Universe Scientific Publishing
Computer System Network and Telecommunications,计算机系统网络和电信(1) 2019,3
ISSN:2661-3719(Print) ISBN:2661-3727(Online)

基于C语言的推箱子游戏的设计与开发

何宇 段华琼
四川大学锦城学院 计算机与软件学院 四川 成都 611731

DOI:10.18686/jsjxt.v1i3.1256

【摘要】推箱子游戏作为一款经典的益智小游戏,广泛应用于Windows系统。本文通过对推箱子游戏的分析,详细阐述了用C语言编写推箱子游戏的过程,对游戏设计过程中的重点和难点进行了详细的分析,并详细介绍了其主要函数。

【关键词】游戏开发;推箱子游戏;C语言

1 引言

推箱子,作为一款老少皆宜的益智游戏,主要考验了操作者思维的灵活性和逻辑思考能力。C语言是一种能以简易的方式编译,仅产生少量的机器码以及不需要任何运行环境支持便能运行的编程语言。本文利用C语言实现了一个推箱子游戏,希望通过此简单实例,给广大游戏开发者一些设计思路和灵感。

2 游戏简介

推箱子游戏最初起源于一个古老的日本游戏,要求在一个狭小的仓库中,把木箱放到指定的位置,且游戏过程中稍不小心就会出现箱子无法移动或者通道被堵住的情况,所以需要在有限的空间,和有限的时间限制的情况下,将规定的箱子放置于目标位置。

推箱子游戏本身属于地图小游戏的一种,所以需要利用二维数组初始化地图map,并在地图上利用随机算法布置出墙壁(游戏人物和箱子均无法通过的障碍物)。游戏目标为在游戏规定步数内,操作者通过键盘控制人物移动方向,最终使游戏人物成功将箱子移动到指定位置。

3 设计方案

本游戏设计主要利用模块化设计的思想,通过主函数调用各个自定义函数来完成,增强了程序的可建设性。

本游戏的主要设计内容包括:

• 游戏界面设计,包括对窗体、人物、障碍物等的设计。

• 游戏功能的实现,包括实现人物移动、人物图片更换和箱子移动等。

• 逻辑判断处理,如通过判断二维数组中的数据来判断人物移动的情况。

• 判断是否闯关成功。

3.1 开始界面设计

游戏的欢迎界面介绍了游戏规则,同时展现友好的游戏界面,有利于提高游戏的趣味性,彰显代码的魅力。

实现开始界面的主要代码如下:

```
void welcome()
{
  printf("* * * * * * * * * * *\n");
  printf(" Welcome! \n");
  printf("键盘输入人物移动方向上下左右来控制箱子移动\n");
  printf("把箱子移动到指定位置即可过关 \n");
  printf("* * * * * * * * * * *\n");
}
```

3.2 游戏界面设计

推箱子作为一个经典且简易的黑白游戏,其界面布局也十分简单,以墙壁、空地、箱子以及人物构成。其界面设计可参考网状地图,把整个游戏地图看做一个二维数组,利用二维数组布局,在i行j列的游戏区域随机分布无法通过的墙壁、人物以及箱子等。在把整个游戏看作一个二维数组之后,利用i*j次循环,随机生成1~5的随机数字,并把不同数字相对应的区域标识为墙壁、箱子等。

本文根据实例,利用二维数组确定一个8*8的一个地图模型,其主要代码如下:

```
int map[8][8] = {
{1,1,1,1,1,1,1,1},      //0代表空地
{1,0,0,0,1,0,0,1},      //1代表墙(箱子与人物均无法通过)
```

ISSN: 2661-3719(Print) ISBN:2661-3727(Online)

```
    {1,0,1,0,1,4,3,1},      //3 代表箱子所要到达的目的地
    {1,0,0,0,1,4,3,1},      //4 代表箱子
    {1,0,1,0,1,4,3,1},      //5 代表游戏人物
    {1,0,0,0,1,0,0,1 },
    {1,1,1,1,1,5,0,1},
    {1,1,1,1,1,1,1,1},
    //绘制地图
    void DrawMap()
    {
      for (int i = 0; i < 8; i++)
      {
        for (intj = 0; j < 8; j++)
        {
          switch (map[i][j])
          {
          case 0:
            printf("  ");
            break;
          case 1:
            printf("■");
            break;
          case 3:
            printf("☆");
          break;
          case 4:
            printf("□");
            break;
          case 5:
            printf("♀");
            break;
          case 7:
            printf("★");      //箱子已到达目的地,二者重合
            break;
          case 8:
             printf("♀");      //人物与目的地重合
            break;
        }
      }
          printf("\n");
        }
    }
```

上述代码先设置了初始化地图的大小,将不同的数转换成其代表标志,描绘了一张比较标准且美观的地图。可以根据各个游戏的不同的布局需求,在以上代码的基础上进行适当的修改和完善。

3.3 确定人物位置

推箱子游戏是通过操作游戏人物移动而进行的,在进行了地图初始化的步骤之后,为了保证游戏有效且有效率地进行,还需要在一开始确定人物所在位置。本文依靠嵌套循环,利用上述二维数组的数组元素,确立人物位置。

主要代码如下:

```
int a,b;
void People()
{
  for (i = 0; i < 8; i++)
   {
    for (j = 0; j < 8; j++)
    {
      if (Map[i][j] == 5 || Map[i][j] == 8)//表示人物的两种标识
        {
          a=i;
          b=j;
        }
    }
    //for 循环结束后 Map[i][j]就是人物的位置
  }
}
```

3.4 游戏操作设计

开始游戏后,游戏界面展示的是一个封闭的地图,需要操控人物来移动以获得游戏的胜利。

游戏规则设计如下:

(1)人物移动方向为上下左右,其对应标志为W、S、A、D;

(2)要判断人前面(上一行),是否为空地,是空地则可以移动,移动后判断并改变人员原位置的数组元素和空地原位置的数组元素,否则不移动;

(3)如果人前面是目的地,人也是可以移动的,同样判断人原位置数组元素,并改变人员位置的数组元素和前一位置的数组元素;

(4)如果人前是在空地上的箱子。如果箱子前面又是空地,则可以移动,改变箱子前方空地位置的数组元素,还是判断原位置的数组元素,然后改之;

(5)如果人前是已经进入目的地的箱子。如果箱子前又是空地,则可以移动箱子,但是要减分,改变空地位置的数组元素,判断人和箱子原地的数组元素,并改之;如果箱子前是另一目的地,方法同上;

ISSN: 2661-3719(Print) ISBN:2661-3727(Online)

(6)当人或箱子就在目标地,退出来时候要回复原来目标地的标志。

(7)不能同时推动两个(以上)箱子,人物和箱子可以同时移动。

每一次的移动,必须要清屏,然后再显示一次地图,这样便可以实现人、人和箱子的移动。故当初始化结束后,设计函数判断横纵坐标的改变,来表示人物的具体移动。

例如,控制人物向左移动,如果人物的移动方向的下一个位置的数据是 4,代表其前面有 1 个箱子存在。如果箱子的下一个位置的数字不等于 1 或 4,代表其前面没有障碍物和箱子。

主要代码如下:

```
void Game()
{
  char ch;  //字符变量
  ch = getch();  //键盘的输入保存到字符中
  switch (ch)
{
  case 'W': //用户可输入大写,小写,ASCII 三值来表示其移动方向
  case 'w':
  case 72:
    if (map[a - 1][b] == 0|| map[a - 1][b] == 3) //上方是空地或者是目的地
    {
      map[a - 1][b] += 5;
      map[a][b] -= 5;
    }
    else if (map[a - 1][b] == 4 || map[a - 1][b] == 7)
    //人的上方是箱子或是箱子加目的地

    {
      if (map[a - 2][b] == 0 || map[a - 2][b] == 3)
      //箱子上方是空地  可以推动
      {
        map[a - 2][b] += 4;
        map[a - 1][b] += 1;
        map[a][b] -= 5;
      }
    }
    break;
  case 'S':
  case 's':
  case 80:
    if (map[a + 1][b] == 0 || map[a + 1][b] == 3)
    {
      map[a + 1][b] += 5;
      map[a][b] -= 5;
    }
    else if (map[a + 1][b] == 4 || map[a + 1][b] == 7)
    {
      if (map[a + 2][b] == 0 || map[a + 2][b] == 3)
      {
        map[a + 2][b] += 4;
        map[a + 1][b] += 1;
        map[a][b] -= 5;
      }
    }
    break;
  case 'A':
  case 'a':
  case 75:
    if (map[a ][b - 1] == 0 || map[a ][b - 1] == 3)
    {
      map[a ][b - 1] += 5;
      map[a][b] -= 5;
    }
    else if (map[a][b - 1] == 4 || map[a][b - 1] == 7)
    {
      if (map[a ][b - 2] == 0 || map[a ][b - 2] == 3)
      {
        map[a ][b - 2] += 4;
        map[a ][b - 1] += 1;
        map[a][b] -= 5;
      }
    }
    Break;
  case 'D':
  case 'd':
  case 77:
    if (map[a][b + 1] == 0 || map[a][b + 1] == 3)
```

ISSN：2661－3719(Print)　ISBN：2661－3727(Online)

```
//向右移箱子，且箱子右边是目的地
{
  map[a][b + 1] += 5;
  map[a][b] -= 5;
}
else if (map[a][b + 1] == 4 || map[a][b + 1] == 7)
{
  if (map[a][b + 2] == 0 || map[a][b + 2] == 3)
  {
    map[a][b + 2] += 4;
    map[a][b + 1] += 1;
    map[a][b] -= 5;
  }
}
break;
}
}
```

3.5 移动函数

为了保持游戏的正常进行与美观，随着人物每一次移动，画面随之变动，调用头文件 stdlib.h 中的清屏函数，清空上一步图层，使最新图层覆盖，实现动态移动效果。

其主要代码如下：

```
void move(int map[8][8]) //定义移动函数
{
system("CLS"); //调用清屏函数
printf("*****************************************\n");
}
```

3.6 游戏胜利

每一个游戏都需要一个胜利条件。根据推箱子游戏的规则，需要判定最后箱子所在位置。如前面所述，用户操纵的是人物，故借助人物所在位置可间接得到箱子位置。知晓箱子所处位置后，结合下面的函数，与数组元素中的数字对比，判断箱子是否到达目的地。

其主要代码如下：

```
void GameWin()
{
for (int i = 0; i < 8; i++)
{
  for (int j = 0; j < 8; j++)
  {
    if (Map[i][j] == 4)
    {
      People4++;
    }
    if (Map[i][j] == 7)
    {
      people7++;
    }
  }
}
if (people4 == people7)
{
    return 0;
      printf("恭喜您，游戏胜利！");
}
else
{
    return 1;
      printf("真遗憾，失败了呢，再来一次吧。");
}
}
```

4 结束语

本文用 C 语言按照结构化程序设计的原则实现了推箱子游戏，展示了用 C 语言开发游戏软件的过程，对 C 语言程序设计教学与实践具有一定的思维启发作用。C 语言是计算机程序设计的基础，而计算机技术的发展，对机械设备的设计工作产生了深远的影响。希望通过本文对推箱子游戏的设计，能对初学程序者提供一些编写程序的思路。

【参考文献】

[1]谭浩强. C 语言程序设计 [M]. 北京：清华大学出版社，2018 年
[2]陈慧杰，郭占祥. 基于 C 语言的五子棋游戏程序设计[J]. 宁波职业技术学院学报，2012 年
[3]马寅璞，孔阳坤. 用 JAVA 实现一个推箱子游戏[J]. 技术与市场，2019 年
[4]俞亮. 基于 C# 的扫雷游戏设计与实现[j]. 企业技术开发，2014 年
[5]徐驰. 基于 C 语言的扫雷游戏设计与实现. [j]. 信息与电脑，2018 年