吉林大学

2016-2017学年第二学期

《C语言程序设计》

项目文档

第8组

|  |  |  |
| --- | --- | --- |
| 项目题目 | C语言课程实践项目——2048游戏 | |
| 组长 | | |
| 陈佩琦 | 33160131 | 理科试验班（唐敖庆数学班） |
| 组员 | | |
| 张剑桥 | 31161203 | 理科试验班（唐敖庆数学班） |
| 赵楷文 | 31160820 | 理科试验班（唐敖庆数学班） |
| 叶良婷 | 32160901 | 理科试验班（唐敖庆物理班） |
| 1. 总体设计   基本设计：  1、首先，我们决定使用二维数组和辅助数组作为我们的主要工具。  2、我们考虑到我们需要使用函数来实现：打印界面，创造随机数，实现基本操作，判断游戏是否结束这些功能。  3、在设计主函数时，我们使用while语句和switch语句来将键盘上的基本操作转变为游戏界面的改变。  进阶设计：  1、为了增加游戏的趣味性，我们设计了“存档”、“读档”和“撤回”等操作。并且当玩家达到游戏目标时，我们使得游戏在可操作的情况下可以继续下去。  2、我们使用结构体数组和指针来代替二维数组和辅助数组，重新编写了第二个版本的程序，增加一个使得结构体数组初始化的函数。  3、我们测试我们的程序来排除并修改出现的漏洞。 | | |
| 1. 主要功能和亮点 2. 从游戏到代码的实现   2048是一个十分有趣味且并不算太复杂的游戏，但是摆在我们面前的第一个难题就是如何将游戏中的一个简单的操作在代码中实现。我们需要将一些简单的操作分解为C语言能够执行的多个函数的叠加以及运用工具的具体选择。  2. 不同方法的尝试  一开始我们选择将这4\*4的方块用二维数组记录，再加上辅助数组。我们通过二维数组对辅助数组赋值，再对辅助数组操作，再反赋值于二维数组来实现对4\*4方块的改变。后来我们尝试使用结构体数组来取代辅助数组，一个结构体包含该处元素的数值以及四个指向上、下、左、右方位结构体的指针。当我们编写代码结束后确实发现结构体数组相对于二维数组有它的简洁性和便于操作性。  3.相对复杂的debug阶段  相对于其他项目而言，2048的debug阶段相对于复杂。我们在编写过程中遇到过很多意想不到的bug。有的时候编译的内容没有error，但是我们需要不停的去试玩我们的游戏以确保在每个步骤中程序能做出我们预想的反应，一旦有问题，我们就需要进入debug阶段。  4.各类游戏元素的增加  一开始我们的程序只能运行到2048，后来为了让玩家不仅限于2048，我们增加了在成功后继续游戏的操作。而且我们也设计了游戏“存档”、“读档”的操作。并且作为奖励在玩家的最大数到达一定值以后会赠送“撤回”的次数 | | |
| 1. 组员分工   陈佩琦：设计main function（1/2）  测试程序寻找修复漏洞（debug）  为程序增加函数并提供更多的操作（保存、读取、撤回），使游戏继续进行当玩家胜利后  使用结构体重写编写了原程序，制作了第二个版本  张剑桥：设计main function（1/2）  设计basic function（实现消零和合并操作）（1/2）  赵楷文：设计game\_interface function（游戏界面）  设计create\_random function（创造随机数）  叶良婷：设计judge\_fun function（包括两个函数来判断游戏是否结束）  设计basic function（实现消零和合并操作）（1/2） | | |
| 1. 程序代码使用说明   （以二维数组加上辅助数组的代码为例，后面会介绍使用结构体的不同之处）  我们使用了额外的conio.h和time.h来实现getch()函数与创造随机数  另外的我们自己编写了以下头文件  game\_interface.h 里面包含game\_interface()函数，用于打印游戏界面  create\_random.h 里面包含create\_random()函数，用于在每次操作后创建随机数  basic\_fun.h 里面包括basic\_fun\_eliminate\_zero()函数，用于对辅助数组进行消零操作  basic\_fun\_combination()函数，用于对辅助数组进行相同元素合并  basic\_fun()函数，包括一个消零函数一个合并函数再一个消零函数  judge\_fun.h 里面包括judge\_1\_fun\_1()和judge\_1\_fun\_2()函数，用于判断是否可以进行  “上、下、左、右”的操作  包括judge\_2\_fun()函数，用于判断4\*4的位置上有没有零元素  system.h 里面包括save()、load()和regret\_save()、regret\_load()函数，用于实现存档、读档以及撤回的操作  主程序中：  一开始执行初始化语句后进入while循环，首先执行清屏和打印界面的操作再进如switch函数，有上、下、左、右、存档、读档、后悔的操作可执行。退出Switch函数后，进行判断函数judge\_1\_fun()和judge\_2\_fun()若两个判断都不能通过则跳出循环进入一个if语句：若16个元素的最大数小于2048，游戏失败；若大于等于2048，游戏胜利，此时玩家可以选择继续游戏，重新进入while循环，在通过两个判断的前提下，玩家可以无限的将游戏进行下去，否则游戏结束。  （对于结构体代码而言，我们多写了一个initialization.h的头文件，包含一个initialization  ()的函数用于初始化结构体数组，由于结构体的优势我们将主函数里的对上、下、左、右的操作都写进了basic\_fun.h这个头文件使主函数更加简洁，当然对其他的头文件中的函数都有不同程度的化简） | | |
| 1. 主要代码   （结构体的basic\_fun头文件中的函数）  int basic\_fun\_eliminate\_zero\_up(void)  {  int i;  int temp=0;  for(i=0;i<=15;i++)  {  if((array[i].up!=NULL)&&(array[i].up->number==0)&&(array[i].number!=0))  {  array[i].up->number=array[i].number;  array[i].number=0;  temp=1;  judge\_3=1;  }  }  return temp;  }  void basic\_fun\_combination\_up(void)  {  int i;  for(i=0;i<=15;i++)  {  if((array[i].up!=NULL)&&(array[i].number!=0)&&(array[i].up->number==array[i].number))  {  array[i].up->number\*=2;  score+=array[i].up->number;  array[i].number=0;  judge\_3=1;  }  if((array[i].up!=NULL)&&(max<(array[i].up->number)))  max=(array[i].up->number);  }  }  （结构体代码main函数中的循环语句）  while(max<WIN)  {  system("cls");  game\_interface();  judge\_4=max;  operation\_1=getch();  switch(operation\_1)  {  case 'w':  case 'W':  regret\_save();  basic\_fun\_up();  break;  （'s' 'S' 'a' 'A' 'd' 'D' case 相同）  case 'o':  save();  printf("you've already saved\n");  getch();  break;  case 'p':  load();  printf("you've already loaded\n");  getch();  break;  case 'l':  if(regret>0)  {  regret\_load();  printf("you've already regreted\n");  regret-=1;  }  else  {  printf("sorry that you can't regret\n");  }  getch();  break;  }  judge\_1=judge\_1\_fun();  judge\_2=judge\_2\_fun();  if((judge\_4!=max)&&(max>=1024))  regret+=1;  if((judge\_2==0)&&(judge\_1==0))  break;  } | | |
| 1. 程序运行截图 2. 游戏界面   C:\Users\18672\AppData\Local\Microsoft\Windows\INetCache\Content.Word\捕获2.png  2、游戏保存  C:\Users\18672\AppData\Local\Microsoft\Windows\INetCache\Content.Word\捕获1.png  3、游戏载入C:\Users\18672\AppData\Local\Microsoft\Windows\INetCache\Content.Word\捕获5.png  4、游戏后悔C:\Users\18672\AppData\Local\Microsoft\Windows\INetCache\Content.Word\捕获4.png  5、游戏失败C:\Users\18672\AppData\Local\Microsoft\Windows\INetCache\Content.Word\捕获6.png  6、游戏成功C:\Users\18672\AppData\Local\Microsoft\Windows\INetCache\Content.Word\捕获3.png | | |
| 1. 收获与感悟   从最初组建小组开始，大家都抱着不断学习的态度来决心做好我们的项目——2048小游戏。开始我们通过在网上找的代码更加深入得理解了这个游戏的工作原理以及代码的运用，在这个过程中，也了解到了一些新的函数比如清屏函数，以及对二维数组的应用；在简单修改了游戏，向老师汇报进展后，我们小组讨论一致认为需要更多的功能来增加游戏的趣味性，并且改用更复杂的结构体数组来实现游戏。  发现问题以及改进问题对刚学c语言的我们来说，有点困难。不断地分析别的代码，查找并理解不认识的函数，在整体框架中调整函数，判断执行中的问题，我们发现在这些过程中我们可以更熟练的运用c语言基础知识，掌握了更多的函数，在最终看见游戏成功运行时的自豪感和喜悦不言而喻。在实践项目的合作过程中，除了更深入学习了c语言知识外，也收获了友谊和认可，大家相互帮助，相互督促，最终才完全实现了游戏的成功。  思考和不断实践应用是我们学习过程中的好老师。每个项目在不断修改后才可能更完美，尝试不同的方法也是一种学习。合作中同学们的水平可能稍有不同，相互体谅与帮助，才能更好地完成项目。 | | |
| 1. 意见与建议 2. 在提前一周讲解时老师尽量慢一点，重点问题应该多重复几遍。 3. 讲解时突出重点，防止一天下来一头雾水的情况发生。 4. 在布置项目时应给出一些学长学姐的例子以供参考。 5. 可以给学生推荐一些网路视频教程。 6. 多深入了解同学的实际需求。 | | |