ITHACA：用于在 Unity 中集成模糊逻辑的工具

1st Alfonso Tejedor Moreno阿尔梅里亚大学计算机科学系西班牙阿尔梅里亚 04120 alfitejedor@gmail.com

2nd Jose A. Piedra-Fernandez阿尔梅里亚大学计算机科学系西班牙阿尔梅里亚 04120 jpiedra@ual.es

3rd Juan Jesus Ojeda-Castelo阿尔梅里亚大学计算机科学系西班牙阿尔梅里亚 04120 juanje.ojeda@ual.es

4th Luis Iribarne*IEEE*会员 阿尔梅里亚大学计算机科学系西班牙阿尔梅里亚 04120 luis.iribarne@ual.es

arXiv:2301.00377v1 [cs.AI] 1 Jan 2023

摘要——**Ithaca 是一款用于在 Unity 游戏引擎中开发人工智能系统的模糊逻辑（FL）插件。其目标是提供一种直观、自然的方式来构建先进的人工智能系统，使这类系统的实现更快捷、更经济。该软件由一个 C# 框架和一个用于编写推理系统的应用程序编程接口（API）组成，并提供一套用于图形化开发和调试的工具。此外，还提供了一个模糊控制语言（FCL）解析器，以便导入先前使用该标准定义的系统。**

索引词—**人工智能、模糊逻辑、模糊控制语言、电子游戏。**

I. 引言

GRAPHICS是电子游戏领域中的一个重要因素，这一点在最新一代电子游戏和游戏主机的开发中得到了体现。每一代新游戏主机都会增强其资源，例如配备强大计算能力的显卡、拥有多个核心的处理器等。因此，电子游戏能够呈现非常逼真的图形，用户也会感觉自己置身于真实环境之中，而不仅仅是虚拟世界。然而，用户的要求越来越高，如今人工智能（AI）在这一领域发挥着重要作用，因为如果一款电子游戏的图形效果十分出色，但其人工智能水平很低，那么它实际上并不会真正受到欢迎。事实上，人工智能的质量是提升玩家沉浸感和娱乐体验的重要功能 [1]。

自国际象棋 [2] 或围棋[3]等游戏被人工智能攻克以来，人工智能一直是电子游戏领域的一项杰出贡献。甚至 DeepMind 开发的算法也能够与人类玩家或其他机器玩家合作 [4]。这些智能体已在《雷神之锤III》游戏中接受测试。据 Pirovano 所述 [5], ，游戏中最常见的两种人工智能技术是有限状态机（FSM）和决策树（DT），其性能远不及人工神经网络（ANN）和遗传算法（GA）等更复杂的技术。因此，我们的主要目标是让规模较小和中等的团队也能使用人工智能，为他们提供一种介于两者之间的技术实现工具：它兼具有限状态机易于使用的特点，以及人工神经网络的功率和灵活性。并且直接基于使用最广泛的游戏引擎 Unity [1]（Unity 概况）。游戏中的人工智能不仅用于非玩家角色，也不只是让玩家在游戏中做出更多决策；人工智能还被用于重制电子游戏，并在数小时内提升这些产品的质量。

另一方面，大多数游戏开发者都面临着巨大的时间和预算限制，因此独立开发者在游戏中实现的人工智能充其量只能算是初级水平，甚至可能根本没有人工智能。这正是我们开发一款基于模糊逻辑的插件的原因，旨在帮助开发者等人以简单的方式在项目中应用模糊逻辑，并充分发挥其作用。此外，该插件是为 Unity 开发的，因为从学生或业余爱好者到专业人士和游戏工作室，许多用户都在使用这款游戏引擎。Unity 允许开发者在多个平台上创建电子游戏，更确切地说，是在30个不同的平台上创建，例如移动设备、个人电脑或游戏主机。因此，对于想学习如何开发游戏的人、只想将制作电子游戏作为业余爱好的人，甚至对于希望建立自己的娱乐公司或制作专业游戏的企业家而言，这款工具都非常有用。此外，该插件采用了模糊逻辑的基本功能，使用起来非常简单，因为它面向的是没有相关知识基础的初学者。而这一点之所以能够实现，是因为 Unity 能够适应许多需求，其学习过程在初期非常容易，尽管如果想制作复杂的游戏，学习难度会逐渐增加。这些功能使得相比其他游戏引擎（如虚幻引擎），更多人使用这款游戏引擎 [2] 事实上，John Riccitiello（Unity首席执行官）表示，所有游戏中有一半是在 Unity 上开发的 [3]。

本项目的目标是：

[1]Unity - https://unity.com/[2]虚幻引擎 - https://www.unrealengine.com/en-US/[3]使用 Unity 开发的游戏 - https://techcrunch.com/2018/09/05/unity-ceo-says-half-of-all-games-are-built-on-unity/

- 创建一个易于使用的工具，让所有游戏开发者都能使用人工智能和模糊逻辑。
- 该工具必须简单灵活，才能用于各种项目和游戏。
- 用户必须能够通过图形用户界面（GUI）或 API 使用该工具。
- 符合 IEC 61131-7 [6] 中定义的 FCL 标准

本文其余部分安排如下：第二节介绍模糊逻辑和人工智能的背景知识。第三节介绍该架构及其模糊逻辑基础。第四节展示构建若干小游戏所取得的成果。第五节总结结论并探讨未来工作。

II. 背景

1940年，Edward Uhler Condon 在西屋电气的展览会上展示了一款数学游戏。这款人机对战游戏吸引了数千人参与，其中只有10%的人能够击败机器。然而，直到20世纪70年代，电子游戏才开始大规模商业化，例如雅达利公司开发的游戏：乒乓[7]。当时，电子游戏的人工智能系统中还未包含神经网络，不过使用了有限状态机等更简单的技术。下一节将简要介绍人工智能对电子游戏世界的影响。

A. 游戏中的人工智能

1959年，Arthur Lee Samuel 将人工智能引入电子游戏领域，更具体地说，他教会了计算机如何玩跳棋 [8]。此后，人们开发了其他融入人工智能的游戏，其中包括70年代的电子游戏太空侵略者和80年代的吃豆人。在太空侵略者中，通过引入能够从玩家行为中“学习”的哈希函数，增加了游戏的复杂性。吃豆人游戏的核心在于幽灵的行为，它们是玩家的敌人，每个幽灵都有不同的行为 [9] ，并由状态机控制 [10]。20世纪90年代，随着各种游戏主机的出现，电子游戏蓬勃发展，例如 Game Boy、超级任天堂和 PlayStation One。这一扩张也促成了更复杂的人工智能系统的引入，其中尤其值得一提的是半条命：在这款电子游戏中，敌人之间能够相互合作来消灭玩家 [11]。进入21世纪后，人工智能取得了长足进步，其算法已经能够在星际争霸II等游戏中击败人类玩家 [12], [13]。

所有这些发展都有一个共同点，无论它们集成了何种人工智能，都是为了改善用户体验，而人工智能正是实现这一目标的绝佳选择。人工智能可用于电子游戏的多个方面，例如寻路、移动、决策或学习 [10], ，但最突出的应用可能是控制非玩家角色的行为。原因在于，如果这些元素的行为真实且贴近现实世界，用户就会觉得自己并非置身于一个人工构建的、预先设定的世界中，而是身处一个更接近自身现实的世界。谈到这一点，重要的是要记住，人工智能的目标并非总是击败玩家，因为这会让游戏失去乐趣和吸引力；它的目标应当是创造尽可能真实的体验。

甚至还有研究分析了将深度强化学习应用于严肃游戏的可行性 [14]

除了深度强化学习，遗传算法也推动了这一领域的发展。在 [15] 中，遗传算法可以通过低成本的流程生成格斗游戏中的人工智能角色。此外，遗传算法还被应用于足球模拟器（Robocup Soccer），以改进足球队模拟中的决策能力 [16]。这一改进的实施结果是，组成该系统的三支队伍中有两支成为 RoboCup 竞赛中的顶尖队伍。[17] 旨在将遗传算法应用于即时战略游戏，以学习适用于这类游戏的新策略和有效策略。问题在于，这类行动对于玩家而言通常较为抽象，因此这项研究着重于学习人们容易理解的策略。此外，这类算法也被用于西洋双陆棋 [18] 和国际象棋 [19] 等游戏中。

前文展示了一系列应用，这些应用与早期电子游戏中使用的人工智能技术相去甚远。早期电子游戏由基于规则的系统定义，而这类系统构成了人工智能中最基础的算法。如今，深度学习或遗传算法等更复杂、更具创新性的结构已得到应用。不过，还有一种名为模糊逻辑的人工智能技术。尽管它不像神经网络那样广为人知，但在这类开发中非常实用，下面将对此进行介绍。

B. 模糊逻辑

模糊逻辑适用于解决许多不确定性起主要作用的问题，例如不确定多准则决策问题 [20]。它广泛应用于难以建立数学模型的控制器，或需要处理模糊性、歧义性的系统。模糊逻辑的一些应用包括医疗应用 [21]、洗衣机等日常生活应用 [22]、基于人脸识别的自然交互 [23] 和姿态识别 [24] ，以及智慧城市。最后这一项或许令人意外，不过，城市的发展已经引发了各种问题。垃圾收集、环境污染、交通、能源分配或供水等方面的管理，正变得日益复杂。模糊逻辑有助于提高智慧城市的能源效率 [25]。

1）游戏中的模糊逻辑：与前文提到的其他人工智能技术一样，模糊逻辑也为电子游戏作出了贡献。针对有特殊需求的人群，已有专为自闭症患者开发的游戏。在 [26] 中，模糊逻辑用于评估自闭症儿童的社交技能，并据此调整游戏难度。在诊断患者的自闭症程度时，患者需要完成一些活动，以确定其程度

。本文作者 [27] 设计了一款能够运用模糊逻辑计算玩家自闭症程度的游戏。不过，严肃游戏不仅面向自闭症患者，也有专为身体障碍人士设计的游戏，例如名为 ReHabGame [28] 的游戏。该游戏通过一系列针对上肢和下肢的活动，评估并改善神经肌肉问题患者的运动和感觉能力。玩家可通过 Kinect 与这款严肃游戏互动，系统由此获取玩家在执行任务过程中的身体动作信息，模糊逻辑系统便能提取并分析与玩家姿势相关的数据，以满足他们的需求。除了 ReHabGame，还开发了其他主题各异的严肃游戏。例如，在另一款严肃游戏中，作者研究了节气门位置传感器、发动机转速和车速等变量，以运用模糊逻辑判断驾驶风格是否高效，从而改善驾驶员的驾驶行为 [29]。作者在此基础上，将模糊逻辑与随机森林相结合，以确定哪种算法更适用于此类严肃游戏。结果表明，模糊逻辑的优势在于能够提供易于理解的语言反馈，而随机森林则能更准确地预测燃油消耗 [30]。另一项值得一提的创新研究发表于 [31] ，旨在创建一款通过归纳控制激发学生情绪、改善其学习过程的游戏。该系统结合模糊逻辑，通过语音分析来评估玩家的表现和情绪状态。

C. *Unity* 的人工智能插件

最近，Unity 引擎终于开始拥抱人工智能，为其寻路工具加入了机器学习（ML）。这款游戏引擎最出色的开源插件名为 ML-Agents。借助该插件，游戏和模拟器可以成为训练环境，通过强化学习来训练人工智能智能体 [32]。不过，如果开发者想要实现其他技术，首先会想到 Unity 资源商店，那里有数以千计的插件。但在种类繁多的可用工具中，用于实现人工智能的工具寥寥无几，其中只有一个基于模糊逻辑。这个插件仅仅是另一款格斗游戏开发工具的模糊逻辑层。最好的解决方案是使用一个名为 AForge.net （ AForge.NET ）的外部库。它以 C# 编写，是一个人工智能库，提供用于神经网络、计算机视觉和机器学习等领域的 API。由于该库涵盖范围广泛，其模糊逻辑实现较为浅显，但易于使用。它没有 Unity 集成，因此缺少图形用户界面，也让设计师等人员更难使用。该项目采用 LGPLv3 许可证开源，因此可以轻松扩展；而我们的工具已经具备了它所缺少的大部分要素，例如更多的隶属函数、算子或去模糊化方法。Ithaca 从零开始构建，专为 Unity 而设计，因此实现了可附加到游戏对象上的组件。它还提供图形用户界面，可用于定义系统并进行调试，无需编写一行代码。所有这些都使我们的解决方案更加完善、更强大且更灵活；得益于与 Unity 的集成，它也更易于使用。

III. ITHACA

A. *Ithaca* 是什么

Ithaca 是一种工具，可将模糊逻辑推理系统（FIS）集成到任何使用 Unity 引擎开发的项目中。它以 C# 编写，但不经过编译，因此所有代码都可供开发者访问，并可根据需要进行修改。这使该工具具有平台独立性，可在 Unity 支持的任何设备上使用。其核心是一个模糊规则系统（FRBS），用于评估由模糊集和模糊逻辑构成的一组规则。

B. 模糊逻辑与推理引擎

Ithaca 中实现的模糊推理系统采用 Mamdani 模型，其中 IF-THEN 规则的后件是如下所示的模糊语句：

如果 X 是 A 且 Y 是 B，那么 Z 是 C

这种规则写法自然直观，更适合开发旨在模拟人类决策的系统。该模型的效率低于 Sugeno 模型，后者的规则后件是代数表达式 [33] ，因此解模糊步骤的计算速度更快。对于 Mamdani 模糊推理系统的解模糊，我们实现了以下方法：

- 质心法或重心法（COG）：

$$COG = \frac{\int_{min}^{max} x\mu(x)dx}{\int_{min}^{max} \mu(x)dx}$$

- 单点质心法（COGS）：

$$COGS = \frac{\sum_i x_i \mu_i}{\sum_i \mu_i}$$

- 平分线法或面积中心法（COA）：

$$COA = u', \int_{min}^{u'} \mu(u)du = \int_{u'}^{max'} \mu(u)du$$

- 最大值平均法（MOM）：

$$MOM = \frac{\int_{A_{max}} xdx}{\int_{A_{max}} dx}$$

- 最右最大值法（RM）：

$$RM = \{x | x > x', \forall x' \in A_{max}\}$$

- 最左（LM）：

$$LM = \{x | x < x', \forall x' \in A_{max}\}$$

对于模糊化接口，我们编写了十二个隶属函数，其中一些是更通用函数的特例，但应用广泛。至于逻辑

规则元素的连接词，合取可以通过一组不同的 t-范数方法来定义。并集也是如此，可以使用多种 t-余范数中的一种来定义。实现的所有方法如下：

- Membership functions:
  – 单点型。– 分段型。–
  三角形。– 梯形。– 升
  阶型。– 降阶型。– 高
  斯型。– 双高斯型。–
  钟形。– 余弦型。– S形。–
  S 形差

- T范数（与运算）：
  – 最小值（G¨odel-Dummett）。–
  乘积 – 有界差（卢卡西维茨）。

- T余范数（或运算）：
  – 最大值。–
  代数和。–
  有界和。–
  归一化和。

C. 架构

系统架构分为三个子系统：核心、图形子系统和解析器。核心子系统包含三个主要组件（图1）：

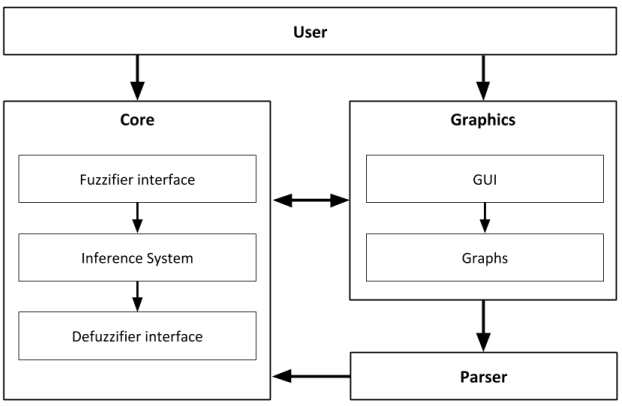


图1：系统架构

1）模糊化接口：清晰输入值在此处根据隶属函数转换为模糊值。

2）推理系统：所有规则都在此处进行评估。前件决定后件的值，结果是一个模糊值。

3）解模糊接口：输出会转换为清晰值，以便在游戏中使用。

图形子系统由图形用户界面类以及一组额外的类组成；这些类位于单独的命名空间中，用于绘制隶属函数图形（图3）。Unity没有内置的绘图系统，因此我们必须自行构建插件。我们将其与图形用户界面的其余部分解耦，使其也能用于其他场景。最后，解析器子系统将FCL文件转换为Ithaca内部格式。

推理引擎的结构基于 FCL 的定义：系统由一个功能块（FB）表示，而功能块又由一组作为输入和输出的语言变量（LV）以及一组规则（即规则块（RB））定义。每个语言变量代表一个知识域；若声明为输入，则可包含所需数量的模糊集。每个模糊集都有一个隶属函数，用于定义输入值与该模糊集的对应程度。另一方面，若语言变量声明为输出，则还具有一个额外的解模糊方法，用于将模糊值转换为清晰值。规则集可在运行时更改，因此能够根据需要调整知识库。

使用 API、图形用户界面或 FCL文件定义系统后，便可在游戏过程中调用推理引擎，根据当前值或新值请求输出。

图形用户界面的主窗口包含三个选项卡（见图2）：
- 用于创建/加载系统
- 另一个用于定义输入和输出变量
- 最后一个用于编写规则集

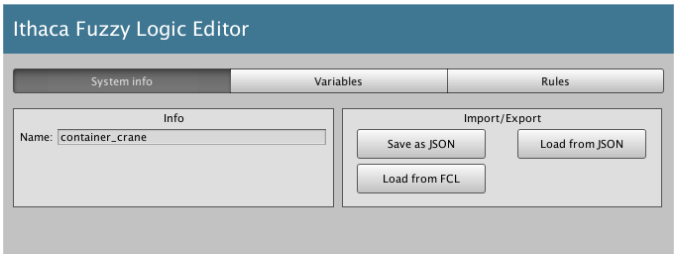


图2：Ithaca窗口

第一个选项卡允许用户创建新系统或导入已定义的系统。系统可以导出为JSON格式，也可以从JSON格式和FCL文件中重新加载。这样，用户便能以可通过版本控制软件跟踪的格式保存和交换系统。系统创建后，Unity项目中会生成一个资源文件。这是一个包含整个系统序列化数据的二进制文件。选中资源文件后，即可在此窗口中进行编辑。第二个选项卡用于声明输入和输出语言变量。创建语言变量后，可以对其进行编辑，以添加模糊集及其隶属函数（图3）。用户可以声明语言变量的范围和默认值。

最后，在第三个选项卡中，用户可以定义以自然语言编写的规则集（图4）。对于每个规则集，用户可以选择运算方法。这些运算包括：

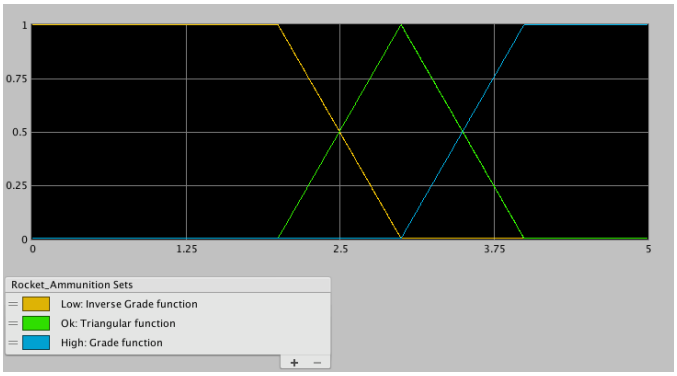


图3：语言变量编辑器

- 逻辑运算与/或的方法，也称为聚合运算。为满足德摩根定律，这些方法必须按以下组合配对：

  – 最小值-最大值–
  乘积 - 代数和 –
  有界差-有界和

- 激活运算的方法。此运算将前件的结果转换为后件。– 最小值 – 乘积

- 累积运算的方法。此运算获取所有触发规则的后件结果，并合并其值以获得一个输出。

  – 最大值–
  有界和 –
  归一化和

如果用户定义了多个规则集，则必须选择一个默认规则集。如前所述，规则库可以在运行时进行更改。

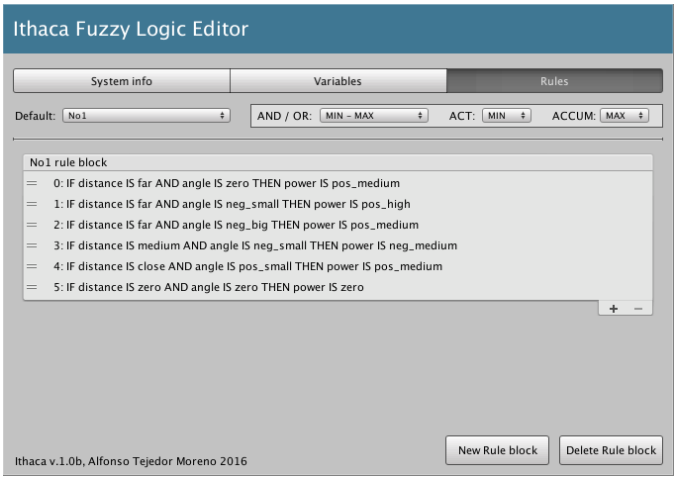


图4：规则编辑器

此外，还有一个额外窗口——调试器。开发者可以使用它调试图形值，以及已触发的规则及其值。（图5）。

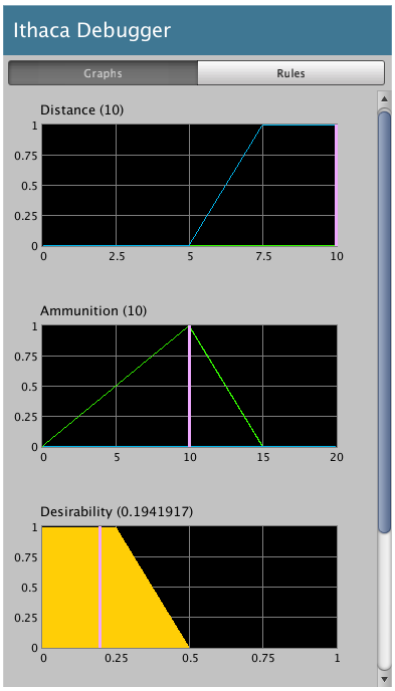


图5：调试窗口

D. 功能

Ithaca 工具的主要功能包括：

- 一个功能强大的图形用户界面，可通过图形定义完整系统。

- 所使用的推理系统是 Mamdani，它将模糊集作为规则的输出。在大多数情况下，这种方式生成的输出更便于开发者使用。

- 涵盖最常用的隶属函数、解模糊方法和运算表达式。用户可以定义新的函数、方法或表达式。

- 该代码未经过编译，因此用户可以根据需要在 Unity 中修改代码。

- 图形化调试系统可用于实时检查输入值和输出值，以及每条规则的值和是否已触发。

- 可用于在 Unity 支持的任何平台上开发游戏。

IV. 结果

A. 起重机

本测试的目标是使用我们的工具构建一个复杂系统，并检查其是否符合 FCL 标准。因此，我们决定使用 FCL 标准中定义的起重机系统 [6]，，该系统控制从船舶上装卸集装箱的集装箱起重机。控制器必须移动集装箱，同时注意速度和角度（图 6）。系统的规则如下：

1) 如果距离为远且角度为零，则功率为正中。2) 如果距离为远且角度为负小，则功率为正高。

–

–

3) 如果距离为远且角度为负大，则功率为正中。4) 如果距离为中且角度为负小，则功率为负中。5) 如果距离为近且角度为正小，则功率为正中。6) 如果距离为零且角度为零，则功率为零。

测试证明，该工具既符合 FCL 标准，也可利用起重机操作员等专家提供的自然语言知识构建复杂系统。

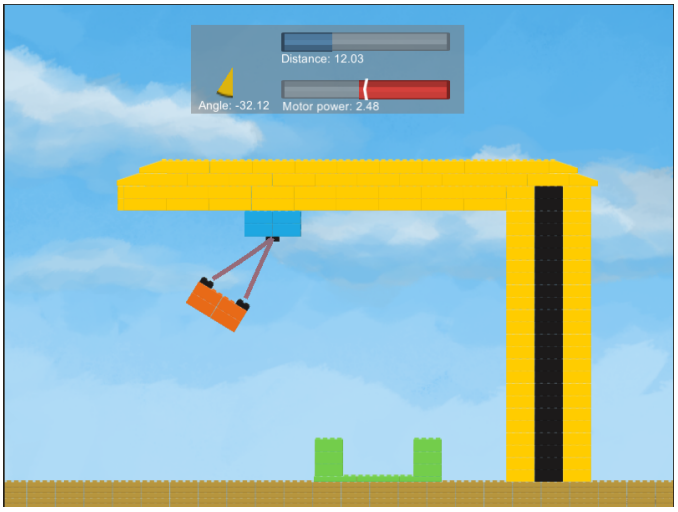


图 6：起重机仿真

B. 赛车

本次测试的目标是开发一个表现自然、逼真的对手人工智能，并完全通过图形用户界面进行开发。很多时候，为了让玩家投入其中，保持一种仿佛正在与真人对战的错觉非常重要，即使这会降低人工智能的最优程度。模糊逻辑的模糊性使其非常适合开发行为不精确、如同人类般的系统。在本例中，我们构建了两辆赛车，它们配备相同的传感器，且规则集几乎相同。其中一辆称为“经典赛车”，采用经典逻辑；另一辆称为“模糊赛车”，采用我们的推理系统。所用的知识库如下：

1) 如果 Front 为 Normal，则 Steering 为 Straight，Speed 为 Somewhat Fast。2) 如果 Front 为 Far，则 Steering 为 Straight，Speed 为 Very Fast。3) 如果 Left 为 Close，则 Steering 为 Right。4) 如果 Right 为 Close，则 Steering 为 Left。5) 如果 VeryLeft 为 Close，则 Steering 为 VeryRight。6) 如果 VeryRight 为 Close，则 Steering 为 VeryLeft。

结果表明，模糊赛车速度更快，行驶轨迹也自然得多（图 7）。至于图形用户界面的使用，结果显示该系统比使用 API 更快。

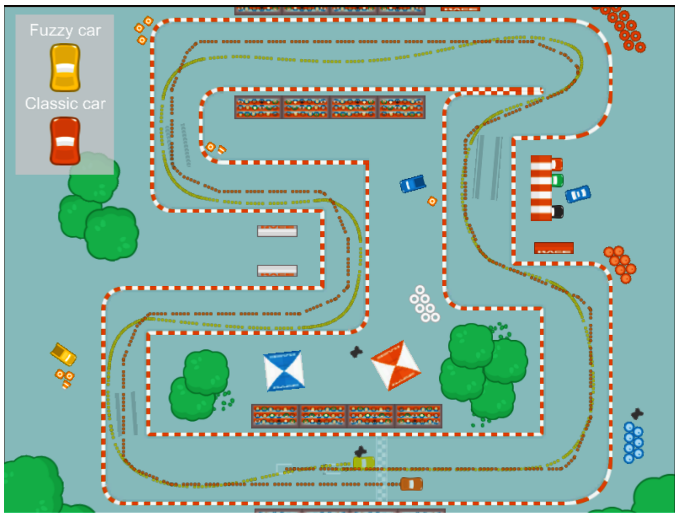


图 7：人工智能汽车行驶的路径

V. 结论

在完成这些测试和一些小型游戏的开发后，我们的工具所具备的功率显而易见。它的安装和集成到项目中都非常简单直接，让任何开发者都能尽可能轻松地迈出第一步，尤其是经验较少的开发者。值得注意的是，Unity 的许多用户都不是经验丰富的程序员。

使用图形用户界面或 FCL，可以让小型团队或独立开发者更容易构建模糊逻辑人工智能；他们可以将更多时间用于设计智能体的行为，而不是编写代码。使用自然语言编写规则并定义变量，使得不懂编程但对相关领域非常了解的工作人员也更容易参与其中。这让这些团队能够专注于玩家所感知的内容、游戏过程，以及其人工智能的行为方式。

它不仅易于开发，而且调试和维护也非常简单。借助自然语言，开发者能够随时理解系统中正在发生的事情，并知道系统将会产生什么结果。调试窗口为这项工作提供了额外帮助，因为用户可以以可视化的方式读取系统状态及其预期结果。

A. 未来工作

- 通过组合规则来实现 Combs 方法，以减少所需规则的数量。
- 添加创建模糊有限状态机（FuFSM）的选项。有限状态机是游戏开发者中非常流行的技术，而模糊逻辑的运用能够引入一定程度的不确定性，在某些情况下非常有价值。
- 用于生成实现自适应网络模糊推理系统（ANFIS）的知识库 [34]。借助神经网络，这将有助于创建更强大的系统。

- 用于实现高木-菅野推理方法。
- 为实现基于可扩展标记语言 [35]的新模糊标记语言（FML）。

致谢

本研究由欧盟欧洲区域发展基金和西班牙经济与竞争力部（MINECO）通过项目 TIN2017-83964-R 提供资助。本研究还获得了 CEiA3 和 CEIMAR 联盟的资助。

参考文献

[1] A. Nareyek，“电脑游戏中的人工智能”，Queue，第1卷，第65页，2004年10月，第58期。[2] M. Newborn，《卡斯帕罗夫对阵深蓝：电脑国际象棋走向成熟》。施普林格科学与商业媒体，2012年。[3] D. Silver、A. Huang、C. J. Maddison、A. Guez、L. Sifre、G. Van Den Driessche、J. Schrittwieser、I. Antonoglou、V. Panneershelvam、M. Lanctot等，“利用深度神经网络和树搜索掌握围棋”，《自然》，第529卷，第7587期，第484页，2016年。[4] M. Jaderberg、W. M. Czarnecki、I. Dunning、L. Marris、G. Lever、A. G. Castaneda、C. Beattie、N. C. Rabinowitz、A. S. Morcos、A. Ruder-man等，“基于种群的强化学习在3D多人游戏中的人类水平表现”，《科学》，第364卷，第6443期，第859–865页，2019年。[5] M. Pirovano，“模糊逻辑在游戏人工智能中的应用”，米兰大学，2012年。[6] 国际电工委员会，“IEC 61131-7委员会草案CD 1.0”，1997年。[7] T. S. Perry和P. Wallich，“微处理器：设计案例研究：雅达利视频计算机系统：通过省去大量硬件，设计师增加了灵活性，也为电子游戏程序员提供了创作空间”，《IEEE Spectrum》，第20卷，第3期，第45–51页，1983年。[8] A. L. Samuel，“使用跳棋游戏进行机器学习的一些研究”，《IBM研究与开发杂志》，第3卷，第3期，第210–229页，1959年。[9] S. M. Lammers，《程序员在工作：采访塑造计算机行业的19位程序员》。微软出版社，1989年。[10] I. Millington，《游戏人工智能》。CRC出版社，2019年。[11] K. Birdwell，“Valve创造《半条命》的设计流程”，1999年。[12] M. ˇCertick`y、D. Churchill、K.-J. Kim、M. ˇCertick`y和R. Kelly，“星际争霸人工智能竞赛、机器人程序与锦标赛管理软件”，《IEEE游戏汇刊》，第11卷，第3期，第227–237页，2018年。[13] O. Vinyals、T. Ewalds、S. Bartunov、P. Georgiev、A. S. Vezhnevets、M. Yeo、A. Makhzani、H. K¨uttler、J. Agapiou、J. Schrittwieser等，“星际争霸II：强化学习的新挑战”，arXiv预印本 arXiv:1708.04782，2017年。[14] A. Dobrovsky、C. W. Wilczak、P. Hahn、M. Hofmann和U. M. Borghoff，“严肃游戏中的深度强化学习：深度神经网络架构的分析与设计”，载于计算机辅助系统理论国际会议。施普林格，2017年，第314–321页。[15] G. Martinez-Arellano、R. Cant和D. Woods，“使用遗传编程为格斗游戏创建人工智能角色”，《IEEE计算智能与游戏人工智能汇刊》，第9卷，第434页，2016年第4期，第423页。[16] A. Tavafi和W. Banzhaf，“用于机器人世界杯足球仿真的混合遗传编程决策系统”，载于遗传与进化计算会议论文集，2017年，第1025–1032页。[17] J. R. Marino，“使用遗传编程学习即时战略游戏策略”，载于AAAI人工智能与互动数字娱乐会议论文集，第15卷，第1期，2019年，第219–220页。[18] Y. Azaria和M. Sipper，“GP-Gammon：通过遗传编程培养西洋双陆棋棋手”，《遗传编程与可进化机器》，第6卷，第3期，第283–300页，2005年。[19] A. Hauptman和M. Sipper，“GP-Endchess：使用遗传编程进化国际象棋残局棋手”，载于欧洲遗传编程会议。施普林格，2005年，第120–131页。

[20] L. Chen、G. Duan、S. Wang和J. Ma，“基于Choquet积分的模糊逻辑方法，用于解决不确定多准则决策问题”，《*Expert Systems with Applications*》，第149卷，第113303页，2020年。[21] R. Nimri和M. Phillip，“人工胰腺：模糊逻辑与血糖控制”，《*Current Opinion in Endocrinology, Diabetes and Obesity*》，第21卷，第4期，第251–256页，2014年。[22] P. Gupta，“模糊逻辑在日常生活中的应用。”《*International Journal of Advanced Research in Computer Science*》，第8卷，第5期，2017年。[23] S. Basak、R. Jia和C. Lei，“基于模糊逻辑的人脸识别”，载于*2018*年*IEEE*国际信息与自动化会议（*ICIA*）。IEEE，2018年，第1317–1322页。[24] D. Brulin、Y. Benezeth和E. Courtial，“基于模糊逻辑的姿态识别：用于老年人居家监测”，《*IEEE*生物医学信息技术汇刊》，第16卷，第982页，2012年第5期，第974页。[25] P. N. Krishna、S. R. Gupta、P. Shankaranarayanan、S. Sidharth和M. Sirphi，“基于模糊逻辑的智能家居能源管理系统”，载于*2018*年第九届计算、通信与网络技术国际会议（*ICCCNT*）。IEEE，2018年，第1–5页。[26] A. Khabbaz、A. Pouyan、M. Fateh和V. Abolghasemi，“一款结合强化学习与模糊逻辑、面向自闭症儿童的自适应学习游戏”，《人工智能与数据挖掘杂志》，第7卷，第2期，第321–329页，2019年。[27] S. Iyer、R. S. Mishra、S. P. Kulkami和D. Kalbande，“评估玩游戏时的自闭症程度”，载于*2017*年第二届通信系统、计算与*IT*应用国际会议（*CSCITA*）。IEEE，2017年，第42–47页。[28] S. S. Esfahlani、S. Cirstea、A. Sanaei和G. Wilson，“用于运动障碍康复严肃游戏的自适应自组织模糊逻辑控制器”，载于*2017*年*IEEE*第*26*届工业电子国际研讨会（*ISIE*）。IEEE，2017年，第1311–1318页。[29] R. Massoud、S. Poslad、F. Bellotti、R. Berta、K. Mehran和A. De Gloria，“用于估算驾驶员燃油消耗的模糊逻辑模块：面向现实增强型严肃游戏”，《国际严肃游戏杂志》，第5卷，第4期，第45–62页，2018年。[30] R. Massoud、F. Bellotti、S. Poslad、R. Berta和A. De Gloria，“在物联网赋能的严肃游戏中探索模糊逻辑与随机森林，以估算汽车驾驶员的燃油消耗”，载于*IEEE*自主分布式系统国际研讨会（*ISADS*），2019年。[31] C. A. Lara、H. Mitre-Hernandez、J. Flores和H. Perez，“通过模糊控制系统在教育电子游戏中诱发情绪状态”，《*IEEE*情感计算汇刊》，2018年。[32] H. C. M. Karangiya，“训练智能体使用强化学习玩游戏”，2019年。[33] A. Kaur和A. Kaur，“Mamdani型与Sugeno型模糊推理系统在空调系统中的比较”，《国际软计算与工程杂志》（*IJSCE*），第2卷，第2期，第323–325页，2012年。[34] J. S. R. Jang，“ANFIS：自适应网络模糊推理系统”，《*IEEE*系统、人类与控制论汇刊》，第23卷，第3期，第665–685页，1993年5月。[35] I. S. Association 等，“IEEE模糊标记语言标准”，1855年。