

## POSUDEK OPONENTA DIPLOMOVÉ PRÁCE

**Student:** Mgr. Michaela Helísková

**Oponent:** Mgr. Bc. Pavel Ugwitz

Studijní program: Inženýrská informatika

Studijní obor: Informační technologie

Akademický rok: 2019/2020

Téma diplomové práce: **Aplikace na ovládání eye trackingového zařízení s možností komunikace se softwarem Hypothesis**

### Hodnocení práce:

Práce se zaměřuje na téma implementace eye-trackingu v kontextu webové diagnostické platformy Hypothesis. Formulace abstraktu se nicméně autorce moc nepovedla, vyznívá vágně. Neznalý si na základě přečtení abstraktu ne nutně uvědomí, co platforma Hypothesis je, resp. jaký výzkumný/aplikační potenciál z této implementace plyne.

Struktura práce je přehledná; v teoretické části se autorka zabývá eye-trackingovou teorií a technologií, softwarovými implementacemi pro eye-tracking, a architekturou diagnostického systému Hypothesis. Praktická část navazuje představením požadavků, implementací, a její verifikací.

V úvodu autorka poukazuje na silnou stránku systému Hypothesis - že potenc. složitost implementovaných testů nemá v počítačovém světě obdoby. Toto je částečně sporné tvrzení, protože toto závisí na implementační složitosti. Je zde též třeba dodat, že v tradiční "papírové" administraci psychodiagnostických testů taková testová složitost/interaktivita též nebyla, a že požadavek na interaktivní diagnostické systémy (kupř. item-response theory) vznikl právě odtud.

Teorie eye-trackingu je zpracována precizně, ilustrativně, vč. fyziologie lidského oka. Uvítal bych mj. uvedení anglických termínů, které jsou oproti českým v literatuře frekventovanější (kupř. "smooth pursuit", namísto "pomalé sledovací pohyby").

Kapitola o stávajících softwarových řešeních pro sběr/analýzu eye-trackingových dat v sobě nezahrnuje širší rešerši - což je škoda, protože přehled existujících výrobců a podporovaných eye-trackerů by mohl představovat přidanou hodnotu pro každého výzkumníka, který se rozhoduje nad užitím/implementací eye-trackingového řešení (kupř. pro výběr řešení bez hardwarových či closed-source omezení). Autorka mohla této kapitole věnovat větší pozornost, protože výčet existujících softwarů, zahrnujících v sobě komponenty jak pro záznam eye-trackingových dat, jejich vyhodnocování, či přípravu eye-trackingových stimulů, je poněkud matoucí - lepší strukturace kapitoly by pak příp. přinesla lepší vhled i do jejího vlastního softwarového řešení v praktické části práce. Kupř. zde není vůbec vedena argumentace, proč dané softwary využívají implementace za pomoci jazyků C++ či Python (rychlost exekuce daného kódu, či výhoda dostupných rozšiřujících knihoven?)

Představení architektury systému Hypothesis je provedeno srozumitelně. Kapitoly 3.3.1 a 3.3.2 (CAT, asynchronní testování) mohly být vynechány, protože se jedná o dosti specifické implementace, které byly využity snad jednou (systém Hypothesis implementuje desítky psychodiagnostických testů). Uvedené příklady XML pseudokódu mohly být názornější, s doprovodnými komentáři.

Výčet požadavků na funkcionalitu (kap. 4.2) je adekvátní, a představuje v sobě defacto základ pro implementaci eye-trackingových metod, na straně serveru, ovládajícího eye-trackingový hardware - s tím, že další implementaci funkcionality (na způsob doSomethingOnGazeEnter(), doSomethingOnGazeExit(), doSomethingGazeOnStay(), apod.) bude třeba vyhotovit na straně systému Hypothesis, či jiného klienta.

Autorka volí implementaci softwarového řešení v jazyce Python "z důvodu předchozí praktické zkušenosti s daným jazykem". Což ovšem není sám o sobě pádný důvod; lépe by bylo uvést výhody/nevýhody této implementace vzhledem k dostupným jazykům C/C++, a to vč. rychlosti implementace, požadavků na výkon, či přihlednutí na možnosti/výzvy budoucí rozšiřitelnosti softwaru.

Achilovou patou celé implementace je závislost na zastaralých/nepodporovaných SW/HW řešeních. Nešťastná je potřeba užití portu USB 2.0 (k roku 2020 je připravována specifikace USB 4.0 a odstoupení od konetkoru USB A), stejně pak i závislosti na eye-trackingových brýlích a softwaru od zaniklé firmy SMI (bez podpory pro Windows 10). Příprava vývojového prostředí na platformě Linux, spolu s emulátorem Wine, je pak s podivem. Autorka uvádí ukázky kódu/terminálů/XML jako různé škálované obrázky s různými fonty a dlouhými řádky, což ubírá na čitelnosti a grafické úpravě práce.

Autorka v implementaci serveru (kap. 5.3.4) zachází do detailů importování Pythonových knihoven; toto ale samo, bez další deskripce, člověku neznalému Pythonového programování moc neřekne. Implementaci eye-tracking serveru (obr. 25), který komunikuje s eye-trackingovým SDK (obr. 18) by bylo příhodné upravit tak, aby tato implementace/komunikace vycházela z obecného interface - aby tak šlo implementovat i jiná eye-tracking zařízení než SMI RED-250 (je škoda, že toto autorka neuvažuje, ani v diskuzi). Krom textového výčtu komunikace mezi serverem a klientem v kap. 5.4 a 5.5 by bylo vhodné uvést i diagram této síťové komunikace, pro lepší přehled o uvedeném.

Kapitola o testování implementace demonstruje vyhotovenou funkcionalitu. Čímž je splněno zadání práce. Nicméně, chybí verifikace algoritmické efektivity předloženého Python kódu.

### **Otázky k obhajobě:**

- Autorka uvádí, že *"aplikace bude programovaná specificky pro zařízení SMI RED-250 (v optimálním případě by samozřejmě byla aplikace univerzální pro všechny eye trackingová zařízení, to však není technicky možné – v kódu jsou volány specifické funkce definované v SDK (Software Development Kit) daného zařízení)"* (kap. 4.2). Autorka dále uvádí, že *"[SMI RED-250] Dalším důvodem výběru tohoto zařízení je fakt, že disponuje velmi robustním SDK, jehož součástí je iView X API, které usnadňuje komunikaci mezi vlastní eye trackingovou aplikací a SMI eye trackingovým zařízením."* Jak byste řešila implementaci pro jiné eye-trackery, a nakolik je vámi vyhotovený kód pro tyto reužitelný?
- Řešený klient je veden na úrovni příkazové řádky. Dovedete uvést jiné případné klienty, vč. GUI a předkládaných stimulů/interaktivity (kupř. Processing, 2D/3D engine), které by mohly figurovat v roli klienta? Jak byste takovou implementaci řešila?
- Jak jste ověřovala algoritmickou efektivitu vašeho kódu? Resp. na základě jakých Python nástrojů toto bude do budoucna možné?

### **Celkové hodnocení práce:**

Předložená práce působí uceleným dojmem, a vychází z konzistentní eye-trackingové teorie. Nemá cenu kritizovat to, co bylo mimo autorčin vliv (chybějící implementace na straně programátorů Hypothesis, neutěšená situace či podpora na poli eye-tracking SDK/výrobců). Práce splňuje zadání. Některé části implementace však působí ne úplně dokončeným dojmem, což je i důvodem pro zvolené hodnocení.

**Práci doporučuji k obhajobě s hodnocením velmi dobře (B) až dobře (C), dle obhajoby.**

Datum 17. 8. 2020

Podpis oponenta diplomové práce