有禁行区的追捕策略设计与仿真

一、目的

在多追一的过程中，通过设计双方不同的策略，解决追逃问题

二、场景

1. 背景

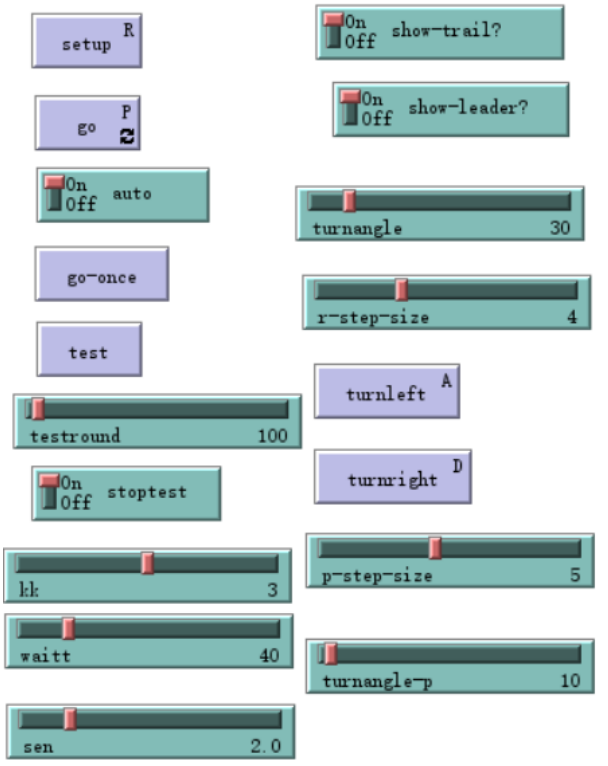
三艘船包围了一个小岛，小岛周围布置了水雷所以这三艘船无法进入，对方为了监测包围状况，每隔一段时间都会派一艘船出来，外边这三艘船的任务就是一旦有船出来就进行追捕，其它时间则在岛外巡逻。

2. 先决条件

- 逃跑者的船与追捕者的船行驶能力不同，即追捕者船的速度更快，逃跑者的转弯能力更强，也就是说逃跑者可以在即将被追上的时候通过突然的大角度转向逃避追捕。这主要是因为二者的加速功率不同并且重量也不一致从而导致
- 当对方进入小岛之后追捕者无法知道其具体位置

3. 相关参数

- 我们使用了Netlogo进行仿真，在仿真中，我们设置了以下参数（即图片中的各种选项，可以自己调整）：



（以下说明按照从左到右，从上到下的顺序）

- setup、go：初始化与运行
- auto：选择自动驾驶还是手动驾驶
- go-once：只运行1秒
- test：运行重复仿真
- testround、stoptest：重复仿真次数与终止重复仿真
- kk、waitt：当逃跑者进入小岛时，追捕者找出最佳等候位置算法的参数
- sen：逃跑者判定追捕者是否太近，要开始进行甩尾动作的参数
- show-trail：是否显示轨迹
- shoe-leader：是否显示逃跑者
- turnangle：逃跑者每秒可转动角度
- r-step-size：逃跑者每秒速度
- turnleft、turnright：手动驾驶模式下，控制逃跑者左转与右转
- p-step-size：追捕者每秒可转动角度
- turnagle-p：追捕者每秒速度

- 在我们的仿真中，假设大地图 150米 × 150米，逃跑者每秒可以跑 4米，转 30°；追捕者每秒可以跑 5米，转 10°。得分为逃跑者存活的秒数。

三、解决方案

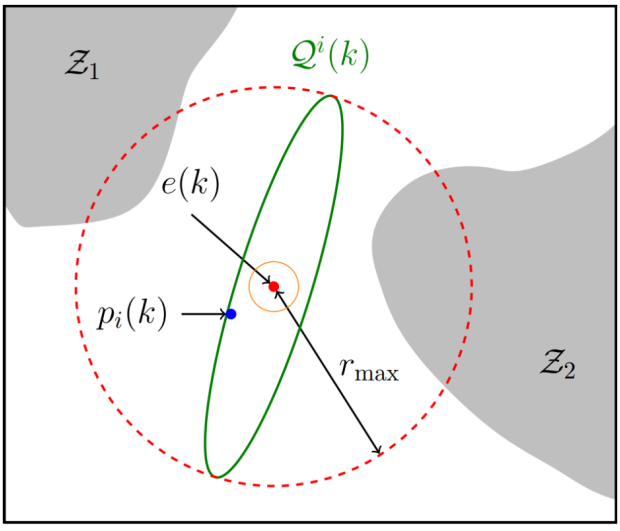
1. 逃跑者策略

- 若自己不在安全区，且计时器的数值过大，则以一定距离绕着安全区走
- 若自己不在安全区，且计时器的数值不大，则直接找最近点进入安全区
- 若自己在安全区，且计时器的数值过大，则直接找最近点离开安全区
- 若自己在安全区，且计时器的数值不大，则在安全区内随机乱走
- 当追捕者太近时，就要采取方式甩掉他们。大致就是找到追捕者来的方向，立马 往此方向的垂直方向转动。

2. 追捕者策略

①逃跑者在海洋

定义p_i (k)与e(k)为追捕者i和逃跑者在k时刻的坐标，考虑这样一种情况，即逃跑者是静止的，并且离追踪者足够近，因此存在一个以逃跑者的位置为中心的椭球体且完全在领海之外，我们把这个椭球称为“ 安全椭球”。 令r_{min} 和r_{max} 为安全椭球半长半短轴的上下界（如下图所示）



令v_κ ，$\kappa = 1, \ldots, n_v$ 为领海上距离K小于等于r_{max}的一组点集，对于追捕者i，定义安全椭球为：

$$Q^i(k) \triangleq \{z \in R^2 | (z - e(k))^T Q_\epsilon^i (z - e(k)) \leq 1\}$$

此时$Q^i(k)$就是半定规划的解：

$$\begin{aligned} & min \quad trace(Q_\epsilon^i) \\ s.t. \ & (v_\kappa - e(k))^T Q_\epsilon (v_\kappa - e(k)) \geq 1 + \epsilon, \\ & (p_i(k) - e(k))^T Q_\epsilon^i (p_i(k) - e(k)) \leq 1 - \epsilon, \\ & \kappa = 1, \ldots, n_v, \quad \frac{1}{r_{max}^2} I \preceq Q_\epsilon^i \preceq \frac{1}{r_{min}^2} I \end{aligned}$$

但是当追捕者和逃跑者的连线过小岛或者二者的距离大于r_{max} 的时候，上面的方法就不成立了，在这种情况下，我们设定了一个虚拟追捕者，它可以使追捕者朝着逃跑者的方向运动。我们记虚拟追捕者e_i^* 为追捕者i对应的虚拟者，定义Z为小岛，因为小岛可能是非凸的，我们计算一个可以包含小岛的体积最小的椭球ϵ_{Z_r} ，这个椭球距离领海最近的距离是r_{min} 。

令$line(p_i, e)$为从p_i到e的线段，定义函数$los(p_i, e)$在线段穿过小岛为0，没穿过为1。

下面的算法判断上面的优化是否可行，如果不可行，则分配一个虚拟追逐者，并不断重复，直到最后可行。

Procedure 1 Proximity Verification Procedure

Input: $p_i, e, \mathcal{Z}_j, \mathcal{E}_{\mathcal{Z}_j}$ $(j = 1, \ldots, n_z)$

Output: e_i^* ▷ position of the dummy evader for pursuer i

1: **procedure** PROXIMITY(p_i, e)
2: $(x_z, d_z, k) \leftarrow$ the closest point along the boundary of $\mathcal{Z}_j$ to $\texttt{line}(p_i, e)$, its distance, and the corresponding no-fly zone number, respectively
3: **if** $d_z \geq r_{\min}$ **then**
4: $e_i^* \leftarrow e$
5: **else**
6: $e_z \leftarrow$ the nearest point on $\mathcal{E}_{\mathcal{Z}_k}$ to x_z with $\texttt{los}(x_z, e_z) = 1$
7: $e_i^* \leftarrow e_z$
8: **end if**
9: **end procedure**

下面的算法是完整的路径规划算法：

Algorithm 1 Path Planning Algorithm for Pursuer p_i

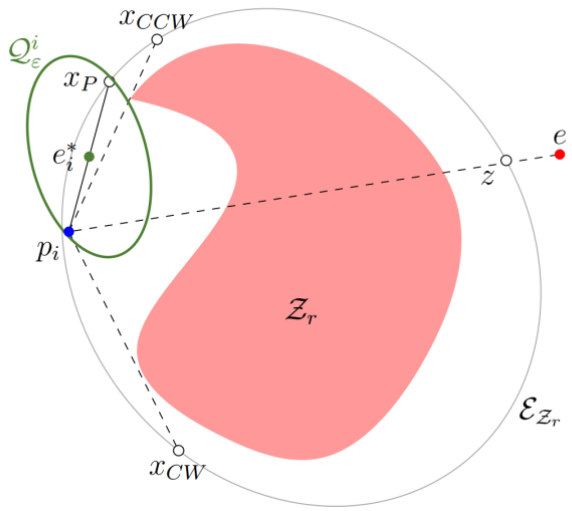
Input: $p_i(k), e_i(k), \mathcal{Z}_j, \mathcal{E}_{\mathcal{Z}_j}\ (j = 1, \ldots, n_z)$
Output: $e_i^*(k), Q_\varepsilon^i$ ▷ position of dummy evader for p_i

Pick $0 < \theta_p < 1$
$e_i^* \leftarrow e_i(k)$ ▷ $e_i(k) = e(k)$ if $e(k)$ is in free space
$\mathcal{L}_1 \leftarrow 1$, $\mathcal{L}_2 \leftarrow 1$
while $\mathcal{L}_1 = 1$ **do**
  if e_i^* is inside a no-fly zone r **then**
    $e_i^* \leftarrow$ nearest point to e_i^* on the boundary of $\mathcal{Z}_r$
  end if
  if $\texttt{los}(p_i, e_i^*) = 1$ **then**
    $e_i^* \leftarrow \text{PROXIMITY}(p_i, e_i^*)$
  end if
  if $\texttt{los}(p_i, e_i^*) = 0$ **then**
    $r \leftarrow$ index of closest obstacle along $\texttt{line}(p_i, e_i^*)$
    $z \leftarrow$ nearest point to e_i^* on $\mathcal{E}_{\mathcal{Z}_r}$
    $x_{\{CW,CCW\}} \leftarrow$ nearest point to z on $\mathcal{E}_{\mathcal{Z}_r}$ in CW and CCW directions with $\texttt{los}(p_i, x_{\{CW,CCW\}}) = 1$
    $e_i^* \leftarrow$ Pick x_{CW} or x_{CCW} that yields a shorter arc to z
    $e_i^* \leftarrow \text{PROXIMITY}(p_i, e_i^*)$
  else
    while $\mathcal{L}_2 = 1$ **do**
      $Q_\varepsilon^i \leftarrow$ Construct max-volume ellipsoid
      if $Q_\varepsilon^i = \emptyset$ **then**
        $x_d \leftarrow \theta_p\, p_i + (1 - \theta_p) e_i^*$
      else
        $\mathcal{L}_1 \leftarrow 0$, $\mathcal{L}_2 \leftarrow 0$
      end if
    end while
  end if
end while

②逃跑者进入小岛

逃跑者一旦进入小岛，追踪者可能没有关于逃兵位置的可靠信息，所以追踪者根据逃兵的最大速度和它的入口点，构建一组可到达的可能位置。追击者们在小岛的周边安排好了自己的位置，准备在逃兵出现时将其抓获。我们采用了基于冯诺依曼的覆盖控制策略来划分逃跑者的可达集，并在不进入小岛的情况下，将追踪者尽可能靠近可达集的中心形Voronoi 镶嵌。

如下图所示：



对于小岛Z_j，q是其中的任一点，v_{max} 是K的最大速度，K在$t=\tau$时刻进入小岛，进入点记为$e(\tau)$ ，这之后我们我们可以得到以$v_{max}(t-\tau)$为半径$e(\tau)$为圆心的圆，这个圆表示着K以最大速度逃跑的可达范围，我们定义这个圆与小岛的重合部分为$R_j(t,\tau,v_{\max})$， 也就是说K一定在这个范围里，即：

$$R_j(t,\tau,v_{\max})=\mathrm{B}(e(\tau),v_{\max}(t-\tau))\bigcap\mathbb{Z}_j$$

同时定义函数ρ：

$$\rho(q,t)=\begin{cases}1,if\ \ q\in\ R_j(t,\tau,v_{\max}),\\0,otherwise.\end{cases}$$

如果追踪者可以在小岛内自由移动，我们会想要分配追踪者到任何可能的位置的平均距离最小化。

令$\overline{p_i}$是追捕者的目标位置，定义基于这些目标位置的可达集R_j 的Voronoi 镶嵌：

$$V_i=\{q\in R_j|\ \ \|q-\overline{p_i}\|^2\le\|q-\overline{p_k}\|^2,k=1,2,\ldots,n_p\}$$

在此基础上，我们定义一个捕获成本：

$$\nu(\overline{p_1},\ldots,\overline{p_n})=\sum_{i=1}^{n_p}\int_{V_i}\|q-\overline{p_i}\|^2\rho(q,t)dq$$

我们可以看出，ν越小则越容易追捕。

我们定义Voronoi镶嵌的质量和质心：

$$M_{V_i}=\int_{V_i}\rho(q,t)dq,\quad C_{V_i}=\frac{1}{M_{V_i}}\int_{V_i}q\rho(q,t)dq$$

上面的捕获成本是对于虚拟追捕者而言的，考虑到我们的追踪者必须留在小岛之外，我们必须调整移动到重心的算法来适应这种受限的情况。在此时，对于虚拟追捕者$\overline{p_i}$而言， 定义真正的追捕者$p_i=\overline{p_i}+d_i$ ，那么我们可以定义真正的捕获函数：

$$H(d_1,\ldots,d_n)=\sum_{i=1}^{n_p}\int_{V_i}\|q-\overline{p_i}-d_i\|^2\rho(q,t)dq$$

其中$\overline{p_i}=C_{V_i}$，同时我们的问题就化为了以下的优化问题：

$$\min_{(d_1,\ldots,d_n)} \quad H(d_1,\ldots,d_n)$$
$$s.t. \quad d_i + \overline{p_i} \notin \mathbb{Z}_j$$

通过求解此优化问题可以得到如下的路径规划算法：

Algorithm 2 No-Fly Zone Planning Algorithm

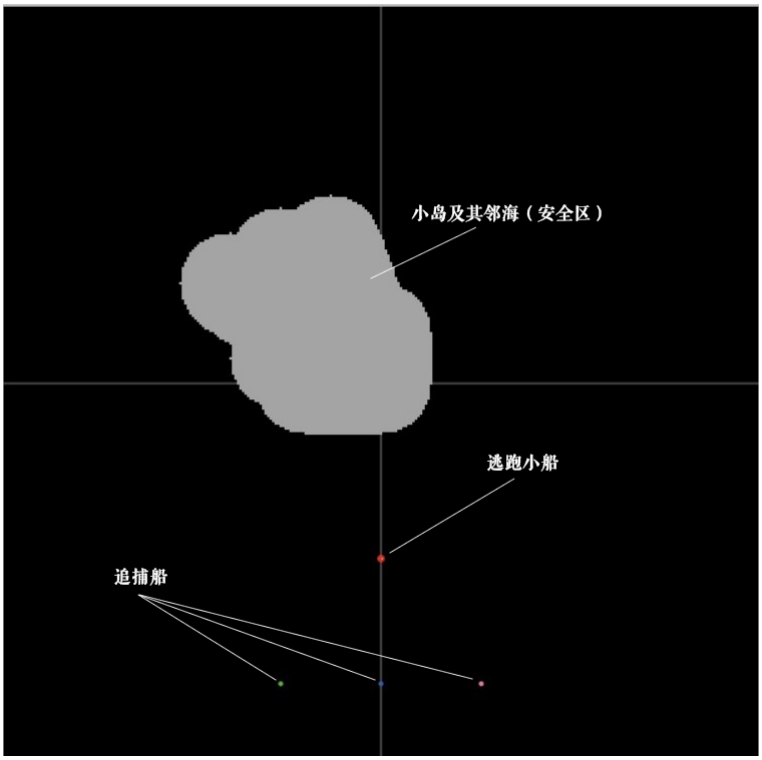
Input: $e(\tau)$, t, $\mathcal{Z}_j$, v_{max}
Output: $\bar{p}_i$

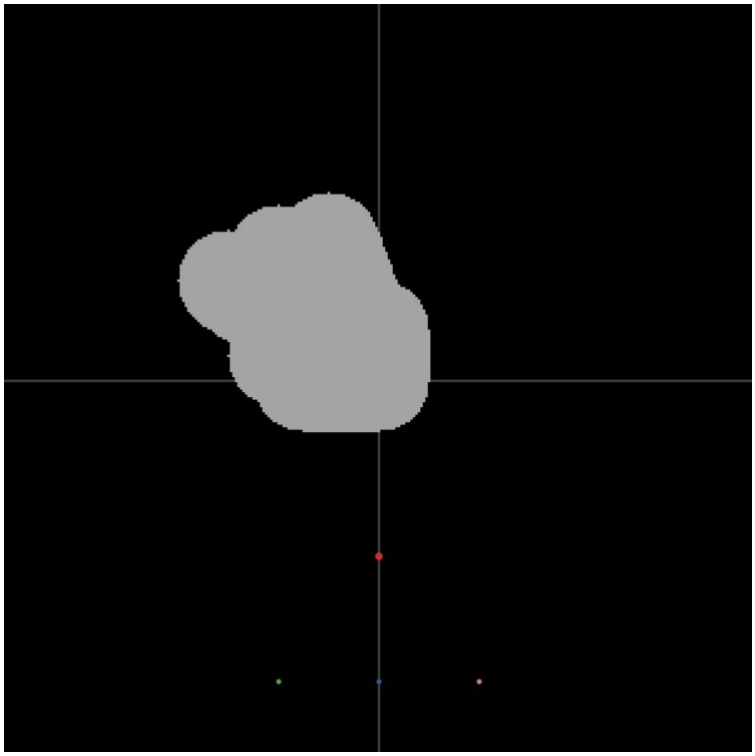
Calculate $\mathcal{B}(e(\tau), v_{\max}(t-\tau))$ and $\mathcal{R}_j(e(\tau), v_{\max}(t-\tau))$
Compute Voronoi tessellation about $\bar{p}_i$
while $\bar{p}_i \neq C_{V_i}\ \forall\ i$ **do**
    Assign $\bar{p}_i = C_{V_i}$
    Recompute Voronoi tessellation
end while
Assign $\bar{p}_i$ as target points for p_i

四、仿真

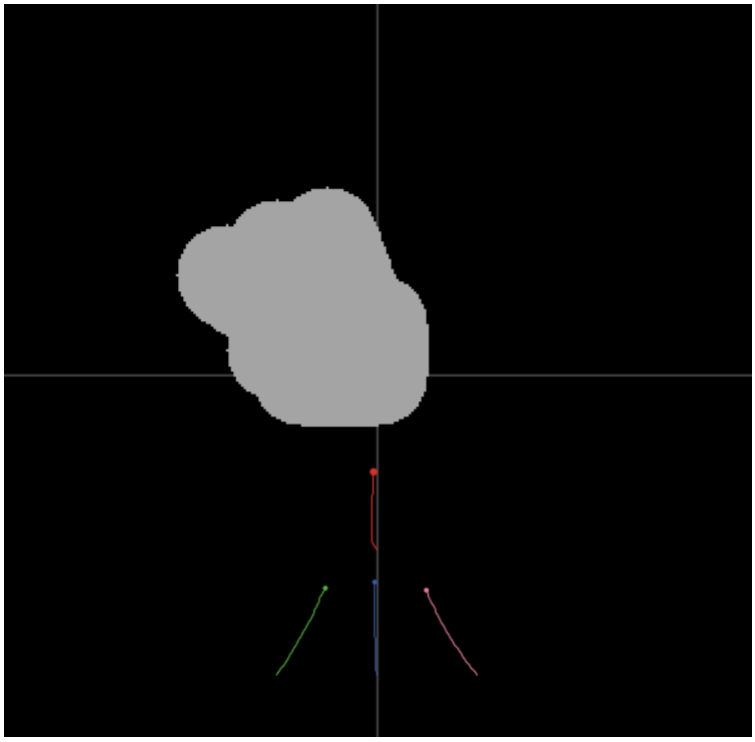
我们使用Netlogo软件进行仿真

- 首先进行初始化

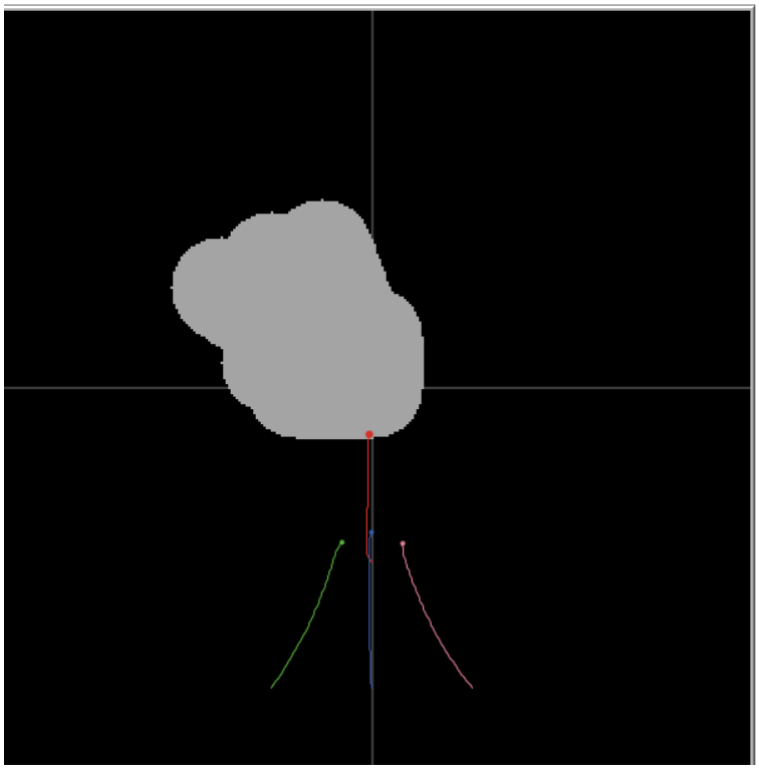


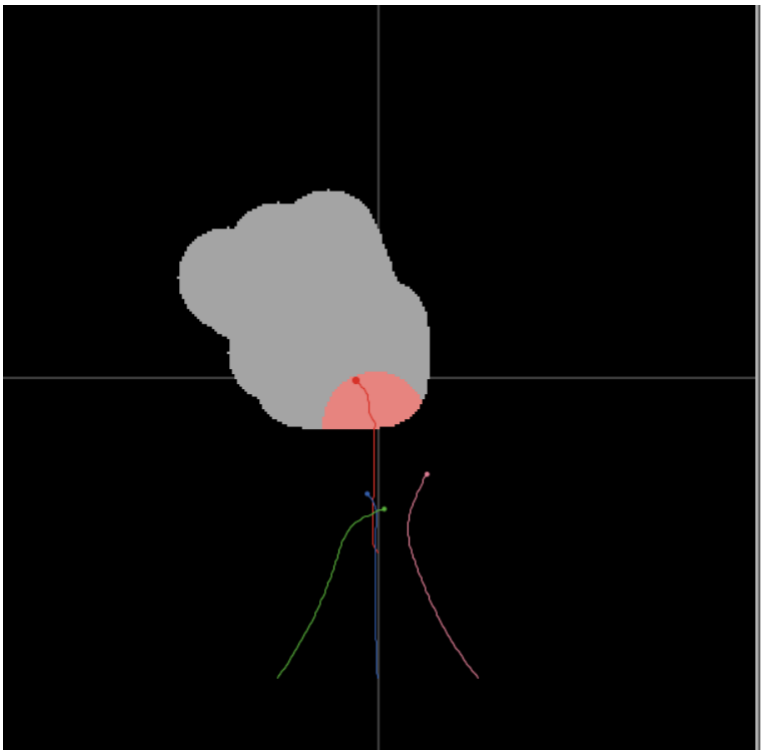


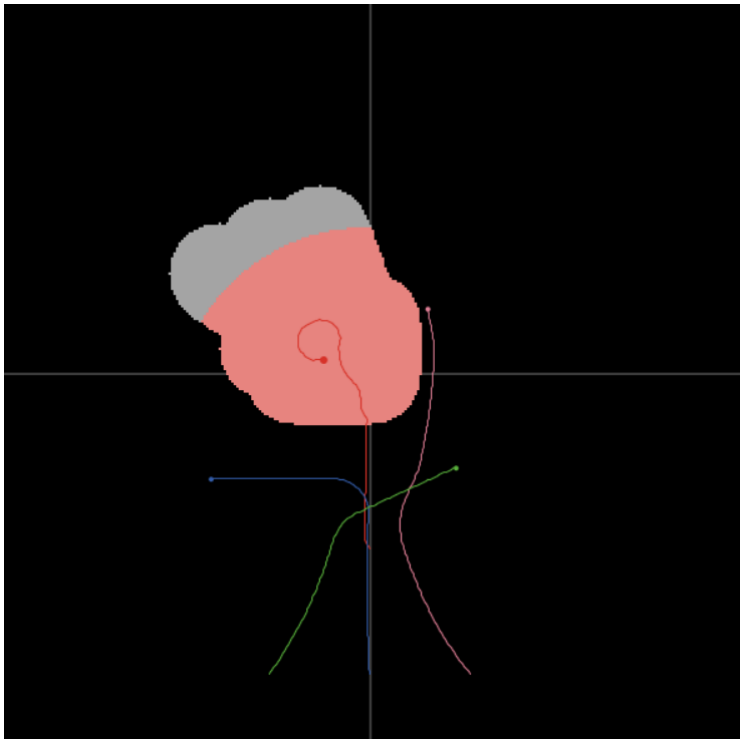
- 然后开始运行：



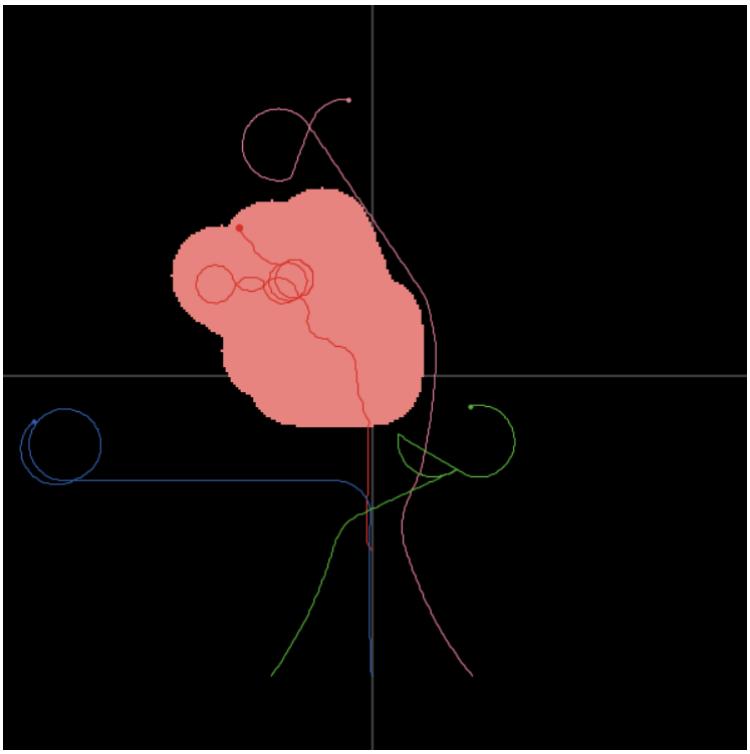
- 逃跑者进入小岛：

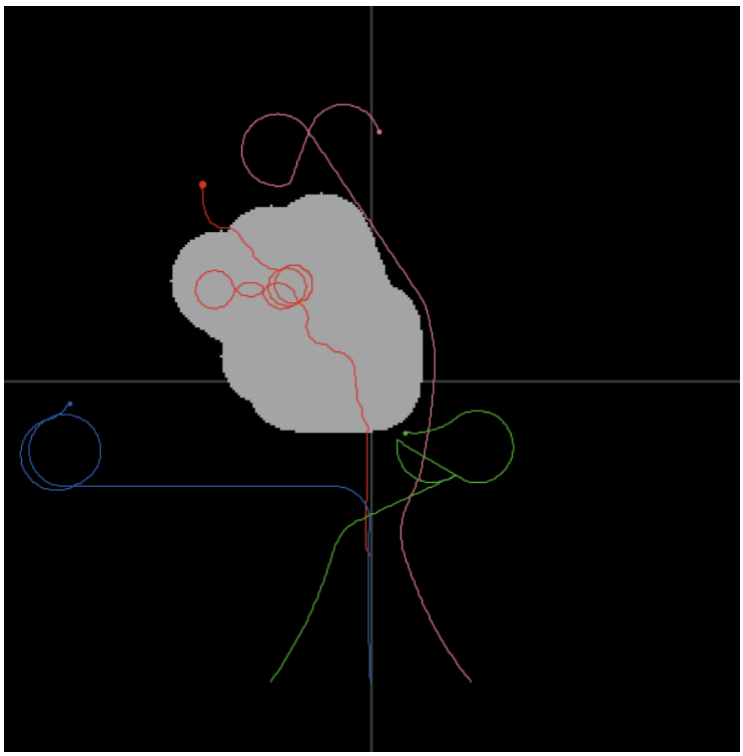




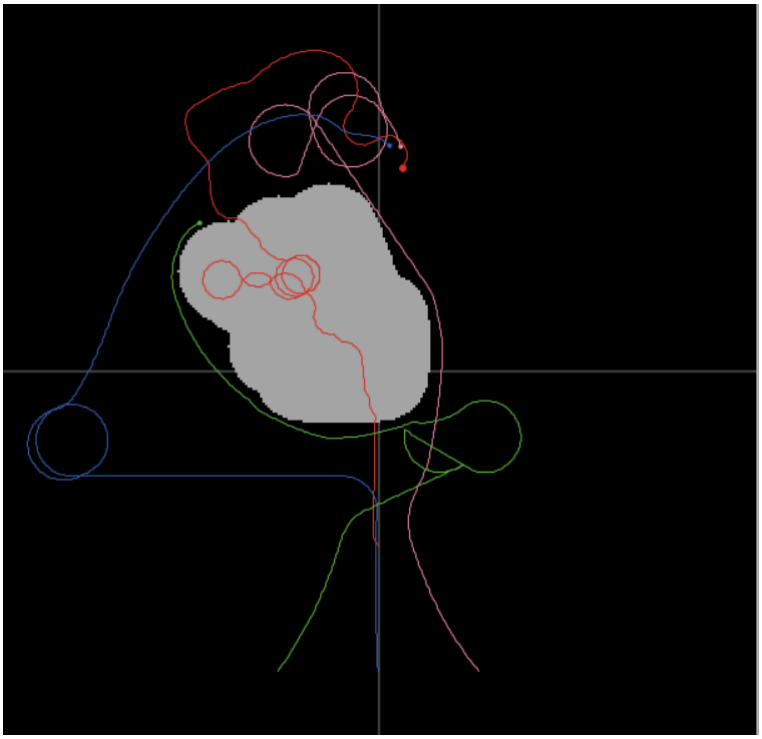


- 即将出小岛

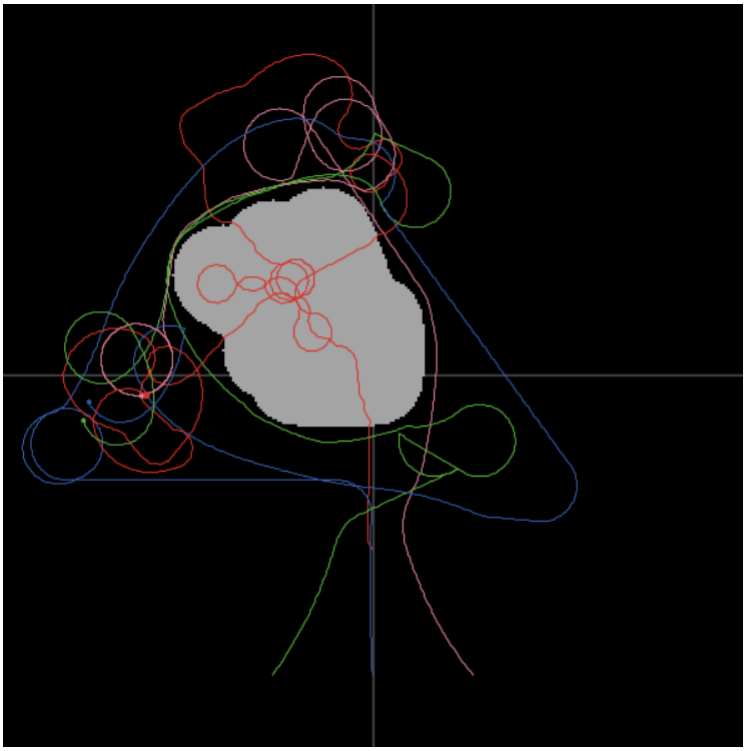




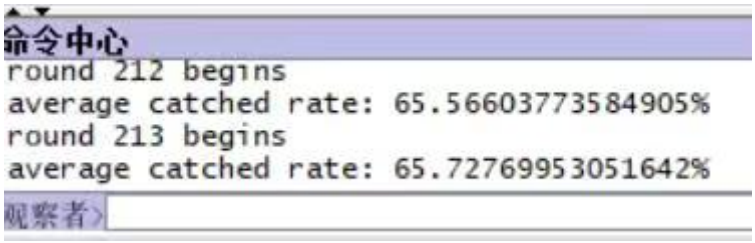
- 在海洋继续追捕：



- 追捕成功



- 此外，我们还进行了200+次的尝试，得到以下结果：



可以看出进行了213次实验之后，追捕成功率在66%左右，说明双方势均力敌，该问题有讨论价值。