

GALA 2008 submission document v2

Title: BotControl, Second Life Agent control application
Track: Student
Category: Application
Authors: Peter Rosina
1st year Master Student
University of Augsburg, Germany

Contact data: **name:** Peter Rosina
e-mail: peter.rosina@gmail.com
url: none
Phone: +491777197374
postal address: Schaezlerstr. 17, 86150 Augsburg, Germany

URL: <http://botcontrol.de.vu>

Movie file submitted: PeterRosina_v2.AVI

Reference teacher: **affiliation:** Dr. Matthias Rehm
e-mail: rehm@informatik.uni-augsburg.de

Process of development: Developing BotControl was a major part of my bachelor's theses. It is a tool for autonomous online evaluation or guidance in Second Life (SL). It offers an intuitive GUI for debugging and testing purposes, an interface to SL, an AIML-Speech generator and a java application (that controls emotions and group behaviour (IPA)) which was developed at the University of Augsburg.

Resources used:

1. Visual Studio 2008 (C#)
2. Eclipse
3. Second Life Client
4. libsecondlife
5. AIML standard set
6. program# - aiml bot

Resources required: Internet connection, Windows Operating System with .NET 2.0-Runtime or higher, for the requirements of Second Life itself see:
<http://secondlife.com/support/sysreqs.php>

BotControl, Second Life agent control application

1. The application and context of the work

BotControl is an application that controls Second Life avatars without the need of human interaction. That way you can make online surveys and evaluations. The program itself is based on different modules to make it easy to extend the logic of the agent for further evaluations. The tool has been tested for several days in SL to gather data about acceptance and to check if other avatars detect the agent behind the avatar. All events (dialogs, users entering/leaving the agent's action area,...)were stored in logfiles for later evaluation. For further use and information, please contact the author.

2. Novelty

Existing agents in SL are built of primitive objects in SL itself and have only a limited amount of storage space for speech and logic (the script language in SL is limited). BotControl controls an avatar itself: it uses the same embodiment and has the same capabilities of movement and animation as real avatars do. The agent's knowledge base and logic are included in BotControl, not on the SL-Server itself.

3. The architecture

The application has a modular design. The speech generation, that uses AIML, is a standalone-assembly included as a library. The AIML structure has been extended with several custom tags to include animation and IPA-information (for agent's emotion towards the different users). The SL-Client uses the opensource library "libsecondlife" for communicating with the SL servers. The final module is for communicating with the "IPA"-Tool, a tool that controls the agent's behaviour towards the different avatars it meets in SL. BotControl itself is the core program that holds the main logic, offers the GUI to control the other modules and establishes communication between the them.

4. Performance

With slight modifications it would be possible to control a lot of agents at the same time to test multi agent scenarios. Every action is calculated in real time. The appropriate speech, animation etc is gathered by parsing the aiml-files. To simulate human behaviour the agent uses a random delay for its answers. The reaction of the users in SL have been very different. Some didn't even notice speaking to an agent. When others detected they started to insult or just ignore the agent. Another third of the test users liked the idea and requested background information.

References

Birgit Endraß. Social group behaviour for multiagent systems. Master's thesis, University of Augsburg, 2006.

Nicholas H.Tollervey. Program# - aimlbot. URL: <http://aimlbot.sourceforge.net/>, 2006.

Second Life Reverse Engineering Team. libsecondlife. URL: <http://www.libsecondlife.org/>