|-----------无名杀技能函数手册-------------------------------------------------------------------------|

| 主编：美妙的世界 |

| 技术主要提供者：苏婆，寰宇星城（众所周知）， |

| 技术次要提供者：霸天，平西镇北征南破东大司马（魔将扩展作者,） |

| 浪琴婊（很奇怪的昵称，总是让人联想到一些奇奇怪怪的事情） |

|-----------------------------------------------------------------------------------------------------|

本手册相当于无名杀技能函数词典，方便有基础的人看，若是新手可配合武将源码慢慢了解理解。

同时也欢迎大佬进行补充

|-----------player对象----------------|

player代表角色或玩家

target代表目标角色/区域

|------------------------------------|

https://tieba.baidu.com/p/6162185467?share=9105&fr=sharewise&see\_lz=0&share\_from=post&sfc=qqfriend&client\_type=2&client\_version=12.16.1.0&st=1640511574&unique=F76ECC17CFC35C49FFEA1E4AF1B66F71&red\_tag=3440467701

半官方

139.196.24.143

159.75.51.253

目录

[触发机制1 2](#_Toc93833667)

[函数 5](#_Toc93833668)

[content操作函数 5](#_Toc93833669)

[Player操作函数 5](#_Toc93833670)

[game操作函数 33](#_Toc93833671)

[Get对象中的函数 34](#_Toc93833672)

[Mod函数 35](#_Toc93833673)

[零散教程 36](#_Toc93833674)

# 触发机制1

　　【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

　　phaseJudgeBefore/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

　　phaseDiscardBegin/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/时/后/结束后

　　dieBefore/Begin

　　死亡 之前/时

gainBefore/Begin/End/After

获得牌 之前/时/后/结束后

loseBefore/Begin/End/After

失去牌 之前/时/后/结束后

wuxieBegin 使用无懈可击时

useCard1(声明使用牌后)和useCard2(使用牌选择目标后)

useCard1目前在无名杀里只有朱雀羽扇（程普没有使用这个时机）

useCard2一般是一些（可以令一些其他角色也成为XXX牌目标）的技能

之后就是大家熟悉的useCard(牌被使用时)时机

一般用于一些纯粹摸牌系和全体强命系的技能

而useCardToPlayer(指定目标时)和useCardToTarget(成为目标时) 则往往是一些会改变卡牌目标数的技能

useCardToPlayer和useCardToTarget这两个时机 对卡牌的所有目标都会依次触发一遍 此时trigger.target即为被触发的目标

需要特别注意的是：这两个时机的trigger本身并不是useCard事件 trigger.parent/trigger.getParent()才是

之后依然是一连串的useCardToPlayered(指定目标后)和useCardToTargeted(成为目标后)的时机了

注意要点和上面差不多

时机中useCardToTargeted， 成为目标后

useCardToTarget， 成为目标时

useCardToPlayered 指定目标后

具体去看规则集，简单一点来说就是"指定"是使用者方的技能，先触发 ,"成为"是挨打方技能

trigger:{target:"useCardToTarget"}, //这句话代表其他角色对你出牌

新版本的伤害事件中 添加了damageBegin1 damageBegin2 damageBegin3 damageBegin4这四个新时机

damageBegin1

能发动的武将技能或能执行的武将技能的效果:〖暗箭〗、〖从谏①〗、〖骄

恣①》、〖竭缘①〗、〖裸衣〗的延时类效果、〖倾袭〗、〖凶镬③〗、〖义绝〗的延廷时类效果、〖赠刀〗

的延时类效果、【忠义】的延时类效果、〖助祭】;能发动的装备技能:〖古锭刀】、【七宝刀②〗

damageBegin2

造成伤害时e:能发动的武将技能:〖豹变》、〖穿心】、〖潜袭(旧将)〗、〖无言①〗、〖制

蜜〗;能发动的装备技能:〖寒冰剑〗、〖麟麟弓〗。

damageBegin3

受到伤害时e:能发动的武将技能:【仇海】、〖从谏②】、〖公清②〗,〖骄矜〗、〖骄恣②〗

〖蝎缘②】、〖恪守〗、〖狂风〗的延时类效果、〖链祸》、〖名士〗、〖天香(1v1)】、〖耀武〗;能

发动的装备技能:〖藤甲②。

damageBegin4

(5)受到伤害时●:能发动的武将技能或能执行的武将技能的效果:〖安东〗、〖白眉①〗、白

眉②〗、〖大雾】的延时类效果、〖公清①》、〖怀橘②〗、〖贿生】、〖宽释①】的延时类效果、〖仁

心》、〖释衅〗、〖天香〗、〖隐士〗、〖远域〗、〖无言②〗;能发动的装备技能:〖白银狮子①〗

〖护心镜〗、〖烂银甲②〗、〖太平要术①〗

phaseZhunbeiBegin/phaseJieshuBegin

新版本的【准备阶段】和【结束阶段】的时机

新版本的judge事件中 可以通过设置callback事件 在judgeEnd和judgeAfter时机之前对判定牌进行操作 避免被曹植截胡

# 函数

element{

## content操作函数

content:{

拼点时失去牌的事件

chooseToCompareLose()

王司徒拼三人

chooseToCompareMultiple()

使用一次技能

useSkill(id)

id填技能代码名称

}

## Player操作函数

player{

判断是男是女

player.hasSex(sex)

sex为"male"或"female"字符串，

返回bool值

判断角色的性别是否和目标角色相同

player.sameSexAs(target)

target为对象，

返回bool值

判断角色的性别是否和目标角色不同

player.differentSexFrom(target)

target为对象，

返回bool值

添加技能屏蔽器。

对一名角色添加技能屏蔽器，无效化其符合筛选条件的技能，直到技能屏蔽器被移除。

player.addSkillBlocker(skill)

skill为对象，

无返回值

参考：

game.js里的baiban技能和fengyin技能。或者贾诩的rewansha技能

移除技能屏蔽器

removeSkillBlocker(skill)

skill为对象，

无返回值

使角色失去牌，并将牌移动到特殊区域，也就是所谓移出游戏。

player.loseToSpecial(cards,tag,target)

tag是技能的id，想让这个牌上显示啥字就填什么

target为目标区域

返回：

给玩家的卡牌添加文字标记参考 董白

player.addGaintag(cards,tag)

卡牌去除文字标记

removeGaintag(tag,cards)

第二个是能否救人

canSave(target)

返回bool

player.canSave(target);

用于判断一名角色能否对另一名濒死角色提供帮助。

一般国战使用，暗置的武将明置

showCharacter(num,log)

暗置的武将明置动画函数

$showCharacter(num,log)

观星的效果

chooseToGuanxing(num)

num为观看的牌数

chooseToGuanxing(X)

令一名角色观看牌堆顶的X张牌并以任意顺序置于牌堆顶或牌堆顶。

送花砸蛋的函数

$throwEmotion(target,name)

判定动画 用来弹出判定失败和判定生效的

tryJudgeAnimate(bool)

技能动画

trySkillAnimate(name,popname,checkShow)

卡牌动画

tryCardAnimate(card,name,nature,popname)

玩家是否能够在当前的情况下使用某张牌

hasUsableCard(name)

name填牌的名称id

返回bool值

player.hasUsableCard()

这个函数用于判断一名角色能否使用XX牌

如果该角色手牌中有XX牌 或者能够通过技能使用XX牌 则返回值为true 否则为false

例：判断能否使用【草船借箭】的代码

...

return player.hasUsableCard('caochuan');

...

直接填写要判断的卡牌名称即可

目标是否在自己攻击范围 判断target是否在player的攻击范围内

player.inRange(to)

自己是否在来源攻击范围 判断player是否在source的攻击范围内

player.inRangeOf(source)

返回玩家已损失的体力值数

player.getDamagedHp()

改变武将势力

changeGroup(group,log,broadcast)

用于切换玩家的国籍

例：player.changeGroup('wei') 即为将玩家的势力切换为魏国

（在国战模式下不影响势力，胜利条件，野心家判断等）

选择对策

chooseToDuiben(target)

对一名其他角色发起对笨(划掉)对策

为审配专门加入的函数 用法和石头剪刀布类似 只不过没有平局的结果

顺便 发起人在对策过程中是选择防御策略的角色 在用的时候不要搞错了

猜拳

chooseToPSS(target)

chooseToPSS()

石头剪刀布的函数 用法和拼点类似

例:player.chooseToPSS(target);

如果发起玩家胜利 那么result.bool值为true 失败则为false 如果平局 那么result.tie值为true

玩家选择恢复一个装备栏

player.chooseToEnable()

玩家选择废除一个装备栏

player.chooseToDisable(horse)

返回玩家已废除的装备栏数量

player.countDisabled()

判断玩家是否处于出牌阶段，参数为true代表不是自己的出牌阶段也行。

player.isPhaseUsing(notmeisok)

玩家和目标交换装备区的牌。

player.swapEquip(target)

能否对目标发起拼点

canCompare(target)

返回bool

废除目标某个装备栏

target.disableEquip(pos)

pos填1-6的数字

1:'武器',2:'防具',3:'防御马',4:'攻击马',5：‘宝物’，6：“所有马”

废除装备栏动画

$disableEquip(skill)

恢复玩家某个装备栏

plyer.enableEquip(pos)

pos填1-6的数字

1:'武器',2:'防具',3:'防御马',4:'攻击马',5：‘宝物’，6：“所有马”

恢复装备栏动画

$enableEquip(skill)

装备区是否被废除

isDisabled(arg)

arg填1-6的数字

1:'武器',2:'防具',3:'防御马',4:'攻击马',5：‘宝物’，6：“所有马”

装备区是否为空

isEmpty(num)

num填1-6的数字

1:'武器',2:'防具',3:'防御马',4:'攻击马',5：‘宝物’，6：“所有马”

判断玩家的某个装备栏是不是“空栏”（即既没有被废除 栏内也没有装备牌） 请使用player.isEmpty()函数

例：一个装着武器的孙笨 player.isEmpty(1)函数得到的结果为false

一个废除了防具栏的陆抗 player.isEmpty(2)得到的结果为false

一个宝物栏里没有宝物牌 也没有废除宝物栏的SP黄月英 player.isEmpty(5)得到的结果为true

括号里面也可以填文字 用法和之前类似

废除玩家的判定区动画

$disableJudge()

恢复玩家的判定区动画

$enableJudge()

废除玩家的判定区

player.disableJudge()

恢复玩家的判定区

player.enableJudge()

联机 联机头像和联机名称用的函数

initOL(name,character)

初始化联机头像和联机名称用的函数

uninitOL()

是联机房间用的函数

initRoom(info,info2)

替换武将牌用的函数跟init相似

reinit(from,to,maxHp,online)

初始化武将 一键白板

uninit()

获取自己的右方位

getLeft()

获取自己的顶方位

getTop()

平整武将图片?在武将换图片的函数出现

smoothAvatar(vice,video)

改变座次？

changeSeat(position,video)

发送联机数据

send()

getId()

发送表情

throwEmotion(target,emotion)

表情

emotion(pack,id)

在人物头像上显示对话框，发送给局内所有玩家。

chat(str)

说话语音

say(str)

显示投降按钮。

showGiveup()

发送联机的技能数据

applySkills(skills)

也是发送数据用的

getState()

联机专属函数

setNickname(str)

这三个三个貌似都是换武将图片的函数

setAvatar(name,name2,video,fakeme)

setAvatarQueue(name, list)

flashAvatar(skill,name)

移除标记

removeMark(i,num,log)

i填标记名id 例如“ming\_mark”

num 填标记的数量

添加标记

addMark(i,num,log)

i填标记代码名id 例如“ming\_mark”

num 填标记的数量

得出目标所标记的数量

target.countMark(i)

i填标记代码名称

发回数字类型

是否含有某个标记

hasMark(i)

i填标记代码名称

更新标记数

updateMark(i,storage)

i填标记技能的ID

更新标记数

updateMarks(connect)

connect填需要全房间同步更新的标记技能的ID

老版本获取手牌的数量，现在不用了

num(arg1,arg2,arg3)

画指示线

line(target,config)

画指示线和line没什么区别

line2(targets,config)

返回下家

getNext()

返回上一家

getPrevious()

牌本回合使用次数

countUsed(card,type)

技能本回合发动次数。

countSkill(skill)

获取固有技能。

getStockSkills(unowned,unique,hidden)

参数：是否包含当前没有的技能。是否包含unique标签的技能。是否包含暗将技能

获取角色的卡牌

player.getCards(arg1,arg2)

arg1填手牌类型例如 'h'

arg2填一个对象或函数例如

function(card){return lib.filter.canBeDiscarded(card,player,target);}

获取可被弃置的牌。player为弃置其牌的角色，剩下的参数和getCards相同

getDiscardableCards(player,arg1,arg2)

获取可被得到的牌。player为弃置其牌的角色，剩下的参数和getCards相同

getGainableCards(player,arg1,arg2)

获取能够获得的技能

getGainableSkills(func)

记录玩家牌的数量

player.countCards(arg1,arg2)

arg1填手牌类型例如 'h'

获取目标有多少张可以被自己弃置的牌

target.countDiscardableCards(player,arg1,arg2)

获取目标有多少张可以被自己获得的牌

target.countGainableCards(player,arg1,arg2)

获取自己武将牌上的技能组

getOriginalSkills()

获取自己拥有的技能

getSkills(arg2,arg3,arg4)

很少用了它被拆成了 player.getCards 和player.getSkills

get(arg1,arg2,arg3,arg4)

刷新技能标记用的

syncStorage(skill)

这两个用的不多

syncSkills()

playerfocus(time)

插入一个额外回合

insertPhase(skill,insert)

skill填技能名，insert填是否优先执行

应该是在当前回合后插入事件什么的

insertEvent(name,content,arg)

执行一个回合。现在一般不使用了。

phase(skill)

下面是时机函数

phaseZhunbei()

phaseJudge()

phaseDraw()

phaseUse()

phaseDiscard()

phaseJieshu()

玩家选择使用某个牌

player.chooseToUse(use)

use填牌代码名称

eg：

player.chooseToUse('梅影枪：是否使用一张【杀】？',function(card){

if(get.name(card)!='sha') return false;

return lib.filter.cardEnabled.apply(this,arguments);

}).set('addCount',false).logSkill='meiyingqiang';

一般常用在时期时

eg：

enable:'chooseToUse',

选择响应打出

chooseToRespond()

一般常用在时期时

eg：

enable:['chooseToUse','chooseToRespond'],

玩家选择目标进行拼点

player.chooseToCompare(target)

和result.bool连用 如果result.bool==true则拼点成功，result.bool==false则拼点失败

选择目标的一个技能和chooseControl一个原理

chooseSkill(target)

玩家发现一张牌

player.discoverCard(list)

list填卡牌的列表

选择窗口里的卡

chooseCardButton()

这些是选择一张虚拟牌获得或使用

chooseVCardButton()

选择窗口里的卡

chooseButton()

加ol 的是在线游戏用的函数，一般会自动调用，扩展用不到

chooseButtonOL(list,callback,ai)

chooseCardOL()

玩家对目标使用牌

player.chooseUseTarget()

参考孙茹的剑影

传入一张牌，直接选择目标使用

chooseCardTarget(choose)

和chooseControl差不多，都是让用户选择

chooseControlList()

摸一张牌或恢复一点体力

chooseControl()

是否发动选项

chooseBool()

会将结果保存在result.bool

需要使用分步

括号里填提示信息 例如：player.chooseBool('是否发动【荐贤】？');

摸一张牌或恢复一点体力

chooseDrawRecover()

选择数张牌和数个目标。

choosePlayerCard()

玩家弃置目标的牌

discardPlayerCard()

eg:player.discardPlayerCard(event.targets[0],'he',true);

展示玩家的所有手牌

player.showHandcards(str)

str填范围例如"h"

展示某个或某类牌

showCards(cards,str)

str填范围例如"h"

cards填卡牌对象

观看牌

viewCards(str,cards)

玩家观看目标的手牌

player.viewHandcards(target)

是否可以移动卡牌

canMoveCard(withatt,nojudge)

player.canMoveCard(true)表示只能移动他人的装备，不能移动自己的装备

移动装备区的牌可参考张郃

moveCard()

技能用不到

useResult(result,event)

玩家对目标使用牌

player.useCard(card,target)

card里填牌的对象

例如{name:"sha"}

玩家使用某个技能

player.useSkill()

括号里填技能代码名称，注意只会执行技能content里面的内容

摸到num张牌。

drawTo(num,args)

随机弃置牌

randomDiscard()

//随机获取牌

randomGain()

弃牌

discard()

使用方法如下

cards.discard();

//响应

respond()

//交换手牌

swapHandcards(target,cards1,cards2)

//木牛牛马相关的 将牌显示在装备区，手牌区

directequip(cards)

直接获得牌

directgain(cards,broadcast,gaintag)

//将牌显示在手牌区

directgains(cards,broadcast,gaintag)

eg: player.directgains(card.cards,null,'muniu');\

张辽的突袭

gainMultiple(targets,position)

给牌

give(cards,target,visible)

失去牌

lose()

eg:

将牌放到特殊的区域

player.lose(result.cards,ui.special,'toStorage');

你的主副将体力上限之和是奇数，是否摸一张牌？

doubleDraw()

改变血量的时机函数

changeHp(num,popup)

改变护甲

changeHujia(num,type)

获取正面效果

getBuff()

获取负面效果

getDebuff()

濒死时机

dying(reason)

ai是否混乱

isMad()

unMad()

添加潜行技能直到下回合开始

tempHide()

添加身份的暴露程度

addExpose(num)

加判定

addJudge(card,cards)

能否添加判定

canAddJudge(card)

将判定牌放进下家判定区

addJudgeNext(card)

判定

judge()

武将牌翻面

turnOver(bool)

离开游戏

out(skill)

进入游戏

in(skill)

玩家横置

player.link(bool)

bool填false的话就是将玩家解除横置

跳过某阶段

skip(name)

等一段时间，联机时的等待 也用在马钧的表演

wait(callback)

unwait(result)

unprompt()

文字提示

prompt(str,nature)

应该是荀攸那样的弹出的

popup(name,className,nobroadcast)

player.popup('悲剧！'); 弹出 文本

是在头像上闪烁技能名吧这个

\_popup()

显示计时器

showTimer(time)

隐藏计时器

hideTimer()

自动刷新标记

markAuto(name,info)

markAuto

快速向storage中添加卡牌并标记(即使storage是undefined)

例：椎名【轻身】

...

'step 2'

if(result.bool&&result.cards.length){

var cards=result.cards;

player.lose(cards,ui.special,'toStorage');

player.markAuto('shiina\_qingshen',cards);

game.log(player,'将',cards,'置于武将牌上');

}

...

其中player.markAuto('shiina\_qingshen',cards);等同于以下内容

if(!player.storage.shiina\_qingshen) player.storage.shiina\_qingshen=[];

player.storage.shiina\_qingshen.addArray(result.cards);

player.markSkill('shiina\_qingshen');

unmarkAuto(name,info)

unmarkAuto

和markAuto类似 快速从storage中移出卡牌并判断数组剩余长度 若为0则移除标记

例：椎名【飞燕】

...

'step 1'

if(result.bool){

var cards=result.links;

player.unmarkAuto('shiina\_qingshen',cards);

game.cardsDiscard(cards);

player.$throw(cards);

...

}

else event.finish();

...

getStorage(name)

player.getStorage('xxx')

若player.storage.xxx不为undefined则返回player.storage.xxx 为undefined则返回空数组

例：椎名【轻身】

mod:{

attackFrom:function(from,to,num){

return num-from.getStorage('shiina\_qingshen').length;

},

maxHandcard:function(from,num){

return num+from.getStorage('shiina\_qingshen').length;

},

},

这样写可以防止因undefined导致出现NaN或报错

//技能标记

markSkill(name,info,card)

//取消技能标记

unmarkSkill(name)

//用到的不多，

markSkillCharacter(id,target,name,content)

例子： target.markSkillCharacter('sgk\_meixin',player,'魅心','本阶段当你使用一张基本牌后，该目标弃置一张牌；当你使用一张锦囊牌后，你获得该目标一张牌；当你使用一张装备牌后，你对该目标造成1点伤害。');

markCharacter(name,info,learn,learn2)

标记，老版本用的多

mark(name,info,skill)

unmark(name,info)

添加铁索

addLink()

removeLink()

玩家能否对目标使用牌

player.canUse(card,target)

canUse(card,target,distance,includecard)

带攻击范围限制的

hasUseTarget(card,distance,includecard)

应该是ai判断牌的收益

hasValueTarget()

getUseValue(card,distance,includecard)

和召唤物相关的:

获得随从

addSubPlayer(cfg)

移出随从

removeSubPlayer(name)

调遣随从

callSubPlayer()

更换随从

toggleSubPlayer()

收回和牺牲随从

exitSubPlayer(remove)

获取随从

getSubPlayers(tag)

添加技能时机

addSkillTrigger(skill,hidden,triggeronly)

添加技能，并显示在日志里

addSkillLog(skill)

添加技能

addSkill(skill,checkConflict,nobroadcast,addToSkills)

为某技能添加附属技能

addAdditionalSkill(skill,skills,keep)

移除某技能的附属技能

removeAdditionalSkill(skill,target)

恢复限定技

restoreSkill(skill,nomark)

//禁用技能

disableSkill(skill,skills)

//启用技能

enableSkill(skill)

检查并清除无效标记

checkMarks()

装备时设置触发器

addEquipTrigger(card)

装备时删除触发器

removeEquipTrigger(card)

删除技能触发器

removeSkillTrigger(skill,triggeronly)

删除技能

removeSkill(skill)

skill填技能代码名

玩家添加技能到什么时机后失效

player.addTempSkill(skill,expire,checkConflict)

skill----技能代码名字

expire---技能什么时候消失的时机

attitudeTo(target)

玩家清空所有技能

player.clearSkills(all)

检查冲突技能。一般外面不用调用。

checkConflict(skill)

获取一轮内历史的函数 一般可以查看出了哪些牌

getHistory(key,filter,last)

作用是获取玩家当前回合内使用/打出卡牌的所有事件 以及获取玩家当前回合跳过过的阶段

例1 沙摩柯【蒺藜】

...

filter:function(event,player){

return player.getHistory('useCard').length+player.getHistory('respond').length==player.getAttackRange();

},

...

括号里填需要获取的行动的类型即可

目前支持2种事件类型:useCard(使用牌) respond(打出牌)

以一种字符串类型及skipped(本回合跳过过的阶段)

函数返回得到的结果为数组型数据

例2 刘琦【屯江】中判断本回合是否跳过了出牌阶段的部分

...

if(player.getHistory('skipped').contains('phaseUse')) return false;

...

skipped返回的数组内 包含角色本回合跳过的阶段的名称

注意和player.skipList进行区分

例3 魏蔡文姬【默识】

...

filter:function(event,player){

return player.getHistory('useCard',function(evt){

return evt.isPhaseUsing()&&['basic','trick'].contains(get.type(evt.card));

}).length>0&&player.countCards('h')>0;

},

...

可以通过在getHistory函数中添加一个函数作为参数 来对事件进行过滤

顺便 在本次更新中 一个事件也可以用isPhaseUsing()函数 判断其是否发生在出牌阶段

玩家是否有当前回合内使用/打出卡牌的所有事件

player.hasHistory(key,filter,last)

getLastHistory(key,filter,last)

getLastHistory

获取一名角色最后一个自己的回合的actionHistory

目前暂无应用实例

类似getHistory 只不过获取的是整局游戏而不仅仅是当前回合的数据

例：Yui喵的【珍宝】判断整局游戏中因【激昂】获得过的牌的数量

var draw=0;

player.getAllHistory('gain',function(evt){

if(evt.getParent(2).name=='yui\_jiang') draw+=evt.cards.length;

});

return draw>=3;

获取玩家所有历史回合内使用/打出卡牌的所有事件

player.getAllHistory(key,filter,last)

玩家是否有所有历史回合内使用/打出卡牌的所有事件

player.hasAllHistory(key,filter,last)

获取玩家本回合内使用倒数第X+1张牌的事件 不填默认为0

player.getLastUsed(X)

getStat(key)

例子：

player.getStat().card.sha--; 出杀次数+1

/\*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）

//千里之视技能使用次数加一

player.getStat().skill.mengzhanc\_千里之视--

//第二种写法

player.getStat('skill').贤者之石--

player.getStat('damage')

getLastStat(key)

getLastStat

用法与getLastHistory类似 但获取的是stat而不是getHistory

例：孙綝孙峻【险害】比较当前回合角色回合内造成的伤害点数和自己上回合造成的伤害点数：

...

filter:function(event,player){

return event.source&&event.source!=player&&event.source.isAlive()&&event.source==\_status.currentPhase&&

(event.source.getStat('damage')||0)>(player.getLastStat('damage')||0)&&

!player.hasSkill('nsxianhai\_round');

},

...

参数写法与getStat一致

queue(time)

获取某张牌还能使用的次数，适用于有回合次数限制的牌

player.getCardUsable(card,pure)

获取玩家的攻击范围

player.getAttackRange(raw)

返回int类型

玩家计算与其他角色的距离增加量

player.getGlobalFrom()

其他角色计算与玩家距离的增加量

player.getGlobalTo()

获取手牌上限

player.getHandcardLimit()

获取玩家的敌人

player.getEnemies(func)

获取玩家的友军

player.getFriends(func)

目标是否是玩家的敌人

player.isEnemyOf(target)

目标是否是玩家的友方

player.isFriendOf(target)

这两函数和上面差不多功能

player.isFriendsOf(target)

player.isEnemiesOf(target)

玩家是否活着

player.isAlive()

玩家是否死了

player.isDead()

玩家是否处于濒死状态

player.isDying()

玩家是否受伤

player.isDamaged()

玩家是否是全场上最大的血量

player.isMaxHp(equal)

玩家是否是全场上最小的血量

player.isMinHp(equal)

玩家是否没扣过血

player.isHealthy()

玩家牌数是否是全场上最大的

player.isMaxCard(equal)

玩家牌数是否是全场上最小的

player.isMinCard(equal)

玩家手牌数是否是全场上最大的

player.isMaxHandcard(equal)

玩家手牌数是否是全场上最小的

player.isMinHandcard(equal)

玩家装备数是否是全场上最大的

player.isMaxEquip(equal)

玩家装备数是否是全场上最小的

player.isMinEquip(equal)

玩家是否被铁索

player.isLinked()

玩家是否翻面

player.isTurnedOver()

玩家是否移出游戏

player.isOut()

第二个战旗函数。

player.isMin(distance)

玩家是否进入游戏

player.isIn()

是否国战潜伏

player.isUnseen(num)

是否玩家操控

player.isUnderControl(self,me)

是否联机

player.isOnline()

player.isOnline2()

是否退出联机

player.isOffline()

国战检查技能是否明置。

checkShow(skill,showonly)

是否需要弃牌，用在ai上面

needsToDiscard(num)

玩家到目标的距离

distanceTo(target,method)

目标到玩家的距离

distanceFrom(target,method)

是否有技能

hasSkill(skill,arg2,arg3,arg4)

一般只填skill （技能代码名）

返回可以窃取的技能。

hasStockSkill(skill,arg1,arg2,arg3)

是否有主公技

hasZhuSkill(skill,player)

一般只填skill （技能代码名）

是否有全局技能标签

player.hasGlobalTag(tag,arg)

玩家是否有tag技能标签 一般配合ai

player.hasSkillTag(tag,hidden,arg,globalskill)

例如：player.hasSkillTag('unequip2') unequip2是防具失效

玩家判定区域是否有某个延时锦囊

player.hasJudge(name)

玩家判定区域是否有乐不

plaeyr.hasJudge('lebu')

玩家是否有队友

player.hasFriend()

如果场上有对其而言身份不清晰的角色，返回true

player.hasUnknown(num)

目标是否不知道（武将牌/身份）玩家 一般用在ai

target.isUnknown(player)

玩家是否有无懈可击

player.hasWuxie()

玩家是否有杀

player.hasSha(respond,noauto)

玩家是否有闪

player.hasShan()

player.mayHaveShan()

玩家是否有某类型的牌

player.hasCard(name,position)

玩家是否有红色的手牌

player.hasCard(function (card) {

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

})

玩家能否装备某个装备

player.canEquip(name,replace)

name填装备的代码名称

获取玩家的装备

player.getEquip(name)

name填1-6的数字

1武器 2 防具 3防御马 4进攻马 5宝物 6所有马

玩家获取判定区的牌的对象

player.getJudge(name)

例子

target.discard(target.getJudge('lebu'));

给角色设置这一函数 从而在特定场合摸牌和判定时（如智斗三国的个人牌堆）改变牌堆顶牌的获取方式

player.getTopCards();

玩家选择目标

player.chooseTarget(mas1,mas2,bool,func)

mas1填提示信息1 字符串

mas2填技能描述 字符串

bool 填true或false（可不填）true代表没有取消按钮，false代表有取消按钮

func 是一个筛选函数必须返回一个bool值

玩家所选择的目标对象存在result.targets[0]

玩家交一张牌给目标

player.chooseCard('h','将一张手牌交给'+get.translation(target),true);

必须配合gain函数

if (result.cards) {targets[0].gain(result.cards,player,"give");}

玩家观看目标的手牌并获得一张牌

player.gainPlayerCard(1,'h',target,true,'visible')

第一个参数填数量是number类型，第二个填手牌类型是字符串，第三个填bool，第四个填字符串（visible为观看手牌）可以不填

1.player.draw(x) 玩家摸x张牌，x∈[-Infinity,Infinity]，x可不填，不填时默认为1

2.player.recover(x) 玩家回复x点体力，x∈[-Infinity,Infinity]，x可不填，不填时默认为1

3.player.damage(a,'b') 玩家受到a点b属性伤害，b为thunder是时雷属性，b为fire时是火属性

4.player.loseHp(x) 玩家流失x点体力，x∈[-Infinity,Infinity]，x可不填，不填时默认为1

5.player.gainMaxHp(x) 玩家体力上限+x，x为Infinity时游戏会出现卡机，x可不填，不填时默认为1

6.player.loseMaxHp(x) 玩家体力上限-x，x为Infinity时游戏会出现卡机，x可不填，不填时默认为1

7.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

8.player.discard(a) 玩家选择失去a张牌

9.player.turnOver(a) a不填时玩家翻面，a填false时玩家解除翻面，填其他时未尝试

9.player.link(a) a不填时玩家横置，a填false时玩家解除横置，填其他时未尝试

10.player.remove() 玩家移出游戏，回不来的这种

11.player.out() 玩家离开游戏

12.player.die() 玩家立刻死亡

13.player.delete() 神圣死亡

14.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');

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

16.player.clearSkills() 玩家清空所有技能

17.player.update() 刷新

18.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

19.player.goMad() 玩家陷入混乱

20.player.skip('a')玩家跳过a阶段，如a为phaseJudge时，跳过判定阶段

21.player.init('a') 将玩家的武将牌替换为a，a为武将名

22.player.addSkill('a') 玩家获得a技能，a为技能名

23.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，如phaseAfter

24.player.revive() 玩家复活

25.player.logSkill('a',b) 游戏记录里出现 玩家+a+b 的说明，其中a为字符串，可以打任意汉字，b为之前代码中已定义的事件或时点或角色，也可不填

26 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

player.gain(targetCard.randomGet(),target,'giveAuto');

或的的牌对象 目标 自动获得，如果填'give' 就是目标选择牌给你

player.gain(player.storage.xxx,'gain2','fromStorage');

这是从游戏外获得牌时的代码 并且前面两项大家应该都懂

27 player.changeHujia(a); 玩家获得a点护甲

player.skip('phaseDiscard');//跳过弃牌阶段

角色直接装备装备牌

player.equip(card)

eg：

var card=game.createCard('jydiy\_xiuhuazhen'); //创造出一张牌

player.equip(card)

eg：

//玩家装备绣花针

game.me.next.equip(game.createCard('jydiy\_xiuhuazhen'))

jydiy\_shenghuoling

//玩家装备圣火令

game.me.equip(game.createCard('jydiy\_shenghuoling'))

game.me.equip(game.createCard('zhuge'))

jydiytaohuazhen\_re

//玩家装备桃花阵

game.me.equip(game.createCard('jydiytaohuazhen'))

game.me.equip(game.createCard('jydiybeidouzhen'))

game.me.equip(game.createCard('jydiy\_jingsibeixin'))

//玩家装备九阳真经

game.me.equip(game.createCard('jydiy\_jiuyangzhengjing'))

//玩家装备倚天剑

game.me.equip(game.createCard('jydiy\_yitianjian'))

//装备加一马

game.me.equip(game.createCard('dilu'))

弃掉对方的牌

player.discardPlayerCard(target,3,'visible','h');

选择弃牌

player.chooseToDiscard(4,true)

获得某张牌

game.me.gain(game.createCard({name:'sha',nature:'jy\_du'})) //获得毒杀

game.me.gain(game.createCard({name:'sha',nature:'jy\_xie'}))

game.me.gain(game.createCard({name:'sha',nature:'fire'})) // 获得火杀

game.me.gain(game.createCard({name:'tao'})) //获得桃

game.me.gain(game.createCard({name:'taoyuan'})) //获得桃园

game.me.gain(game.createCard({name:'shunshou'})) //获得顺手

game.me.gain(game.createCard({name:'juedou'})) //获得决斗

game.me.gain(game.createCard({name:'bingliang'})) //获得兵粮寸断

game.me.next.gain(game.createCard({name:'wugu'})) //获取五谷

game.me.next.gain(game.createCard({name:'shan'})) //下一个玩家获得闪

//玩家获得桃花阵卡牌

game.me.gain(game.createCard({name:'jydiytaohuazhen'}))

game.me.gain(game.createCard({name:'sha',nature:'jy\_xie'}))

game.me.previous.getHistory('damage', function (evt) {

console.log(evt.getParent(0));

console.log(evt.getParent(1));

console.log(evt.getParent(2));

console.log(evt.getParent(3));

console.log(evt.getParent(4));

console.log(evt.getParent(5));

console.log(evt.getParent(6));

console.log(evt.getParent(7));

return evt.getParent(2);

}).length > 0

player.getHistory('useCard') //获取出牌阶段使用的牌 （括号里面填阶段）

player.getHistory('useCard').length;//出牌阶段使用的牌的数量

player.getAllHistory('useCard').length;//游戏开始后使用的牌的总数量

}

}

frequent:true,//自动发动

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

forced:true,//锁定技

direct:true,//等于上面两个相加

popup:false,//不弹出提示(也就是发动技能时不提示发动技能名)

silent:true,//等于上面两个相加

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体力（即是血量）， maxHp体力上限， damage受伤 ， lose失去， recover回复体力， discard弃置牌， draw摸牌， step 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，

skill技能， add增加……

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 equip6所有马

# game操作函数

game{

控制台的换人

game.swapPlayer(target);

游戏结束

game.over(bool)

bool填true表示游戏胜利

bool填false表示游戏失败

设置同时失去牌的函数,例如拼点

loseAsync(arg)

loseAsync

用于处理失去牌后需要二次操作的情况和需要多名角色同时失去牌的情况的事件 时机为loseAsyncAfter 用getl获取失去牌

根据需要设置自定义content

game.createCard2

用法和game.createCard完全一致 只不过生成的卡牌洗牌时不会消失

}

card{

//卡牌是否添加了文字标记

card.hasGaintag(strid)

strid填技能代码名字

}

# Get对象中的函数

get{

get.strNumber(num)

若num为1/11/12/13 则返回A/J/Q/K

否则返回num.toString()

//获取目标离自己的距离是否为1

get.distance(player, target) <= 1

get.isLuckyStar()

用于判断是否开启了【幸运星模式】的选项

然后是两个国战模式专用的函数

get.guozhanRank()

用于判断一个武将在国战模式下的强度评级

评级有10个级别 从-1,0,1...一直到6,7,8 级别越高说明武将强度越强

士兵会被判断为-1 非国战原生武将会被判断为0 君主会被判断为7

其他武将将根据自己在lib.guozhanRank（若为鏖战模式 则改为优先以lib.aozhanRank）中的位置来判断强度

例：get.guozhanRank('gz\_xunyou')

判断荀攸的强度 结果无论鏖战还是非鏖战都为8

另外 强度小于4的武将是会被AI荀攸马谡吕范直接换掉的

get.guozhanReverse(xxx,yyy)

用于AI判断是否要调整主副将顺序的函数 xxx为主将 yyy为副将 当AI觉得要换时返回值为true 否则为false

荀攸吕范必定会被AI放在主将

其次是是搭配三血武将的邓艾

搭配 三血将/周泰 的 姜维/孙策 则必定会被AI放在副将

关于处理区：

不知道处理区是什么的同学们 请自行查阅凌天翼规则集相关内容太长了我懒得贴

处理区在无名杀的代码为ui.ordering

为方便兼容旧扩展 使用get.position(card)方法读取处理区的卡牌 默认得到的仍然是弃牌堆（'d'）

使用get.position(card,true) 才会得到处理区（'o'）的结果

getLastUsed(X)

获取玩家本回合内使用倒数第X+1张牌的事件 不填默认为0

例：留赞【奋音】

...

filter:function(event,player){

var evt=player.getLastUsed(1);

if(!evt) return false;

var color1=get.color(evt.card);

var color2=get.color(event.card);

return color1&&color2&&color1!='none'&&color2!='none'&&color1!=color2;

},

...

}

关于get.effect和get.effect\_use

get.effect用于获取直接收益

比如说加目标 桃子全场多开就完事了

get.effect\_use则需要考虑【桃】的存留等问题

【桃】的收益函数里分为了target:function和target\_use:function 对应get.effect和get.effect\_use

get.is.yingbian(card)

用于判断一张牌能否被应变

}

# Mod函数

//写在武将的技能中和content同级别

mod:{

//无限范围

targetInRange:function(card){

if(card.storage.bjzm孤走装备变杀) return true;

},

//无限次数

cardUsable:function(card){

if(card.storage.bjzm孤走装备变杀) return Infinity;

},

//将装备牌变杀

cardname:function(card){

if (get.type(card, null, false) == 'equip') {

card.storage.bjzm孤走装备变杀=true;

return 'sha';

}

},

//不计入手牌数

ignoredHandcard:function(card,player){

if(card.hasGaintag('bjzm集铩')){

return true;

}

},

},

# 零散教程

----------------------------------------

步骤停止

event.finish();

eg：

"step 3"

if(result.control!='cancel2'){

。。。。。

}

else{

event.finish();//有了这行代码，如果找执行else，就会停止步骤，不执行下面的step 4 等后面的内容

}

"step 4"

...

"step ..."

---------------------------------------------

date为需要输出的数据 name为.txt结尾的文档名称

//将内容保存到txt文件

function exportRaw(name, data) {

　　　　var urlObject = window.URL || window.webkitURL || window;

　　　　var export\_blob = new Blob([data]);

　　　　var save\_link = document.createElementNS( "a")

　　　　save\_link.href = urlObject.createObjectURL(export\_blob);

　　　　save\_link.download = name;

　　　　fakeClick(save\_link);

　　}

//删除卡牌的技能 删除card触发技能的代码

var info=get.info(card);

if(info.skills){

for(var i=0;i<info.skills.length;i++){

player.removeSkillTrigger(info.skills[i]);

}

}

//添加卡牌的技能 添加card触发技能的代码

var info=get.info(card);

if(info.skills){

for(var i=0;i<info.skills.length;i++){

player.addSkillTrigger(info.skills[i]);

}

}

//终极锁定技，加charlotte不会被封掉技能

charlotte: true,

想让一名角色能和陆抗一样 不能成为拼点目标 请参考下方写法：

ai:{

noCompareTarget:true,

},

新版本在执行shaBegin等和杀有关的时机时 可以直接在此步骤调整【杀】伤害 而不需要建立额外的技能

例：新黄忠【烈弓】的部分代码

.........

trigger:{player:'shaBegin'},

.........

content:function(){

.........

if(typeof trigger.extraDamage!='number'){

trigger.extraDamage=0;

}

trigger.extraDamage++;

.........

},

在useCard等时机也可以对【杀】的伤害进行调整

和上面的时机区别在于：如果一张【杀】指定了多个目标 在useCard相关的时机修改【杀】的伤害值时对所有的目标角色都有效 而上面的只对特定的目标有效

例：新版本【酒】的部分代码：

trigger:{player:'useCard'},

filter:function(event){

return event.card&&event.card.name=='sha';

},

forced:true,

content:function(){

if(!trigger.baseDamage) trigger.baseDamage=1;

trigger.baseDamage+=player.storage.jiu;

.........

},

众所周知 有些技能 看起来什么时候都能发动 然而在OL上却只能在出牌阶段内发动如徐盛的【破军】 马云禄的【凤魄】等

判断一名角色是否处在出牌阶段内 使用player.isPhaseUsing()判断即可

关于自定义势力颜色的问题

在以前 自定义势力的颜色需要修改get.groupnature()函数

而现在不需要了 在lib.groupnature里添加需要的势力颜色映射即可

例：新版本「Key杀」中自定义【键】势力的相关代码

//设置势力的颜色

//这个步骤是在样式库中写入需要的势力颜色 rgba里面的四个数字就代表颜色的RGB值和透明度

var style2=document.createElement('style');

style2.innerHTML=".player .identity[data-color='key'],";

style2.innerHTML+="div[data-nature='key'],";

style2.innerHTML+="span[data-nature='key'] {text-shadow: black 0 0 1px,rgba(203, 177, 255,1) 0 0 2px,rgba(203, 177, 255,1) 0 0 5px,rgba(203, 177, 255,1) 0 0 10px,rgba(203, 177, 255,1) 0 0 10px}";

style2.innerHTML+="div[data-nature='keym'],";

style2.innerHTML+="span[data-nature='keym'] {text-shadow: black 0 0 1px,rgba(203, 177, 255,1) 0 0 2px,rgba(203, 177, 255,1) 0 0 5px,rgba(203, 177, 255,1) 0 0 5px,rgba(203, 177, 255,1) 0 0 5px,black 0 0 1px;}";

style2.innerHTML+="div[data-nature='keymm'],";

style2.innerHTML+="span[data-nature='keymm'] {text-shadow: black 0 0 1px,rgba(203, 177, 255,1) 0 0 2px,rgba(203, 177, 255,1) 0 0 2px,rgba(203, 177, 255,1) 0 0 2px,rgba(203, 177, 255,1) 0 0 2px,black 0 0 1px;}";

document.head.appendChild(style2);

//在lib.groupnature中建立相应的映射

lib.groupnature.key='key';

//将势力添加到势力库中 并指定势力的中文名称

lib.group.push('key');

lib.translate['key']='键';

game.cardsDiscard();

这个函数的作用是「将不属于任何人的卡牌置入弃牌堆」

把要弃置的卡牌填进括号就行

将玩家的技能清空

player.addSkill("baiban")

有charlotte: true,还是清不掉

另外关于新版本卡牌强命的方法

除原有的wuxieRespondable的mod和norespond的skillTag之外 还可以通过以下的方法实现卡牌强命/不可无懈

例1：麴义【伏骑】

...

content:function(){

trigger.directHit.addArray(game.filterPlayer(function(current){

return current!=player&&get.distance(current,player)<=1;

}));

},

...

useCard事件的directHit列表 即为【不能使用或打出牌响应卡牌】的角色

想让此牌完全不能被响应？把场上所有角色都加进这个列表就行了

例2:SP贾诩【缜略】

...

content:function(){

trigger.nowuxie=true;

},

...

只想让卡牌不能被无懈？直接把nowuxie属性设置为true即可

另外 这一属性也可以在useCard函数的括号内直接设置

例3：貂蝉【离间】

...

targets[1].useCard({name:'juedou'},'nowuxie',targets[0],'noai').animate=false;

...

那个nowuxie就是关键

useCard流程中的customArgs

这个恕我难以使用语言描述不如贴个例子来的实在

例：黄忠【烈弓】部分代码

trigger:{player:'useCardToTargeted'},

logTarget:'target',

check:function(event,player){

return get.attitude(player,event.target)<=0;

},

filter:function(event,player){

if(event.card.name!='sha') return false;

if(event.target.countCards('h')<=player.countCards('h')) return true;

if(event.target.hp>=player.hp) return true;

return false;

},

content:function(){

if(trigger.target.countCards('h')<=player.countCards('h')) trigger.getParent().directHit.push(trigger.target);

if(trigger.target.hp>=player.hp){

var id=trigger.target.playerid;

var map=trigger.getParent().customArgs;

if(!map[id]) map[id]={};

if(typeof map[id].extraDamage!='number'){

map[id].extraDamage=0;

}

map[id].extraDamage++;

}

},

新版本出牌阶段主动技的新参数（均仅在discard为false且lose不为false时有效）

例：卡牌重铸类技能：

...

visible:true,

loseTo:'discardPile',

...

visible参数是让角色失去卡牌的过程中强制视为正面朝上失去

loseTo则是指定将卡牌的去向（默认为.ui.special）

新版本的judge事件中 可以通过设置callback事件 在judgeEnd和judgeAfter时机之前对判定牌进行操作 避免被曹植截胡

例：黄巾雷使【助祭】：

...

content:function(){

trigger.source.judge().callback=lib.skill.fuji.callback;

},

callback:function(){

var evt=event.getParent(2);

if(event.judgeResult.color=='black'){

game.cardsDiscard(card);

evt.\_trigger.num++;

}

else{

evt.\_trigger.source.gain(card,'gain2');

}

},

...

想把卡牌送进处理区 有两个方法

对于有主的卡牌 直接player.lose(cards,ui.ordering)伺候即可

无主的卡牌则是通过game.cardsGotoOrdering()解决 用法和那两个一样

想不插入卡牌置入弃牌堆的事件 请设置事件的noOrdering属性

在1.9.97.8.1或更高版本中 在lose事件里 可以直接通过event.hs/es/js 来判断一张卡牌在此次失去事件中原先所处的区域

例：陆逊【连营】：

filter:function(event,player){

if(player.countCards('h')) return false;

return event.hs&&event.hs.length;

},

三个参数均为包含卡牌的数组

相比使用card.original判断卡牌来源而言 这种方法不会受到其他插入结算的影响 结果更可靠

game.getGlobalHistory();

类似于getHistory() 用于获取本回合内发生过的 不属于任何角色的一些事件

目前仅支持cardMove参数（cardsDiscard cardsGotoOrdering cardsGotoSpecial等涉及卡牌位置改变的事件）

例：张昭张纮【固政】

var cards=[];

var cards2=[];

game.getGlobalHistory('cardMove',function(evt){

if(evt.name=='cardsDiscard'&&evt.getParent('phaseDiscard')==trigger) cards.addArray(evt.cards.filterInD('d'));

});

在1.9.99.1版本更新之后，所有角色经历的lose事件会同时记录到globalHistory中

例：岩泽雅美【奋音】

...

filter:function(event,player){

if(player!=\_status.currentPhase) return false;

if(event.name=='lose'&&event.position!=ui.discardPile) return false;

var list=[];

for(var i=0;i<event.cards.length;i++){

var card=event.cards[i];

list.add(card.suit);

}

game.getGlobalHistory('cardMove',function(evt){

if(evt==event||(evt.name!='lose'&&evt.name!='cardsDiscard')) return false;

if(evt.name=='lose'&&evt.position!=ui.discardPile) return false;

for(var i=0;i<evt.cards.length;i++){

var card=evt.cards[i];

list.remove(card.suit);

}

});

return list.length>0;

},

...

event.relatedLose

这是gain事件中 如果获得其他角色的卡牌时 其他角色失去卡牌的事件

例：蒋干【伪诚】

filter:function(event,player){

if(event.source==player&&event.player!=player&&player.hp>player.countCards('h')){

return event.relatedLose&&event.relatedLose.hs&&event.relatedLose.length>0;

}

return false;

},

现在cardsGotoOrdering函数 可以通过指定一个事件而不一定是强制使用当前事件 在这个事件之后丢弃所有还在处理区未被移动的卡牌

例：五谷丰登：

contentBefore:function(){

...

var cards=get.cards(num);

game.cardsGotoOrdering(cards).relatedEvent=event.getParent();

...

}

通过将relatedEvent设置为当前事件的parent（即useCard） 在这一事件结束时而不是contentBefore结束时再丢弃所有卡牌

在1.9.98.2以后的新版本中 useCard过程中中的event.card 不再是简单暴力地使用event.cards[0] 而是重新生成的一个{}

在这种情况下 判断这张牌是否为【转化】的卡牌的方法如下：

...

filter:function(event){

return (get.type(event.card,'trick')=='trick'&&event.card.isCard);

},

...

当event.card.isCard为true时 卡牌为非转化卡牌 否则为转化卡牌

content:function(){

...

trigger.changeToZero();

...

},

为什么要使用changeToZero 而不是直接trigger.cancel呢？

因为直接cancel的话 会导致后面的摸牌阶段结束时等时机直接无法生成司马昭队友直接哭了

同时 要判断一个出牌阶段「有没有被放弃摸牌」的话 需要使用下面的代码

filter:function(event,player){

return !event.numFixed;

},

两种特效的设置方式 都是通过在lib.animate中添加函数 来实现特定卡牌的特效

例：设置【杀】的自定义特效

lib.animate.card.sha=(card,name,nature){

this.chat('杀一下');

}

这样的话 使用【杀】的时候便不会弹出文字 而是喊一句「杀一下」

this为卡牌的使用/打出者 card为卡牌 nature为不使用特效的情况下卡牌的文字颜色(metal-使用-黄色，wood-打出-绿色)，可以用来判断卡牌是因使用还是因打出而播放特效

当然这里的示例很简单 你也可以自己整些高级的

技能特效的设置方式类似

例：设置孙笨【激昂】的特效

lib.animate.skill.jiang=function(name,popname,checkShow){

this.chat('吾乃江东的小霸王，孙伯符！')

}

name为技能名称 popname为原先发动技能时弹出的文字（绝大多数情况下与name相同） checkShow为双将模式下技能的来源（vice为主将 其他情况下为副将）

新版本中 get.name, info, suit, color, type等函数 可以通过添加参数，判断一张牌在一名角色手牌区时的牌面信息

例：假设card是一张红桃闪

get.name(card,player);

如果是player这种普通人 返回的依然是shan

但如果player是神关羽 返回的就是sha了

player不填时默认视为卡牌的拥有者 填false时不做判断

另外 get.type的时候要注意

card和player之间记得加一个null 或者在需要把延时锦囊牌视为trick时把null改为'trick'

get.type(card,null,player)

函数式viewAs

以神赵云【龙魂】举例

relonghun:{

audio:'longhun',

//技能发动时机

enable:['chooseToUse','chooseToRespond'],

//发动时提示的技能描述

prompt:'将♦牌当做杀，♥牌当做桃，♣牌当做闪，♠牌当做无懈可击使用或打出',

//动态的viewAs

viewAs:function(cards,player){

var name=false;

var nature=null;

//根据选择的卡牌的花色 判断要转化出的卡牌是闪还是火杀还是无懈还是桃

switch(get.suit(cards[0],player)){

case 'club':name='shan';break;

case 'diamond':name='sha';nature='fire';break;

case 'spade':name='wuxie';break;

case 'heart':name='tao';break;

}

//返回判断结果

if(name) return {name:name,nature:nature};

return null;

},

//AI选牌思路

check:function(card){

if(ui.selected.cards.length) return 0;

var player=\_status.event.player;

if(\_status.event.type=='phase'){

var max=0;

var name2;

var list=['sha','tao'];

var map={sha:'diamond',tao:'heart'}

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

var name=list[i];

if(player.countCards('he',function(card){

return (name!='sha'||get.value(card)<5)&&get.suit(card,player)==map[name];

})>0&&player.getUseValue({name:name,nature:name=='sha'?'fire':null})>0){

var temp=get.order({name:name,nature:name=='sha'?'fire':null});

if(temp>max){

max=temp;

name2=map[name];

}

}

}

if(name2==get.suit(card,player)) return (name2=='diamond'?(5-get.value(card)):20-get.value(card));

return 0;

}

return 1;

},

//选牌数量

selectCard:[1,2],

//确保选择第一张牌后 重新检测第二张牌的合法性 避免选择两张花色不同的牌

complexCard:true,

//选牌范围：手牌区和装备区

position:'he',

//选牌合法性判断

filterCard:function(card,player,event){

//如果已经选了一张牌 那么第二张牌和第一张花色相同即可

if(ui.selected.cards.length) return get.suit(card,player)==get.suit(ui.selected.cards[0],player);

event=event||\_status.event;

//获取当前时机的卡牌选择限制

var filter=event.\_backup.filterCard;

//获取卡牌花色

var name=get.suit(card,player);

//如果这张牌是梅花并且当前时机能够使用/打出闪 那么这张牌可以选择

if(name=='club'&&filter({name:'shan',cards:[card]},player,event)) return true;

//如果这张牌是方片并且当前时机能够使用/打出火杀 那么这张牌可以选择

if(name=='diamond'&&filter({name:'sha',cards:[card],nature:'fire'},player,event)) return true;

//如果这张牌是黑桃并且当前时机能够使用/打出无懈 那么这张牌可以选择

if(name=='spade'&&filter({name:'wuxie',cards:[card]},player,event)) return true;

//如果这张牌是红桃并且当前时机能够使用/打出桃 那么这张牌可以选择

if(name=='heart'&&filter({name:'tao',cards:[card]},player,event)) return true;

//上述条件都不满足 那么就不能选择这张牌

return false;

},

//判断当前时机能否发动技能

filter:function(event,player){

//获取当前时机的卡牌选择限制

var filter=event.filterCard;

//如果当前时机能够使用/打出火杀并且角色有方片 那么可以发动技能

if(filter({name:'sha',nature:'fire'},player,event)&&player.countCards('he',{suit:'diamond'})) return true;

//如果当前时机能够使用/打出闪并且角色有梅花 那么可以发动技能

if(filter({name:'shan'},player,event)&&player.countCards('he',{suit:'club'})) return true;

//如果当前时机能够使用/打出桃并且角色有红桃 那么可以发动技能

if(filter({name:'tao'},player,event)&&player.countCards('he',{suit:'heart'})) return true;

//如果当前时机能够使用/打出无懈可击并且角色有黑桃 那么可以发动技能

if(filter({name:'wuxie'},player,event)&&player.countCards('he',{suit:'spade'})) return true;

return false;

},

ai:{

respondSha:true,

respondShan:true,

save:true,

//让系统知道角色“有杀”“有闪”“有桃”

skillTagFilter:function(player,tag){

var name;

switch(tag){

case 'respondSha':name='diamond';break;

case 'respondShan':name='club';break;

case 'save':name='heart';break;

}

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

},

//AI牌序

order:function(item,player){

if(player&&\_status.event.type=='phase'){

var max=0;

var list=['sha','tao'];

var map={sha:'diamond',tao:'heart'}

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

var name=list[i];

if(player.countCards('he',function(card){

return (name!='sha'||get.value(card)<5)&&get.suit(card,player)==map[name];

})>0&&player.getUseValue({name:name,nature:name=='sha'?'fire':null})>0){

var temp=get.order({name:name,nature:name=='sha'?'fire':null});

if(temp>max) max=temp;

}

}

max/=1.1;

return max;

}

return 2;

},

},

//让系统知道玩家“有无懈”

hiddenCard:function(player,name){

return name=='wuxie'&&player.countCards('he',{suit:'spade'})>0;

},

group:['xinlonghun\_num','xinlonghun\_discard'],

},

\_status.pileTop

牌堆顶的一张牌 随game.updateRoundNumber()函数的运行实时更新

关于界太史慈【酣战】随机拼点

本质为「通过其他方式提前选择拼点牌」

hanzhan:{

audio:2,

trigger:{

global:'chooseToCompareBegin',

},

filter:function(event,player){

if(player==event.player) return true;

if(event.targets) return event.targets.contains(player);

return player==event.target;

},

logTarget:function(event,player){

if(player!=event.player) return event.player;

return event.targets||event.target;

},

check:function(trigger,player){

var num=0;

var targets=player==trigger.player?(trigger.targets?trigger.targets.slice(0):[trigger.target]):[trigger.player];

while(targets.length){

var target=targets.shift();

if(target.getCards('h').length>1) num-=get.attitude(player,target);

}

return num>0;

},

content:function(){

var targets=player==trigger.player?(trigger.targets?trigger.targets.slice(0):[trigger.target]):[trigger.player];

if(!trigger.fixedResult) trigger.fixedResult={};

while(targets.length){

var target=targets.shift();

var hs=target.getCards('h');

if(hs.length) trigger.fixedResult[target.playerid]=hs.randomGet();

}

},

},

令其他角色的手牌对自己可见

只需令自己拥有viewHandcard的技能标签即可

例：水濑秋子【洞察】

ai:{

viewHandcard:true,

skillTagFilter:function(player,tag,arg){

if(player==arg) return false;

},

}

arg为目标 通过调整skillTagFilter即可实现对特定角色的手牌可见

关于dynamicTranslate

指定lib.dynamicTranslate.xxx为一个函数 即可让技能xxx显示的描述随玩家状态而变化 并实现技能修改等

例：SP姜维【困奋】

kunfen:function(player){

if(player.storage.kunfen) return '结束阶段开始时，你可以失去1点体力，然后摸两张牌。';

return '锁定技，结束阶段开始时，你失去1点体力，然后摸两张牌。';

},

第一个参数为技能的拥有者

在一个judge判定事件中，如果是因为卡牌导致的，则可以通过event.cardname判断卡牌名称

例：界刘禅【思蜀】

...

mod:{

judge:function(player,result){

if(\_status.event.cardname=='lebu'&&player.countMark('sishu2')%2==1){

if(result.bool==false){

result.bool=true;

}

else{

result.bool=false;

}

}

}

},

...

关于getl机制

由于无名杀没有正规的牌移动事件 因此在新版本中 使用getl机制进行拟合

例：孙尚香【枭姬】

...

//时机：单纯的“失去牌后”，牌被获得后，牌被人直接插进判定区后，牌被人拿进装备区后

trigger:{

player:'loseAfter',

global:['equipAfter','addJudgeAfter','gainAfter'],

},

frequent:true,

//通过getl函数 获取玩家在事件过程中“失去过”的牌，同时排除因其他事件导致的lose事件 避免二次发动技能

filter:function(event,player){

var evt=event.getl(player);

//只有获取的列表存在 并且有失去装备区的牌的记录 才能发动技能

return evt&&evt.player==player&&evt.es&&evt.es.length>0;

},

\_trueMe

直接将一名角色的\_trueMe赋值给另一名角色 即可实现控制权的夺取

例：神貂蝉【惑心】

huoxin\_control:{

audio:'huoxin',

forced:true,

trigger:{global:'phaseBeginStart'},

//避免多重控制 而必须在没有\_trueMe的情况下发动

filter:function(event,player){

return player!=event.player&&!event.player.\_trueMe&&event.player.countMark('huoxin')>1;

},

logTarget:'player',

skillAnimation:true,

animationColor:'key',

content:function(){

trigger.player.removeMark('huoxin',trigger.player.countMark('huoxin'));

//夺取控制权

trigger.player.\_trueMe=player;

//加入操作切换的全局技能

game.addGlobalSkill('autoswap');

//如果是AI抢玩家

if(trigger.player==game.me){

//强制托管且不可取消

game.notMe=true;

if(!\_status.auto) ui.click.auto();

}

//增加一个技能用于换回来

trigger.player.addSkill('huoxin2');

},

},

huoxin2:{

trigger:{

player:['phaseAfter','dieAfter'],

global:'phaseBefore',

},

lastDo:true,

charlotte:true,

forceDie:true,

forced:true,

silent:true,

content:function(){

player.removeSkill('huoxin2');

},

onremove:function(player){

if(player==game.me){

//将视角切回

if(!game.notMe) game.swapPlayerAuto(player.\_trueMe)

//取消强制托管

else delete game.notMe;

if(\_status.auto) ui.click.auto();

}

delete player.\_trueMe;

},

},

在chooseButton类出牌阶段技能中调用chooseControl函数而不是chooseButton函数进行第一段选择

例：神户小鸟【花绽】

（篇幅所限，仅列出chooseButton内部部分）

...

chooseButton:{

//和chooseButton一样 直接生成对话框

dialog:function(event,player){

return ui.create.dialog('###花绽###'+lib.translate.kotori\_huazhan\_info);

},

//生成chooseButton中所有选项（记得加上cancel2！）

chooseControl:function(event,player){

var list=['wei','shu','wu','qun','key'];

var list2=[];

for(var i of list){

if(player.hasMark('kotori\_yumo\_'+i)&&!player.getStorage('kotori\_huazhan2').contains('kotori\_yumo\_'+i))list2.push('kotori\_yumo\_'+i);

}

list2.push('cancel2');

return list2;

},

//AI选择

check:function(){

var player=\_status.event.player;

var list=['wei','shu','wu','qun','key'];

var list2=[];

for(var i of list){

if(player.hasMark('kotori\_yumo\_'+i)&&!player.getStorage('kotori\_huazhan2').contains('kotori\_yumo\_'+i))list2.push('kotori\_yumo\_'+i);

}

if(list2.contains('kotori\_yumo\_wei')) return 'kotori\_yumo\_wei';

if(list2.contains('kotori\_yumo\_wu')) return 'kotori\_yumo\_wu';

if(list2.contains('kotori\_yumo\_qun')) return 'kotori\_yumo\_qun';

if(list2.contains('kotori\_yumo\_key')) return 'kotori\_yumo\_key';

if(list2.contains('kotori\_yumo\_shu')&&game.hasPlayer(function(current){

return current.group=='shu';

})) return 'kotori\_yumo\_shu';

return 'cancel2';

},

//这里的result是整个result result.control才是你选择的选项

backup:function(result,player){

return {

markname:result.control,

viewAs:{name:'kaihua',isCard:true},

filterCard:function(){return false},

selectCard:-1,

precontent:function(){

delete event.result.skill;

var name=lib.skill.kotori\_huazhan\_backup.markname;

if(!player.storage.kotori\_huazhan2) player.storage.kotori\_huazhan2=[];

player.storage.kotori\_huazhan2.push(name);

player.addTempSkill('kotori\_huazhan2');

player.popup('花绽',get.groupnature(name.slice(12)));

game.log(player,'发动了技能',lib.translate[name].replace(/魔物/g,'【花绽】'));

player.removeMark(name,1,false);

;game.log(player,'移去了一个',lib.translate[name].replace(/魔物/g,'【魔物】'));

},

}

}

},

对特定角色使用牌无次数限制

旧版本写法为加减100判断

新版本添加新的mod 用于进行直接判断

例：张绣【雄乱】

...

mod:{

targetInRange:function (card,player,target){

if(target.hasSkill('drlt\_xiongluan2')){

return true;

}

},

cardUsableTarget:function(card,player,target){

if(target.hasSkill('drlt\_xiongluan2')) return true;

},

},

...

在useCard事件中设置.throw属性 即可在使用牌时不播放扔牌特效

例：player.useCard(card).throw=false

如果要在chooseUseTarget函数中使用 则使用nothrow

例：player.chooseUseTarget(card,'nothrow');

关于武将评级

通过角色在lib.rank.rarity里的位置判断

分为legend(传说) epic(史诗) rare(稀有) junk(平凡)

例：界孙策位于lib.rank.rarity.legend，因此为传说武将

评级仅影响战棋君主模式下的武将价格 无其他实际影响

小吧主13

当一张牌的updateUsable为'phaseUse'时 会在出牌阶段结束时而而不是回合结束时更新使用数据

比如杀是出牌阶段限一次 酒是每回合限一次

现在chooseToRespond事件也可以像chooseToUse一样onChooseToRespond chooseButton useSkill了

具体请参考诸葛恪【傲才】等类似技能

lib.filter.targetEnabledx。

用于在考虑次数限制的情况下判断目标合法性。

loseMaxHp事件中的loseHp

为数值 用于判断减体力上限事件中减少的溢出体力值

chooseTarget事件和chooseCardTarget事件

可以将animate属性设置为false 用于避免先辅目标因动画效果暴露等情况

$throwEmotion

用于投掷表情

例：player.$throwEmotion(target,'egg');

请文明游戏 避免滥用

lib.characterReplace。

用于进行特定模式下的同名武将切换。

例：lib.characterReplace.xuzhu=['re\_xuzhu','xuzhu'];

就可以对标界许褚进行同框切换了

lib.card[i].yingbian\_prompt。

可以为函数（参数为卡牌）或者字符串。

用于展示卡牌长按时的应变描述。

lib.filter.cardSavable(card,player,target)。

用于判断一名角色能否使用某张卡牌对另一名濒死角色提供帮助。

save的AI标签的skillTagFilter

现在参数3为寻求帮助的濒死角色。

双势力武将

...

gz\_tangzi:['male','wei',4,['gzxingzhao'],['doublegroup:wei:wu']],

...

可以设置为多势力（如冈崎汐）

cardChongzhuable(card,player)

checkMod的一种 用于强制判断卡牌能否重铸

gaintag

用于给卡牌添加标签（多个标签可叠加）失去后清除

添加方法1

孟达【量反】

...

var cards=player.getStorage('qiuan');player.gain(cards,'gain2','fromStorage').gaintag.add('liangfan');

...

在gain事件中添加

draw事件也可以

添加方法2

例：吕凯【闭境】

...

if(result.bool){

player.logSkill('xinfu\_bijing');

player.addGaintag(result.cards,'xinfu\_bijing');

}

...

移除方法1

王元姬【宴戏】

...

player.removeGaintag('yanxi');

...

全部移除 一张不留

移除方法2

card.removeGaintag('xxx');

单独移除 精准爆破

multitarget:true,//防止多目标的主动技能运行两遍content里面的内容

selectCard:-1, 是选择所有手牌

//觉醒技标识

derivation:['bjzm更储'],

不计入次数

if (player.stat[player.stat.length - 1].card.jiu > 0) {

player.stat[player.stat.length - 1].card.jiu--;

}

if (player.stat[player.stat.length - 1].card.sha > 0) {

player.stat[player.stat.length - 1].card.sha--;

}

1. 添加/移除技能必须使用addSkill/removeSkill或addAdditionalSkill/removeAdditionalSkill函数

2. 添加/移除全局技能必须使用addGlobalSkill/removeGlobalSkill函数

3. 禁用/启用技能必须使用disableSkill/enableSkill函数

4. 检查是否有某技能必须使用hasSkill函数

5. 装备不再能直能通过appendChild函数添加，也不再能通过remove函数移除

6. 使用createEvent创建事件，事件的content必须用setContent添加

移除牌堆与弃牌堆中所有【南蛮入侵】与【万箭齐发】

var cards=[];

for(i=0;i<ui.discardPile.childNodes.length;i++){

var currentcard=ui.discardPile.childNodes[i];

if(currentcard.name=='naman'||currentcard.name=='wanjian'){

ui.discardPile.removeChild(currentcard);

cards.push(currentcard);

}

}

for(i=0;i<ui.cardPile.childNodes.length;i++){

var currentcard=ui.cardPile.childNodes[i];

if(currentcard.name=='naman'||currentcard.name=='wanjian'){

ui.cardPile.removeChild(currentcard);

cards.push(currentcard);

}

}

game.cardsGotoSpecial(cards);

ai:{combo:'jutu'}, //仅当角色有jutu这个技能时，此技能有意义

全局的储存

player.storage.bjzm主媾

只持续单个回合的储存

player.getHistory('custom').push({bjzm主媾:true,bjzm伏计card:trigger.card});

event.getParent('phaseUse').skipped = true; //结束出牌阶段