

# Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

<b>Autor práce</b>	Petr Martinek	
<b>Název práce</b>	Vizualizace změny kódu	
<b>Rok odevzdání</b>	2022	
<b>Studijní program</b>	Informatika	
<b>Studijní obor</b>	Informatika se specializací Obecná informatika [IPP1]	
<b>Autor posudku</b>	Filip Kliber	Oponent
<b>Pracoviště</b>	Katedra distribuovaných a spolehlivých systémů	

## K celé práci

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Obtížnost zadání		X		
Splnění zadání		X		
Rozsah práce <i>... textová i implementační část, zohlednění náročnosti</i>		X		
<p>Cílem práce bylo vytvořit aplikaci, která ze dvou podobných zdrojových souborů jazyka JavaScript vyrobí animaci (GIF) vizualizující postupné změny kterými bychom z jednoho zdrojového souboru dostali druhý. Součástí práce bylo umožnit uživateli do tohoto procesu zasáhnout a případně animaci ručně doladit editací stromových popisů. Textová část práce je až na nepřítomnost vývojové dokumentace zdařilá. K implementaci mám nějaké výhrady.</p>				

## Textová část práce

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Formální úprava <i>... jazyková úroveň, typografická úroveň, citace</i>		X		
Struktura textu <i>... kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu</i>	X			
Analýza		X		
Vývojová dokumentace				X
Uživatelská dokumentace	X	X		
<p>Autor v textu práce poskytne dostatečný úvod do problematiky, na který navazuje analýza, ze které autor vyvodí rozumné závěry. Detailně se zabývá problematikou reprezentování zdrojových kódu nejen v jazyce JS, ale i v podobných „C-like“ jazycích. Dobře si uvědomuje jaká úskalí jednotlivé přístupy mají a vybírá vhodnou reprezentaci. Součástí práce je kvalitní uživatelská dokumentace, která dobře popisuje ovládání programu (z příkazové řádky). V příloze je také specifikace, která pravděpodobně vznikla ve spolupráci s firmou Apify, kterou student ve své práci zmiňuje. Jasně jsou tak definované cíle práce.</p> <p>V textu téměř zcela chybí vývojová dokumentace. Autor pouze nastíní architekturu programu a popíše ve kterém zdrojovém souboru jsou implementované které komponenty aplikace. Poté stručně nastíní jak algoritmus probíhá a které komponenty se o to starají. Zcela chybí jak popis jednotlivých fází algoritmu z pohledu implementace, tak některá zásadní implementační rozhodnutí, se kterými se autor musel vypořádat.</p> <p>Autor v textu zmiňuje, že bílé znaky jsou v případě porovnávání rozdílů mezi zdrojovými soubory důležité. Pro implementaci parseru se však opírá o existující nástroj Antlr4 i s popisem gramatiky jazyka JS, která dle mého pozorování množství bílých znaků ignoruje.</p> <p>Rád bych vyzdvihl kvalitní popis použitých algoritmu pro detekci posloupnosti úprav, které autor vymyslel sám.</p>				

## Implementační část práce

lepší OK horší nevyhovuje

Kvalita návrhu ... architektura, struktury a algoritmy, použité technologie			X	
Kvalita zpracování ... jmenné konvence, formátování, komentáře, testování			X	X
Stabilita implementace		X		

Implementační část práce je výrazně slabší. Postupem z dokumentace se mi nepodařilo program bez chyb spustit. Hlavním problémem je nutnost existence adresáře `.output` v adresáři se zdrojovým kódem, který se používá pro dočasné soubory. Dokumentace toto vůbec nezmiňuje. Lepší řešení by bylo vytvářet dočasnou složku automaticky, ideálně nějakým mechanismem stylu `mktemp`, což by zároveň umožňovalo paralelní běh programu v různých spuštěních. V dokumentaci rovněž schází informace o tom, že je pro fungování aplikace vyžadována knihovna *ImageMagick*. Knihovna *GraphicsMagick*, kterou si autor zvolil pro práci s animovanými GIFy, se pak používá pouze tím způsobem, že se volání delegují na knihovnu *ImageMagick*, což je zvláštní řešení, které není zdokumentované (a které by mělo být ve vývojové dokumentaci). Ve zdrojovém kódu se objevuje enormní množství asynchronních funkcí, které se až na výjimky používají v synchronním kontextu, a jejich užitek je pak velmi sporný (a opět chybí rozprava ve vývojové dokumentaci). Kód obecně působí stylem, ve kterém autor kopíruje fungující kusy z různých zdrojů, aniž by jejich významu příliš rozuměl. Objevují se zde nesmyslné callback parametry, které nejsou nijak využité a pouze zatemňují orientaci v kódu.

Pro otestování funkčnosti aplikace autor zvolil sadu víceméně integračních testů, jejichž výstup je kontrolován ručně, což se u aplikace generující animované obrázky dá očekávat. Nicméně některé části aplikace by bylo možné testovat více důkladně nějakou formou unit testů (např. komplikované parsování argumentů příkazové řádky, generování mezi-výstupu nebo zjednodušené reprezentace kódu). Zároveň nerozumím tomu, z jakého důvodu autor nevyzkoušel svůj program na operačním systému Linux, když žádná z použitých technologií není nijak závislá na platformě.

V aplikaci není vůbec možné konfigurovat výsledný obrázek, např. co se týče rozlišení (velikosti fontu) nebo rychlosti animace (FPS).

**Celkové hodnocení**    Neprospěl(a)

**Práci navrhuji na zvláštní ocenění**    Ne

Datum

Podpis