

# Posudek bakalářské práce

## Matematicko-fyzikální fakulta Univerzity Karlovy v Praze

<b>Autor práce</b>	Lucia Bečvarová	
<b>Název práce</b>	Implementace hry Quoridor ve variantě pro tři hráče	
<b>Rok odevzdání</b>	2015	
<b>Studijní program</b>	Informatika	
<b>Studijní obor</b>	<del>Programování</del> – Obecná informatika – <del>Správa počítačových systémů</del> *	
<b>Autor posudku</b>	Mgr. Pavel Veselý	Vedoucí – Oponent *
<b>Pracoviště</b>	IUUK	

Prosím vyplňte hodnocení křížkem u každého kritéria. Hodnocení *OK* označuje práci, která kritérium vhodným způsobem splňuje. Hodnocení *lepší* a *horší* označují splnění nad a pod rámec obvyklý pro bakalářskou práci, hodnocení *nevyhovuje* označuje práci, která by neměla být obhájena. Hodnocení v případě potřeby doplňte komentářem. Komentář prosím doplňte všude, kde je hodnocení jiné než *OK*.

### K celé práci

lepší OK horší nevyhovuje

K celému zadání	lepší	OK	horší	nevyhovuje
Obtížnost zadání		X		
Splnění zadání		X		
Rozsah práce <i>... textová i implementační část, zohlednění náročnosti</i>		X		
Komentář	Autorka navrhla novou variantu hry Quoridor speciálně pro 3 hráče (hra předtím byla určena pro 2 nebo 4 hráče). Úpravy pravidel jsou dle mého názoru dobře zdůvodněné a výsledná varianta je hratelná. Autorka pro hru vytvořila grafické rozhraní (GUI). Líbí se mi, že jsou na výběr dvě velikosti herní desky. Hlavním cílem práce bylo vytvořit pro novou variantu počítačové protivníky, tedy umělou inteligenci (AI). Autorka vytvořila tři úrovně AI vyladěné testy. Přestože podle mě není příliš těžké AI nejvyšší úrovně porazit pomocí jednoho triku (při hře člověka proti dvěma AI), myslím, že vytvořit AI porážející i zkušenější hráče bude pro tuto hru těžké (asi už ve variantě pro dva hráče). Efektivnější implementace (viz níže), tedy prohledávání herního stromu o trochu hlouběji však AI zřejmě moc nepomůže (což autorka podložila experimentem) a jinak nevím, jak by bylo možné AI podstatně vylepšit. Program by se hodilo dát na internet ke stažení, nejlépe i se zdrojovými kódy.			

Posudek pokračuje na druhé straně.

\* Nehodící se škrtněte.

**Textová část práce**

lepší OK horší nevyhovuje

Formální úprava	... jazyková úroveň, typografická úroveň, citace			X	
Struktura textu	... kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu		X		
Analýza			X		
Vývojová dokumentace				X	
Uživatelská dokumentace			X		
<p><b>Komentář</b> Text je přehledně strukturovaný. Oceňuji obrázky k pravidlům, i samotný popis pravidel se mi líbí. V podkapitole 4.2.2. je popsán Dijkstrův algoritmus, přestože graf je neorientovaný a bohatě by stačilo BFS. Na str. 18 se rozebírá, že položení stěny je smysluplné jen těsně vedle nějakého hráče. To podle mě není správné a může se vyplatit zablokovat protihráči uličku ještě než k ní dojde (aby protihráč nezabránil blokování). Navíc by měl algoritmus určitě stihnout vyzkoušet všechny možné tahy, když prohledává jen do hloubky 1. Nebo má ořezávání tahů při prohledávání do hloubky 1 nějaký jiný důvod?</p> <p>Uživatelská dokumentace je dostatečná a navíc s obrázky. V textu chybí programátorská dokumentace (mimo pár zmínek o metodách v kapitole Implementace). V příloze je pouze automaticky vygenerovaná dokumentace k třídám, ale chybí celkový popis struktury kódu.</p> <p>Typografická úroveň se mi zdá slabší (mj. sazba vzorce na str. 17 a 20, rozlišení některých obrázků, vzhled tabulek)</p> <p>Drobné nedostatky:</p> <ul style="list-style-type: none"> <li>- V úvodu se píše, že popularita Quoridoru roste, ale v kap. 3.1 stojí, že není příliš populární (obojímů taky chybí citace).</li> <li>- Z textu není jasné, jestli <math>Max^n</math> zkouší všechny polohy stěn při prohledávání (narozdíl od heuristického přístupu),</li> <li>- Str. 21, bod 5 je pro hráče C (ne B). Nedá se taky spolehnout na to, že B zablokuje C (pokud B má ještě stěny)?</li> </ul>					

**Implementační část práce**

lepší OK horší nevyhovuje

Kvalita návrhu	... architektura, struktury a algoritmy, použité technologie		X		
Kvalita zpracování	... jmenné konvence, formátování, komentáře, testování			X	
Stabilita implementace			X		
<p><b>Komentář</b> Program je stabilní a GUI přehledné (jen není pohodlné přetahovat stěny pouze za jejich střed). Je škoda, že v programu nelze zvolit pořadí hráčů (tj. např. 1. hráč bude AI level 3, 2. hráč člověk a 3. bude AI level 1).</p> <p>Co se týče volby jazyka, místo Javy by bylo vhodnější mít pro umělou inteligenci zvláštní program např. v C, který by GUI programu zavoloval vždy, když by byl počítačový hráč na tahu.</p> <p>Jednotlivé verze AI byly mezi sebou porovnány pomocí partií (s různým rozložením, která verze hraje 1., 2., nebo 3.), podle mého názoru v dostatečném rozsahu. Protože tato varianta hry je nová, nebylo možné provést rozsáhlejší testy.</p> <p>Oceňuji dobré okomentování metod a vlastností, nicméně ve zdrojovém kódu se míchá angličtina a slovenština (někdy i v rámci jednoho názvu metody / proměnné). Především se mi nelíbí, že Dijkstrův algoritmus se volá několikrát, když stačí jen jednou. Navíc mi přijde, že implementace Dijkstrova algoritmu běží v kvadratickém čase, i když jde jednoduše napsat se složitostí <math>O(n \log n)</math>, kde <math>n</math> je počet vrcholů (uvažovaný graf je rovinný). Nelíbí se mi také, že graf je reprezentován maticí sousednosti.</p> <p>Věřím, že s lepší volbou jazyka pro AI a efektivnější implementací by program mohl prohledávat herní strom do hloubky 4 nebo i 5 (místo 3), aniž by byl pomalejší. Nicméně počítačového protivníka to asi zlepší jen o trochu.</p>					

**Celkové hodnocení**Výborně – Velmi dobře – Dobře – ~~Neprospěl(a)~~ \***Práci navrhuji na zvláštní ocenění**~~Ano~~ – Ne \*

Datum 4.6.2015

Podpis

\* Nehodící se škrtněte.