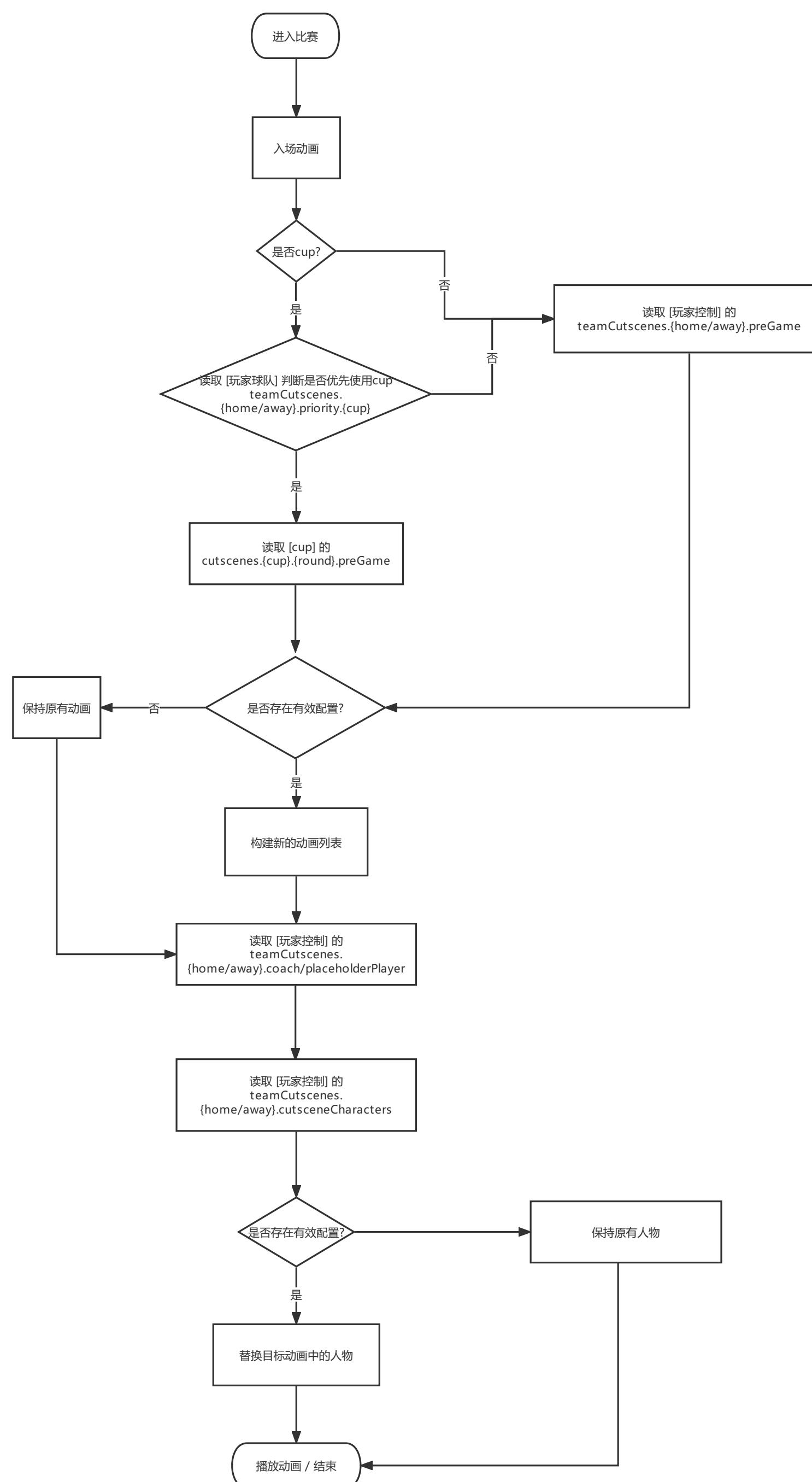


preGame / halftime / afterGame



> teamCutscenes.{teamid}.{?home/away}.*:

表示根据 [玩家球队] (afterGame为winner)的主场场读取对应的配置, 配置中没有对home/away分开配置的*, 则尝试读取teamCutscenes.{team id}.*
比如读取球队id为114的coach的配置, 当前为客场:
1. 先尝试读取teamCutscenes.114.away.coach,
2. 如果不存在, 则读取teamCutscenes.114.coach, (对于一些主场通用的配置, 可以如此配置).
3. 如果依然不存在, 则判定为未配置

> teamCutscenes.{teamid}.{?home/away}.priority.{matchType}:

如果priority.{matchType}设置为true, 表示当前情况下, teamCutscenes优先于cup.

matchType:
opening, groupStage, firstRound, secondRound, quarterRound, semiRound, final

比如球队114, 在groupStage和firstRound优先使用team配置:
teamCutscenes:
114:
priority:
groupStage: true
firstRound: true

> teamCutscenes.{teamid}.{?home/away}.coach:

coach只是一个alias, 作用等同于placeholderPlayer的第一个人, 当coach未配置时, 默认读取placeholderPlayer的第一个人作为coach

这个alias的作用是可读性提高, 并且可以只配置 teamCutscenes.{teamid}.coach 即可在所有比赛中为此球队开启ML专属coach动画显示.(无需其他配置).. teamCutscenes.{teamid}.coach/placeholderPlayer均未配置时, 当作无教练, 不适用ML动画

> teamCutscenes.{teamid}.{?home/away}.dynamicCutscenes:

对 [犯规(警告, 获得黄/红牌) / 入球] 的动态动画进行动态替换/插入

处理方式为, 针对原动画进行替换, 将原动画替换为你需要的其他动画. 选择替换+ wildcard, 可以更灵活, 你可以自己控制是在前面insert还是后面insert, 或者是直接删掉本来的动画替换成其他的.

配置支持[单个id, 多个id, id范围]三种配置方式,
范围越小(越精确), 优先级越高, 单个id > 多个id > id范围.

>>>1. 单个id:
dynamicCutscenes:
0x20001:
- [0x20002, +, 0x20005]
其中的+是一个wildcard, 在最终的动画列表中会被替换为匹配到的目标动画

>>>2. 多个id(中间用英文输入(半角)的逗号, 隔开):
dynamicCutscenes:
0x20001,0x20003,0x20004:
- [0x20006, 0x20007, +]
wildcard 放在最后, 效果等于在前面insert

>>>3. 范围id(中间用-隔开):
dynamicCutscenes:
0x20000-0x2FFF:
- [0x20006, +, 0x20007]
- [+ , 0x20004, 0x20005, 0x20010]
wildcard放在最前面, 效果等于在后面insert
当然都可以配置多组, 使用时会进行随机

> teamCutscenes.{teamid}.{?home/away}.dynamicCutscenesRecurrence:

是否针对配置中的动画再递归匹配, 不配置默认认为false

假设配置如下, 且当前original动画为[0x300F8]

dynamicCutscenes:

0x300F8:
- [+ , 0x300F6]
0x30000-0x3FFF:
- [+ , 0x30130]

那么,

1. true 递归, 最终结果为 [0x300F8, 0x300F6, 0x30130]

2. false(或未配置) 不递归, 最终结果为[0x300F8, 0x300F6]

这里0x300F6就是你配置的, 如果为true 则会针对0x300F6再次尝试匹配, 如果false则会跳过你从配置中insert的id

dynamicCutscenes

