The Malmo Platform for Artificial Intelligence Experimentation*

Matthew Johnson, Katja Hofmann, Tim Hutton, David Bignell
Microsoft
{matjoh,katja.hofmann,a-tihutt,a-dabign}@microsoft.com

Abstract

We present Project Malmo – an AI experimentation platform built on top of the popular computer game Minecraft, and designed to support fundamental research in artificial intelligence. As the AI research community pushes for artificial general intelligence (AGI), experimentation platforms are needed that support the development of flexible agents that learn to solve diverse tasks in complex environments. Minecraft is an ideal foundation for such a platform, as it exposes agents to complex 3D worlds, coupled with infinitely varied game-play.

Project Malmo provides a sophisticated abstraction layer on top of Minecraft that supports a wide range of experimentation scenarios, ranging from navigation and survival to collaboration and problem solving tasks. In this demo we present the Malmo platform and its capabilities. The platform is publicly released as open source software at IJCAI, to support openness and collaboration in AI research.

1 Introduction

A fundamental in artificial intelligence research is how to develop *flexible* AI that can learn to perform well on a wide range of tasks, similar to the kind of flexible learning seen in humans and other animals, and in contrast to the vast majority of current AI approaches that are primarily designed to address narrow tasks. As the AI research community pushes towards more flexible AI, or artificial general intelligence (AGI) [Adams *et al.*, 2012], researchers need tools that support flexible experimentation across a wide range of tasks.

In this demo we present the Malmo platform, designed to address the need for flexible AI experimentation. The platform is built on top of the popular computer game Minecraft,[1] which we have instrumented to expose a clean and intuitive API for integrating AI agents, designing tasks, and running experiments. A wide range of tasks is supported. Figure 1 shows an example. In the following sections we give an overview of Malmo and how it can support AGI research.

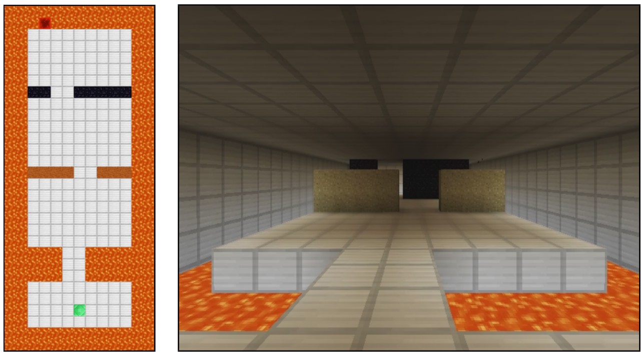

Figure 1: Example 3D navigation task from a top-down and first-person perspective. Here, the agent has to navigate to a target. A wide range of tasks can be easily defined in Malmo.

2 Project Malmo as an AGI environment

The Project Malmo platform is designed to support a wide range of experimentation needs and can support research in *robotics*, *computer vision*, *reinforcement learning*, *planning*, *multi-agent systems*, and related areas. It provides a rich, structured and dynamic environment to which agents are coupled through a natural sensorimotor loop. More generally, we believe that implements the characteristics of "AGI Environments, Tasks, and Agents" outlined in [Laird and Wray III, 2010] and refined by [Adams *et al.*, 2012] as detailed below.

C1. The environment is complex, with diverse, interacting and richly structured objects. This is supported by exposing the full, rich structure of the Minecraft game.

C2. The environment is dynamic and open. The platform supports infinitely-varied environments and mission, including, e.g., navigation, survival, and construction.

C3. Task-relevant regularities exist at multiple time scales. Like real-world tasks, missions in Malmo can have complex structure, e.g., a construction project requires navigation, mining resources, composing structures, etc.

C4. Other agents impact performance. Both AI-AI and human-AI interaction (and collaboration) are supported.

C5. Tasks can be complex, diverse and novel. New tasks can be created easily, so the set of possible tasks is infinite.

*We thank Evelyne Viegas, Chris Bishop, Andrew Blake, Jamie Shotton for supporting this project, our colleagues at MSR for insightful suggestions and discussions, and former interns Dave Abel, Nicole Beckage, Diana Borsa, Roberto Calandra, Philip Geiger, Cristina Matache, Mathew Monfort, and Nantas Nardelli, for testing an earlier version of Malmo and for providing invaluable feedback.

[1] `https://minecraft.net/`

C6. Interactions between agent, environment and tasks are complex and limited. Perception and action couple environment and agents. Several abstraction levels are provided to vary complexity within this framework.

C7. Computational resources of the agent are limited. Real-time interaction naturally constrains available resources. Additional constraints can be imposed if required.

C8. Agent existence is long-term and continual. This is naturally provided by persistent Minecraft worlds, supporting long-term agent development and lifelong learning.

In addition to addressing the above requirements, we subscribe to a set of design principles aimed to support high *speed of innovation*: (1) *Complexity gradient* – because is difficult to predict how rapidly AI technology will advance, Project Malmo supports increasingly complex tasks that can be designed to challenge to current and future technologies. (2) A *low entry barrier* is supported by providing different levels of abstractions for observations and actions. (3) *Openness* is encouraged by making the platform fundamentally cross-platform and cross-language, and by relying exclusively on widely supported data formats. Finally, we make the platform open source along with the demo at IJCAI.

3 Content of the Demo

In this demo, we show the capabilities of Project Malmo, and outline the kind of research it can support. It provides an abstraction layer and API on top of the game Minecraft. Conceptually, its abstraction is inspired by RLGLue [Tanner and White, 2009] in that the high-level components are agents that interact with an environment by perceiving observations and rewards, and taking actions. We extend this concept to support real-time interactions and multi-agent tasks.

The following high level concepts support the AI researcher who uses the Malmo platform: The *MissionSpec* specifies a mission (task) for agents to solve. This may include a definition of a map, reward signals, consumable goods, and the types of observations and action abstractions available to agents. It can be specified in XML to ensure compatibility across agents, and can be further manipulated through the API. The *MissionSpec* can use task generators, e.g., to from a task distribution. The *AgentHost* instantiates missions according to the *MissionSpec* in a (Minecraft) world and binds agents to it. It can include a *MissionRecord* to log information (e.g., to record timestamped observations, action, rewards, images and videos). The mission is then started by the *AgentHost*. During the mission the agents interact with the *AgentHost* to observe the *WorldState* and execute actions. Finally, we provide a *HumanActionComponent*, which supports missions with human interaction (for data collection, or multi-agent missions involving human players).

Demo visitors will see a variety of agent implementations complete a series of tasks, primarily focusing on navigation in diverse and increasingly challenging 3D environments. They will be able to try the HumanActionComponent on some classes of tasks that are too challenging for current AI technologies to learn to complete (e.g., complex multi-agent missions). They can learn how to set up a mission, implement an agent and run an experiment within the Malmo platform.

4 Related Work

This work builds on a long and incredibly fruitful tradition of game-supported and inspired AI research. Particularly notable recent examples include the Atari Learning Environment (ALE) [Bellemare *et al.*, 2013], which provides an experimentation layer on top of the Atari emulator Stella[2] and can run hundreds of Atari games. The General Video Game Playing Competition [Perez *et al.*, 2015] is an ongoing AI challenge series that builds on a set of computer games that is similar in spirit to the ALE set, but supports larger state spaces and the development of novel games.

[Whiteson *et al.*, 2011] first noted the danger of overfitting to individual tasks, and proposed evaluation on multiple tasks that are sampled from a distribution to encourage generality. Their proposal was fist implemented in the Reinforcement Learning Competitions [Whiteson *et al.*, 2010]. It also insipred subsequent work on generating tasks includes [Schaul, 2013] and [Coleman *et al.*, 2014]. The task generators in Malmo build on this line of work.

5 Conclusion and Outlook

We present Project Malmo – a experimentation platform designed to support fundamental research in artificial intelligence. Malmo exposes agents to a consistent 3D environment with coherent, complex dynamics. Within it, experimenters can construct increasingly complex tasks. The result is a platform on which we can begin a trajectory that stretches the capabilities of current AI technology and will push AI research towards developing future generations of AI agents that collaborate and support humans in achieving complex goals.

References

[Adams *et al.*, 2012] S. Adams, I. Arel, J. Bach, R. Coop, R. Furlan, B. Goertzel, J.S. Hall, A. Samsonovich, M. Scheutz, M. Schlesinger, et al. Mapping the landscape of human-level artificial general intelligence. *AI Magazine*, 33(1):25–42, 2012.

[Bellemare *et al.*, 2013] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *JMLR*, 47:253–279, 06 2013.

[Coleman *et al.*, 2014] O.J. Coleman, A.D. Blair, and J. Clune. Automated generation of environments to test the general learning capabilities of AI agents. *GECCO*, pages 161–168, 2014.

[Laird and Wray III, 2010] J.E. Laird and R.E. Wray III. Cognitive architecture requirements for achieving AGI. *AGI*, pages 79–84, 2010.

[Perez *et al.*, 2015] D. Perez, S. Samothrakis, J. Togelius, T. Schaul, S. Lucas, A. Couëtoux, J. Lee, C. Lim, and T. Thompson. The 2014 general video game playing competition. *IEEE Trans. Comp. Int. and AI in Games*, 2015.

[Schaul, 2013] T. Schaul. A video game description language for model-based or interactive learning. *CIG*, pages 1–8, 2013.

[Tanner and White, 2009] B Tanner and A. White. Rl-glue: Language-independent software for reinforcement-learning experiments. *JMLR*, 10:2133–2136, 2009.

[Whiteson *et al.*, 2010] S. Whiteson, B. Tanner, and A. White. The reinforcement learning competitions. *AI*, 31(2):81–94, 2010.

[Whiteson *et al.*, 2011] S. Whiteson, B. Tanner, M.E. Taylor, and P. Stone. Protecting against evaluation overfitting in empirical reinforcement learning. *ADPRL*, pages 120–127. IEEE, 2011.

[2] `http://stella.sourceforge.net/`