

An Intelligent and Collaborative Multiagent System in a 3D Environment [†]

Alejandro Rodríguez-Arias ^{*} , Bertha Guijarro-Berdiñas  and Noelia Sánchez-Maróño 

Campus de Elviña s/n, Universidade da Coruña, CITIC, 15071 A Coruña, Spain; berta.guijarro@udc.es (B.G.-B.); noelia.sanchez@udc.es (N.S.-M.)

^{*} Correspondence: alejandro.rodriguez.arias@udc.es

[†] Presented at the 3rd XoveTIC Conference, A Coruña, Spain, 8–9 October 2020.

Published: 21 August 2020



Abstract: Multiagent systems (MASs) allow facing complex, heterogeneous, distributed problems difficult to solve by only one software agent. The world of video games provides problems and suitable environments for the use of MAS. In the field of games, Unity is one of the most used engines and allows the development of intelligent agents in virtual environments. However, although Unity allows working in multiagent environments, it does not provide functionalities to facilitate the development of MAS. The aim of this work is to create a multiagent system in Unity. For this purpose, a predator–prey problem was designed in which the agents must cooperate to arrest a thief driven by a human player. To solve this cooperative problem, it is required to create the representation of the environment and the agents in 3D; to equip the agents with vision, contact, and sound sensors to perceive the environment; to implement the agents' behaviors; and, finally but not less important, to build a communication system between agents that allows negotiation, collaboration, and cooperation between them to create a complex, role-based chasing strategy.

Keywords: intelligent agents; multiagent systems; unity; agent communication language

1. Introduction

Multiagent systems (MASs) are an artificial intelligence discipline that allows facing complex, heterogeneous, and distributed problems that cannot be solved by a single software agent. A MAS is a society of homogeneous or heterogeneous intelligent agents specialized in solving part of a problem who cooperate or compete for resources to achieve their goals [1]. In a MAS, each agent acts autonomously. Therefore, agents need to be able to negotiate, cooperate, and coordinate to achieve their goals [2]. This is only possible if some agent communication language (ACL) has been developed.

The world of video games provides problems and suitable environments for the use of MAS. Unity is one of the most used video game engines and allows the development of intelligent agents in virtual 2D/3D environments [3]. The Unity Machine Learning (ML) Agents project enables games and simulations run in Unity to be used as intelligent agent training environments using machine learning algorithms, enabling developers and researchers to work with Unity and machine learning [4]. Even though both Unity and ML Agents allow working in environments with multiple agents, none provide functionalities to work with MAS, that is, there are no tools for agents to communicate and, consequently, they cannot carry out tasks in a coordinated way.

The objective of this work is to create a multiagent system in Unity. To test our development, we use a predator–prey problem where the MAS, in the role of security guards, must use a collaborative strategy that allows it to catch a human player, in the role of a thief, who has stolen a treasure and escapes using the exit door in a virtual 3D environment.

2. New Functionalities for the Development of Multiagent Systems in Unity

For solving problems, an agent must be able to perceive the environment and act in consequence. In the case of complex problems in which the actions of several agents are necessary, communication actions are also mandatory to permit coordination and/or negotiation between agents. Therefore, it is necessary to build a communication system in Unity that allows this communication.

2.1. Single Agents

In our system, each agent has been endowed with three sensors that allows them to perceive the environment they are in (in our example, mainly walls, doors, and the player). These three sensors are: (1) a vision sensor that allows the agents to recognize any object in their front view cone; (2) a sound sensor that allows the agents to hear, through a nearby radio, the noise of the steps generated by the player; (3) a contact sensor that allows the agents to identify the objects that they collide with.

These agents can perform three different actions: (1) move freely around the stage using the A* algorithm to find the optimal route to their objective; (2) open/close doors to go through them; (3) capture the player once an agent collides with him.

2.2. The Agent Communication System

A communication system between agents is based on three key points: (1) the structure of messages; (2) a system for sending, creating, and receiving messages; and (3) a system for managing conversations between agents.

For the structure of the messages, it is necessary to use an agent communication language (ACL). An ACL is a standard that provides high-level methods for an agent to exchange information. The ACL itself defines the types and meanings of the messages that agents can exchange. Most ACLs are based on speech act theory [5], which means that messages are communicative acts.

As Unity lacks a communication language between agents, we developed a plugin based on the ACL proposed by the Foundation for Intelligent Physical Agents (FIPA) [6,7], reducing the number of communicative acts and parameters available in FIPA-ACL to those essential for communication and developing a collaborative strategy complex enough to solve problems.

2.3. Behavior

In our system, the agents' behavior varies over time. Initially, everyone is assigned a limited area where they patrol. When one detects the player, the agents form a team and each one assumes a role according to their current position with respect to the player, other agents, and the player's objective.

An agent can take on different roles: a *goalkeeper*, who patrols near the player's target; *defense*, attempting to block accessible rooms from the player's current position; an *attacker*, who will attempt to block exits from the player's current room; and a *pursuer*, who will chase the player across the scenario.

Agents with the same role form a team with randomly assigned leader who helps coordinate negotiations on the objective of each agent in their role (for example, which door access will be blocked). Thanks to the designed communication system, each agent maintains negotiations with other agents with the same role and shares information with all of the agents in the system to try to use the most effective strategy at all times.

An example of how the system works can be seen in <https://youtu.be/CEw7XkMIKfc>

3. Conclusions

In this work we have provided Unity and Unity ML Agents with the necessary facilities to develop a MAS, that is: (a) a communication system; (b) an ACL based on FIPA to understand the interchanged messages; (c) basic sensors to allow agents to perceive the environment. These functionalities allow

the resolution of problems that require collaborative solutions between agents, as demonstrated in the predator–prey problem implemented.

Author Contributions: These authors contributed equally to this work.

Funding: This work has been supported in part by the Xunta de Galicia (Grants ED431C 2018/34 and ED431G/01), the European Union’s H2020 SMARTEES project (Grant 763912), and the European Union FEDER funds.

References

1. Sycara, K. Multiagent Systems. *AI Magazine*, 15 June 1998.
2. Wooldridge, M. *An Introduction to Multiagent Systems*; John Wiley & Sons: Hoboken, NJ, USA, 2009.
3. Toftedahl, M.; Engström, H. A Taxonomy of Game Engines and the Tools that Drive the Industry. In Proceedings of the Digra International Conference 2019, Game, Play and the Emerging Ludo-Mix, Kyoto, Japan, 6–10 August 2019.
4. Juliani, A.; Berges, V.P.; Vckay, E.; Gao, Y.; Henry, H.; Mattar, M.; Lange, D. Unity: A general platform for intelligent agents. *arXiv* **2018**, arXiv:1809.02627.
5. Austin, J.L. *How to Do Things with Words*; Oxford University Press: Oxford, UK, 1975; Volume 88.
6. Fipa, A. *Fipa Acl Message Structure Specification*; Foundation for Intelligent Physical Agents: Geneva, Switzerland, 2002.
7. FIPA, T. Fipa communicative act library specification. Available online: <https://www.fipa.org/specs/fipa00037/PC00037E.html> (accessed on 30 November 2000).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).