缓存听牌检测时可共用的胡牌组合信息【key=手牌映射值（3N+1）】

性能优化相关文件：

bin\game\common\core\Base\CacheManager.lua 【缓存管理类】

bin\game\common\core\Base\mahjongUtil\cacheUtils.lua 【缓存方法集】

bin\game\common\core\Base\mahjongUtil\CheckUtils.lua 【性能优化方法集】

适用情况：

调用了听牌检测且无特殊将、刻、顺组合，且未使用缓存胡牌组合【key=手牌映射值（3N+1）】缓存逻辑的无癞子麻将。

一、 使用方式

略。

二、 流程图

暂无

三、 流程解析

该优化方式，主要是针对听牌检测中，减少胡牌组合检测的调用次数。所以会分别在听牌检测和胡牌检测的逻辑中插入听牌缓存信息的获取和处理逻辑。

以下简要说明缓存信息获取及处理的简要步骤：

1. 获取及保存听牌缓存信息（对应CacheManager中的getTingGroupInfoList函数）【调用位置：tingUtil中的isTing方法】

该逻辑主要分为五个步骤：

（1） 根据传入的玩家的手牌，获取对应的映射值（可调用CacheManager:getMappingValByHandCards获取）；

（2） 判断缓存数据中是否已存在该映射值的信息，若已存在则直接跳到（5）；

（3） 通过传入的groupMap参数（默认有将、刻、顺、四人帮），从手牌中获取所有对应的组合（调用CacheManager:getGroupInfoList获取）；

（4） 将所获得的缓存数据信息保存到指定的变量；

（5） 返回对应该手牌映射值的缓存数据。

2. 听牌缓存信息的处理逻辑（对应CacheManager中的getResultGroupsByCacheTingGroupInfoList函数）【调用位置：huUtil的getCardsGroups方法】

该逻辑主要分为六个步骤：

（1） 判断params.exParams.tingCheckCacheInfo参数是否存在，若不存在则直接返回nil值，该函数外面会调用默认的胡牌组合检测逻辑；

（2） 通过checkGroupMap函数，判断当前基础牌型所要检测的的组合类型配置，是否包含了不在传入参数中的baseGroupMap（对应上文出现的groupMap）的组合类型，若不存在则直接返回nil值，该函数外面会调用默认的胡牌组合检测逻辑；

（3） 判断缓存数据中，是否已存在对应手牌映射值和基础牌型ID的信息，若已存在则直接跳到（5）；

（4） 从手牌中移除要胡的牌，然后根据手牌、组合类型配置和听牌缓存信息，找出缺一张牌就能胡的胡牌组合信息，然后将要胡的牌加入到手牌中【有点类似缓存胡牌组合【key=手牌映射值（3N+1）】中的第三大点的第1小点】；

（5） 根据胡牌组合信息，调用CacheManager:getTotalGroupsByCacheHuGroupsInfos函数，获取所有胡牌组合【跟缓存胡牌组合【key=手牌映射值（3N+1）】中的第三大点的第2小点】；

（6） 返回所有胡牌组合。

四、 特点分析

1. 优点

（1） 在听牌检测中，会减少胡牌组合检测函数的调用次数；

（2） 具有可扩展性，对于特殊将刻顺组合类型，可以通过判断是否启用某规则ID，来扩展CacheManager中相应的组合校验函数；

（3） 由于目前是一局结束后，立即清除缓存，所以不会造成多局后运行内存增长过快的情况。

2. 缺点

（1） 跟缓存胡牌组合【key=手牌映射值（3N+1）】同时使用时，会造成部分缓存逻辑和数据上的重叠，可能导致胡牌检测的耗时增加。

（2） 当前优化只针对无癞子且普通的将、刻、顺组合类型的麻将，至于有特殊的顺子（如 风牌可以组合成顺子）的麻将，会出现组合检测不全的情况。若有该特殊顺子组合的麻将要使用该优化逻辑，需要对优化逻辑进行扩展。

3. 其他

（1） 对于不调听牌检测函数的麻将，并无性能上的提升（如 福建泉州麻将）。