

# Posudek diplomové práce

Matematicko-fyzikální fakulta Univerzity Karlovy

**Autor práce** Bc. Roman Borufka  
**Název práce** Performance Testing Suite for Unity DOTS  
**Rok odevzdání** 2020  
**Studijní program** Informatika      **Studijní obor** Počítačová grafika a vývoj počítačových her

**Autor posudku** Mgr. Jakub Gemrot, Ph.D.      **Role** Vedoucí  
**Pracoviště** KSVI

## Text posudku:

Student se ve své práci věnoval testování výkonu nového systému DOTS implementovaného v rámci herního enginu Unity. DOTS, zkratka pro Data-Oriented Technology Stack, je řešení Unity Technologies pro implementaci vysoce výkonného C# kódu exekuoaného na CPU. DOTS systém obsahuje tři části. 1) Entity Component System, který umožňuje použití data-oriented designu pro tvorbu herní logiky; 2) C# Job System, což je knihovna, která zjednodušuje tvorbu multivláknového kódu jeho dělením na tzv. joby, které jsou rozvrhovány pro exekuci na základě závislostí specifikovaných tvůrcem; 3) Burst Compiler, což je kompilátor podmnožiny C# jazyka, tzv. High-performance C#, založený na LLVM, který při překladu dokáže využít SIMD instrukcí procesoru. Student se ve své práci zaměřil zejména na testování C# Job Systemu a Burst compiler.

Vzhledem k tomu, že DOTS je systém s uzavřeným kódem, přistoupil student k jeho testování formou black-box testování. Student vytvořil jednoduchý framework pro implementaci a on-demand spouštění testů, jejichž dobu běhu loguje do CSV souborů. Student vytvořil celkem 39 testů, které zjišťují navýšení výkonu při použití 2+3 na elementárních operátorech a matematických operacích, prací s tzv. native kolekcemi a impaktu použití konstruktů C# jazyka (např. použití vlastností, hníždění struktur, či použití rozhraní) včetně různé parametrizace C# Job System. Tyto testy pak spouští na třech zařízeních – dvou různých PC se systémem Windows 10 (liší se typem CPU a grafickou kartou – dedikovaná vs. integrovaná) a mobilním telefonem se systémem Android.

Výsledky jsou sumarizovány v přehledných grafech. Většina testů potvrzuje lineární navýšení výkonu za použití 2+3, některé testy však přinášejí překvapivé výsledky: nečekaná pomalejší inicializace nativních kolekcí za použití Allocator.TempJob oproti Persistent či různorodost výkonu při menších datech. Na základě těchto testů student sestavuje sadu doporučení pro tvorbu kódu pro Unity DOTS. Účinnost těchto doporučení pak ilustruje na implementaci komplexní úlohy simulaci boidů, kdy kód sledující doporučení studenta je o 7-25% výkonnější (časově) na daném příkladu.

Práci považuji za zdařilou, implementace je stabilní a výsledky práce jsou dobře využitelné v praxi při tvorbě her v herním enginu Unity. Práci doporučuji k obhajobě.

**Práci doporučuji k obhajobě.**

**Práci nenavrhují 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** 27. 1. 2020

**Podpis**