



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PROGRAMOVÁNÍ ROBOTY POMOCÍ 3D MYŠI

ROBOT PROGRAMMING USING A 3D MOUSE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ LUKÁČ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ZDENĚK MATERNA, Ph.D.

BRNO 2023

Zadání bakalářské práce



140500

Ústav: Ústav počítačové grafiky a multimédií (UPGM)
Student: **Lukáč Tomáš**
Program: Informační technologie
Specializace: Informační technologie
Název: **Programování robota pomocí 3D myši**
Kategorie: Uživatelská rozhraní
Akademický rok: 2022/23

Zadání:

1. Proveďte rešerši existujících řešení využívajících 3D vstupní zařízení pro interakci s robotem.
2. Seznamte se se systémem pro programování robotů ARCOR2.
3. Proveďte návrh integrace 3D myši do systému ARCOR2 tak, aby bylo umožněno ovládání robota a ukládání 3D pozic.
4. Návrh realizujte.
5. Proveďte uživatelské testování pro ověření použitelnosti.
6. Zdrojové kódy a dokumentaci publikujte na GitHubu.
7. Vytvořte video prezentující vaši práci, její cíle a výsledky.

Literatura:

- MATERNA Zdeněk, et al. Teleoperating Assistive Robots: A Novel User Interface Relying on Semi-Autonomy and 3D Environment Mapping. *Journal of Robotics and Mechatronics*, roč. 29, č. 2, 2017, s. 381-394. ISSN 0915-3942.
- Guerin, Kelleher R., et al. "Adjutant: A framework for flexible human-machine collaborative systems." *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014.

Při obhajobě semestrální části projektu je požadováno:

- Body 1-2.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Materna Zdeněk, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, prof. Dr. Ing.
Datum zadání: 1.11.2022
Termín pro odevzdání: 10.5.2023
Datum schválení: 31.10.2022

Abstrakt

Práca nadväzuje na projekt skupiny Robo@FIT s názvom "Test-it-off: robotizované offline testování produktů". Projekt je zameraný na vývoj programu umožňujúceho jednoduchú obsluhu robotického pracoviska. Implementované riešenie dovoľuje obsluhu cez tablet s použitím rozšírenej reality. Moja práca sa zaoberá rozšírením existujúceho riešenia o podporu programovania pomocou 3D myši. Zameraním aplikácie je urýchlenie vytvárania jednoduchých programov pre obsluhu robotického pracoviska. Podľa týchto požiadaviek prebehla implementácia cieľovej aplikácie a následne užívateľské testy. V testoch sa použitie postupu s 3D myšou ukázalo ako rýchlejšia voľba pre tvorbu jednoduchých programov, užívatelia taktiež vybrali túto možnosť ako intuitívnejšiu.

Abstract

This thesis expands the project of group Robo@FIT named "Test-it-off: robotizované offline testování produktů". The project is targeted at the development of a program, that allows a simple operation of a robot workstation. The implemented solution supports the operation of a workstation using a tablet device, which makes use of augmented reality. My thesis expands on this solution by adding support for robot operation using a 3D mouse. The goal of this project is an acceleration of simple program creation for robot workstation operation. With these goals in mind, the application has been implemented, and afterward, user testing had been carried out. The results of testing conclude, that the new method indeed is faster and also more intuitive, than the previous solution.

Klíčové slová

Zjednodušené programovanie robotov, interakcia človek-robot, ARCOR2, Test-it-off, UI, 3D myš, Docker

Keywords

Simplified programming of robots, human-robot interaction, ARCOR2, Test-it-off, UI, 3D mouse, Docker

Citácia

LUKÁČ, Tomáš. *Programování robota pomocí 3D myši*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Zdeněk Materna, Ph.D.

Programování robota pomocí 3D myši

Prehlásenie

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Zdenka Maternu Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Tomáš Lukáč
9. mája 2023

Podakovanie

Na tomto mieste by som rád poďakovali pánovi Ing. Zdenkovi Maternovi, Ph.D., za vedenie tejto práce, za cenné rady, vynaložený čas, odbornú pomoc, a tiež trpezlivosť a veľkú ochotu pri konzultáciách a spolupráci na platforme github.

Obsah

1	Úvod	4
2	ARCOR2	5
2.1	Architektúra	5
2.2	Popis práce so systémom pomocou tabletu	5
2.3	Komunikácia	7
2.4	Popis uloženia scény	8
2.5	Popis uloženia projektu	8
2.6	Preklad projektu	9
2.7	Architektúra docker	9
2.8	Pohyb robota v systéme	10
3	Použitie 3D myši	13
3.1	Popis zariadenia	13
3.2	3D myš ako ovládač robota	15
4	Návrh	18
4.1	Vytýčenie požiadaviek	18
4.2	Postupnosť krokov	18
4.3	Vytvorenie výberu	19
4.4	Formy interakcie so systémom	19
4.5	Prvotný návrh	19
4.6	Zmeny v návrhu	20
4.7	Finálny návrh	22
5	Implementácia	23
5.1	Komunikácia so serverom	23
5.2	Čítanie vstupov z 3D myši	24
5.3	Zvuková odozva	25
5.4	Predpoklady pre stav servera	25
5.5	Vrstvenie programu	25
5.6	Výber robota	27
5.7	Volenie akcie	29
5.8	Ovládanie robota 3D myšou	30
5.9	Pridanie akcie	32
5.10	Zaradenie akcie do kontextu programu	33
5.11	Prispôbenie pre architektúru Docker	34
5.12	Spustenie programu	35

6	Užívateľské testovanie	36
6.1	Teória užívateľského testovania	36
6.2	Popis vykonaného testu	37
7	Záver	42
	Literatúra	44

Zoznam obrázkov

2.1	Menu tabletovej aplikácie ARCOR2	6
2.2	Používateľ pracujúci s tabletovou verziou aplikácie	7
2.3	Ukážka pohybového widgetu	7
2.4	Popis ukládanie informácií v systéme popísaný uml diagramom	9
2.5	Porovnanie architektúr	10
2.6	Priama kinematika	11
2.7	Inverzná kinematika	11
3.1	Obrázok 3D myši s popisom ovládacích prvkov	13
3.2	Prenos vstupu z myši do 3D priestoru.	14
3.3	Použitý robot	16
3.4	Ilustrácie k projektu	17
4.1	Schéma pôvodného návrhu	20
4.2	Schéma návrhu po úpravách	22
5.1	Stavový automat zobrazujúci tok programu	27
5.2	Tvorba zoznamu robotov	28
5.3	Schéma vrstvy výberu robota	29
5.4	Schéma vrstvy výberu akcie	31
5.5	Ilustrácia reťazenia akcií	34
5.6	Ilustrácia pridania novej akcie	35

Kapitola 1

Úvod

Jedným z problémov pri automatizácii býva nedostatok kvalifikovaných pracovníkov schopných naprogramovať robotickú linku. Tento problém nie je až tak podstatný pri veľkých pracoviskách, ktoré je potrebné naprogramovať len raz, a následne roky vykonávajú rovnakú funkciu. V prípade, že sa ale jedná o malé alebo stredne veľké spoločnosti, rozpočet na nákup špecializovaných robotov nie je až tak veľký. Preto by bolo vhodné mať možnosť použiť jedno robotické pracovisko na viacej účelov. Práve takéto časté preprogramovanie pracoviska ale môže byť nákladné.

Projekt TIOROTP ako riešenie vyvinul systém ARCOR2¹. Tento systém umožňuje programovanie robotického pracoviska pomocou tabletu s rozšírenou realitou. Týmto spôsobom odstraňuje potrebu kvalifikovaného pracovníka pri vývoji programu a dovoľuje koncovému užívateľovi programovať bez jeho pomoci. Práca s tabletom umožňuje užívateľovi ucelený prístup k funkciám, ktoré ARCOR2 ponúka a zároveň jednoduché využite možnosti manuálneho nastavenia robota do koncovej polohy. Programovanie robotov, ktorí ale túto možnosť nemajú, môže byť o niečo náročnejšie.

Preto sa moja práca zaoberá vytvorením alternatívnej metódy k tabletu. Zameranie metódy je na rýchle vytváranie základného programu robotickej linky, ktorý môže byť následne dokončený s využitím tabletu ak to bude potrebné.

Práca obsahuje predstavenie existujúcej implementácie, prieskum existujúcich projektov používajúcich 3D myš, predstavenie návrhu a implementácie programu a na záver zhodnotenie využiteľnosti 3D myši na základe výsledkov užívateľského testovania.

¹<https://github.com/robofit/arcor2>

Kapitola 2

ARCOR2

Moja práca bola navrhnutá ako rozšírenie systému ARCOR2, ktorý považujem za nutné predstaviť. Budem sa ale zameriavať na zložky systému podstatné pre moje rozšírenie.

ARCOR2 je systém pre jednoduché programovanie kolaboratívnych robotov s použitým rozšírenej reality vyvinutej skupinou ROBO@FIT¹[8].

2.1 Architektúra

Systém ARCOR2 sa skladá z nezávislých služieb, ktoré tvoria serverovú časť, a užívateľských rozhraní v roli klientov. Hlavnou službou servera je služba ARServer, ktorá sprostredkováva komunikáciu medzi službami, užívateľmi a zariadeniami v systéme [8].

Predpokladá sa, že každé pracovisko má samostatnú inštanciu systému, alebo sa nepre-
rajuje s viacerými pracoviskami naraz. Jedna inštancia servera totiž uchováva informácie iba o jednom stave, s ktorým môžu užívatelia komunikovať.

2.2 Popis práce so systémom pomocou tabletu

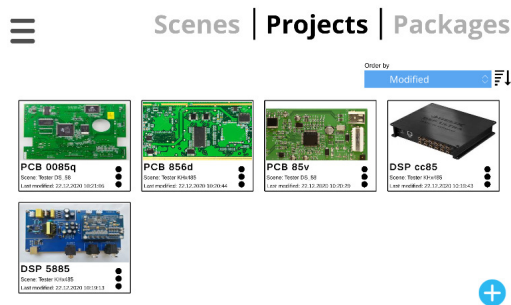
Pri používaní systému ARCOR2 sa užívateľ pripája k systému za pomoci aplikácie AREditor². Po zapnutí aplikácie užívateľ musí vyplniť ip servera, port a tiež meno, pod ktorým bude v systéme identifikovaný. Meno používateľa je použité na zamykanie objektov v systéme pri práci. Ak je server práve spustený a nedošlo k ďalšej manipulácii so systémom, tak sa užívateľovi zobrazí hlavné menu aplikácie. V danom momente nie je spustená úprava žiadneho projektu ani scény.

2.2.1 Kalibrácia

Po otvorení scény alebo projektu prebehne kalibrácia zariadenia. Užívateľ je vyzvaný na pohyb tabletu tak, aby zachytil čo najviac bodov na pracovisku. Užívateľ by sa mal zamerať na zachytenie značky, ktorá sa na pracovisku nachádza. Po úspešnej kalibrácii by sa na značke mala zobrazíť kalibračná kocka. Táto kocka oznamuje úspešnú kalibráciu a jej zvolením je možné kalibráciu opakovať.

¹<https://www.fit.vut.cz/research/group/robo/.en>

²https://github.com/robofit/arcor2_areditor



Obr. 2.1: Menu tabletovej aplikácie ARCOR2

2.2.2 Úprava scén

Užívateľ si môže vybrať scénu z existujúcich, alebo vytvoriť novú. Po otvorení scény prebehne kalibrácia. Užívateľ v móde úpravy scény môže pridávať na scénu nové objekty, ako sú napríklad roboty alebo prekážky. Užívateľ ovláda aplikáciu výberom z menu v ľavej časti obrazovky. Novo pridaný objekt je umiestnený na scénu v mieste, na ktoré je kamera tabletu natočená. Umiestnenie pridaného objektu je možné upravovať pomocou prídavného menu vpravo. Po ukončení úprav musí užívateľ scénu manuálne uložiť.

2.2.3 Úprava projektu

Užívateľ môže vybrať projekt z existujúcich alebo vytvoriť nový. Po otvorení projektu prebehne kalibrácia. Grafické rozhranie úpravy projektu a úpravy scény je rovnaké. Užívateľ programuje v systéme nasledovne:

- Zvolí si robota, ktorý má vykonať akciu.
- Prepne robota do módu učenia rukou.
- Posunie robota do želanej koncovkej pozície.
- Vytvorí nový akčný bod pomocou robota.
- Pridá akčnému bodu novú akciu.
- Vytvorí väzbu v programe, ktorá novú akciu napojí na koniec programu.

Alternatívne užívateľ môže akčný bod pridať samostatne a jeho pozíciu následne upraviť.

Po ukončení úprav musí užívateľ projekt manuálne uložiť.

2.2.4 Posun a rotácia

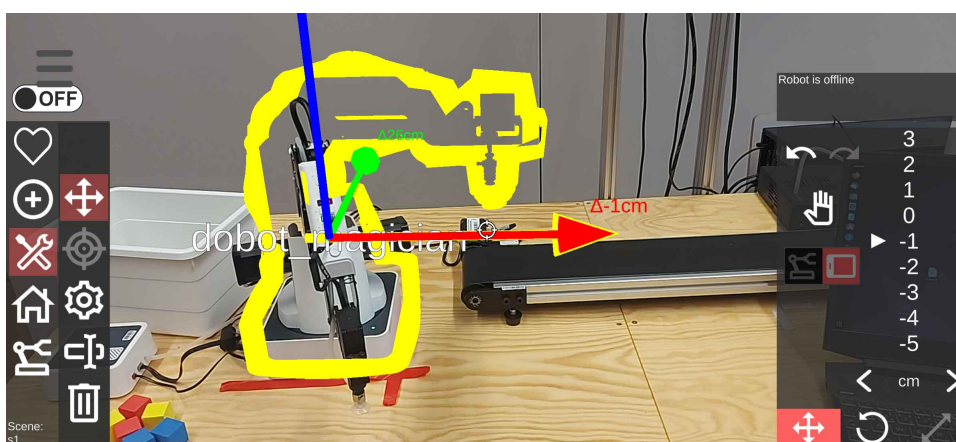
Pri pohybe objektu na scéne alebo v projekte užívateľ musí tento posun vykonať pomocou pomerne zložitej metódy. Po zvolení objektu sa na tablete zobrazí 3D pohybový widget.

Pohybový widget sa skladá z troch šípok vychádzajúcich zo zvoleného objektu. Každá zo šípok ukazuje pozdĺž inej súradnicovej osi. Daný dizajn funguje tak, že pri práci s 2D myšou v 3D priestore sa potiahnutím za šípku zvolený bod pohne v priestore v ose, ktorú šípka znázorňuje. Pri ovládaní tabletom ale smerovanie funguje o niečo rozdielnejším spôsobom.

Užívateľ musí namieriť tablet na šípku v smere, v ktorom chce pohyb vykonať. Na pravej strane obrazovky sú pod sebou napísané čísla. Užívateľ vykonáva pohyb v zvolenom smere



Obr. 2.2: Používateľ pracujúci s tabletovou verzou aplikácie



Obr. 2.3: Ukážka pohybového widgetu

posunom týchto čísel dohora alebo dole. Zvolený dizajn sa mal podobať mechanickému koliesku s hodnotami. Rovnaký postup funguje aj pri volení rotácie. Užívateľ si ale musí uvedomiť, okolo ktorej osi potrebuje vybraný objekt rotovať.

2.3 Komunikácia

Jednotlivé služby systému ARCOR2 medzi sebou komunikujú pomocou frameworku flask³.

Komunikácia medzi užívateľskými rozhraniami a službou ARServer prebieha cez web-sockets pomocou protokolu RPC (Remote procedure call) [8]. Protokol RPC volá funkciu zo vzdialeného serveru, akoby bola lokálna, a následne čaká na odpoveď [1]. V prípade, že by nebolo možné nadviazať spojenie, alebo by došlo k vypršaniu času na odpoveď, je vyhlásená výnimka. Táto funkcionality je implementovaná v triede ARServer.

³<https://flask.palletsprojects.com/en/2.3.x/>

2.3.1 Komunikácia server-klient

Predpokladá sa, že naraz môže byť na server napojených viacero užívateľov. Aby sa zachoval konzistentný stav, server posiela každému užívateľovi správy o vykonaných zmenách. Jednotlivé správy neobsahujú celý projekt, ale len informácie o konkrétnej zmene. Celý projekt je každému užívateľovi zaslaný len raz, pri otvorení projektu alebo prihlásení užívateľa. Preto každé užívateľské rozhranie musí spracovávať správy servera a upraviť lokálnu kópiu projektu tak, aby presne zodpovedala serverovej kópii.

2.3.2 Komunikácia klient-server

Užívateľ môže vykonávať zmeny v systéme zasielaním RPC správ serveru. Po zaslaní správy server posiela odpoveď, ktorá buď oznamuje úspešné vykonanie akcie, alebo zápornú odpoveď a chybovú správu.

2.4 Popis uloženia scény

Systém ARCOR2 si uchováva informácie o robotických pracoviskách v triede scéna. Trieda okrem základnej identifikácie obsahuje zoznam objektov.

2.4.1 Objekt v systéme

Trieda objekt obsahuje:

- `id` – `str`, unikátne id objektu
- `name` – `str`, názov objektu
- `type` – `str`, typ objektu
- `pose` – `Pose`, pozícia a rotácia objektu na scéne
- `parameters` – `list[Parameter]`, zoznam dodatočných parametrov

Informácie o typoch objektov a ich parametrov môže užívateľ získať pomocou správy `GetObjectTypes`. Pridávanie a úprava typov objektov je možné pomocou ďalších RPC.

Medzi objektami na scéne funguje dedičnosť a všetky objekty dedia z abstraktného typu `generic`.

Medzi špeciálne objekty, ktoré možno pridať na scénu patria napríklad objekt kamera a objekty prekážok, ktoré majú roboty obchádzať.

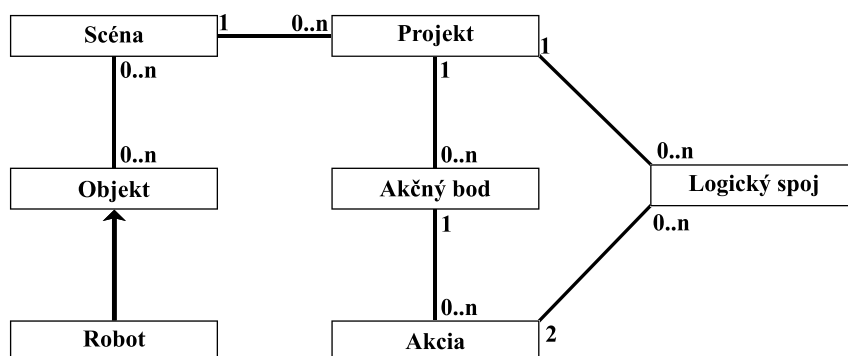
2.5 Popis uloženia projektu

Projekt v systéme ARCOR2 si uchováva základnú identifikáciu projektu a odkaz na scénu, na ktorú sa projekt vzťahuje. Informácie o robotoch a ich akciách sú čerpané zo scény.

2.5.1 Akčné body a akcie

Akcie v projekte sú uložené v akčných bodoch. Jeden bod môže obsahovať 0 až n akcií. Každý akčný bod môže mať priradeného príslušného robota a mať uložené informácie o postavení robota. Akcia si okrem identifikácie uchováva informácie o type akcie a robotovi,

ktorý akciu vykonáva. Zároveň má uložené ďalšie parametre akcie, ak sú potrebné. Na zoradenie akcií v programe slúži logika, ktorá je uložená zvlášť. Podrobnejšie informácie sú v sekcii implementácia.



Obr. 2.4: Popis ukládanie informácií v systéme popísaný uml diagramom

2.6 Preklad projektu

Užívateľom napísaný program je preložený do Python skriptu. Vygenerovaný skript bežne spúšťa vykonávacia služba, ale môže byť spustený aj samostatne.

2.7 Architektúra docker

Systém je nasadený v samostatných Docker kontajneroch. Docker je platforma, ktorá zjednodušuje proces vývoja a distribúcie aplikácií [9]. Aplikácie vytvorené pomocou platformy Docker sú zabalené spolu so všetkými závislosťami do samostatných balíčkov nazývaných kontajnery. Jednotlivé kontajnery sú izolovane spustené na kernely operačného systému. Táto technológia nahrádza potrebu spustenia jednotlivých aplikácií pomocou virtuálneho stroja.

2.7.1 Porovnanie architektúr Docker a virtuálneho stroja

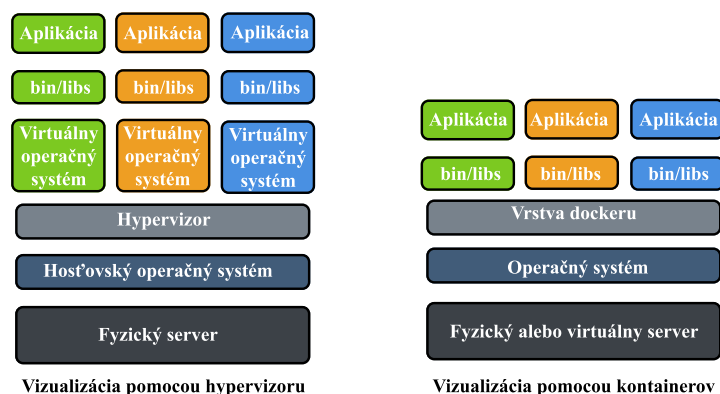
Výhody architektúry Docker sú nasledovné:

Rýchlosť

Kontajnery v architektúre Docker sú menšie, preto ich výroba je rýchlejšia a následne aj vývoj, testovanie a nasadenie je urýchlené.

Prenosnosť

Aplikácie vyrobené pomocou technológie Docker môžu byť prenesené ako jeden objekt bez ovplyvnenia výkonu.



Obr. 2.5: Porovnanie architektúr

Škálovateľnosť

Docker môže byť nasadený na fyzických, dátových alebo cloudových platformách. Kontajnery môžu byť rýchlo upravované a škála projektu môže byť ľahko upravovaná.

Rýchle nasadenie

Formát Dockerových kontajnerov je štandardizovaný, takže je možné ľahšie rozdeliť zodpovednosť vývojárov. Kontajnery majú všetky závislosti pribalené a otestované a môže byť nasadené do rôznych prostredí.

Hustota

Docker nepoužíva hypervizor (technika používaná pri spúšťaní virtuálnych strojov), takže môže lepšie rozdeliť potrebné zdroje. Docker môže spustiť viac kontajnerov naraz a ich výkon bude lepší.

Architektúra Docker má ale aj niekoľko nevýhod:

- Neposkytuje kompletnú virtualizáciu.
- Funguje len na 64bitových architektúrach.
- Na strojoch s Windowsom alebo Macoch je potrebná plná virtualizácia.
- Problémy so zaručením bezpečnosti.

2.8 Pohyb robota v systéme

Systém ARCOR2 a existujúce užívateľské rozhranie pre komunikácie s ním je navrhnuté tak, aby bolo možné čo najľahšie využiť možnosť robota manuálne ho posunúť do koncovkej polohy. Takáto manipulácia s robotom sa volá "hand teaching", v preklade učenie rukou.

V prípade, že robot túto možnosť manipulácie nepodporuje alebo sa užívateľ rozhodne ju nevyužiť, systém ponúka alternatívne možnosti ovládania robota. Jedna z možností je zaslание správy `MoveToPose` s argumentom špecifikujúcim cieľovú polohu a rotáciu zvoleného efektoru. Systém sa potom pokúsi vypočítať jednotlivé rotácie kĺbov robota pomocou

inverznej kinematiky. Ďalšie možnosti manipulácie zahŕňajú poslanie správy s argumentom, ktorý špecifikuje konkrétny akčný bod v priestore, alebo krokovanie.

2.8.1 Inverzná kinematika

Kinematika popisuje pohyb a rotáciu objektov a skupín objektov v priestore, bez zohľadňovania príčin pohybu alebo vlastností objektov. V danom prípade použijeme kinematiku na popis pohybu robotického ramena z počiatočného do cieľového bodu. [2, 3]

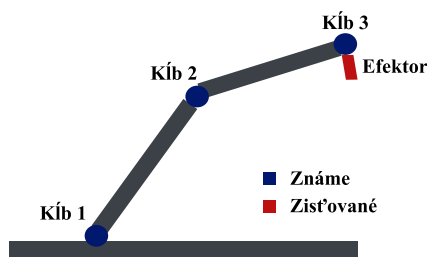
Kinematický popis sa delí na dva typy priamu a inverznú.

Priamy kinematický popis

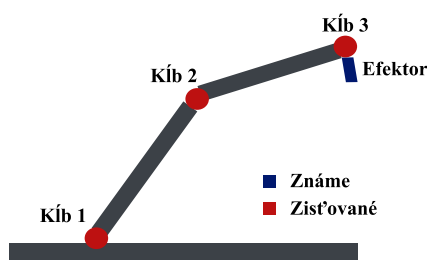
Priama kinematika počíta pozíciu a rotáciu koncového efektoru robota z polohy robota a rotácií jednotlivých kĺbov robotického ramena.

Inverzný kinematický popis

Inverzná kinematika bola založená v robotike ako riešenie problému pre posun robotického ramena s určeným počtom stupňov voľnosti do zadaného bodu a rotácie v priestore.



Obr. 2.6: Priama kinematika



Obr. 2.7: Inverzná kinematika

2.8.2 Popis aktívneho robotického pracoviska

Robotické pracovisko napojené na ARCOR2 na fakulte informatiky ponúka užívateľom spolu tri rôzne funkčné mechanické objekty:

- Robot typu Dobot Magician
- Robot typu Dobot M1

- dopravný pás

Dopravný pás

Dopravný pás v systéme nie je rozoznaný ako robot. Hoci mu boli systémom priradené akcie, ktoré pás dokáže vykonať, tieto akcie si vyžadujú väčšie množstvo manuálnej manipulácie s parametrami. Preto manipulácia s dopravníkom nie je možná pomocou 3D myši, a ani nebol zámer podporu dopravníku a iných nie robotických objektov dodať. V prípade použitia dopravníku, alebo iných objektov, ktoré sú schopné vykonávať funkcie, užívateľ musí dané akcie pridať pomocou tabletu.

Kapitola 3

Použitie 3D myši

Pri čítaní vstupu od užívateľa je použitá výhradne 3D myš. Model použitý pri vývoji je model SpaceNavigator od spoločnosti 3DConnexion ¹. Zariadenia podobného typu sú používané prevažne na rýchlu a intuitívnu orientáciu kamery pri práci s editormi 3D objektov, ako je napríklad Blender alebo CAD aplikácie.

3.1 Popis zariadenia

Zariadenie 3D myš sa skladá z joysticku, schopného snímať pohyb a dvoch programovateľných tlačidiel. Tlačidlá označujem ako ľavé a pravé pre pohľade na zariadenia zo strany, kde sa nachádza nápis.

3.1.1 Popis vstupu

Joystick 3D myši je schopný snímať pohyb v šiestich smeroch. Táto vlastnosť sa nazýva "Six degrees of freedom".

Six degrees of freedom (6DoF)

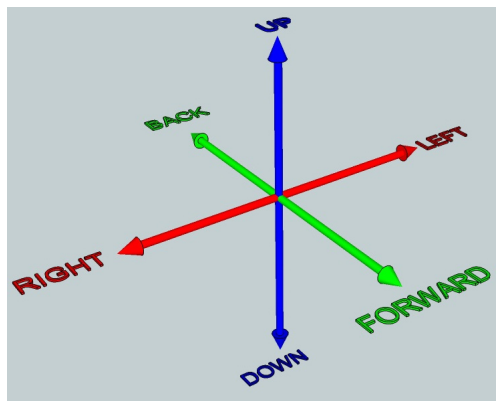
Pod pojmom "Six degrees of freedom" rozumieme možnosti pohybu a rotácie telesa v trojdimenzionálnom priestore.

Tri stupne voľnosti označujú možnosť pohybu telesa pozdĺž súradnicových osí.

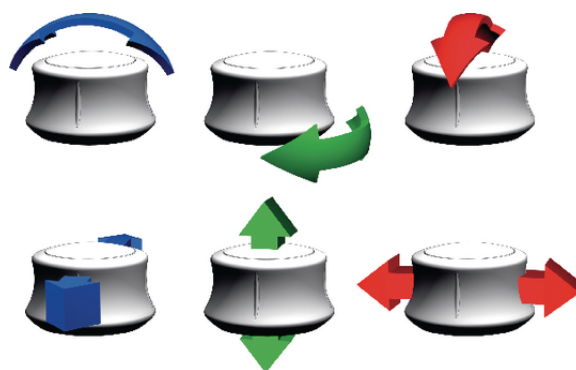
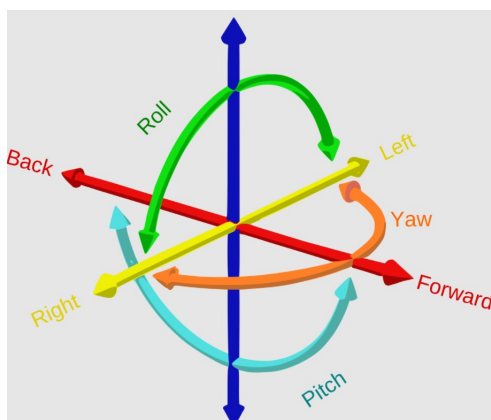
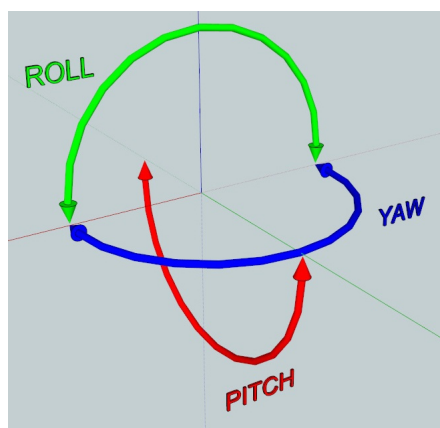
¹<https://3dconnexion.com/cz/product/spacemouse-compact/>



Obr. 3.1: Obrázok 3D myši s popisom ovládacích prvkov



Zvyšné tri stupne označujú možnosť rotácie objektu okolo súradnicových osí.



Obr. 3.2: Prenos vstupu z myši do 3D priestoru.

Načítaný vstup má teda podobu šiestich hodnôt. Tri hodnoty predstavujú posun myši, tri hodnoty predstavujú rotáciu.

3.1.2 Použitá knižnica

Čítanie vstupu z myši v jazyku python sprostredkováva knižnica `pyspacemouse`². Knižnica používa surový HID vstup.

²<https://pypi.org/project/pyspacemouse/>

3.2 3D myš ako ovládač robota

Použitie 3D myši na ovládanie robota nie je nová myšlienka. Joystick a podobné zariadenia sa už dlho používajú na ovládanie mechanických zariadení. Medzi projekty, ktoré využili 3D myš na operáciu robota patria napríklad tieto projekty:

- Towards Robot Avatars: Systems and Methods for Teleinteraction at Avatar
- Teleoperative assisting robots

3.2.1 Towards Robot Avatars: Systems and Methods for Teleinteraction at Avatar

Cieľom tohto projektu bolo pokračovať vo vývoji komunikácie na diaľku [6]. Autori sa snažili vyvinúť nástroj, ktorý by používateľovi umožnil prenos pohybu užívateľa ako rozšírenie možností videohovoru.

Opis postupu

Autori vybudovali mobilnú robotickú platformu, ktorú je možné diaľkovo ovládať. Robotická platforma pozostáva z:

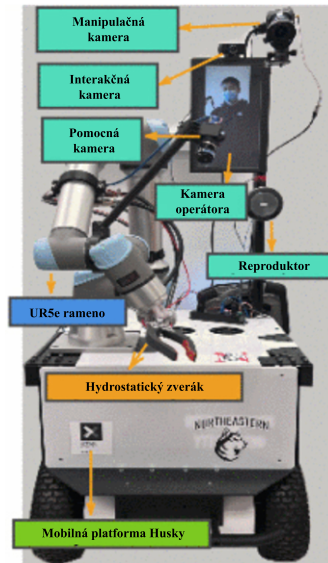
- Mobilná pásová platforma Husky
- Robotické rameno UR5e.
- Dve rôzne kamery kontrolujúce pohyb platformy
- Kameru sprostredkujúcu videohovor.
- Pár reproduktorov slúžiacich na zvukový výstup.
- Tabletové zariadenie slúžiace na ovládanie platformy a na výstup obrazovej časti videohovoru

Na ovládanie platformy užívateľ použil dva rôzne ovládače. Na ovládanie robotického ramena bol použitý špecializovaný hardware, ktorý mal priblížiť užívateľovi pocit pohybu ruky. Ako druhotný ovládač bola použitá 3D myš model SpaceNavigator.

Opis úlohy 3D myši v projekte

3D myš bola v projekte použitá na ovládanie mobilnej pásovej platformy Husky. Pásová platforma má obmedzené možnosti pohybu, ktoré zahŕňajú len pohyb dopredu/dozadu, a rotáciu doľava/doprava. Na pohyb platformy bola vybraná 3D myš pretože ponúka intuitívny spôsob vykonávania daných príkazov. Jej využitie v projekte je ale obmedzené, dalo by sa povedať, že sú využité len dva stupne voľnosti zo šiestich, ktoré myš ponúka.

Zdalo sa mi ale vhodné projekt zmieniť, hoci len stručne, pretože používa dva rôzne typy vstupu, ktoré je 3D myš schopná zaznamenať. Aj keď je ovládanie mobilnej platformy len podporná časť projektu, využitie 3D myši na daný účel je vhodné zaznamenať. Použitie obmedzenej časti funkcionality myši pri vybraných úlohách, v snahe spraviť určitý úkon v programe viac intuitívny pre užívateľa, je myšlienka, ktorú by som z tohto projektu rád vyzdvihol. V danom projekte bolo možné použiť aj iné zariadenia, ako príklad uvádzam jednoduché šípky na klávesnici, ktorými by bolo ovládanie platformy tiež možné, ale nie až tak intuitívne.



Obr. 3.3: Použitý robot

3.2.2 Teleoperating assistive robots

Projekt Teleoperating assistive robots: A novel user interface relying on semi-autonomy and 3D environment mapping mal za cieľ riešiť problém, že za súčasného stavu robotiky a umelej inteligencie nie je možné vytvoriť plne samostatného robota schopného vykonávať úlohy v domácnosti [7]. Projekt preto vytvoril riešenie, ktoré necháva robota riešiť úlohy prevažne samostatne, ale dovoľuje ľudskému neexpertnému užívateľovi diaľkovo ovládať robota, ak by sa robot dostal do situácie, ktorú sám nevie vyriešiť. Jedným z hlavných prínosov projektu bolo vytvorenie užívateľského rozhrania, ktoré mieša údaje zo senzorov robota s 3D vygenerovaním obrazom okolia a dovoľuje ľahkú a presnú diaľkovú prácu s robotom.

Projekt je navrhnutý tak, aby robota bolo možné ovládať pomocou bežného a lacného hardwaru, napríklad klávesnicou a myšou, ale bola pridaná aj alternatívna možnosť ovládania robota pomocou 3D myši a stereoskopického displeju.

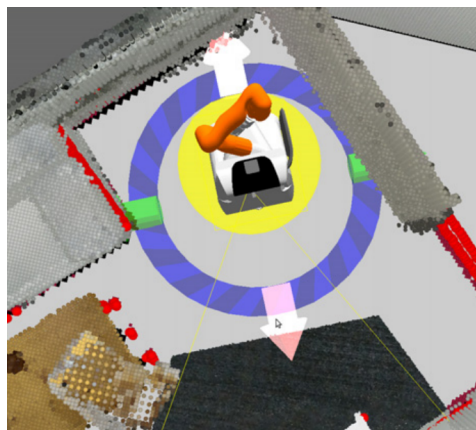
Popis ovládania pomocou 3D myši

Vyvinutý systém by mal byť použiteľný na viacero typov robotov, ale v príklade budem rátať s použitím robota, ktorý sa skladá z mobilnej platformy a jedného robotického ramena.

Operácia mobilnej platformy robota v priestore v projekte predstavovala hneď niekoľko prekážok. Robot v projekte je schopný snímať svoje okolie a vypočítať na základe vizuálneho vstupu 3D model okolia, ale výsledný model nemusí byť úplne presný. Aby nedochádzalo ku kolíziám kvôli nepresnému modelu, robot má určenú istú bezpečnostnú vzdialenosť, na ktorú sa nemôže priblížiť k prekážkam. Ak by došlo k situácii, keď robot nedokáže terén bezpečne navigovať sám, ľudský operátor môže robota ovládať manuálne. Ovládanie robotického ramena pomocou myši je možné s pomocou neúplného 3D gizmu, posunom ktorého je možné robota otáčať a hýbať do strán. Alternatívou k danému prístupu je použitie 3D myši na posun platformy. Ako jednu z výhod projekt špecifikuje možnosť súčasného prístupu k pohybu a rotáciám.



(a) Robot použitý v projekte



(b) UI navigácie robota

Obr. 3.4: Ilustrácie k projektu

Operácia robotického ramena prebieha posunom a rotáciou koncového bodu efektoru ramena. Túto akciu je možné vykonať buď pomocou klasickej myši a 3D gizma, alebo pomocou 3D myši. Projekt sa zameriava na bezpečnú manipuláciu s ramenom a dovoľuje si prezrieť simuláciu akcie pred jej vykonaním, prípadne ju meniť.

Daný prístup k ovládaniu robota je použitý aj v systéme ARCOR2. Ovládanie robotického ramena pomocou 3D myši by som vo svojom projekte rád implementoval podobným spôsobom. Pri programovaní sa, na rozdiel od daného projektu, predpokladá priama prítomnosť programátora pri robotickom pracovisku, a preto možnosť kontroly akcie pred jej vykonaním nie je potrebná.

Kapitola 4

Návrh

V tejto kapitole popíšem hlavné požiadavky na výslednú aplikáciu a predstavím prvotný návrh aplikácie. Ďalej popíšem problémy a ďalšie nedostatky, ktoré pôvodný návrh mal, a tiež zmeny, ktorými boli vyriešené. Kapitulu uzatvára popis výsledného návrhu.

4.1 Vytýčenie požiadaviek

Základné požiadavky na návrh sú:

- jednoduchosť ovládania
- rýchlosť programovania
- použitie zariadenia 3D myš na vstup
- minimálne použitie iných zariadení

Hlavnou požiadavkou na návrh je teda možnosť zjednodušenia a urýchlenia práce oproti prístupu s tabletom.

4.2 Postupnosť krokov

Prieskum programu ARCOR2 odhalil, že užívateľské akcie potrebné na vytváranie programu v systéme sú:

- Výber robota, ktorý má akciu vykonať
- Výber efektoru, s ktorým má robot akciu vykonať
- Výber akcie, ktorú má robot vykonať
- Výber fyzického miesta, kde sa akcia vykoná
- Výber miesta v programe, kedy má byť akcia pridaná

Ak užívateľ uskutoční tieto výbery, je možné pridať novú akciu.

4.3 Vytvorenie výberu

Základnou požiadavkou na užívateľský vstup je teda možnosť výberu objektu z množiny. Ako vhodné riešenie, ktoré nevyžaduje veľkú mentálnu záťaž, bolo vybraná možnosť vytvorenia virtuálneho zoznamu.

Užívateľ by mohol zoznamom prechádzať pohybom myši do strán. Pre potvrdenie výberu užívateľ stlačí ľavé tlačítko myši, rovnako ako pre potvrdenie akcie s normálnou myšou. Pre zrušenie výberu je potrebné stlačiť pravé tlačidlo myši. Niektoré výbery môžu ponúkať dodatočné informácie pre užívateľa. Tieto informácie obsahujú popis priložený k danej položke v zozname alebo ďalšiu formu odozvy pre užívateľa.

Pre vyžiadanie týchto informácií musí užívateľ pohnúť myšou od seba.

4.4 Formy interakcie so systémom

V posledných rokoch výskum možností interakcie s robotmi rapídne pokročil [10]. Roboty schopné pokročilých interakcií ale ešte nie sú veľmi dostupné. Najmä pre cieľovú skupinu používateľov systému ARCOR2, napríklad malých a stredných podnikateľov, by daná investícia nebola veľmi prínosná. Preto sa zariadenia použité na zachytenie vstupu od užívateľa obmedzujú na bežne dostupné a nenákladné zariadenia, ako sú tablet a 3D myš.

Okrem toho je ale tiež potrebná istá forma odozvy zo systému, napríklad pri výbere objektu zo zoznamu je potrebné, aby užívateľ mohol ľahko zistiť, ktorý objekt má momentálne zvolený. Najlepšou formou odozvy je vizuálna odozva, preferované je zobrazenie celého zoznamu naraz [4]. V programe ale nie je použitý monitor počítača ani iná forma obrazovky, preto zobrazenie celého zoznamu neprípadá do úvahy. Napriek tomu je ale použitá vizuálna forma odozvy a to v podobe priameho pohybu na scéne. Konkrétne, pri voľbe robota sa pri listovaní zoznamu robotov na scéne, zvolený robot pohne, aby upútal pozornosť užívateľa.

V miestach, kde vizuálna odozva nie je použiteľná, ako je napríklad pri výbere akcie, je namiesto toho použitá audio odozva.

4.5 Prvotný návrh

Návrh je štrukturovaný na vrstvy. V každej vrstve užívateľ uskutoční jeden výber. Ak užívateľ potvrdí posledný výber, akcia je pridaná do zoznamu a riadenie programu sa premiestni naspäť do prvého výberu. V každej vrstve má užívateľ možnosť zrušiť predchádzajúci výber a presunúť sa o vrstvu vyššie. Každá vrstva má zároveň inú formu odozvy.

V prvej vrstve si užívateľ vyberie robota. Odozva v tejto vrstve je len zvuková, na vytvorenie vizuálnej odozvy je potrebný výber efektoru.

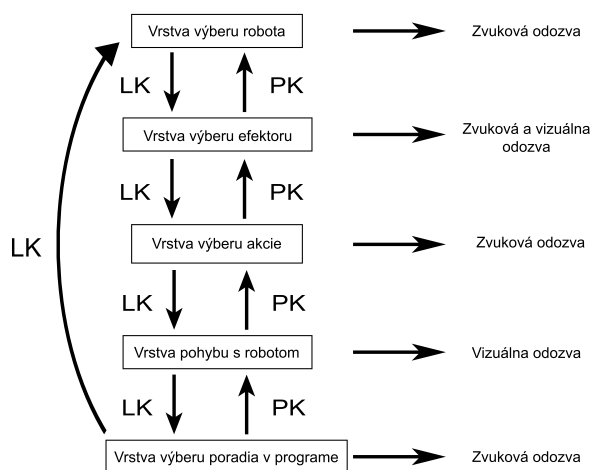
V druhej vrstve užívateľ vyberie konkrétny efektor daného robota. Odozva je vizuálna aj zvuková.

Tretia vrstva sa zaoberá výberom akcie. Odozva je len zvuková, vizuálna odozva v tejto vrstve nie je možná.

V štvrtej vrstve užívateľ vyberie miesto, kde sa má akcie vykonať. Užívateľ pomocou myši presunie efektor robota do koncovej pozície.

V poslednej vrstve je vybrané miesto v programe. Odozva je opäť len zvuková.

Po potvrdení výberu sa program presunie naspäť do prvej vrstvy.



Obr. 4.1: Schéma pôvodného návrhu

4.6 Zmeny v návrhu

Po začatí implementácie pôvodného návrhu ale vyšlo najavo niekoľko nedostatkov. V tejto sekcii popíšem nájdené problémy a nedostatky pôvodného návrhu, a tiež zmeny, ku ktorým to viedlo.

4.6.1 Úprava programu

V pôvodnom návrhu programu bolo možné výsledný program aj upravovať. Táto funkcionality ale vyžadovala vyššiu úroveň mentálnej záťaže, aby bolo možné si predstaviť, čo vykonávané úkony menia v programe.

Pridaná záťaž bola neúmeraná benefitom, ktoré by dané rozšírenie prinieslo. Okrem toho by táto funkcionality išla proti jednému z hlavných požiadaviek práce a to jednoduchosti.

Táto funkcionality preto do výsledného programu implementovaná nebola. Potrebnú úpravu programu je možné rýchlo a prehľadne vykonať pomocou tabletovej aplikácie, prípadne môže vytvoriť program znovu.

4.6.2 Výber efektoru

V pôvodnom návrhu je výber efektoru samostatná vrstva v programe. Ale väčšina robotov má len jeden efektor. Ak má robot viacero efektorov, ich počet nebýva veľký. Preto bol výber robota a výber efektoru zlúčený do jednej vrstvy vo výslednom programe. V špecifických prípadoch síce môže nastať situácia, že má robot väčší počet efektorov, ale tieto prípady nie sú veľmi bežné. Ani v týchto prípadoch nie je vzniknuté oneskorenie spôsobené listovaním rozšíreného zoznamu príliš veľké. Zároveň to posúva možnosť vizuálnej odozvy o vrstvu vyššie.

4.6.3 Výber robota

Vo vrstve výberu robota v pôvodnom návrhu bola predpokladaná odozva vizuálna aj odozva pohybom robota. Robot sa zároveň mal pohnúť do viacerých polôh, akoby zakrúžiť okolo

pôvodnej pozície, aby nevznikla situácia, kde sa robot pokúša pohnúť do polohy, do ktorej sa fyzicky nemôže pohnúť a žiadny pohyb vykonaný nie je.

Problém vzniká pri listovaní zoznamu, ktorý obsahuje väčší počet robotov alebo efektorov. Každá položka v zozname pri zvolení použije zvukovú odozvu, ktorá môže trvať až vyše sekundu. Okrem toho táto odozva nie je nevyhnutná, užívateľ môže fyzicky vidieť pohyb efektora.

Systém ARCOR2 zároveň oznamuje výsledok pokusu o pohyb robota, čiže aplikácia má odozvu o vykonaní pohybu. Na zachytenie pozornosti užívateľa zároveň postačuje jeden pohyb a návrat do pôvodnej pozície. Odstránením nadbytočných pohybov možno urýchliť prácu s programom.

Ak by ale došlo k prípadu, kde si užívateľ nevšimne pohyb robota, alebo robot neposkytne možnosť pohybu, užívateľ si môže prehrať audio odozvu posunutím myši od seba.

4.6.4 Výber miesta v programe

V pôvodnom návrhu je samostatná vrstva pre výber miesta v programe, kam sa má akcia pridať. Táto funkcionálna ale nie je až tak potrebná. Väčšina užívateľov totiž program vytvára chronologicky. Nová akcia je preto skoro vždy pridaná na koniec programu. Preto bola vrstva výberu miesta v programe odstránená a nová akcia je pridaná vždy na koniec.

Táto úprava môže spôsobovať niekoľko problémov. Napríklad v prípade, že so systémom pracuje niekoľko užívateľov naraz. Tieto problémy by ale mali byť ľahko riešiteľné a čas ušetrený zjednodušením programu by mal prevyšovať čas strávený opravou prípadných chýb.

4.6.5 Opakované použitie robota

Pri programovaní pracoviska je veľmi pravdepodobné, že používateľ zadá niekoľko akcií po sebe jednému robotovi. Ako príklad môže byť použitie akcií ako je zdvihnúť a položiť alebo opakované akcie pri navádzaní robota. Takéto opakovanie je väčšinou častejšie ako samostatná akcia jedného robota, najmä pri malých pracoviskách. Preto bol pôvodný návrh pozmenený a po pridaní novej akcie sa užívateľ nepresunie do vrstvy výberu robota, ale zostane zvolený rovnaký robot a užívateľ sa presunie do vrstvy výberu akcie.

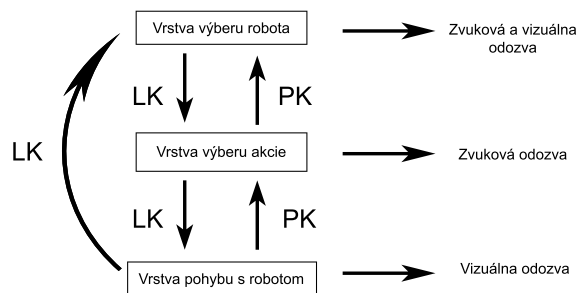
4.6.6 Pridanie čakacej vrstvy

V pôvodnom návrhu bol program rozdelený len na aktívne vrstvy. Úspešný chod programu ale vyžaduje určitý stav serveru. Užívateľ musí tento stav nastaviť pomocou tabletovej aplikácie.

Pôvodný návrh predpokladal zapnutie aplikácie až po nastavení serveru do požadovaného stavu. Ak by po zapnutí server v požadovanom stave nebol, aplikácia by sa vypla s chybovým hlásením.

Daný systém ale nebol veľmi efektívny, pri úprave viacerých projektov naraz opakované zapínanie a vypínanie aplikácie pôsobilo ťažkopádne.

Ako riešenie daného problému bola vytvorená tzv. čakacia vrstva. Program sa do tejto vrstvy prepne hneď po spustení, a zakaždým keď počas výkonu programu server opustí želaný stav. Program sa z tejto vrstvy automaticky presunie do vrstvy výberu robota, ak server prejde do potrebného stavu. Prechody medzi čakacou a inými vrstvami sú sprevádzané zvukovou odozvou.



Obr. 4.2: Schéma návrhu po úpravách

4.7 Finálny návrh

Program je vo finálnom návrhu stále štrukturovaný na vrstvy, ale ich počet bol zredukovaný. Úspešne sa podarilo odstrániť najmä vrstvy, ktoré sa spoliehali na zvukovú odozvu. Výsledné vrstvy sú teda nasledovné:

- Vrstva výberu robota a efektoru
- Vrstva výberu akcie
- Vrstva výberu miesta

Prvá vrstva má vizuálnu odozvu s možnosťou zvukovej odozvy. Druhej vrstve zostáva zvuková odozva. Posledná vrstva má opäť vizuálnu odozvu.

Vrstva výberu robota bola zlúčená s vrstvou výberu efektoru. Vrstva výberu miesta v programe bola odstránená, novo pridaná akcia bude pridaná na koniec programu.

Čakacia vrstva nie je na obrázku vykreslená, pretože sa nejedná o vrstvu, ktorá vyžaduje vstup od užívateľa.

Kapitola 5

Implementácia

V kapitole implementácia bližšie popíšem technickú stránku aplikácie. Postupne popíšem bližšie fungovanie komunikácie so serverom a myšou a tiež jednotlivé vrstvy programu.

5.1 Komunikácia so serverom

Priama komunikácia so serverom prostredníctvom RPC správ je sprostredkovaná triedou `ARServer`¹. Vymieňané správy spadajú do jednej z nasledovných kategórii.

- Zmena stavu servera
- Zamykanie objektov
- Zisťovanie informácii
- Pohyb robota
- Pridanie akcie

5.1.1 Zmena stavu servera

Do kategórie zmena stavu servera patria správy posielané serverom, ktoré informujú aplikáciu o zmenách prebiehajúcich na serveri. Tieto správy môžeme ďalej rozdeliť na správy, ktoré informujú o toku programu a správy, ktoré informujú o zmene scény alebo projektu.

Správy posielané serverom dedia z triedy `Event`, ktorá obsahuje:

- `event` – `str`, názov eventu
- `change_type` – `Type(enum)`, obsahuje typ vykonanej zmeny
- `parent_id` – `str`, id rodiča objektu ak je potrebné

Program nemá vyhradený proces na čítanie správ. Správy sa ukladajú do štruktúry radu a v programe sú vyhradené miesta, kde sa program zastaví a spracuje všetky správy. Okrem toho sa v programe nachádzajú miesta, kde program čaká na príchod konkrétnej správy. Akékoľvek iné správy zachytené v týchto miestach sú tiež spracované.

Správy informujúce o toku programu, ktoré sú pre program podstatné sa týkajú stavu projektu (otvorený alebo zatvorený) a stavu scény (zastavená alebo spustená).

¹https://github.com/robotfit/arcor2/blob/master/src/python/arcor2_rserver_data/client.py

5.1.2 Zamykanie objektov

Server je prispôsobený na prácu viacerých používateľov naraz. Aby nedošlo ku konfliktu dát, server implementoval mechaniku zamykania. Zamykanie funguje tak, že keď jeden užívateľ zamkne konkrétny objekt na scéne, ostatným užívateľom server zašle správu o zamknutí objektu. Pri snahe o interakciu so zamknutým objektom sa užívateľovi interakcia nepodarí a dostane chybovú správu.

5.1.3 Zisťovanie informácií

O zaslanie informácií žiada užívateľ pomocou použitia RPC správ. Pre program dôležité informácie sa týkajú zoznamu robotov na scéne a tiež akcií, ktoré jednotlivé roboty dokážu vykonávať.

5.1.4 Pohyb robota

Medzi správy o pohybe robota radím RPC správy, ktorými užívateľ žiada o zasielanie aktualizácií, ktoré sa týkajú vybraného robota, a správy, ktoré server po vyžiadaní zasiela užívateľovi. Mechanizmus zasielania správ o robotovi je použitý pri každom pohybe robota.

5.2 Čítanie vstupov z 3D myši

Čítanie vstupu z 3D myši prebieha pomocou knižnice `pyspacemouse`. Pri používaní tejto knižnice ale dochádza k unikátnemu problému. Pri spracovaní vstupov dochádza k ich hromadeniu v rade. To nie je problém, ak dochádza k čítaniam pravidelne v krátkych intervaloch, ale ak sú medzi čítaniami dlhé medzery, čítania prestávajú byť presné.

Spomínanú chybu uvediem na príklade. Nasledujúci kód číta a vypíše vstup z myši každé 2 sekundy. Pri prvom čítaní je myš v pokoji, v druhom je posunutá do krajnej polohy na jedenej z osí a v treťom je v pokoji. Prečítané údaje sú ale nasledovné:

```
SpaceNavigator(t=-1, x=0, y=0, z=0, roll=0, pitch=0, yaw=0, buttons=[0, 0])
SpaceNavigator(t=13004.930459407, x=0.0, y=-0.002857142857142857, z=0.0, roll=0, pitch=0, yaw=0, buttons=[0, 0])
SpaceNavigator(t=13006.93267579, x=0.0, y=-0.002857142857142857, z=0.0, roll=0.0, pitch=0.0, yaw=0.0, buttons=[0, 0])
```

V programe je daný problém vyriešený vytvorením separátneho čítacieho procesu pomocou triedy `Process` knižnice `Multiprocessing`. Výmena hodnôt s procesom prebieha pomocou triedy `Queue` z rovnakej knižnice. Upravený program a výstup sú nasledovné:

```
SpaceNavigator(t=0, x=0.0, y=0.0, z=0.0, roll=0.0, pitch=0.0, yaw=0.0, buttons=[0, 0])
SpaceNavigator(t=0, x=0.03142857179045677, y=1.0, z=-0.05428571254014969, roll=0.0, pitch=0.0, yaw=0.0, buttons=[0, 0])
SpaceNavigator(t=0, x=0.0, y=0.0, z=0.0, roll=0.0, pitch=0.0, yaw=0.0, buttons=[0, 0])
```

5.2.1 Formát načítaných údajov

Údaje načítané zo vstupu 3D myši majú nasledovný formát:

- `t` – float, čas, kedy bol vstup načítaný
- `x` – float, posun myši pozdĺž osi x
- `y` – float, posun myši pozdĺž osi y
- `z` – float, posun myši pozdĺž osi z

- `roll` – float, rotácia okolo osi z
- `pitch` – float, rotácia okolo osi x
- `yaw` – float, rotácia okolo osi y

Hodnota čas `t` v programe nie je využitá, ani sa preto neposiela cez zdieľané hodnoty z čítacieho procesu. Zvyšné hodnoty sú v intervale `<-1;1>`. Hodnoty `x`, `y`, `z` čítajú posun a hodnoty `roll`, `pitch`, `yaw` rotáciu.

5.3 Zvuková odozva

Zvuková odozva v programe je implementovaná pomocou knižníc `gtts`² a `pydub`³. Predvolený nastavený jazyk čítania je angličtina. V pôvodnom návrhu aplikácie bola použitá knižnica `playsound` ale táto knižnica nefungovala s prostredím architektúry `docker`. Ako náhrada bola vybraná knižnica `pydub`. Táto knižnica ale obsahuje jednu chybu. Nedokáže prehrať zvuky, ktoré sú kratšie ako jedna audio jednotka definovaná knižnicou. Táto jednotka má približne 0,2 sekundy. Niektoré akcie, napríklad akcia `home`, boli vygenerované kratšie. Ako riešenie daného problému, boli prehrávané zvuky, kratšie ako minimálna jednotka predĺžené pridaním popisu, o akú položku sa jedná. Napríklad akcia `home` sa v programe prečíta ako `action home`, ale akcia `update_base` zostane ako `update_base`.

Vizuálna odozva

Program nepoužíva displej na vykresľovanie aplikácie. Preto sa vizuálnou odozvou v programe označujú prípady, keď užívateľovi pri práci tvorí odozvu pohyb samotných robotov na pracovisku. Táto forma odozvy je preferovanou voči zvukovej odozve.

5.4 Predpoklady pre stav servera

Na zahájenie práce s robotickým pracoviskom pomocou myši musí byť splnených niekoľko predpokladov zo strany servera.

Za prvé, server musí byť v stave úpravy konkrétneho projektu. Výber a spustenie projektu nie je možné uskutočniť pomocou myši, a preto táto akcia musí byť vykonaná s pomocou tabletu.

Za druhé, načítaný projekt musí byť spustený. Ak by projekt spustený nebol, server by neumožňoval pohybovať robotmi pomocou zasielania správ. Pohyb robotov je nevyhnutný pre funkciu aplikácie, ktorá ich pohyb využíva už v prvom kroku programu pri výbere robota.

Ak tieto podmienky splnené nie sú, program sa prepne do čakacieho režimu.

5.5 Vrstvenie programu

Program počas behu prechádza medzi vrstvami výberu užívateľa a čakacou vrstvou. Premenná riadiaca tok programu je typu `ProgramPosition` a môže nadobudnúť tieto stavy:

- `WAITING` – program je v stave čakania na spustenie projektu

²<https://pypi.org/project/gTTS/>

³<https://pypi.org/project/pydub/>

- ROBOTMENU – program je v stave výberu robota
- ACTIONMENU – program je v stave výberu akcie
- ROBOTMOVEMENT – program je v stave pohybu robota

5.5.1 Prechody medzi stavmi

V rôznych vrstvách reaguje program na vstup z myši rôznym spôsobom. Ale aby sa zachovala intuitívnosť programu vo všetkých vrstvách platí, že stlačenie ľavého tlačidla výber potvrdzuje a pravé tlačidlo výber ruší a posúva sa o vrstvu vyššie.

Čakací stav

Program v čakacom režime nereaguje na podnety z myši, okrem podnetu na zatvorenie aplikácie a čaká na správu zo strany servera o otvorení projektu. Program otvorenie projektu oznámi hlasovou správou, ale ostáva ďalej v čakacom režime. Až keď je projekt spustený, program prechádza do vrstvy výberu robota.

Ak by bol projekt počas manipulácie s programom zatvorený alebo zastavený, je táto zmena ohlásená hlasovou správou a projekt sa prepne naspäť do čakacieho stavu.

V čakacom režime môže užívateľ aplikáciu ukončiť stlačením pravého tlačidla myši.

Vrstva výberu robota

Vo vrstve výberu robota program reaguje na pohyby myšou doprava a doľava zmenou vybraného robota. V tejto vrstve je možné program ukončiť stlačením pravého tlačidla myši. Zvolením robota ľavým tlačidlom myši sa program presunie do vrstvy výberu akcie. Ak by v tejto vrstve došlo k uzatvoreniu alebo zastaveniu projektu, program sa prepne do čakacej vrstvy.

Vrstva výberu akcie

Program vo vrstve výberu akcie reaguje na pohyby myšou do strán, ktoré menia aktuálne vybranú akciu. Stlačením pravého tlačidla myši sa užívateľ presunie do vrstvy výberu robota. Stlačením ľavého tlačidla myši užívateľ zvolí vybranú akciu a presunie sa do vrstvy pohybu robota.

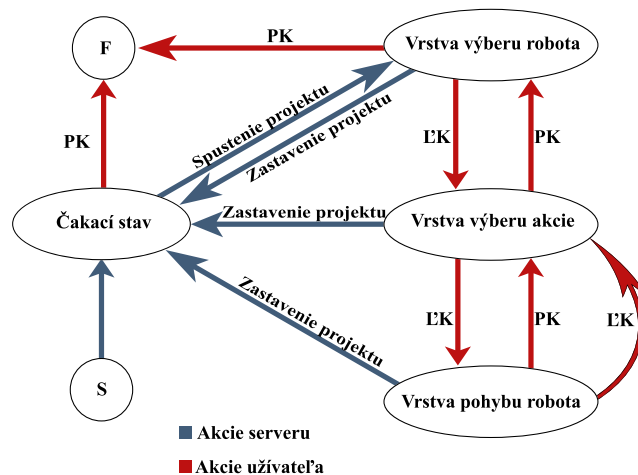
Ak by došlo k uzatvoreniu alebo zastaveniu projektu, program sa prepne do čakacej vrstvy.

Vrstva pohybu robota

Užívateľ v tejto vrstve pohybuje vybraným robotom pomocou 3D myši. Všetky pohyby joysticku sú poslané robotovi. Ak je robotovi zadaný pohyb do miesta, kde sa nemôže fyzicky dostať, alebo rotácia, ktorá nie je podporovaná, pohyb sa nevykoná.

Stlačením ľavého tlačidla užívateľ potvrdí pridanie akcie a program sa presunie do vrstvy výberu akcie. Stlačením pravého tlačidla sa užívateľ tiež presunie do vrstvy výberu akcie, ale žiadna akcia pridaná nie je.

Ak by došlo k uzatvoreniu projektu alebo zastaveniu scény program sa prepne do čakacej vrstvy.



Obr. 5.1: Stavový automat zobrazujúci tok programu

5.6 Výber robota

V projekte nie je priamo uložený zoznam robotov, obsahuje len zoznam objektov. Preto najprv prebehne výber robotov zo zoznamu objektov. Zavolaním správy `GetRobotMeta` získame zoznam všetkých robotov, ktoré systém dokáže rozpoznať. Následne program prechádza zoznam objektov a vyberie iba tie objekty, ktoré sú zároveň roboty.

5.6.1 Uložené informácie

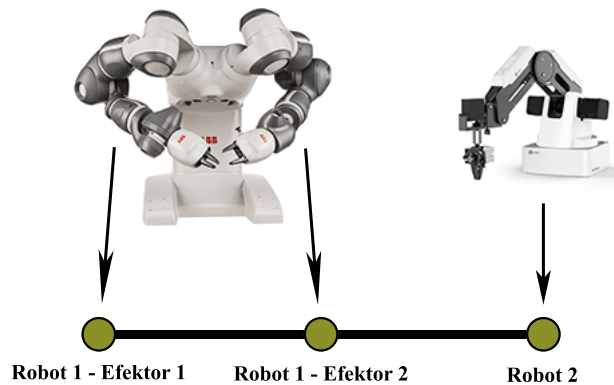
O jednotlivých robotoch si program ukladá názov, id, typ, id efektoru a poznámku, či má daný robot viac než jeden efektor. Názov robota v programe je použitý na zvukovú odozvu pre užívateľa. Id robota a efektoru sú potrebné pre ďalšiu prácu s robotom. Typ robota je použitý na získanie zoznamu akcií, ktoré robot dokáže vykonať. Položka `read_effector` nadobúda pravdivostnú hodnotu 1 v prípade, že ide o robota s viacerými efektormi. V prípade, že má robot viac než jeden efektor, program ponúka zvukovú odozvu rozšírenú o id efektoru.

5.6.2 Vytvorenie výberu

Systém ARCOR2 podporuje prácu s robotmi, ktoré majú viacero efektorov. Keďže sa ale nepredpokladá práca s robotom, ktorý ich má neprijateľne veľký počet, program sa vysporiadáva s výberom efektoru v rovnakom kroku ako výber robota. Ak má robot viacero efektorov, je do výberu pridaná jedna položka pre každý efektor.

5.6.3 Zamykanie robota

V systéme ARCOR2 je konflikt súbežnej práce s robotom riešený systémom zamykania. Zamknutie robota jedným užívateľom znemožní jeho použitie ostatnými užívateľmi. V programe prebieha zamknutie objektu v momente jeho výberu vo vrstve výberu robota a pretrváva až pokiaľ nie je zvolený iný robot, alebo nie je potvrdené pridanie novej akcie do programu.



Obr. 5.2: Tvorba zoznamu robotov

5.6.4 Vrsta výberu robota

Položky výberu sú pripravené. Poradie robotov nie je usporiadané, roboty sú v rovnakom poradí, v akom boli načítané.

Zvolenie prvého robota

Pri výbere robota je zvolený prvý nezamknutý robot. Ak nezamknutý robot nájdený nie je, program prejde do módu čakania na vstup. Ďalší pokus o vyhľadávanie nezamknutého robota prebehne až po vstupe od užívateľa. Pri výbere ďalšieho robota môže nastať rovnaká situácia a je riešená rovnakým spôsobom. Užívateľ nemôže zvoliť robota, ak žiadny nie je k dispozícii.

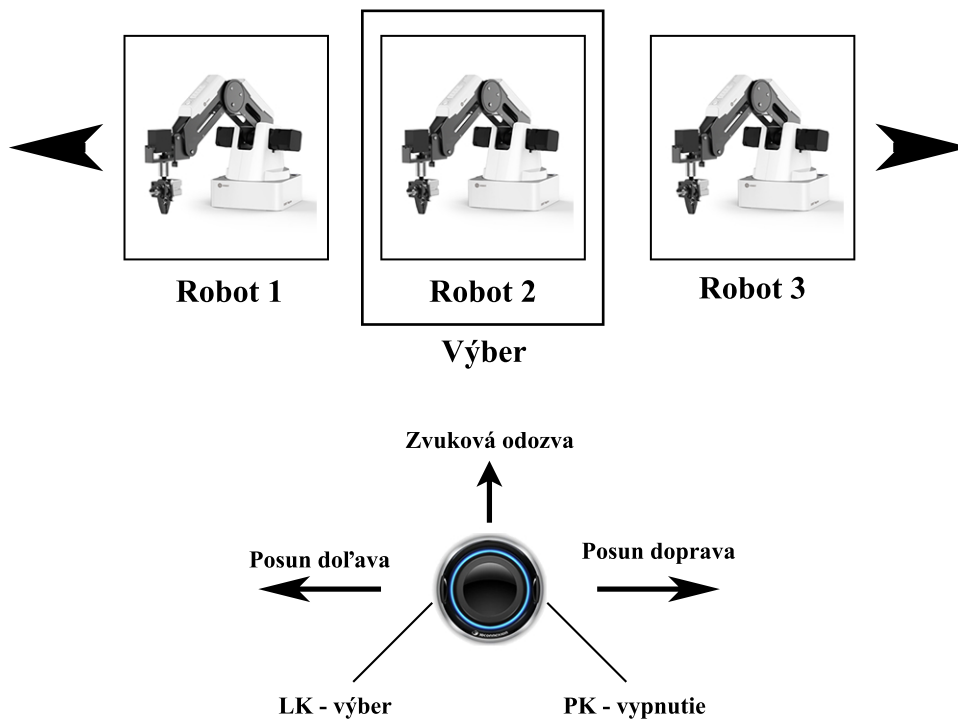
Vizuálna odozva

Pri zvolení robota sa program pokúsi označiť vybraného robota krátkym pohybom a návratom do pôvodnej pozície. Pohyb má dĺžku 1 cm. Môže ale nastať situácia, kde sa robot pokúša pohnúť na miesto, kde nie je pohyb možný. Ak by k tomu došlo, robot by nezmenil svoju pozíciu a nedošlo by k jeho označeniu. V snahe danému stavu predísť program vyskúša postupne všetky smery pozdĺž súradnicových osí, okrem smeru dole. Ako prvý bude vyskúšaný pohyb dohora. Pohyb dole na zobrazovanie nie je použitý pre prípad, že by sa medzi podkladom a efektorom nachádzalo neočakávané teleso. Ak by sa v tom prípade pohyb vykonal, riskovalo by sa jeho poškodenie. Ak sa žiadny z možných pohybov nepodarí vykonať, program pristúpi k použitiu zvukovej odozvy.

Zvuková odozva

Ak program nedokáže pohnúť robotom alebo je vyžiadaná zvuková odozva, program prečíta meno zvoleného robota uložené v triede `RobotInfo`. V prípade, že daný robot má viacero efektorov, je prečítaný text vo formáte "meno_robota - id efektoru".

Ak nie je možné zvoliť robota, program prečíta frázu "All robots locked". Ak sa užívateľ pokúsi o zvolenie robota, ak žiadny nie je k dispozícii, program prečíta frázu "No robot chosen".



Obr. 5.3: Schéma vrstvy výberu robota

Pri zvolení robota je prečítaný text vo formáte "Chosen meno_robota". V prípade, že robot má viacero efektorov je prečítaný text vo formáte "Chosen meno_robota - id efektoru".

5.7 Volenie akcie

Informácie o akciách dostupných pre zvoleného robota je možné získať zaslaním správy `GetActions` pre zvolený typ robota. Navrátený objekt je typu `list[ObjectAction]`.

5.7.1 Popis uloženia informácie o dostupnej akcií

Typ `ObjectAction` je uložený nasledovne:

- `name` – `string`, názov akcie
- `description` – `string`, popis akcie
- `parameters` – `list[ParameterMeta]`, zoznam parametrov akcie
- `meta` – `ActionMetadata`, obsahuje ďalšie informácie o akcií
- `origins` – `string`, typ robota, ktorý umožňuje danú akciu (v systéme ARCOR2 je podporovaná dedičnosť robotov)
- `returns` – `list[string]`, zoznam návratových typov
- `disabled` – `bool`, prepínač, či je akcia v prevádzke

- `problem` – `Optional[string]`, informácia o prípadnom probléme s akciou

Pre riešený projekt sú dôležité položky `name`, `description`, `disabled` a `parameters`. Meno a popis sú pre každú akciu uložené, aby boli následne použité na zvukovú odozvu pri výbere akcie. Dôležitou časťou sú ale parametre.

5.7.2 Popis uloženia metadát o parametri

Typ `ParameterMeta` je uložený nasledovne:

- `name` – `string`, názov parametru
- `type` – `string`, typ očakávanej hodnoty
- `description` – `string`, popis parametru
- `dynamic_value` – `bool`, ak je prepínač zapnutý, klient sa musí na povolené hodnoty opýtať
- `dynamic_value_parents` – `Optional[set[string]]`, dopĺňujúca hodnota k položke `dynamic_value`
- `default_value` – `Optional[string]`, predvolená hodnota parametru
- `extra` – `Optional[string]`, ďalšie informácie o parametri

5.7.3 Povinné parametre

Parametre, ktoré majú vyplnenú predvolenú hodnotu sú nepovinné. Parameter `pose`, ktorý obsahuje väčšina akcií, je tiež nepovinný, pretože sa pri zvolenej metóde parameter `pose` načíta automaticky z akčného bodu vytvoreného robotom v neskoršom kroku. Je ale potrebné vyriešiť zvyšné parametre.

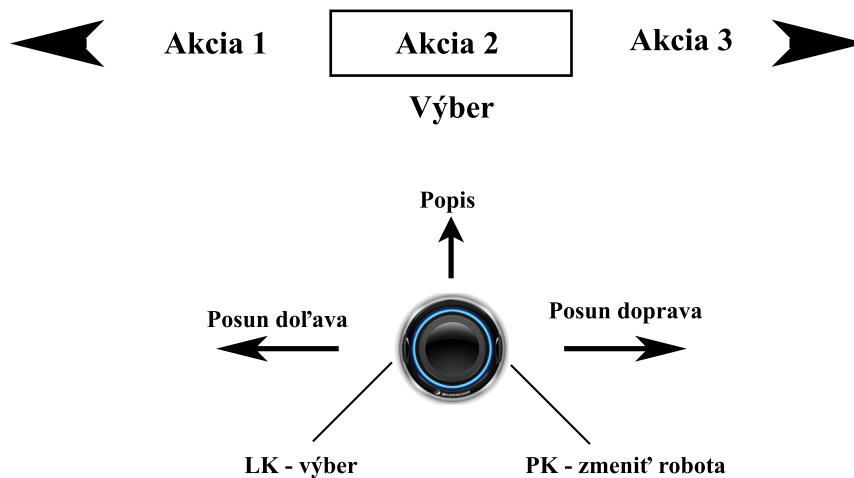
Ak nie je nastavená predvolená hodnota a nejedná sa o parameter `pose`, je pravdepodobné, že sa v premennej `extra` nachádza položka `allowed values`. Ak sa tam táto položka nachádza, program vyberie prvú hodnotu zo zoznamu. Ak sa tam táto položka nenachádza, akcia sa neobjaví vo výbere.

5.7.4 Vrstva výberu akcie

Vrstva pracuje obdobne ako vrstva výberu robota s niekoľkými rozdielmi. Užívateľ má k dispozícii len zvukovú odozvu. Preto bol program rozšírený o možnosť prečítania popisu danej akcie, pri pohybe myšou od seba.

5.8 Ovládanie robota 3D myšou

Vo vrstve pohybu robota sa pohyb 3D myši priamo prejaví pohybom vybraného efektoru. Pohyb v danej vrstve je možné popísať nasledovne.



Obr. 5.4: Schéma vrstvy výberu akcie

5.8.1 Jeden posun robota

Jednoduchý pohyb robota do zvoleného bodu v programe prebieha v niekoľkých krokoch.

Najskôr program zašle správu `RegisterForRobotEvents`. Server po spracovaní správy vo zvolenej variante začne zasielať programu správy o polohe koncového efektoru robota. Ak je ale robot v pohybe, server zasiela aj správy o začatí a ukončení pohybu.

Samotný pohyb robota vykonáva zaslanie správy `MoveToPose`. Ako parametre správy je zaslaná cieľová pozícia a rotácia efektoru. Ak by mal byť problém so zaslanými súradnicami, správa sa vráti s negatívnym výsledkom.

Program potom spracováva správy a čaká na správu o ukončení pohybu robota. Ak boli vykonané všetky požiadavky na pohyb, či už pri ukážke robota alebo samostatnom ovládaní, program odregistruje zasielanie správ o pohybe robota.

5.8.2 Cyklus pohybu

Užívateľ ovláda zvolený efektor robota myšou. Systém ARCOR2 ale nedokáže spracovať dlhý súvislý pohyb. Program teda pracuje v nasledujúcom cykle:

- Prečíta užívateľov pohyb myšou
- Vypočíta čiastkový pohyb robota z jeho momentálnej pozícia a vstupu od užívateľa
- Pošle robotovi správu s cieľovými súradnicami
- Čaká, kým robot dosiahne cieľovú pozíciu

Program daný cyklus opakuje, až kým užívateľ nepotvrdí pridanie novej akcie, nezruší pridávanie akcie alebo sa nezastaví projekt.

5.8.3 Výpočet nového bodu

Nový bod sa v programe vypočítava z aktuálnej pozície vybraného efektoru a aktuálneho vstupu z 3D myši.

Posun

Pri výpočte posunu prebehne pre každú zo súradníc nasledujúci výpočet:

$$\text{Nová hodnota} = \text{Predchádzajúca hodnota} + \text{Hodnota vstupu z myši} * \text{Koefficient citlivosti}$$

Koefficient citlivosti bol experimentálne stanovený na hodnotu 0.015.

Rotácia

Rotácia sa v systéme ARCOR2 spracováva pomocou triedy `Quaternion`. Pri výpočte rotácie sa využívajú metódy tejto triedy. Výpočet novej rotácie prebieha nasledovne:

$$\begin{aligned} \text{Quaternion myši} &= \text{Quaternion.from_rotation_vector}(\text{roll}, \text{pitch}, \text{yaw}) \\ \text{Výsledný Quaternion} &= \text{Predchádzajúci Quaternion} * \text{Quaternion myši} \end{aligned}$$

Hodnoty `roll`, `pitch` a `yaw` sú hodnoty rotácie načítané z myši.

Funkcia `Quaternion.from_rotation_vector` z načítaných hodnôt vyrobí triedu typu `Quaternion`. Výslednú rotáciu dostaneme vynásobením predchádzajúcej rotácie vyrobenou triedou.

5.9 Pridanie akcie

V tomto momente máme vybraného robota, efektor aj akciu, ktorú má vykonať. Efektor sa zároveň nachádza v pozícii, kde má byť daná akcia vykonaná.

5.9.1 Popis uloženia akcie

Trieda `Action` obsahuje:

- `name` – `string`, názov akcie zadaný užívateľom
- `id` – `string`, id inštalácie triedy `Action`
- `type` – `string`, skladá sa z id robota a názvu akcie
- `parameters` – `list[ActionParameter]`, zoznam parametrov potrebných pre vykonanie akcie
- `flows` – `list[Flow]`, zoznam tokov, v ktorých sa akcia vykoná

Mechanika tokov ešte nie je plne implementovaná v programe ARCOR2, preto sa počíta len so základným tokom.

5.9.2 Popis uloženia akčného bodu

Bod akcie v programe nemusí byť naviazaný na konkrétneho robota. Podpora naviazať ho je ale podporovaná, viz. položky `orientations` a `robot_joints`.

- `name` – `string`, názov zadaný užívateľom
- `position` – `Position`, pozícia v priestore programu
- `id` – `string`, id inštancie triedy `ActionPoint`
- `orientations` – `list[NamedOrientation]`, zoznam orientácií, ktoré musí robot nadobudnúť aby sa jeho efektor dostal do danej polohy
- `robot_joints` – `list[ProjectRobotJoints]`, zoznam kĺbov robota
- `actions` – `list[Action]`, zoznam akcií naviazaných na danú polohu

5.9.3 Pridanie akčného bodu

V programe ARCOR2 sú podporované dve metódy pridania akčného bodu.

Pri prvej metóde pomocou zaslania správy `AddActionPoint` je pridaný akčný bod, ktorý nie je nastavený na konkrétneho robota a teda položky `orientations` a `robot_joints` nie sú vyplnené. Položka `position` pri tejto metóde musí byť zadaná manuálne.

Pri druhej metóde pomocou zaslania správy `AddApUsingRobot` je pridaný akčný bod v mieste, kde sa nachádza efektor zadaného robota. Nový akčný bod sa vyrobí nastavený na zadaného robota a jeho položky `orientations` a `robot_joints` sú automaticky vyplnené. Pri vývoji môjho ovládača som sa rozhodol použiť druhú metódu.

5.9.4 Pridanie akcie

Väčšina parametrov už bola vyplnená v predchádzajúcom kroku. Zvyšné parametre budú vyplnené automaticky z triedy `ActionPoint`.

5.10 Zaradenie akcie do kontextu programu

Projekt vytvorený v programe ARCOR2 má podporu pre sériové radenie akcií do programu. Ak je táto funkcia zapnutá, je pravdepodobne žiadané, aby nami tvorené akcie boli tiež pridané do programu. V pôvodnom návrhu bol tento proces priamo volený užívateľom, ale pre zjednodušenie a urýchlenie metódy bola táto voľba odstránená. Novo pridaná akcia sa automaticky pridáva nakoniec programu.

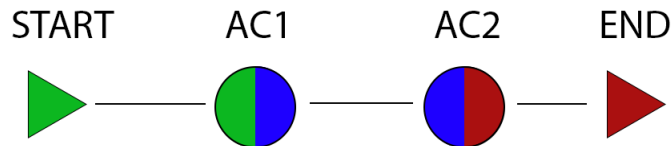
5.10.1 Popis uloženia programu

Projekt má atribút `logic` typu `list[LogicItem]`. Prvky zoznamu nie sú zoradené priamo. Trieda `LogicItem` obsahuje

- `id` – `string`, id inštancie triedy `LogicItem`
- `START` – `string`, id vykonávanej akcie (typ `Action`), alebo špeciálna hodnota `START`
- `END` – `string`, id nasledujúcej akcie (typ `Action`), alebo špeciálna hodnota `END`

- `condition` – `ProjectLogicIf`, podmienka vykonania akcie, ak je určená.

Každý prvok zoznamu obsahuje id vykonávanej akcie a id nasledujúcej akcie. Špeciálne kontrolné objekty `START` a `END` označujú začiatok a koniec programu vykonávaného v nekonečnom cykle. Pri preklade celého programu sa teda vyhľadá objekt `LogicItem`, ktorý má odkaz na objekt `START`. Program následne vyhľadá nasledujúci objekt `LogicItem` podľa príslušného odkazu. Takto program postupuje, pokiaľ nenájde odkaz na objekt `END`.



Obr. 5.5: Ilustrácia reťazenia akcií

5.10.2 Implementácia zaradenia akcie

Pri pridávaní akcie do programu sa najprv skontroluje, či daný projekt podporuje reťazenie akcií. Ak pre daný projekt reťazenie zapnuté nie je, žiadne sa nevykoná.

Ak reťazenie zapnuté je, akcia sa vždy pridá na koniec programu a program sa zakončí objektom `END`. Vykoná sa to najprv poslaním správy `UpdateLogicItem`, ktorý upraví posledný prvok na nový odkaz a následne `AddLogicItem`, ktorý pridá nový prvok.

Podmienky

System ARCOR2 je navrhnutý tak, aby užívateľ mohol pridávať podmienky do programu. Jednoduchosť implementácie môjho programu totiž neponúka dostatok odzvy na to, aby bolo možné prehľadne pracovať s vetvením programu.

Ak sa v užívateľovom programe nachádza vetvenie, program nebude vedieť, do ktorej vetvy má danú akciu zaradiť. V danom prípade program vytvorí nový akčný bod a pridá novú akciu, ale k zaradeniu do programu nedôjde.

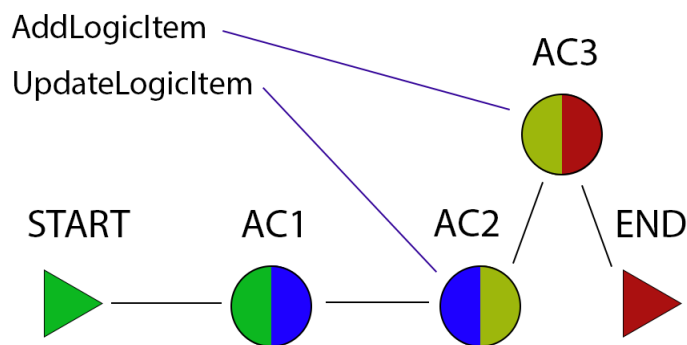
Užívateľ následne môže novú akciu do programu pridať na želané miesto pomocou rozhrania v tablete.

5.11 Prispôsobenie pre architektúru Docker

Pri prispôbovaní aplikácie pre nasadenie pomocou platformy docker nastalo niekoľko ďalších zmien v programe, ktoré považujem za zaujímavé zmeniť.

5.11.1 Docker a 3D myš

Architektúra docker si pre prácu so zariadeniami používajúcimi raw HID vstup vyžaduje privilegované práva pre konkrétny kontajner. Taktiež si to vyžaduje, aby bol program v kontajneri vykonávaný pod užívateľom `root`.



Obr. 5.6: Ilustrácia pridania novej akcie

5.11.2 Docker a audio

Vytvorenie balíčku docker s použiteľným prehrávačom zvukov si vyžadovalo niekoľko zmien. Ako príklad uvediem výmenu použitej zvukovej knižnice z playsound na pydub. Knižnica pydub totiž používa systém ALSA (Advanced Linux Sound Architecture)⁴, ktorý je vhodný pre prehrávanie zvuku z docker kontajneru. Aby daný systém fungoval je nutné aby bol program v kontajneri vykonávaný pod rovnakým užívateľským id, s akým bol spustený.

5.11.3 Riešenie konfliktu

Medzi popísanými požiadavkami vzniká konflikt. 3D myš vyžaduje ako užívateľa root, zatiaľ čo ALSA vyžaduje konkrétne užívateľské id. Daný problém bol vyriešený tvorbou separátnych procesov pri prehrávaní zvukov. Tieto procesy sa vykonávali so zadaným užívateľským id, zatiaľ čo hlavný program je vykonávaný s oprávneniami root užívateľa.

5.12 Spustenie programu

Program je možné spustiť pomocou príkazu docker compose up so špecifikáciou konkrétnej služby.

5.12.1 Docker a prostredie

Pre úspešné spustenie programu je nutné dockeru zadať id aktívneho užívateľa. Toto id je v premennej UID. Pri spustení pomocou príkazu docker compose, je nutné napred vytvoriť v adresári s .yml súborom súbor .env a vložiť tam nasledujúci riadok UID=1000. 1000 je užívateľské id užívateľa na mojom systéme, pre zistenie aktívneho užívateľa stačí zadať príkaz echo \$UID.

⁴https://www.alsa-project.org/wiki/Main_Page

Kapitola 6

Užívateľské testovanie

Po ukončení implementácie programu prebehlo užívateľské testovanie na Fite. Ako vhodný spôsob testovania bolo vybrané porovnávacie testovanie.

6.1 Teória užívateľského testovania

Existuje niekoľko druhov delenia testovania [5]. Pred výberom správneho testovania je potrebné najskôr vybrať správny druh testovania podľa cieľov testovania, dostupných zdrojov a cieľového publika.

Moderované a nemoderované testovanie

Moderované testovanie prebieha za prítomnosti výskumníka, ktorý účastníkom testu vysvetlí test, zodpovie ich otázky a po ukončení testu pokladá dotazy. Moderovaný prístup je používaný v prípadoch, kedy je potrebné zozbierať presnejšie dáta, ktoré nie je možné zozbierať na diaľku. Používa sa tiež na zisťovanie príčin užívateľského správania pri používaní aplikácie.

Nemoderované testovanie prebieha bez priameho dohľadu výskumníka, Účastníci tiež nemusia byť priamo prítomní na konkrétnom mieste, dáta sú zbierané používaním testovanej aplikácie. Nemoderovaný prístup je ľahší na organizáciu a pri veľkých projektoch lacnejší.

Zvolený bol moderovaný prístup z nasledujúcich dôvodov:

- Aplikáciu nie je možné distribuovať na diaľku, prítomnosť účastníkov je nevyhnutná
- Testovanie aplikácie vyžaduje zároveň určitú znalosť práce s tabletovou verziou aplikácie. Vysvetlenie práce s aplikáciou je preto nevyhnutné.
- Potrebné dáta nie je možné zozbierať pomocou aplikácie, ďalšie dotazy sú potrebné.

Diaľkové a osobné testovanie

Diaľkové testovanie, ako napovedá názov, prebieha online alebo telefonicky, zatiaľ čo osobné testovanie si vyžaduje fyzickú prítomnosť testovacích subjektov na zvolenom mieste.

Ako bolo povedané v predchádzajúcom delení, aplikácia si vyžaduje testovanie v lokácii, ktorá poskytuje prístup k robotickému pracovisku s bežiacim serverom ARCOR2. Testovanie na diaľku preto nie je možné.

Prieskumné, posudkové a porovnávacie testy

Prieskumné testovanie bežne prebieha v prvých štádiách vývoja aplikácie. Cieľom testovania je zozbieranie požiadavok na fungovanie aplikácie.

Posudkové testovanie slúži na meranie spokojnosti používateľov s vyvinutou aplikáciou, prípadne na naplánovanie ďalšieho postupu vo vývoji. Je použité na otestovanie celkovej funkčnosti produktu.

Porovnávacie testovanie je používané na porovnanie dvoch rôznych riešení problému, väčšinou na porovnanie vyvinutej aplikácie s konkurenciou.

Cieľom užívateľského testovania je porovnanie existujúceho riešenia ovládania systému ARCOR2 cez tablet a novej metódy ovládania systému pomocou 3D myši. Zvolené bolo preto porovnávacie testovanie.

6.2 Popis vykonaného testu

Testovanie prebehlo počas dvoch dní v robotickom laboratóriu na FITe.

6.2.1 Naplánované otázky

Každý z účastníkov testu bol požiadaný o vyplnenie dotazníku s nasledujúcimi otázkami v priebehu testu.

1. Aká je vaša skúsenosť s prácou s aplikáciami na tabletovom zariadení?

Nemám žiadnu skúsenosť ○ ○ ○ ○ ○ **Mám veľa skúseností**

2. Do akej miery ste pochopili vysvetľovanie postupu práce s tabletovou verziou aplikácie.

Nepochopil som ○ ○ ○ ○ ○ **Perfektne som pochopil**

3. Ako hodnotíte vašu skúsenosť s prácou s tabletovou aplikáciou.

Veľmi zlá ○ ○ ○ ○ ○ **Výborná**

4. Aká je vaša skúsenosť s prácou s aplikáciami ovládanými 3D myšou alebo joystickom.

Nemám žiadnu skúsenosť ○ ○ ○ ○ ○ **Mám veľa skúseností**

5. Do akej miery ste pochopili vysvetľovanie postupu práce s verziou aplikácie používajúcou 3D myš.

Nepochopil som ○ ○ ○ ○ ○ **Perfektne som pochopil**

6. Ako hodnotíte vašu skúsenosť s prácou s aplikáciou používajúcou 3D myš.

Veľmi zlá ○ ○ ○ ○ ○ **Výborná**

7. Ktorý prístup hodnotíte ako lepší?

Okrem toho som počas testu účastníkom zodpovedal dotazy a snažil sa určiť miesta v programe, ktoré boli pre účastníkov testu problematické. Meraný bol aj čas, ktorý účastníci potrebovali na naprogramovanie testovacieho zadania jednotlivými prístupmi.

6.2.2 Testovacie zadanie

Účastníci boli pri testovaní požiadaní o naprogramovanie nasledujúceho zadania najskôr pomocou tabletovej aplikácie AREditor a potom pomocou 3D myši. Zadanie bolo navrhnuté veľmi jednoduché, aby účastníkom nezabralo veľa času ho naprogramovať obidvoma spôsobmi.

- Robot zoberie predmet (testovacie kocku) z pozície číslo 1.
- Robot položí predmet na pozíciu číslo 2.
- Robot sa vráti do pôvodnej pozície.

S niektorými účastníkmi potom bolo vykonané aj druhé testovanie, ktoré zahŕňalo rozšírené zadanie. Toto zadanie obsahovalo aj použitie dopravného pásu, s ktorým nie je možné manipulovať pomocou 3D myši. Cieľom tohto testu bolo otestovať možnosť kombinácie oboch spôsobov.

- Robot zoberie predmet (testovacie kocku) z pozície číslo 1.
- Robot položí kocku na dopravný pás.
- Dopravný pás presunie kocku na iné miesto.
- Robot zoberie kocku z dopravného pásu.
- Robot položí kocku na na pozíciu číslo 2.

Tabuľka výsledkov						
Pohlavie	M	M	M	M	Ž	Ž
Skúsenosť – tablet	5	4	5	5	5	5
Pochopenie vysvetlenia – tablet	4	5	3	3	4	4
Hodnotenie aplikácie – tablet	4	2	2	5	5	4
Výsledný čas – tablet	4:15	5:30	5:12	4:30	4:15	4:40
Skúsenosť – 3D myš	3	1	2	2	1	1
Pochopenie vysvetlenia – 3D myš	4	5	4	5	5	5
Hodnotenie aplikácie – 3D myš	3	4	4	1	3	4
Výsledný čas – 3D myš	3:45	3:50	3:15	4:30	3:30	3:45
Preferovaná metóda	tablet	myš	myš	tablet	tablet	myš

Tabuľka 6.1: Tabuľka výsledkov testu č.1

Hodnotenie – tablet	4	2
Výsledný čas – tablet	7:37	8:39
Hodnotenie – 3D myš	3	3
Výsledný čas – 3D myš	6:07	6:51
Preferovaný prístup	tablet	3D myš

Tabuľka 6.2: Tabuľka výsledkov testu č.2

6.2.3 Plán testu

Test bol naplánovaný nasledovne:

- Účastníkom testu bolo vysvetlené ovládanie systému ARCOR2 pomocou tabletu.
- Každý účastník v systéme vyrobil projekt obsahujúci riešenie zvoleného zadania pomocou tabletu.
- Účastníci boli vyzvaní na vyplnenie prvých 3 otázok
- Účastníkom bolo vysvetlené ovládanie systému pomocou 3D myši
- Každý účastník v systéme vyrobil projekt obsahujúci riešenie zvoleného zadania pomocou 3D myši
- Účastníci boli vyzvaní na vyplnenie zvyšných otázok.

6.2.4 Výsledky testov

Prvé kolo testovania prebehlo so šiestimi účastníkmi a prinieslo nasledujúce výsledky (hodnoty v tabuľkách).

Druhého testu sa zúčastnili prví dvaja účastníci z testu číslo jedna. Jedná sa o prvé dva stĺpce tabuľky z prvého testu.

6.2.5 Vyhodnotenie výsledkov

V nasledujúcej sekcii vyhodnotím nazbierané výsledky a popíšem aj verbálne dotazy účastníkov, rovnako ako aj odozvu, ktorú mi účastníci poskytli navyše. Taktiež v krátkosti popíšem námety na vylepšenia, ktoré by bolo možné v budúcnosti pridať.

Vyhodnotenie dotazníkových výsledkov

V oblasti skúsenosti so zariadením je možné povedať, že každý z účastníkov mal bohaté skúsenosti prácou s tabletom, prípadne inými zariadeniami ovládanými dotykovou obrazovkou, zatiaľ čo len jeden z účastníkov pred testami vyskúšal použiť 3D myš. Účastníci mali preto niekedy problém s použitím funkcií 3D myši a potrebovali viac času na efektívne použitie zariadenia.

Ak ale išlo o pochopenie vysvetlenia práce so zariadením, užívatelia pozitívnejšie hodnotili prístup s 3D myšou, aj počet dotazov počas testu bol podstatne menší.

Celkové hodnotenia aplikácie sa zhodujú s preferovanou metódou prístupu jednotlivých účastníkov. Možno len zmieniť, že prístup s tabletom účastníci hodnotili v priemere 3,67/5 a prístup s 3D myšou 3.17/5.

Taktiež je možné poznamenať, že pri programovaní rozšíreného programu, kde bolo nutné použiť pri práci aj tablet, bolo hodnotenie prístupu s 3D myšou o niečo horšie. Rozdiel v časoch je tiež možné čiastočne vysvetliť lepšou znalosťou používania zariadenia, v porovnaní s predchádzajúcim testom.

Dotazy účastníkov

Počas testu sa ma účastníci mohli pýtať dotazy, v prípade, že by si neboli istý ďalším postupom. Väčšina dotazov sa týkala tabletovej metódy, konkrétne to boli dotazy na potvrdenie si správneho tlačidla.

Dotazy počas práce s tabletom sa týkali prevažne rozdelenia na vrstvy. Užívatelia opakovane zabudli, v ktorej vrstve sa momentálne nachádzajú. Daný problém netrval dlho užívatelia sa pri dlhšej práci so zariadením začali vo vrstvách orientovať. Prípadne by v ďalších verziách programu bolo možné preskúmať dodatočné formy odozvy, ktoré by napovedali užívateľovi stav programu.

Niektoré dotazy sa týkali zariadenia 3D myši, niektorí účastníci testov nemali predtým skúsenosť priamo z 3D myšou, ale len s joystickom. Jeden z účastníkov nevedel nájsť tlačidlo myši a snažil sa potvrdiť akciu v menu stlačením joysticku. V ďalšej verzii programu by možno bolo vhodné túto možnosť tiež pridať. Okrem sa účastníci často dotazovali na pohyb s joystickom. Často totiž namiesto pohybu joystickom účastník vykonal rotáciu joystickom. V menu táto akcia nič nevykoná. Danú funkciu by tiež bolo možné pridať.

Ďalšie postrehy

Niektorí účastníci sa rozhodli prístupy skombinovať iným než očakávaným spôsobom. Pri práci s myšou, vo vrstve pohybu robota, použili možnosť prepnutia robota do stavu učenia rukou a manuálne premiestnili robota do koncovej polohy. Následne novú akciu pridali pomocou 3D myši.

Tu popíšem ešte zopár trendov, ktoré som si všimol počas testovania. Väčšinu účastníkov testovania osobne poznám, a preto by som doplnil ešte niekoľko ďalších postrehov, ktoré sa týkajú technickej zdatnosti používateľa. Účastníci, ktorí zvolili ako preferované riešenie

tablet boli prevažne technicky zdatnejší, ako účastníci, ktorí volili riešenie s 3D myšou. Aj keď nie je možné potvrdiť túto koreláciu z tak malej vzorky testovaných účastníkov, ak by táto korelácia potvrdená bola, je možné z nej vyvodiť pomerne zaujímavý záver.

Projekt ARCOR2 je zameraný na presun programovania robotického pracoviska do rúk koncového užívateľa. Pri takom užívateľovi je možné predpokladať menšiu technickú zdatnosť, aspoň čo sa programovania týka. Ak by sa daná korelácia potvrdila, je možné, že prístup k programovaniu pomocou 3D myši by bol preferovanou metódou programovania pre cieľovú skupinu užívateľov.

Kapitola 7

Záver

Cieľom mojej práce bolo vytvoriť rozšírenie pre systém ARCOR2, ktoré by povoľovalo aspoň čiastočné ovládanie systému pomocou 3D myši.

V prvom kroku práce som sa dôkladne oboznámil s fungovaním systému ARCOR2. Vyskúšal som si prácu so systémom pomocou tabletu a určil nedostatky, ktoré by mal prístup s 3D myšou riešiť. Taktiež som sa oboznámil s aplikačným rozhraním, ktoré systém ARCOR2 používa, a ktorým mala výsledná aplikácia so systémom komunikovať.

V ďalšom kroku boli vytýčené hlavné požiadavky na novú aplikáciu. Stanovilo sa, ktoré úlohy má táto aplikácia spĺňať, čo sú podmienky a priority projektu. V danom kroku sa vychádzalo zo zamerania systému ARCOR2, ktorý už isté požiadavky mal, a tiež nedostatkov, ktoré vzišli z analýzy prvotného prístupu ovládania systému pomocou tabletu. Výsledkom daného kroku bol zoznam požiadaviek na aplikáciu. Medzi priority sa radili intuitívnosť a rýchlosť výslednej aplikácie.

Nasledoval prieskum možností komunikácie medzi robotom a človekom. Daný prieskum mal za úlohu nájsť čo najlepší spôsob odozvy, aby sa splnila požiadavka na intuitívnosť aplikácie. Prieskum zahŕňal prevažne rôzne formy odozvy pre užívateľa, ako vstupné zariadenie bola už vybraná 3D myš. Prieskum ukázal, že najlepšou formou odozvy je odozva vizuálna, preto bola výsledná aplikácia navrhnutá tak, aby využívala prevažne túto formu odozvy. Aplikácia je ale navrhnutá pre pracoviská, ktoré nemusia mať prístup k monitoru a preto sa vizuálna odozva viaže priamo na pracovisko. Vizuálna odozva mala byť dopĺňaná audio odozvou.

Pred začiatkom implementácie bol vytvorený prvotný návrh aplikácie. Tá sa mala skladať z niekoľkých vrstiev, v ktorých užívateľ prechádzal jednotlivé možnosti. Vytvorený návrh mal tiež v snahe zachovať veľkú časť funkcionality prístupu s tabletom. Daný prístup sa už v prvých kolách testov ukázal ako nevhodný, pretože sťažoval prácu s aplikáciou. Nasledoval preto popis vykonaných úprav, ktoré uprednostňovali predovšetkým jednoduchosť výslednej aplikácie.

Taktiež došlo k niekoľkým ďalším zmenám v pôvodnom návrhu, ktoré boli vynútené nesprávnym fungovaním knižníc, prípadne ich nekompatibilitou.

V kapitole implementácia som popísal systém funkcie všetkých tried a vrstiev programu, ako aj komunikáciu aplikácie so systémom. Popísal som problémy, ktorým som v priebehu práce čelil a riešenia, ktoré som implementoval.

Po dokončení implementácie prebehlo v laboratóriu na FITe užívateľské testovanie, ktorého sa zúčastnilo šesť študentov. Účastníci si vyskúšali prácu s tabletovou aplikáciou, aj s novo vytvorenou aplikáciou využívajúcou 3D myš. Po skončení testovania účastníci zodpovedali niekoľko dotazov a výsledný produkt ohodnotili. Boli zaznamenané aj otázky, ktoré sa

účastníci pýtali počas testovania. Zozbierané údaje boli vyhodnotené a hovoria o pomerne úspešnom výsledku projektu.

Podľa nazbieraných údajov bolo navrhnutých aj niekoľko ďalších smerov, ktorými by sa mala práca v budúcnosti uberať. Medzi tieto smery patrí napríklad vylepšenie odozvy pre signalizáciu vrstvy aplikácie, v ktorej sa užívateľ momentálne nachádza.

Nakoniec som natočil video, ktoré v krátkosti demonštruje použitie 3D myši na naprogramovanie krátkeho ukázkového programu¹.

Počas práce som nadobudol mnohé pozitívne skúsenosti s prácou na rozsiahlejšom projekte pomocou platformy github. Taktiež som si vyskúšal prácu s platformou docker a systémom pants, ktoré napriek spôsobeným nepríjemnostiam hodnotím prevažne pozitívne a som rád, že som mal príležitosť ich využiť. Pozitívne hodnotím aj príležitosť podieľať sa na projekte v oblasti robotiky, ako jednej z mojich obľúbených tém.

¹<https://www.youtube.com/watch?v=z8-ONNddoMk>

Literatúra

- [1] 2020, I. C. *Remote Procedure Call* [online]. 2020 [cit. 2023-5-1]. Dostupné z: <https://www.ibm.com/docs/en/aix/7.1?topic=concepts-remote-procedure-call>.
- [2] ARISTIDOU, A., LASENBY, J., CHRYSANTHOU, Y. a SHAMIR, A. Inverse Kinematics Techniques in Computer Graphics: A Survey. *Computer Graphics Forum*. 2018, zv. 37, č. 6, s. 35–58. DOI: <https://doi.org/10.1111/cgf.13310>. Dostupné z: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13310>.
- [3] D'SOUZA, A., VIJAYAKUMAR, S. a SCHAAL, S. Learning inverse kinematics. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*. 2001, sv. 1, s. 298–303 vol.1. DOI: 10.1109/IROS.2001.973374.
- [4] ENOCH, J., McDONALD, L., JONES, L., JONES, P. R. a CRABB, D. P. Evaluating whether sight is the most valued sense. *JAMA Ophthalmol*. American Medical Association (AMA). november 2019, zv. 137, č. 11, s. 1317.
- [5] HOTJAR. *The different types of usability testing methods for your projects* [online]. 2022 [cit. 2023-5-1]. Dostupné z: <https://www.hotjar.com/usability-testing/methods>.
- [6] LUO, R., WANG, C., SCHWARM, E., KEIL, C., MENDOZA, E. et al. Towards Robot Avatars: Systems and Methods for Teleinteraction at Avatar XPRIZE Semi-Finals. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, s. 7726–7733. DOI: 10.1109/IROS47612.2022.9982258. Dostupné z: https://ieeexplore.ieee.org/abstract/document/9982258?casa_token=t04_FemcU-kAAAAA:RbZoXvfFw6zVplb8GUcLhyAA2axgF00pkELKcWwaWHTnATe4m_j8z1chI9cw0zrhqhszJT-haks0.
- [7] MATERNA, Z., ŠPAÑEL, M., MAST, M., BERAN, V., WIESSHARDT, F. et al. Teleoperating assistive robots: A novel user interface relaing on semi-automy and 3D enviroment mapping. *Journal of Robotics and Mechatronics*. 2017, zv. 29, č. 2, s. 381–394.
- [8] PH.D., I. Z. M. *ARCOR2* [<https://github.com/robofit/arcor2>]. GitHub, 2023. Dostupné z: <https://github.com/robofit/arcor2>.
- [9] RAD, B. B., BHATTI, H. J. a AHMADI, M. An introduction to docker and analysis of its performance. *International Journal of Computer Science and Network Security (IJCSNS)*. International Journal of Computer Science and Network Security. 2017, zv. 17, č. 3, s. 228.

- [10] REN, F. a BAO, Y. A Review on Human-Computer Interaction and Intelligent Robots. *International Journal of Information Technology & Decision Making*. 2020, zv. 19, č. 01, s. 5–47. DOI: 10.1142/S0219622019300052. Dostupné z: <https://doi.org/10.1142/S0219622019300052>.