

# 2025 年 北京建筑大学 萌新种子杯

## “视觉循迹仿真”赛道竞赛命题与运行

### 一、 比赛背景与意义

计算机视觉是机器人感知外界环境的核心能力，使机器人能够像人类一样“看见”世界并进行理解、判断和决策。在移动机器人中，视觉循迹技术被广泛应用于智能物流车辆、智能小车、自动驾驶等领域，是机器人自主运动控制的重要基础。本次竞赛旨在让参赛者在仿真环境中深入理解视觉信息的获取、处理与控制执行之间的关系，提升编程能力、图像处理能力以及机器人运动控制算法设计能力，为后续机器人科研与竞赛打下坚实基础。

### 二、 比赛环境

#### 1. 软件环境

Webots R2021a。下载地址：

[https://github.com/cyberbotics/webots/releases/download/R2021a/webots-R2021a\\_setup.exe](https://github.com/cyberbotics/webots/releases/download/R2021a/webots-R2021a_setup.exe)

#### 2. 仿真环境

组委会将统一提供完整赛道场景与机器人模型。

下载地址：<https://github.com/imfanyu/bucea-seed-cup-2025/archive/refs/heads/main.zip>

**注意：**文件托管在 GitHub，下载速度可能较慢，建议加入 QQ 群（群号 455362758）获取加速下载。

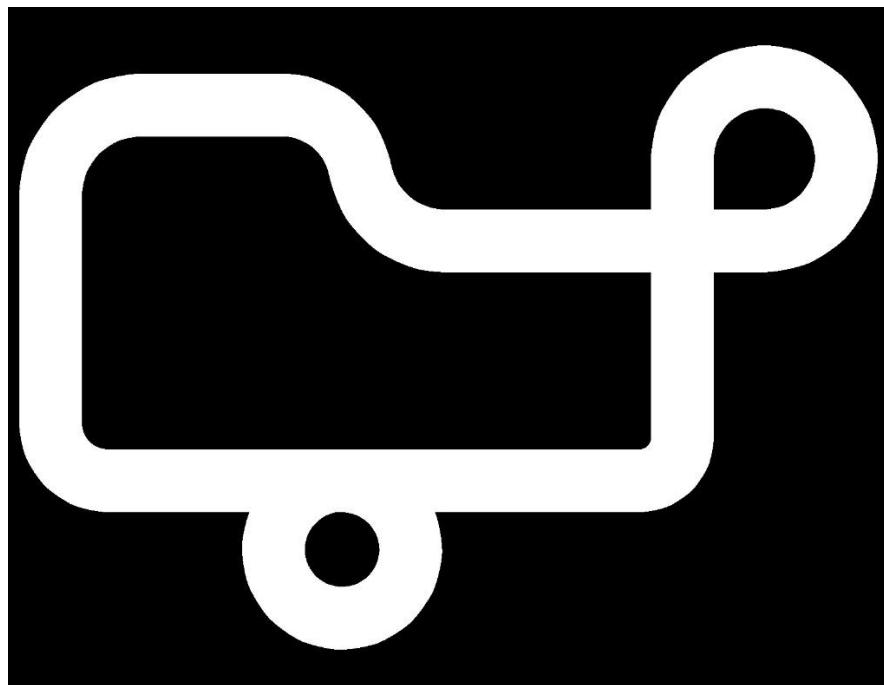
### 三、 任务要求

1. 个人赛：参赛选手需实现小车在赛道上的**自主循迹行驶**，不得人工干预。
2. 参赛选手**不得修改**仿真场景的物理参数，且**不得更改**机器人结构（包括但不限于摄像头位置、朝向及安装角度等）。
3. 控制与视觉算法 必须使用 Python 完成。
4. 图像处理部分如涉及**直接调用**现成分析算法函数（包括阈值处理、平滑处理、颜色空间转换、边缘检测、轮廓检测、形态学操作等），总成绩记为原分数的 50%。（允许使用 cv2.resize 和 cv2.imshow，不扣分）

- 问题求解策略不作限制，参赛者可自行设计控制逻辑、算法模型及状态策略。

#### 四、 仿真环境说明

##### 1. 场景地图



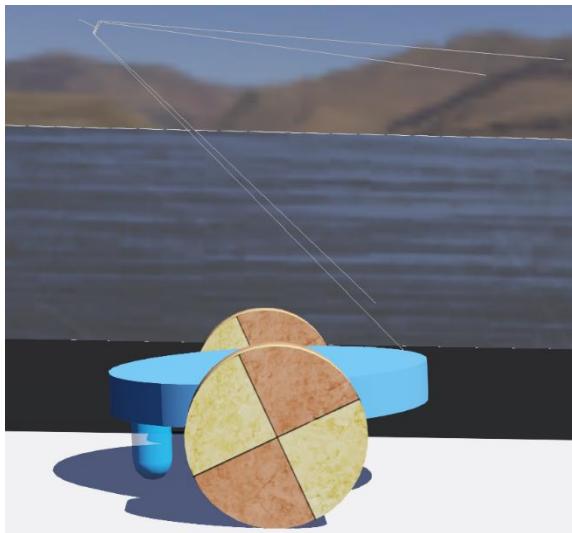
##### 2. 赛道元素

赛道为 白色路径区域，周围环境为 黑色背景；

赛道中包含以下道路结构元素：

- 直行道路
- S 弯道
- 十字路口
- 直角转弯
- 圆环环道

### 3. 机器人模型结构



机器人采用 双轮差速驱动

前置单摄像头用于道路视觉识别

## 五、 控制接口说明

比赛将提供两类基本控制接口函数：

功能	描述
摄像头图像读取	获取当前帧图像数据
轮子速度控制	控制左右轮转速实现小车方向与速度调整

参赛选手需基于摄像头图像，自主设计视觉识别与偏差控制策略，使机器人实现稳定循迹

## 六、 评分规则

项目 / 赛道元素	得分	判定条件
直行道路	1 分	稳定完成直线段
直角转弯	2 分	成功通过任意一个直角转弯
S 弯道	2 分	平稳通过 S 形弯道
十字路口	3 分	做出正确路径跟随并顺利通过
圆环环道	5 分	成功识别并完成圆环绕行
完整跑完全程	额外 +3 分	连续完成所有赛道元素后到达终点

注意： 1. 各元素分值 不重复累计，即每类元素只计一次最高分。

2. 图像处理部分如涉及直接调用现成分析算法函数(包括阈值处理、平滑处理、颜色空间转换、边缘检测、轮廓检测、形态学操作等)，总成绩记为原分数的 50%。

**零分判定：** 若比赛过程中出现机器人双轮同时完全脱离赛道区域（即车辆全体驶出白色轨迹），则本次成绩记为 0 分。

## 七、 提交方式

请将仿真运行过程录制为完整视频，并整理好最终使用的 controller 的 Python 代码文件。

将上述两个文件放入同一文件夹中，再将该文件夹压缩为 .zip 格式。

压缩包命名格式如下：

**视觉循迹仿真-班级姓名学号.zip**

例如：

**视觉循迹仿真-电气 241 张三 2024040201xx.zip**

完成后，请在**比赛网站的对应提交入口**上传该压缩包。

## 八、 补充说明

- 比赛将安排赛前说明会与算法讲解；
- 本命题及解释权归“萌新种子杯”竞赛组委会所有。

## 九、附录：控制接口与示例代码

```
1 """Demo controller."""
2 from controller import Robot, Motor, Camera
3
4 # -----
5 # 创建机器人实例
6 # -----
7 robot = Robot()
8 timestep = int(robot.getBasicTimeStep())
9
10 # -----
11 # 获取左右电机
12 # -----
13 left_motor = robot.getDevice('MotorLeft')
14 right_motor = robot.getDevice('MotorRight')
15 left_motor.setPosition(float('inf'))
16 right_motor.setPosition(float('inf'))
17 left_motor.setVelocity(0.0)
18 right_motor.setVelocity(0.0)
19
20 # -----
21 # 启用摄像头
22 # -----
23 camera = robot.getDevice('camera')
24 camera.enable(timestep)
25
26 # -----
27 # 主循环
28 # -----
29 max_speed = min(left_motor.getMaxVelocity(), right_motor.getMaxVelocity())
30
31 while robot.step(timestep) != -1:
32     # 前进
33     left_motor.setVelocity(0.5*max_speed)
34     right_motor.setVelocity(-0.5*max_speed)
35
36     # 获取图像缓冲区
37     image = camera.getImage()
```