

Zápočtová úloha z předmětu KIV/ZSWI

DOKUMENT SPECIFIKACE POŽADAVKŮ

28. března 2017

Tým: Carel

Členové:

Kateřina Kopřivová

kcermak@students.zcu.cz

Jakub Šantora

santoraj93@gmail.com

Valentin Horáček

valentin.horacek@gmail.com

Eye tracker

DOKUMENT SPECIFIKACE POŽADAVKŮ

pro Software umožňující výběr předmětu na monitoru

Verze 3.0

Historie dokumentu

Datum	Verze	Popis	Autor
19. 3. 2017	1.0	Úvodní popis	Kateřina Kopřivová
25. 3. 2017	1.1	Sepsání specifikace	Kateřina Kopřivová
27. 3. 2017	2.0	Úprava specifikace podle zadavatele	Kateřina Kopřivová
4. 4. 2017	3.0		Kateřina Kopřivová

Obsah

1. Úvod.....	1
1.1 Předmět specifikace	1
1.2 Typografické konvence.....	1
1.3 Cílové publikum, návod ke čtení	1
1.4 Rozsah projektu.....	1
1.5 Odkazy	1
2. Obecný popis	2
2.1 Kontext systému.....	2
2.2 Funkce produktu.....	2
2.3 Třídy uživatelů	2
2.4 Provozní prostředí	2
2.5 Omezení návrhu a implementace	2
2.6 Uživatelská dokumentace	2
2.7 Předpoklady a závislosti	2
2.8 Zdrojové kódy aplikace.....	2
3. Funkce systému	3
3.1 Nastavení kamery	3
3.2 Kalibrace kamery	3
3.3 Načtení nových obrázků do aplikace.....	3
3.4 Výběr obrázků z galerie	4
3.5 Úprava pořadí obrázku v mřížce.....	4
3.6 Vykreslení mřížky.	5
3.7 Výběr pole v mřížce	5
4. Požadavky na vnější rozhraní	6
4.1 Uživatelská rozhraní	6
4.2 Hardwarová rozhraní.....	7
4.3 Softwarová rozhraní	7
4.4 Komunikační rozhraní.....	7
5. Další parametrické (mimofunkční) požadavky	8
5.1 Výkonnostní požadavky.....	8
5.2 Bezpečnostní požadavky	8
5.3 Kvalitativní parametry	8
6. Ostatní požadavky.....	8
6.1 Testování	8

1. Úvod

1.1 Předmět specifikace

Tato specifikace popisuje software pro Eye Tracker umožňující výběr předmětu na monitoru vytvářeného v rámci předmětu KIV/ZSWI.

1.2 Typografické konvence

Pro odlišení významných prvků v textu jsou použity následující konvence:

- Názvy komponent grafického uživatelského rozhraní jsou psány kurzívou

1.3 Cílové publikum, návod ke čtení

Tato specifikace je určena pro zadavatele projektu a vývojový tým projektu. Tento dokument poskytuje základní popis toho, co má systém dělat a jak bude vypadat.

1.4 Rozsah projektu

Hlavním účelem programu je vybrat jeden z obrázků na monitoru pouze snímáním pohybu zorničky. Obrázky představují činnosti nebo potřeby člověka. Projekt má do budoucna za úkol pomoci lidem bez možnosti pohybu v komunikaci s okolním světem.

Samotný Eye Tracker by měl reagovat pouze na pohyb oka a na tlačítko pro ukončení aplikace. Uživatelské rozhraní aplikace je blíže popsáno v kapitole 4.1.

1.5 Odkazy

Veškeré informace k softwaru Pupil jsou k dispozici na uložišti GitHub:

Moritz Kassner, William Patera, Pupil Github Repository - <https://github.com/pupil-labs/pupil>

2. Obecný popis

2.1 Kontext systému

Tento systém je založen na open-source platformě Pupil. Pupil je software určený ke sledování a nahrávání pohybu zorničky v předem definované oblasti jednoznačně určené kalibrační kamerou. Systém obsahuje jednoduché uživatelské rozhraní pro nastavení, kalibraci a výběr obrázků, a komponentu pro vybrání konkrétního obrázku z matice podle zaměření zorničky uživatele.

2.2 Funkce produktu

- Grafické uživatelské rozhraní
- Výběr velikosti obrázkové matice
- Výběr přednastavených nebo vlastních obrázků
- Vyhodnocení cílového obrázku podle souřadnic pohledu očí

2.3 Třídy uživatelů

Produkt je určen pro uživatele s tělesným postižením. Tito uživatelé mohou potřebovat pomoc se spuštěním a kalibrací aplikace, v závislosti na jejich imobilitě. Je nutné přečtení uživatelské dokumentace. Ovládání aplikace nevyžaduje žádné specifické dovednosti.

2.4 Provozní prostředí

Notebook MSI GE72 2QC Apache, 8GB RAM, Intel Core i7 – 5700HQ CPU, frekvence 2.7 GHz.
Operační systém Ubuntu 16.04.2 LTS (Xenial).

2.5 Omezení návrhu a implementace

Systém nelze vyvíjet pod OS Windows z důvodu nekompatibility knihoven s operačním systémem. Výsledná aplikace bude spustitelná jak na OS Linux tak OS Windows.

2.6 Uživatelská dokumentace

S projektem bude dodáván uživatelský manuál s popisem funkcí a návodem k použití. Ve spuštěné aplikaci bude tlačítko pro nápovědu k ovládání aplikace.

2.7 Předpoklady a závislosti

Aby nedošlo ke kolizím našeho systému se softwarem Pupil a jeho budoucími aktualizacemi, rozhodli jsme se použít stabilní verzi Pupilu z data 17. března 2017 a neaktualizovat ji.

2.8 Zdrojové kódy aplikace

Zdrojové kódy aplikace jsou k dispozici ke stažení na úložišti GitHub.
github.com/teamCarel/EyeTracker

3. Funkce systému

3.1 Nastavení kamery

3.1.1 Popis a priorita

Funkce zobrazí pohled kamery, která snímá oko, a její možná nastavení jako obraz kamery, oblast zájmu a algoritmus. Celá tato funkce je převzatá od Pupil.

Priorita vysoká.

3.1.2 Události a odpovědi

Funkce se spustí tlačítkem *Camera Settings* v grafickém uživatelském rozhraní. Proces nastavení kamery lze spustit opakovaně, výsledný stav je trvalý a v dalším běhu programu jej nelze změnit.

Více informací lze nalézt v dokumentaci Pupilu.

3.2 Kalibrace kamery

3.2.1 Popis a priorita

Funkce zobrazí kalibrační proces přes celou plochu monitoru. Uživatel se dívá na jednotlivé kalibrační body. Funkce snímá jednou kamerou pohyb oka a zaměřenou oblast, druhou kamerou sleduje okolní svět (oblast, kterou před sebou uživatel skutečně má). Sjednocením kalibračních bodů s mapovanými koordináty se dosáhne optimální kalibrace.

Priorita vysoká.

3.2.2 Události a odpovědi

Funkce se spustí tlačítkem *Calibration* v grafickém uživatelském rozhraní. Proces kalibrace kamery lze spustit opakovaně, výsledný stav je trvalý a v dalším běhu programu jej nelze změnit.

Více informací lze nalézt v dokumentaci Pupilu.

3.3 Načtení nových obrázků do aplikace

3.3.1 Popis a priorita

Funkce otevře průzkumníka a uživatelem vybrané obrázky přidá do obrázkové galerie.


Priorita střední.

3.3.2 Události a odpovědi

- Uživatel vybere jeden nebo více obrázků – vybrané obrázky se zobrazí v galerii
- Uživatel nevybere žádné obrázky – galerie zůstává ve stejném stavu, nic se nekládá
- Uživatel vloží nepodporovaný formát – špatné soubory se do galerie nevloží a zobrazí se chybová hláška o nekompatibilitě souborů.


3.4 Výběr obrázků z galerie

3.4.1 Popis a priorita

Pro spuštění aplikace je nutné vybrat stejný počet obrázků jako je obsah mřížky (pro mřížku 3x3 je nutno vybrat 9 obrázků). Vybraný obrázek se od ostatních liší symbolem  v levém dolním rohu obrázku. Nad obrázkovou galerií se nachází počítadlo zvolených obrázků.

Priorita střední.

3.4.2 Události a odpovědi

- Uživatel klikne na obrázek, který není vybrán – obrázek se označí symbolem  a započítá se do celkového počtu vybraných obrázků.
- Uživatel klikne na obrázek, který je vybrán – označení obrázku v levém dolním rohu zmizí a počítadlo celkového počtu vybraných obrázků se zmenší o 1.

3.5 Úprava pořadí obrázku v mřížce

3.5.1 Popis a priorita

Funkce otevře okno s náhledem na mřížku. Pole v mřížce lze uspořádat podle preferencí uživatele.

Priorita střední.

3.5.2 Události a odpovědi

Pole v mřížce lze přemístit tažením myši na požadované místo. Obrázek, který je tažený myší, se vymění s obrázkem, který je na požadovaném místě. Uspořádání lze uložit tlačítkem *Confirm*.

3.6 Vykreslení mřížky.

3.6.1 Popis a priorita

Funkce z obrázků vybraných v galerii vytvoří mřížku požadované velikosti. Pozice obrázků jsou vybrány náhodně.

Priorita vysoká.

3.6.2 Události a odpovědi

Funkce se spouští tlačítkem Run v hlavní nabídce aplikace.

- Uživatel se pokusí spustit aplikaci se správným počtem zvolených obrázků – přes celou plochu monitoru se vykreslí mřížka s obrázky a spustí se sledování zorničky.
- Uživatel se pokusí spustit aplikaci s méně nebo více zvolenými obrázky – aplikace se nespustí a vyskočí chybová hláška.

3.7 Výběr pole v mřížce

3.7.1 Popis a priorita

Funkce sleduje zorničku a podle sledované oblasti vybere příslušnou sekci a zvýrazní obrázek v sekci. Funkce vybere obrázek, pokud se oko zaměří na jednu oblast v časovém intervalu třiceti sekund. Zvýrazní se ta oblast, na kterou se oko zaměřilo nejvíce. Zvýraznění bude trvat deset sekund, po té se mřížka vrátí zpět do původního stavu a znovu se spustí sledování oka. Tato akce se opakuje, dokud nedojde k vypnutí aplikace nebo návratu do menu.

Všechny časové intervaly budou vhodně upraveny během testování.

Priorita vysoká.

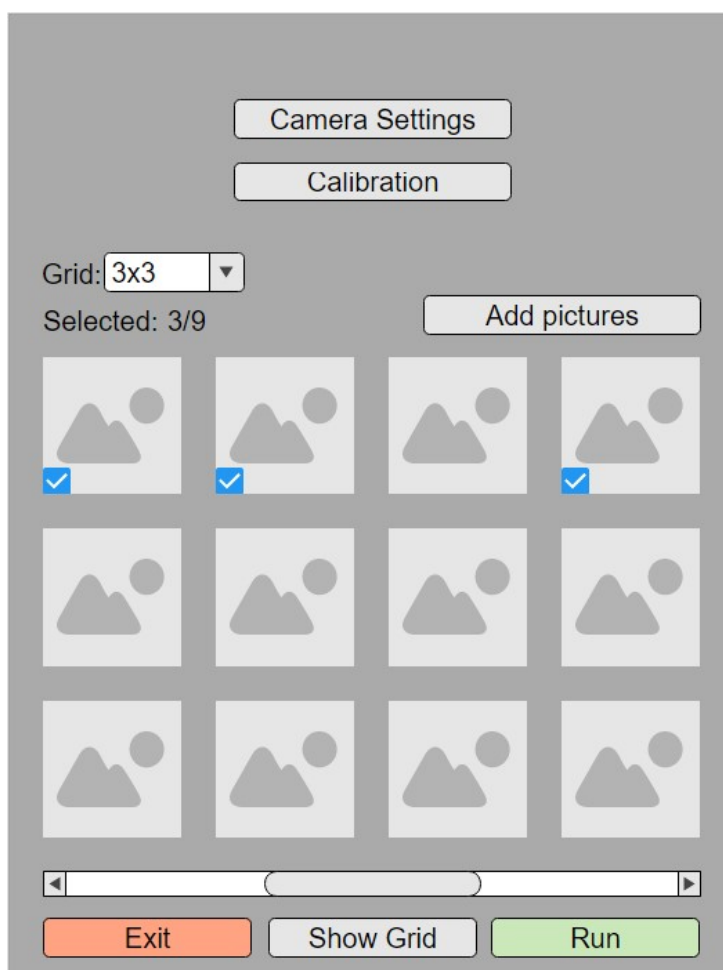
3.7.2 Události a odpovědi

V pravém horním menu bude možnost otevření možností, které podle výběru uživatele aplikaci vrátí do hlavního menu nebo celý proces ukončí. Více informací naleznete v části 4.1 Uživatelské rozhraní.

4. Požadavky na vnější rozhraní

4.1 Uživatelská rozhraní

Aplikace používá vlastní uživatelské rozhraní vytvořené knihovnou Tkinter v programovacím jazyce Python. Návrh spouštěcího okna je na obrázku 1. Rozhraní se skládá z několika tlačítek a obrázkové galerie.



Obrázek 1 Návrh spouštěcího okna

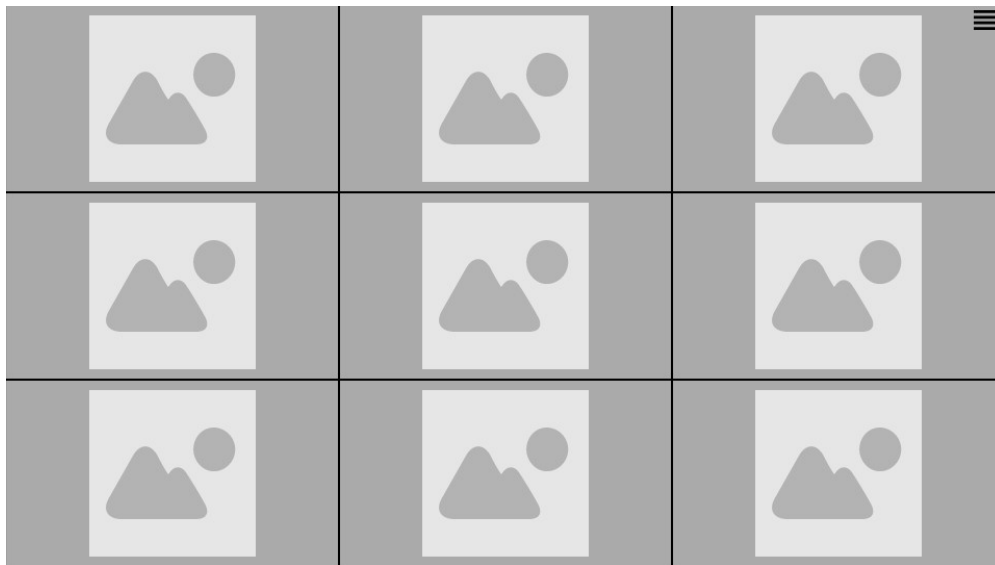
Tlačítko *Camera Setting* otevře okno s přenosem kamery zaměřené na sledované oko. Slouží k nastavení zaměření na zorničku oka. Tlačítko *Calibration* spustí kalibraci kamery a softwaru na oblast monitoru na kterém aplikace běží. V rozbalovacím menu lze vybrat velikost mřížky s obrázky.

Tlačítko *Add Pictures* otevře průzkumníka a umožní nahrání vlastních obrázků do aplikace. Největší prostor okna zabírá obrázková galerie, ve které je nutné vybrat přesný počet obrázků v mřížce.

Tlačítko *Show Grid* otevře okno s náhledem na mřížku, ve kterém lze obrázky přemísťovat do požadovaného uspořádání. Obrázky lze přemísťovat tažením myši. Obrázek, který je tažený myší, se vymění s obrázkem, který je na požadovaném místě.

Tlačítkem *Run* se spustí obrázková mřížka, před spuštěním je nutné zařízení zkalibrovat. *Exit* celou aplikaci vypne. V pravém horním rohu se nachází tlačítko pro nápovědu.

Na obrazovce s mřížkou s obrázky je také malé rolovací menu s možnostmi pro ukončení aplikace a návrat do menu. Návrh zobrazení je na obrázku 2.



Obrázek 2 Návrh obrazovky s mřížkou obrázků

4.2 Hardwarová rozhraní

Pro spojení kamer Eye Trackeru se softwarem slouží rozhraní USB 3.0.

4.3 Softwarová rozhraní

Pro komunikaci s aplikací Pupil využíváme její IPC (Inter process communication) Backbone. Pomocí PUBSUB Proxy Python modulu ZeroMQ, sloužícímu k asynchronnímu předávání zpráv získáváme data z aplikace pupil a odesíláme jí příkazy (například pro spuštění kalibrace atp.).

IPC Backbone má dvě síťové adresy, jejichž porty jsou náhodně voleny při startu a známy všem komponentám aplikace - PUB, umožňující odesílání zpráv a SUB, umožňující čtení vybraných zpráv ze streamu.

Všechny zprávy jsou složeny ze dvou rámců, první obsahuje řetězec nazývaný topic (téma), určující komu je zpráva určena, a druhý samotnou zprávu, která je dvojicí klíč - hodnota, zakódovanými pomocí modulu Python modulu msgpack.

4.4 Komunikační rozhraní

Není.

5. Další parametrické (mimofunkční) požadavky

5.1 Výkonnostní požadavky

Pro správný běh program je doporučena dedikovaná grafická karta, procesor s frekvencí větší než 2GHz a s minimálně třemi jádry, kvůli paralelismu vláken programu.

Je možné, že aplikace bude spustitelná i na počítači s horším vybavením. Jelikož nemáme možnost otestovat aplikaci na takovém stroji, nezaručujeme její funkčnost.

Spustitelná verze aplikace poběží pod OS Windows, developerská verze spustitelná ze zdrojových kódu bude spustitelná pouze pod OS Linux.

5.2 Bezpečnostní požadavky

Žádné.

5.3 Kvalitativní parametry

- nízký počet havárií během používání
- přesnost sledování a správnost výběru obrázku
- rychlé naučení ovládání programu

6. Ostatní požadavky

6.1 Testování

Software bude úspěšně otestován na deseti zdravých lidech.