

本图说地图元素关系；

字母+数字：地区 s 海洋地区，h 高地

蓝线，有的一座坝

深蓝线：一座坝连接海洋 s1

褐色线：空的一座坝；

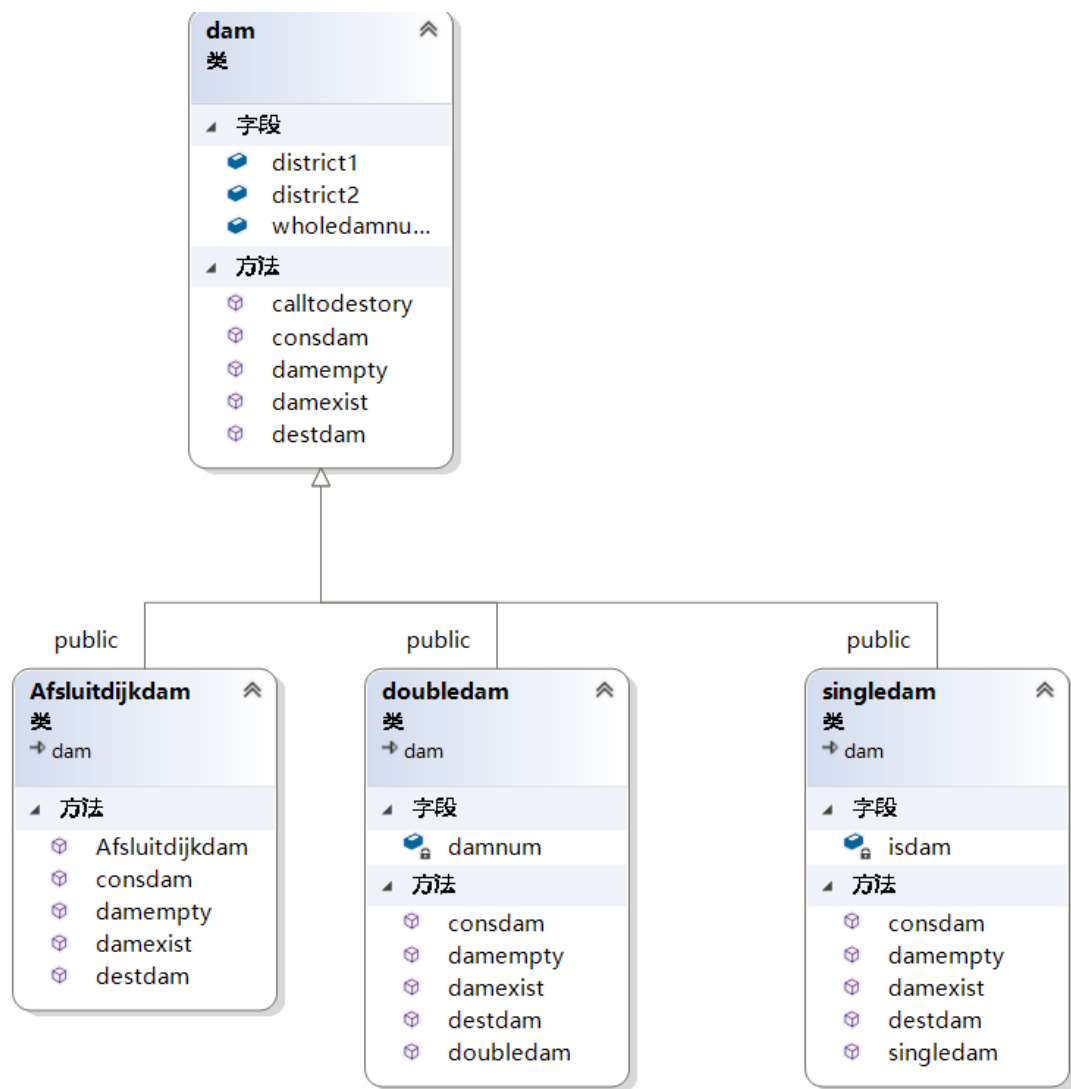
红线：两座坝

地区数字编号

S1—0，o1-o7——1—7，g1—g7——8—14，y1—y7——15—21，p1—p7——

22—28，s2—29

UML



```

string district1;
string district2;    //储存大坝两端地区
static int wholedamnumber; //剩余大坝总数

```

其他 dam 类函数为纯虚函数

singledam 类描述只有一座坝位的大坝；（图上蓝线和褐色线）

doubledam 类描述两座坝位置的大坝；（图上红线）

virtual void consdam();建造大坝，并且维护水流流动条件，并且在

virtual void destdam();摧毁大坝，并且维护水流流动条件

virtual bool damexist();返回是否空

virtual bool damempty();返回是否有

singledam(const string &a, const string& b, const bool& c)构造函数

doubledam(const string& a, const string& b, const int& c)

对 Afsluitdijkdam：一座特殊的坝

建成以后该坝不能摧毁，

建设条件是 water 类的函数 `static void Afsluitdijkdam();`//阿夫鲁戴克大

water 类	district 类	people 类
<div>字段</div> <div><div>h20ocube</div><div>isAfsluitdijkdam</div><div>isdeltaengine</div><div>isriverbuffer</div><div>isstandardeng...</div><div>protec</div><div>waterlevel</div></div> <div>方法</div> <div><div>Afsluitdijkdam</div><div>deltaengine</div><div>drain</div><div>inundation</div><div>isfail</div><div>issuccess</div><div>oceanheight</div><div>renew</div><div>riverbuffer</div><div>setbegin</div><div>setwetpath</div><div>standardengine</div><div>waterflow</div></div>	<div>字段</div> <div><div>fac</div><div>facnum</div><div>hnear</div><div>index</div><div>port</div><div>portnum</div><div>stricname</div><div>waterheight</div></div> <div>方法</div> <div><div>district (+ 1 重...</div><div>isfacconstruct</div><div>isportconstruct</div><div>operator=</div><div>pathjudge</div></div>	<div>字段</div> <div><div>card</div><div>career</div><div>didian</div><div>ind</div><div>movepoint</div><div>peoplenum</div></div> <div>方法</div> <div><div>move</div><div>movepjudgu</div><div>movepoin</div><div>movpoi</div><div>people (+ 1 ...</div><div>renew</div><div>special</div></div>

坝

Water 类：主要的函数功能类

变量：关于游戏四个主要目标是否实现与是否失败，静态，

私有 `static bool isstandardengine;` 标准化工程是否建成

`static bool isriverbuffer;` 河流缓冲区是否建成

`static bool isdeltaengine;` 三角洲工程是否建成

`static bool isAfsluitdijkdam;` 阿夫鲁戴克大坝是否建成

`static int protec;` 保护地区的编号，默认为1000，500指跳过普通的水流动

公有 `static int h20ocube;`//剩余水方块

`static int waterlevel;`//全局水位

函数

`static void setwetpath();`//重置wetpath

`static void setbegin();`//初始化，设定数组damarray和低海拔地区联通判

```

据vector<list<string>>hpaph;
static int oceanheight();将waterlevel换算成2, 3, 4大级
static void standardengine(int set = 4);//标准化工程：黄色大地区放置
至多4个（默认值）大坝
    static void riverbuffer(int set = 6);//河流缓冲带：绿色大地区排除
至多6个（默认值）水方块
    static void deltaengine();//三角洲工程
    static void Afsluitdijkdam();//阿夫鲁戴克大坝
    static bool issuccess();判定是否成功
    static bool isfail();判定是否失败
    static void waterflow();普通的水流动函数
    static void inundation(const string & orign);orign是洪水流动起始
地区，函数用来结算洪水流动
    static void renew();//每回合开始前更新荷兰男孩，water::protec变
量
    static void drain();//调用排水厂

```

district 类：描述不同地区

```

static int portnum; //总剩余码头
static int facnum; //总剩余排水厂
int waterheight; 地区水位
char stricname; 地区大地区名
int index; 地区编号
vector <string> near; //人的行动，判定人之间是否可行动，用于
pathjudg ( )
vector <string> hnear; //坝位置
bool port; //是否有码头
bool fac; //是否有排水厂
bool isportconstruct(); //用于建设地区码头，针对码头建筑数量有限，
不建设返回0，建设成功返回1
bool isfacconstruct(); //类似上述
district(char a, int n, vector <string> ne, vector <string>
hne, int id) : stricname(a), waterheight(n), port(0), fac(0) //构造函数

district(const district& t) : waterheight(t.waterheight),
stricname(t.stricname), port(t.port), fac(t.fac), near(t.near),

```

```

hnear(t.hnear)//复制构造
    district() = default;//默认构造，默认的地区晚些时候有通过
diqupanding ( string , district ) 赋值
    district &operator =(const district &tm);//重载等于的复制，用于
diqupanding ( string , district ) 赋值
    bool pathjudge(string a);//判断两地是否相邻；
people 类：描述玩家
int movepoint;//活动点数，一回合一人四个
public:
    static int peoplenum;//总角色人数
    static void special();//特殊活动，使用特殊卡牌或者后门
    int career;//职业
    int ind;//编号，用于string allpeople[5]获得名字
    people(int n):ind(n) { }构造
    people() =default;//默认构造， 随后inline bool
isnotplayer(string& a, people& c, const int& k);赋值
    void renew() {回合开始重载活动点数
        movepoint = 4;
    }
    bool movepjugu();//活动点数是否到0

    void move(const int & b);//花费活动点数的活动，b=1—11，用来决定
活动类型
    list<string>card;//人的卡牌
    string didian;//所在地区
    void movpoi();//活动点数减1，利用了movepoin ( -1 )
    void movepoin(int n); //活动点数加 n

```

设计难点

一、相邻水系判断

广度搜索算法

用维护邻接结构

vector<list<int>> hpath;//由大坝维护

vector<list<int>> wetpath;//每次水流动后，由void water：

setwetpath()维护，遍历hpath，将其中联通又相互有水的写入，判断联通的

水系

`bool issearch[30] = 0;`用于描述是否到达

用在`bool bfs_chain(const string a, const string b)` //单纯判断是否两地联通，排水厂用

`void water::inundation(const string &origin)`//连锁结算洪水

二、普通的水流动可以连锁流入

`void water::waterflow()`//由于，比如相邻的3—0—0变成3—2—1，所以利用`bool issearch[30]`计数水位 ≥ 2 的地区，然后遍历两次（由于水的上限是3）

三、不同职业不同职业技能

函数指针，

`bool (*carfun)(people & p);`//职业技能的函数指针
`switch (this->career)`

应对职业技能

```
bool portmanager(people& p) { ... }
bool warehousemanager(people& p) { ... }
bool monitor(people& p) { ... }
bool carpenter( people& p) { ... }
bool pumpworker(people& p) { ... }
bool waterengineer(people &p) { ... }
bool medicalworker(people &p) { ... }
bool nothing(people& p) { ... }
```

四、卡牌操作

洗牌使用`stl`中`random_shuffle`

打出某卡牌使用`auto de=find (a.begin(),a,end(),value)`

当`(de!= a,end())`用`erase`删去卡牌。

五，字符串，数字和类的联系

`if`语句嵌套，靠传入`district`类或者`people`类或者`int`（对应大坝指针数组

damarray坐标)的引用,判断

六、大坝不同座数

利用继承基类dam,并使用dam* damarray[62]指针数组,可以统一处理大坝

七、随机数

srand(unsigned(time(0))); 传入系统时间做种子

```
double random(double start, double end)
{
    return start + (end - start) * rand() / (RAND_MAX + 1.0);
}
```

用除法,生成随机区间的数

八、异常处理

针对整数

```
do {
    cin.sync(); //清空流
    cin.clear(); //清除流错误标记
    cin >> peoplenum; //cin输入到int i, 类型不符, 超长将设置cin.fail为1
    if (cin.fail() || peoplenum > 6 || peoplenum < 2)
        cout << "输入错误, 请重新输入" << endl; //出错显示信息
} while (cin.fail() || peoplenum > 6 || peoplenum < 2);
```

针对district类名

```
bool raincome() { ... }
inline bool isnotdistrict(string& a, district& c)
{
    bool is = 0;
    do {
        cout << "输入地区名\n";
        cin >> a;
        diqupanding(a, c, is);
        if (!is)
        {
            cout << "错误, 地区不存在\n" << "是否返回 (按q停止, 选择其他地区按其他键) \n";
            cin >> a;
            if (a == "q")
                return 1;
        }
    } while (!is);
    return 0;
}
```

其他同理