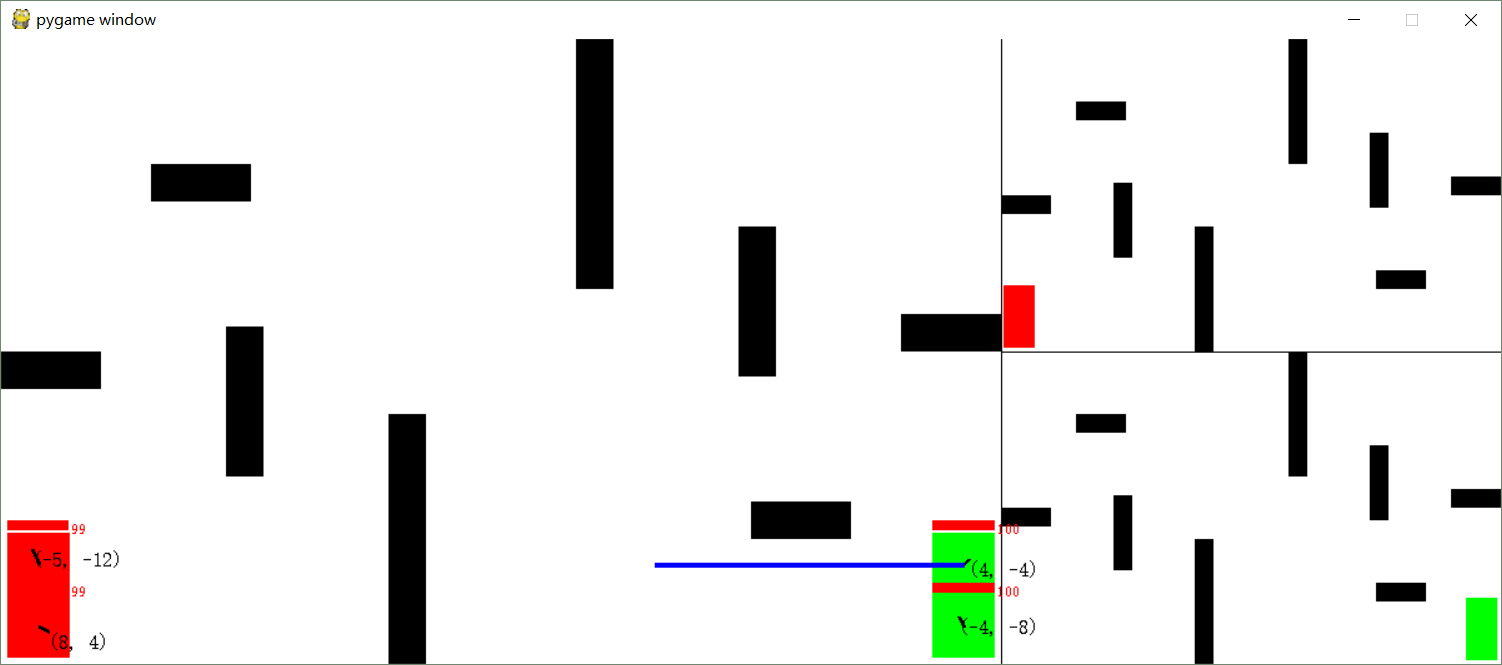
# strategy overview

经过可行性研究后，我们放弃了深度强化学习端到端的控制方式，选择了传统的基于状态机架构。共有两个状态机，每个状态机控制己方一个机器人。每个状态机有Start，巡逻Patrol，追击Pursuit，站桩输出DPS四种状态。第一步我们实现了基本功能，即单个机器人的状态机，包括路径规划，并在仿真环境中模拟进行修改迭代，最后在实车上与嵌入式，视觉组协调测试。第二步，我们制定了两个机器人间的协调策略，以及为了争夺buff点制定的策略。第三步，我们考虑了敌我血量，针对比如一个机器人阵亡等突发情况制定了相关对策。

# field simulation

采用python的pygame模块搭建，5m\*8m 的场地用500\*800的网格来表示。x方向最大长度为800，y方向最大长度为500. 显示机器人的各种状态（血量，是否射击，是否被击中，下一步的动作），在右边有两个小窗口分别是我方视角和敌方视角。



# data structures:

## info

机器人的状态用一个2\*4的数组表示，每行表示一个机器人，机器人的四个状态分别为其x, y坐标，剩余血量，和是否获得bonus。

## action

机器人每走一步（每步时间暂定为0.1s），机器人大脑（即策略部分）会根据当前我方及敌方机器人的方位和血量等信息产生下一步动作的决策。

下一步动作由一个2\*15的数组表示，每行表示一个机器人，每个机器人动作的[0:7]号元素代表x方向的位移，[7:14]号元素代表y方向的位移，[14]号元素代表是否射击。

对每个方向上的位移，有7个可能的动作，分别为沿该方向移动-3a, -2a, -a, 0, +a, +2a, +3a个单位长度。其中a=（1/每步时间内机器人能移动的最大距离）。

# states:

## 状态转移

如图所示，状态机部分共定义了四个状态，分别是Start，巡逻Patrol，追击Pursuit，站桩输出DPS。每个状态的具体转入转出条件如图所示，分别如下：

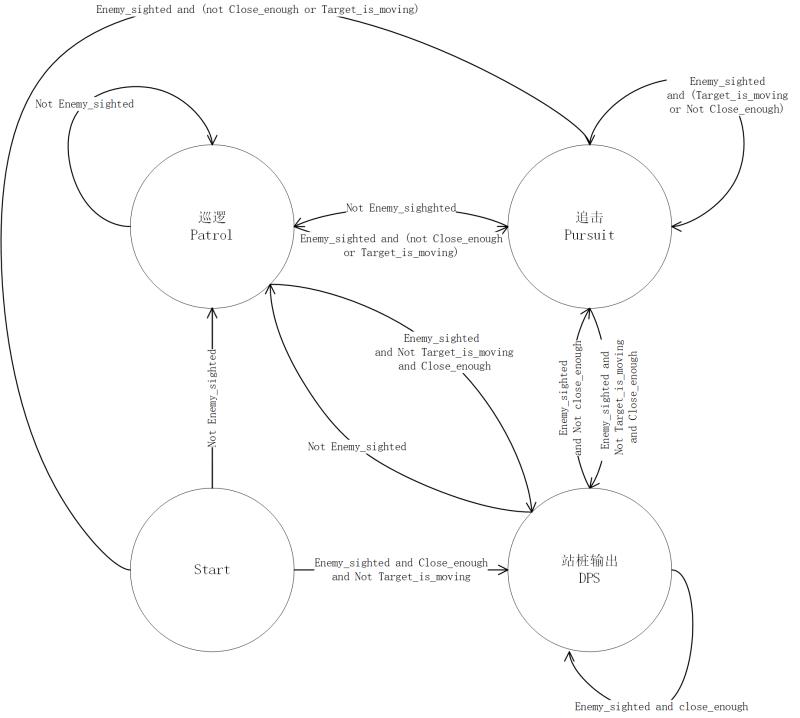
1. Start：初始状态转入，转出条件：·Not Enemy\_sighted（没有发现敌方）到巡逻Patrol，Enemy\_sighted and Close\_enough and Not Target\_is\_moving（没有发现敌方而且足够近而且目标不在移动）到站桩输出DPS，Enemy\_sighted and (not Close\_enough or Target\_is\_moving)（发现敌方而且不是足够近或者敌方在移动）到追击Pursuit。
2. 巡逻Patrol：转入条件：Not Enemy\_sighted（没有发现敌方），转出条件：Enemy\_sighted and (not Close\_enough or Target\_is\_moving)（发现敌方而且不是足够近或者敌方在移动）到追击Pursuit，Enemy\_sighted and Close\_enough and Not Target\_is\_moving（没有发现敌方而且足够近而且目标不在移动）到站桩输出DPS。
3. 追击Pursuit：转入条件：Enemy\_sighted and (not Close\_enough or Target\_is\_moving)（发现敌方而且不是足够近或者敌方在移动）到追击Pursuit，转出条件：Not Enemy\_sighted（没有发现敌方）到巡逻Patrol，Enemy\_sighted and Close\_enough and Not Target\_is\_moving（没有发现敌方而且足够近而且目标不在移动）到站桩输出DPS。
4. 站桩输出DPS：转入条件：Enemy\_sighted and Close\_enough and Not Target\_is\_moving（没有发现敌方而且足够近而且目标不在移动），转出条件：Not Enemy\_sighted（没有发现敌方）到巡逻Patrol，Enemy\_sighted and (not Close\_enough or Target\_is\_moving)（发现敌方而且不是足够近或者敌方在移动）到追击Pursuit。

根据目前的状态（视线中是否有敌人，敌人的距离及是否在移动）转移到三个实际运行状态：pursuit, dps, patrol中的一个。

当视线中没有敌方机器人，进入巡逻状态(patrol)。

当视线中至少有一个机器人静止在自己的打击范围内，进入站桩输出状态(DPS)。

当视线中没有机器人静止在自己的打击范围内，进入追击状态(pursuit)。



## start

初始状态。进行初始化及状态转移。

## pursuit

路径规划：采用A\*算法实现，来达到在最短时间中找到从自己到追击目标的最优路径，即避开障碍物的（近似？）最短路径。

* 每个机器人将独立选择追击视线中离自己较近的敌人。
* 用A\*算法找到从自己到目标敌人的最优路径。
* 机器人下一步的位置（动作）确定为在这条路径上的，一步能走到的最远的点。

在追击过程中持续发弹打击被追击的敌人。

## dps

机器人不移动，且持续发弹打击视线中距离自己最近的敌人。

## patrol

将场地分为左上，右上，左下，右下四个分区，机器人按A\*算法计算出从当前位置走到与自己所在分区为对角分区的路径。在该状态时机器人不发射子弹。