**无名杀零门槛技能编写教程**

写在前面的话

众所周知，无名杀主要由JavaScript编写，css布局的HTML开源工程。所以，只要学会了JS，就可以很轻易的进行技能编程，DIY武将。而即使没学过JS，在看完以下的教程后，你也能进行一些对技能的简易操作。



**本教程参考废城（QQ2375749244），永远的萌新，QQ2691330040 等制作，**我只是进行了重新整理和提供一点自己的想法，非常感谢大佬们。

无名杀下载地址及FQA 贴吧原贴:

[http://tieba.baidu.com/p/4206202301?share=9105&fr=share](http://tieba.baidu.com/p/4206202301?share=9105&fr=share" \o "http://tieba.baidu.com/p/4206202301?share=9105&fr=share)

可能需要的工具（安卓端）:

**MT管理器:[https://ww.lanzous.com/ic97vdc](https://ww.lanzous.com/ic97vdc" \o "https://ww.lanzous.com/ic97vdc)**

**apk编辑器Pro:[https://ww.lanzous.com/ic981af](https://ww.lanzous.com/ic981af" \o "https://ww.lanzous.com/ic981af)**

**搜图神器:[https://ww.lanzous.com/ibcetsj](https://ww.lanzous.com/ibcetsj" \o "https://ww.lanzous.com/ibcetsj)**

**snapseed:[https://ww.lanzous.com/ic98fti](https://ww.lanzous.com/ic98fti" \o "https://ww.lanzous.com/ic98fti)**

**三国杀武将牌制作:[https://ww.lanzous.com/ic98h8j](https://ww.lanzous.com/ic98h8j" \o "https://ww.lanzous.com/ic98h8j)**

re管理器:http://mobile.baidu.com/#/item?pid=1482773&from=1011512c

PS TOUCH:

http://mobile.baidu.com/#/item?docid=7321464&source=aladdin@wise\_app\_13@title&ala=wise\_app@strong@PS%20TOUCH

Xposed:

http://mobile.baidu.com/#/item?docid=7815786&source=aladdin@wise\_app\_13@title&ala=wise\_app@strong@X-posed

编辑文件:mt管理器，apk编辑器，es文件浏览器，re管理器，Xposed

照片查找:搜图神器

处理照片:snapseed，PS TOUCH

加粗的本人正在使用，确保链接应用无任何病毒，仅为推荐，不一定要下载

好了，前期的准备工作已经结束，让我们翻到下一页，进入正式的教程吧！

我们会先看到常用的词汇，技能效果，基本框架，技能模板，以及AI逻辑和理论教学。

**目录**

‖第一章 技能效果······p5

‖第二章 基本框架······p7

1.触发技 ············p7

2.主动技 ············p9

3.视为技 ············p11

4.A I框架············p12

‖第三章 AI 逻辑······p14

‖第四章 理论教程······p25

**附录······p2**

**如果你发现了本教程的错误，或者想反馈&建议那就告诉我吧:读书人QQ2375161944**

*为了显示附录的重要性，我把原本计划放在后面的附录，放到了最前面*

**附录:**

触发时机

　　【阶段类】

　　gameStart

　　//所有人都展示武将牌后 (前缀必须为global)

　　gameDrawBefore/Begin/End/After

　　//游戏开始阶段 前/时/后/结束后 (前缀必须为global)

　　phaseBefore/Begin/End/After

　　//回合阶段 开始前/时/后/结束后

　　judgeBefore/Begin/End/After

　　//判定阶段 开始前/时/后/结束后

　　phaseDrawBefore/Begin/End/After

　　//摸牌阶段 开始前/时/后/结束后

　　phaseUseBefore/Begin/End/After

　　//出牌阶段 开始前/时/后/结束后

　　discardBefore/Begin/End/After

　　//弃牌阶段 开始前/时/后/结束后

【卡牌类】

　　shaBefore/Begin/End/After/Miss

　　使用杀 之前/时/后/结束后/被闪后

player:"shaMiss",使用杀未命中

　　juedouBeforeBegin/End/After

　　使用决斗 之前/时/后/结束后

　　loseBefore/Begin/End/After

　　失去卡牌 之前/时/后/结束后

　　gainBefore/Begin/End/After

　　获得卡牌 之前/时/后/结束后

　　useCardBefore/Begin/End/After

　　使用卡牌 之前/时/后/结束后

　　useCardToBefore/Begin/End/After

　　使用卡牌指定 之前/时/后/结束后

　　respondBefore/Begin/End/After

　　打出卡牌 之前/时/后/结束后

　　drawBefore/Begin/End/After

　　从牌堆摸牌 之前/时/后/结束后

　　equipBefore/Begin/End/After

　　装备装备牌 之前/时/后/结束后

　　【体力类】

　　damageBefore/Begin/End/After

　　受到伤害 之前/时/后/结束后

　　(若前缀为source则为你造成伤害)

　　loseHpBefore/Begin/End/After

　　失去(流失)体力 之前/时/后/结束后

　　recoverBefore/Begin/End/After

　　回复体力 之前/时/后/结束后

　　changeHp

　　体力值发生改变后

　　loseMaxHpBefore/Begin/End/After

　　减少体力上限 之前/时/后/结束后

　　gainMaxHpBefore/Begin/End/After

　　增加体力上限 之前/时/后/结束后

　　【状态类】

　　chooseToRespondBefore/Begin/End/After

　　打出卡牌 之前/时/后/结束后

　　turnOverBefore/Begin/End/After

　　武将牌翻面 之前/时/后/结束后

　　linkBefore/Begin/End/After

　　武将牌横置(连环) 之前/时/后/结束后

　　dyingBefore/Begin/End/After

　　进入濒临状态 之前

常用词汇：

**trigger触发，phase阶段， player玩家，current当前玩家， target目标， global场上所有角色， event事件，card牌， handCard手牌 ，equip装备牌， trick锦囊牌，delay延时性锦囊， h手牌， e装备牌 ， j判定牌，**

**hp当前体力， maxHp体力上限， damage受伤 ， lose失去， gain获得，recover回复体力， discard弃置牌， draw摸牌， sep 步骤，**

**fire火焰伤害，thunder雷属性伤害，true成功， false失败，storage可理解为技能的标记，mark标记，countCard牌数， result结果，random随机，compare比较、拼点，gain获得，$gain给出牌，**

**type类型，basic基本牌，color卡牌颜色，spade黑桃，heart红桃，club梅花，diamond方块equip1武器 equip2 防具 equip3防御马 equip4进攻马 equip5宝物，**

**skill技能， add增加……**

常用效果（详见第一章:技能效果）:

**damage().\_triggered=null 神圣伤害**

**player.disableEquip(2); 废除装备**

**player.enableEquip() 回复装备**

**player.countCards('h') 当前的牌数**

**player.getStat().card.sha--; 出杀次数+1**

**game.countPlayer() 场上的玩家**

**player.chooseToDiscard(true,'h')玩家选择弃置一张手牌**

**return player.isMinHp(); 体力为最少或其中之一**

**var num=game.countPlayer(function(current){**

**return current.hp>player.hp;**

**});场上血量比玩家多的角色数**

**game.countPlayer(function(current){**

**return current.group=='wei'});场上的魏势力角色数**

**game.countGroup();场上的势力**

**player.draw(); 摸一张牌**

**player.recover(); 回复一点体力**

**player.damage(); 受到一点伤害**

**player.changeHujia(); 增加一点护甲**

**player.addSkill('技能的名字'); 玩家增加技能**

**player.addTempSkill('技能的名字',{时机}); 玩家获得临时技能时机那里填持续到什么时间**

**player.loseHp(); 玩家失去X点体力,X为（）里填的数字**

**player.init('武将的名字')；玩家将武将替换为括号里所填写的武将**

**player.link(); 玩家横置**

**player.turnOver(); 玩家翻面**

selectTarget目标 如 selectTarget:[1,5],指主动技选择1~5个目标 content:function ()里写技能的效果

filter:function (event,player){},里写发动技能的条件 小括号内的词随技能而变

如 filter:function (event,player){

return get.color(event.card)=='black';

},判断黑色牌的条件 红色是red替换black即可

filter:function (event,player){

return event.card.name=='shan'

},判断某种牌的条件 将牌名替换即可

filter:function (event,player){

return get.type(event.card)=='basic';

},判断基本牌的条件 trick,绵囊equip装备

filter:function (event,player){

return event.player.sex=='female';

},判断性别的条件 male男 female女

filterCard 选择卡牌的条件

position 可选择卡牌的类型

如position:"he",表可选择手牌和装备牌

filterCard:function (card,player){

return get.suit(card)=='club';

},判断某种花色牌的条件 **一丶 技能效果**

也可在游戏时填在命令里（如无特殊说明，以下x均可不填，x不填时数值默认为1，x∈[-Infinity,Infinity] Infinity为正无穷大）

1.player.draw(x) 玩家摸x张牌

2.player.recover(x) 玩家回复x点体力

3.player.damage(a,'thunder')玩家受到a点雷属性伤害

（目标受到一点雷电伤害：target.damage('thunder')）

4.player.damage(a,'fire')玩家受到a点火属性伤害

（目标受到一点火焰伤害：target.damage('fire')）

5.player.loseHp(x) 玩家流失x点体力

6.player.gainMaxHp(x) 玩家体力上限+x，x为Infinity时游戏会出现卡机

7.player.loseMaxHp(x) 玩家体力上限-x，x为Infinity时游戏会出现卡机

8.player.chooseToDiscard(a,b) 玩家选择弃置a张牌，a不填时默认为1，b不填时可以取消弃置，b为true等杂七杂八的东西时强制弃置

9.player.discard(a) 玩家选择失去a张牌

10.player.turnOver(a) a不填时玩家翻面，a填false时玩家解除翻面，填其他效果未知

11.player.link(a) a不填时玩家横置，a填false时玩家解除横置，填其他效果未知

12.player.remove() 玩家移出游戏，回不来的这种

13.player.out() 玩家离开游戏

14.player.die() 玩家立刻死亡

15.player.delete() 神圣死亡

16.player.awakenSkill('a') 限定技专用，a为技能名，让技能不可再次使用，该语句需要和player.storage.a=true连用 如player.storage.神音=true; player.awakenSkill('神音');

17.player.removeSkill('a') 玩家失去a技能，a为技能名

18.player.clearSkills() 玩家清空所有技能

19.player.update() 刷新

20.player.setIdentity('a') 设置玩家身份/势力为a，需要和player.identity=a 和player.node.identity.dataset.color='xxx'和player.identityShown=true等语句连用

21.player.goMad() 玩家陷入混乱

22.player.skip('a')玩家跳过a阶段

（判定阶段phaseJudge ，摸牌阶段phaseDraw，出牌阶段phaseUse，弃牌阶段discard）

23.player.init('a') 将玩家的武将牌替换为a，a为武将名

24.player.addSkill('a') 玩家获得a技能，a为技能名

25.player.addTempSkill('a','b') 玩家获得a技能直至b时刻，a为技能名，b为时机。 （gameDrawBefore/Begin/End/After//游戏开始阶段 前/时/后/结束后 (前缀必须为global)

phaseBefore/Begin/End/After//回合阶段 开始前/时/后/结束后

judgeBefore/Begin/End/After//判定阶段 开始前/时/后/结束后

　phaseDrawBefore/Begin/End/After//摸牌阶段 开始前/时/后/结束后

phaseUseBefore/Begin/End/After//出牌阶段 开始前/时/后/结束后

discardBefore/Begin/End/After//弃牌阶段 开始前/时/后/结束后）

26.player.revive() 玩家复活

27. player.gain(a,b) 玩家获得a张牌，a不填时默认为1，b不填时可以取消弃置，b为true等杂七杂八的东西时强制获得

如 player.gain(2,target);

target.$giveAuto(1,player);

//获得目标的2张牌，还给他1张牌（不想弃置转第一条）

28.player.changeHujia(a); 玩家获得a点护甲

**二丶 基本框架**

1.触发技：

trigger:{//触发者:'触发时机'（触发者：场上的人global，玩家player，自己为来源source；触发时机:见附录）

global:'damageBefore',//例子，任意一名角色受到伤害前

player:'damageBefore',//例子，玩家受到伤害前

player:['damageBefore', 'recoverAfter'], //例子，玩家受到伤害前或玩家恢复体力后

},

priority: 1,//同时机技能发动的优先度

check:function(event,player){//告诉ai是否发动这个技能，返回true则发动此技能

//无例子，因为不同时机evevt参数内的元素不同

//要了解evevt参数内的元素，具体看game.js内lib.element.event内的函数

},

frequent:true,//此技能可以被设置为自动发动

forced:true,//表示锁定技

nopop:true,//player是否logSkill('此技能')，true为不

direct:true,//相当于上面两行相加

filter:function(event,player){//触发条件，返回true则发动此技能

//无例子，因为不同时机evevt参数内的元素不同

//要了解evevt参数内的元素，具体看game.js内lib.element.event内的函数

},

content:function(){//触发内容

//参数trigger相当于filter:function(event,player){}内的event

player.recover()//例子，玩家回复一点体力

},

技能模板（郭嘉〖遗计〗为例）

skill={

audio:2,

trigger:{

player:"damageEnd",

},

frequent:true,

filter:function(event){

return (event.num>0)

},

content:function(){

"step 0"

event.count=trigger.num;

"step 1"

event.count--;

event.cards=get.cards(2);

"step 2"

if(event.cards.length>1){

player.chooseCardButton('将“遗计”牌分配给任意角色',true,event.cards,[1,event.cards.length]).set('ai',function(button){

if(ui.selected.buttons.length==0) return 1;

return 0;

});

}

else if(event.cards.length==1){

event.\_result={links:event.cards.slice(0),bool:true};

}

else{

event.goto(5);

}

"step 3"

if(result.bool){

for(var i=0;i<result.links.length;i++){

event.cards.remove(result.links[i]);

}

event.togive=result.links.slice(0);

player.chooseTarget('将'+get.translation(result.links)+'交给一名角色',true).set('ai',function(target){

var att=get.attitude(\_status.event.player,target);

if(\_status.event.enemy){

return -att;

}

else if(att>0){

return att/(1+target.countCards('h'));

}

else{

return att/100;

}

}).set('enemy',get.value(event.togive[0],player,'raw')<0);

}

"step 4"

if(result.targets.length){

result.targets[0].gain(event.togive,'draw');

player.line(result.targets[0],'green');

game.log(result.targets[0],'获得了'+get.cnNumber(event.togive.length)+'张牌');

event.goto(2);

}

"step 5"

if(event.count>0) player.chooseBool(get.prompt2(event.name)).set('frequentSkill',event.name);

else event.finish();

"step 6"

if(result.bool){

player.logSkill(event.name);

event.goto(1);

}

},

2.主动技：

enable:"phaseUse",//出牌阶段使用

usable:1,//数字可以更改，表示可发动次数

round:2,//数字可以更改,表示每几轮限1次

//限制使用次数为变量时需写在filter:function(event,player){}内

filter:function(event,player){ //发动条件，返回true则可以发动此技能

/\*限制使用次数为变量时:

if(player.getStat().skill['技能名']>=变量) return false;（变量可以写player.hp等）

return true;

\*/

return player.hp<2;//例子，玩家体力小于2时可以发动

},

filterCard:function(card){//选择的牌需要满足的条件

return get.color(card)=='red';//例子，只能选择红色牌

},

position:'hej',//可以选择什么位置的牌（h手牌， e装备牌 ， j判定牌）

selectCard:2,//需要选择多少张牌才能发动（为-1时，选择所有牌）

/\*为变量时（具体情况具体分析）:

function(card){//例子，变量为玩家体力时

var pl=get.owner(card);//pl定义为选择牌中的牌的拥有者，即“你”

return [pl.hp,Infinity];//返回可选手牌数为你的体力值到无限

},

\*/

check:function(card){//ai应该如何选牌，返回值为正则此牌可选，反之，此牌不可选

/\*在check中定义玩家（你）：

var player=get.owner(card);

\*/

return 6-get.value(card);//例子，ai可选价值大于6的卡牌

},

discard:false,//是否弃牌，若没有这一行，选择牌发动技能后，被选择的牌都要弃置

prepare:'give',//不弃牌，准备用这些牌来干什么

filterTarget:function(card,player,target){//选择的目标需要满足的条件

return target!=player;//例子，目标不是玩家

},

selectTarget:1, //需要选择多少个目标才能发动

//为-1时，选择全部人

//为数组时，这个数组就是选择目标数的区间

multitarget:true,//是否每个目标都结算一次，true为否

targetprompt:[],//选择的目标武将牌上出现什么字，数组第几元素对应第几个目标

content:function (){//触发内容

player.recover()//例子，玩家回复一点体力

},

技能模板（张辽〖突袭〗为例）

skill={

audio:2,

trigger:{

player:"phaseDrawBegin1",

},

direct:true,

filter:function(event,player){

return !event.numFixed;

},

content:function(){

"step 0"

var check;

var i,num=game.countPlayer(function(current){

return current!=player&&current.countCards('h')&&get.attitude(player,current)<=0;

});

check=(num>=2);

player.chooseTarget(get.prompt('tuxi'),'获得其他一至两名角色的各一张手牌',[1,2],function(card,player,target){

return target.countCards('h')>0&&player!=target;

},function(target){

if(!\_status.event.aicheck) return 0;

var att=get.attitude(\_status.event.player,target);

if(target.hasSkill('tuntian')) return att/10;

return 1-att;

}).set('aicheck',check);

"step 1"

if(result.bool){

player.logSkill('tuxi',result.targets);

player.gainMultiple(result.targets);

trigger.changeToZero();

}

else{

event.finish();

}

"step 2"

game.delay();

},

3.视为技：

（1）转化为一张卡牌：

基本框架：

name:'技能',//技能按钮名字，不写则默认为此技能的翻译

enable:['phaseUse','chooseToRespond'],//发动视为技的时机，现在为出牌阶段和响应时

viewAsFilter:function(player){//视为技按钮出现条件（即发动条件）

return player.maxHp==4;//例子，玩家最大体力值为4时可以发动 },

filterCard:{type:'basic'},//选择的牌需要满足什么条件

/\*上面那种是简便写法，多次判断的话还是要写回旧格式：

filterCard:function(card){

return get.color(card)=='black';//例子，只能选择黑色牌

},

\*/

position:'hej',//可以选择什么位置的牌

selectCard:2, //需要选择多少张牌才能发动

check:function(card){// ai应该如何选牌

return 1;//例子，ai可以使用所有手牌

},

viewAs:{name:'tao'},//目标卡牌

prompt:'将一张基本牌当桃使用或打出',//选择时弹出的提示

onuse:function(result,player){//使用视为牌时触发内容，result.cards是视为前的牌

player.recover();//例子，使用视为牌时玩家回复一点体力

},

（2）转化为多张卡牌：

基本框架：

enable:'phaseUse', //发动视为技的时机，现在为出牌阶段

usable:1, //每回合限制使用次数，和主动的usable相同，不解释

filter:function(event,player){ //发动条件，和主动的filter相同，不解释

return player.countCards('h')>0//例子，玩家手牌数大于0时可以发动

},

chooseButton:{//选择按钮（牌）

dialog:function(){//选择内容，一般为选择卡牌，有能力的可以写成选择其他

return ui.create.dialog([['sha','shan'],'vcard']);//可以选择杀和闪（卡牌）

},

filter:function(button,player){//卡牌选择条件

return lib.filter.filterCard({name:button.link[2]},player,\_status.event.getParent());

//例子，你可以选择你可以使用的卡牌

},

check:function(button){//ai如何选牌

/\*定义player：

var player=\_status.event.player;

\*/

if(button.link=='sha') return 1;//ai选择杀

},

backup:function(links,player){//返回视为（1）部分

return {//例子

filterCard:false,//不用弃牌

selectCard:0, //弃牌数为零

viewAs:{name:links[0][2]},//视为你使用你选择的牌

};

},

prompt:function(links,player){ //选择时弹出的提示

return '请选择'+get.translation(links[0][2])+'的目标';

//例子，弹出提示：选择你选择的牌的目标

},

},

技能模板（甘宁〖奇袭〗为例）

skill={

audio:2,

audioname:["re\_ganning","re\_heqi"],

enable:"chooseToUse",

filterCard:function(card){

return get.color(card)=='black';

},

position:"he",

viewAs:{

name:"guohe",

},

viewAsFilter:function(player){

if(!player.countCards('he',{color:'black'})) return false;

},

prompt:"将一张黑色牌当过河拆桥使用",

check:function(card){return 4-get.value(card)},

4.AI框架:

ai:{

save:true, //此技能可以用于自救

respondTao:true, //此技能可以用于救人，一般用于视为技

respondShan:true,//此技能可以响应杀，一般用于视为技

respondSha:true, //此技能可以响应闪，一般用于视为技

order:8, //ai发动技能的优先度

/\*比什么先：

order:function(){//例子，比打出杀的优先度高1

return get.order({name:'sha'})+1;

},

\*/

expose:0.2, //发动技能是身份暴露度（0~1，相当于概率）

maixie:true, //卖血技

threaten:0.5,//嘲讽值

result:{//主动技专属

target:function(player,target){//ai如何选择目标

//返回负，选敌人，返回正，选队友

//没有返回值则不选

//写了这个就不用写player:function(player){}了

//因为player可以在这里进行判断

If(player.hp>2) return -1;//例子，玩家体力大于2时，发动技能并选择敌人

},

player:function(player){//ai是否发动此技能，返回正，发动，否则不发动

if(player. countCards('h')<2) return 1;//例子，玩家手牌数小于2时，发动 },

},

effect:{//影响ai出牌（例如什么时候不出杀）等

},

skillTagFilter:function(player){//视为技专属，ai什么时候可以发动视为技

player.countCards('h',{type:'basic'})>0;//例子，手牌中有基本牌时，ai发动技能

},

},

技能模板（李典〖忘隙〗为例）

　　ai:{

　　effect:{

　　player:function(card,player){

　　if(card.name=='nanman'||card.name=='wanjian') return [1,3];

　　}}}

后两章因为时间等原因没能整合制作，所以就直接copy了，希望别被打（墙角害怕）后续版本会更新。**三丶 AI逻辑**

这里复制的是 *废城* 的外篇AI教程

　　『1』名词解释:

　　1.1 基本名词

　　order:『优先级』,用来设定技能发动的优先程度,此程度以卡牌使用的优先级为参考。为了实现合理的牌序(例如先拆后顺再杀),须将每种卡牌设定一个优先级,优先级越大的卡牌ai会优先使用,技能也是如此。

　　设拆的优先级为9,顺为7,杀为4:

-10(枪舞)--9(拆)--8(苦肉)--7(顺)--6--5--4(杀)--;

　　expose:『身份暴露程度』,取值范围为0~1,用于帮助AI判断身份,AI中未写expose其他AI将会无法判断其身份;

　　threaten:『嘲讽值』,嘲讽值越大的角色越容易遭受到敌方的攻击,默认为1,一般在0~4中取值即可(缔盟threaten值为3);

　　respondShan:true,『响应闪』,作用是告诉AI手里没『闪』也可能出『闪』,防止没『闪』直接掉血;

　　respondSha:true,与respondShan类似;

　　save:true,告诉AI可能救人,防止没桃xxx;

　　maixie:true,技能标签『卖血』,用于其他AI检测是否是卖血流(if(target.hasSkillTag('maixie')))。并非加了这个AI就会卖血;

　　noh:true,技能标签『无牌』,目前只出现在“连营”和“伤逝”中,用于其它AI检测是否含有标签『无牌』,从而告诉其他AI不要拆迁(因为生生不息)。

　　1.2 AI获取(玩家):

　　ai.get.attitude(玩家1,玩家2):『态度值检测』检测玩家1对玩家2的态度值;

　　ai.get.effect(target,content,player,viewer):『效果检测』检测卡牌/技能对目标角色的效果值,content代表卡牌或者技能,例如

　　{name:'sha'}/'挑衅';viewer代表视角,一般填player或者target,例如我方杀敌方满血『曹丕』,对我方来说是负效果,但对敌方是正效果,

　　视角决定效果的正负。

　　ai.get.damageEffect(玩家1,玩家2,viewer,nature):『检测伤害效果』这里检测的是[玩家2对玩家1]的伤害效果,nature代表属性(雷/火);

　　ai.get.recoverEffect(玩家1,玩家2,viewer):『检测回复效果』 与上段类似;

　　1.3 AI获取(卡牌)：

　　useful：回合外留牌的价值(弃牌阶段按useful顺序选)

　　value：牌的使用价值(五谷按value顺序选)

　　举例:ai.get.useful(card)/ai.get.value(card)

　　『2』AI的分类与写法：

　　2.1 AI的合理位置

　　1.AI须选择角色/卡牌/按钮等等时

　　2.check中检测条件时的收益时

　　3.主动技的结尾,用来检测是否发动和对谁发动

　　4.需要声明有某技能时某牌的使用效果发生改变时(大部分是卖血技)

　　2.2 不同位置的AI写法

　　2.2.1 选择(choose类):

　　target.chooseToDiscard().ai=function(card){

　　return 8-ai.get.value(card);

　　}

　　return 8-ai.get.value(card)这句是告诉AI如果手里有价值小于8的牌就弃置,否则不弃置

　　弃置的规则是价值越小的牌优先丢,一个return 5和return 3 AI会选return 5的卡牌。

　　当return值不大于0时,例如最后一张手牌价值为9 (return 8-9)=-1,AI将会放弃。

　　target.chooseTarget().ai=function(target){

　　return ai.get.attitude(player,target);

　　}

　　同理,att值越大的会优先选。注意att值只与身份有关,一个鲁肃一个于禁队友,并不一定会选鲁肃。

　　player,chooseControl('摸牌','回血').ai=function(event,player){

　　if(player.hp>=2) return '摸牌';

　　return '回血';

　　}

　　2.2.2 检测(check类)：

　　举例：『魅步』

　　check:function(event,player){

　　if(ai.get.attitude(player,event.player)>=0) return false;

　　//对当前角色态度大于等于0,不发动;

　　var e2=player.get('e','2');

　　//定义一个变量e2代表获取玩家的装备区的防具;

　　if(e2){

　　if(e2.name=='tengjia') return true;

　　if(e2.name=='bagua') return true;

　　//有藤甲和八卦就发动。。。老大这是瞧不起仁王吗。。。。

　　}

　　return player.num('h','shan')>0;

　　//如果上述条件均不满足,如果玩家手里有『闪』 发动,否则不发动;

　　},

　　发动技能规则:check最终的return值大于0;

　　举例:『峻刑』(这是个主动技,不过也需要check决定是否发动)

　　check:function(card){

　　if(ui.selected.cards.length) return -1;

　　//告诉AI只选一张牌;

　　if(get.type(card)=='basic') return 8-ai.get.value(card);

　　//只有价值小于8的基本牌可被选;

　　return 5-ai.get.value(card);

　　//如果手中有价值小于5的牌就发动,否则不发动;

　　},

　　2.2.3 主动技结尾

　　一般格式为:

　　ai:{

　　order:xx,

　　threaten:xx,

　　//上述为定义/标签

　　result:{

　　player:xx;

　　target:xx;

　　}

　　//收益值

　　}

　　2.2.4 卖血技能

　　一般格式为:

　　ai:{

　　maixie:true,

　　threaten:xx,

　　//上述为定义/标签

　　effect:{

　　xx;

　　}

　　//效果值的重新定义

　　}

　　更多内容详见第三章『收益论』;

　　『3』收益论

　　3.1 收益论用途

　　1.告诉AI会发动技能,

　　2.告诉AI会卖血,

　　3.告诉AI倾向于卡牌/行为,

　　3.2 收益论名词:

　　result:『收益』收益值未在AI声明默认为0(对玩家对目标均是如此).收益值可以直接在AI声明;

　　result在player部分里声明就是对玩家的收益,在target部分里声明就是对目标的收益;

　　effect:『效果』效果值为正代表正效果,反之为负效果,AI会倾向于最大效果的目标/卡牌;

　　attitude:『态度』态度只由identity决定。不同身份对不同身份的att不同。例如在身份局中,主对忠att值为6,

　　忠对主att值为10;

　　3.3 收益论应用:

　　3.3.1 【主动技】:

　　?如果技能发动无须指定目标: effect=result\*ai.get.attitude(player,player);

　　即: 总效果=收益值\*使用者对自己的att;

　　举例『制衡』:

　　ai:{

　　result:{

　　player:1

　　}}

　　\*由于制衡基本是每回合都会发动,所以AI中将『收益值』直接定为1,假设孙权对自己的att为10,这将代表：

　　总效果=(1)收益值\*(10)=10;由于效果值为正(正收益) 所以AI将会发动『制衡』

　　举例『急袭』:

　　ai:{

　　order:10,

　　result:{

　　player:function(player){

　　return player.storage.tuntian.length-1;

　　}}}

　　\*这个很容易看出来,只有一个田就不发动了...只要田大于1,对使用者就是正收益,再乘以对自己的att,也是正效果,所以AI将会发动『急袭』

　　?如果技能发动须指定目标

　　总效果=对使用者的收益值\*使用者对自己的att+对目标的收益值\*使用者对目标的att;

　　实际还会考虑嘲讽值,这里简化了;

　　举例『挑衅』:

　　ai:{

　　order:4,

　　expose:0.2,

　　threaten:1.1,

　　result:{

　　target:-1,

　　//直接定义目标的收益值为-1,

　　player:function(player,target){

　　if(target.num('h')==0) return 0;

　　if(target.num('h')==1) return -0.1;

　　if(player.hp<=2) return -2;

　　if(player.num('h','shan')==0) return -1;

　　return -0.5;

　　//上述表示目标角色不同的情况,玩家将会return不同的收益值;

　　}}}

　　『挑衅』是一个作死技,技能发动对自己没有正收益,只会给目标角色带来负收益。所以

　　player部分中收益值最大也不会超过0,因为是弃置对方一张牌,所以直接将目标角色的收益

　　定为了-1;如果对自己的负收益大于对敌方的负收益的话,姜维将不会发动『挑衅』

　　举例『排异』:

　　ai:{

　　order:1,

　　result:{

　　target:function(player,target){

　　if(player!=target) return 0;

　　if(player.num('h')+2<=player.hp+player.storage.quanji.length) return 1;

　　return 0;

　　}}}

　　『排异』为什么没有声明player部分呢,因为除非你排异自己,否则你自己是没收益的,只是对你选择的目标

　　有收益(摸2牌)。

　　3.3.2 【卖血技】

　　总效果=对使用者的收益值\*使用者对自己的att+对目标的收益值\*使用者对目标的att;

　　实际还会考虑嘲讽值,这里简化了;

　　ai.effect：告诉ai有某技能时某牌的使用效果发生改变,

　　设对目标的原收益为n,对使用者的原收益为n',n>0代表正收益,n<0代表负收益;

　　函数传入4个参数,分别为卡牌、使用者、目标以及n;

　　返回值可有3种形式

　　1. 一个数a,对目标总的收益为a\*n;

　　2. 一个长度为2的数组[a,b],对目标的总收益为a\*n+b;

　　3. 一个长度为4的数组[a,b,c,d],对目标的总收益为a\*n+b,对使用者的总收益为c\*n'+d;

　　假设n代表火杀对目标的效果

　　1. 技能为防止火焰伤害：return 0;

　　2. 技能为防止火焰伤害并令目标摸一张牌：return [0,1];

　　3. 技能为防止火焰伤害并令使用者弃置一张牌：return [0,0,1,-1];

　　举例『遗计』:

　　ai:{

　　maixie:true,

　　effect:{

　　target:function(card,player,target){

　　if(get.tag(card,'damage')){

　　if(player.skills.contains('jueqing')) return [1,-2];

　　if(!target.hasFriend()) return;

　　//如果没有存活队友,也就是说你方阵营就你一个活着的了,直接跳出effect,避免单挑永远不杀你;

　　if(target.hp>=4) return [1,get.tag(card,'damage')\*2];

　　if(target.hp==3) return [1,get.tag(card,'damage')\*1.5];

　　if(target.hp==2) return [1,get.tag(card,'damage')\*0.5];

　　//以『杀』为例,杀对目标角色为-1.5收益,计算效果得:

　　4血及以上:-1.5\*1+2=0.5 正收益;

　　3血:-1.5\*1+1.5=0 无收益;

　　2血:-1.5\*1+0.5=-1 负收益;

　　}}}}

　　这个get.tag是什么意思呢？其实和技能标签一样,卡牌也有自己的标签,例如

　　杀:{

　　tag:{

　　respond:1,

　　respondShan:1,

　　damage:function(card){

　　if(card.nature=='poison') return;

　　return 1;

　　}}}

　　为了方便删了一些tag,那么『杀』有如下tag:respond;respondShan;damage等等,所以get.tag(card,'damage')

　　代表有伤害属性的卡牌,例如杀,南蛮,决斗等等。

　　\*特别注意ai里面的effect是上帝视角,target不代表目标角色,player也不代表拥有此技能的玩家本身,

　　因为effect是写给别的AI看的,所以target代表玩家本身,player代表其他人,可以是正在犹豫是否要杀

　　你的任何一位玩家。

　　那么队友会杀卖血流队友还是杀敌方呢？还是套公式:

　　总效果=对使用者的收益值\*使用者对自己的att+对目标的收益值\*使用者对目标的att;

　　因为『杀』对使用者是没收益的,所以公式变为:

　　总效果=对目标的总收益值\*使用者对目标的att;

　　比如杀对目标角色的收益是-1.5,满血曹丕的effect是return [0.5,1],那最终的收益是0.25,如果对曹丕的attitude是1,对某白板敌人的attitude是-1,那杀曹丕的效果为1\*0.25,

　　杀敌人的效果为-1\*-1.5,所以最终会选敌人(实际稍复杂点,详见ai.get.effect函数)

　　3.3.2 【倾向技】

　　这个占比例非常少,略说明一下

　　举例『忘隙』:

　　ai:{

　　effect:{

　　player:function(card,player){

　　if(card.name=='nanman'||card.name=='wanjian') return [1,3];

　　}}}

　　这里的意思是尽量放南蛮万剑之类,类似的,激昂也是强调AI尽量用红杀;

　　因为上述AI中effect里面的为player,所以代表的是对使用者的收益。

**四丶 理论教程**

本章复制 *废城* 的JS语法规则和css

1：变量(自由改变的数字)

　　" "或者' '将数字括起来

　　2:运算符

　　输出 return

　　真(成立/正确)true

　　假(不成立/错误)false

　　加+

　　减-

　　乘\*

　　除/

　　取模%

　　数值加1 ++

　　数值减1 --

　　小于<

　　等于=

　　大于>

　　大于等于>=

　　与&

　　与之后赋值&=

　　或之后赋值|=

　　小于等于<=

　　等于等于==

　　不等于!=

　　绝对值abs()

　　四舍五入round()

　　空值null

　　//这个符号是解释用的，无实际意义，符号后不翻译，但仅隐藏一行文字或数据

　　2.1常量：

　　无穷大，最大

　　lnfinity

　　负无穷

　　-lnfinity

　　不能和自己比较的唯一值

　　NaN： not a number

　　2.2number的特殊常量

　　number.MAX\_VALUE最大值

　　number.MIN\_VALUE最小值

　　number.NaN空值

　　2.3:语句(共5种，目前介绍一种)

　　if:加上判断条件，如果成立(成立用true表示)则执行

　　else:不成立(用false表示)则执行

　　例：

　　if(1==1&&6>3)

　　{

　　alert(return true);

　　}

　　else

　　{

　　alert(return false);

　　}

　　执行语句:Function{

　　}

　　2.35算数式

　　var pingguo=6;

　　var lizi=3;

　　var huaqian=pingguo+li;

　　document.write(“花费：”=huaqian)

　　//苹果6元，李子3元，花钱=苹果和李子加起来，输出花钱=

　　2.4赋值语句：

　　var a=10;

　　//赋值让a=10

　　var x=1/0;

　　//x==lnfinity

　　var y=-1/0;

　　//y==-lnfinity

　　var x=0/0;

　　//x==NaN

　　var m=0+“1”;

　　//m==1

　　var n=0+“string”;

　　//n==NaN

　　注：首字母大写，区分大小写，不大写和大写分别表示二个意义

　　工具电脑版：firebox和firebug

　　编辑器：

　　Aptana

　　记事本

　　WebStrom

　　DreamWeaver

　　……

　　第二章

　　简易css

　　外部样式：

　　内部样式：

　　属性名：内容(content)、填充(padding)、边框(border)、边界(margin)， CSS盒子模式都具备这些属性。