

Developing Intelligent Robots using Computer Game Engines

Félix Trépanier, Masters project in Computer System Engineering
 Supervisor: Dr. Bruce MacDonald
 Tel: 373-7599 ext. 4811, Email: frp001@ec.auckland.ac.nz
 Web Site: <http://www.ele.auckland.ac.nz/~frp001/>

Project Description

Background

Simulators are useful development tools. Many of them already exist for robotic applications. However, they are often specific to either one task or one robot. Furthermore, there is now a tendency to break robot application into distributed components. This improves the reusability of these components while establishing a framework for program sharing among research groups [3][4].

Objective

The objective of this project is to provide a simulator that will allow research groups to test and share their artificial intelligence algorithms.

Requirements

The simulator must be flexible enough to support a wide range of robots and environments. It must provide an API that allows any robot application to use the simulator and perform tests. The simulator must also be distributed and allow real robot programs to be tested in the simulation without any modification. Users must also be able to interact within the virtual environment.



Many Viewer applications can connect to the same running simulation. Users can use an Avatar to affect the environment.

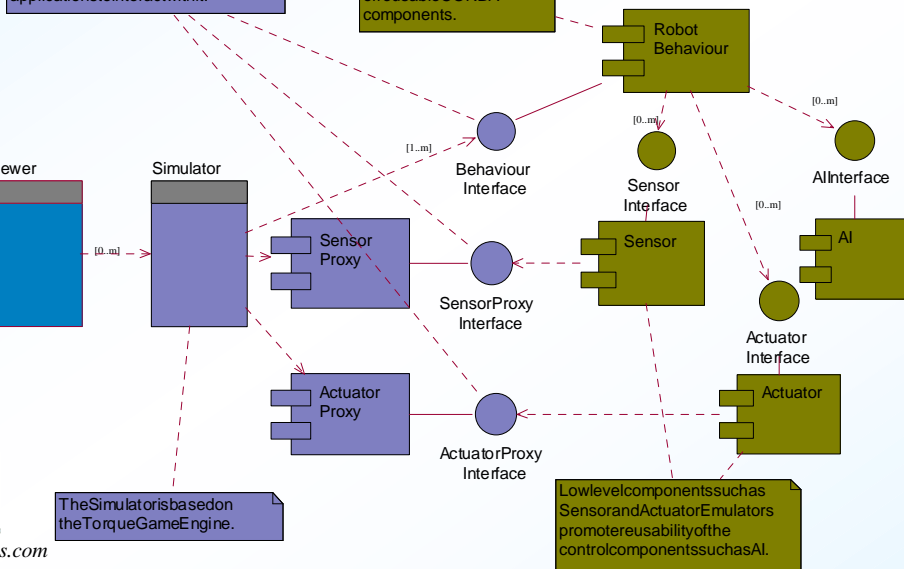


The Simulator is based on the Torque Game Engine.

Software Architecture

The simulator will provide a set of public interfaces that allow other applications to interact with it.

The Robot Application consists of a collection of reusable CORBA components.



Low level components such as Sensor and Actuator Emulators promote reusability of the control components such as AI.



How to use it?

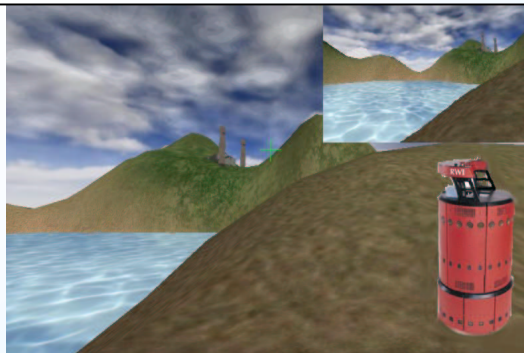
Outcomes

- A flexible simulation tool to test and develop new Artificial Intelligence algorithms.
- A distributed framework to develop robotic applications.
- A virtual meeting room for AI discussion.
- A virtual world for intelligent agents.

1. Build a 3D geometrical description of the robot.

2. Program the new components as CORBA Object servers.

3. Program the Robot Behaviour control program.



4. Create the test Environment based on a real environment or customize it to test specific algorithms' weaknesses.

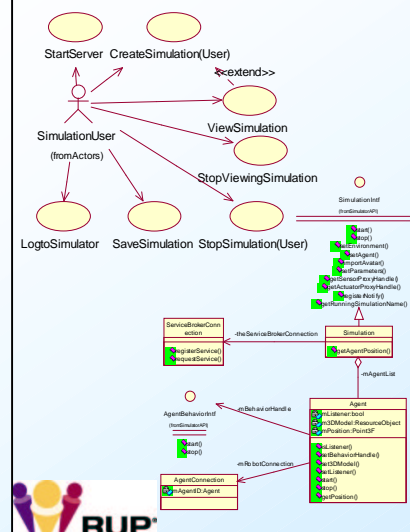
5. Many researchers can join in a watch or affect the simulation.

6. Analyze results from the simulation (e.g. sensor readings, robot positions, etc).

Software Process

This project's software development process is based on the Rational Unified Process [1] in an effort to follow industry development standards.

The Rational Rose™ C++ software was provided by the Rational company. The tool is used to model the simulator using the Unified Modelling Language. Rose™ also provides documentation and automatic code generation.



Bibliography

- [1] I. Jacobson, G. Booch and J. Rumbaugh. *The Unified Software Development Process*. Addison-Wesley
- [2] J. E. Laird. Research in human-level AI using computer games. *Communication of the ACM*, Jan. 2002
- [3] M. Humphrys and C. O'Leary. Constructing complex minds through multiple author. *Proc. of the 6th European Conference on Artificial Life*, 2001
- [4] D. Brugalí and M. E. Fayad. Distributed Computing in Robotics and Automation. *IEEE Trans. on robotics and automation*, Aug. 2002