

# Posudek diplomové práce

Matematicko-fyzikální fakulta Univerzity Karlovy

**Autor práce** Bc. Jan Pacovský  
**Název práce** Navigace jednotek v počítačových hrách za pomoci toků v sítích  
**Rok odevzdání** 2019  
**Studijní program** Informatika      **Studijní obor** Diskrétní modely a algoritmy

**Autor posudku** Mgr. Jakub Gemrot, Ph.D.      **Role** Vedoucí  
**Pracoviště** Katedra softwaru a výuky informatiky

## Text posudku:

Student předkládá práci zabývající se navigací skupiny jednotek (agentů) ve virtuálním prostředí (zejména počítačových her) v prostředích (mapách) typu 2D mřížka. Mřížka je zde použita pouze při definici mapy, agenti jsou aproximováni kolizním kruhem a jejich pozice v rámci buňky mřížky může být libovolná.

Práce je dobře motivovaná. Pohyb skupin jednotek je v počítačových hrách adresován pouze částečně, zejména pak pro předem definované skupiny (např. armádní jednotky v rámci válečných her v Total War: Rome či Company of Heroes), kdežto ve strategických hrách, kde hráč má možnost ovládat jednotky individuálně (např. Star Craft II), je to opomíjeno a skupiny se pohybují často po neideálních trasách skrz úzká hrdla map, což vypadá nevěrohodně.

Práce má tři části: 1) teoretickou, ve které student navrhuje pro řešení problému algoritmus založený na toku v síti, 2) implementační, ve které student vytváří simulátor pro pohyb skupin jednotek, 3) experimentální, ve které student evaluuje navrhnoutou metodu vůči baseline, kdy jsou všechny jednotky navigovány v prostředí individuálně, pomocí A\* či vektorového pole.

Návrh algoritmu je předmětem páté kapitoly a sestává se z: 1) transformace mřížky na síť tvořenou úzkými hrdly v mapě; 2) vyhledávání možných cest v této síti; 3) přiřazení agentů k jednotlivým cestám. Ad 1) Řešení používá existující algoritmus pro hledání úzkých hrdel v mapě. Student následně diskutuje možné varianty převodu mřížky na síť, kde diskutuje zejména aproximaci délky hrany výsledné sítě; experimentálně pak porovnává dva typy těchto převodů. Ad 2) Množina možných cest je hledána formou nalezení maximálního toku za použití zlepšujících cest. Problém tohoto přístupu pro navigaci je však nemožnost použití cest, které jsou konfliktní. Toto řeší student vlastním algoritmem pro křížení těchto konfliktních cest. Ad 3) Výsledné cesty se setřídí dle vzdálenosti a následně se postupně saturují jednotkami.

Implementace simulátoru je předmětem čtvrté kapitoly a zabývá se hlavně exekuční částí navigace, tedy implementací samotného pohybu agentů. Zde se student vhodně snaží přiblížit praxi v počítačových hrách, kdy pohyb agentů je založen na Raynoldsových steerinzích a obsahuje také detekci a zotavení se ze selhání exekuce přeplánováním.

Evaluace je pak popsána v šesté kapitole. Evaluace se sestává z 132 testů nad některými z map z Moving AI, které obsahují jak syntetické mapy tak mapy z počítačových her. Mezi testy jsou rovnoměrně zastoupeny ty, kde dává smysl jednotky posílat po skupinách a naopak.

Vyhodnocení metod je pak zaměřeno zejména na počet kroků, které jednotky potřebují k dosažení svého cíle (zde se studentův algoritmus jeví jako nejlepší), ale nediskutuje již reálnou časovou náročnost studentova algoritmu v porovnání s baseline, což je druhá velmi důležitá metrika pro počítačové hry.

Na práci hodnotím kladně zejména návrh algoritmu, který úlohu co do délky exekuce řeší lépe než základní metody. U simulátoru pak kladně hodnotím netriviální implementaci exekuci cesty, která se blíží reálným postupům používaných v počítačových hrách, a tedy následná evaluace není příliš syntetická a dává dobrý náhled na to, jak by se jednotky ve hrách mohly chovat. Výsledky pak ukazují, že přístup použití toků v síti by mohl pohyb velkých skupin jednotek opravdu vylepšit.

Vzhledem k tomu, že student provedl spoustu experimentů a v příloze jsou vygenerovány grafy ke všem testům, uvítal bych delší diskuzi, která by ukázala data pro testy různých tříd. Podrobně se diskutuje pouze test s ID 225, který je z třídy testů vhodných pro rozdělení jednotek do skupin. Z přehledové tabulky 6.7, která sumarizuje výsledky algoritmů na všech testovaných mapách, lze nahlédnout, že existují testy, kde baseline metody vítězí.

Otázky k obhajobě:

- 1) Porovnejte praktickou časovou náročnost jednotlivých algoritmů.
- 2) Ukažte zástupce testů, na kterých vítězí různé algoritmy (např. testy s ID 120, 77, 135, 196 a 222) a proveďte diskuzi, proč tomu tak je. Lze alespoň neformálně zobecnit, na jakých mapách či v jakých situacích je vhodné použít tu kterou metodu?

**Práci doporučuji k obhajobě.**

**Práci nenavrhuji na zvláštní ocenění.**

*Pokud práci navrhuje na zvláštní ocenění (cena děkana apod.), prosím uveďte zde stručné zdůvodnění (vzniklé publikace, významnost tématu, inovativnost práce apod.).*

**Datum** 3. June 2019

**Podpis**