

Evolution of GameBots Project

Michal Bída, Martin Černý, Jakub Gemrot, Cyril Brom

▶ **To cite this version:**

Michal Bída, Martin Černý, Jakub Gemrot, Cyril Brom. Evolution of GameBots Project. 11th International Confernece on Entertainment Computing (ICEC), Sep 2012, Bremen, Germany. pp.397-400, 10.1007/978-3-642-33542-6_37 . hal-01556185

HAL Id: hal-01556185
https://inria.hal.science/hal-01556185

Submitted on 4 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Evolution of GameBots project

Michal Bída, Martin Černý, Jakub Gemrot, Cyril Brom

Charles University in Prague, Faculty of Mathematics and Physics,
Department of Software and Computer Science Education
Malostranské nám. 2/25, Prague, Czech Republic

Abstract. GameBots is a project started in early 2000s by A. N. Marshall and G. A. Kaminka. The project aims at providing researchers a real-time virtual environment testbed for their agents. GameBots utilized environment of Unreal Tournament first-person shooter game providing several scenarios for the agents. GameBots project was continued by several research groups resulting in many interesting applications. In this paper we summarize evolution of the GameBots project and contributions made since the first appearance. We focus on Pogamut GameBots branch which has been steadily developed for six years with many improvements and optimizations.

1 Introduction

The intelligent virtual agents (IVAs) research community grows increasingly interested in the use of computer games as primary evaluation testbeds. The advantage of this approach is that contemporary computer games provide ready-made rich environments with many properties of the real world without exposing developers to difficulties inherent to robotics. However, the implementation of interfaces between games and research software is a repetitive, time consuming task and introduces many caveats.

The GameBots (GB) project [1] aimed at solving this problem regarding a 3D first-person shooter game. In particular, the original GB provided access to the environment of the first Unreal Tournament (UT) [2] by TCP/IP text-based protocol. This allowed the researchers to directly connect agents written in high level languages to a complex virtual environment.

The 3D engine behind Unreal Tournament called Unreal Engine (UE) was further developed while keeping the same scripting language (UnrealScript). Thus the GB code could be ported to its newer versions without any great effort. The UE is used in the Unreal Tournament series as well as many other popular games.

Since the inception of GB, several research groups built upon the GB codebase, resulting in multiple quite different platforms for communication with the environments of various versions of the games and platforms built upon UE. Currently, there are versions of GameBots for a) the original UT, b) its sequel Unreal Tournament 2004 (UT 2004), c) its sequel Unreal Tournament 3 (UT 3), d) Unreal Engine 2 Runtime Demo (UE 2) – a free version of UE and e) Unreal Development Kit (UDK) – a free version of the latest UE.

The JavaBot project [3] features a Java library for communication with GB, allowing a user to control agents in UT. JavaBot comes with a relatively simple application for running and observing the agents. The Rochester Institute of Technology GameBots project (RIT GB) [4] ported GB to UT 2004. While RIT GB supported multiple game types, the code was not optimized for performance and did not cover all of the possible bot actions. USARSim [5] is a robotic simulator built for UT 2004, UDK and UT 3. The communication protocol is based on the original GB. USARSim extended UT 2004, UDK and UT 3 by adding multiple robots (Khepera, Aibo, flying drones, etc.) with complete physics and sensor simulation, e.g., laser, infra and sonic. To our knowledge, none of the above projects features any IDE for debugging and only USARSim project is still being actively developed.

The Pogamut project [6] ported and extended GB for UT 2004, UE 2, UT 3 and UDK, optimizing its performance, adding many more features and building an extensive Java front-end (a NetBeans plugin).

The rest of this paper discusses the Pogamut GameBots, its advantages, new features, optimizations and research applications.

2 Pogamut GameBots improvements

A major issue in previous GB implementations was performance. Our experience is that Unreal Engine in general is slow in concatenating strings and exporting large amount of textual data. This greatly limited the number of bots that could be run in parallel. By optimizing the amount of data exported by the engine, we have increased the number of bots running in parallel from approximately 4 to approximately 10 (depending on actual hardware configuration) measured in UT 2004. In this comparison, synchronous data containing updates of bots' sensory information were exported with the frequency of 4 Hz in both cases. Among the modifications made is a filtering mechanism preventing the bot to be flooded with hear sound events and we have also reduced the amount of exported synchronous data (e.g. positions of most navigation points are static and need to be exported only once).

One of common techniques used in robotics is to detect nearby obstacles by various kinds of sensors. We have added automatic raycasting for bots in Pogamut GB. This allows for steering research [7] with Pogamut GB.

To ease the development of IVAs with Pogamut GB we have introduced various debugging features into the graphical interface of the Unreal Engine for UT 2004, UT 3, UE 2 and UDK. Those include navigation graph visualization, agent information visualization (health, field of view, focus, etc.), raycasting visualization and general game info (Fig. 1).

In order to be able to make scientific comparison between bot and human behavior, a new type of connection called Observer connection was implemented. The connection a) enables listening to events triggered by a human player or a bot, and b) exports all movement and visibility information of the observed entity.

Following the interest in believable IVAs and interactive digital storytelling we have created a standalone package for UE 2 and UDK consisting of a virtual city and numerous agent avatars suitable for non-violent everyday life scenarios [8]. We also

introduced a simple communication model driven by displaying comic-like balloons with icons. This package is freely distributed at our site[1] for research and educational purposes.

To develop a human-like bot for the BotPrize competition [9] Pogamut GB were extended to better interact with moving projectiles and to allow the bots to double jump and dodge-jump. This allows for complex human-like movement in the environment that (to our knowledge) no other GB version is capable of.

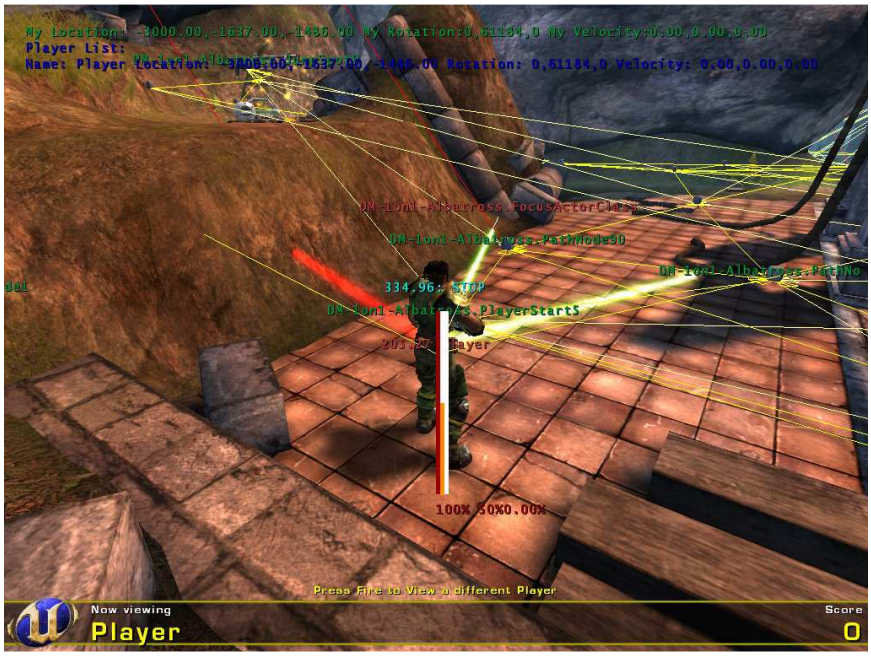


Fig. 1. Debugging features of Pogamut GB include navigation points and navigation grid visualization, raycasting visualization, bot health, armor and field of view visualization, general game information, etc. The screenshot is from UT 2004.

3 Conclusions

According to Google Scholar the use of Pogamut GB branch has been reported in nearly 100 published papers (23 by our group), including [7, 10, 11]. We believe this is a strong proof of versatility of the Pogamut GB for IVA research in general and also of their quality, stability and comprehensive documentation. While Pogamut GB are a standalone project (and are used as such), one of their main strength is the tight integration with the Pogamut platform [6], which introduces many higher level agent-oriented features on top of Pogamut GB. Porting Pogamut GB to newer versions of Unreal Engine brings opportunities to combine research with cutting-edge

[1] http://pogamut.cuni.cz/main/tiki-index.php?page=Download

entertainment computing technologies. The many new features ease the IVA development process and widen the usability spectrum to further research topics.

In general, Pogamut GB opens complex and challenging environments for IVAs research while letting the researchers focus on the implementation of the actual agent logic instead of the technical details.

Acknowledgments. This work was partially supported by a project P103/10/1287 (GACR), by student grants GA UK No. 0449/2010/A-INF/MFF and GA UK No. 655012/2012/A-INF/MFF and partially supported by SVV project number 265 314. Development of graphical content created for GameBots in Unreal Engine 2 Runtime Demo and Unreal Development Kit was supported by the project CZ.2.17/3.1.00/33274 that is financed by the European Social Fund and the Budget of the Municipality of Prague. We would also like to thank Zbynek Krulich who created majority of this graphical content using Mayang's Free Textures library: http://mayang.com/textures/.

References

1. Adobbati, R., Marshall, A.N., Scholer, A., Tejada, S., Kaminka, G.A., Schaffer, S., Sollitto, C.: GameBots: A 3D virtual world test bed for multiagent research. In: Proceedings of the Second International Workshop on Infrastructure for Agents, MAS, and Scalable MAS, Montreal, Canada (2001).
2. Epic Games, Inc.: Unreal Tournament, 2001. URL: http://en.wikipedia.org/wiki/Unreal_Tournament (30.5. 2012)
3. Marshall A., N. et. al.: JavaBot, 2002. URL: http://utbot.sourceforge.net/ (30.5. 2012)
4. Bayliss, J., Garwood, T.: GameBots for Unreal Tournament 2004, Rochester Institute of Technology, 2005. URL: http://www.cs.rit.edu/~jdb/gamebots/ (30.5. 2012)
5. Carpin, S., Lewis, M., Jijun Wang, Balakirsky, S., Scrapper, C.: USARSim: a robot simulator for research and education. In: Proceedings of the IEEE International Conference on Robotics and Automation (2007) pp. 1400-1405.
6. Gemrot, J., Kadlec, R., Bida, M., Burkert, O., Pibil, R., Havlicek, J., Zemcak, L., Simlovic, J., Vansa, R., Stolba, M., Plch, T., Brom C.: Pogamut 3 Can Assist Developers in Building AI (Not Only) for Their Videogame Agents. In: Agents for Games and Simulations, LNCS 5920, Springer (2009) pp. 1-15. URL: http://pogamut.cuni.cz (30.5. 2012)
7. Popelova, M., Bida, M. & Brom, C.: When a Couple Goes Together: Walk along Steering. In: Motion in Games, LNCS 7060, Springer Berlin / Heidelberg (2011) pp. 278-289.
8. Bida, M., Brom, C., Popelova, M., Kadlec, R.: StoryFactory – A Tool for Scripting Machinimas in Unreal Engine 2 and UDK. In: Interactive Storytelling, LNCS 7069, Springer Berlin / Heidelberg (2011) pp. 334-337.
   URL: http://www.storyfactory.cz/index.php?lang=en (30.5. 2012)
9. Hingston, P,: The 2K BotPrize. In: Proceedings of CIG 2009, Milano, Italy, 2009. URL: http://www.botprize.org (30.5. 2012)
10. Gemrot, J., Brom, C., Bryson, J., Bida, M.: How to compare usability of techniques for the specification of virtual agents' behavior? An experimental pilot study with human subjects. In: Proceedings of Agents for Games and Simulations, AAMAS workshop, Taipei, Taiwan. (2011) pp. 33-57.
11. Arrabales, R. Ledezma, A. and Sanchis, A.: Towards Conscious-like Behavior in Computer Game Characters. In: Proceedings of the IEEE Symposium on Computational Intelligence and Games 2009 (2009) pp. 217-224.