强化学习

第九讲：COG RoboMaster Sim2Real 竞赛作业

教师：赵冬斌　朱圆恒

教师助教：李浩然

1 中国科学院大学人工智能学院

2 中国科学院自动化研究所

April 25, 2025

课程目录

- 2022 COG RoboMaster Sim2Real 竞赛

  比赛背景

  比赛介绍

  基线算法1

  基线算法2

  实体机器人测试

  参赛队算法分析

竞赛组织者：李浩然[1,2]，陈亚冉[1,2]，刘莎莎[1,2]，郑博培[12]，曾令泽[12]，赵冬斌[1,2]

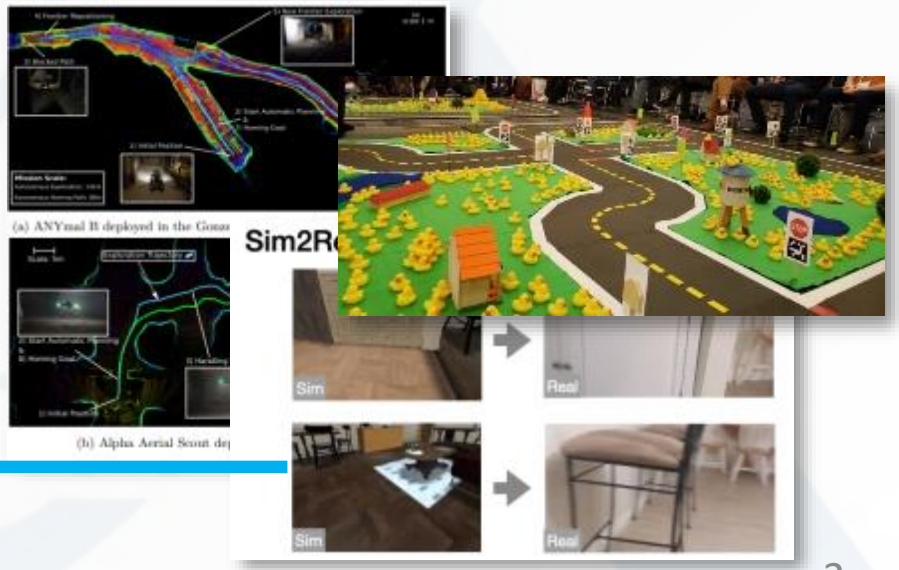
2022 CoG Robomaster Sim2Real 竞赛

机器人在生产和生活中发挥着越来越重要的作用




机器人自主探索和sim2real是当前人工智能领域的研究热点

- 2019-2021持续3年的DARPA地下挑战赛
- CVPR 2020 Sim2Real Challenge with iGibson
- NeurIPS 2021 AI Driving Olympics

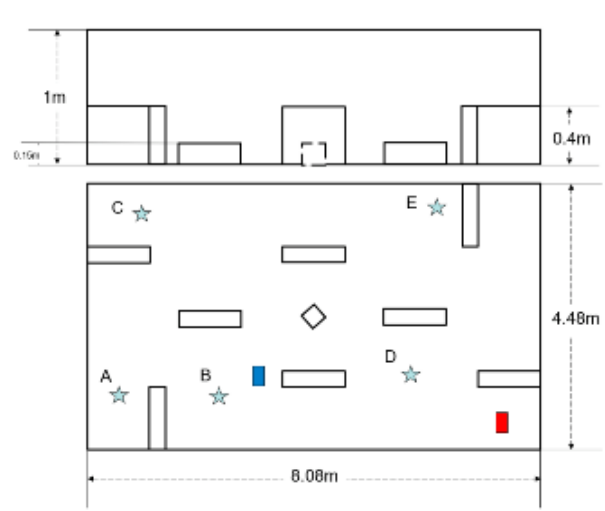


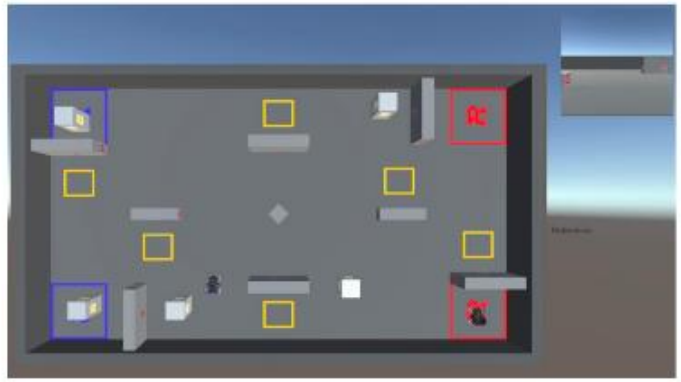
比赛介绍

开发了一个框架[1]，具有速度快的敏捷物理机器人，并可用于训练机器人导航和对抗策略。

任务：参赛机器人在固定场地的随机初始位置开始，寻找在场地中随机生成的5个目标点，并按照固定顺序依次完成激活。此后，防守机器人被激活开始射击对抗。

目标：在3分钟内，以最快并且安全的速度完成5个目标点的激活，会对参赛机器人进行攻击，参赛机器人需要尽可能地对防守机器人射击并保持自身血量





评价指标：

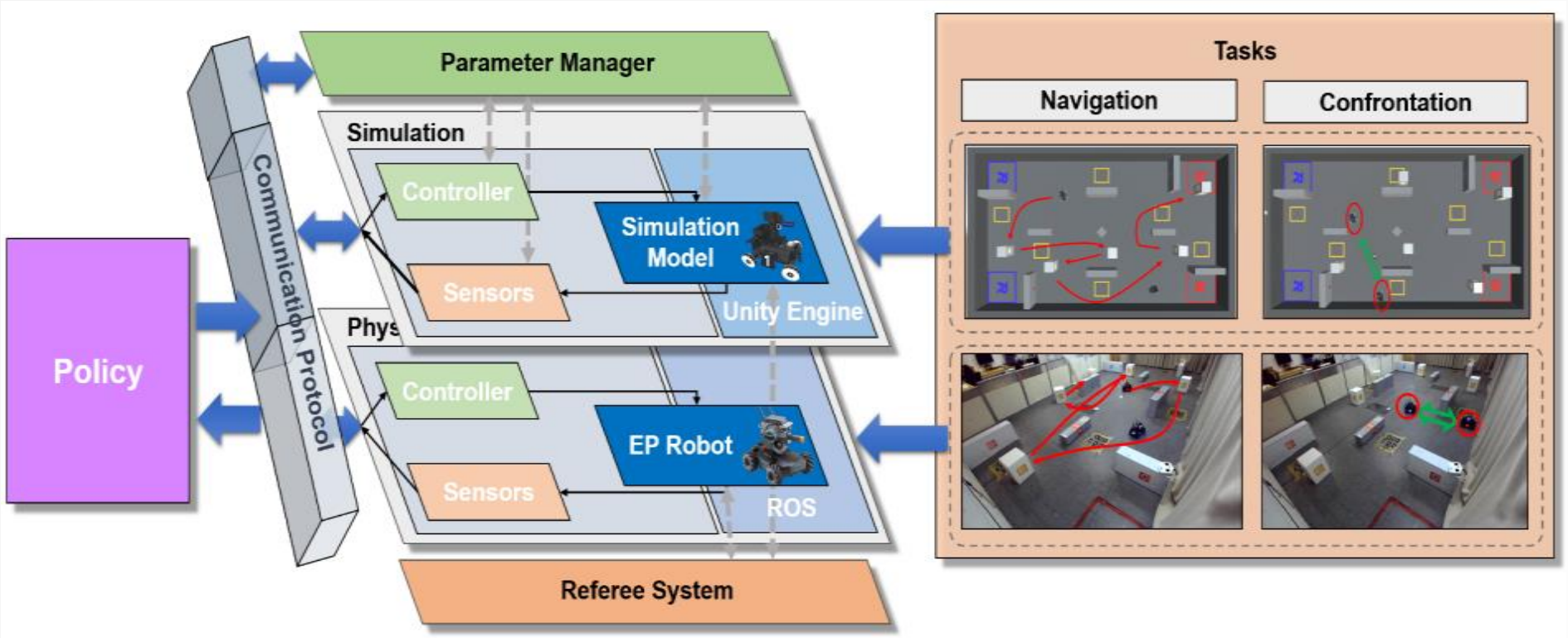
$$Score=60\times N + A\times 0.5\times (D+H) - T-10K$$

N是成功激活目标点数量，T是所用时间，K为碰撞次数，D为对方受到伤害，H为机器人剩余血量

[1]https://github.com/DRL-CASIA/NeuronsGym

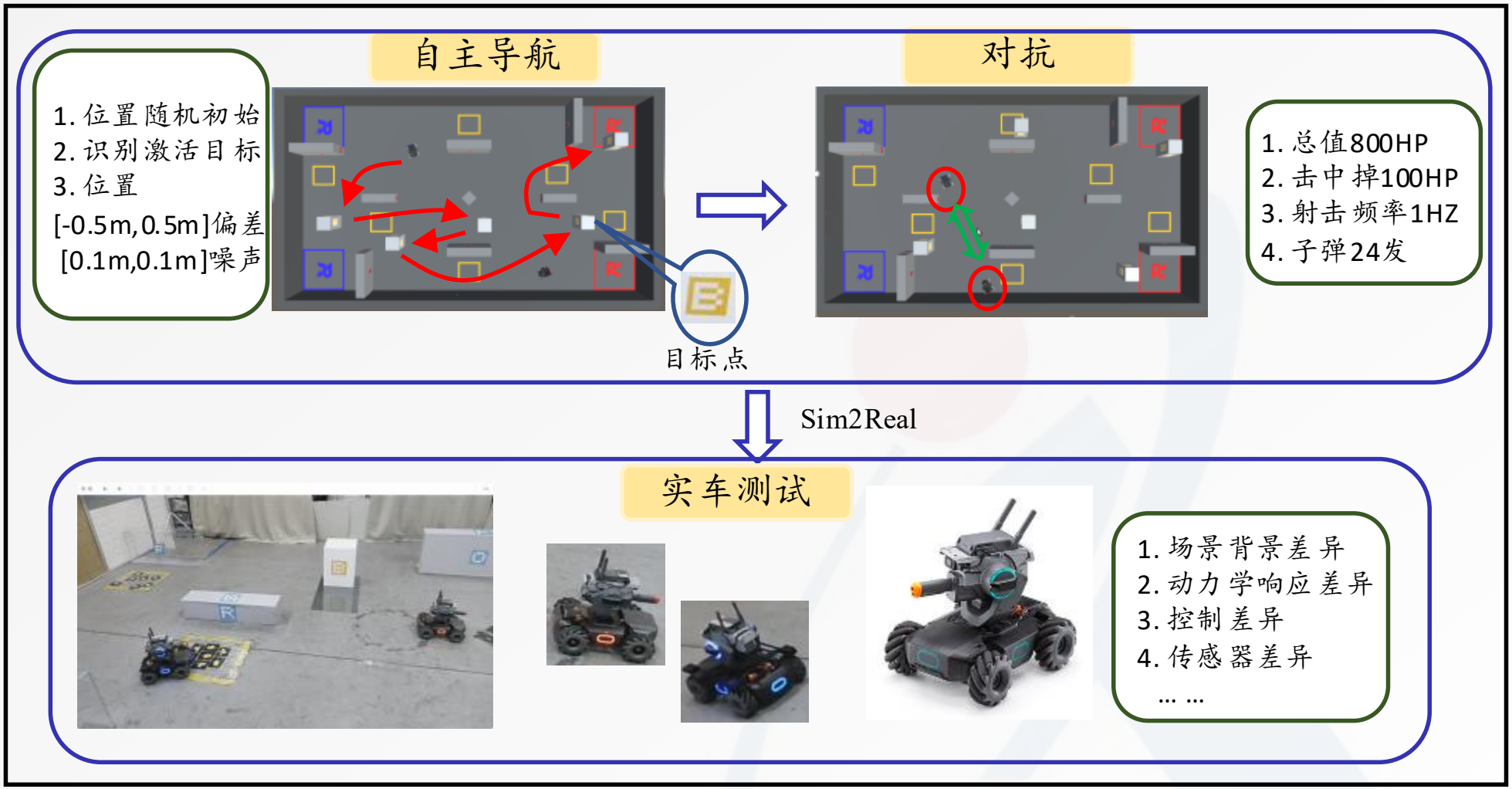
https://eval.ai/web/challenges/challenge-page/1513/overview

比赛介绍



- 仿真系统：Unity3D的仿真平台，包括多种场景、机器人模型、控制器和各种传感器。
- 实体系统：RoboMaster EP、相机、雷达
- 参数管理系统：摩擦系数、电机特性、控制器参数、机器人的重量、控制响应延时

比赛介绍



- 仿真场景存在位置等观察量的噪声和偏差，传统的导航算法对位置精度依赖较高 ⇨ 考虑RL算法
- 实体场景和仿真场景存在差异 ⇨ 考虑Sim2Real 方法

比赛介绍

Track 1

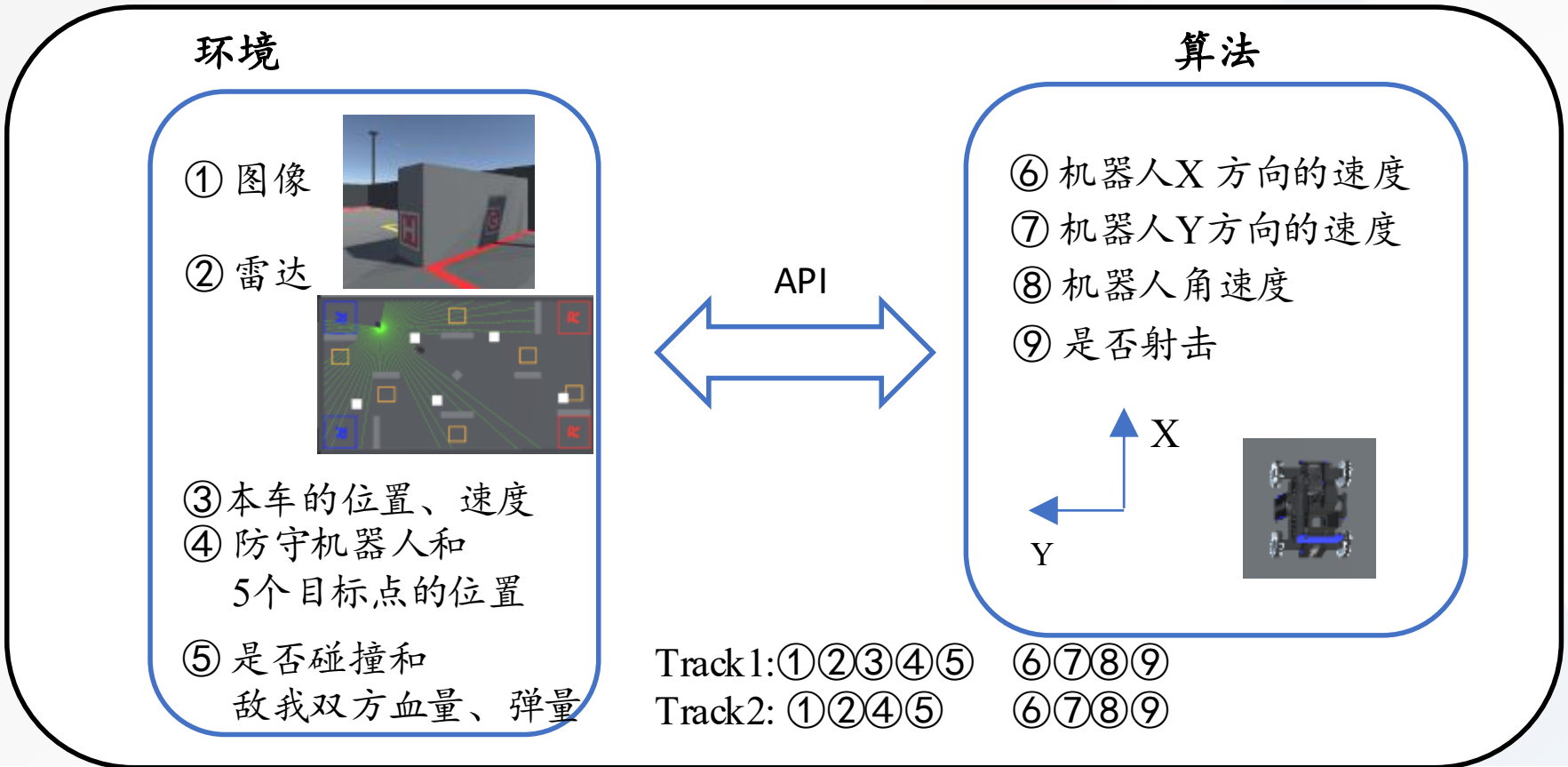
EP机器人可获取状态包括①②③④⑤：本车在地图中的位置，速度；当前时刻的图像；目标点的位置；

算法应输出⑥⑦⑧⑨机器人的速度控制指令和是否射击指令。

Track 2

EP机器人可获取状态包括①②④⑤：当前时刻的图像；目标点的位置；

算法应输出⑥⑦⑧⑨机器人的速度控制指令和是否射击指令。

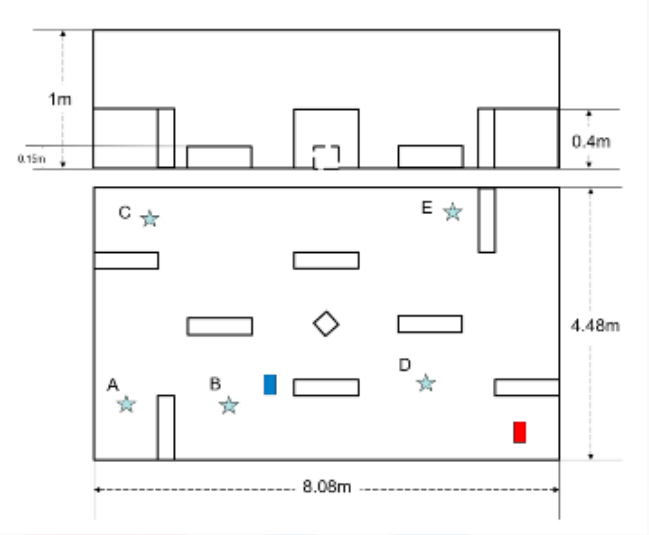


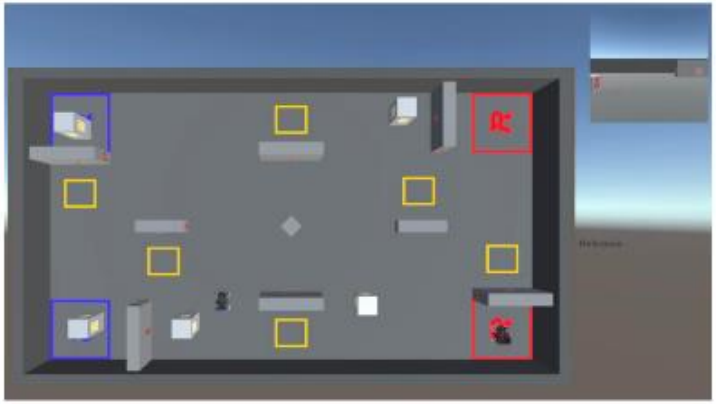
比赛介绍

目标块激活条件

比赛过程中参赛机器人需要**按照顺序依次进行激活**，目标块激活需要满足下述所有条件：

- 参赛机器人与目标块的距离**小于1m**
- 参赛机器人与目标块之间**没有障碍物**
- 参赛机器人在世界坐标系下的朝向与机器人与目标块的连线所构成的夹角**小于30度**
- 目标块是按照ABCDE的顺序激活(例如抵达目标块C之前，依次抵达过AB，否则目标块无法激活)

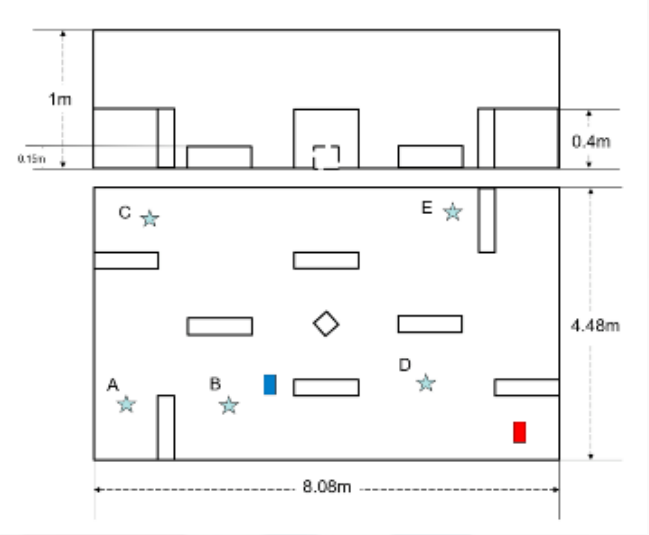


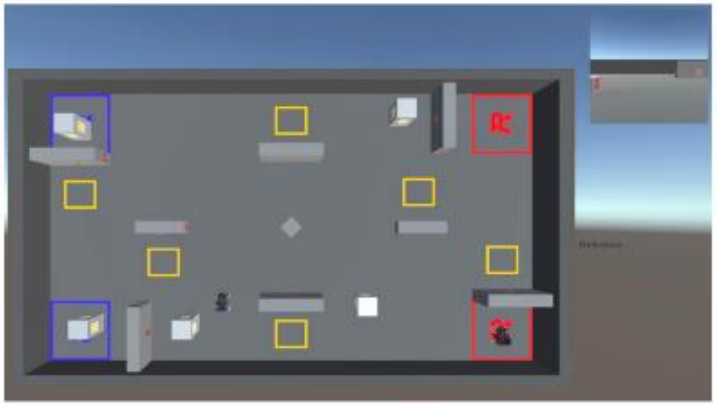


比赛介绍

机器人对抗规则

- 比赛开始时，防守机器人作为固定障碍物出现在场地中，只有**当五个目标块都激活完成之后，比赛进入对抗阶段**，防守机器人才被激活，开始移动与参赛机器人射击对抗。
- 对抗开始时，防守机器人和参赛机器人的初始血量为800，双方机器人的射击频率为1Hz，**被击中一次掉血为100**。每个机器人可**发射射击指令次数为24次**，当下发射击指令多于24次时，射击指令将不会被执行。





比赛介绍

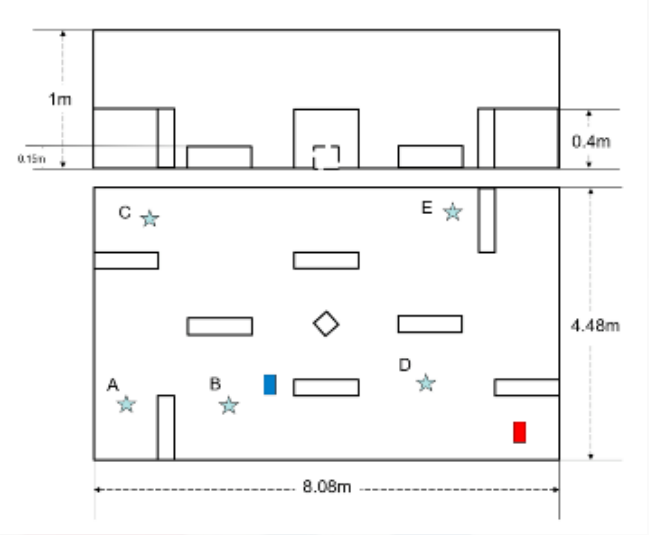
计分规则

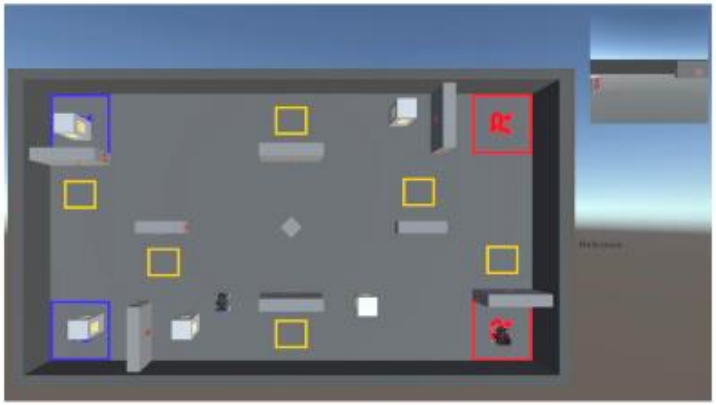
参赛机器人的得分Score=60×*N*+0.5×(*D*+*H*)-*T*-10*K*，考虑以下四个部分：

- **寻找目标的得分**：60×*N*，按顺序激活一个目标获得奖励60，*N*为成功激活目标的个数，最高得分为300；
- **防守机器人对抗的得分**：0.5×(*D*+*H*)，敌我双方初始血量为800，比赛结束时敌方机器人的伤害为*D*=800-防守机器人血量剩余，参赛机器人的血量剩余为*H*，希望参赛机器人在保持不被击中的情况下，尽可能的对防守机器人造成伤害，最高得分为800；
- **比赛总用时**：*T*(s)，希望参赛机器人快速完成任务，用时越长，得分越低，最多扣减为180；
- **碰撞惩罚**：$K = 2 \times T_k$，T_k为连续碰撞时间(单位为秒)，碰撞时间越长，得分越低，最多扣减3600。

例如：参赛机器人成功到达5个目标块后，与防守机器人对抗并取胜，剩余血量100，总用时150s，发生碰撞时间5秒，那么其总得分为

60×5+0.5×(800+100)-150-10×(2×5)=500。



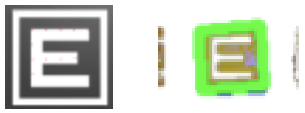


基线算法1

仿真

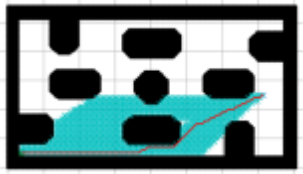
感知

AprilTag
目标识别



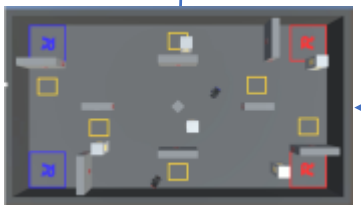
规划

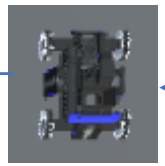
A* 规划算法



决策控制

有限状态机





实体

目标识别
定位算法

A*规划算法

有限状态机

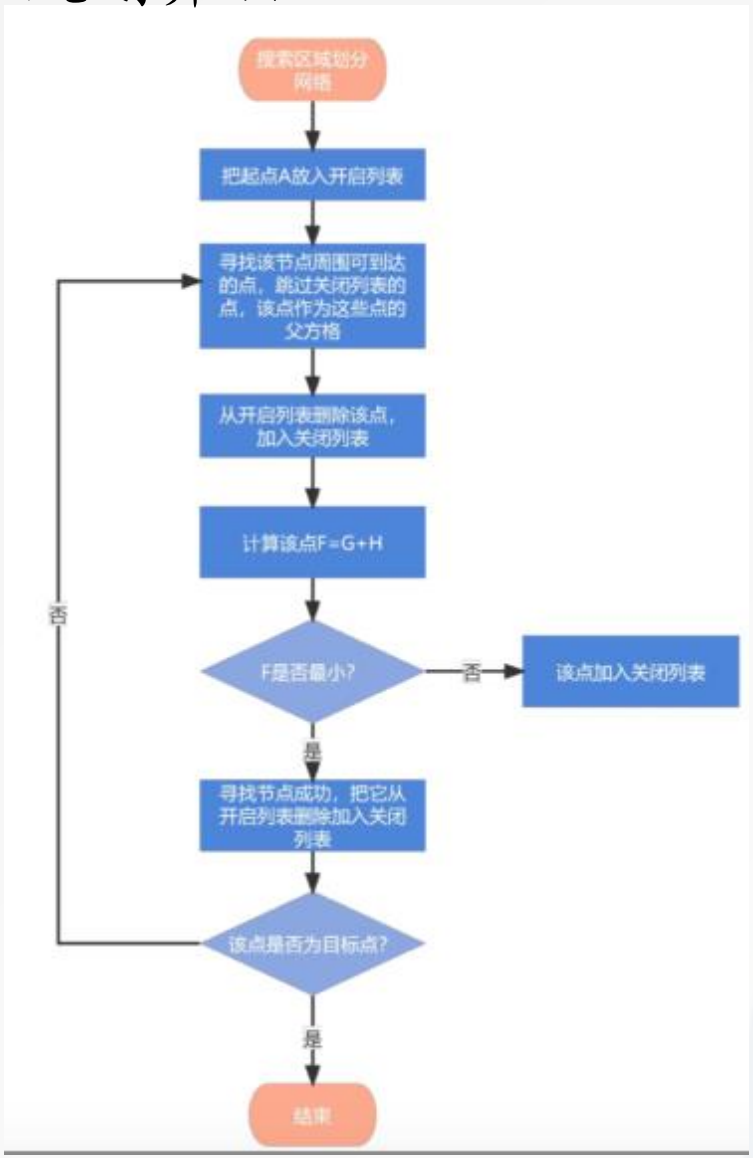
背景差异
相机内外参差异

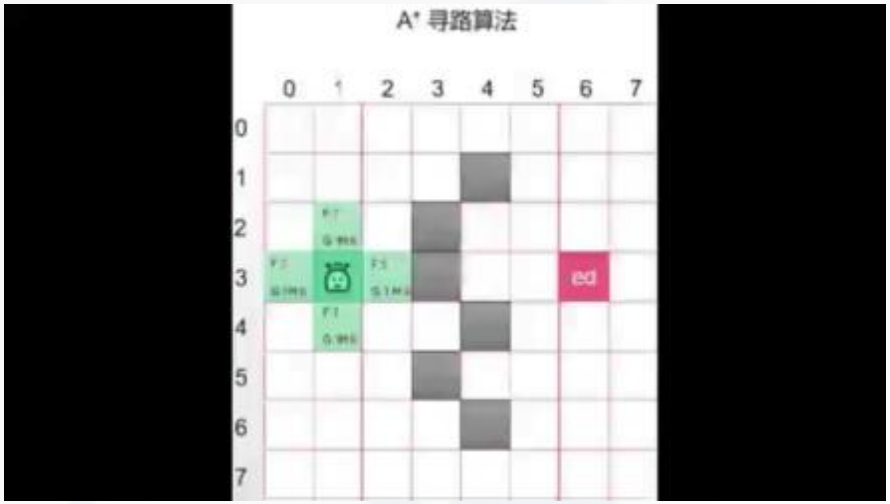
运动学响应差异

机械控制差异

基线算法1

A* 规划算法





F = G+H

G：当前点与起始点的距离

H：点前点与终点的距离

基线算法1

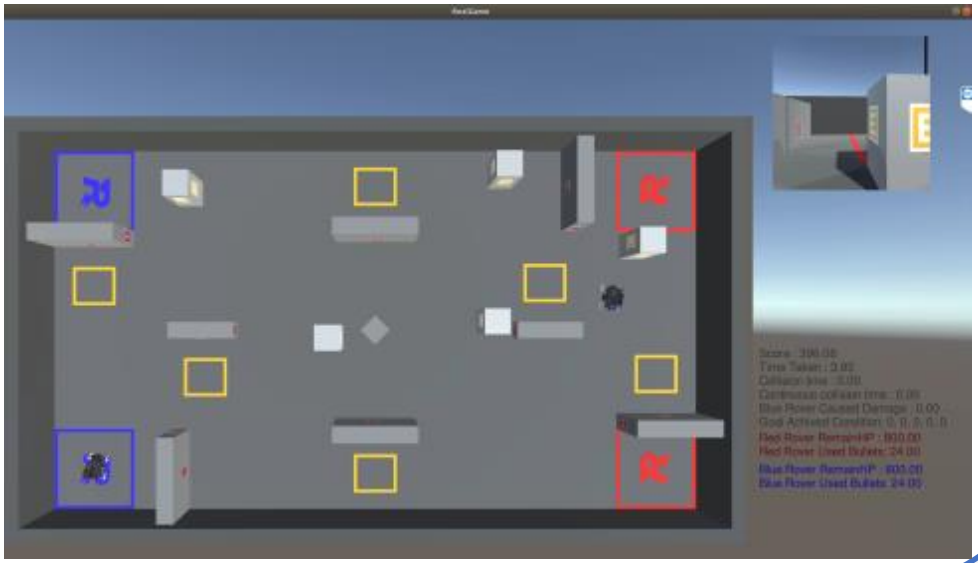
仿真结果

不加噪声
得分：328.66
时间：51.96
连续撞击时间：2.96s
红色伤害：400HP
剩余血量：0HP

(统计5次实验
连续撞击时间平均值**3.84s**)

加噪声[-0.1,0.1]
得分：70.42
时间：43.59
连续撞击时间：8.40
红色伤害：500HP
剩余血量：0HP

(统计5次实验
连续撞击时间平均值**7.88s**)



基线算法1

问题-位置出现较大偏差

里程计累计误差越来越大

地面光滑空转导致误差

撞击导致误差

无法完成任务

60%概率位置偏差
无法完成任务

状态估计（位置）存在偏差
传统方法验证依赖于状态

强化学习方法，输入可以考虑：位置、图像、雷达

基线算法2

基于强化学习的导航

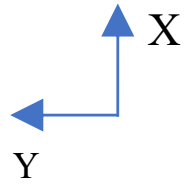
- **环境State:**
  - 本车的精确位置
  - 目标点的位置
- **Reward**：是否碰撞,距离目标点的距离（在目标点附近速度很小，且停留0.5s）

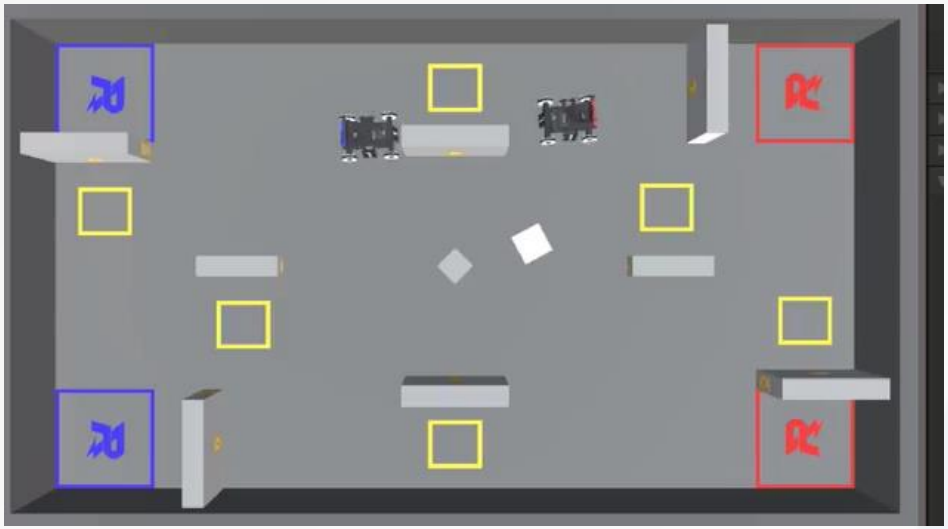
- **输出：**

机器人X 方向的速度

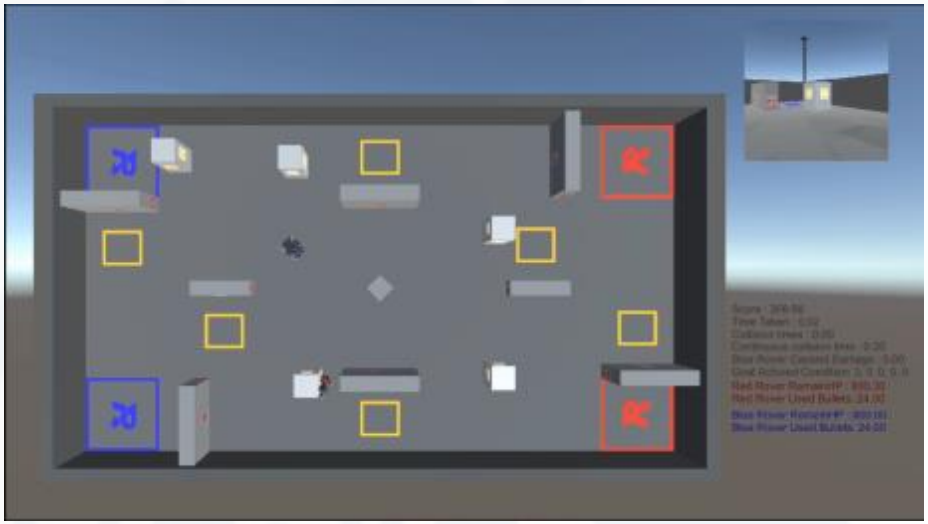
机器人Y方向的速度

固定**朝向目标点**





训练训练结果（1.5倍速）
快速到达5个目标点：时间10s



迁移到5个目标点的导航

基线算法2

基于强化学习的导航

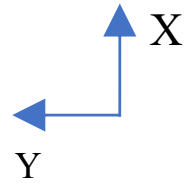
- **环境State:**
  - 本车的精确位置
  - 目标点的位置
- **Reward**：是否碰撞,距离目标点的距离（在目标点附近速度很小，且停留0.5s）

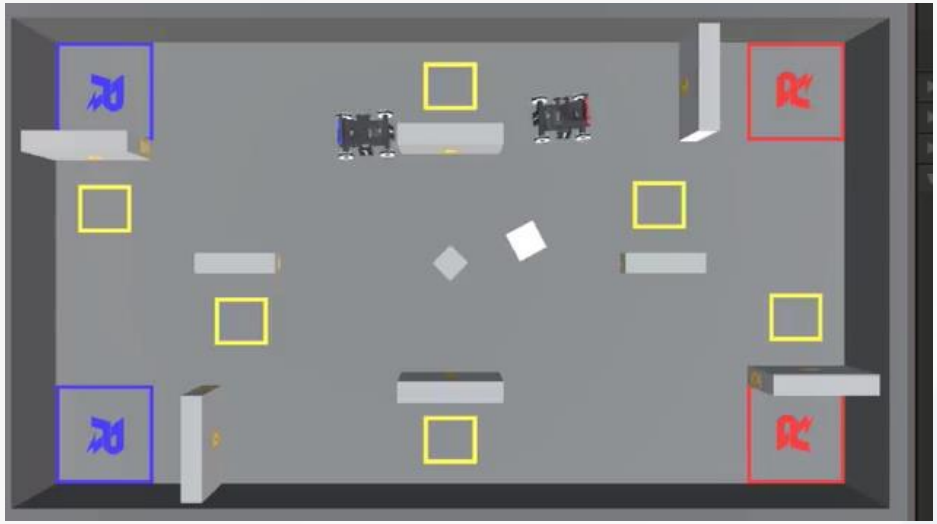
- **输出：**

机器人X 方向的速度

机器人Y方向的速度

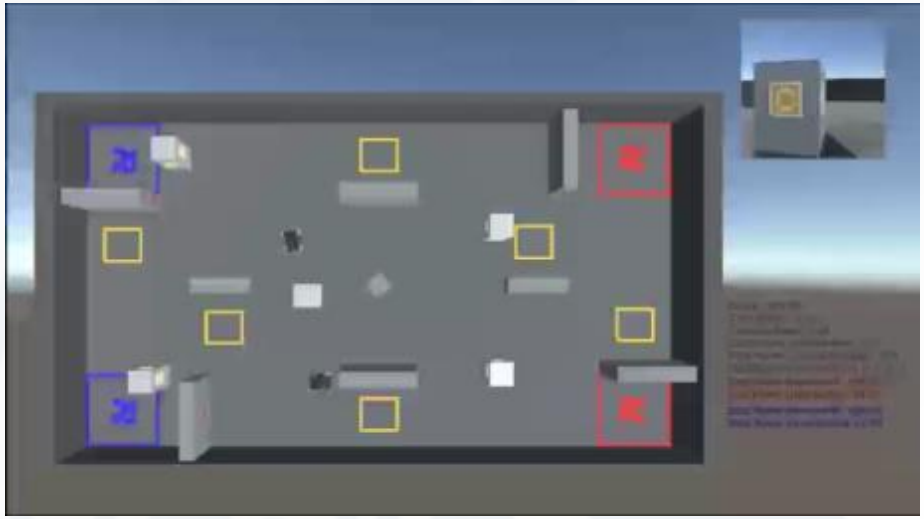
固定**朝向防守机器人**





训练训练结果（1.5倍速）

快速到达5个目标点：时间10s



迁移到比赛的导航

基线算法2

基于强化学习的导航

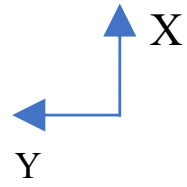
- **环境State:**
  - 本车的精确位置
  - 目标点的位置
  - **雷达信息**
- **Reward:** 是否碰撞,距离目标点的距离（在目标点附近速度很小，且停留0.5s）

- **输出:**

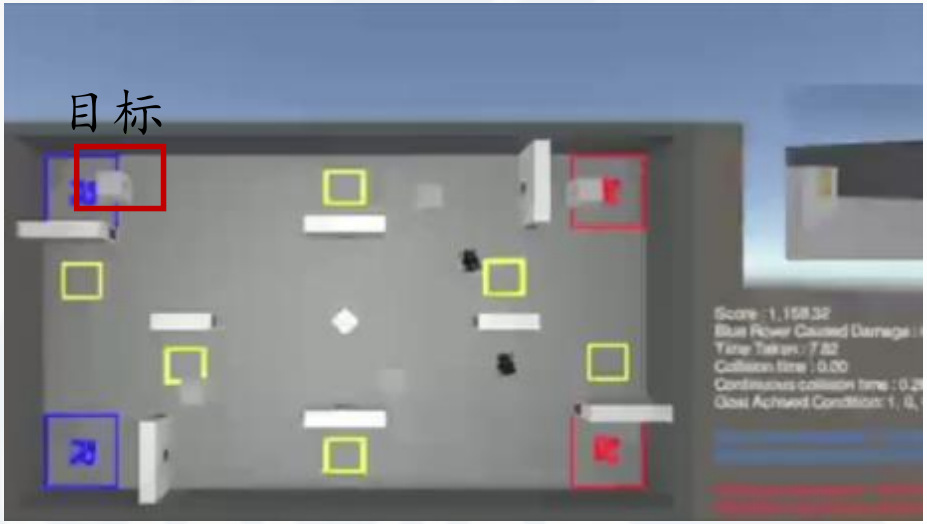
机器人X 方向的速度

机器人Y方向的速度

机器人的角速度



可以到达指定位置
增加了方向控制，动作更加灵活



红色机器人作为动态障碍物
蓝色机器人到达指定的某一目标

基线算法2

基于强化学习的导航：本车的位置存在偏差和噪声，纯位置的导航无法满足要求

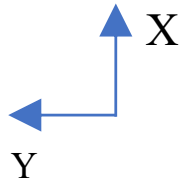
- **环境State:**
  - 本车的位置
    [-0.5m,0.5m]偏差
    [0.1m,0.1m]噪声
  - 目标点的位置
  - **雷达信息**
- **Reward**：是否碰撞,距离目标点的距离（在目标点附近速度很小，且停留0.5s）

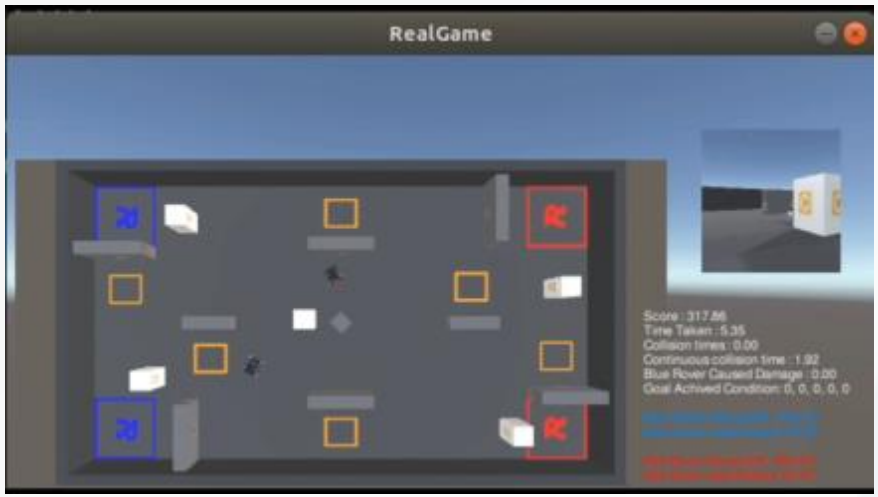
- **输出：**

机器人X 方向的速度

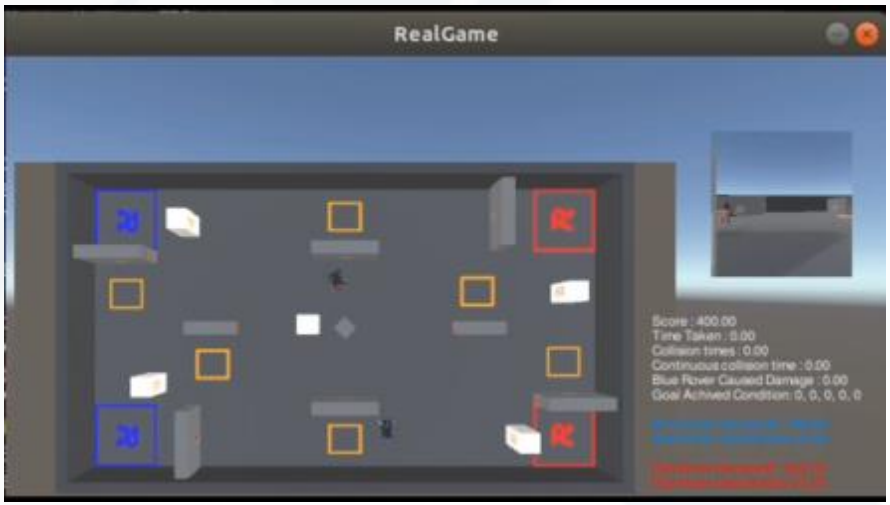
机器人Y方向的速度

机器人的角速度





纯位置输入：撞击障碍物



误以为到达目标点实际未到达

基线算法2

基于强化学习的导航

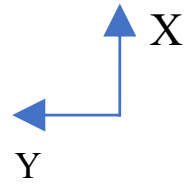
- **环境State:**
  - 本车的位置
    [-0.5m,0.5m]偏差
    [0.1m,0.1m]噪声
  - 目标点的位置
  - **雷达信息**
- **Reward**：是否碰撞,距离目标点的距离（在目标点附近速度很小，且停留0.5s）

- **输出：**

机器人X 方向的速度

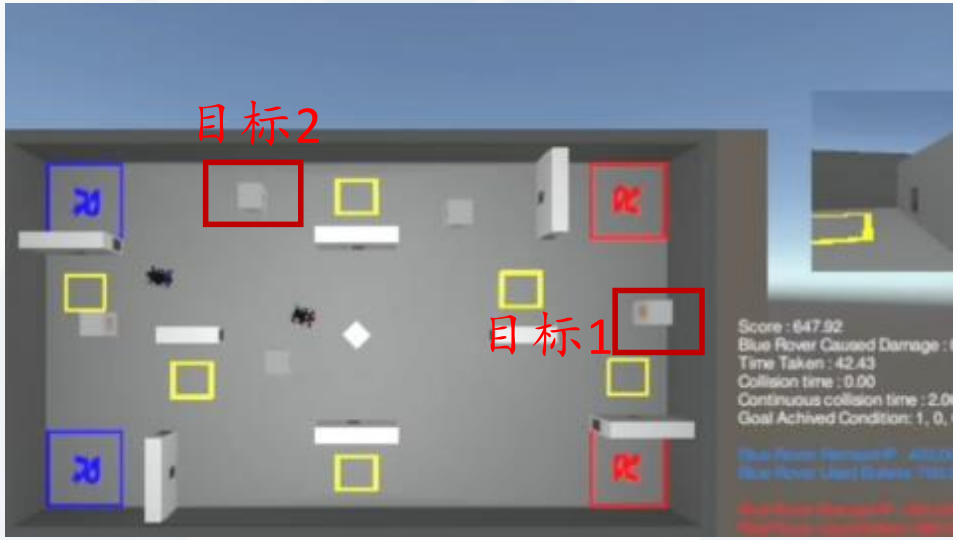
机器人Y方向的速度

机器人的角速度



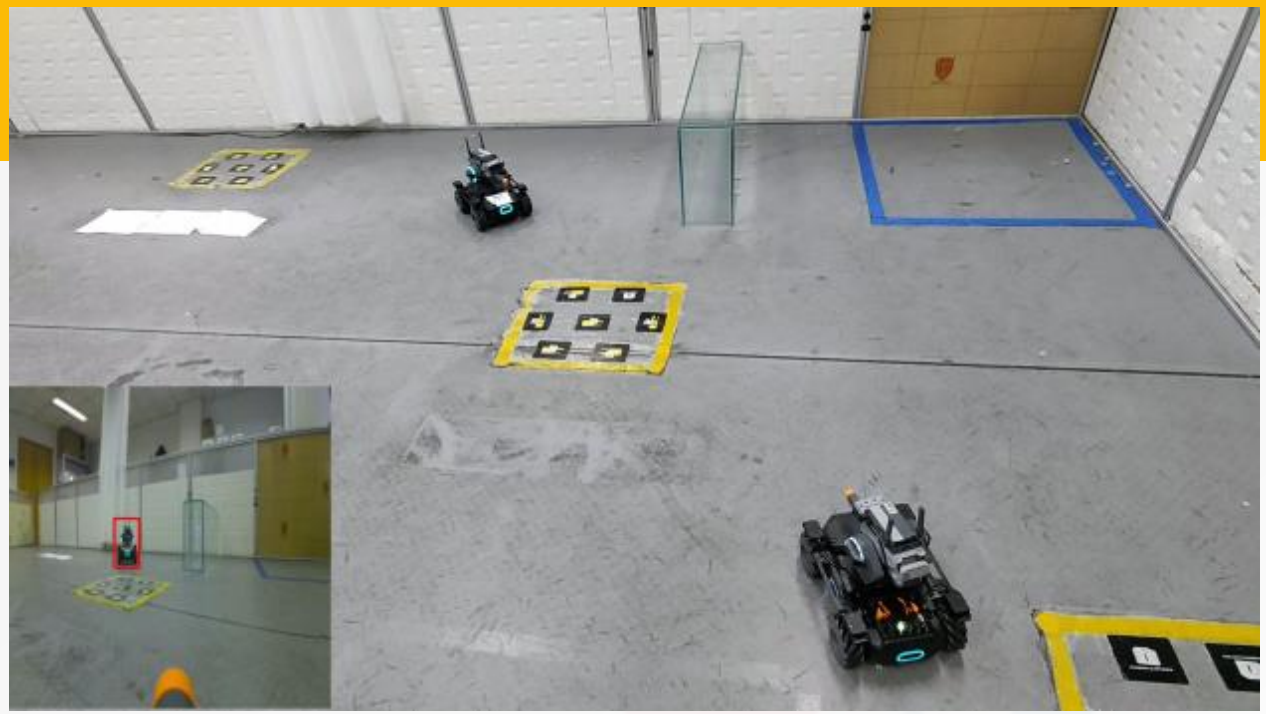
雷达作为输入，可以到达指定目标

针对观测量存在偏差的情况，初步验证强化学习可以解决这类问题。

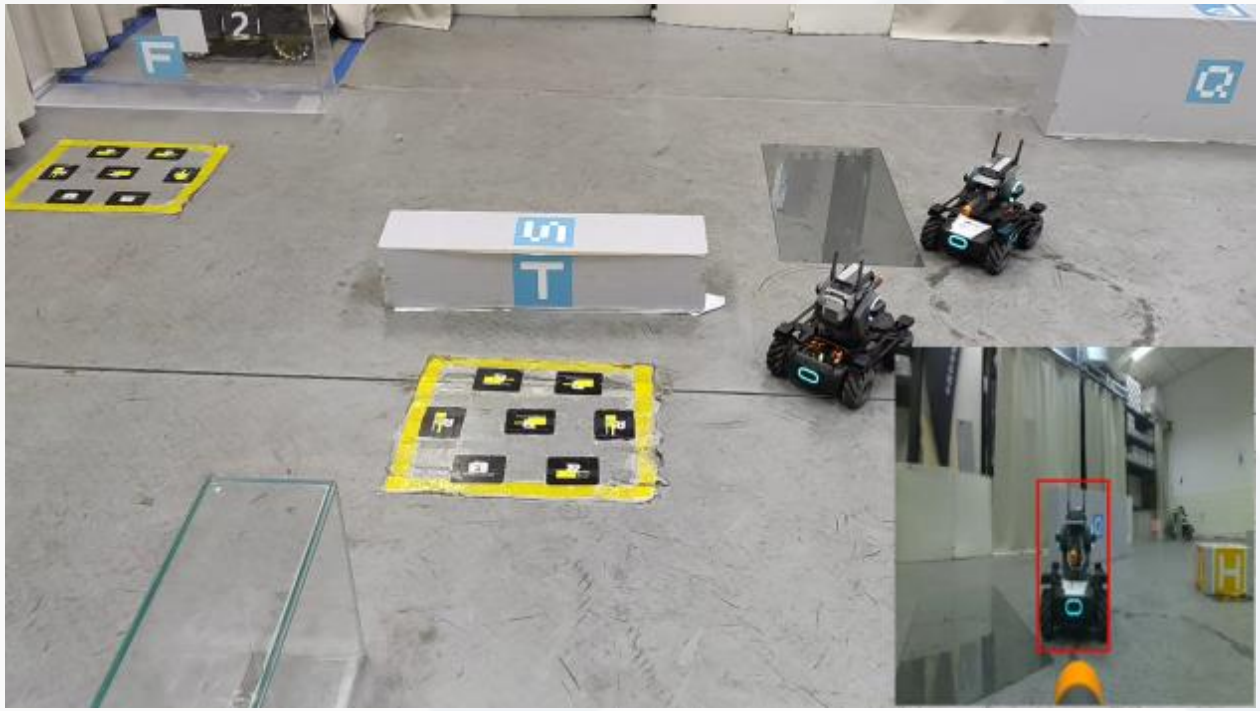


实体测试

机器人检测

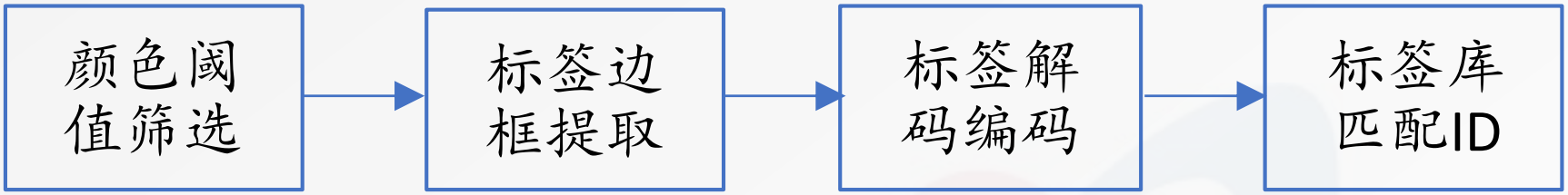


追踪



实体测试

AprilTag辅助定位



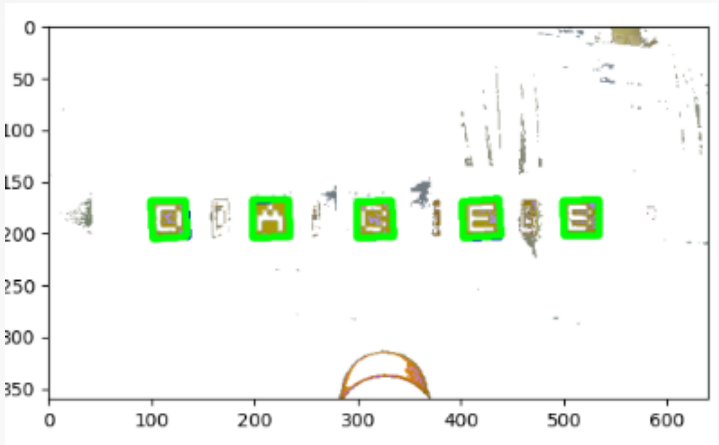
- 检测二维码
- 辅助定位

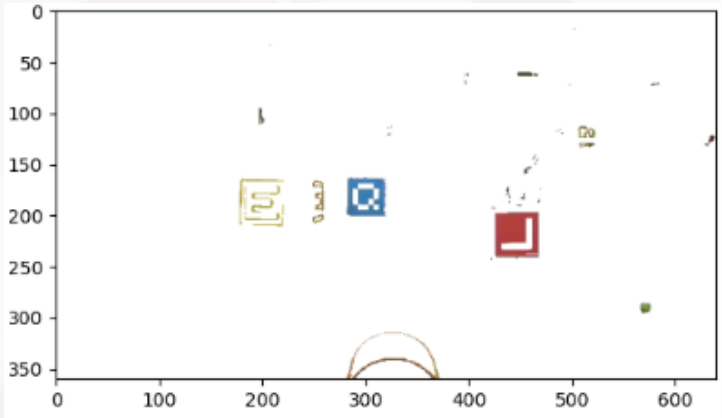
机器人打滑，或者撞击障碍物，定位出现偏差，利用AprilTag检测到的二维码辅助机器人定位。

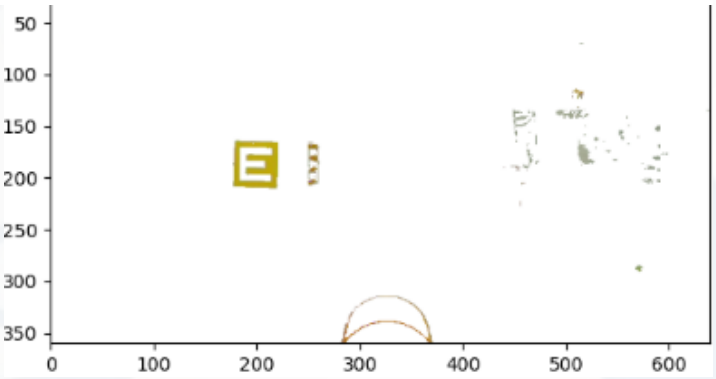
实体测试

AprilTag辅助定位

区分红蓝颜色、黄色标签识别

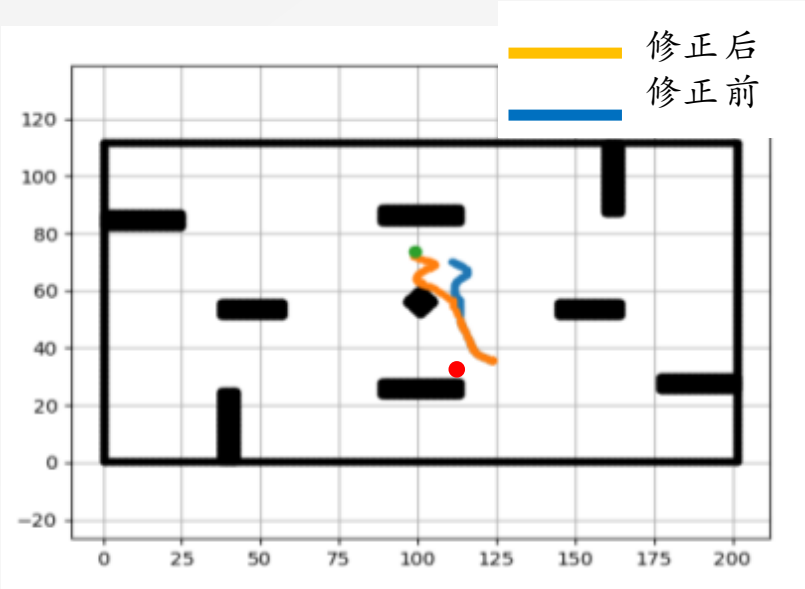






实体测试

AprilTag辅助定位



位置存在偏差，AprilTag辅助定位

里程计更新位置：已知初始位置(self.x_offset, self.y_offset)和里程计返回的信息 (x_info, y_info)得到最新的位置：

$$self.x = x\_info + self.x\_offset$$
$$self.y = y\_info + self.y\_offset \quad (1)$$

问题：使用里程计定位，遇到撞击打滑等导致路径反馈有误，规划为蓝色路径，无法到绿色目标点

视觉标签定位 (vision.x, vision.y)
用来更新初始位置（视觉坐标减里程计坐标的差值(gap_x_offset , gap_y_offset)加上初始坐标：

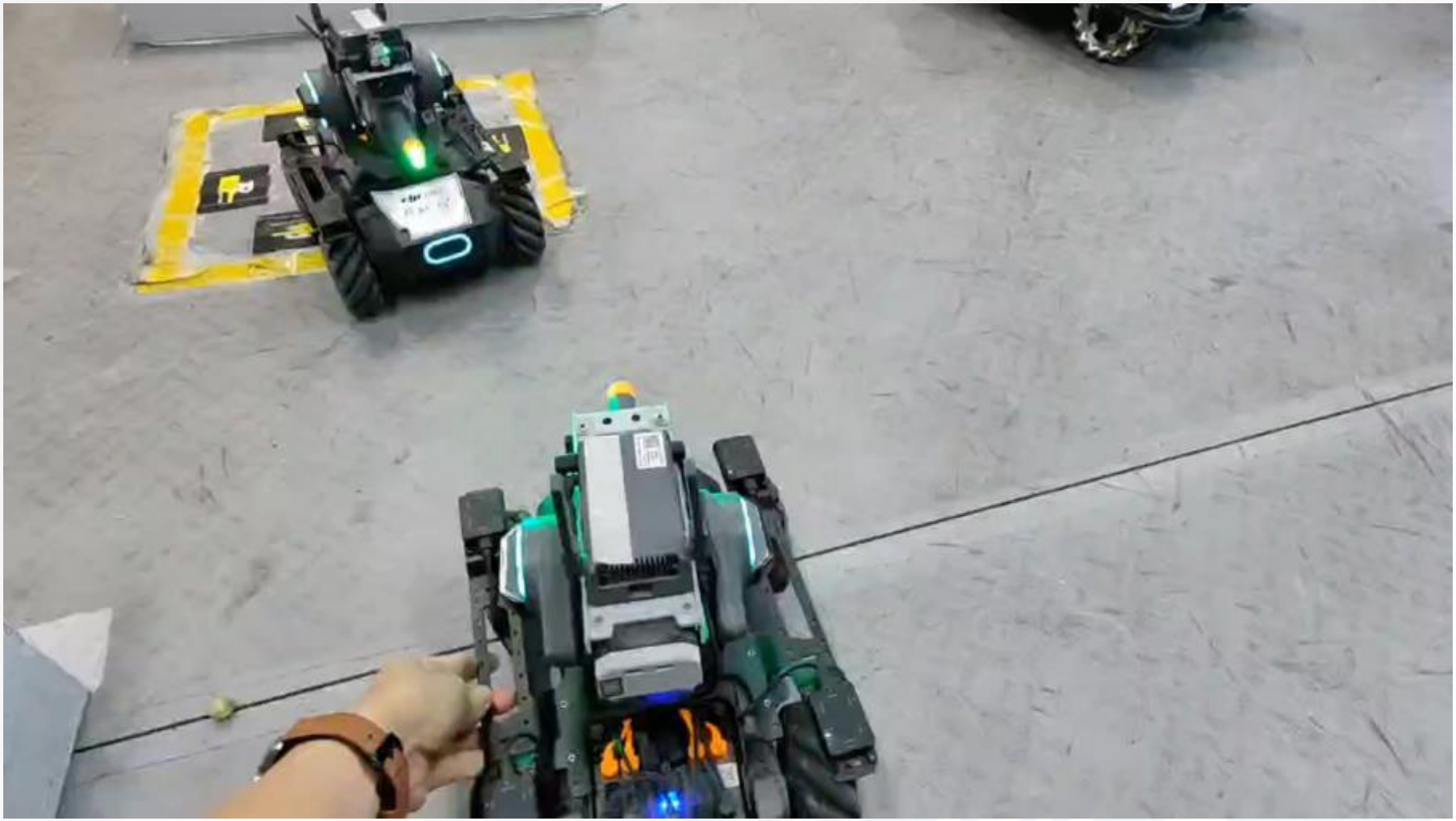
(self.x_offset ,self.y_offset) ）
$$gap\_x\_offset = self.x\_wheel - vision.x$$
$$gap\_y\_offset = self.y\_wheel - vision.y \quad (2)$$

视觉标签定位更新：Weight表示视觉定位的置信，和距离有关（1-2m时权重视觉定位为0.3，0-1m时权重为0.7），将（3）带入（1）更新定位

$$self.x\_offset = self.x\_offset - gap\_x\_offset * weight$$
$$self.y\_offset = self.y\_offset - gap\_y\_offset * weight \quad (3)$$

实体测试

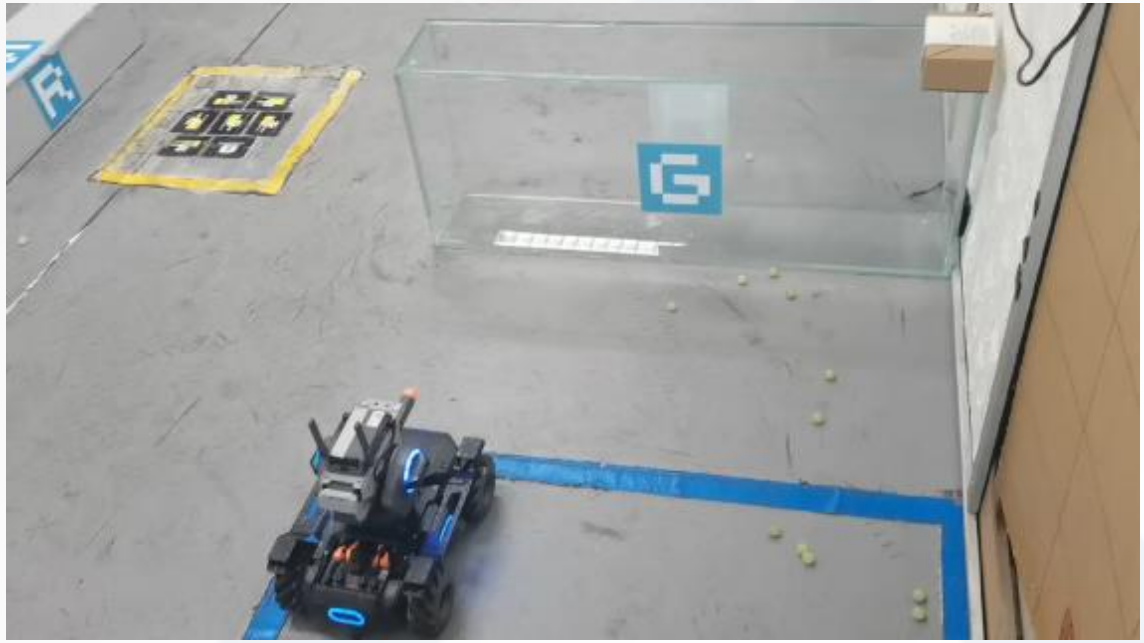
基于PID控制的云台追踪敌方装甲板



实体测试

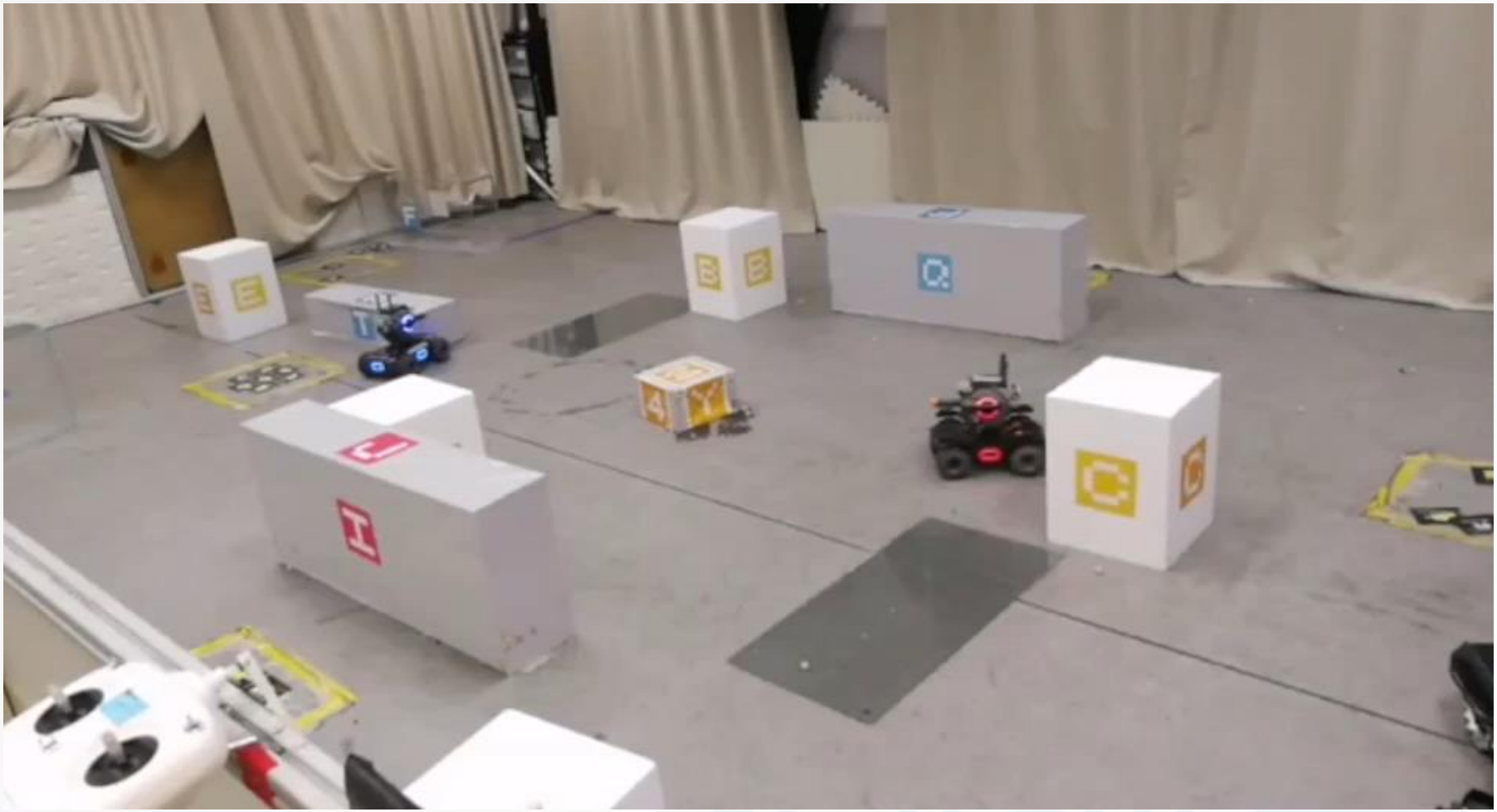
导航示例

由于位置偏差
导航失败示例



实体测试

对抗示例



参赛队算法分析

参赛队：据不完全统计，共有48支来自全国各高校的队伍报名参加了Sim2real比赛，截止第一阶段结束，共有13支队伍能够完成比赛，提交代码，经测试，筛选出10支队伍进入第二阶段

♠ Rank(Simulation Phase)

Participant team	Submitted time	Mean activated goals(N)	Mean score
Asterism	2022/6/18 6:52:45	5.0	659.2
D504	2022/6/14 19:03:48	5.0	610.5
HKU_Herkules2	2022/6/16 0:51:05	5.0	572.8
SEU-Abang	2022/6/18 1:49:50	4.8	78.2
SEU-AutoMan	2022/6/17 23:22:02	3.8	36.9
KnownAsSuperEast	2022/6/8 10:41:05	3.8	-8.0
THU_RLC_A	2022/6/17 13:12:40	4.3	-98.5
QGRFH	2022/6/9 15:14:49	4.7	-344.8
You are my god	2022/6/10 17:16:08	4.7	-685.1
HKU_Herkules1	2022/6/15 16:06:42	3.6	-3084.7
stay healthy for ddl	2022/6/8 0:11:51	1.8	-1284.9
King of Dog Point	2022/6/18 0:26:50	2.8	-3382.4
LuoXiangSaysAI	2022/6/8 10:08:00	2.6	-4354.3

参赛队员算法分析

Track 1 参赛队伍表现—导航阶段

参赛队员算法分析

Track 1 参赛队伍表现—对抗阶段

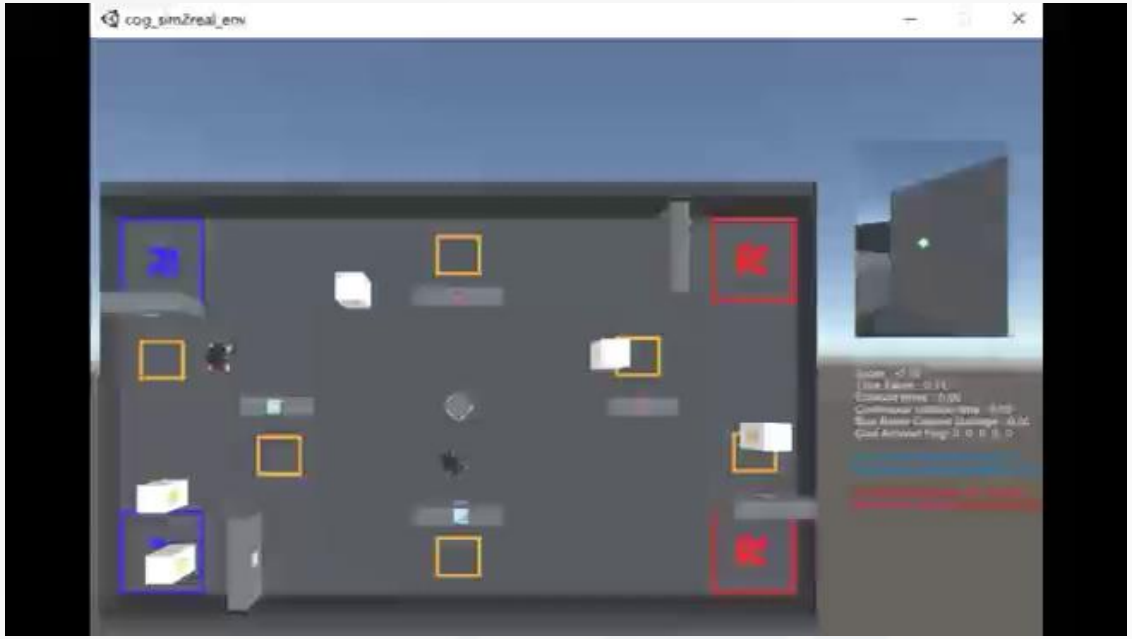
参赛队员算法分析

Tack 2 参赛队伍表现

参赛队员算法分析

第二阶段Sim2Real展示

仿真视频



实体视频

参赛队算法分析

赛道1算法总结—传统方法

	定位修正	规划器	对抗策略
HKU_Herkules2	简易粒子滤波算法在局部区域内修正位置	RRT+纯跟踪	进攻：选点导航 防守：4个防御点
THU_RLC_A	自己定义的规则	不是标准规划器，用规则写的	追击+摇摆进攻
HKU_Herkules1	局部区域内用lidar与占用地图的相似度做暴力搜索	A*+Stanley	摇摆进攻

参赛队算法分析

赛道1算法总结—学习方法

	动作空间	状态输入	网络结构	学习算法	对抗策略
Asterism	离散动作（9）	周围障碍物(俯视图->向量)，修正后位置，目标点 周围障碍物，修正后位置，敌方位置，进攻状态	MLP MLP	DQN	移动学习，射击规则
D504	连续动作	修正后位置，激光雷达，目标点 修正后位置，激光雷达，敌方位置，敌方速度，到敌方的距离和夹角，射击冷却时间	MLP MLP	SAC	移动学习，射击规则
SEU-Abang	连续动作	己方位置，激光雷达，目标点/敌方位置	MLP	AC	移动学习，射击规则
SEU-AutoMan	连续动作	己方位置，与目标点/敌方的相对位置	MLP	DDPG	移动学习，射击规则

参赛队算法分析

赛道1算法总结—学习方法

	动作空间	状态输入	网络结构	学习算法	对抗策略
KnownAs SuperEast	连续动作	己方位置，目标点及其距离和夹角	MLP	DPG	进攻：移动和射击学习 防守：左右摇摆，射击学习
QGRFH	连续动作 离散动作(2)	己方位置，目标点 图像，激光雷达，己方位置和敌方位置	MLP CNN+MLP	PPO	前后移动学习，左右移动，旋转和射击规则
You are my god	连续动作	激光雷达，与目标点的夹角和距离 激光雷达，与目标点的夹角和距离，双方剩余血量和子弹量	MLP MLP	None	移动和射击均学习

参赛队算法分析

赛道2算法总结—D504

动作空间	状态输入	网络结构	学习算法
连续动作	1 预测位置， 2 目标点， 3 图像编码特征	MLP	SAC

1 位置预测网络：CNN+RNN,预测本车位置

2 敌方位置
- 基于图像：敌方位置预测网络（CNN+MLP），输出当前图像有目标的置信度，与敌方的距离和角度
- 基于历史数据：假定敌方车辆线性运动，预测位置

3 图像编码网络：VAE（FCN）

作业要求

1、组队完成任务

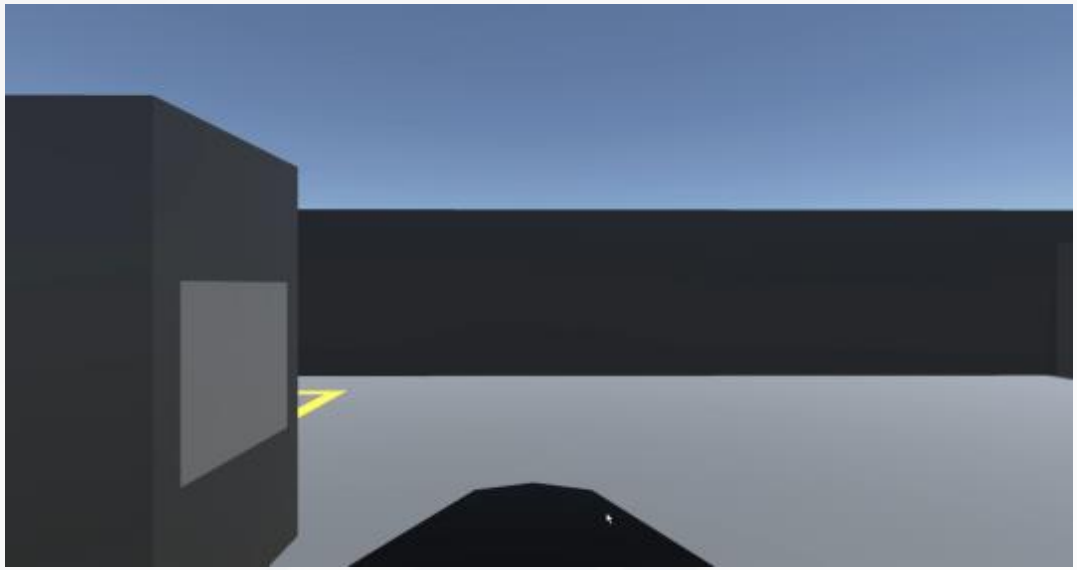
2、完成导航到固定点的任务

（状态输入/图像输入）

3、提交代码（可执行的脚本）、

视频和报告分析

4、可用华为云资源



要求：

- 基于MindSpore架构下搭建代码会有加分
- 在大报告里体现每个人分工完成的部分
- 必须涉及到强化学习/深度强化学习方法
- 大报告可以以会议论文，或者以PPT汇报的形式
- 建议进行不同算法、不同状态的对比分析