AngryHEX: An Angry Birds-playing Agent based on HEX-Programs

Francesco Calimeri[1] and **Stefano Germano**[1] and **Aldo Marzullo**[1] and **Nikolaus Funk**[2] and **Bianca Löhnert**[2] and **Christoph Redl**[2] and **Zeynep G. Saribatur**[2] and **Peter Schüller**[2] and **Mahak Sarin**[3] and **Daria Stepanova**[3]

calimeri@mat.unical.it, stefanogermano0@gmail.com, marzullo.aldo@gmail.com, e1425838@student.tuwien.ac.at
e1525825@student.tuwien.ac.at, redl@kr.tuwien.ac.at, zgsaribatur@gmail.com, ps@kr.tuwien.ac.at
2014csb1020@iitrpr.ac.in, dariastepanova7@gmail.com

Abstract

Our agent is a joint development of groups at Università della Calabria (UNICAL), Technische Universität Wien (TUWIEN), and Max Planck Institut für Informatik (MPI). A distinctive characteristic of our agent is that it uses a declarative, logic-programming based module for reasoning about the target to shoot at next. It is realized as a so-called HEX-program [Eiter *et al.*, 2016], i.e., by means of Answer Set Programming (ASP) with external sources and other extensions.

UNICAL[1]

Team members from Università della Calabria (Rende, Italy) are affiliated with the **Computer Science Group** at the **Department of Mathematics and Computer Science**[1]. The group focuses on formal aspects and applications of knowledge-based systems and Artificial Intelligence.

TUWIEN[2]

Team members from Technische Universität Wien (Vienna, Austria) are affiliated with and/or students of the **Knowledge Based Systems Group (KBS)**[2] at the **Institut für Logic and Computation**. The group focuses on foundations and formal aspects of knowledge-based systems and Artificial Intelligence.

MPI[3]

The team member from Max Planck Institut für Informatik (Saarbrücken, Germany) is affiliated with the department for **Databases and Information Systems**[3]. The group focuses on machine learning, data mining, and the automatic construction of knowledge-based systems.

The AngryHEX agent

Our agent, called AngryHEX, builds on the Base Framework provided by the organizers and extends it with declarative means for decision making models by means of an Answer Set Programming (ASP). Declarative logic programming controls two different layers for AngryHEX: the **Tactics** layer, which plans shots, and decides how to complete a level; and the **Strategy** layer, which decides the order of levels to play and repeated attempts to solve the same level.

Tactics is declaratively realized by HEX-programs, i.e., an extension of ASP to incorporate external sources of computation via so-called *external atoms*. It is implemented using the DLVHEX solver and computes optimal shots based on information about the current scene and on domain knowledge modeled within the HEX-program. Its *input* comprises scene information encoded as a set of logic program facts (position, size and orientation of pigs, ice, wood and stone blocks, slingshot, etc.); its *output* are *answer sets* that contain a dedicated atom describing the target to hit, and further information about the required shot. Physics simulation results and other information are accessed via external atoms.

As an extension of previous years' approach that considered only the next bird that can be shot at a target, we worked on *planning-based* declarative reasoning where multiple future shots are considered. Our agent uses single-bird tactics for playing a level the first time and attempts repetitions with the planning-based tactics which is computationally more expensive but more anticipatory.

The **Strategy** layer decides, at the end of each level, which level to play next. This layer is also realized declaratively as an (ordinary) ASP program encoding our strategy on three priority levels: (1) each available level is played once; (2) levels where the agent score differs most from the current best score are selected; (3) levels where AngryHEX achieved a score higher than the current best scores and that have the minimum difference from the best score, are selected. For each level, the **Strategy** layer keeps tracks of previously achieved scores and previously selected initial target objects.

References

[Eiter *et al.*, 2016] T Eiter, M Fink, G Ianni, T Krennwallner, C Redl, and P Schüller. A Model Building Framework for Answer Set Programming with External Computations. *Theory and Practice of Logic Programming*, 16(04):418–464, 2016.

[1] http://www.mat.unical.it

[2] http://www.kr.tuwien.ac.at

[3] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/