

Posudek na diplomovou práci

„Multiplatformní prostředí pro vývoj mobilních her“

Autor diplomové práce: **Ivan Vacula**

Cílem práce bylo navrhnout a naimplementovat prostředí pro tvorbu her na mobilní zařízení - podmínkou byla implementace na minimálně dvou odlišných platformách. Autor práce si zvolil platformy Windows Mobile 6 a Windows Phone 7 a pro ně naimplementoval své herní prostředí využívající jazyk Lua pro přenositelný popis her. Pro WM6 bylo použito standardní C++ implementace virtuálního stroje Lua, pro WP7 autor vytvořil vlastní implementaci Lua VM v jazyce C#. Součástí odevzdané práce je i jednoduchá ukázková hra, která beze změny Lua skriptů bez problému funguje na obou podporovaných platformách.

Přestože hlavní cíl práce autor splnil, mám k práci několik připomínek:

1. Součástí zadání práce je i požadavek, aby "návrh rozhraní herního stroje zohledňoval i ostatní významné platformy". Rozhraní herního prostředí směrem ke hrám je dosti jednoduché (podporuje základní funkcionalitu potřebou pro tvorbu 2D her) a proto asi tento základní požadavek i splňuje. Nicméně bych očekával, že s myšlenkou přenositelnosti na jiné platformy bude navržena i samotná implementace herního prostředí. Pokud jsem implementaci pochopil správně, tak tomu tak ale není - C++ implementace je na tvrdo svázána s GapiDraw, C# implementace s XNA. Pokud by bylo třeba prostředí implementovat na jinou platformu (např. využít C# implementaci pro podporu iPhone pomocí MonoTouch), tak myslím, že by ani jedna z implementací nešla použít a bylo by třeba ji vytvořit zcela znovu. To se mi zdá jako zásadní problém, hlavně při budoucím rozšiřování projektu.
2. Jako velmi slabou bych hodnotil i analytickou část práce. Ve druhé kapitole autor např. srovnává vhodné skriptovací jazyky, ale srovnání je velmi povrchní a vlastně z něj vůbec nevyplývá, zda je některý z jazyků opravdu lepší, či horší - to platí i o samotném výběru jazyka Lua (v čem je třeba lepší než zmiňovaný GameMonkey Script?). Možnosti návrhu samotného herního prostředí nebo virtuálního stroje Lua nejsou diskutovány v zásadě vůbec. Přičemž právě u implementace Lua VM by to bylo velmi přínosné, zvláště když autor občas nějaké netriviální rozhodnutí v práci zmíní, ovšem bez důkladnějšího zdůvodnění (např. reprezentace typu Variant). Velkým přínosem by též bylo srovnání autorovi implementace VM v C# vůči standardní C++ implementaci - takové srovnání ale v práci opět zcela chybí.
3. Jako velmi nedostatečnou hodnotím také vývojovou dokumentaci. Zdrojové kódy v C++ i v C# jsou sice velmi pečlivě komentované (což oceňuji), ale v práci v zásadě zcela chybí jakýkoliv shrnující popis implementace jako celku. Není tedy vůbec jasné, jak spolu jednotlivé třídy souvisí, jaké algoritmy a postupy jsou v implementaci použity. Text práce (mající bez příloh 31 stran) se věnuje pouze popisu užití herního prostředí z Lua skriptů, ale chybí zde kapitola věnující se samotné implementaci.
4. Popis rozhraní herního prostředí není úplně ideální, resp. je v mnoha místech nedostatečně detailní - např. jaký je rozdíl mezi funkcemi `engine.drawTextSimple` a `engine.drawText`, když obě mají stejné parametry i popis v textu; podrobnější zdůvodnění nepodporování některých knihovnických funkcí jazyka Lua, apod. V práci chybí také nějaký tutoriál k vytváření her v autorově prostředí - je sice dodána implementace ukázkové hry, nicméně opět (jako pro samotnou implementaci herního prostředí) je v práci obsažen pouze "uživatelský popis" a chybí k ní vývojová dokumentace.

Explicitně bych chtěl zmínit také několik pozitiv:

1. Výsledek práce se zdá být zcela funkční a stabilní.
2. Autor vytvořil kompletní implementaci virtuálního stroje Lua pro prostředí .NET.
3. Zdrojové kódy aplikace jsou přehledné a důkladně komentované.
4. Až na uvedené nedostatky je text práce docela pěkně napsaný a bez zbytečných gramatických a stylistických chyb.

I přes uvedené nedostatky ale doporučuji, aby práce byla přijata jako diplomová a připuštěna k obhajobě.