境界任务

刻画系统 模块名lib\_player\_realm\_task

**原理 通过player\_module文件实现模块初始化,加载、卸载模块,保存模块和进入游戏时完成所需事务**

**因为旧的境界任务功能没有子境界字段,故要在db\_upgrade文件中添加新版本用来更新旧玩家的字段.**

**在gm指令中添加”realm 主境界 子境界”用于快四提升玩家境界,用指令”realmfinish”快速完成当前境界所有任务.**

**前端通过协议号15007发送境界任务领取任务奖励(境界,子境界,任务id)和15008发送升级境界.**

**玩家进入游戏根据功能开启模块来初始化.**

**根据玩家主境界初始化玩家刻画任务列表,如果玩家主境界为0则将玩家主境界和子境界初始化为初始值,这个初始值通过宏定义?** **INIT\_REALM和?** **INIT\_SUB\_REALM来决定,通过init\_data(主境界,子境界)来将玩家境界任务数据保存到记录r\_player\_realm\_task{}中.**

**这个功能模块有一个事件列表监听,监听玩家任务完成情况.当玩家完成这些事件的时候添加对应任务完成次数或者从历史完成次数添加到当前任务次数.**

**check\_task\_info用来获取[主境界,副境界,当前境界是否全部完成,当前境界任务列表]主要用来协议15008发于前端.**

**get\_flag\_reward 领取当前境界小任务奖励,检查任务是否完成和未领取通过check\_get\_flag\_reward,获得奖励通过do\_get\_flag\_reward发送奖励.**

**up\_realm 升级境界,通过check\_up\_realm**

**判断能不能升级,通过do\_up\_realm实现升级境界.**

**check\_task\_info主要逻辑,获取玩家记录r\_player\_realm\_task**

**经过遍历任务状态返回任务完成状态,取其个数判断是否与总任务个数相同返回全部任务是否完成标记,以及将子任务记录从记录转成元组形式方便协议发送.**

**check\_get\_flag\_reward 主要逻辑,**

**①判断是否有玩家任务数据**

**②配置表是否有当前境界的配置**

**③当前境界是否与配置表对的上(可能多余,为了稳定加上)**

**④判断任务状态是否为1**

**⑤尝试将奖励添加到背包**

**如果上述都成功返回{ok,任务配置}否则通过throw({error,错误号})跳出**

**do\_get\_flag\_reward 主要逻辑 获取旧的玩家境界任务数据**

**将新任务状态存进境界任务数据,并将新的全部任务状态标记存进新的境界任务数据,马上将数据存入mongo表并把遍历奖励发于玩家最后发送15007协议,并发送新的玩家任务信息15006给前端.**

**check\_up\_realm 主要逻辑**

**①判断是否有玩家任务数据**

**②全部任务状态标记是否为1**

**③配置表是否有当前境界的配置**

**④当前玩家等级是否大于当前境界开启等级(可能多余,为了稳定加上)**

**⑤检查是否为最大境界**

**⑥判断是添加主境界还是子境界**

**如果上述都成功返回{ok,添加境界类型}否则通过throw({error,错误号})跳出**

**do\_up\_realm 主要逻辑**

1. **获取玩家升主境界奖励并通过邮件发送给玩家,通过check\_up\_realm返回的升级境界类型用init\_data()初始化玩家新境界,设置玩家数据为新境界,分发境界升级事件,更新玩家境界到地图和将新境界信息(玩家名字,境界名字)广播出去,13037通知前端更新场景对象信息,发送新玩家境界任务信息,计算玩家数据,最后发送15008[成功or失败]**

**任务计算逻辑:**

**宏定义一个全部需要监听的事件的变量,后面的任务需要监听的事件就在宏定义下面加监听事件.当这个功能开启的时候就遍历这个宏定义用player\_event:add\_listener()添加监听.**

**监听事件handle\_event(),当监听的事件dispatch后会调用这个接口,因为事件有多个所以要写一个映射函数ref\_task\_type(),**

**映射函数ref\_task\_type()的作用是把监听的事件id转换成新定义的境界任务事件id(写在文件def\_realm\_task),如果对境界任务事件id用函数do\_handle\_event进行遍历,作用是计算累计值或重新计算新等级.**

**do\_handle\_event()获取玩家的境界等级和子境界等级和任务列表,从配置表中根据玩家的境界和子境界等级获取当前的境界任务,用update\_flag\_list()更新玩家任务列表,用do\_update\_flag\_list()尾递归检测flagid与传入值一样的任务,在任务配置表中找对应任务的参数值用不同的check\_taks\_count()检测不同对应参数的任务,如果状态为1(已完成未领取),2(已完成已领取)直接返回配置表数量,如果不是上述状态则根据配置的参数数量调用不同的do\_check\_task\_count()检测对应任务id的任务完成数量,如果是累计值在原有的基础上加上这次的值,如果是需要计算的则重新计算并返回.**

**新加任务**

**-define(REALM\_SOUL\_BOSS, 31). % 魂卡boss累计挑战N次**

**-define(REALM\_SOUL\_CARD\_QUALITY,32).% 小境界N魂卡X张达到品质Y**

**-define(REALM\_SOUL\_CARD\_LEVEL,33). % 小境界N魂卡强化总等级X级**

**-define(REALM\_SOUL\_CARD\_TREASURE,34).%魂卡寻宝累计寻N次**

**:** **-define(REALM\_ELEMENT\_BOSS,35). %元素boss击杀N次，按归属次数算**

**-define(REALM\_AWAKE\_DUP,36). %觉醒塔挑战N次**

**-define(REALM\_CALL\_LUOSHA\_BOSS,37). %罗刹boss召唤N次**

**-define(REALM\_CROSS\_TRINITY,38). %圣魔禁地参与N次，进入活动场景就算**

**-define(REALM\_STRANGE\_BOX,39). %异世宝箱开启N次，扣钥匙就算**

**-define(REALM\_CROSS\_ZODIAC,40). %生肖战场参与N次，进入活动参加就算**

**%%-define(REALM\_SOUL\_CARD\_TREASURE,41).%创世boss击杀N个，采水晶也算**

**-define(REALM\_CROSS\_DOMINATE,42). %主宰boss击杀N个，用归属次数做判断，采水晶也算**

**-define(REALM\_LUOSHA\_FIGHT,43). %罗刹系统总战力达到N**

**-define(REALM\_EQUIP\_STRENGTHEN\_LV,44).%装备强化总等级N**

**-define(REALM\_GEM\_LV,45). %宝石总等级N级**

**-define(REALM\_GOD\_CASTING\_LV,46). %神铸总等级N级**

**-define(REALM\_SHIELD\_LEVEL,47). %护盾N级**