

C 语言扫雷小程序概要设计

2011-12-06
xu_ke@bupt.edu.cn

目录

扫雷小程序概要设计.....	1
1.数据结构设计.....	2
1.1 地雷分布矩阵 Bomb	2
1.2 周边地雷计数矩阵 Count	2
1.3 显示矩阵 Display	3
2.主要函数设计.....	3
2.1 void displayGrids()	3
2.2 void initBomb()	3
2.3 void initCount()	4
2.4 void initDisplay()	4
2.5 void init()	4
2.6 void digGrid(int x, int y)	5
2.7 void signGrid(int x, int y)	5
2.8 int checkGameState()	5
2.9 void main()	5

1.数据结构设计

使用三个二维数组（可以改成动态分配大小的数组，这里固定大小）

```
int Bomb[10][10] = {0}; //地雷分布矩阵
int Count[10][10] = {0}; //周边地雷计数矩阵
char Display[10][10] = {0}; //显示字符的矩阵
```

1.1 地雷分布矩阵 Bomb

`Bomb[10][10]`表示格子的哪个位置上有雷，如下图矩阵，`Bomb[3][3] == 1` 和 `Bomb[4][4]==1` 则说明这两个位置有雷。可以使用随机数函数生成随机数 `x` 和 `y`，然后在 `Bomb[x][y]`的位置上布雷。

[illegible]

1.2 周边地雷计数矩阵 Count

Count[10][10]表示每个格子周围的地雷总数,根据 Bomb 数组可以计算出 Count 数组的取值。下面是 (1) 中 Bomb 矩阵对应的 Count 矩阵

[illegible]

1.3 显示矩阵 Display

Display[10][10]表示格子的每个位置上应该显示什么符号，这是一个字符二维数组，根据每次挖开的位置，以及 Count 数组和 Bomb 数组计算得出。

例如，可以定义几个字符作为该数组的取值：

‘#’ 表示格子未翻开时的显示

‘1’~‘8’ 表示翻开格子后没有雷，字符显示的周边雷数

‘*’ 表示翻开格子后有雷的显示

‘ ’（空格）表示翻开格子后周边没有雷时的显示

‘!’表示该位置被标识的显示

2.主要函数设计

2.1 void displayGrids()

功能：在屏幕上显示当前的格子矩阵

设计：

```
void displayGrids()
{
    二层循环，遍历 Display 数组
    {
        printf(Display[i][j]);
        如果一行结束
        printf("\r\n");
    }
}
```

2.2 void initBomb()

功能：初始化 Bomb 矩阵，随机布雷

设计

```
void initBomb()
{
    while(i<雷的总数)
    {
        随机生成 x,y
        如果 x,y 的位置已经有雷
            continue;
    }
}
```

```

        Bomb[x][y] = 1;
        i++;
    }
}

```

2.3 void initCount()

功能：根据 Bomb 矩阵，初始化 Count 矩阵

设计：

```

void initCount()
{
    二层循环，遍历 Bomb[][]
    {
        如果 Bomb[][] 的(i,j)位置有雷
        {
            for( (m,n) 是 (i,j) 周围的 8 个格子)
            {
                Count[m][n] ++;
            }
        }
    }
}

```

2.4 void initDisplay()

功能：初始化显示矩阵

设计：

数组初始化成 ‘#’

2.5 void init()

功能：发起各矩阵的初始化，在开始时调用

设计：

```

void init()
{
    initBomb();
    initCount();
    initDisplay();
}

```

2.6 void digGrid(int x, int y)

功能：挖开一个指定坐标的格子

参数：

x 矩阵的行坐标

y 矩阵的列坐标

void digGrid(int x, int y)

```
{
    如果 Bomb[x][y]位置有雷
        Display[x][y] = '*'
    否则
        {
            如果 Count[x][y]是 0
                递归的翻开(x,y)周围的格子
            否则
                根据 Count[x][y]的取值给 Display[x][y]赋值
        }
}
```

2.7 void signGrid(int x, int y)

功能：标识一个指定坐标的格子

参数：

x 矩阵的行坐标

y 矩阵的列坐标

2.8 int checkGameState()

功能：检查所有的格子，看游戏是否结束

返回值：

1 表示成功结束

2 表示失败结束

0 表示游戏继续

自己设计

2.9 void main()

功能：主函数

设计：

```
void main()
{
    初始化各矩阵 init()
    显示格子矩阵 displayGrids()

    循环等待用户输入
    {
        提示用户输入操作字符;
        switch(输入的操作字符)
        {
            case 挖雷操作:
                提示, 输入坐标 x, y;
                digGrid(x, y);
                break;
            case 标识操作
                .....
            case 退出操作
                .....
        }

        显示矩阵 displayGrids()
        获得状态 checkGameState()

        如果成功结束
            则打印提示信息, 退出
        如果失败结束
            则打印提示信息, 退出
        如果未结束
            则继续循环
    }
}
```