

Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy v Praze

Autor práce Lukáš Meduna
Název práce Terra-Settler: vesmírná real-time strategie s API pro umělou inteligenci
Rok odevzdání 2016
Studijní program Informatika **Studijní obor** Programování a softwarové systémy
Autor posudku Mgr. Pavel Ježek, Ph.D. **Role** Oponent
Pracoviště KDSS MFF UK

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
Obtížnost zadání	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Splnění zadání	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Rozsah práce ... <i>textová i implementační část, zohlednění náročnosti</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<p>Komentář V zadání se často objevuje slovo „jednoduchý“ – bohužel mám pocit, že ho autor bral dost doslova, a zvláště část s umělou inteligencí – dále budu značit AI – (kterou dle oficiálního názvu práce a zadání vnímám jako klíčovou) zpracoval zcela minimalisticky (a zdá se, že si je toho autor vědom, protože z finálního názvu práce byla dokonce v rozporu se zadáním část o AI vypuštěna). Autor v práci obšírně popisuje, jak navrhl API pro AI, které ale spočívá jen v tom, že k veřejným metodám několika tříd přidal wrapper metody v jedné třídě AI API – není zde ale žádná podpora pro samotnou tvorbu AI (např. pro tvorbu stavových automatů nebo behavior trees, apod.), a novou AI je třeba dodat formou modifikace zdrojových kódů práce. Ukázková AI má formu jednoho obrovského switche implementující stavový automat s NEPOJMENOVANÝMI stavy 0 až 10 (s tím, že ostatní to mají dělat podobně). Na explicitní otázku „jaký je rozdíl mezi touto prací, která dle autora má podporu pro tvorbu AI, a v práci zmiňovaným Starcraftem II (zmiňovaný jako prototyp hry bez podpory vlastní AI)“ mi autor odpověděl, že to je jen to, že autorova hra je „open-source“ a tedy tím si AI mohou doprogramovat. Autor v práci ani v nejmenším neřeší možnost skutečného dodání vlastní AI běžné v jiných hrách (mimořádně třeba původní Starcraft používaný běžně na univerzitách jako prototypová platforma pro implementaci AI) formou vlastních modulů, vlastních stavových automatů, apod. dodatelných do hry bez nutnosti překládat její jádro!</p> <p>Rozsah kódu samotné hry je na první pohled na bakalářskou práci zcela dostačující (300 kB zdrojových kódů v C#), nicméně při hlubším studiu se ukazuje, že je spíše na hranici: (1) obsahuje obrovské množství duplikovaného kódu, viz níže, (2) velkou část rozsahu kódu tvoří to, že objekty ve hře mají velké množství z programátorského pohledu triviálních statistik, tedy kód roste velkým množstvím getterů a setterů (zvláště jejich explicitním rozepisováním do metod, viz níže); a práce tedy obsahuje minimum skutečné „business“ logiky aplikace.</p>				

Textová část práce

lepší OK horší nevyhovuje

Formální úprava ... <i>jazyková úroveň, typografická úroveň, citace</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Struktura textu ... <i>kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Analýza	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Vývojová dokumentace	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Uživatelská dokumentace	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<p>Komentář Autor v práci velmi pěkně popisuje historii strategických her a rozmyšlení herních mechanik jeho hry po game designové stránce, což je jistě užitečné, ale je to bohužel zcela na úkor části zabývajících se samotnou implementací práce. V práci nejsou představeny žádné cíle stran samotné implementace a např. ani požadavků na AI. V práci zcela chybí analytická část – autor ani na jediném místě nediskutuje ŽÁDNÁ svá rozhodnutí stran volby cílových platforem (např. volbu Unity engine), knihoven (způsobu serializace, ukládání XML dat, apod.), algoritmů, návrhu datových struktur, návrhu AI pro API, atd. Velká část vývojové dokumentace není součástí svázaného textu, ale přiložena jako samostatný dokument na přiloženém CD.</p> <p>Vývojová dokumentace se zabývá hlavně strukturou Unity engine, se zmínkami typu „a např. tohle je také v naší hře použito“, chybí zde celkový přehled architektury a vztahů mezi jednotlivými částmi, autor dokumentuje jen velmi vybranou podmnožinu tříd a metod – ke zbývajícím chybí i např. alespoň dokumentační komentáře (jelikož samodokumentační kód svým návrhem API rozhodně není – jeden příklad za všechny: třída MineSpeed s „magickým indexerem“ s hodnotami 0 až 4, reprezentující různé druhy zdrojů: (1) proč je vůbec takto zvláště navržena? (2) zcela chybí jakákoliv dokumentace, co čísla 0 až 4 znamenají, že 0 je biomasa, 1 voda, atd.).</p> <p>Uživatelská dokumentace dobře popisuje samotné grafické uživatelské rozhraní aplikace, pro uživatele hráče (až na to, že už mu příliš nevysvětluje samotné herní mechaniky, význam jednotlivých statistik, apod.), ale bohužel zcela ignoruje uživatele designery levelů a umělé inteligence. Autor v práci píše, že je hra konfigurovatelná XML soubory, ale nikde není zmíněno, kde je lze v práci nalézt, stejně tak k nim není poskytnuta žádná dokumentace povolené struktury (např. formou XML schématu, apod.) ani významu hodnot – např. jaký je povolený číselný rozsah, co hodnoty ve hře opravdu znamenají a jak ovlivňují její chování. Stejně tak je v práci dost textu věnováno „debugovací konzoli“, ale jako dokumentace jejího použití je uveden jen příklad několika z podporovaných proměnných, a je zmíněno, že si uživatel může zavést i své vlastní proměnné k tomu, aby si je pak mohl vypsát ...</p> <p>K čemu je taková „rozšiřitelnost“, když s takovými novými proměnnými hra vůbec nepočítá? Práce také obsahuje velké množství překlepů a typografických chyb.</p>				

Implementační část práce

lepší OK horší nevyhovuje

Kvalita návrhu ... <i>architektura, struktury a algoritmy, použité technologie</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Kvalita zpracování ... <i>jmenné konvence, formátování, komentáře, testování</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Stabilita implementace	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<p>Komentář Kromě základního OO návrhu objektů herního světa, má celá hra velmi nejasnou architekturu (přijde mi, že autor psal kód tam, kde mu to zrovna přišlo bez hlubšího zamyšlení nad nějakou abstrakcí, nebo nad udržovatelností, testovatelností a rozšiřitelností kódu). Zásadním problémem je nadměrné používání Copy&Paste místo vyčlenění do metod, např. NewGameVersus (2-krát stejný kód), nebo NewGameConquer a NewGameSurvival, TransferWorkersFromUnassigned (5-krát stejný kód), různá přetížení metody MineOutResources lišící se jen 2 řádky z 25 (cílová proměnná = co třeba místo toho udělat fci s návratovou hodnotou), a mnoho dalších!</p> <p>Dále mě při prohlížení kódu napadla spousta dalších otázek stran zvláštností návrhu – pro příklad uvedu několik z nich: Proč pro koncept vlastností nepoužívá C# vlastností, ale explicitní Get a Set metody? Proč jsou známé AI dány hodnotou statické datové položky (viz výše)? Proč jsou desítky implicitních hodnot všech možných parametrů světa v C# zdrojáku a nikoliv v nějakém konfiguračním souboru? Proč je velikost vesmíru zadána číselnou hodnotou pevně na několika místech? Proč nejsou vyřešena TODO označená na několika místech kódu? Proč se hodnoty enumů předávají do metod jako int a tam přetypovávají?</p> <p>Pojmenovávací konvence netypická pro C#, navíc mixování různých pojmenovacích konvencí (někdy camelCase, jindy podtržítka_konvence, jindy _uvodniPodtržitkoCamelCase). V kódu nejsou skoro žádné komentáře, ani u složitějších částí.</p>				

Celkové hodnocení Neprospěl(a)**Práci navrhuji na zvláštní ocenění** Ne**Datum** 10. června 2016**Podpis**