Agents

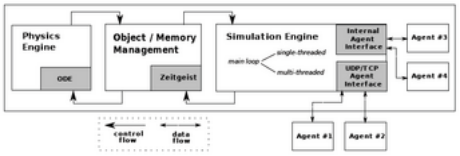
From Simspark

Contents

Programming an Agent

There are virtually unlimited ways in which you could program an agent. Out-of-process agents communicate with the simulator via TCP, so any language on any platform that supports TCP socket communication is a candidate for your agent.

In order to learn how to programming an agent read the User Manual (especially the 4th and 6th chapters) and once you understand the system, take a look at some of the various libraries and existing agents listed below.



Running an agent as a SimSpark plugin

It is possible to run agents as plugins directly in the simulator process, with no network communication required. An example is given in the SoccerbotBehavior in rcssserver3d/plugin/soccer/agentintegration. This should also be interesting for machine learning and debugging purposes.

More information is in this PDF http://monicareggiani.net/simpar2008/RoboCupSimulators/SIMPAR_simspark.pdf

What kinds of challenges are involved in creating a soccer playing agent?

In order to act in a meaningful way on the playing field the first challenge is to localize your agent on the playing field. To support this the agents perceive their relative position to a set of fixed landmarks such as the flags and goals. Dynamic objects such as other players and the ball are also seen.

If an agent knows where it is and where it wants to be in the near future the next challenge is to walk there. The structure of the humanoid models are sufficiently realistic to make this mobility non trivial. The agent must also recover from falling over.

Another challenge is kicking the ball. As trivial this sounds to a human it is far from trivial for a robot to keep its dynamic balance when kicking and controlling the direction of the ball.

Agents that are able to move and kick the ball need to cooperate and form a team. Only the effective application of strategic and cooperative behaviors forms a successful team.

SimSpark & RoboCup 3D Libraries

Several libraries exist that take care of basic communication tasks and, in many cases, provide utility classes for geometry, localisation, etc...

Library			
Zigorat	home (http://sites.google.com/site/zigorat3d/)	code (http://sites.google.com/site/zigorat3d/base-code)	ohloh (http
TinMan	home (http://code.google.com/p/tin-man)	code (http://code.google.com/p/tin-man/downloads/list)	ohloh (http
libbats	home (https://github.com/sgvandijk/libbats)	code (https://github.com/sgvandijk/libbats)	ohloh (http
AUT	home (http://www.parsianrobotic.ir/soccer3d/)	code (http://www.sarrafi.org/Parsian2008.tar.bz2)	
tsubamegaeshi-base	home (http://code.google.com/p/tsubamegaeshi-base/)	code (http://code.google.com/p/tsubamegaeshi-base/downloads/list)	
UIAI		code (http://hedayat.fedorapeople.org/uiai3d/uiai_base01Aug_12-49.tar.bz2)	
robo-cup-soccer3d-framework	home (http://code.google.com/p/robo-cup-soccer3d-framework/)	code (http://code.google.com/p/robo-cup-soccer3d-framework/downloads/list)	
JZUbase	home (http://www.nlict.zju.edu.cn/nlictrobocup/Robocup.htm)		
RoboLog	home (http://sourceforge.net/projects/robolog/)	code (http://sourceforge.net/projects/robolog/files/)	ohloh (http
Nakkaya	home (http://nakkaya.com/2010/04/26/simple-robocup-simspark-agent-in-clojure/)		
node-simspark	home (http://daizoru.github.com/node-simspark)	code (http://github.com/daizoru/node-simspark)	

Sample Agent Source Code

The following teams have made their agent source code available.

Team			
Apollo3D	home (http://sourceforge.net/projects/apollo3d/)		
magmaOffenburg	home (http://hs-offenburg.de/magma)	code (http://robocup.hs-offenburg.de/nc/downloads/)	
FIIT	home (http://sourceforge.net/projects/fiitrobocup3d/)		
Little Green Bats	home (https://launchpad.net/~littlegreenbats)	code (http://launchpad.net/littlegreenbats)	ohloh (http://ohloh.n
OPU_hana_3D		code (http://www.cs.osakafu-u.ac.jp/~nakashi/download/hana08_3D_src.tar.gz)	
opuCI_3D		code (http://www.cs.osakafu-u.ac.jp/~nakashi/download/opuCI_3D_RC2010.tar.gz)	
Nao Team Humboldt	home (http://www.naoteamhumboldt.de/projects/simple-soccer-agent/)	code (https://launchpad.net/simplesocceragent/+download)	
Delta3D	home_sourceforge (http://sourceforge.net/projects/delta3d2010/) home_google (https://sites.google.com/site/kameltareen/delta3d-source-code)		

A rudimentary agent implementation in python is available here.

Agent Binaries from Previous Events

If you know of other archives, please extend this table.

Year	Event	Location	Files	
2015	RoboCup (http://www.robocup2014.org/)	Hefei - China		[1] (http://chaosscripting.ne
2014	RoboCup (http://www.robocup2014.org/)	João Pessoa - Brazil		[1] (http://chaosscripting.ne
2013	RoboCup (http://www.robocup2013.org/)	Eindhoven - Netherlands		[1] (http://chaosscripting.ne
2012	RoboCup (http://www.robocup2012.org/)	Mexico City - Mexico		[1] (http://chaosscripting.ne
2011	RoboCup (http://www.robocup2011.org/)	Istanbul - Turkey	files (http://simspark.sourceforge.net/binaries/RoboCup2011/)	[1] (http://chaosscripting.ne
2011	IranOpen (http://2011.iranopen.ir/)	Tehran - Iran	files (http://hedayat.users.sourceforge.net/iranopen_binaries/2011/)	[1] (http://chaosscripting.ne
2010	RoboCup (http://www.robocup2010.org/)	Singapore	files (http://robocup.martenvdsanden.net/document.php?id=2)	[1] (http://chaosscripting.ne
2009	RoboCup (http://www.robocup2009.org/)	Graz - Austria	files (http://robocup.martenvdsanden.net/document.php?id=1)	[1] (http://chaosscripting.ne
2008	RoboCup (http://www.robocup-cn.org/)	Suzhou - China	files (http://robocup.martenvdsanden.net/)	[1] (http://chaosscripting.ne
2007	RoboCup (http://www.robocup-us.org/Old/robocup-2007/index.html)	Atlanta - USA	files (http://www.uni-koblenz.de/~murray/robocup/rc07/Binaries/binaries_3D.html)	

There is also a table of Log Files from previous events.