程序设计报告：wikuAI

计52 赵梓硕

May 24, 2016

Chapter 1

介绍

此程序为我参加第20届智能体大赛使用的AI代码，实现操控游戏角色自动进行采矿、打野、团战、进攻、防守等操作，以求最终获得比敌方更多的经济或是攻破对方基地而取得胜利。我的AI以样例中的SimpleAI为基础，初步运用了OOP的思想，实现了对游戏局势进行大致评估并得出决策。

由于编程水平与时间精力有限，我只用了670行代码实现了一个相当粗糙的AI，且部分功能并未完全使用OOP的设计模式，甚至还有部分架构未完善。

在期中最繁忙的阶段参加这次持续5个星期的比赛，面对并克服了很多挑战，与大量编程高手相互较量并进入16强，无论是结果和过程都获得了很大的收获，并且明显提高了编程熟练度与水平。

Chapter 2

基本架构

程序使用了自带的console接口，并将整个AI封装为wiku_ai类，类中的成员变量存储了各种局势信息，不同的成员函数实现移动、攻击、施法、撤退、回防等不同功能。程序运行顺序如下：

- $wiku\_ai()$ 构造函数读取地图信息
- $work()$
  - $initialize()$ 初始化英雄信息
  - $base\_check()$ 判断基地是否需要防守
  - $get\_targets()$ 获取每个英雄攻击目标
  - 判断人数是否过少 若过少则暂不进攻
  - 判断是否处于防守状态
  - 遍历每个存活英雄 $think()$
    - 已分配（防守）任务？
      - 是 不分配新任务
      - 否 有攻击目标？
        - 是 施法>走位>进攻
        - 否 $occupy(who)$

$occupy(who)$：是否满足进攻条件？

- 是 $push(who)$ 集结进攻敌方基地
- 否 寻找对应野区打野

Chapter 3

特色功能实现

3.1 攻击优先级判断

“伤其九指，不如断其一指”，在大多数情况下，优先消灭一个敌方单位对战局有利，因此*priority*()函数实现了对敌方优先级的判断，每个英雄都会尝试攻击视野范围内*priority*()值最小的敌军。

由于最终目的是攻破基地，所以基地的优先级最高。（这可能值得商榷，但我的AI就这么处理）
“鞭尸”显然毫无意义，因此死亡敌军的优先级最低。（事实上死亡者只会出现在敌方基地附近）
在发生战斗时，相比于眼和野怪，一般应当优先消灭敌方英雄，因此眼和野怪的优先级次低。
在防守状态下，可以攻击我方基地的英雄应该优先消灭，因此优先级次高。
如果敌方单位不在己方英雄攻击范围内，为了避免被“勾引”，其优先级会有不同程度的降低。
一般情况下，敌方生命值越低，优先级越高。

基地攻击的优先级略有不同。由于Berserker开启大招可能对基地造成极大伤害，因此会优先攻击Berserker。事实上，生命值越高的Berserker威胁越大，因此对hp的判定应当反转，且有没有大招的Berserker威胁程度也不同，因此可以进行进一步优化，但我的程序中未进行此优化。

3.2 双矿打野

我的AI采取放弃中矿，通过野矿发展经济的策略。Roshan的战斗能力比Dragon强很多，因此在游戏前期应避免打Roshan，且后期相比Dragon，Roshan应聚集优势兵力去消灭。

我选择前期单矿中后期双矿，因此兵力调配很重要。*wildmine*()函数就根据当前英雄人数与游戏时长分配兵力。

分配原则如下：
如果双野均存活，则只有中后期（400回合以后）且人数极多（达到7人）才两个野怪一起打，并给Roshan分配较多的兵力。
如果已到200回合，至少有5人存活且Dragon死亡，那么可以尝试打Roshan。
如果有且只有一个野怪存活，那么无野怪的矿仅分配1人，其余兵力集火另一野怪。

3.3 进攻基地

进攻策略由变量*push_sign*控制，共有3个可能状态：

$$\begin{cases} 0 & \text{兵力不足，不进攻} \\ 1 & \text{尝试进攻} \\ 2 & \text{进攻不利，刷野至后期再进攻} \end{cases}$$

*push_sign*的初始值为0。当场上有8名己方英雄且其值为0时，*initialize*()函数将其改变为1，表示开始尝试进攻。

进攻集结状态由成员变量*group*控制。

在*push_sign*为1时，*occupy*()会调用*push*()函数控制进攻。进攻时首先集结到位置(45,120)，人员到齐后*group*变为1，集结至敌方基地附近(118,136)（以Player0为例）。当有足够数量的英雄到达敌方基地附近时，*group*将会变为2，开始无条件进攻敌方基地。当开始进攻基地或者减员严重时，*group*变回0，表示这一波进攻“完成”，避免新英雄复活后无序进攻。

如果有英雄被敌军截获或因其他原因无法到达集结点，且已经集结的英雄较多时，变量*push_count*会开始增加，英雄越多增加越快。当达到一定阈值时将不再等待，开始进攻。*push_count*也会在一波进攻完成时重置。然而与较弱对手对战时此功能几乎不起作用，而较强对手常使用某些技巧“隐藏实力”，导致*push_count*功能未能调试成熟，存在较多bug，不过它在很多情况下确实可以起作用。

在进攻敌方基地时，未被攻击的Berserker会在攻击基地前开启大招，试图造成较大伤害。当然这个策略依然可以被针对，但对不针对的玩家有非常强大的效果。

3.4 反针对策略

作为一个竞技性极强的游戏，尤其到了后期只有32名玩家对峙的状况下，玩家之间的针对现象极其明显。写出一个强的AI并不容易，但对于一个不弱的对手，写一个针对性的程序来“专攻”是相对简单的。尤其是允许在同一场比赛中使用不同AI，使针对的套路变得更深。本着“少一份套路，多一份真诚”和“友谊第一，比赛第二”的想法，我没有针对其他玩家，但对于其他玩家的针对策略进行了一定的反制。

3.4.1 push_wait的设置

在比赛过程中，有些玩家针对我的AI想出了一种极其naive的针对方式：基地血量较多时不怎么防守，而在基地残血时无脑防守，这样我的第一波进攻会较为顺利，而在之后的进攻中则会每次都失利而处于经济劣势，最终落败。因此我在较新的版本中采取“刷一波偷一波”的策略，即在一波进攻完成后刷一段时间的野怪弥补经济，若干回合后再进行进攻。（间隔在50至150回合之间，越到后期间隔越短，700回合后不间断。）这样在对方无脑防守时，我方也能积累较多的经济，升级兵种以有效消耗对方基地，争取胜利。

3.4.2 在敌方基地附近的特判

我的作战策略中有一条，即陷于人数上不利的境地，或是远程被敌方近战贴脸时会撤退，于是有人在守家时故意聚集基地周围较远（我的第二集结点之外）的地方，而我的*occupy*()只有在周围无敌方单位时才会触发，因此会导致我的队伍在敌方基地附近徘徊不前。因此我的*dodge*()做了一个特别判定：如果距离敌方基地较近，则会向着敌方基地（而不是我方基地）方向“逃跑”。

此外，*push_check*()函数还实现了一个判定，即有足够多的己方英雄在敌方基地附近时，无视敌方所有英雄，直接攻击基地。

这些想法或许非常粗糙，但确实能解决很多问题。

Chapter 4

待改进问题

由于时间有限，以及害怕不成熟的改动反而使AI变弱或出现预料之外的bug，我意识到了一些问题，但在比赛中未能改进。

（1）人数未齐但足以进行一次进攻时，$push_count$有时不能正确地触发进攻，且$push_count$和$group$有时不能正确地在一波进攻之后清零。事实上，如何界定“一波”进攻并不容易。

（2）每次总在同样的野区打野，由于采取双野策略，对针对性抓野的应对能力较差，且在局部战斗触发时缺乏有效的集合机制。事实上$assembly()$函数是一个雏形，但暂时只用于防守基地，而为将其用在战斗中的支援上。

（3）每个英雄仅对视野范围内的单位和全局的战略性指令（推家，防守，刷野）感兴趣，缺乏小规模支援和局部判断的能力。这是因为我多半使用定性方式判断局势，缺乏用定量方法估计局势和计算战斗力、价值的算法。

（4）缺乏对已决定的策略进行记录并进行一定程度的“坚持”的设定，导致少数时候当局势处于临界值时，可能会反复改变策略导致英雄出现“徘徊”与卡死的状态。