



# 智能体决策1v1：游戏机制介绍

汇报人：薛源  
2025/10/27

# Contents 目录



1 初步介绍

3 代码思路

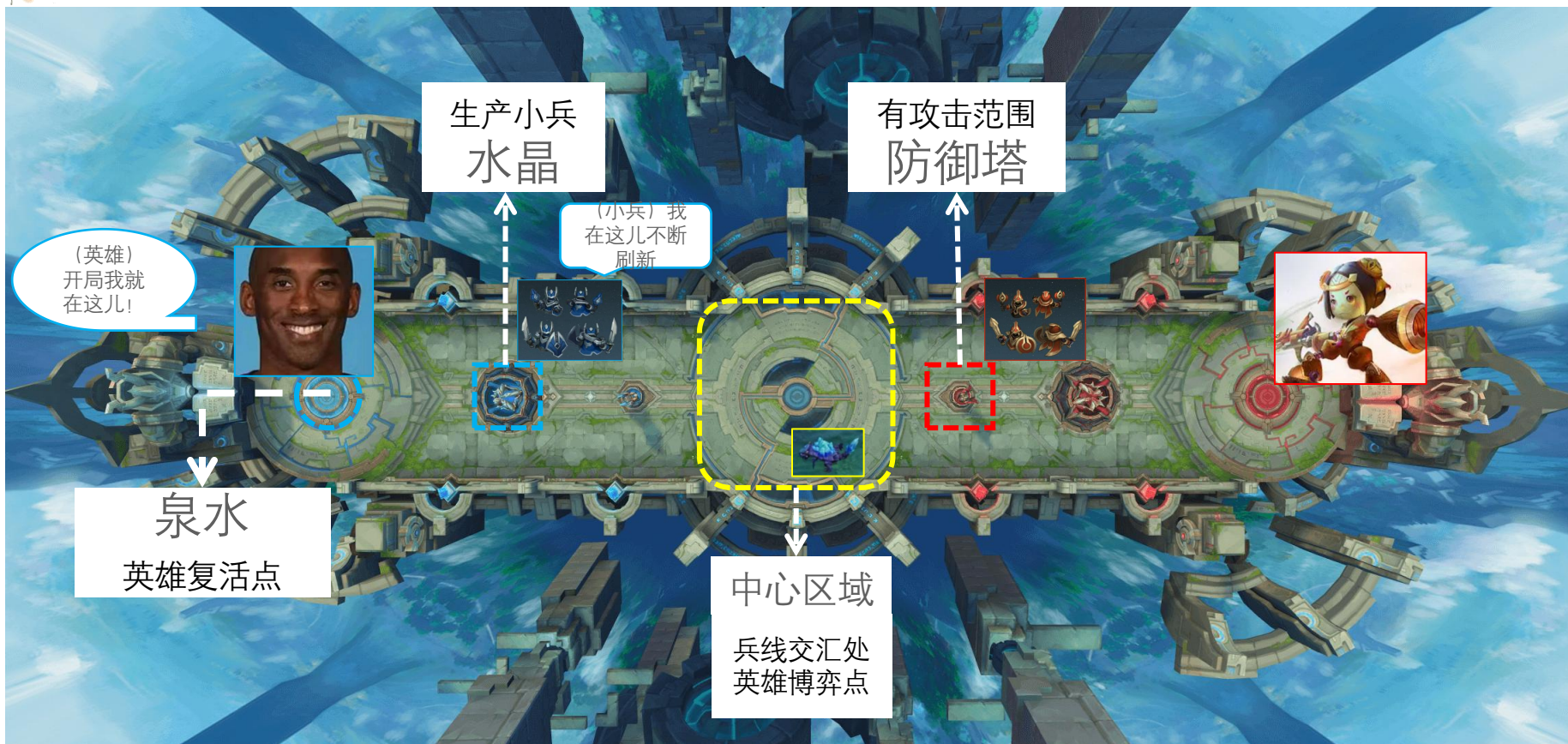
2 对战视频

4 总结建议





# 墨家机关道地图



目标：我方英雄率领小兵摧毁敌方第一座防御塔



英雄

火力压制 伤害

被动

鲁班七号第五次普攻或使用技能后的下次普攻会强化为扫射，对敌人造成3次普攻伤害  
对英雄伤害：70~140+4.5%物理伤害(每点额外物理攻击提升0.01%)  
对非英雄伤害：120~240+50%物理伤害

	LV1	LV15
英雄伤害	70	140
非英雄伤害	120	240

收藏

英雄试玩

全身

已穿戴

点击面板任意处关闭

5315221619+

鲁班七号

机关造物

精英射手 505/1000

来一局

游戏攻略

游戏职业 射手

细分定位 连珠射手

游戏特长 远程消耗

发育路

故事

强势时期 后期

生存能力

输出强度

上手难度

伤害

伤害

伤害

伤害

短视频

备战

英雄

河豚手雷 伤害 减速 视野

7.5秒 40

鲁班七号投掷河豚手雷造成520+120%物理伤害和25%减速(2秒)并获得敌人视野

	LV1	LV2	LV3	LV4	LV5	LV6
基础伤害	520	624	728	832	936	1040
冷却时间	7.5	7.2	6.9	6.6	6.3	6
降低移速	25%	30%	35%	40%	45%	50%

收藏

英雄试玩

全身

已穿戴

点击面板任意处关闭

5315221619+

鲁班七号

机关造物

精英射手 505/1000

来一局

游戏攻略

游戏职业 射手

细分定位 连珠射手

游戏特长 远程消耗

发育路

故事

强势时期 后期

生存能力

输出强度

上手难度

伤害

伤害

伤害

伤害

短视频

备战





英雄

无敌鲨嘴炮

伤害

控制

伤害

回

极简

12秒 60

鲁班七号发射火箭炮击退身前英雄并且造成450+100%物理伤害与5%法术伤害

	LV1	LV2	LV3	LV4	LV5	LV6
基础伤害	450	540	630	720	810	900
冷却时间	12	11.4	10.8	10.2	9.6	9
已损生命	5%	6%	7%	8%	9%	10%

收藏

英雄试玩

全身

已穿戴

点击面板任意处关闭

5315 2216 19+

鲁班七号

机关造物

精英射手 505/1000

来一局

攻略

游戏职业 射手

细分定位 连珠射手

发育路

故事

游戏特长 远程消耗

强势时期 后期

生存能力

输出强度

上手难度

伤害 伤害 伤害 伤害

短视频

备战

英雄

空中支援

伤害

减速

视野

回

极简

35秒 80

鲁班七号召唤河豚飞艇(持续14秒)，每秒随机对一个敌人投掷炸弹造成520+120%物理伤害并造成25%减速(1秒)  
飞艇将照亮支援范围内视野

	LV1	LV2	LV3
基础伤害	520	780	1040
冷却时间	35	27.5	20
减速效果	25%	37.5%	50%

收藏

英雄试玩

全身

已穿戴

点击面板任意处关闭

5315 2216 19+

鲁班七号

机关造物

精英射手 505/1000

来一局

攻略

游戏职业 射手

细分定位 连珠射手

发育路

故事

游戏特长 远程消耗

强势时期 后期

生存能力

输出强度

上手难度

伤害 伤害 伤害 伤害

短视频

备战



```

▼ 'skill_state' = {'slot_states': [{...}, {...}, {...}, {...}, {...}, {...}, {...}]
  > special variables
  > function variables
▼ 'slot_states' = [{'configId': 11200, 'slot_type': 'SLOT_SKILL_0', 'level': 1, 'usable': True, 'cooldown': 0, 'cooldown_max': 891, 'usedTimes': 0, 'hitHeroTimes': 0, 'succUsedInFrame': 0, 'nextConfigID': 0, 'comboEffectT...
  > special variables
  > function variables
  > 0 = {'configId': 11200, 'slot_type': 'SLOT_SKILL_0', 'level': 1, 'usable': True, 'cooldown': 0, 'cooldown_max': 891, 'usedTimes': 0, 'hitHeroTimes': 0, 'succUsedInFrame': 0, 'nextConfigID': 0, 'comboEffectTime': 0}
  > 1 = {'configId': 11210, 'slot_type': 'SLOT_SKILL_1', 'level': 0, 'usable': False, 'cooldown': 0, 'cooldown_max': 7000, 'usedTimes': 0, 'hitHeroTimes': 0, 'succUsedInFrame': 0, 'nextConfigID': 0, 'comboEffectTime': 0}
  > 2 = {'configId': 11230, 'slot_type': 'SLOT_SKILL_2', 'level': 0, 'usable': False, 'cooldown': 0, 'cooldown_max': 15000, 'usedTimes': 0, 'hitHeroTimes': 0, 'succUsedInFrame': 0, 'nextConfigID': 0, 'comboEffectTime': 0}
  > 3 = {'configId': 11220, 'slot_type': 'SLOT_SKILL_3', 'level': 0, 'usable': False, 'cooldown': 0, 'cooldown_max': 30000, 'usedTimes': 0, 'hitHeroTimes': 0, 'succUsedInFrame': 0, 'nextConfigID': 0, 'comboEffectTime': 0}
  > 4 = {'configId': 90003, 'slot_type': 'SLOT_SKILL_4', 'level': 1, 'usable': True, 'cooldown': 0, 'cooldown_max': 60000, 'usedTimes': 0, 'hitHeroTimes': 0, 'succUsedInFrame': 0, 'nextConfigID': 0, 'comboEffectTime': 0}
  > 5 = {'configId': 80115, 'slot_type': 'SLOT_SKILL_5', 'level': 1, 'usable': True, 'cooldown': 0, 'cooldown_max': 120000, 'usedTimes': 0, 'hitHeroTimes': 0, 'succUsedInFrame': 0, 'nextConfigID': 0, 'comboEffectTime': 0}
  > 6 = {'configId': 90005, 'slot_type': 'SLOT_SKILL_6', 'level': 1, 'usable': True, 'cooldown': 0, 'cooldown_max': 0, 'usedTimes': 0, 'hitHeroTimes': 0, 'succUsedInFrame': 0, 'nextConfigID': 0, 'comboEffectTime': 0}
  len() = 7
  len() = 1

```

技能（cooldown\_max表示冷却，单位是ms）：

11200 – 被动

11210 – 一技能

11230 – 二技能

11220 – 三技能

90003 – 恢复键

80115 – 闪现键

90005 – 回城键

重点关注：usedTimes、hitHeroTimes



## 时间信息

帧(frame)和步(step)存在一定映射关系。

帧是场景的一个时间单位，表示场景的一个完整更新周期。在每一帧中，场景的所有元素(如英雄状态等)都会根据当前的状态和输入进行更新。

步是强化学习环境中的一个时间单位，表示智能体(agent)在环境中执行一个动作并接收反馈的过程。在每一步中，智能体选择一个动作，环境根据该动作更新状态，并返回新的状态、奖励和终止信号。

在本环境中，1 个 step 由 6 个 frame 组成。这意味着每个动作对应一个步，在每一步中，智能体将在六个连续的帧中执行同一个动作。环境将在每一步结束后更新状态并返回反馈，场景只有在完成六帧后，环境状态才会返回一次状态的更新。

- **步更新**：在每一步中，智能体选择一个动作，环境更新状态并返回。
- **帧更新**：在一步中，场景进行六次帧更新，更新所有场景中对象的状态并渲染新的画面。

帧(frame)，步(step)，现实时间秒(s)和现实时间毫秒(ms)的关系如下：

1 frame 约等于 66 ms

1 step 执行 6 frame

1 s 等于 1000 ms

**注意：**由于运行环境的差异，每一帧的时间会在 66 毫秒上下浮动

意味着env.step返回的是时隔6帧的信息

每秒15帧（其实挺卡的），可实际上实验中似乎还是每秒30帧

```
Enemy Tower Target: 0
Behavior Mode: Direction_Move

=== Frame 5432 Agent 1 Status ===
Attack Target: 0
Enemy Tower Target: 0
Behavior Mode: State_Idle

=== Frame 5438 Agent 0 Status ===
Attack Target: 0
Enemy Tower Target: 0
Behavior Mode: State_Idle

=== Frame 5438 Agent 1 Status ===
Attack Target: 0
Enemy Tower Target: 0
Behavior Mode: UseSkill_1

=== Frame 5444 Agent 0 Status ===
Attack Target: 0
Enemy Tower Target: 0
Behavior Mode: UseSkill_1

=== Frame 5444 Agent 1 Status ===
Attack Target: 0
Enemy Tower Target: 0
```

针对某个技能：

If in this step usedTimes change:

```
if hitHeroTimes == last_hitHeroTimes+1:
    give reward（延后的奖励）
```

Ps：一局游戏结束后累计也是可以的

```
=== Frame 5456 Agent 1 Status ===
Attack Target: 0
Enemy Tower Target: 0
Behavior Mode: Direction_Move
```



此处应有个对战视频



# 分阶段权重

兵线最初是15秒出，以后每隔27秒出；  
每波兵线抵达战场约11秒；  
一分20刚好是前三波兵线交汇完

```
# 设置各个回报项的权重，在reward_manager中使用
REWARD_WEIGHT_DICT = {
    "hp_point": 2.0,
    "tower_hp_point": 10.0,
    "money": 0.01,
    "exp": 0.006,
    "ep_rate": 0.2,
    "death": 2.0,
    "kill": -0.6, # 先设置为负值，后续再调整
    "last_hit": 0.75,
    "forward": 0.01,

    "grass_penalty": 0.05, # 新增草丛惩罚权重，过
    "flash_skill": 1.0, # 闪现奖励权重
    "heal_skill": 0.5,
    "skill1": 0.8,
    "skill2": 0.8,
    "skill3": 1.0,
    "chase": 0.04, # 适度提高追击奖励
}
```

```
# 新增的动态权重调整配置（不要用乘积）
DYNAMIC_WEIGHT_CONFIG = {
    "early_game_threshold": 2700, # 前期结束的帧号（1分半=1秒30帧，1分钟60秒1800帧）
    "late_game_threshold": 5400, # 后期开始的帧号（3分）

    # 前期的权重
    "early_game_weights": {
        "hp_point": 2.0,
        "last_hit": 1.0,
        "flash_skill": 0.5, # 前期谨慎使用闪现
        "chase": 0.03, # 前期追击权重较低
    },

    # 中期的权重
    "mid_game_weights": {
        "money": 0.008,
        "hp_point": 1.0,
        "tower_hp_point": 15.0,
        "last_hit": 0.75,
        "forward": 0.012,
        "flash_skill": 1.0,
        "chase": 0.05, # 中期追击权重适中
    },

    # 后期的权重
    "late_game_weights": {
        "money": 0.006,
        "tower_hp_point": 12.0, # 推塔奖励提升
        "kill": 5.0, # 击杀奖励提升
        "forward": 0.015,
        "chase": 0.06, # 后期提高追击积极性
    },
}
```

- ◆ 权重具体的值该怎么设，建议设监控diy\_1~5，然后自己算。
- ◆ 确保奖励值大致符合 前期（1分20之前，即frame\_no=80\*30=2400）：发育类>人头类>推塔类；  
中期：发育类~人头类~推塔类；后期（4分钟以后）：发育类<人头类<推塔类  
在reward\_process.py中通过判断帧号来给予不同的权重。讨论一下kill的权重？



## 奖励设置

```
# 草丛惩罚，这个应该是状态型奖励，如果说在草里，就给予一个持续性的惩罚，这样
elif reward_name == "grass_penalty":
    main_hero_in_grass = main_hero.get("isInGrass", False)
    reward_struct.cur_frame_value = -1.0 if main_hero_in_grass else 0.0
elif reward_name == "chase":
    reward_struct.cur_frame_value = self.calculate_chase_reward(main_hero, enemy_hero)
elif reward_name in ["flash_skill", "heal_skill", "skill1", "skill2", "skill3"]: # 瞬时奖励
    reward_struct.cur_frame_value = skill_rewards[reward_name]
self.last_dist2enemy = dist2enemy
```

零和奖励：money、hp\_point、kill、death、tower\_hp\_point、exp

独立奖励：forward、ep\_rate

事件触发奖励（属于稀疏奖励，kill和death也属于）：last\_hit

以上是为了告诉大家在get\_reward函数中怎么写。

并非所有新设置的奖励都适合零和奖励。

比如一二三技能命中。

```
# 草丛惩罚是为了避免影响长时间待在草里，是一个单边的负奖励，不涉及双方相减。可恶，原本把reward_name写
elif reward_name == "grass_penalty":
    reward_struct.value = self.m_main_calc_frame_map[reward_name].cur_frame_value
elif reward_name == "flash_skill":
    reward_struct.value = self.m_main_calc_frame_map[reward_name].cur_frame_value
elif reward_name == "heal_skill":
    reward_struct.value = self.m_main_calc_frame_map[reward_name].cur_frame_value
elif reward_name in ["skill1", "skill2", "skill3"]:
    reward_struct.cur_frame_value = self.m_main_calc_frame_map[reward_name].cur_frame_value
    reward_struct.last_frame_value = self.m_main_calc_frame_map[reward_name].last_frame_value
    reward_struct.value = reward_struct.cur_frame_value - reward_struct.last_frame_value
elif reward_name == "hit_tower":
    reward_struct.value = self.m_main_calc_frame_map[reward_name].cur_frame_value
elif reward_name == "lazy":
    reward_struct.value = self.m_main_calc_frame_map[reward_name].cur_frame_value
else:
    # Calculate zero-sum reward
```



# 防御塔

```
> 'npc_states' = [{ 'config_id': 1114, 'runtime_id': 16, 'actor_type': 'ACTOR_ORGAN', 'sub_type': 'ACTOR_SUB_CRYSTAL', 'camp': 'PLAYERCAMP_2', 'behav_mode': 'Attack_Move', 'location': {...}, 'forward': {...}, 'hp': ... },  
> special variables  
> function variables  
> 0 = { 'config_id': 1114, 'runtime_id': 16, 'actor_type': 'ACTOR_ORGAN', 'sub_type': 'ACTOR_SUB_CRYSTAL', 'camp': 'PLAYERCAMP_2', 'behav_mode': 'Attack_Move', 'location': {'x': 31250, 'y': 0, 'z': 31350}, 'forward': ... },  
> 1 = { 'config_id': 1112, 'runtime_id': 15, 'actor_type': 'ACTOR_ORGAN', 'sub_type': 'ACTOR_SUB_TOWER', 'camp': 'PLAYERCAMP_2', 'behav_mode': 'Attack_Move', 'location': {'x': 13131, 'y': 0, 'z': 13258}, 'forward': ... },  
> 2 = { 'config_id': 1111, 'runtime_id': 14, 'actor_type': 'ACTOR_ORGAN', 'sub_type': 'ACTOR_SUB_TOWER', 'camp': 'PLAYERCAMP_1', 'behav_mode': 'Attack_Move', 'location': {'x': -13138, 'y': 0, 'z': -12940}, 'forward': ... },  
> 3 = { 'config_id': 1113, 'runtime_id': 8, 'actor_type': 'ACTOR_ORGAN', 'sub_type': 'ACTOR_SUB_CRYSTAL', 'camp': 'PLAYERCAMP_1', 'behav_mode': 'Attack_Move', 'location': {'x': -31310, 'y': 50, 'z': -31120}, 'forward': ... },  
> 4 = { 'config_id': 46, 'runtime_id': 7, 'actor_type': 'ACTOR_ORGAN', 'sub_type': 'ACTOR_SUB_TOWER_SPRING', 'camp': 'PLAYERCAMP_2', 'behav_mode': 'Attack_Move', 'location': {'x': 45785, 'y': 1120, 'z': 45856}, 'forward': ... },  
> 5 = { 'config_id': 44, 'runtime_id': 2, 'actor_type': 'ACTOR_ORGAN', 'sub_type': 'ACTOR_SUB_TOWER_SPRING', 'camp': 'PLAYERCAMP_1', 'behav_mode': 'Attack_Move', 'location': {'x': -45396, 'y': 1120, 'z': -45467}, 'forward': ... },  
> 6 = { 'config_id': 21, 'runtime_id': 1, 'actor_type': 'ACTOR_ORGAN', 'sub_type': 'ACTOR_SUB_NONE', 'camp': 'PLAYERCAMP_2', 'behav_mode': 'Attack_Move', 'location': {'x': -29000, 'y': -8100, 'z': -152100}, 'forward': ... },  
len() = 7
```

```
> 1 = { 'config_id': 1112, 'runtime_id': 15, 'actor_type': 'ACTOR_ORGAN', 'sub_type': 'ACTOR_SUB_TOWER', 'camp': 'PLAYERCAMP_2', 'behav_mode': 'Attack_Move', 'location': {'x': 13131, 'y': 0, 'z': 13258}, 'forward': {'x': 707, 'y': 0, 'z': -707}, 'hp': 12000, 'max_hp': 12000, 'values': {'phy_atk': 470, 'phy_def': 200, 'mgc_atk': 470, 'mgc_def': 200}, 'abilities': [False, True, False, False, False, False], 'attack_range': 8800, 'attack_target': 0, 'kill_income': 50, 'camp_visible': [True, True], 'sight_area': 8800 }
```

◆ 一个闪现400码=4000

◆ 英雄视野=3个闪现=12000

◆ 防御塔攻击范围8800，射手射程8000

设置回城的话，一种走回去，一种通过回城键

走回去的话要参考和ACTOR\_SUB\_TOWER\_SPRING的距离





## 奖励设置

```
reward_struct.cur_frame_value = self.calculate_forward(main_hero, ma  
# 草丛惩罚, 这个应该是状态型奖励, 如果说在草里, 就给予一个持续性的惩罚, 这样  
elif reward_name == "grass_penalty":  
    main_hero_in_grass = main_hero.get("isInGrass", False)  
    reward_struct.cur_frame_value = -1.0 if main_hero_in_grass else 0.0  
elif reward_name == "chase":
```

躲在草里就给予惩罚

```
def calculate_chase_reward(self, main_hero, enemy_hero):  
    """计算战斗相关奖励, 一个情景奖励"""  
    chase_reward = 0.0  
  
    main_hp_ratio = main_hero["actor_state"]["hp"] / float(main_hero["actor_state"]["max_hp"])  
    enemy_hp_ratio = enemy_hero["actor_state"]["hp"] / float(enemy_hero["actor_state"]["max_hp"])  
  
    hero_pos = (main_hero["actor_state"]["location"]["x"],  
                main_hero["actor_state"]["location"]["z"])  
    enemy_pos = (enemy_hero["actor_state"]["location"]["x"],  
                 enemy_hero["actor_state"]["location"]["z"])  
    dist = math.dist(hero_pos, enemy_pos)  
  
    if main_hp_ratio > enemy_hp_ratio + 0.2 and enemy_hp_ratio < 0.5:  
        if dist < 6000:  
            chase_reward = 1.0  
        elif dist < 8000:  
            chase_reward = 0.6  
  
        if self.last_dist2enemy is not None: # 确保不是第一帧 (虽然可能性也不大)  
            if dist < self.last_dist2enemy: # 如果帧距离在缩小  
                chase_reward *= 1.2 # 给予额外奖励  
    return chase_reward
```

鼓励残血追杀



## 奖励设置

```
# Tower health points
# 塔血量
elif reward_name == "tower_hp_point":
    reward_struct.cur_frame_value = 1.0 * main_tower["hp"] / main_tower["max_hp"] + 1.0 - 1.0 * enemy_tower["hp"] / enemy_tower["max_hp"]
    # reward_struct.cur_frame_value = 1.0 * main_tower["hp"] / main_tower["max_hp"]
# Last hit
# 补刀
```

虽然是零和奖励，但是这样可以放大奖励，同时鼓励推塔和守塔

```
# 河道之灵
elif reward_name == "last_hit":
    reward_struct.cur_frame_value = 0.0
    frame_action = frame_data["frame_action"]
    if "dead_action" in frame_action:
        dead_actions = frame_action["dead_action"]
        for dead_action in dead_actions:
            if dead_action["death"]["sub_type"] == "ACTOR_SUB_SOLDIER": # 如果死的是小兵
                if dead_action["killer"]["runtime_id"] == main_hero["actor_state"]["runtime_id"]: # 运行id, 应该是英雄的唯一标识
                    reward_struct.cur_frame_value += 1.0 # 补刀奖励
                elif dead_action["killer"]["runtime_id"] == enemy_hero["actor_state"]["runtime_id"]:
                    reward_struct.cur_frame_value -= 1.0 # 敌方补刀奖励
            elif dead_action["death"]["config_id"] == 6827: # 河道之灵
                if dead_action["killer"]["runtime_id"] == main_hero["actor_state"]["runtime_id"]:
                    reward_struct.cur_frame_value += 4.0 # 击杀河道之灵奖励
                elif dead_action["killer"]["runtime_id"] == enemy_hero["actor_state"]["runtime_id"]:
                    reward_struct.cur_frame_value -= 4.0 # 敌方击杀河道之灵惩罚
# Experience points
```

河道之灵的奖励

```
# 计算向前奖励, 从残血到满血的距离, 计算向前奖励
def calculate_forward(self, main_hero, main_tower, enemy_tower):
    main_tower_pos = (main_tower["location"]["x"], main_tower["location"]["z"])
    enemy_tower_pos = (enemy_tower["location"]["x"], enemy_tower["location"]["z"])
    hero_pos = (
        main_hero["actor_state"]["location"]["x"],
        main_hero["actor_state"]["location"]["z"],
    )
    forward_value = 0
    dist_hero2emy = math.dist(hero_pos, enemy_tower_pos)
    dist_main2emy = math.dist(main_tower_pos, enemy_tower_pos)
    if main_hero["actor_state"]["hp"] / main_hero["actor_state"]["max_hp"] > 0.5 and dist_hero2emy > dist_main2emy:
        forward_value = (dist_main2emy - dist_hero2emy) / dist_main2emy
    return forward_value
```

Forward奖励：从0.99修改为0.5，不然英雄残血后会缩在塔下



谢谢观看

Thanks for watching

