

2020RoboCup 机器人世界杯中国赛比赛规则

RoboCup 救援仿真组

2020RoboCup 机器人世界杯中国赛
RoboCup 救援仿真组技术委员会

2020 年 1 月 5 日

目 录

一、项目简介	3
二、技术委员会	3
三、赛项说明	4
四、比赛场地及器材	5
五、机器人要求	6
六、评分标准	7
七、赛程赛制	7

一、项目简介

RoboCup 机器人救援仿真是是一个通过模拟现实生活中的城市灾难场景，用机器人进行救援的仿真系统，它是和应用领域结合十分密切的新兴工程。其主要目的是使救援智能体进行有效的分工协作，完成营救市民和灭火的任务，以最大限度地减小灾难带来的损失，进而在灾难救援这个重大的社会问题上促进研究和发展。

RoboCup 机器人救援仿真是一个用计算机对真实的城市灾难情况进行模拟的系统，如在地震发生时的仿真模拟环境中：房屋，建筑物等都倒塌了；道路、轨道和其他一些公共交通设施都被毁坏了；基础的城市设施比如电力，下水道系统也都被毁坏了；通信设施和信息的传播被中断了，许多受害者被埋在倒塌的房屋下；地震引起的火灾开始很快的蔓延；消防车要通过的道路被倒塌的房屋碎片挡住了等场景。为了减小灾难带来的损失，参赛队伍需要开发一支强有力的救援智能体队伍，在仿真系统提供的灾难场景下进行有效的救援，并且尽快地营救受伤的民众，抢救人们的生命财产，把灾难的损失降低到最小。

二、技术委员会

负责人：彭军，中南大学，pengj@csu.edu.cn，13875979898

成 员：蒋富，中南大学

许映秋，东南大学

赵清杰，北京理工大学

三、赛项说明

1. 编译

每个队伍均需要提交可运行的源码（非二进制码），与一个可用于编译该源码的脚本。这个脚本的名字必须为 `compil.sh`，并且放在所提交代码的根目录中。

2. 比赛要求

- 1) 参赛队伍需要按照 `ADF-template1.0` 规定的格式书写启动脚本。
- 2) 智能体不能使用任何形式的共享内存，包括所有智能体可访问的静态内存、智能体之间的直接函数调用或编写在方案模拟过程中由其他智能体使用的文件。
- 3) 预计算阶段允许每种类型的智能体与预处理特定于特定场景的数据和具体的方案有关，并将它们存储在模拟阶段中使用的文件中。每个类型的智能体只有一个可以连接到服务器并执行算算法。此阶段限制为 2 分钟，经过时间后，服务器将被终止。
- 4) 正式比赛阶段对应于实际队伍在竞争场景中的模拟。为了执行场景模拟，队伍必须在 3 分钟内将所有智能体连接到内核。在第一个智能体开始与内核握手之后，场景模拟将不迟于 3 分钟开始。

3. 代码提交时间与次数

提交代码的时间、次数以及要求将会在比赛准备时间做进一步介绍。技术委员会将有权在比赛期间调整代码提交时间。同时，技术委员会将有权预览所有已提交的源码。

4. 队伍说明及技术交流

队伍准备一份 PPT，用于介绍队伍及相关技术。每个队伍将有 20 分钟的时间来展示，并有 10 分钟的问答时间。该报告将由专家小组和其他小组的组长进行评估。成绩将算作一个地图的成绩，并计入最终比赛成绩。拟进行技术挑战赛。

四、比赛场地及器材

1. 软件环境

- a) 仿真环境：Linux Ubuntu 16.04 (64bit) 或更高版本，使用 Open JDK 8 运行时或更高版本。
- b) 仿真服务器：比赛采用最新稳定的服务器版本，服务器下载地址：<https://sourceforge.net/projects/roborescue/files/roborescue/>。

2. 硬件环境

- a) 比赛所用的 PC：每支队伍在比赛时使用 4 台 PC，其中一台用于运行仿真服务器，另外 3 台运行参赛队伍。
- b) 运行服务器的机器硬件配置：i7， 32GB RAM， 独立显卡。
- c) 运行参赛队伍的机器硬件配置：i7， 16GB RAM。

3. 地图：比赛地图在比赛时由技术委员会决定。

- a) 可能有一个或更多的随机地图用于比赛，将由技术委员会在比赛时决定。
- b) 地图将被限定在 10000 个建筑物和 10000 个道路以内。
- c) 道路是正确连接，建筑物有相应的入口并且所有道路和建筑物

都是可以访问的，然而，当出现道路或建筑物入口没有完全连通并被证实是验证工具出错时，各参赛队伍无权投诉。

- d) 仿真参数与难度等级将会随着地图的比例不同而调整，以匹配真实世界。

五、机器人要求

队伍应使用 ADK 框架来实现所有类型的智能体。队伍只能实现自己的代码来替换或扩展以下类：

```
adf.component.extraction.ExtAction  
adf.component.module.algorithm.Clustering  
adf.component.module.algorithm.PathPlanning  
adf.component.module.complex.TargetDetector  
adf.component.module.complex.TargetAllocator  
adf.component.centralized.CommandPicker  
adf.component.centralized.CommandExecutor
```

不允许队伍更改其他类，尤其是基础类。队伍的职责是确保其代码将智能体连接到服务器。

队伍必须在以其团队名称命名的包中实现其代码。

```
$TEAM. ExtAction  
$TEAM. algorithm. Clustering  
$TEAM. algorithm. PathPlanning  
$TEAM. TargetDetector  
$TEAM. TargetAllocator  
$TEAM. centralized. CommandPicker  
$TEAM. centralized. CommandExecutor
```

例如，如果队伍的简称是测试的，则队伍应该提供包含方法的包。

```
TEST.extraction.ExtAction  
TEST.algorithm.Clustering  
TEST.algorithm.PathPlanning  
TEST.complex.TargetDetector  
TEST.complex.TargetAllocator  
TEST.centralized.CommandPicker  
TEST.centralized.CommandExecutor
```

队伍应提供一个配置文件，其中包含它们已从原始 ADK 框架中更改的类的信息，以及在其代码中的类、包路径和文件之间的映射。

所有参赛的队伍必须保证所使用的代码与提交的可执行码一致。队伍对代码的贡献率至少在 30% 以上，技术委员会有权采取措施鉴别有疑问的队伍，一旦确认队伍违反了公平竞赛原则，将立即取消其参赛资格。

各个队伍需要确保其代码能很好与各给定的仿真器一起工作。尽管技术委员会尽一切努力以提供一个可靠的仿真环境，但对在比赛期间出现的任何软件故障，技术委员会不负任何责任。仿真错误不能作为充分的理由来要求重新运行。

六、评分标准

比赛将由若干轮组成，在每一轮中，各参赛队伍都会有一个地图分数，将地图分数按从高到低排序，设 n 为参加这轮比赛的队伍数目，则地图分数最高的队伍得 n 分，地图分数最低的队伍得 1 分。每轮比赛的队伍最终成绩为每个地图的得分之和。

每轮各个队伍根据这轮的最终成绩进行排名，分数最高的为第一名，第二高的为第二名，并依此类推。如果最终成绩相同，则按总的地图分数进行排名。

七、赛程赛制

1. 队伍分组

比赛分为预赛和决赛两个阶段，参赛队伍比赛在预赛中分为三个小组同时进行。

A1	B1	C1
A2	B2	C2
A3	B3	C3

在赛前抽签决定具体分组情况。

每个小组中，对于每个地图的比赛，队伍上场的顺序循环调整，例如在第一小组中，第一个地图比赛的顺序为 A1-A2-A3，则第二个地图比赛的顺序为 A2-A3-A1，依次循环。

所有参加比赛的队伍必须在比赛当天上午 9:00 前将各自队伍的可执行码上传到比赛用服务器上。进入决赛的队伍将要求进行现场编译源代码。

2. 晋级方法

比赛采用积分制，按规则中的计算方式算分。队伍的最终得分为所有得分之和。如果有队伍积分相同，则以每场地图分数的总分高的队伍胜出。

3. 赛程安排

预赛和决赛中分别使用 8-10 个地图；

（比赛所用地图的数目和比赛时间会根据比赛的进程做出临时调整,请大家注意赛场通知）