

Modelovanie algoritmu pre šachový engine*

Martin Čajka

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
xcajka@stuba.sk

20. október 2021

Abstrakt

...

1 Úvod

„Šach je všetkým: umením, vedou aj športom“ – vyhlásil ruský šachový veľmajster Anatolij Karpov. Za posledné dve dekády sa vďaka rapídneho pokroku v oblasti umelej inteligencie tento jeho výrok aspoň z jednej časti potvrdil – šachu vládnu stroje, ktoré dominujú ľudským protivníkom na základe ich bleskových kalkulácií a schopnosti naozaj sa poučiť z vlastných chýb.

Fascinuje ma, čo stojí za takouto rýchlou evolúciou. Práve preto je cieľom mojej práce objasniť a zanalyzovať na základe akých princípov sa stavia model pre šachový algoritmus, aký prístup strojového učenia sa volí v tejto oblasti, ako sa vyhodnocuje nadsadenosť jednotlivých ťahov a mnoho ďalších problémov s ktorými sa stretnú osoby podieľajúce sa na tvorbe algoritmu pri takej komplexnej hre, akou je šach.

2 Nejaká časť

Z obr. 1 je všetko jasné.

Aj text môže byť prezentovaný ako obrázok. Stane sa z neho označný plávajúci objekt. Po vytvorení diagramu zrušte znak % pred príkazom `\includegraphics` označte tento riadok ako komentár (tiež pomocou znaku %).

Obr. 1: Rozhodujúci argument.

*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2021/22, vedenie: Vladimír Mlynarovič

3 Iná časť

Základným problémom je teda... Najprv sa pozrieme na nejaké vysvetlenie (časť 3.1), a potom na ešte nejaké (časť 3.1).¹

Môže sa zdať, že problém vlastne nejestvuje [1], ale bolo dokázané, že to tak nie je [2,3]. Napriek tomu, aj dnes na webe narazíme na všelijaké pochybné názory [4]. Dôležité veci možno *zdôrazniť kurzívou*.

3.1 Njaké vysvetlenie

Niekedy treba uviesť zoznam:

- jedna vec
- druhá vec

– x

– y

Ten istý zoznam, len číslovaný:

1. jedna vec
2. druhá vec

(a) x

(b) y

3.2 Ešte nejaké vysvetlenie

Veľmi dôležitá poznámka. Niekedy je potrebné nadpisom označiť odsek. Text pokračuje hneď za nadpisom.

4 Dôležitá časť

5 Ešte dôležitejšia časť

6 Záver

Literatúra

- [1] James O. Coplien. *Multi-Paradigm Design for C++*. Addison-Wesley, 1999.
- [2] Krzysztof Czarnecki, Simon Helsen, and Ulrich Eisenecker. Staged configuration through specialization and multi-level configuration of feature models. *Software Process: Improvement and Practice*, 10:143–169, April/June 2005.
- [3] Krzysztof Czarnecki and Chang Hwan Peter Kim. Cardinality-based feature modeling and constraints: A progress report. In *International Workshop on Software Factories, OOPSLA 2005*, San Diego, USA, October 2005.

¹Niekedy môžete potrebovať aj poznámku pod čiarou.

- [4] Carnegie Mellon University Software Engineering Institute. A framework for software product line practice—version 5.0. http://www.sei.cmu.edu/productlines/frame_report/.