

C++实验报告三：类与对象

实验目的

- 1.掌握基本的类与对象使用方法
- 2.掌握基本的交互页面
- 3.掌握基本的随机数使用方法
- 4.掌握基本的分支语句的使用
- 5.熟练运用数组和结构体

实验内容

实验：类和对象

请设计一个精灵类**Spirit**，其成员要求如下：

1.具有private的data members

- (1) **name**，精灵的名字，键盘输入
- (2) **x**，精灵的位置坐标X轴的值，键盘输入
- (3) **y**，精灵的位置坐标Y轴的值，键盘输入
- (4) **health**，精灵的**health**生命力初值设为1000
- (5) **aliveState**，精灵的生存状态，初值设为**true**

2.具有静态数据成员

- (1) **number**，记录已经创建的精灵数量，初值为0
- (2) **alivenumber**，记录生存状态为**true**的精灵数量，初值为0

3.具有的public的member function

- (1) 构造函数1，无参数的构造函数，创建精灵对象并初始化，**number+1**，**alivenumber+1**
- (2) 构造函数2，有参数的构造函数（精灵的名字和位置坐标），创建对象并初始化，**number+1**，**alivenumber+1**
- (3) **takeDamage**函数：实现对精灵的伤害，首先需要判断**aliveState**的值，如果是**true**，将**health - damage**，若伤害后的精灵**health<=0**，则**health**设为0，将**aliveState**设为**false**，**alivenumber-1**
- (4) **setPosition**函数：需要判断新位置是否在**0-200**之间，如果是的话，就设置精灵新位置，否则报错，仍保持原位置，并输出错误提示信息。
- (5) **getPositionX**函数：返回精灵的位置坐标X
- (6) **getPositionY**函数：返回精灵的位置坐标Y
- (7) **getName**函数：返回精灵的名字
- (8) **getHealth**函数：返回精灵的生命值
- (9) **getNumber**函数：返回精灵世界一共有多少精灵
- (10) **getAliveNumber**函数：返回精灵世界一共有多少**alive**的精灵
- (11) **getInfo**函数：输出精灵的各种信息，包括姓名、生命值、生存状态、坐标等

4.main函数功能

- (1) 在main函数中通过不同的构造函数创建不同精灵。
- (2) 调用**takeDamage**函数，测试精灵的生存状态。

- (3) 测试类中定义的所有函数。比如设置新位置，输出精灵的各种信息等。
- (4) 设置对象数组管理10个精灵并初始化
- (5) 用随机数模拟10次随机事件，即对精灵*i*的*k*点伤害，输出相关信息，随机事件结束后，输出所有对象的基本信息

5. 选做任务

- (1) 尝试在屏幕上用符号（比如“*”）代表精灵，以精灵坐标为屏幕的行和列，在屏幕上“画”出所有精灵
- (2) 对应各种事件在屏幕画中展现相应变化，比如新增精灵，精灵消亡，精灵移动
- (3) 添加精灵事件，任何你希望精灵具备的技能

实验方案

```
//by dpcc
//精灵类来袭
#include<iostream>
#include<algorithm>
#include<string>
#include<random>
#include<ctime>
using namespace std;
using ll = long long;
#define rep(i,a,n) for(int i=a;i<=n;i++)
#define frep(i,a,n) for(int i=a;i>=n;i--)
ll maze[300][300];
struct Node
{
    ll x, y;
};
struct jineng
{
    string name;
    ll gongji;
};
ll cnt_number = 0;
class Spirit
{
public:
    jineng m_jineng[10];
    ll jineng_number = 0;
    static ll number;//创建过的精灵
    static ll alivernumber;//存活的精灵
    //无参构造函数
    Spirit()
    {
        number++;
        alivernumber++;
    }
    //有参构造函数
    Spirit(string m_name, Node m_node)
    {
        name = m_name;
        node.x = m_node.x;
        node.y = m_node.y;
```

```

        maze[node.x][node.y] = 1;
        number++;
        alivernumber++;
    }
    void setallDetail()
    {
        string m_name;
        Node m_node;
        cout << "请输入精灵的名字" << '\n';
        cin >> m_name;
        cout << "请输入精灵的横坐标和纵坐标" << '\n';
        cout << "请注意x,y坐标全部控制在10以内的整数" << '\n';
        cin >> m_node.x >> m_node.y;
        name = m_name;
        node.x = m_node.x;
        node.y = m_node.y;
        maze[node.x][node.y] = 1;
    }
    void takeDamage(int damage)
    {
        if (!aliveState)
        {
            cout << "很不幸，该精灵已经失败了" << '\n' << '\n';
        }
        else
        {
            if (health - damage > 0)
            {
                health -= damage;
                cout << name << "成功承受住了攻击" << '\n';
            }
            else
            {
                health = 0;
                aliveState = false;
                alivernumber -= 1;
            }
        }
    }
}
void setPosition(Node m_node)
{
    if (node.x >= 0 && node.y >= 0 && node.x <= 200 && node.y <= 200 &&
!maze[m_node.x][m_node.y])
    {
        maze[node.x][node.y] = 0;
        node.x = m_node.x;
        node.y = m_node.y;
        maze[node.x][node.y] = 1;
    }
    else
    {
        cout << "该位置非法" << '\n';
    }
}
}

```

```

11 getPointx()
{
    return node.x;
}
11 getPointy()
{
    return node.y;
}
string getName()
{
    return name;
}
11 getHealth()
{
    return health;
}
11 getNumber()
{
    return number;
}
11 getAlivernumber()
{
    return alivernumber;
}
void printmyspirit()
{
    cout << "名字是" << name << '\n';
    cout << "生命值是" << health << '\n';
    if (aiveState)
    {
        cout << "该精灵仍伴你左右" << '\n';
    }
    else
    {
        cout << "该精灵已经离你而去" << '\n';
    }
    cout << "-----" << '\n';
}
void m_showhealth()
{
    cout << name << "精灵的血量为" << health << '\n';
}
void add()
{
    cout << "请输入添加的技能的名称" << '\n';
    cin >> m_jineng[++jineng_number].name;
    cout << "请输入添加技能的攻击点数" << '\n';
    cin >> m_jineng[jineng_number].gongji;
    cout << "技能汇入完毕" << '\n';
}
void showjineng()
{
    rep(i, 1, jineng_number)
    {
        cout << name << "的第" << i << "个技能是" << m_jineng[i].name << '\n';
    }
}

```

```

        cout << "该技能可对其他精灵造成" << m_jineng[i].gongji << "点伤害" << '\n';
        cout << "-----" << '\n';
    }
}

private:
    //五个初始值实现
    Node node; //点, 包含x,y
    string name;
    ll health = 1000;
    bool aliveState = true;
};

ll Spirit::number = 0;
ll Spirit::alivernumber = 0;
void show()
{
    cout << "-----" << '\n';
    cout << "输入1代表创建你的精灵" << '\n';
    cout << "输入2代表攻击你的精灵" << '\n';
    cout << "输入3代表查看你的精灵" << '\n';
    cout << "输入4查看每只精灵的位置" << '\n';
    cout << "输入5为精灵添加技能" << '\n';
    cout << "输入6展示精灵技能" << '\n';
    cout << "输入7可选择两个精灵进行比武" << '\n';
    cout << "输入8可选择退出本款小游戏" << '\n';
    cout << "-----" << '\n';
}

int main()
{
    std::default_random_engine e;
    std::uniform_int_distribution<int> u(1, 1000); // 左闭右开区间
    e.seed(time(0));
    cout << "欢迎来到精灵国度" << '\n' << '\n';
    cout << "尽情的开始创建精灵吧" << '\n' << '\n';
    //一些基础的调试
    //无参构造函数的调用
    //Spirit first();
    //有参构造函数的调用
    /*Node second_x_y;
    second_x_y.x = 80;
    second_x_y.y = 40;
    Spirit second("康娜", second_x_y);
    cout << '\n';
    //成员函数的调用
    second.printmyspirit();
    //1.攻击并判断受到攻击后是否存活
    second.takeDamage(18);
    //2.获得存活精灵的数量
    ll alivernumber = second.getAlivernumber();
    //3.获得精灵的名字
    string name = second.getName();
    //4.获得精灵的健康指数
    ll health = second.getHealth();
    //5.获得精灵的x坐标
    ll x = second.getPointx();
    //6.获得精灵的y坐标

```

```

    ll y = second.getPointy();
    //7.获得创建过的精灵的数量
    ll number = second.getNumber();
    //8.重设精灵的位置
    Node second_set_x_y;
    second_set_x_y.x = 20;
    second_set_x_y.y = 20;
    second.setPosition(second_set_x_y);
    //9.打印出精灵所有的参数
    second.printmyspirit();
    */
    //第一种对象数组的创建
    /*
    Spirit jingling[10] = { //定义对象数组
        Spirit("汤姆",node), //调用构造函数
        Spirit("杰瑞",node), //调用构造函数
        Spirit("加菲猫",node), //调用构造函数
        Spirit("球球鼠",node), //调用构造函数
        Spirit("笑猫",node), //调用构造函数
        Spirit("虎皮猫",node), //调用构造函数
        Spirit("黑猫",node), //调用构造函数
        Spirit("白猫",node), //调用构造函数
        Spirit("黑旋风",node), //调用构造函数
        Spirit("红猪",node) //调用构造函数
    };
    */
    Spirit jingling[1000] = {};
    while (1)
    {
        show();
        int cnt;
        cin >> cnt;
        switch (cnt)
        {
            case 1:
            {cout << "可以开始创建你的精灵了" << '\n';
            jingling[++cnt_number].setalldetail();
            cout << "创建完毕" << '\n';
            system("pause");
            system("cls");
            break;
            }
            case 2:
            {cout << "想攻击你创建的第几只精灵呢" << '\n';
            ll n;
            cin >> n;
            cout << "选择你攻击的点数" << '\n';
            ll damage;
            cin >> damage;
            jingling[n].takeDamage(damage);
            system("pause");
            system("cls");
            break;
            }
            case 3:
            {

```

```

        int n;
        cout << "想查看哪知精灵的全部属性呢? " << '\n';
    cin >> n;
    if (n > cnt_number)
    {
        cout << "该精灵没有被创建过" << '\n';
    }
    else {
        cout << "这是" << n << "号精灵的信息: " << '\n';
        jingling[n].printmyspirit();
    }
    system("pause");
    system("cls");
    break;
}
case 4:
{cout << "*号表示有精灵驻守" << '\n';
rep(i, 1, 10)
{
    rep(j, 1, 10)
    {
        if (maze[i][j])
        {
            cout << '*';
        }
        else
        {
            cout << '#';
        }
    }
    cout << '\n';
}
system("pause");
system("cls");
break;
}
case 5:
{
    int n;
    cout << "想为哪一只精灵添加技能呢" << '\n';
    cin >> n;
    jingling[n].add();
    system("pause");
    system("cls");
    break;
}
case 6:
{
    cout << "想查看哪个精灵的技能呢" << '\n';
    int chakan;
    cin >> chakan;
    jingling[chakan].showjineng();
    system("pause");
    system("cls");
    break;
}
}

```

```

case 7:
{
    cout << "接下来是两个精灵的战斗" << '\n';
    cout << "输入进入角斗场的精灵选手" << '\n';
    bool flag = 0;
    int x, y;
    cin >> x >> y;
    jingling[x].m_showhealth();
    jingling[y].m_showhealth();
    jingling[x].showjineng();
    jingling[y].showjineng();
    int n_x = jingling[x].jineng_number;
    int n_y = jingling[y].jineng_number;
    string name1 = jingling[x].getName();
    string name2 = jingling[y].getName();
    for (int i = 1; i <= max(n_x, n_y); i++)
    {
        jingling[y].takeDamage(jingling[x].m_jineng[i].gongji);
        cout << name2 << "受到来自" << name1 << "的" <<
jingling[x].m_jineng[i].name << "的攻击" << '\n';
        ll m_health1 = jingling[x].getHealth();
        ll m_health2 = jingling[y].getHealth();
        if (m_health2 == 0)
        {
            flag = 0;
            cout << name1 << "获得比赛胜利" << '\n';
            break;
        }
        else if (m_health1 == 0)
        {
            flag = 0;
            cout << name2 << "获得比赛胜利" << '\n';
            break;
        }
        jingling[x].takeDamage(jingling[y].m_jineng[i].gongji);
        cout << name1 << "受到来自" << name2 << "的" <<
jingling[y].m_jineng[i].name << "的攻击" << '\n';
        m_health1 = jingling[x].getHealth();
        m_health2 = jingling[y].getHealth();
        if (m_health2 == 0)
        {
            flag = 0;
            cout << name1 << "获得比赛胜利" << '\n';
            break;
        }
        else if (m_health1 == 0)
        {
            flag = 0;
            cout << name2 << "获得比赛胜利" << '\n';
            break;
        }
    }
    if (flag)
    {
        cout << "两个精灵打成了平局" << '\n';
    }
}

```



```

        system("pause");
        system("cls");
        break;
    }
    case 8:
    {
        cout << "本游戏由爱笑的龙猫出品" << '\n';
        cout << "感谢体验" << '\n';
        exit(0);
    }
    default:
    {
        system("pause");
        system("cls");
        break;
    }
}

/*
rep(i, 0, 9)
{
    Node node;
    string m_name;
    cout << "请输入第"<<i+1<<"只精灵的名字和x, y坐标" << '\n';
    cin >> m_name;
    cin >> node.x;
    cin >> node.y;
    jingling[i].setalldetaili(m_name, node);
}
rep(i, 0, 9)
{
    ll damage = u(e);
    jingling[i].takeDamage(damage);
}
rep(i, 0, 9)
{
    jingling[i].printmyspirit();
}
/*
rep(i, 0, 300)
{
    rep(j, 0, 300)
    {
        if (maze[i][j])
        {
            cout << "*";
        }
        else
        {
            cout << ' ';
        }
    }
    cout << '\n';
}

```

```
*/  
    return 0;  
}
```

输出结果

欢迎来到精灵国度

尽情的开始创建精灵吧

输入1代表创建你的精灵
输入2代表攻击你的精灵
输入3代表查看你的精灵
输入4查看每只精灵的位置
输入5为精灵添加技能
输入6展示精灵技能
输入7可选择两个精灵进行比武

输入8可选择退出本款小游戏

1
可以开始创建你的精灵了
请输入精灵的名字
Tom
请输入精灵的横坐标和纵坐标
请注意x,y坐标全部控制在10以内的整数
5
5
创建完毕

输入1代表创建你的精灵
输入2代表攻击你的精灵
输入3代表查看你的精灵
输入4查看每只精灵的位置
输入5为精灵添加技能
输入6展示精灵技能
输入7可选择两个精灵进行比武

输入8可选择退出本款小游戏

1
可以开始创建你的精灵了
请输入精灵的名字
nick
请输入精灵的横坐标和纵坐标
请注意x,y坐标全部控制在10以内的整数

6

6

创建完毕

输入1代表创建你的精灵

输入2代表攻击你的精灵

输入3代表查看你的精灵

输入4查看每只精灵的位置

输入5为精灵添加技能

输入6展示精灵技能

输入7可选择两个精灵进行比武

输入8可选择退出本款小游戏

1

可以开始创建你的精灵了

请输入精灵的名字

hugua

请输入精灵的横坐标和纵坐标

请注意x,y坐标全部控制在10以内的整数

7

1

输入1代表创建你的精灵

输入2代表攻击你的精灵

输入3代表查看你的精灵

输入4查看每只精灵的位置

输入5为精灵添加技能

输入6展示精灵技能

输入7可选择两个精灵进行比武

输入8可选择退出本款小游戏

2

想攻击你创建的第几只精灵呢

1

选择你攻击的点数

12

Tom成功承受住了攻击

输入1代表创建你的精灵

输入2代表攻击你的精灵

输入3代表查看你的精灵

输入4查看每只精灵的位置

输入5为精灵添加技能

输入6展示精灵技能

输入7可选择两个精灵进行比武

输入8可选择退出本款小游戏

2

想攻击你创建的第几只精灵呢

1

选择你攻击的点数

12

Tom成功承受住了攻击

输入1代表创建你的精灵

输入2代表攻击你的精灵

输入3代表查看你的精灵

输入4查看每只精灵的位置

输入5为精灵添加技能

输入6展示精灵技能

输入7可选择两个精灵进行比武

输入8可选择退出本款小游戏

3

想查看哪知精灵的全部属性呢？

1

这是1号精灵的信息：

名字是Tom

生命值是988

该精灵仍伴你左右

输入1代表创建你的精灵

输入2代表攻击你的精灵

输入3代表查看你的精灵

输入4查看每只精灵的位置

输入5为精灵添加技能

输入6展示精灵技能

输入7可选择两个精灵进行比武

输入8可选择退出本款小游戏

4

号表示有精灵驻守

#####

#####

#####

#####

#####

#####*#####

*#####

#####

#####

#####

输入1代表创建你的精灵
输入2代表攻击你的精灵
输入3代表查看你的精灵
输入4查看每只精灵的位置
输入5为精灵添加技能
输入6展示精灵技能
输入7可选择两个精灵进行比武

输入8可选择退出本款小游戏

5
想为哪一只精灵添加技能呢
1
请输入添加的技能的名称
火焰流星拳
请输入添加技能的攻击点数
100
技能汇入完毕

输入1代表创建你的精灵
输入2代表攻击你的精灵
输入3代表查看你的精灵
输入4查看每只精灵的位置
输入5为精灵添加技能
输入6展示精灵技能
输入7可选择两个精灵进行比武

输入8可选择退出本款小游戏

6
想查看哪个精灵的技能呢
1
Tom的第1个技能是火焰流星拳
该技能可对其他精灵造成100点伤害

输入1代表创建你的精灵
输入2代表攻击你的精灵
输入3代表查看你的精灵
输入4查看每只精灵的位置
输入5为精灵添加技能
输入6展示精灵技能
输入7可选择两个精灵进行比武

输入8可选择退出本款小游戏

5
想为哪一只精灵添加技能呢
2
请输入添加的技能的名称
爱的抱抱

请输入添加技能的攻击点数

10000

技能汇入完毕

输入1代表创建你的精灵

输入2代表攻击你的精灵

输入3代表查看你的精灵

输入4查看每只精灵的位置

输入5为精灵添加技能

输入6展示精灵技能

输入7可选择两个精灵进行比武

输入8可选择退出本款小游戏

6

想查看哪个精灵的技能呢

2

nick的第1个技能是爱的抱抱

该技能可对其他精灵造成10000点伤害

输入1代表创建你的精灵

输入2代表攻击你的精灵

输入3代表查看你的精灵

输入4查看每只精灵的位置

输入5为精灵添加技能

输入6展示精灵技能

输入7可选择两个精灵进行比武

输入8可选择退出本款小游戏

7

接下来是两个精灵的战斗

输入进入角斗场的精灵选手

1

2

Tom精灵的血量为1000

nick精灵的血量为1000

Tom的第1个技能是火焰流星拳

该技能可对其他精灵造成1000点伤害

nick的第1个技能是爱的抱抱

该技能可对其他精灵造成10000点伤害

Tom获得比赛胜利

输入1代表创建你的精灵

输入2代表攻击你的精灵

输入3代表查看你的精灵

输入4查看每只精灵的位置

输入5为精灵添加技能
输入6展示精灵技能
输入7可选择两个精灵进行比武

输入8可选择退出本款小游戏

8
本游戏由爱笑的龙猫出品
感谢体验

实验心得

经过该实验，我对随机数的用法的理解更深一层。同时，困扰了我已久的类与对象也开始有了些许眉目，我开始对C++面向对象的编程思想有了一定的体悟，同时在书写代码的时候，我对结构体，数组和一些分支循环结构体悟更深。

实验日期

11月24日