

*DANIEL JACHYRA\**

*KAMIL WASYL\**

*PRZEMYSŁAW DZIAK\**

Wyższa Szkoła Informatyki i Zarządzania w Rzeszowie

## ANALIZA DARMOWYCH SYMULATORÓW ROBOTÓW

### Wprowadzenie

Artykuł ma charakter doświadczalny i dotyczy programów symulujących odruchy i zachowania robotów. Symulatory zostały przetestowane pod kątem funkcjonalności i możliwości dalszego rozwoju. Wszystkie testowane aplikacje są oparte na darmowej lub otwartej licencji. W artykule zamieszczono krótki opis każdego z symulatorów, a także analizę i ocenę dokonaną przez autora. Analizie poddano siedem najpopularniejszych symulatorów, skupiając się na budowie systemu, funkcjonalności technicznej oraz obszarach zastosowań. W artykule zaprezentowano aktualny stan wiedzy, jeśli chodzi o rozwój tego typu aplikacji w dziedzinie robotyki.

### 1. Symulatory robotów

Symulatory robotów są używane do tworzenia aplikacji wbudowanych dla robotów bez fizycznej obecności docelowej maszyny (roboty). Tak stworzona aplikacja może być później przeniesiona na rzeczywistego

---

\* [djachyra@wsiz.rzeszow.pl](mailto:djachyra@wsiz.rzeszow.pl); [w39057@student.wsiz.rzeszow.pl](mailto:w39057@student.wsiz.rzeszow.pl); [w41627@student.wsiz.rzeszow.pl](mailto:w41627@student.wsiz.rzeszow.pl).

robota. Skorzystanie z takiego rozwiązania pozwala zaoszczędzić czas oraz zmniejszyć koszty produkcji. Na rynku dostępne są symulatory robotów do różnych typów symulacji. Na przykład korzystając z symulatora robota mobilnego, można stworzyć prosty świat wraz z obiektami i źródłami światła, a następnie zaobserwować, jak zachowuje się w nim robot. Wykorzystując symulator behawioralny, można zauważyć bardziej naturalne akcje w porównaniu z symulacjami, które bazują tylko na obliczeniach. Co więcej, symulator behawioralny może „uczyć się” na błędach oraz wykazywać zdolności antropomorficzne.

Z symulatorami robotów związane jest także określenie *robotyka*. Jest to interdyscyplinarna dziedzina wiedzy na styku mechaniki, automatyki, elektroniki, sensoryki, cybernetyki oraz informatyki. Domeną robotyki są również rozważania nad sztuczną inteligencją, a w niektórych środowiskach robotyka jest wręcz z nią utożsamiana.

Najchętniej stosowanymi symulatorami robotów są aplikacje do modelowania w 3D oraz tworzenia renderingu robota i jego środowiska. Oprogramowanie tego typu pozwala stworzyć wirtualnego robota i emulować jego ruchy, takie jak w rzeczywistym środowisku. Niektóre symulatory mają silnik fizyki (odpowiadający za obliczenia własności fizycznych obiektów dwu- i trójwymiarowych) dla uzyskania bardziej realistycznych ruchów robota.

Należy mieć na uwadze, że sukces robota zasymulowanego *off-line* zależy od tego, na ile podobnie będzie on działał w rzeczywistości. Działania robota oparte o faktyczne odczyty z sensorów są o wiele bardziej trudne do zasymulowania. Niemniej jednak korzystanie z symulatora robota do programowania i testowania go w trybie *off-line* jest zalecane, niezależnie od tego, czy mamy do niego dostęp.

## 2. Symulatory robotów dostępne na rynku

Segment aplikacji do symulowania robotów zawiera zarówno darmowe, jak i komercyjne oprogramowanie. Programy tego typu pozwalają najczęściej na:

- szybkie prototypowanie robota;
- użycie silnika fizycznego dla uzyskania realistycznych ruchów;

- realistyczne renderowanie środowiska 3D;
- dynamizowanie robotów przez skrypty (m.in. URBI, MATLAB, Python).

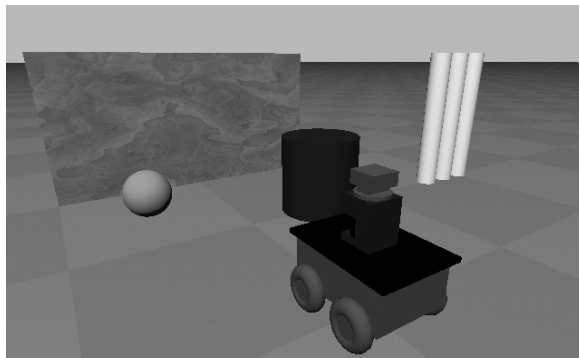
Wśród symulatorów *open source* należy wymienić: Gazebo, The Blender for Robotics software, Stage: 2.5D, breve 3d, LpzRobots, OpenSim, Simbad 3d, Khepera, Moby, UCHILSIM, SimRobot.

Do komercyjnych oprogramowań tego typu zaliczają się: Virtual Robot Experimentation Platform, anyCode Marilou, Webots Microsoft Robotics Studio, RoboLogix, Visual Components, Cogmation robotSim, SimplyCube, Actin Simulation toolkit for Robotics, Workspace 5 Robot and Automation Simulation, WorkCellSimulator WorkCellSimulator.

### 3. Projekt Gazebo

Gazebo to symulator robotów w 3D (licencja GNU), który umożliwia symulowanie wielu robotów, sensorów i obiektów. Rozpoznaje komunikaty od sensorów, a także interakcje pomiędzy obiektami. Gazebo wchodzi w skład *The Player Project*. Player jest serwerem, który po zainstalowaniu na robocie pozwala go kontrolować, na przykład przez połączenie bezprzewodowe. Player jest zgodny z systemami: Linux, BSD, Mac OS X, QNX, Solaris oraz Windows. W skład *The Player Project* wchodzi też Stage – symulator 2D. Projekt ten stworzono do wspomagania badań nad robotyką. Został on uruchomiony w 2000 roku. Jak zapewniają twórcy, Player można uruchomić na różnym sprzęcie (roboty). Istotne, że zawiera on wsparcie bibliotek klienckich dla wielu języków programowania, takich jak: C, C++, Python i Ruby. Symulatory Stage i Gazebo mogą być używane samodzielnie lub też osiągalne przez serwer Playera.

Na uwagę zasługuje znaczna liczba tutoriali związanych z projektami Gazebo i Player. Sposób instalacji i używania jest dość dobrze opisany. Wersje tego oprogramowania można uznać za aktualne – daty ich ostatniej aktualizacji to odpowiednio 13 stycznia 2010 roku oraz 10 lipca 2010 roku.



Rys. 1. Robot w środowisku projektu Gazebo

Źródło: opracowanie własne.

Wśród wspieranych robotów znajdują się: Acroname's Garcia, Botrics's Obot d100, CoroWare Inc. Corobot and Explorer, Evolution Robotics' ER1 and ERSDK robots, iRobot's Roomba vacuuming robot, K-Team's Robotics Extension Board (REB) attached to Kameleon 376BC, K-Team's Khepera, MobileRobots' (formerly ActivMedia) PSOS/P2OS/AROS-based robots, Nomadics' NOMAD200 (and possibly related) mobile robots, RWI/iRobot's RFLEX-based robots (e.g., B21r, ATRV Jr), Segway's Robotic Mobility Platform (RMP), UPenn GRASP's Clodbuster, Videre Design's ERRATIC mobile robot platform, White Box Robotics' 914 PC-BOT.

Strona internetowa symulatora Gazebo to: <http://playerstage.sourceforge.net/wiki/Gazebo>. Obsługuje on platformy: Windows, Linux i Mac OS. Autor przyznaje mu ocenę 4/6.

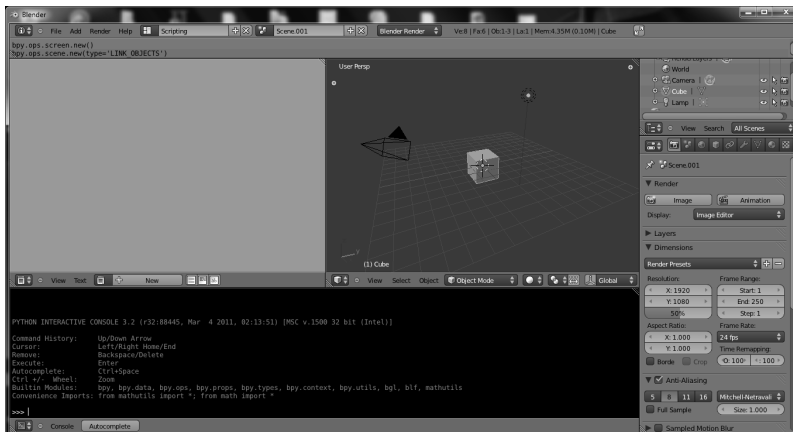
#### 4. The Blender for Robotics

Jak sugeruje nazwa projektu *The Blender For Robotics*, związany jest on z oprogramowaniem Blender, służącym do modelowania i renderingu obrazów oraz animacji trójwymiarowych. Twórcy projektu chcą ułatwić projektowanie złożonych systemów, między innymi pomagając w implementacji algorytmów

dotyczących: kontroli ruchu, dynamiki, przetwarzania sensorów itp. Oto kilka przykładowych zastosowań Blendera w robotyce:

- wizualizacja – prezentacja świata otaczającego robota;
- symulacja – sprawdzenie, jak zachowuje się uruchomiony program;
- emulacja – symulacja w pętli, reprezentacja zachowania robota.

Autorzy projektu zadbali o tutoriale pomagające osobom pragnącym zastosować Blendera w robotyce. Aby rozpocząć, konieczne jest posiadanie implementacji języka Python oraz odpowiednio skonfigurowanego programu Blender. Wszelkie niezbędne informacje zawarte są w witrynie: <http://wiki.blender.org/index.php/Robotics:Tutorial>.



Rys. 2. Program Blender z linią komend języka Python

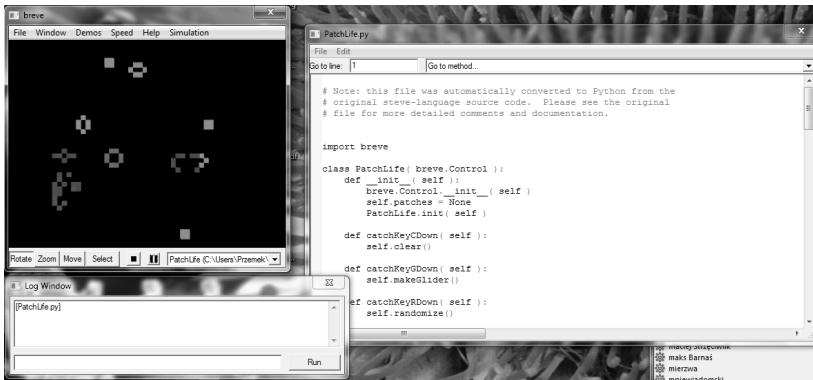
Źródło: opracowanie własne.

Strona internetowa symulatora to: <http://www.blender.org/>. Obsługiwane platformy to: Windows, Linux, Mac OS. Według autora artykułu symulator ten zasługuje na ocenę 4/6.

## 5. Breve 3D

Breve to oprogramowanie, które pozwala stworzyć w 3D sztuczny świat oraz symulować systemy wieloagentowe. Zachowanie robota programowane jest tu w języku Python. Symulator Breve w czasie pracy wspomaga się bibliotekami OpenGL. Oprogramowanie ma możliwość wykrywania kolizji. Co ważne, Breve jest dostępny w wersjach zarówno dla Mac OS X, Linux, jak i systemu Windows. Ostatnia aktualizacja projektu została przeprowadzona na początku 2008 roku.

Na stronie <http://www.spiderland.org/breve/documentation.php> dostępna jest kompletna, a zarazem niezbyt szczegółowa dokumentacja. Przygotowano również tutoriale, przez co rozpoczęcie pracy z Breve nie powinno stanowić problemu, tym bardziej, że standardowo po instalacji dostępnych jest kilka demonstracyjnych skryptów.



Rys. 3. Skrypt demo uruchomiony na symulatorze Breve

Źródło: opracowanie własne.

Strona internetowa symulatora to: <http://www.spiderland.org/>, a do obsługiwanych platform należą: Windows, Linux, Mac OS. Autor przyznaje mu ocenę 4,5/6.

## 6. OpenSim

Oprogramowanie OpenSim zostało napisane w języku C#, zatem jego uruchamianie jest możliwe pod systemem Windows, na którym zainstalowano środowisko .NET Framework. Jest też dostępna wersja dla platformy Unix, która działa w środowisku Mono (open source'owy .NET Framework dla Linuksa).

OpenSim pozwala na symulowanie wirtualnego świata dostępnego dla wielu użytkowników przez różne protokoły. Wspiera symulowanie fizyki w czasie rzeczywistym. Zawartość prezentowana w 3D może być także generowana przez klientów podłączonych do serwera OpenSim. Aktualna stabilna wersja OpenSim oznaczona jest numerem 0.7.0.2 i datowana na 10 września 2010 roku, jednak projekt nieustannie się rozwija. Wersja RC 0.7.1-rc1 (kandydat do wydania) została udostępniona 21 kwietnia 2011 roku.



Rys. 4. Przykładowy świat zbudowany przez OpenSim

Źródło: opracowanie własne.

Należy zauważyć, że światy symulowane przez OpenSim wyglądają niezwykle atrakcyjnie. Bez wątpienia pod tym względem OpenSim wyróżnia się na tle „konkurencji”. OpenSim rozpoznaje skrypty napisane w językach LSL,OSL, a także C#.

Materiały potrzebne do uruchomienia OpenSima są dostępne na stronie [http://opensimulator.org/wiki/Build\\_Instructions](http://opensimulator.org/wiki/Build_Instructions). Treści są dość obszerne, a całość napisana w języku angielskim lub japońskim. Dość duży manual jest też dołączony do paczki z plikami projektu.

Budowanie odbywa się przez program Visual Studio 2008. Stosując pewne modyfikacje opisane na stronach projektu, można też budować w Visual Studio 2010.

Strona internetowa symulatora to: <http://opensimulator.org>, zaś do obsługiwanych platform należą: Windows, Linux, Mac OS. Ocena, jaką przyznał symulatorowi autor to 5/6.

## **7. SimBad 3D**

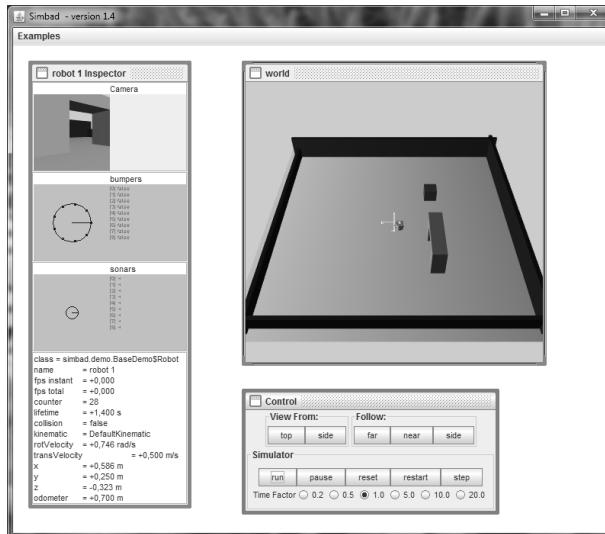
Kolejny wartą uwagi symulator to napisany w całości w języku Java SimBad 3D. Został on stworzony w celach naukowych, by umożliwić osobom zainteresowanym tą tematyką poznanie podstaw sztucznej inteligencji czy robotyki. SimBad umożliwia:

- wizualizację w 3D i teledetekcję;
- symulacje jednego bądź wielu robotów;
- wyczuwanie interakcji zachodzących w świecie.

Oprogramowanie posiada interfejs napisany z użyciem komponentu Swing. Aplikacja wymaga do pracy środowiska Java 3D. SimBad posiada domyślnie kilka przykładowych symulacji. Do programu dołączona jest dokumentacja API. Dodatkowe materiały dostępne są na stronie <http://simbad.sourceforge.net/guide.php#controller> sprawiają, że SimBad jest symulatorem naprawdę godnym uwagi.

SimBad (aktualnie dostępny w wersji 1.4) jest stale rozwijany. Ostatnia aktualizacja na stronach projektu datowana jest na 1 maja 2011 roku.





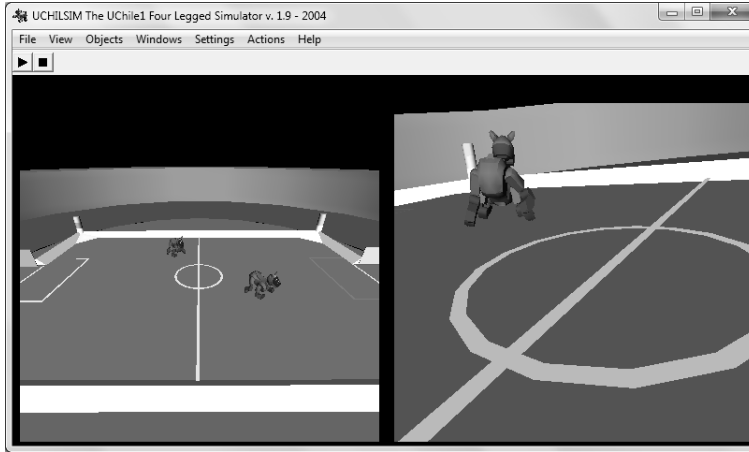
Rys. 5. Przykładowa symulacja uruchomiona w SimBad

Źródło: opracowanie własne.

Strona internetowa symulatora to: <http://simbad.sourceforge.net/>, a do obsługiwanych przez niego platform należą: Windows, Linux, Mac OS. Autor przyznał temu symulatorowi ocenę 4/6.

## 8. UchiSim

UchiSim to oprogramowanie stworzone na Uniwersytecie Chilijskim, dedykowane przedsięwzięciu RoboCup (międzynarodowe zawody robotów odbywające się od 1993 r.). Celem zawodów jest tworzenie autonomicznie działających robotów grających w piłkę nożną. Symulacja odbywa się użyciem Open Dynamics Engine (ODE).



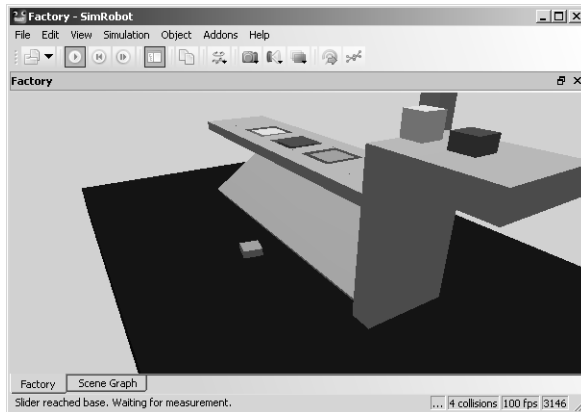
Rys. 6. Symulacja z udziałem dwóch robotów

Źródło: opracowanie własne.

Ostatnia wersja Uchilsim – 1.9 pochodzi z 2004 roku. Trudno też znaleźć dokumentację projektu. Strona internetowa symulatora to: <http://www.robocup.cl/uchilsim/>, a obsługiwane platformy to: Windows i Linux. Autor przyznał temu symulatorowi ocenę 3/6.

## 9. SimRobot

SimRobot został stworzony na Uniwersytecie w Bremen. Do kompilacji projektów potrzebny jest program Microsoft Visual Studio, co najmniej w wersji 6. Aplikacja działa we wszystkich wersjach systemów Windows, począwszy od XP, a także pod systemami Linux i Mac OS X. Program posiada prosty, a zarazem funkcjonalny interfejs z obsługą skrótów klawiaturowych. SimRobot rozwijany jest od 1994 roku, dzięki czemu jest on bardzo popularny. Dostępnych jest wiele materiałów pomocniczych do tego programu, w tym kompletnych manuali. Dobra dokumentacja z 2008 roku znajduje się pod adresem: <http://monicareggiani.net/simpar2008/RoboCupSimulators/SimRobot-LaueRoefler.pdf>.



Rys. 7. Okno główne programu SimRobot

Źródło: opracowanie własne.

Strona internetowa symulatora to: <http://www.informatik.uni-bremen.de/simrobot/>, a do obsługiwanych przez niego platform należą Windows i Linux. Autor przyznał temu symulatorowi ocenę 5,5/6.

## Podsumowanie

Analizę każdego z symulatorów zakończono jego oceną, której dokonano, uwzględniając następujące kryteria:

- aktualność – data wydania najnowszej wersji stabilnej;
- kompletność;
- funkcjonalność;
- łatwość instalacji i konfiguracji;
- liczba dostępnych tutoriali/skryptów demonstracyjnych;
- dokładność dokumentacji;
- dostępność, rozmiar społeczności użytkowników.

Oprócz komercyjnych symulatorów, dostępnych jest wiele darmowych, które doskonale sprawdzają się w celach edukacyjnych. Najbardziej zaawansowanymi i wartymi użycia symulatorami typu *open source* są: SimRobot

oraz OpenSimulator. Oba posiadają dokładną dokumentację, dużą funkcjonalność oraz aktualne wersje. W klasyfikacji ogólnej wygrywa SimRobot. Oprogramowanie to rozwijane jest od 1994 roku (stąd nawet wersja na OS/2). SimRobot pracuje w systemach Windows XP, Vista, 7, a także Linux i Mac OS X. Jak piszą twórcy, symulator służy do badań nad autonomicznymi robotami. SimRobot w odróżnieniu od innych symulatorów składa się z osobnych modułów – jądra, symulatora, interfejsu, kontrolera. Do korzystania z tego symulatora potrzebny jest program Microsoft Visual Studio 2008. SimRobot jest symulatorem najbardziej kompletnym, rozbudowanym i godnym uwagi. Wszystkie przedstawione symulatory robotów są darmowe, co czyni je również doskonałą alternatywą dla osób pragnących rozpocząć pracę w obszarze robotyki.

## Literatura

- Blender Foundation, *Blender For Robotics*, <http://wiki.blender.org/index.php/Robotics:Contents>, 06.04.2011.
- Cristóbal Zagal J., Ruiz-del-Solar J., *UCHILSIM: A Dynamically and Visually Realistic Simulator for the RoboCup Four Legged League*, 8th International Workshop on RoboCup 2004.
- Fishwick P.A., *An Introduction to OpenSimulator and Virtual Environment agent-based M&S Applications*, Winter Simulation Conference – WSC, 2009.
- Hugues L., Bredeche N., *Simbad: an Autonomous Robot Simulation Package for Education and Research*, Proceedings of The International Conference on the Simulation of Adaptive Behavior, Springer LNCS/AI series, 2006.
- Klein J., *Breve: a 3D simulation environment for the simulation of decentralized systems and artificial life*, Proceedings of Artificial Life VIII, the 8th International Conference on the Simulation and Synthesis of Living Systems, The MIT Press, 2002.
- Laue T., Rofer T., *SimRobot – Development and Applications*, *Workshop Proceedings of SIMPAR*, 2008.
- Vaughan R.T., *Massively multi-robot simulations in Stage*, “Swarm Intelligence” 2008, nr 2 (2–4).

---

**FREE ROBOTICS SIMULATORS ANALYSIS****Summary**

This work has an experimental character of programs simulating movements and robots behaviors. Simulators were tested by the author taking into account functionality and possibility for future development. All tested applications based on freeware or open source license. These work focuses on short description of each simulators, their analysis and self evaluation by author. It was analyzed seven the most popular robotics simulators concentrating on their useful interface, technical functionality and areas of uses. These article presents current state of the art in developing these kind of applications in robotics field.

**Keywords:** robotics, simulators, gazebo, OpenSim, Breve 3d

*Translated by Daniel Jachyra*

