普通游戏

名称：普通游戏

要素：卡片、攻防

卡片的定义：

struct card{//卡片属性面板

string name;//卡片名字

int cardtype;//卡片类型

int cardattack\_wu;//卡片物理攻击

int cardattack\_fa;//卡片法术攻击

int carddefend\_wu;//卡片物理防御

int carddefend\_fa;//卡片法术防御

int carddefend\_fan;//卡片防御反伤

int cardtypenames[MAX]={};//卡片属性

int cardtypenames\_index;//记录卡片属性数

//可能还有更多，敬请期待。

};

如默认卡牌：

CARDs[0].name="默认卡牌";

CARDs[0].cardtype=defaults;

CARDs[0].cardattack\_wu=0;

CARDs[0].cardattack\_fa=0;

CARDs[0].carddefend\_wu=0;

CARDs[0].carddefend\_fa=0;

CARDs[0].carddefend\_fan=0;

CARDs[0].cardtypenames[0]=none;

CARDs[0].cardtypenames\_index=1;

对应无任何功能的牌。

card中，cardtype有以下几种：

enum cartype{

defaults=-1,//默认

NPC=0,//电脑卡牌，玩家不可获得

often=1,//普通卡牌

sometimes=2,//优秀卡牌

never=3,//大师卡牌

amazing=4,//顶级卡牌

unbeliveble=5,//史诗卡牌

legend=6//传奇卡牌

};

card中，cardtypenames有：

enum cartypenames{

none=-1,//无属性

knife=0,//刀

attacker=1,//剑

fu=2,//斧

dun=3,//盾

fire=4,//火

ice=5,//冰

magical=6//神奇

};

定义的卡牌：（暂时）

CARDs[1].name="破剑";

CARDs[1].cardtype=often;

CARDs[1].cardattack\_wu=6;

CARDs[1].cardattack\_fa=0;

CARDs[1].carddefend\_wu=0;

CARDs[1].carddefend\_fa=0;

CARDs[1].carddefend\_fan=0;

CARDs[1].cardtypenames[0]=attacker;

CARDs[1].cardtypenames\_index=1;

CARDs[2].name="破盾";

CARDs[2].cardtype=often;

CARDs[2].cardattack\_wu=0;

CARDs[2].cardattack\_fa=0;

CARDs[2].carddefend\_wu=7;

CARDs[2].carddefend\_fa=0;

CARDs[2].carddefend\_fan=0;

CARDs[2].cardtypenames[0]=dun;

CARDs[2].cardtypenames\_index=1;

战斗模式：回合制，双方轮流出卡牌，谁的生命值先≤0谁输；在双方的生命值为正的情况下，谁的卡牌先耗光谁输；若一回合后双方卡牌同时耗光，先手输。

玩家的定义：

struct people{

string name;//名字

card cards[MAX]={};//拥有的卡牌

int cards\_num;//拥有的卡牌数

int things[MAX]={};//拥有的物品

int things\_num;//拥有的物品数

map<card,int> cardnums;//对应卡片的张数

map<int,int> thingnums;//对应物品的个数

int level;//等级

int health; //生命值

int experience;//经验值

};

携带卡牌上限：

inline int max\_card(int level){

if(level==1) return 2;

return level\*level+health/level;

}

简单列个表：（不含生命值）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 等级 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 卡牌上限 | 2 | 4 | 9 | 16 | 25 | 36 | 49 | 64 | 81 |
| 等级 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 卡牌上限 | 100 | 121 | 144 | 169 | 196 | 225 | 256 | 289 | 324 |
| 等级 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 卡牌上限 | 361 | 400 | 441 | 484 | 529 | 576 | 625 | 676 | 729 |
| 等级 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
| 卡牌上限 | 784 | 841 | 900 | 961 | 1024 | 1089 | 1156 | 1225 | 1296 |

但是对应的经验值却很难获得：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 等级 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 经验值 | 1 | 2 | 3 | 5 | 10 | 20 | 40 | 80 | 160 |
| 等级 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 经验值 | 320 | 640 | 1000 | 2000 | 4000 | 6000 | 10000 | 15000 | 20000 |
| 等级 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 经验值 | 25000 | 30000 | 35000 | 40000 | 45000 | 50000 | 55000 | 60000 | 65000 |
| 等级 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
| 经验值 | 70000 | 75000 | 80000 | 85000 | 90000 | 95000 | 100000 | 105000 | 110000 |

游戏的基础生命值的计算公式：（可能有物品加成）

inline int basic\_health(int level){

return level\*10+1;

}

有NPC（电脑玩家）的定义：

struct NPC{

string name;//名字

card cards[MAX]={};//拥有的卡牌

int cards\_num;//拥有的卡牌数

int things[MAX]={};//掉落的物品

int things\_num;//掉落的物品数

map<card,int> cardnums;//对应卡片的张数

map<int,int> thingnums;//对应物品的个数

int level;//等级

int health; //生命值

int experience;//奖励的经验

};