江 苏 海 洋 大 学 计算机工程学院

验收报告

**课程名称：** Linux课程设计

**选题名称：** “基于Linux的填色游戏的设计与实现”

**成 员：** 代婷婷 2017123260

张冬玲 2017123293

秦佳宁 2017123190

徐 周 2017123325

**专业班级：** 软件171

**系 （院）： 计算机工程学院**

**设计时间：** 2019.11.18~2018.12.1

|  |
| --- |
| 指导教师评语：  **成绩：**  签名：  年 月 日 |

|  |
| --- |
| 1．课程设计目的 |
| 本课程设计是计算机科学与技术专业重要的实践性环节之一，是在学生学习完《Linux程序设计》课程后进行的一次全面的综合练习。本课程设计的目的和任务：  （1）巩固和加深学生对Linux程序设计课程的基本知识的理解和掌握；  （2）掌握基于Linux的软件开发过程中编程和程序调试的基本技能；  （3）利用Linux系统提供的API进行简单软件设计的基本思路和方法；  （4）提高在Linux环境下解决实际问题的能力；  （5）掌握书写程序设计说明文档的能力。  本次课程设计利用《Linux程序设计》课程中所学到的编程知识和编程技巧，完成具有一定难度和工作量的程序设计题目，帮助学生掌握编程、调试的基本技能，使学生能够独立完成所布置的任务。 |
| 1. 课程设计任务与要求： |
| **要求：**  （1）设计题目、要求以及系统功能需求分析；  （2）总体设计：包括模块说明、模块结构图和系统流程图；  （3）详细设计：包括主要功能模块的算法设计思路以及对应的工作流程图；  （4）源程序代码及设计成果：完整源程序清单和设计成果；  总结：调试分析过程描述：包括测试数据、测试输出结果以及对程序调试过程中存在问题的思考（列出主要问题的出错现象、出错原因、解决方法及效果等）包括设计过程中的学习体会与收获、对本次课程的认识等内容。  **任务：**  在Linux系统中，设计一个以炸弹人为原型改良后的填色游戏。  **实现功能：**  玩家可通过键盘控制角色的移动（玩家一：W、A、S、D，玩家二：I、J、K、L），并通过放置炸弹来毁坏部分墙壁，  玩家所走过的点即变为对应颜色，以达到“占地”的目的，在规定时间到达后，比较两个玩家色块数目的多少来判断输赢。  本程序实现了游戏逻辑、角色移动、放置炸弹、炸毁墙壁、色块填充、分数统计、结果分析等功能。 |
| 3．课程设计说明书 |
| （1）游戏分析  游戏主要功能：  1.从文件读取并输出对应的地图；  2.从键盘读取对应数据实现人物的移动；  3.人物移动对应路径填充色块；  4.炸弹爆炸并销毁对应障碍物；  5.时间结束后不同人物对应填充的色块数量统计，色块数量多者获胜；  （注：level2中不出现炸弹与障碍物，人物移动只有遇到墙体时停止。）  （2）课设分工  1.展示界面+场景跳转（张冬玲）  2.人物移动+路径颜色覆盖（徐周）  3.炸弹爆炸+障碍物清除+对应颜色的方格数统计+倒计时界面（代婷婷）  4.判断障碍物是否可炸+编辑报告及流程图（秦佳宁）  （3）概要设计  //获取用户输入并不回显  char getch();  //检测键盘的输入状态  int kbhit(void);  //从文件读取对应等级的地图  void ReadMap(int level);  //地图刷新，做为线程，一直刷新  void\* MapChange();  //拆毁墙壁  void BreakWall(int x, int y);  //炸弹炸墙(用于制作炸弹拆毁墙壁特效，目前还未实现)  void Bomb(int x, int y);  //分数统计，返回得分  int Score(int id);  //判断玩家对应的色块数，并输出输赢结果  void Result(int a, int b);  //主菜单  void GameMenu();  //玩家一获胜界面设计  void P1\_Win();  //玩家二获胜界面设计  void P2\_Win();  //游戏等级界面设计  void LV1();  void LV2();  //倒计时时间界面设计  void One();  void Two();  void Three();  //游戏说明界面  void GameExplain();  //炸弹填色游戏  void\* BombGame();  //滑动填色游戏  void\* SlideGame();  //玩家选择游戏等级界面  void Game(int x);  （4）详细设计：主要写各功能模块流程图、算法的设计、函数的设计、列出文件清单  各功能模块流程图：  1.主流程图设计如下：    图1 主流程图  2.游戏逻辑流程图；    图2 游戏逻辑流程图  2.地图刷新流程图；    图3 地图刷新流程图  2.炸弹爆炸流程图；    图4 炸弹爆炸流程图 |
| 4．课程设计成果 |
| 1. 代码设计：   #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <fcntl.h>  #include <pthread.h>  #include <time.h>  #include <termios.h>  #define DDL 10 //游戏计时时间  int map[15][17] = {0};  typedef struct player  {  int head;  int body;  int score;  int x;  int y;  }Player;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //获取用户输入并不回显  char getch()  {  char c;  system("stty -echo");  system("stty -icanon");  c = getchar();  system("stty -icanon");  system("stty -echo");  return c;  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //检测键盘的输入状态  int kbhit(void)  {  struct termios oldt, newt;  int ch;  int oldf;  tcgetattr(STDIN\_FILENO, &oldt);  newt=oldt;  newt.c\_lflag &= ~(ICANON | ECHO);  tcsetattr(STDIN\_FILENO, TCSANOW, &newt);  oldf=fcntl(STDIN\_FILENO, F\_GETFL, 0);  fcntl(STDIN\_FILENO, F\_SETFL, oldf | O\_NONBLOCK);  ch=getchar();  tcsetattr(STDIN\_FILENO, TCSANOW, &oldt);  fcntl(STDIN\_FILENO, F\_SETFL, oldf);  if(ch != EOF)  {  ungetc(ch, stdin);  return 1;  }  return 0;  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  void ReadMap(int level)  {  int i=0,j=0;  FILE \*fp;  if(level == 1)  {  fp = fopen("map.txt","r");  }  else if(level == 2)  {  fp = fopen("map2.txt","r");  }  //打开文件  if(!fp)  {  printf("地图读取失败\n");  }  //将地图数据读到数组map  for(i=0; i<15; i++)  {  for(j=0; j<17; j++)  {  fscanf(fp,"%d",&map[i][j]);  }  }  //关闭文件  close(fp);  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //地图刷新，做成线程，一直刷新  void\* MapChange()  {  //获取时间  time\_t tPast = -1;  time\_t tNow = -1;  time(&tPast); //获取线程开始时的系统时间  time(&tNow);  int i=0,j=0,k=0;  //不停地刷新，直到DDL  while((tNow-tPast) <= DDL)  {  //获取当前系统时间  time(&tNow);  //计算时间  k = DDL+1 - tNow + tPast;  //清屏  system("clear");  printf("\n\n");  //图形化  for(i=0; i<15; i++)  {  printf(" ");  for(j=0; j<17; j++)  {  switch(map[i][j])  {  case 0: printf(" ");break;  case 1: printf(" \033[42m○\033[0m ");break; //Player1\_Body  case 11:printf(" \033[32m●\033[0m ");break; //Player1\_Head  case 2: printf(" \033[44m○\033[0m ");break; //Player2\_Body  case 12:printf(" \033[34m●\033[0m ");break; //Player2\_Head  case 3: printf(" \033[41m \033[0m ");break; //BombEffect  case 5: printf(" \033[31m★\033[0m ");break; //Bomb  case 4: printf(" \033[33m□\033[0m ");break; //Wall can be broken  case 9: printf("\033[47m \033[0m");break; //Wall can not be broken  }  }  if(i==0)  {  printf("\033[34m -------------------\033[0m");  }  else if(i==1)  {  printf("\033[34m | |\033[0m");  }  else if(i==2)  {  printf("\033[34m | 倒计时 |\033[0m");  }  else if(i==3)  {  if(k > 9)  {  printf("\033[34m | %d |\033[0m", k);  }  else  {  printf("\033[34m | %d |\033[0m", k);  }  }  else if(i==4)  {  printf("\033[34m -------------------\033[0m");  }  else if(i==5)  {  printf("\033[34m | |\033[0m");  }  else if(i==6)  {  printf("\033[34m | player1移动: |\033[0m");  }  else if(i==7)  {  printf("\033[34m | w |\033[0m");  }  else if(i==8)  {  printf("\033[34m | a s d |\033[0m");  }  else if(i==9)  {  printf("\033[34m | player2移动: |\033[0m");  }  else if(i==10)  {  printf("\033[34m | i |\033[0m");  }  else if(i==11)  {  printf("\033[34m | j k l |\033[0m");  }  else if(i==12)  {  printf("\033[34m | 在规定时间内， |\033[0m");  }  else if(i==13)  {  printf("\033[34m |占方格数目多者胜 |\033[0m");  }  else if(i==14)  {  printf("\033[34m -------------------\033[0m");  }  printf("\n");  }  }  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //拆毁墙壁  void BreakWall(int x, int y)  {  //上  if(map[x-1][y] == 4)  {  map[x-1][y] = 0;  }  //下  if(map[x+1][y] == 4)  {  map[x+1][y] = 0;  }  //左  if(map[x][y-1] == 4)  {  map[x][y-1] = 0;  }  //右  if(map[x][y+1] == 4)  {  map[x][y+1] = 0;  }  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //炸弹炸墙  void Bomb(int x, int y)  {  //放置炸弹  map[x][y] = 5;  //炸弹特效  //拆墙  BreakWall(x, y);  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //分数统计，返回得分  int Score(int id)  {  int score = 0;  int i=0, j=0;  for(i=0; i<15; i++)  {  for(j=0; j<17; j++)  {  if(map[i][j] == id)  {  score++;  }  }  }  return score;  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  void Result(int a, int b)  {  if(a>b)  {  P1\_Win();  }  else  {  P2\_Win();  }  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //主菜单  void GameMenu()  {  int i;  while(!kbhit())  {  for(i=0; i<=7; i++)  {  system("clear");  printf("\n\n");  printf(" \033[35m-------------------------------------------------------------\033[0m\n");  printf(" \033[35m| |\033[0m\n");  printf(" \033[35m|\033[3%dm \* \* \* \* \* \* \* \* \* \* \* \* \* \033[35m|\033[0m\n",i);  printf(" \033[35m|\033[3%dm \* \* \* \* \* \* \* \* \* \* \033[35m|\033[0m\n",i);  printf(" \033[35m|\033[3%dm \* \* \* \* \* \* \* \* \* \* \033[35m|\033[0m\n",i);  printf(" \033[35m|\033[3%dm \* \* \* \* \* \* \* \* \* \033[35m|\033[0m\n",i);  printf(" \033[35m|\033[3%dm \* \* \* \* \* \* \* \* \* \* \* \033[35m|\033[0m\n",i);  printf(" \033[35m|\033[3%dm \* \* \* \* \* \* \* \* \* \033[35m|\033[0m\n",i);  printf(" \033[35m|\033[3%dm \* \* \* \* \* \* \* \* \033[35m|\033[0m\n",i);  printf(" \033[35m|\033[3%dm \* \* \* \* \* \* \* \* \033[35m|\033[0m\n",i);  printf(" \033[35m|\033[3%dm \* \* \* \* \* \* \* \* \* \* \* \033[35m|\033[0m\n",i);  printf(" \033[35m| |\033[0m\n");  printf(" \033[35m| |\033[0m\n");  printf(" \033[35m|\033[33m 游戏名：占地为王 \033[35m|\033[0m\n");  printf(" \033[35m| |\033[0m\n");  printf(" \033[35m|-----------------------------------------------------------|\033[0m\n");  printf(" \033[35m|\033[33m 1.开始游戏 2.游戏说明 3.退出游戏 \033[35m|\033[0m\n");  printf(" \033[35m-------------------------------------------------------------\033[0m\n");  printf("\n 请输入您的选择！ \n");  }  }  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //玩家获胜界面  void P1\_Win()  {  system("clear");  int i;  while(!kbhit())  {  for(i=0;i<=7;i++)  {  system("clear");  printf("\n\n");  printf(" \033[3%dm ------------------------------------------------------------------------ \n\033[0m",i);  printf("\n");  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf("\n");  printf(" \033[3%dm ------------------------------------------------------------------------ \n\033[0m",i);  }  }  }  void P2\_Win()  {  system("clear");  int i;  while(!kbhit())  {  for(i=0;i<=7;i++)  {  system("clear");  printf("\n\n");  printf(" \033[3%dm ----------------------------------------------------------------------- \n\033[0m",i);  printf("\n");  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf(" \033[3%dm | \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* |\n\033[0m",i);  printf("\n");  printf(" \033[3%dm ----------------------------------------------------------------------- \n\033[0m",i);  }  }  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //游戏等级界面、倒计时时间界面  void LV1()  {  system("clear");  printf("\n\n\n");  printf(" \033[33m \* \* \* \* \n\033[0m");  printf(" \033[33m \* \* \* \n\033[0m");  printf(" \033[33m \* \* \* \* \* \n\033[0m");  printf(" \033[33m \* \* \n\033[0m");  printf(" \033[33m \* \* \* \* \n\033[0m");  printf(" \033[33m \* \* \n\033[0m");  printf(" \033[33m \* \* \* \* \n\033[0m");  printf(" \033[33m \* \* \n\033[0m");  printf(" \033[33m \* \* \* \* \* \* \* \* \* \* \* \n\033[0m");  }  void LV2()  {  system("clear");  printf("\n\n\n");  printf(" \033[35m \* \* \* \* \* \* \n\033[0m");  printf(" \033[35m \* \* \* \n\033[0m");  printf(" \033[35m \* \* \* \* \n\033[0m");  printf(" \033[35m \* \* \n\033[0m");  printf(" \033[35m \* \* \* \* \n\033[0m");  printf(" \033[35m \* \* \n\033[0m");  printf(" \033[35m \* \* \* \* \n\033[0m");  printf(" \033[35m \* \* \n\033[0m");  printf(" \033[35m \* \* \* \* \* \* \* \* \* \* \* \n\033[0m");  }  void One()  {  system("clear");  printf("\n\n");  printf(" --------------- \n");  printf(" | |\n");  printf(" | |\n");  printf(" | \*\* |\n");  printf(" | \* \* |\n");  printf(" | \* |\n");  printf(" | \* |\n");  printf(" | \* |\n");  printf(" | \* |\n");  printf(" | \* |\n");  printf(" | \* |\n");  printf(" | \* |\n");  printf(" | \*\*\*\*\*\*\* |\n");  printf(" | |\n");  printf(" --------------- \n");  }  void Two()  {  system("clear");  printf("\n\n");  printf(" --------------- \n");  printf(" | |\n");  printf(" | |\n");  printf(" | \*\*\* |\n");  printf(" | \* \* |\n");  printf(" | \* \* |\n");  printf(" | \* |\n");  printf(" | \* |\n");  printf(" | \* |\n");  printf(" | \* |\n");  printf(" | \* |\n");  printf(" | \*\*\*\*\*\*\* |\n");  printf(" | |\n");  printf(" | |\n");  printf(" --------------- \n");  }  void Three()  {  system("clear");  printf("\n\n");  printf(" --------------- \n");  printf(" | |\n");  printf(" | \*\*\*\* |\n");  printf(" | \* \* |\n");  printf(" | \* |\n");  printf(" | \* |\n");  printf(" | \* |\n");  printf(" | \*\*\*\*\* |\n");  printf(" | \* |\n");  printf(" | \* |\n");  printf(" | \* |\n");  printf(" | \* \* |\n");  printf(" | \*\*\*\* |\n");  printf(" | |\n");  printf(" --------------- \n");  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //游戏说明界面  void GameExplain()  {  system("clear");  printf("\n\n");  printf(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf(" \* 游戏说明 \*\n");  printf(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf(" \* \*\n");  printf(" \* 在游戏时间结束之后，所占颜色格数多的玩家获胜 \*\n");  printf(" \* \*\n");  printf(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //炸弹填色游戏  void\* BombGame()  {  //读入地图  ReadMap(1);  Player p1,p2;  //player1，属性设置  p1.head = 11;  p1.body = 1;  p1.score = 0;  p1.x = 1;  p1.y = 1;  //player2  p2.head = 12;  p2.body = 2;  p2.score = 0;  p2.x = 13;  p2.y = 15;  //玩家头部初始位置  map[p1.x][p1.y] = p1.head;  map[p2.x][p2.y] = p2.head;  //设定游戏时间  time\_t tPast = -1;  time\_t tNow = -1;  //获取线程调用时的系统时间  time(&tPast);  time(&tNow);  //玩家移动  int i=0,j=0;  char x = 'a';  while(x != '0' && (tNow-tPast) <= DDL)  {  //获取用户输入  x = getch();  switch(x)  {  //player1  case 'w':  if(map[p1.x-1][p1.y]<4)  {  map[p1.x][p1.y] = p1.body;  p1.x = p1.x-1;  map[p1.x][p1.y] = p1.head;  }  break;  case 's':  if(map[p1.x+1][p1.y]<4)  {  map[p1.x][p1.y] = p1.body;  p1.x = p1.x+1;  map[p1.x][p1.y] = p1.head;  }  break;  case 'a':  if(map[p1.x][p1.y-1]<4)  {  map[p1.x][p1.y] = p1.body;  p1.y = p1.y-1;  map[p1.x][p1.y] = p1.head;  }  break;  case 'd':  if(map[p1.x][p1.y+1]<4)  {  map[p1.x][p1.y] = p1.body;  p1.y = p1.y+1;  map[p1.x][p1.y] = p1.head;  }  break;  case 'x':  Bomb(p1.x, p1.y);  break;    //player2  case 'i':  if(map[p2.x-1][p2.y]<4)  {  map[p2.x][p2.y] = p2.body;  p2.x = p2.x-1;  map[p2.x][p2.y] = p2.head;  }  break;  case 'k':  if(map[p2.x+1][p2.y]<4)  {  map[p2.x][p2.y] = p2.body;  p2.x = p2.x+1;  map[p2.x][p2.y] = p2.head;  }  break;  case 'j':  if(map[p2.x][p2.y-1]<4)  {  map[p2.x][p2.y] = p2.body;  p2.y = p2.y-1;  map[p2.x][p2.y] = p2.head;  }  break;  case 'l':  if(map[p2.x][p2.y+1]<4)  {  map[p2.x][p2.y] = p2.body;  p2.y = p2.y+1;  map[p2.x][p2.y] = p2.head;  }  break;  case 'm':  Bomb(p2.x, p2.y);  break;  }  //获取最新系统时间  time(&tNow);  }  //结果结算  p1.score = Score(p1.body);  p2.score = Score(p2.body);  sleep(2);  Result(p1.score, p2.score);  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //滑动填色游戏  void\* SlideGame()  {  //读入地图  ReadMap(2);  Player p1,p2;  //player1  p1.head = 11;  p1.body = 1;  p1.score = 0;  p1.x = 1;  p1.y = 1;  //player2  p2.head = 12;  p2.body = 2;  p2.score = 0;  p2.x = 13;  p2.y = 15;  //玩家头部初始位置  map[p1.x][p1.y] = p1.head;  map[p2.x][p2.y] = p2.head;  //设定游戏时间  time\_t tPast = -1;  time\_t tNow = -1;  //获取线程调用时的系统时间  time(&tPast);  time(&tNow);  //玩家移动  int i=0, j=0;  char x = 'a';  while(x != '0' && (tNow-tPast) <= DDL)  {  //获取用户输入  x = getch();  switch(x)  {  //player1  case 'w':  while(map[p1.x-1][p1.y]<4)  {  map[p1.x][p1.y] = p1.body;  p1.x = p1.x-1;  map[p1.x][p1.y] = p1.head;  }  break;  case 's':  while(map[p1.x+1][p1.y]<4)  {  map[p1.x][p1.y] = p1.body;  p1.x = p1.x+1;  map[p1.x][p1.y] = p1.head;  }  break;  case 'a':  while(map[p1.x][p1.y-1]<4)  {  map[p1.x][p1.y] = p1.body;  p1.y = p1.y-1;  map[p1.x][p1.y] = p1.head;  }  break;  case 'd':  while(map[p1.x][p1.y+1]<4)  {  map[p1.x][p1.y] = p1.body;  p1.y = p1.y+1;  map[p1.x][p1.y] = p1.head;  }  break;  //player2  case 'i':  while(map[p2.x-1][p2.y]<4)  {  map[p2.x][p2.y] = p2.body;  p2.x = p2.x-1;  map[p2.x][p2.y] = p2.head;  }  break;  case 'k':  while(map[p2.x+1][p2.y]<4)  {  map[p2.x][p2.y] = p2.body;  p2.x = p2.x+1;  map[p2.x][p2.y] = p2.head;  }  break;  case 'j':  while(map[p2.x][p2.y-1]<4)  {  map[p2.x][p2.y] = p2.body;  p2.y = p2.y-1;  map[p2.x][p2.y] = p2.head;  }  break;  case 'l':  while(map[p2.x][p2.y+1]<4)  {  map[p2.x][p2.y] = p2.body;  p2.y = p2.y+1;  map[p2.x][p2.y] = p2.head;  }  break;  }  //获取最新系统时间  time(&tNow);  }  //结果结算  p1.score = Score(p1.body);  p2.score = Score(p2.body);  sleep(2);  Result(p1.score, p2.score);  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  void Game(int x)  {  //定义两个线程号  pthread\_t update;  pthread\_t gm1;  pthread\_t gm2;  //创建线程  pthread\_create(&update, NULL, MapChange, NULL); //地图刷新  if(x == 1)  {  //炸弹填色游戏  pthread\_create(&gm1, NULL, BombGame, NULL);  pthread\_join(update, NULL);  pthread\_join(gm1, NULL);  }  else if(x == 2)  {  //滑动填色游戏  pthread\_create(&gm2, NULL, SlideGame, NULL);  pthread\_join(update, NULL);  pthread\_join(gm2, NULL);  }  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  int main()  {  char p = '1';  //初始主菜单  GameMenu();  //用户输入选择  p = getch();  //游戏逻辑  while(p != '3')  {  //选择游戏阶段  switch(p)  {  case '1':  //第一关界面、倒计时  LV1();  sleep(2);  Three();  sleep(1);  Two();  sleep(1);  One();  sleep(1);  Game(1); //从游戏1开始  getch();  LV2();  sleep(2);  Three();  sleep(1);  Two();  sleep(1);  One();  sleep(1);  Game(2);  p = '0';  break;  case '2':  GameExplain();  p = getch();  break;  case '3':  system("clear");  //直接结束游戏  p = '3';  break;  default:  GameMenu();  p = getch();  break;  }  }  printf("\n\n 感谢您的使用！\n");  return 0;  }  （2）界面展示    图4 游戏菜单    图5 游戏说明界面    图6 游戏倒计时界面    图7 第一关跳转界面    图8 第一关初始化游戏地图    图9 第一关游戏场景    图10 第二关跳转界面    图11 第二关初始化游戏地图    图12 第二关游戏场景    图13 玩家一获胜界面    图14 玩家二获胜界面    图15 游戏结束界面  **参考文献：**  [1]岳 浩《Linux操作系统教程》机械工业出版社 2005.4  [2]柳 青《Linux应用教程》北京：清华大学出版社 2008  [3]浦 滨 《C游戏编程从入门到精通（第二版）》科学出版社 2006  [4]（美）施耐德曼 《用户界面设计》 电子工业出版社 2006  [5]吴 岳 《LINUX C程序设计大全》 清华大学出版社 2009  [6]杨宗德 吕光宏 刘雍《Linux 高级程序设计（第三版）》 北京：人民教育出版社 2012 |

|  |
| --- |
| **5.课程设计心得** |
| 这次的课程设计看起来简单，但是做起来困难重重，深刻体会到做一个游戏是多么的不容易，尽管里面需要的很多知识我们都接触过，但是去图书馆找书的时候发现，我们学的仅仅是皮毛，还有很多东西需要我们去发掘，就算是借一本书看完它，我们还是会发现还有很多知识没有吃透，这需要我们不断的实践，不断地自学习，不断地发现问题去思考问题。  从确定主题的时候我们就不断地搜索资料，想游戏的逻辑及规则，最后我们参考了小时候玩过的7k7k炸弹人和填色游戏，做了这个“占地为王”填色小游戏，在开始的时候我们对要做的东西分了几个部分，一人做一部分，最后再整合到一块。  由于这门课学的不深，尤其是由于c语言库和linux库存在差异，所以我们翻阅了很多资料，经过研究探讨，不断地测试，不断地改进，从而更好的解决了这一问题。  张冬玲:在本次的课程设计中，我负责的是界面设计，本次的界面设计没有什么太大的难度，在后期添加特效或者功能时会导致原来的设计的界面出现问题。添加特效，我们主要是做了颜色的变化以及多级界面的跳转，使其更具有耐玩性，后来我们也有与老师商量解决字体投放以及颜色变化的问题，在之后会通过文件读取的方法将其改得更加完善与严谨。在本次课程设计的过程中，主要是Linux库与C语言非标准库之间的函数转变，我们也通过搜集资料完美解决了这些问题，我这次最大的收获可能就是自己第一次意义上的敲出完整代码，也更加深层地了解到了代码的奇妙所在。  秦佳宁：在本次课设中负责炸弹爆炸清除障碍物的游戏逻辑实现，这一部分的代码比较简单出的问题也不多，除了这个部分之外，还想搞出两个电脑联机玩游戏用socket实现，先是研究了实验七的代码，然后有上网百度，想把游戏逻辑代码放到服务器然后两个客户端连接从而实现联机玩游戏，试验了网上的几段代码都没有实现，然后自己先从实验七改起，想先实现简单的功能，然后就一直报错，后来没有搞出来，当然后来放弃之后，就辅助队友搞代码，界面啥的，做完此次课设我对linux的特点还有了深刻的了解，比如说linux是多用户多任务、稳定性、良好兼容性、强大的可移植性、支持多种文件操作、高效内存管理等。  代婷婷：我在这次课程设计中主要负责的是游戏的逻辑设计、游戏地图文件读取、炸弹放置、游戏计时器设计和所有代码的整合工作。我之前有过制作小游戏的经验，所以对于游戏逻辑的设计还是比较有思路的。但是我们第一次项目完成时，因为大家技术有限，很多功能并没有实现，比如游戏开始时的倒计时界面和计时器都没有完成，并且所有的功能模块是放在一个线程里完成的。经过老师的指导之后，我们将地图刷新和人物移动分为了不同的线程，这样就可以保证在刷新地图时不会有停顿感。第二次项目修改时，我们又为项目添加了一个“滑冰”填色的游戏方式，作为游戏升级的关卡。同时，整个游戏的界面设计也美化了很多，游戏计时器也添加完成并且能够显示在游戏界面上。此次设计过程中还有一个遗憾之处。我们为了使用线程而删改了一项游戏结束的判定，因为我们无法从外部终止一个正在运行的线程。后来老师告诉我们，可以通过设以一个全局变量来当作信号，通过修改全局变量的值来结束线程，而这一点是我们没有想到的。  徐周：这是第一次通过小组合作的方式来完成课程设计，本来以为小组合作会比个人完成要来的简单，但是在前期讨论的时候还是遇到了好多问题。通过本次实验，我们对Linux的系统调用有了深入的理解，通过同学们的协作，分工，大家整体完成了这次实验，实验中当然也遇到了不少的问题，但最终，我们通过查阅相关资料，大家研究探讨，最终还是解决了大部分的问题，虽然其中还是有不少的小问题，但最终，还是能够实现最终的需求，当然，改进的方式也有许多，许多地方还是需要优化，本次实验之后，我们依旧会对实验中问题部分进行优化，也加强对Linux以及，编程思想的学习，进一步加强我们的能力。 |