异兽装备(现金版功能)

将魔幻版功能移植到现金版

因为异兽装备是在现有的装备系统上添加功能,所以模块名和这个模块系统对应的功能名一样:

1. 宠物 模块名: lib\_player\_pet
2. 翅膀 模块名: lib\_player\_wing
3. 神兵 模块名: lib\_player\_godweapon
4. 护盾 模块名: lib\_player\_shield
5. 战灵 模块名: lib\_player\_fight\_spirit
6. 兵魂 模块名: lib\_player\_weapon\_soul
7. 特戒 模块名: lib\_player\_ring

一共移植七个系统的左右装备

给每个系统分别添加接口,

special\_up\_lv(功能id, 位置Pos)   
get\_special\_attr([(异兽装备列表)]) [{位置, 等级}]  
get\_skill\_attr()获取技能属性  
get\_special\_suit\_attr([(异兽装备列表)]) [{位置, 等级}]

功能主要逻辑:前端通过协议16873调用对应模块的special\_up\_lv()函数,如果成功升级一般需要重新计算属性,所以要在属性计算里加上get\_special\_attr()计算异兽装备等级添加的属性,get\_special\_suit\_attr()计算异兽装备套装添加的属性, get\_skill\_attr()获取异兽装备技能添加的属性,最后将这些属性与之前的属性列表合并起来通过attribute\_tool:sum\_partion\_attr(属性列表)添加所有属性.

因为进游戏需要让前端知道现在的异兽装备信息所以要在enter\_world()加入新的信息用信息号向前端发送当前灵兽装备信息.

special\_up\_lv(功能id,位置)函数主要逻辑:获取玩家ets的左右装备数据,根据位置找对应的装备等级如果找不到对应返回1,如果找到则返回对应等级+1,在配置表中根据功能id和位置获取消耗列表和技能列表,判断消耗物品是否足够扣除,如果足够则扣除并设置新的左右装备等级,通过获取技能配置调用lib\_player\_skill:add\_skill(SkillCfgL, 0)添加技能,计算最新属性,最后通过协议号向前端发送[0,升级后等级,功能id,位置],如果有错误发送[对应错误码,0,功能id,位置]

get\_special\_attr([异兽装备列表], 保存值(尾递归用)),通过尾递归实现遍历异兽装备列表[{位置, 等级}]然后通过功能id和位置和等级在配置表找对应的属性加成,然后将属性合并成一个新列表.

get\_special\_attr([异兽装备列表]),先通过get\_special\_min\_lv()获取这套装备里最低的装备等级,然后用尾递归函数do\_get\_special\_suit\_attr(MinLv(套装最低装备等级), 尾递归保存列表)遍历从套装最低装备等级到0这些等级在套装配置表获取对应的属性值,然后用尾递归的方式叠加起来.

(通俗的说法是:获取当前系统的最低异兽装备等级,假如有两件装备一件5,一件4则最低等级为4,这时用尾递归方式从配置表中遍历从4到0的套装等级对应的属性值然后合并成一个号新的列表)

get\_skill\_attr()获取异兽装备的技能属性加成(PS:这里的技能id是写死的),先获取玩家的技能列表,如果有对应的英雄id则添加到属性列表,这里有两种加属性的方式,一种是装备的阶数,每进阶一级加对应的属性值和另外一只,每击杀多少个boss加相应的属性值.

因为击杀boss数量需要计算所以先在场景管理进程svr\_map文件处处找到击杀boss处理函数do\_kill\_monster()里添加场景工具函数lib\_map\_util:handle\_kill\_boss1()这个函数主要处理计算boss数量的计算,并在lib\_player\_map():enter\_world用协议号12006向前端发送玩家击杀数量列表,以供前端显示玩家击杀数量.

每次击杀boss都会算出归属者信息和怪物信息然后跑到lib\_map\_util:handle\_kill\_boss1()函数,每次先检查玩家是不是在当前场景,再判断是不是跨服场景如果是跨服场景则需要远程调用函数lib\_player\_equip:update\_cross\_kill\_mon\_num(),

如果不是跨服则直接调用玩家进程函数lib\_player\_equip: update\_kill\_mon\_num ()即可.

lib\_player\_equip:update\_cross\_kill\_mon\_num()直接调用玩家进程函数lib\_player\_equip: update\_kill\_mon\_num ().

lib\_player\_equip: update\_kill\_mon\_num ()函数的主要逻辑,先获取玩家数据ets得到之前累计的击杀boss列表,检查需要计算击杀boss的技能当拥有其中一个则开始计算击杀boss类型数量,获取怪物类型当怪物类型为指定的怪物类型则开始计算,获取激活的需要计算该怪物类型击杀数量的技能列表,对该技能列表进行遍历,寻找玩家击杀对应怪物数量如果没找到返回0,找到则返回对应值,配置表获取当前技能需要击杀的最大值,获取他与玩家击杀对应数量较大值,如果配置表的较大则存入新的击杀数量到玩家数据中,如果玩家击杀数量已超过配置表配置最大数量则不作处理.

击杀boss需要在存玩家信息处添加新字段kill\_mon\_num格式为[{技能id, 击杀数量}],并通过协议号12017向前端发送最新击杀怪物数量信息.

(PS:添加字段需要在db\_upgrate文件处添加新的版本号和新的字段,上述的每个功能左右装备列表也需要加入到db\_update文件)

技能移植:

添加属性类的技能(1、根据装备阶数提升属性,2、根据击杀怪物数量添加对应的属性值)的技能写在对应的功能模块的获取属性下,对应配置技能id.(这种技能的实现已经在上面表达清楚)

被动技能移植:如攻击、被攻击、暴击、闪避等触发或概率触发的技能写入对应的功能技能模块。

技能id 30002 嗜血追击 攻击时概率对周围目标造成伤害。

这个被动技能需要写check\_30002（）函数判断有没有触发，现在玩家的被动技能列表里面寻找有没有这个被动技能id，然后找出玩家对应技能的技能等级在配置表中获取这个技能造成的伤害和触发几率，用随机数判断是否触发，如果触发了检测技能cd，如果在技能不在cd内则设置技能cd并获取该技能的范围和是否飘字，因为这个技能是对范围内的3个目标进行攻击,所以要通过find\_range\_target()寻找目标扣血,最后返回{伤害值,[{玩家id,技能id}]},与其他伤害和展示列表形成一个新列表.

find\_rang\_targe(攻击者战斗记录,半径,目标id,伤害值)逻辑:

如果附近对象是玩家:

首先通过进程函数lib\_9\_grid:get\_sight\_players(ActorPos)获取玩家坐标视野内可以看到的人,如果看不到人则返回空列表,对看到的玩家id进行遍历,如果id是攻击者或被攻击者则跳过,对其他玩家id进行通过lib\_map\_util:is\_in\_ellipse()判断攻击者坐标和其他玩家的id是不是在半径范围内,不是则跳过,是则用lib\_map\_util:get\_pixel\_distance()获取玩家和其他玩家的直线距离值,判断攻击者的攻击模式,如果是全体则可以对该其他玩家进行攻击,如果是帮会,如果攻击者帮会和其他玩家不是一个帮会这可以攻击,如果可以攻击返回[{玩家id, 1, 距离,当前血量值}]列表

如果附近对象是怪物:

通过进程函数lib\_9\_grid:get\_sight\_players(ActorPos)获取玩家坐标视野内可以看到的怪物,对怪物id进行遍历,获取不是被攻击怪物id和收集类型怪物, 通过lib\_map\_util:is\_in\_ellipse()判断攻击者坐标和怪物的id是不是在半径范围内, 用lib\_map\_util:get\_pixel\_distance()获取玩家和怪物的直线距离值,返回[{怪物id, 2, 距离,当前血量值}]列表

将这两个列表合并起来,如果空则跳过,如果不是空对列表里的距离进行排序,从这个新列表里面获取对应数量的目标列表,对这个列表进行遍历.如果目标当前血量少于伤害值返回0如果不是则返回当前血量值和伤害值的差,通过lib\_battle\_send:send\_see\_hurt()向前端发送伤害信息,根据之前获得的类型对玩家或怪物扣除伤害值血量.

技能id 30004 余震 攻击时概率增加伤害。

这种被动技能需要写一个check\_30004(对应技能id)()函数判断触发是否成功,先在玩家的技能列表寻找该技能id,如果找到用该技能id和该技能等级去配置表中获取概率触发值和添加的属性id和添加属性的值,用util:rand(1,最大)函数获取当前随机数,如果获取的随机数小于等于概率触发值则用函数lib\_battle\_passive\_talent:check\_passive\_skill\_cd(使用者id,技能id,技能等级)检查该技能id,如果不在技能cd时间范围内,则让技能进入cd并从配置表中获取buff配置然后构造一个新的buff记录,最后判断当前地图是不是跨服地图,如果不是直接通过svr\_player:map\_cast\_player(收到buff的玩家的pid,{add\_buff, buff变量}),通过这个接口地图进程向指定玩家pid进程调用add\_buff这个函数,如果是跨服则用svr\_map:player\_cast\_map(当前地图pid,{cross\_add\_buff, 收到buff的玩家id, buff变量}),最后如果发送成功则返回[{使用技能的玩家id, 技能id}]如果各种情况没法送buff成功则返回空列表,这个列表用于前端展示这个被动技能.

技能id 30006 反刺追击 触发闪避时提升自身攻击力.

因为这个技能是在玩家触发闪避的时候概率触发的所以在lib\_battle\_effect:check\_passive\_skill\_hit\_false()处添加check\_30006()函数.

check\_30006()当每次触发闪避的时候都会使用这个函数判断是否添加这个buff给玩家,首先在玩家被动技能列表寻找这个技能id获取技能等级,在配置表中获取这个技能添加的属性的键和值并将它们加入到buff记录的value值处以供buff\_on\_event遍历buff的pid为对应的pid值将这个键和值的属性加入到玩家当中,最后后通用的添加buff方法(如check\_30004()所示)添加到指定玩家中,并返回[{收到buff的玩家id, 技能id}]列表用于与其他闪避的技能列表合并用于放入show\_list列表一般用于前端飘字等.

技能id 30008 致命打击 攻击时有几率让被攻击方所受伤害增加(易伤).和上面技能类似.

技能id 30010 刀刀暴击 攻击时有几率触发buff接下来的n秒的攻击必定暴击,技能的添加和上述类似,但是这个技能机制需要必定暴击,所以在判断暴击lib\_battle\_expression:is\_critical()处添加我的技能展示列表否则这个技能将不会飘字.

技能id 30012 持续追击 暴击时有概率增加攻击力.

和上述技能不同是需要在check\_30012()添加个检查暴击的参数,当这个参数是true的时候才会进入函数判断概率触发.

技能 id 30014 圣盾反击 收到伤害有几率反弹本次伤害的一部分给攻击者

这个技能因为效果是以本次伤害的值作为基础计算反弹值或者恢复生命值所以要写在lib\_battle\_effect文件下的do\_calc\_skill\_rebound\_or\_addhp()里,check\_30014()通用检查技能是否触发,不一样的是这个技能加的不是buff而是反弹伤害所以要攻击者类型进行判断,如果是怪物就用api\_map:update\_monster\_hp(self(), 攻击者id,{dec\_hp, 反弹伤害}, 受到攻击者战斗记录),如果是玩家则用api\_map:update\_player\_hp(self(), 攻击者id,{dec\_hp, 反弹伤害值}, 0, 受到攻击者战斗记录),最后如果触发成功返回[{受到伤害者id, 技能id}]作为前端显示的列表,如果触发不成功则返回空列表.

技能 id 30018 战灵追击 永久增加战灵现身buff存在时攻击附加伤害的概率同时增加附加伤害值

因为这个技能是在其他技能的基础上增加概率和伤害,所以要在战灵现身的buff下的添加判断,在svr\_player文件下的make\_buff()中加上判断技能在不在玩家被动技能里面,如果在返回{增加概率,增加伤害万分比}添加到buff数值和概率判断中,如果没有这个技能则返回{0,0}.

技能 id 30020 受击时有概率恢复最大生命值

因为这个技能是有关血量变化的所以在lib\_battle\_effect文件下的check\_passive\_skill\_4()中添加check\_30020(),通用检测技能触发,如果触发返回{[技能发出者id, 技能id], [{加血者id, {recover, 加血量}, 血量恢复类型(在def\_player可以找到对应类型), [加血者id]}]}前者用于飘字显示,后者用于血量回复列表统一计算,如果没触发返回{[],[]}.

技能 id 30022 攻击时有概率提升攻速

通用加buff方式,唯一不同是配置表需要将fid改为403用于通用攻速计算,通用返回飘字列表.

技能 id 30024 受击时减少目标300点攻速

与技能30022一样不同的是需要将buff fid改为3004,.

技能 id 30026 受击有概率添加防御可以叠加n层.

配置表配置叠加类型以及叠加层数因为是百分比加防御故要改fid为402

技能 id 30028 生命低于30%时，受到攻击有30%几率产生护盾，可抵消最大生命10%的伤害，冷却30秒

Lib\_battle\_player\_shield文件下的launch\_on\_player中添加技能id判断,需要分开写判断技能触发判断和技能触发效果.

技能触发判断写在同文件下的get\_trigger\_rate里检测生命值是否低于30%和技能触发概率和技能cd.

技能触发效果写在launch\_on\_player()下, 因为和之前的技能不一样,没有返回值所以需要通过lib\_battle\_send:send\_show\_skill()发送飘字信息,设置技能进入cd,获取玩家最大护盾值计算添加护盾值,判断添加护盾值与最大护盾值较少值,用player\_data:set\_player\_map()设置玩家上述判断得到护盾值,然后用lib\_player:update\_to\_map()更新玩家护盾值,因为战斗计算是从地图进程里读取地图玩家战斗值#r\_map\_player{}所以在地图进程添加的护盾值可以被应用到战斗计算中.

Ps 飘字列表[{用技能的id, 技能id}]

属性移植:

(PS:属性更新不用添加字段,因为属性计算在玩家进入游戏的时候会计算一次不用保存到数据库)

新增

|  |  |  |
| --- | --- | --- |
| 属性类型 | 属性说明 | 备注 |
| 中毒几率： | 增加对目标造成伤害时的中毒几率 | 默认中毒时每秒造成最大生命3%的伤害，持续3秒，最终中毒伤害为每秒造成最大生命的3%\*（1+中毒伤害加成-中毒伤害减免）  中毒对怪物无效 |
| 中毒抗性 | 增加自身受到伤害时抵抗中毒的几率 |  |
| 中毒伤害加成 | 增加对目标造成中毒伤害时的伤害倍率 |  |
| 中毒伤害减免 | 减少受目标造成中毒伤害时的伤害倍率 |  |
| 卓越几率 | 增加对目标造成伤害时的致命几率 | 卓越=魔幻的致命 |
| 卓越抗性 | 增加自身受到伤害时抵抗致命的几率 |  |
| 卓越伤害加成 | 增加对目标造成致命伤害时的伤害倍率 |  |
| 卓越伤害减免 | 减少受目标造成致命伤害时的伤害倍率 |  |
| 双倍几率 | 增加对目标造成伤害时的双倍几率 |  |
| 双倍抗性 | 增加自身受到伤害时抵抗双倍的几率 |  |
| 双倍伤害加成 | 增加对目标造成双倍伤害时的伤害倍率 |  |
| 双倍伤害减免 | 减少受目标造成双倍伤害时的伤害倍率 |  |
| 混沌几率 | 增加对目标造成伤害时的混沌几率 | 混沌=魔幻的神圣 |
| 混沌抗性 | 增加自身受到伤害时抵抗混沌的几率 |  |
| 混沌伤害加成 | 增加对目标造成混沌伤害时的伤害倍率 |  |
| 混沌伤害减免 | 减少受目标造成混沌伤害时的伤害倍率 |  |

首先在rec\_attribute定义文件处添加各个属性字段.

然后在属性计算工具attribute\_tool文件处添加映射定义(ATTRIB\_MAP).

在属性系数定义(POWER\_COEF(Job))处添加定义, ,目的是自动在配置表对应属性和职业类型获取属性评分系数.

在计算总战力函数calc\_sum\_fight()处的战斗属性记录添加新的字段

在合并属性函数sum\_battle\_attr()处的三个战斗属性记录分别添加新的字段.

在计算属性差值函数diff\_attribute()处的两个战斗属性记录分别添加新字段字段,在属性差值列表添加新的子元组

在game\_admin\_player文件的请求玩家信息接口的字段存档里添加属性字段转换成json返回给前端.

在player\_login文件中的推送玩家基础属性函数push\_base()处Info2添加新字段,用于通过协议号12003向前端发送人物数据(战斗属性).

在战斗表现文件lib\_battle\_expression处,添加计算概率的函数.

暴击伤害需要增加pvp战斗属性计算,所以在lib\_battle\_expression文件下的获取暴击伤害倍率get\_critical\_rate()添加pvp的计算倍率,需要判断攻击目标对象是怪物或者玩家返回pve计算值或pvp计算值,pve计算值=基础暴击倍率和被动技能lib\_battle\_passive\_magic:critical\_add()的添加以及buff暴伤加成,加上技能爆伤加成.

PVP的爆伤加成用公式计算PVP=1.5+max(min(暴击伤害倍数PVE-1.5,0.5),0)+0.5\*max（min（暴击伤害倍数PVE-2，0.5），0）+0.3\*max（暴击伤害倍数PVE-2.5，0）,判断目标的类型如果是怪物则返回PVE值如果是玩家则返回PVP值,最后如果对方抗暴击加成太大会造成玩家暴击伤害比普通攻击伤害还低,所以需要对该伤害进行判断,返回较大值,防止暴击伤害低于普通攻击伤害.

卓越伤害、双倍伤害、创世伤害移植

首先需要在战斗效果文件lib\_battle\_effect下的damage\_normal()函数下添加三个函数分别计算这三个伤害is\_double(),IsExcellent(),IsCreation()计算这三个的触发,

需要分别传入攻击者和被攻击者的战斗记录,用攻击者的双倍攻击属性加上加双倍攻击属性的buff所加双倍攻击属性值,然后计算被攻击者的抵抗双倍攻击属性值和buff所加的抵抗双倍攻击属性值,用攻击者的双倍攻击属性值之和减去被攻击的抵抗双倍攻击属性值之和,然这个差值和0比较返回一个较大值防止获得负值,这个值为PVE值,如果是攻击者和被攻击者其中一个不是玩家返回PVE值,如果攻击者和被攻击都是玩家则返回pvp概率计算max(min(PVE概率, 2000), 0) + 0.8\*max(min(DoublePVE-2000, 1000), 0) + 0.5\*max(DoublePVE-3000, 0)

,然后给一个1到10000的随机数如果随机数小于等于获得的新值则触发双倍攻击否则不触发双倍攻击.

其余同理

最后他们返回true或者false以判断是不是触发这种攻击.

伤害计算:

将上面算出的true或false 传入lib\_battle\_expression:get\_final\_rate中分别算他们的实际倍率与上面计算暴击伤害倍率的方式一致.

只是计算公式有所不同

双倍: PVP=2+max(min(双倍伤害倍数PVE-2,0.5),0)+0.5\*max（min(双倍伤害倍数PVE-2.5，0.5），0）+0.3\*max（双倍伤害倍数PVE-3，0）

混沌: PVP=2.5+max(min(混沌伤害倍数PVE-2.5,0.5),0)+0.5\*max（min(混沌伤害倍数PVE-3，0.5），0）+0.3\*max（混沌伤害倍数PVE-3.5，0）

卓越: PVP=1.8+max(min(卓越伤害倍数PVE-1.8,0.5),0)+0.5\*max（min(卓越伤害倍数PVE-2.3，0.5），0）+0.3\*max（卓越伤害倍数PVE-2.8，0）

最后返回{他们之中最大的伤害倍率, 战斗飘字类型(是对应的伤害类型或者都没触发为普通攻击),真实战斗计算伤害类型(一般和飘字类型是一样的)}