



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Marek Bečvář

Evoluce robotů v simulovaném fyzikálním prostředí

Katedra softwaru a výuky informatiky

Vedoucí bakalářské práce: RNDr. František Mráz, CSc.

Studijní program: Informatika

Studijní obor: Informatika se specializací Umělá
inteligence

Praha 2023

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Poděkování.

Název práce: Evoluce robotů v simulovaném fyzikálním prostředí

Autor: Marek Bečvář

Katedra: Katedra softwaru a výuky informatiky

Vedoucí bakalářské práce: RNDr. František Mráz, CSc., Katedra softwaru a výuky informatiky

Abstrakt: Abstrakt.

Klíčová slova: klíčová slova

Title: Evolution of robots in a simulated physical environment

Author: Marek Bečvář

Department: Department of software and computer science education

Supervisor: RNDr. František Mráz, CSc., Department of software and computer science education

Abstract: Abstract.

Keywords: key words

Obsah

Úvod	2
1 Základní pojmy	3
1.1 Evoluční algoritmy	3
2 Specifikace	4
2.1 Funkční požadavky	4
3 Implementace	5
3.1 Programovací jazyk	5
3.2 Simulované fyzikální prostředí	5
3.2.1 Roboti	6
3.3 Genetické algoritmy	6
3.4 Implementace knihovny	6
3.5 Grafické rozhraní	6
4 Experimenty a výsledky	7
4.1 Vývoj řízení robotů	7
4.2 Vývoj řízení a morfologie robotů	7
4.3 Diskuze výsledků	7
Závěr	8
Seznam použité literatury	9
A Přílohy	10
A.1 První příloha	10

Úvod

Následuje několik ukázkových kapitol, které doporučují, jak by se měla bakalářská práce sázet. Primárně popisují použití T_EXové šablony, ale obecné rady poslouží dobře i uživatelům jiných systémů.

1. Základní pojmy

1.1 Evoluční algoritmy

2. Specifikace

2.1 Funkční požadavky

3. Implementace

V předchozí kapitole jsme prošli funkční požadavky, očekávané od vyvíjeného souboru programů. Následuje rozbor jednotlivých modulů, které vznikly při vlastní implementaci. Zároveň zde projdeme možné alternativy, které se pro vývoj nabízejí a probereme důvody stojící za zvolením jednotlivých z možností.

Nejprve vysvětlíme volbu programovacího jazyka, ve kterém je celá knihovna vytvořena. Poté projdeme systémy umožňující vývoj ve fyzikálním prostředí a ovládání uživatelem definovaných robotů. Zde představíme i možnosti tvorby vlastních robotů. Dále ukážeme možné varianty modulů umožňující vývoj řízení robotů pomocí genetických algoritmů a popíšeme vlastní implementaci. Následně projdeme všechny části implementace spojující tyto moduly do přístupné rozšiřitelné knihovny. V poslední části představíme implementaci grafického rozhraní, které slouží uživateli, který chce používat knihovnu a provádět experimenty, bez nutnosti využití příkazové řádky.

3.1 Programovací jazyk

3.2 Simulované fyzikální prostředí

Jelikož chceme vyvíjet řízení robotů založených na korektních fyzikálních pravidlech a interakcích, je pro tuto práci důležité vybrat dostatečně robustní, deterministický fyzikální simulátor. Dále bychom od tohoto simulátoru chtěli, abychom mohli dle vlastních potřeb měnit vlastnosti a podobu simulovaného prostředí. Zároveň chceme, aby nám simulátor dovolil konfigurovat morfologii vlastních robotů a případně nějakým stylem umožnil morfologii v průběhu běhu vývoje měnit. V poslední řadě by bylo užitečné, aby modul spravující zvolený fyzikální simulátor byl open-source, což nám dá volnost v případě, že si budeme chtít chování systémů v prostředí nějak vlastnoručně upravit.

Při hledání fyzikálních simulátorů, které by umožňovali kontrolu a ovládání prostředí skrz Python jsme narazili na několik možností.

- **Gazebo**

Gazebo je sada open-source knihoven pro vývoj, výzkum a aplikaci robotů. Umožňuje simulaci dynamického 3D prostředí s více agenty, generování dat ze simulovaných senzorů a fyzikálně korektní interakce robotů s prostředím. Uživatel s knihovnou pracuje skrz grafické rozhraní nebo příkazovou řádku. Prostředí a roboti mohou být tvořené buď skrz grafické prostředí, nebo v textovém formátu XML. Koenig a Howard (2004).

- **Webots**

Webots je open-source víceplatformní robotický simulátor, umožňující programování a testování virtuálních robotů a následnou aplikaci softwaru na reálné roboty. Využívá programovacích jazyků C a C++ s možností přístupu skrz Python. Prostředí dále nabízí využití připravených modelů robotů a možnosti vložení vlastních robotů z 3D modelovacích softwarů v CAD formátu Michel (2004). Více informací v dokumentaci Webots (Webots).

3.2.1 Roboti

3.3 Genetické algoritmy

3.4 Implementace knihovny

3.5 Grafické rozhraní

4. Experimenty a výsledky

4.1 Vývoj řízení robotů

4.2 Vývoj řízení a morfologie robotů

4.3 Diskuze výsledků

Závěr

Seznam použité literatury

- KOENIG, N. a HOWARD, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*(IEEE Cat. No. 04CH37566), volume 3, pages 2149–2154. IEEE.
- MICHEL, O. (2004). Cyberbotics ltd. webotsTM: professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1), 5.
- WEBOTS. <http://www.cyberbotics.com>. URL <http://www.cyberbotics.com>. Open-source Mobile Robot Simulation Software.

A. Přílohy

A.1 První příloha