The Galaxian Project: A 3D Interaction-Based Animation Engine

Philippe Mathieu and Sébastien Picault

Université Lille 1, Computer Science Dept. LIFL (UMR CNRS 8022),
Cité Scientifique 59655 Villeneuve d'Ascq Cedex, France
firstname.surname@univ-lille1.fr
http://www.lifl.fr/SMAC/

Abstract. This demonstration paper presents a space battle simulation, which was designed using the Interaction-Oriented approach IODA, and implemented within the 3D professional game engine Unity. After giving an overview of the simulation, we explain how it was built through a step-by-step setup, and how the use of interactions enable infinite extensions.

Keywords: Agent-based simulation, Interaction-Oriented Design, Video Games

1 Introduction

The Galaxian Project [1] was developed in 2011–2012 among the Multiagent research team (SMAC) at Lille 1 University, France. It was designed to run for hours on the 6 m-wide screen of the PIRVI platform[1] at the Computer Science Laboratory (LIFL).

It aims at demonstrating the capabilities of the design principles and algorithms we promote through the Interaction-Oriented approach, the IODA method [2, 3], especially for video game applications. Actually, this simulation presents an application of the IODA method to a 3D space battle, involving several kinds of spacecrafts. It is implemented within a commercial video game engine (Unity 3D[2]), with embedded pedagogical material.

2 Main purpose

Two teams are fighting: white against black. The simulation relies upon the following agents:

- the white **fighters** are the atomic units of the white team, and assault the black fighters – and vice-versa;
- the white **crusers** (resp. black) send new white fighters from time to time so as to revive the battle (actually they do nothing more);

[1] "Plateforme Interactions-Réalité Virtuelle-Images": http://www.lifl.fr/pirvi/
[2] http://unity3d.com

- **frigates** are found only in the black team: though smaller than the crusers, they are strong ships, armed with powerful turrets, and are able to fight off several white fighters at a time;
- **squads** are built only by white fighters which decide to join together, in order to be strong enough to attack black frigates, then they break up.

Galaxian relies upon the IODA method, therefore the behaviors of all those agents are expressed as *interactions*, i.e. conditions-actions rules involving two agents (a *source* agent which can perform the interaction, and a *target* which can undergo it). In addition, thanks to the modularity of this method, the behaviors were designed and tuned incrementally.

Since the simulation is aimed at running for hours, the navigation interactions were tuned first (SeekTarget, Confront, Engage, Escape) so as to ensure that spaceships keep moving in front on the camera. Then, struggle behaviors were implemented (Fire, Explode, LaunchFighter, etc.) and tested. Finally, team behavior such as creating, joining and disbanding squads were added. Additional agents and behaviors could be easily introduced in the same way to extend the simulation.

The resulting simulation model is summarized in the *interaction matrix* below (for more details about this approach, please refer to [3]):

Table 1. This *interaction matrix* describes which interactions are assigned to source/target agent families. For instance, the intersection of line "WFighter" and column "BFighter" contains all interactions that a white fighter agent can perform on a black fighter agent. Each interaction is followed by a priority level (from the point of view of the source agent) and a limit distance (which constrains the distance between source and target to make the interaction possible). The "$\emptyset$" column contains *reflexive* interactions (where the target is the source itself).

Sources \ Targets	$\emptyset$	BFighter	WFighter	BFrigate	WSquad
BFighter	(SeekTarget; 0) (Fire; 8) (Explode; 9)		(Confront; 2; 1000) (Intercept; 3; 1000) (Engage; 4; 1000) (Escape; 5; 1000)		
WFighter	(SeekTarget; 0) (Fire; 8) (Explode; 9)	(Confront; 2; 1000) (Intercept; 3; 1000) (Engage; 4; 1000) (Escape; 5; 1000)	(CreateSquad; 2; 30)		(Join; 6; 100) (Follow; 7; 1000)
BFrigate					(FightBack; 0; 200)
BCruser	(LaunchFighter; 0)				
WCruser	(LaunchFighter; 0)				
WSquad	(Disband; 0) (Fire; 3)			(Engage; 1; 5000)	(Merge; 5; 50)

The original IODA simulation engine, called JEDI, was developed in Java [3]. The main transformations to integrate it within Unity were the following:

- a rewriting from Java to C#
- a specific scheduler, so as to let the Unity engine deal with 3D computations, collisions, and rendering at its own variable framerate, while the IODA engine is in

charge of making agents select their behavior at fixed time steps, according to their perceived neighbors, their state and the interaction matrix.

3 Demonstration

The simulation is able to run a battle for hours. In order to visualize which behaviors occur during the simulation, the simulation is endowed with gizmos showing:

- the perception links between agents (see fig. 1)
- the interactions that are performed by agents (as sources), with a link to the corresponding target (see fig. 2).

Due to the portability of the Unity models, a web version is provided on-line [1]. In addition, the full demonstration (stand-alone application) encapsulates slides to explain how the IODA simulation engine selects the appropriate interactions in several concrete cases.

4 Conclusion

This demonstration shows how complex simulations, e.g. in video games, can easily be designed through an incremental process with the IODA approach. Any model can be extended simply by writing new interactions and assigning them to agents in a declarative structure, the interaction matrix. Besides, this method provides a homogeneous representations of all entities (as agents) and all behaviors (as interactions).

Acknowledgments

The Galaxian project was implemented by Marc-Antoine Dupré (associate engineer, supported by the French research institute INRIA), under the supervision of David Panzoli (post-doctoral researcher).

References

1. Dupré, M.A., Mathieu, P., Panzoli, D., Picault, S.: Galaxian. `http://www.lifl.fr/SMAC/projects/galaxian`
2. Kubera, Y., Mathieu, P., Picault, S.: Interaction-Oriented Agent Simulations : From Theory to Implementation. In: Proceedings of the 18th European Conference on Artificial Intelligence (ECAI'08). (2008) 383–387
3. Kubera, Y., Mathieu, P., Picault, S.: IODA: an interaction-oriented approach for multi-agent based simulations. Journal of Autonomous Agents and Multi-Agent Systems (2011) 1–41

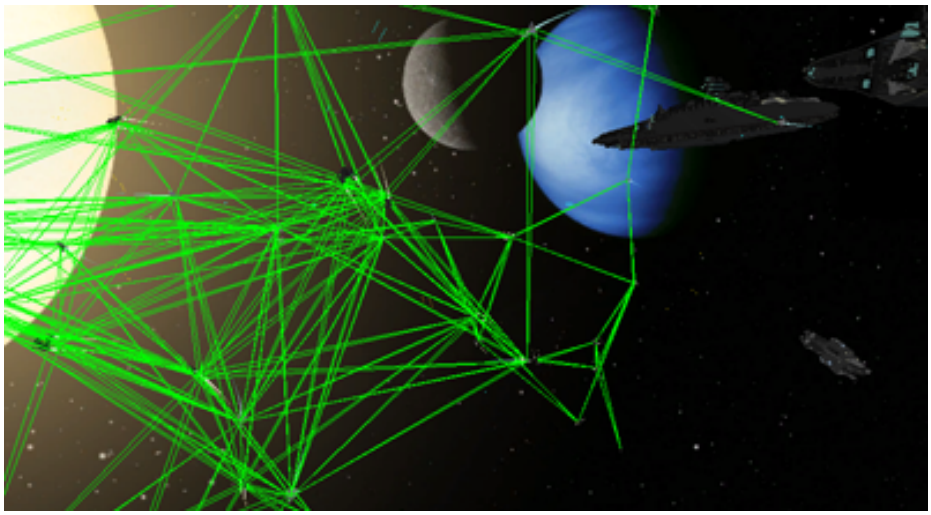


Fig. 1. A view of the Galaxian simulation. The green lines between agents show their perceptions.

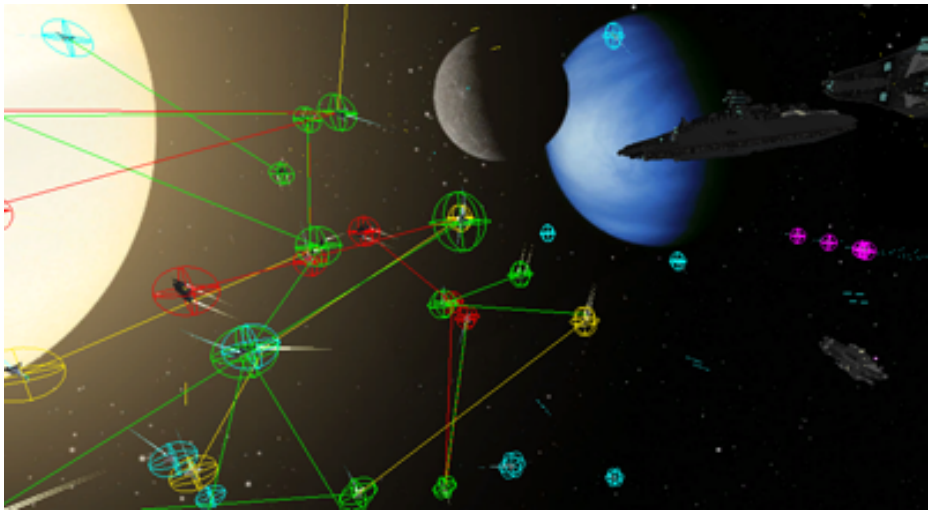


Fig. 2. A view of the Galaxian simulation. The colored circles surrounding the agents represent the interaction they are currently performing and the line ends on the corresponding target agents (the agent that is undergoing this interaction).