':

\_stprintf\_s(guide\_text, \_T("状态：借阅中（还剩%d个回合）"), d);

settextcolor(BLACK);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

Sleep(1000);

settextcolor(WHITE);

settextstyle(15, 0, \_T("宋体"));

drawtext(guide\_text, &guide\_r, DT\_CENTER | DT\_WORDBREAK);

d -= 1;

if (d == -1 || playerX->position == LIBRARY && d != borrowing\_round - 1)

{

d = borrowing\_round;

playerX->state[BORROWING] = false;

if (playerX->position != LIBRARY)

playerX->gold -= 250;

failure(playerX);

}

break;

}

}

if (playerX->name == 'D')

{

FILE\* Borrowing = fopen("D:\\C\_Project\_Resource\\continous\_state\\Borrowing.txt", "w");

fwrite(&a, sizeof(int), 1, Borrowing);

fwrite(&b, sizeof(int), 1, Borrowing);

fwrite(&c, sizeof(int), 1, Borrowing);

fwrite(&d, sizeof(int), 1, Borrowing);

fclose(Borrowing);

}

}

1. **测试**

|  |  |  |
| --- | --- | --- |
| 测试项目 | 预期结果 | 实际结果 |
| 游戏性测试 | 游戏平衡性良好，无逻辑错误，在80个回合内可结束游戏，可玩性、策略性较强 | 基本达成预期结果，且在50-60回合内即可结束游戏。 |
| 健壮性测试 | 可以处理玩家的非法输入，玩家可以无限试错，玩家使用道具失败后可以重新进行回合操作，键盘和鼠标交互不会干扰。 | 达成预期结果。 |
| 图形界面测试 | 画面美观，交互性强，操作方便，文本框位置、字体和字号适宜，音乐正常播放、避免画面闪烁现象。 | 基本达成预期结果，偶尔出现的画面闪烁现象疑似为EasyX函数本身所致。 |
| 存读档测试 | 自动存档和读档功能正常。 | 达成预期结果 |



