

Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra informatiky

**Robot NAO jakožto nástroj na procvičování
matematických dovedností**

NAO robot as a tool for practicing math skills

Bakalářská práce

Vypracovala: Daniela Pevná

Vedoucí práce: Mgr. Václav Šimandl, Ph.D.

České Budějovice 2021

Zadání bakalářské práce

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta
Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Daniela PEVNÁ
Osobní číslo: P18483
Studijní program: B7507 Specializace v pedagogice
Studijní obor: Informační technologie a e-learning
Téma práce: Robot NAO jakožto nástroj na procvičování matematických dovedností
Zadávající katedra: Katedra informatiky

Zásady pro vypracování

Cílem práce je naprogramovat robota NAO pro procvičování jednoduché matematiky ve skupině žáků 1. stupně základní školy. Robot by měl pomocí svého hlasového výstupu dávat dětem matematické příklady a jejich správnou či nesprávnou hlasovou odpověď odměnit odpovídající reakcí. Příklady budou na sčítání, odčítání, násobení a dělení. Vytvořený program bude možno použít k procvičování nejen jedné konkrétní matematické operace, ale také všech typů příkladů v náhodném sledu za sebou. Náročnost příkladů bude taková, aby dítě bylo schopno je spočítat z paměti v krátkém časovém intervalu. Hlasový výstup by měl být pro dětskou cílovou skupinu zajímavý, případně doprovázený pohybovou aktivitou, která dítě zaujme a zpříjemní procvičování. Robot bude schopen identifikovat ve svém okolí lidské tváře a na základě toho pokládat příklady konkrétním lidem. Využitelnost vytvořeného programu bude opakovaně ověřena v dětském kolektivu. Součástí ověřování bude pozorování reakce dětí na tuto formu procvičování a analýza získaných dat. V teoretické části práce studentka popíše existující výukové roboty a robotické hračky, které jsou využitelné ve výuce předmětů jiných než informatika.

Rozsah pracovní zprávy: 40
Rozsah grafických prací: CD ROM
Forma zpracování bakalářské práce: tištěná

Seznam doporučené literatury:

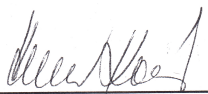
1. ALDEBARAN ROBOTICS. First steps with NAO. In: Aldebaran documentation [online]. 2015. Dostupné z: <http://doc.aldebaran.com/documents/first-steps-with-nao.pdf>
2. BEITER, M., COLTIN, B., LIEMHETCHARAT, S. An Introduction to robotics with NAO [online]. Aldebaran Robotics: Francie, 2012, 204 s. Dostupné z: http://www.kramirez.net/Robotica/Material/Nao/AnIntroductionToRoboticsWithNao_TextBook_2012_US.pdf
3. HENDL, J. Kvalitativní výzkum: základní metody a aplikace. Vyd. 1. Praha: Portál, 2005, 407 s. ISBN 80-736-7040-2.
4. SOFTBANK ROBOTICS DOCUMENTATION [online]. 2020. Dostupné z: http://doc.aldebaran.com/2-8/home_nao.html#
5. SEO, K. Using NAO: Introduction to interactive humanoid robots. Aldebaran Robotics: Francie, 2013, 276 s.
6. ŠVAŘÍČEK, R., ŠEĐOVÁ, K. et al. Kvalitativní výzkum v pedagogických vědách. Praha: Portál, 2007, 377 s. ISBN 978-80-7367-313-0.

Vedoucí bakalářské práce: Mgr. Václav Šimandl, Ph.D.
Katedra informatiky

Zadání bakalářské práce

Datum zadání bakalářské práce: 1. dubna 2020
Termín odevzdání bakalářské práce: 30. dubna 2021

Pro informaci: Tato práce je určena pro studenty bakalářského studijního programu. Studenti jsou povinni dodržovat všechny podmínky a termíny uvedené v zadání. Práce musí být odevzdána v termínu a v souladu s požadavky na formátování. V případě jakýchkoli dotazů se obraťte na svého vedoucího práce.



doc. RNDr. Helena Koldová, Ph.D.
děkanka



L.S.



doc. PaedDr. Jiří Vaníček, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 1. dubna 2020

Prohlášení

Prohlašuji, že jsem svoji bakalářskou práci vypracovala samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. V platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejich internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 19. dubna 2021

Daniela Pevná

Abstrakt

Má bakalářská práce se zabývá využitím robota Nao pro procvičování matematických dovedností ve skupině žáků 1. stupně základní školy. Robot pomocí svého hlasového výstupu dává dětem matematické příklady a jejich správnou či nesprávnou hlasovou odpověď odměňuje odpovídající reakcí. Příklady procvičují sčítání, odčítání, násobení a dělení.

Tento program je cílený k procvičování nejen jedné konkrétní matematické operace, ale také všech typů příkladů v náhodném sledu za sebou. Příklady jsou tvořeny tak, aby dítě bylo schopno je spočítat z paměti v krátkém časovém intervalu. Hlasový výstup je přizpůsoben, aby byl pro dětskou cílovou skupinu zajímavý, případně doprovázený pohybovou aktivitou, která dítě zaujme a zpříjemní procvičování. Robot je schopen identifikovat ve svém okolí lidské tváře a na základě toho pokládat příklady konkrétním lidem.

V teoretické části práce jsou popsáni existující výukoví roboti a robotické hračky, které jsou využitelné ve výuce předmětů jiných než informatika. Praktická část se zabývá tvorbou programu.

Klíčová slova

robot, Nao, Choregraphe, matematika, procvičování, programování, žáci

Abstract

My bachelor thesis deals with the use of the robot Nao for practising mathematical skills in a group of pupils of the 1st grade of primary school. The robot gives mathematical operations to pupils using its voice output, and then the robot waits for the answers to his questions. He rewards their correct or incorrect responses with their appropriate reaction. Mathematical operations based on addition, subtraction, multiplication and division.

This program is focused on practising not only one specific mathematical operation but also on all types of operations in a random sequence. The examples are designed so that the pupil is able to count them by heart in a short period of time. The voice output is adapted to be interesting for the pupil's target group or accompanied by a physical activity that will attract the child and make the exercise more pleasant. The robot is able to identify human faces in its surroundings and, based on that, give examples to specific people.

The theoretical part of the bachelor thesis describes the existing educational robots and robotic toys that are useful in other school subjects than ICT. The practical part deals with the creation of the program.

Keywords

Robot, Nao, Choregraphe, mathematics, practising, programming, pupils

Poděkování

Chtěla bych poděkovat Mgr. Václavu Šimandlovi, Ph.D. za odborné vedení mé práce a cenné rady, které mi poskytl, ale také za trpělivost a ochotu, se kterou se mi věnoval.

Obsah

1	Úvod	11
1.1	Cíle práce	11
1.2	Metody práce	11
2	Programovatelní roboti na trhu	13
2.1	Humanoidní roboti	13
2.1.1	Walker (UBTECH Robotics)	13
2.1.2	Asimo (Honda)	14
2.2	Roboti srovnatelní s Nao	15
2.2.1	Pepper (Softbank Robotics)	15
2.2.2	Manav (A-SET Robotics)	16
2.2.3	Robotis Premium - Bioloid (Robotis Co Ltd)	16
2.2.4	Robotis Mini (Robotis Co Ltd)	18
2.2.5	Lego Mindstorms	19
2.3	Shrnutí	20
3	Robot Nao	21
3.1	Hardwarové vlastnosti robota Nao	21
3.1.1	Pohyb	21
3.1.2	Zvuková zařízení	22
3.1.3	Vizuální zařízení	22
3.1.4	Senzory	23
3.2	Softwarové vlastnosti robota Nao	25
3.2.1	Dialog	25
3.2.2	Analýza lidské tváře	26
3.2.3	Paměť	28
4	Prostředí Choregraphe	29
4.1	Práce s prostředím Choregraphe	29
4.2	Boxy	30
4.2.1	Rozdělení boxů	30
4.2.2	Propojování boxů	31

4.2.3	Vstupy	31
4.2.4	Výstupy	32
4.3	Virtuální robot	32
4.4	Nabízené funkce	33
4.5	Tvorba vlastních funkcí	34
5	Návrh programu	35
5.1	Analýza požadavků	35
5.2	Návrh implementace stanovených požadavků	36
5.3	Diagram programu	37
5.3.1	Základní schéma	37
5.3.2	Úplné schéma	38
6	Tvorba programu	40
6.1	Tvorba dílčích částí	40
6.1.1	ALMemory	40
6.1.2	Funkce podmínky	41
6.1.3	Převod řeči na text	41
6.1.4	Vyslovení textu	43
6.1.5	Tvorba dialogu	44
6.1.6	Generování náhodné hodnoty	45
6.1.7	Matematické operace	45
6.2	Výsledné funkce	45
6.2.1	Výpočet příkladu	46
6.2.2	Vyslovení příkladu	49
6.2.3	Poslouchání výsledku od žáka	50
6.2.4	Kontrola výsledku	51
6.2.5	Zábava	51
6.3	Shrnutí poznatků z praktické části	53
6.3.1	Práce s robotem	53
6.3.2	Práce s Choregraphe	54
7	Ověření použitelnosti	58

8 Závěr	59
Seznam použité literatury a zdrojů	66
Seznam obrázků	67
Seznam příkazů	68
A Přílohy	69

1 Úvod

1.1 Cíle práce

Cílem mé práce bylo naprogramovat robota NAO pro procvičování jednoduché matematiky ve skupině žáků 1. stupně základní školy. Robot je schopen pomocí svého hlasového výstupu dávat dětem matematické příklady a jejich správnou či nesprávnou hlasovou odpověď odměnit odpovídající reakcí.

Příklady procvičují sčítání, odčítání, násobení a dělení. Vytvořený program je možno použít k procvičování nejen jedné konkrétní matematické operace, ale také všech typů příkladů v náhodném sledu za sebou. Náročnost příkladů je taková, aby dítě bylo schopno je spočítat z paměti v krátkém časovém intervalu. Hlasový výstup je přizpůsoben, aby byl pro dětskou cílovou skupinu zajímavý a je doprovázený pohybovou aktivitou, která dítě zaujme a zpříjemní procvičování. Robot také identifikuje ve svém okolí lidské tváře a na základě toho pokládá příklady konkrétním lidem. Reaguje také na výrazy nudím se, nuda, mě nezkoušej a já jsem učitel. Ačkoliv využitelnost vytvořeného programu měla být opakovaně ověřena v dětském kolektivu, vzhledem k nepříznivému vývoji epidemiologické situace muselo být od tohoto cíle upuštěno.

1.2 Metody práce

Při tvorbě programu jsem využívala software Choregraphe, jehož grafické rozhraní umožňuje graficky uspořádat jednotlivé události a propojovat jejich výstupy s dalšími událostmi, které mají následovat. Samotné události je možno upravovat, ať už jejich vstupní i výstupní parametry, či přímo zdrojový kód v jazyce Python.

Postupně jsem z dílčích celků vytvořila kompletní program, který byl opakovaně testován na robotovi. Program využívá mimo jiné funkci rozpoznávání obličejů, která je zásadní pro možnost pokládat příklady konkrétní žákům a tyto žáky střídát. Mimo jiné jsem vytvořila šest zábavných aktivit pro pobavení žáků a povzbuzení jejich zájmu o robota. Aktivity jsou tvořeny částečně z pohybů obsažených ve vývojovém prostředí a částečně z webové stránky Univerzity Notre Dame [1] nabízející několik krátkých pohybů k volnému použití. Tyto části jsou doplněny o několik mnou vytvářených pohybů a následně poskládány do smys-

luplných celků, některé doplněny hudbou získanou k volnému použití v rámci předplatného.

Dále jsem využila Python knihovnu Text to speech, která umožňuje přehrávat text jako zvukový výstup, což je nezbytné pro komunikaci robota s žáky. Úlohy se vytvářejí ze dvou náhodně vygenerovaných čísel ve zvoleném rozsahu a mezi ně je vloženo matematické znaménko, které je právě procvičováno. Při tvorbě i testování jsem brala v úvahu znalosti i projev mladších žáků, proto jsou výpočty příkladů zvládnutelné z paměti a odpovědi v průběhu programu jsou očekávány krátké, převážně jednoslovné.

Testování jsem z důvodu protiepidemických opatření v souvislosti s šířením onemocnění COVID-19 prováděla pouze sama.

2 Programovatelní roboti na trhu

V dnešní době je na trhu velké množství různých robotů. Nejen kuchyňský robot, který se z principu nedá úplně považovat za robota, ale právě mnoho robotů s vlastním chováním a iniciativou, kteří jsou schopni situaci vyhodnocovat a reagovat na podněty od člověka. Takových robotů je mnoho druhů. Mohou to být domácí spotřebiče, jako jsou robotické vysavače nebo sekačky. Také známe různé hračky pro děti v podobě postavičky nebo psa. Nyní se dostávají do povědomí i roboti, kteří roznášejí zásilky nebo pečují o starší osoby.

2.1 Humanoidní roboti

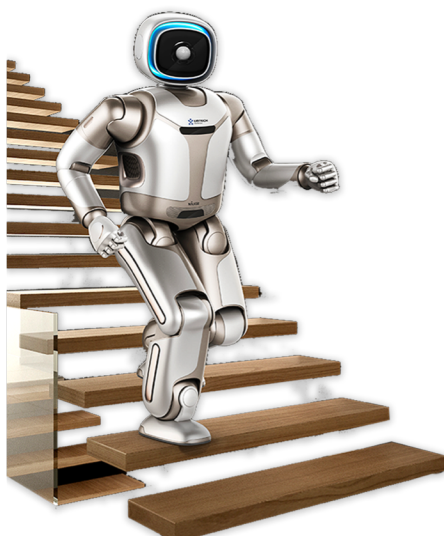
Robotů humanoidního typu je na trhu několik, většinou jsou to modely určené spíše pro další vývoj než pro prodej. Přestože někteří jsou vyvíjeni i déle než 15 let, jejich vývoj ještě není u konce. Jsou cíleni většinou do zdravotnictví, pečovatelských domů nebo do služeb. Takovým příkladem může být roznášení zásilek, případně robot, který vám umyje gril nebo podá toaletní papír. Vyvinout takto rozsáhlé autonomní chování, které by splnilo potřeby člověka a obstálo před nástrahami běžného prostředí, je velmi náročné.

2.1.1 Walker (UBTECH Robotics)

Tento robot je 1,45 m vysoký a váží 77 kg. Poslední model vyšel v roce 2019. Z pohledu vybavenosti Walker disponuje rozpoznáváním tváří, analýzou postoje a nálady člověka. Je schopen rozpoznat a sledovat předměty a komunikovat s uživatelem podobně jako robot Nao. Je cílen do domácnosti a pečovatelských zařízení. Mezi jeho přednosti patří velmi jemné a přesné pohyby, které jsou ve srovnání s ostatními roboty velmi plynulé. Také je to jeden z mála dvounohých robotů zvládající chůzi po schodech. Díky jeho plynulým pohybům, obdivuhodné vyváženosti při chůzi a jemné motorice, dokáže přenést sklenici s vodou, otevřít dveře, hrát na klavír, kreslit fixem, přelévát kapalinu mezi sklenicemi a mnoho dalšího [2, 3].

Jinak je vybaven podobně jako Nao. Rozpoznává tváře, mapuje okolí svou kamerou a komunikuje prostřednictvím hlasu. Velkou nevýhodou je, že Walker

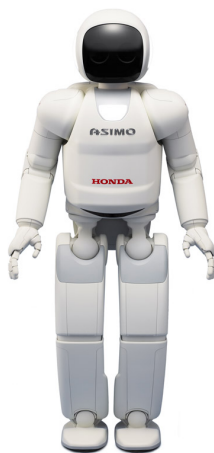
v současné době neumožňuje propojení s osobními asistenty jako například Alexa, ale jeho výrobci připravují vlastního asistenta na míru tomuto robotovi [3].



Obrázek 1: Robot Walker [4]

2.1.2 Asimo (Honda)

Robot Asimo je podobně vybaven jako již zmíněný Walker. Je možné ho ovládat hlasem a mluvit s ním, rozpoznává lidské jednání a řeč. Asimo se ale liší především tím, že byl vyroben v roce 2000, což je o 19 let dříve než Walker. Další specialitou je možnost ovládat tohoto robota myšlenkou pomocí helmy, kterou má uživatel nasazenou [5, 6].



Obrázek 2: Robot Asimo [5]

2.2 Roboti srovnatelní s Nao

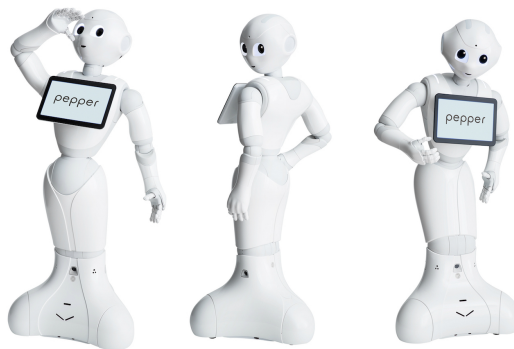
V této kapitole popisují roboty, kteří mají podobné možnosti jako robot Nao. Především tedy možnost programovatelnosti a prostředí vhodné pro výuku a méně zkušené programátory.

2.2.1 Pepper (Softbank Robotics)

Tohoto robota nemohu opomenout, a to nejen proto, že mají s robotem Nao společného výrobce. Dalším společným znakem programovatelnost obou robotů. Tento robot byl představen v roce 2015 a je cílen především pro informování a komunikaci s návštěvníky v obchodech, galeriích, veletrzích apod. Mimo to našel využití i v běžných domácnostech a školách, a to pro výzkum, ale i výuku. Pepper vyniká především svou schopností detekovat výrazy, nálady a emoce z tváře. Rozpoznává i tón hlasu člověka, a díky této schopnosti snadno naváže dialog s uživatelem a snadno si ho získá během několika okamžiků [7, 8].

Z pohledu vybavenosti, nabízí podobné možnosti jako Nao. Rozdílem je, že Pepper se pohybuje pomocí podvozku se třemi kolečky.

Zajímavostí je, že kvůli propuknutí nemoci COVID-19 byla vydána aktualizace, která Pepperovi umožňuje hlídat, zda mají lidé nasazenou ochranu úst. Tato funkce je žádaná nejen v obchodech, ale také na letištích. Jeden Pepper hlídá dodržování opatření i na Letišti Václava Havla [9].



Obrázek 3: Robot Pepper [10]

2.2.2 Manav (A-SET Robotics)

Tento robot byl vyvinut v Indii v roce 2015 a jedná se o prvního robota vytištěného na 3D tiskárně. Robot je cílen na soukromou sféru, kde se počítá s jeho přizpůsobením na míru, ale také již našel využití ve vzdělávání a výzkumu. Z pohledu programovatelnosti je jeho software napsán v jazyce C++, což není příliš vhodné pro méně zkušené uživatele. Také nemá žádné přizpůsobené vývojové prostředí, čímž se hodí především na odborné katedry vysokých škol. Robot je schopen dialogu, několika tanečků, aktivit a v rámci jiného projektu byl naprogramován pro autonomní fotbalový zápas robotů. Robot také vyniká svou cenou, která v přepočtu začíná na 45 000 Kč [11, 12].

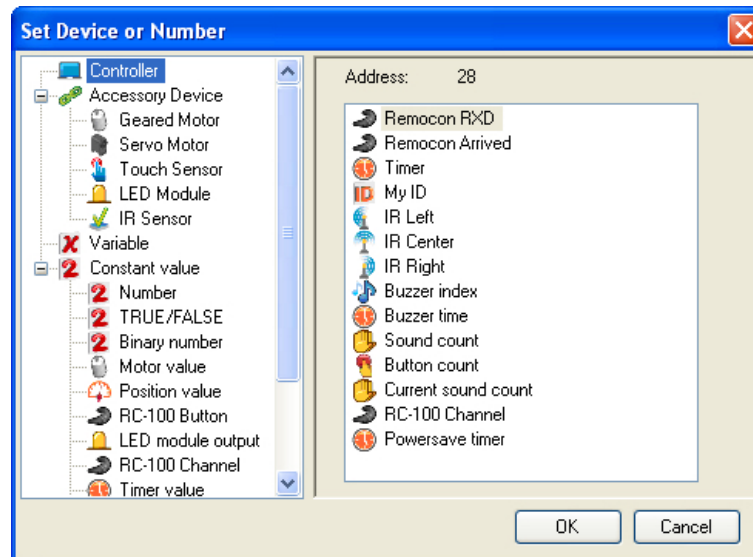


Obrázek 4: Robot Manav [12]

2.2.3 Robotis Premium - Bioloid (Robotis Co Ltd)

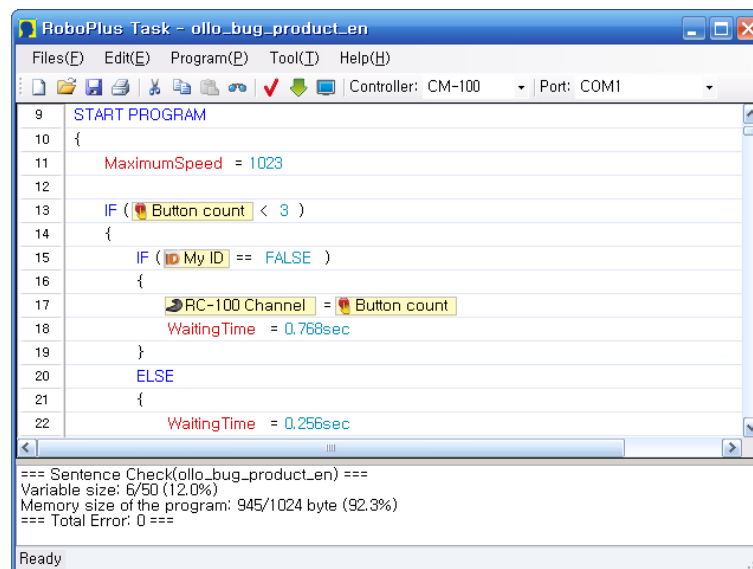
Jedním z mála programovatelných robotů, kteří by měli vhodně upravené vývojové prostředí je Bioloid. Robot se skládá z lehkých a pevných hliníkových rámců. Má velmi dobrou mobilitu, rychlou chůzi díky gyroskopu, je schopen mnoha autonomních póz a postojů. Obsahuje také mnoho senzorů [13].

Programovací software RoboPlus sice nevyniká svým designem, ale obsahuje funkce a zkratky, které stáčí vložit a není nutné je programovat ručně (viz obrázek 5) [13].



Obrázek 5: Software RoboPlus [14]

Software obsahuje také jakési šablony pro programování samotného kódu, jako například šablonu cyklu včetně popisu částí. Samotné programování jednotlivých skriptů působí přehledně a v příručce je i několik příkladů a vzorových kódů (viz. obrázek 6) [13].



Obrázek 6: Kód v programu RoboPlus [14]

Další výhodou je možnost rozšíření o další senzory a komponenty, čímž lze robota přestavět na jiný tvar a programovat ho k novým účelům [15].



Obrázek 7: Přestavby robota Bioloid [15]

2.2.4 Robotis Mini (Robotis Co Ltd)

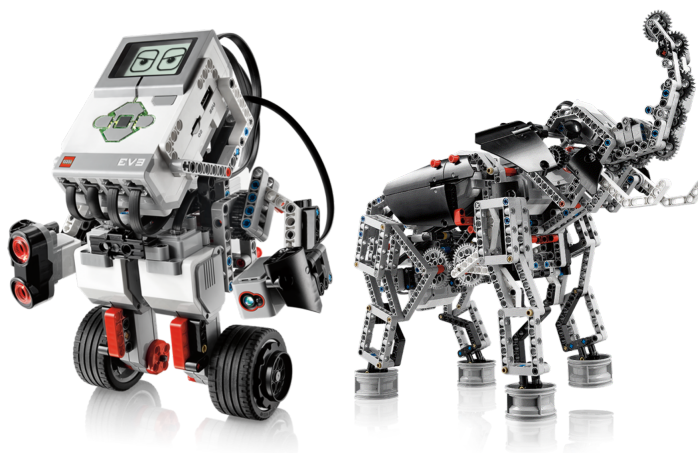
Další zajímavou možností je Robotis Mini, který je také programovatelný stejně jako Bioloid. Oba také pocházejí od stejného výrobce. Robotis Mini lze pořídit za zhruba čtvrtinu ceny Bioloida. I přes nižší cenu ale nabízí příjemné a moderní vývojové prostředí R+ Task 2.0, a to pro počítače i mobilní telefony. Robot je vysoký asi 27 cm a z pohledu vybavenosti nabízí senzory barev, teploty, náklonu, doteku, ale také infračervený senzor. Dále umožňuje rozpoznání hlasu a objektů [16].



Obrázek 8: Robotis Mini [16]

2.2.5 Lego Mindstorms

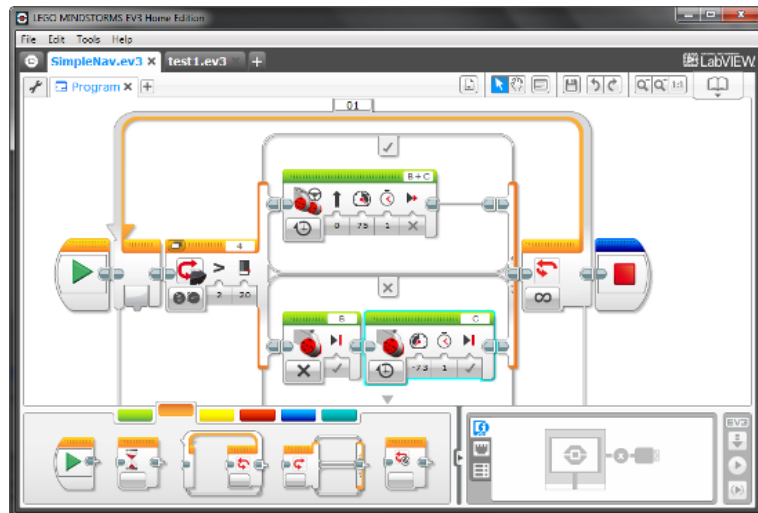
Podobné přestavby jako Bioloid umožňuje i Lego Mindstorms. Jedná se o velmi propracovanou stavebnici nabízející mnoho přestaveb dle manuálů i mimo něj. Obsahuje programovatelný Brick sloužící pro uložení a volání programu. Brick také napájí veškeré motory a senzory a přebírá od nich data [17].



Obrázek 9: Lego Mindstorms [18]

Velkou výhodou Mindstormu je nízká cena, která se pohybuje do 10 tisíc korun za základní balíček obsahující Brick a velké množství komponentů. Velmi příjemné jsou také obrázkové návody na mnoho různých sestav i materiály v češtině dostupné na internetu [17].

Asi nejvíce propracované je vývojové prostředí. Na spodním okraji je nabídka dostupného hardwaru, který lze ovládat, ale také programovací metody, jako jsou podmínky, cykly, proměnné, start a stop programu a další. Vše lze navzájem kombinovat a vnořovat, a i když program působí hravým dojmem, je možné vytvořit rozsáhlé projekty, které potrápí i zkušenější studenti na dlouhé hodiny. V rámci výuky nikdy nedojdou zajímavé úkoly, protože i samotná přestavba na jiný tvar může být náročná [18].



Obrázek 10: Lego Mindstorms software [18]

2.3 Shrnutí

Při hledání dalších robotů s podobnými vlastnostmi jako robot Nao jsem nenašla přímou konkurenci. Vynechala jsem modely, jejichž cena se pohybuje kolem milionu korun a výše, jako je robot Sophia. Roboti humanoidního vzhledu nebývají programovatelní nebo jen výjimečně. Většina běžně se prodávajících robotů vyniká spíše nízkou cenou než důmyslnou konstrukcí. Také nutno dodat, že nízká cena je obvykle vykoupena nepřizpůsobeným nebo neexistujícím vývojovým prostředím, čímž bude trpět každý, kdo chce robota programovat.

Podobně kvalitní provedení má robot Pepper, který využívá i stejné vývojové prostředí jako Nao. Pepper je ovšem určen spíše pro komerční účely. Pokud by se jednalo o výuku programování na školách, kde je potřeba nakoupit více strojů za přiměřené náklady, jasnou volbou je Lego Mindstorms nebo Robotis Mini.

3 Robot Nao

Robot Nao je autonomní, programovatelný robot ze skupiny humanoidů neboli robotů s lidskou podobou. Jeho původním vývojářem je Aldebaran Robotics z Francie. Od roku 2015 je Nao společně s dalším modelem nazývajícím se Pepper prodáván pod společností SoftBank Robotics.

Jeho vývoj začal již v roce 2004 a je určen nejen pro školy a univerzity, ale je hojně využíván i ve výzkumných centrech a ve zdravotnictví.

Pro svou práci mám k dispozici robota Nao 6. verze, vydaného v roce 2016.

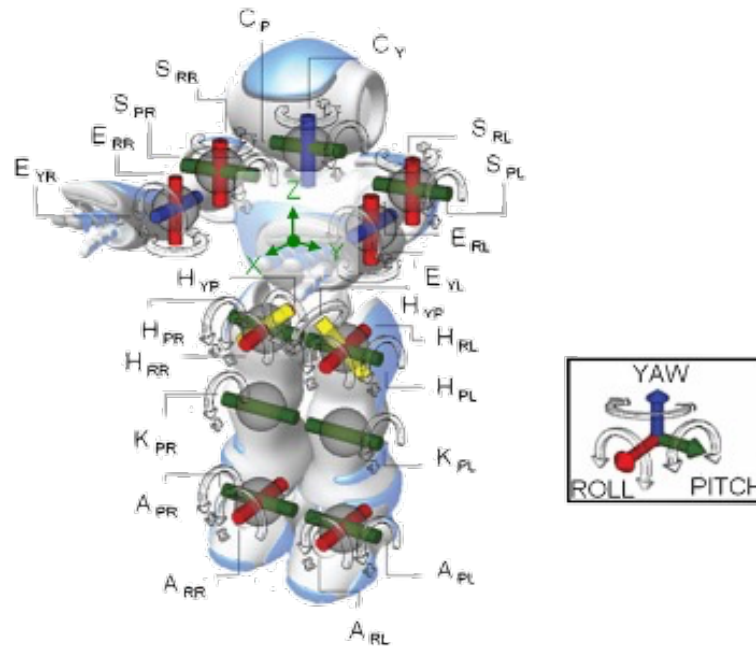
3.1 Hardwarové vlastnosti robota Nao

Nao je vysoký 57,4 centimetru a váží 5,5 kg. Doba provozu se dle manuálu pohybuje někde mezi 45 minutami a 2 hodinami v závislosti na náročnosti používání. Z pohledu konektivity nabízí Ethernet RJ45 a USB port, oba skryté v zadní části hlavy. Také je možná komunikace přes WiFi v předem zvolené síti [19, 20].

3.1.1 Pohyb

Pohyb robota je realizován motory. Důležitou součástí jsou také snímače polohy kloubů. Tyto snímače poskytují informace o pohybu kloubních spojů, zejména o kolik stupňů se spoj pootočil. Tento senzor využívá magnetický rotační enkodér, jehož princip spočívá v posunu několika magnetů vůči snímači. Senzor použitý v robotovi je 12bitový s přesností odpovídající přibližně $0,1^\circ$ [21].

V robotovi je použito 5 různých typů motorů, liší se především rychlostí a točivým momentem. Dále se dělí ještě na dvě kategorie dle omezení maximální rychlosti. Z pohledu pohyblivosti se většina spojů může pohybovat ve dvou osách jako například spojení hlavy a krku. Hlava se tak může jednat horizontálně, také pomocí náklonu vůči vertikální ose lze hlavou například kývat. Výjimkou jsou robotova kolena, která umožňují pohyb pouze v jedné ose nebo také spoj kyčlí schopný se pohybovat ve všech třech osách, a to především při chůzi, kde je nutné přesné balancování při přenášení váhy mezi končetinami [22]. Jednotlivé osy pohybu pro každý spoj jsou znázorněny na obrázku 11.



Obrázek 11: Osy pohybu robota Nao [23]

3.1.2 Zvuková zařízení

Robot je vybaven stereo systémem se dvěma reproduktory po stranách jeho hlavy. Dále obsahuje čtyři všesměrové mikrofony. Dva jsou umístěny na vrchu hlavy a dva na zadní straně [24, 25].

3.1.3 Vizualní zařízení

Z pohledu vizuálních zařízení jsou zde k dispozici dvě kamery s podobnými vlastnostmi. První kamera je umístěna na čele robota a umožňuje rozlišení v rozsahu od 320 x 240 px do 2560 x 1920 px. Druhá kamera se nachází na místě pusy a její maximální rozlišení je pouze 1280 x 960 px [26].

Rozlišení	Frame rate	Dostupné pro kamery
2560 x 1920 px	1-15	Pouze horní kamera
1280 x 960 px	1-30	Obě
640 x 480 px	1-30	Obě
320 x 240 px	1-30	Obě

Tabulka 1: Rozlišení kamer ve vztahu k počtu snímku za sekundu

LED diody Dále je robot osazen velkým množstvím LED diod, a to na hlavě, uších, očích a na nártech nohou.

Na vrchní části hlavy se nachází 12 diod po obvodu dotykových ploch. Tyto diody jsou schopny svítit jen bílou barvou v 16 různých intenzitách.

Další skupina diod se nachází na bočních stranách hlavy. Po obvodu každého z reproduktorů je umístěno 10 modrých diod.

Nejvýraznější diody jsou vidět v očích. Zde je 8 RGB diod do tvaru prstence v každém oku.

Poslední skupinou jsou diody na nártech nohou. Na každém nártu se nachází jedna RGB dioda, která se rozsvěcí například při stisknutí dotykové plochy na špičce nohy [27].

3.1.4 Senzory

Senzor tlaku Asi nejvíce specifickým senzorem je senzor tlaku v chodidlech. Celým názvem Force Sensitive Resistors. Tyto senzory jsou celkem čtyři, každém rohu chodidla jeden a udávají hodnotu v rozmezí 0 – 25 N dle odporu v závislosti na změně tlaku [28].

Inerciální jednotka Tato jednotka je umístěna v trupu robota spolu s vlastním procesorem a měří hodnotu náklonu relativní k poloze trupu. Výstupní data z této jednotky umožňují vypočítat rychlost pohybu robota a jeho náklon vůči svislé ose.

Výpočet probíhá díky údajům ze dvou senzorů. Akcelerometr poskytuje úhel trupu ve statickém režimu. Když je detekován nějaký pohyb, je výstupní úhel vypočítán pomocí gyrometrů. Protože rychlost pohybu je proměnlivá, je nutné vypočítat úhel z akcelerometru a gyrometru. Oba tyto senzory měří ve třech osách a k oběma těmto hodnotám je možné přistupovat pomocí ALMemory [29].

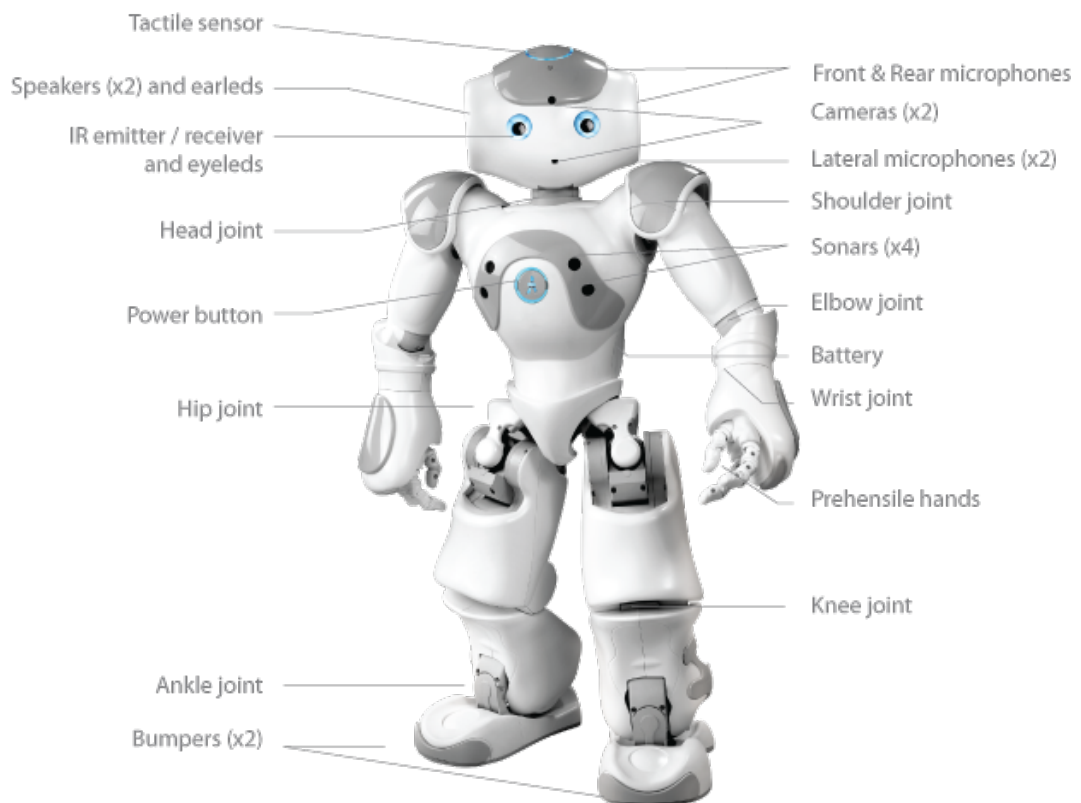
Sonar Sonar nebo také ultrazvukový senzor umožňuje robotovi, aby se vyhnul překážkám v jeho přirozeném prostředí. Tuto funkci zajišťují dva vysílače a dva přijímače ultrazvukových signálů na jeho hrudníku. Je schopen detekovat překážku v rozmezí od 20 do 80 cm v úhlu 60° [30].

Kontaktní a taktilní senzory Lze je nazývat také senzory dotyku, na robotovi se nachází na čtyřech místech. Na vrchu hlavy se nacházejí tři dotykové plochy, které mohou sloužit jako tlačítka. Nemají fyzický krok při kliknutí, reagují pouze na dotek. Na tyto tlačítka lze naprogramovat libovolnou odezvu během programu, například dotaz na stav baterie.

Dalším senzorem dotyku jsou šedé plochy na hřbetu rukou. Tyto senzory lze použít například pro vedení robota za ruku.

Na hrudi robota se nachází klasické fyzické tlačítko s nápisem Nao. Tímto tlačítkem se robot zapíná a také slouží pro přehrání IP adresy, na které se lze k robotovi připojit.

Posledním senzorem jsou nárazníky na přední části nohou. Tento senzor reaguje například, když při chůzi robot kopne do překážky. Při aktivaci tohoto senzoru se zároveň rozsvítí dioda na nártu nohy [31].



Obrázek 12: Části robota Nao [32]

3.2 Softwarové vlastnosti robota Nao

Robot Nao je nejen seskupení součástí, ale také obsahuje důmyslný software, který mu umožňuje fungovat. V této podkapitole zmíním jen funkce robota, které přímo souvisí s mým programem. Mimo zde popsané umí také rozpoznávat objekty, reprezentovat slova, sledovat vybrané tváře a objekty, autonomně se hýbat během nečinnosti a mnoho jiného.

3.2.1 Dialog

Funkce dialogu je pro robota klíčová právě z důvodu, že umožňuje komunikovat s libovolným publikem. Lze předem zadat otázky, na které se uživatel bude ptát a robotovi připravit odpovědi, případně přiřadit celé téma.

Syntaxe dialogu probíhá v samostatném rozhraní nazývaném QiChat, které má i vlastní pravidla a mnoho samostatných a intuitivních funkcí.

V příkladu 1 je vzor dialogu z oficiální dokumentace. V prvním řádku se uvede název tématu, aby robot k tématům mohl přistupovat dle názvu. V druhém řádku je uveden jazyk, který má robot nejen sám využít pro svůj projev, ale také jazyk, kterým bude mluvit uživatel a robot bude výrazy v rámci zvoleného jazyka rozpoznávat [33].

Dále je zde koncept, což je definování otázky a odpovědi nebo lépe definování reakce robota v závislosti na projevu uživatele. V prvním konceptu jsou definovány pozdravy. Výraz `greetings` je název reakce, k níž se budeme odkazovat níže. V hranatých závorkách jsou uvedeny výrazy, které robot může v rámci pozdravů používat. Slovo `rand` před závorkou umožňuje robotovi volit pozdravy náhodně. Stejným způsobem jsou definovány i ostatní dva koncepty s názvem `vino` a `alcohol`. Tyto dvě skupiny jsou vnořeny do sebe. Ve skupině `vino` robot vždy vybere červené nebo bílé a přidá ke zvolenému druhu slovo `vino`. Ve skupině `alcohol` je robot schopen nabídnout buď pivo nebo nahlédne do skupiny `vino`, kde zvolí „bílé + `vino`“ nebo „červené + `vino`“ [33].

Níže v kódu vidíme volání jednotlivých témat. Na počátku robot čeká, než uslyší některý ze zvolených pozdravů. Pokud pozdrav rozpozná, pozdraví také. Poté čeká, že se uživatel zeptá „Do you have“ + cokoliv z kategorie `alcohol` včetně

vnořené kategorie víno. Robot odpoví „Ano, mám“ a doplní nápoj, na který se uživatel ptal [33].

```
topic: ~introduction ()
language: English

concept:(greetings) ^rand[hi hello "hey there"]
concept:(wine) [red white] wine
concept:(alcohol) [beer ~wine]

u:(~greetings) ~greetings
u:(do you have _~alcohol) yes, I have $1
u:(I want to drink something) do you want ~alcohol?
```

Příklad 1: Zdrojový kód dialogu

```
> hey there
hello
> do you have white wine?
yes, I have white wine
> I want to drink something
do you want beer?
> I want to drink something
do you want red wine?
```

Příklad 2: Výsledný dialog

Tímto způsobem s mnoha dalšími funkcemi a usnadněními lze vytvořit dojem vědomého dialogu s robotem. Dialog umožňuje různé ukládání do proměnných, volání pohybů jako je posazení, vstání, jednotlivá gesta, přehrání zvuků a další. Také je možné, aby robot dialog zahájil a jako první oslovil uživatele [33].

3.2.2 Analýza lidské tváře

Robot je schopen rozpoznat mnoho informací z tváře člověka, tvář si zapamatovat a pojmenovat, což je jistě funkce, která zaujme mnoho pozorovatelů. Tato analýza probíhá pomocí API ALFacesCharacteristics a při tvorbě na úrovni API lze využít skóre spolehlivosti v rozsahu od 0 do 1 k vyfiltrování nejistých výsledků z provedené analýzy [34].

Detekce výrazu ve tváři Detekce analyzuje mimiku tváře a snaží se odhadnout, zda je tvář neutrální, šťastná, překvapená, našťvaná nebo smutná [35].

Detekce nálady Tato funkce se liší od předchozí v tom, že nepozoruje výraz celého obličeje, ale pravděpodobně je více zaměřená na ústa. Také je umístěna v samostatném API s názvem ALMood přesto, že se jedná o funkci, která pracuje s tváří. Posledním rozdílem je, že tato funkce rozlišuje pouze tři hodnoty (neutrální, šťastná, nebo smutná). Pravděpodobně dle tvaru rtů, zda jsou rty prohnuté do úsměvu, naopak, nebo jsou rovné.

Detekci nálady i výše zmíněnou detekci tváře je vhodné propojit s další reakcí, například „Myslím si, že jsi smutný. Zkus se trochu usmát.“ a případně připojit zábavnou aktivitu [36].

Rozpoznání věku V rámci této funkce robot odhadne, kolik je uživateli let. Robot se na uživatele zadívá několik vteřin, spíše desítek vteřin, tvář analyzuje a poté řekne „Myslím si, že ti je * let.“. V návaznosti na rozpoznanou hodnotu lze navázat další projev k tomuto výsledku. Především je nutné nebrat výsledný věk příliš vážně, robot často hádá vyšší věk [37].

Určení pohlaví Tato funkce probíhá podobně jako předchozí zmíněné. Robot chvíli pozoruje tvář a poté řekne „Myslím si, že jsi (muž/žena).“ Může i dojít k situaci, že robot pohlaví nerozpozná.

Časté obtíže Výsledky detekce mohou být nepřesné vlivem světelných podmínek a fyzických překážek jako jsou brýle nebo čepice.

Na odhad věku mohou mít vliv také deformace rysů obličeje, například výrazný úsměv nebo úšklebek.

Odhad pohlaví u malých dětí může být také méně přesný než u dospělých, protože mají méně výrazné rysy obličeje.

Výsledky analýzy mohou být s další detekcí odlišné, protože jsou pokaždé počítány na různých obrázcích obličeje. To lze očekávat a například odhad věku je vhodné rozdělit do věkových kategorií (například dítě, dospělý, starší), čímž nebudou rozdíly tolik patrné.

Je vhodné brát v úvahu, že odhad výrazu obličeje není schopen rozpoznat příliš jemné výrazy obličeje [34].

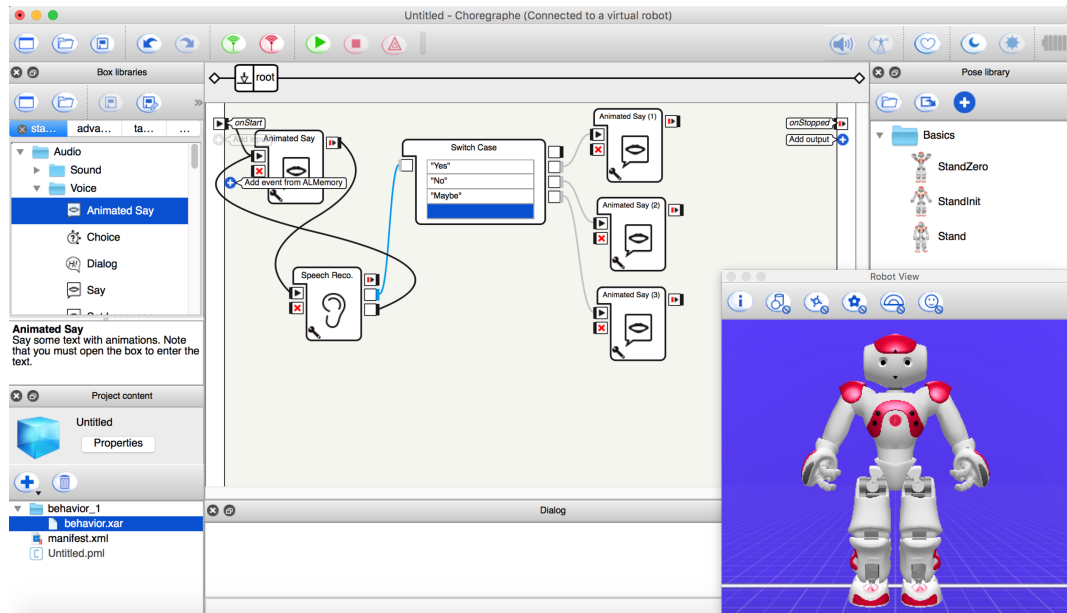
3.2.3 Paměť

Robot obsahuje jednu centrální paměť uchovávající všechny aktuální data a informace klíčové pro běh robota, například o poloze trupu, úhlech jednotlivých kloubů, veškerá data ze snímačů a další.

V přizpůsobeném použití za pomoci další funkce lze v této paměti uchovávat také data inicializovaná uživatelem nebo během uživatelského programu. Tyto údaje jsou pojmenovány a uchovávány ve tvaru key – value. Právě zmíněný klíč je tvořený jedním slovem nebo složeninou slov bez mezer a speciálních znaků. Pod tímto klíčem jsou data v paměti uchována a je k nim na základě klíče přístupováno. Lze vytvářet nové klíče s daty (pokud již existuje, bude přepsán), také lze data načítat a klíč i s daty smazat [38].

4 Prostředí Choregraphe

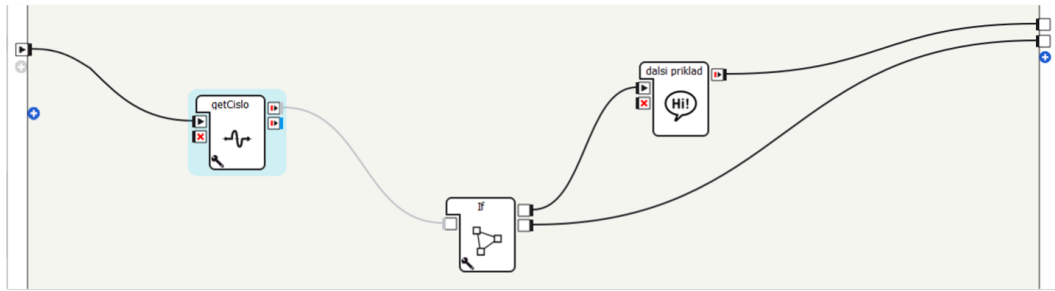
Choregraphe je multiplatformní desktopová aplikace, která umožňuje ovládat a programovat robota. Nabízí tvorbu chování, dialogů, animování jednotlivých pohybů, programování vlastních skriptů v jazyce Python a mnoho dalšího. Vytvořené projekty je možné testovat bez přítomnosti robota pomocí funkce virtuálního robota [39].



Obrázek 13: Software Choregraphe [40]

4.1 Práce s prostředím Choregraphe

Program jsem vytvářela ve vývojovém prostředí Choregraphe verze 2.8.6.23 pro robota Nao V6. Toto prostředí je velmi intuitivní a přehledné. Tvorba základních částí programu probíhá v grafickém prostředí, které znázorňuje jednotlivé části programu a návaznosti mezi nimi. Souvislosti mezi těmito částmi jsou znázorněny spojnícemi, které zobrazují nejen směr, kterým se program může ubírat, ale při běhu programu je vidět, která funkce je právě volána, což usnadňuje orientaci nejen při tvorbě rozsáhlejších projektů, ale především při odstraňování chyb.



Obrázek 14: Grafické znázornění funkcí

4.2 Boxy

Grafické prostředí softwaru se skládá z boxů, které tvoří elementární část chování robota. Box může obsahovat jednu funkcionalitu, ale i více funkcí, které společně tvoří logický celek. Takovým příkladem může být box Say, který zajišťuje vyslovení vloženého textu [41].

4.2.1 Rozdělení boxů

Podle obsahu můžeme boxy rozdělit do čtyř kategorií.

- Python box – obsahuje skript v jazyce Python
- Flow diagram box – obsahuje skupinu boxů
- Timeline box – obsahuje časovou osu pro pohyby a chování robota
- Dialog box – odkazuje na téma dialogu, který se zahájí

Python box Python box obsahuje skript v jazyce Python. Tento skript většinou obsahuje soubor příkazů, který má za úkol vykonat jednu určitou funkci. Například vstát, posadit se, popojít, udělat gesto rukama nebo rozpoznat obličej. Zdrojový kód těchto funkcí je možné libovolně editovat, rozšiřovat a importovat balíčky, které Python nabízí [42].

Flow box Flow diagram box obsahuje skupinu boxů, které jsou vzájemně propojeny alespoň jednou spojnicí. Tento box při vytvoření přejímá vstupy a výstupy, které jsou použité uvnitř a slouží především pro větší přehlednost a snadnější orientaci v programu. Je vhodné vložit do zmíněných boxů samostatné části programu jako např. dialog, který je doprovázen větším množstvím gest a pohybů.

Ve světě programování lze flow diagram přirovnat k metodě, která obsahuje jednotlivé příkazy, které spolu kooperují [43].

Timeline box Timeline box umožňuje díky časové ose synchronizovat volání jednotlivých boxů a k tomu robotovy pohyby. Také slouží k tvorbě pohybů a gest. Aktuální postoj robota lze zaznamenat do časové osy a sestavu pohybů posléze přehrát jako robotovo gesto nebo pohyb [44].

Dialog box Dialog box je posledním druhem a slouží pouze k zahájení dialogu ve zvoleném jazyce. Obsah samotného dialogu je nutné předem vytvořit jako téma přes rozhraní QiChat. Toto rozhraní má vlastní pravidla i syntaxi a nabízí mnoho funkcí, díky kterým lze vytvořit dojem vědomého dialogu ze strany robota. Robot se např. zeptá, o čem si chce uživatel povídat a poté používá náhodně vybrané, předem zadané odpovědi jen v rámci zvoleného tématu [45].

4.2.2 Propojování boxů

Se spuštěním programu je inicializován bod na levém okraji okna a na základě spojnic mezi jednotlivými boxy dochází k inicializaci těchto boxů. Průběh programu je znázorněn svítivým bodem, který prochází po aktivních spojnících.

Mezi boxy je možné předávat několik typů impulzů. Nejjednodušším je Bang, který s sebou nenesení žádná data, pouze informaci o tom, že má být inicializován další box. Složitějším je Number, který obsahuje číselnou hodnotu z předchozího boxu ve formátu Integer, Float, anebo může být tvořen řadou čísel. Text je přenesen pomocí impulzu String a poslední je impulz Dynamic, který může obsahovat textová i číselná data, ale také je schopen fungovat i bez jakýchkoliv dat [46]. Stejně jako metody ve světě programování obsahuje každý box rozdílné množství vstupů a výstupů. V tomto prostředí jsou vstupy znázorněny na levé straně boxu a výstupy funkcí na pravé straně.

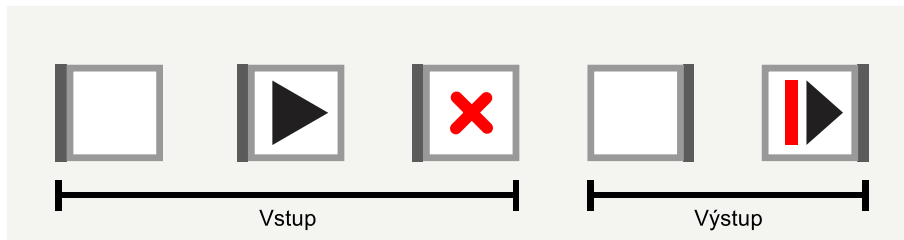
4.2.3 Vstupy

V tomto prostředí existují tři základní typy vstupů pro box. Inicializační vstup je znázorněn černým trojúhelníkem a nejen, že zahajuje běh programu uvnitř boxu,

ale je také schopen přijímat vstupní data. Tento vstup se nazývá onStart (na Obrázku 15 druhý symbol zleva).

Vstup s červeným křížkem se nazývá onStop a slouží pro zastavení běhu programu uvnitř boxu. Toto zastavení je vhodné například pro box rozpoznávání řeči (Speech recognition), který probíhá do doby, než zachytí některé z očekávaných slov. Pokud by tento box poslouchal dále a nebyl by ukončen, výrazně by to ovlivnilo následný běh programu (na Obrázku 15 třetí symbol zleva).

Třetím typem je vstup, který nemá žádný specifický vliv na box, pouze předává informaci o inicializaci a případná data, která s sebou impuls může nést [47].



Obrázek 15: Vstupy a výstupy boxů

4.2.4 Výstupy

Výstup se specifickým chováním je pouze jeden. Tento výstup se aktivuje, pokud je aktivita uvnitř boxu dokončena a již nepokračuje. Zároveň může předávat výstupní data dalším boxům. Nazývá se onStopped a je znázorněn červeným obdélníkem a černým trojúhelníkem (na Obrázku 15 zcela vpravo). Druhý typ výstupu nemá žádnou specifickou funkci nebo vlastnost. Nese pouze impuls pro inicializaci dalšího boxu a případně výstupní data pro následující funkci [47].

4.3 Virtuální robot

Virtuální robot je samostatný program, který se spouští společně s Choregraphe na počítači uživatele a simuluje téměř všechny možnosti a funkce fyzického robota. Virtuální robot dokáže simulovat veškeré pohyby, gesta, chůzi, výpočty, operace s proměnnými i odpovídající reakce a dialogy.

Hlavní výhodou této funkce je, že odpadá riziko pádu nebo poškození. Na druhou stranu některé funkce tento režim přeci jen nenabízí. Například v programu není nastavena gravitace, tedy není schopen simulovat pád při přílišném náklonu nebo nestabilní póze. Zda se robot převáží a upadne tedy zjistíme pouze se skutečným robotem.

Dále simulovaný režim sice umožňuje přehrání zvuku ve formátu mp3 nebo ogg, ale text, který robot vyslovuje je interpretován pouze písemnou formou do okna dialogu. Stejně tak hlasový vstup od uživatele není zaznamenán mikrofonom počítače, ale je nutné odpověď uživatele vepsat do okna dialogu jako běžný text, aby virtuální robot byl schopen reagovat na projev uživatele.

Poslední funkcí, která není při simulaci robota dostupná, jsou dotykové plochy umístěné přímo na robotovi. Tyto plochy se nachází na hlavě, pažích i nohách a je možné naprogramovat odezvu při stisknutí těchto ploch. Bohužel program neumožňuje používat tyto reakce a s virtuálním robotem je tedy nelze otestovat [48].

Přestože virtuální robot nenabízí přesné funkce jako fyzický robot, shledávám tuto možnost jako velký přínos pro uživatele, protože tak mají možnost vytvářet a z velké části i testovat program bez ohledu na dostupnost fyzického robota. Většinu programu jsem tak mohla vytvářet bez přítomnosti robota a stačily mi cca 4 testovací dny, které sloužily převážně pro hlubší otestování.

4.4 Nabízené funkce

Program Choregraphe také nabízí velké množství funkcí, které jsou zde již vytvořeny a které uživatel bez znalosti programovacích jazyků může využívat. Jak jsem se dočetla v dokumentaci, v prvních verzích robota Nao bylo nutné mnoho funkcí programovat ručně. V současné verzi je ale většina z nich vytvořena. Nabízené funkce jsou tříděny do sedmi hlavních kategorií na animace, projev, LED diody, multimédia, pohyby, senzory a programování.

Mezi nabízené funkce patří například mnoho gest, pohybů, ukládání dat do paměti, jednoduché matematické výpočty, rozpoznávání tváří, získávání provozních dat, přehrávání zvuků, ale také několik druhů chůze, kdy stačí zadat vzdálenost

a směr, kam má robot dojít. Všechny funkce je možné libovolně upravovat v jejich Python skriptu. Pohyby nabízí možnost úprav na jejich časové ose.

4.5 Tvorba vlastních funkcí

Vlastní funkce je možné vytvářet pomocí boxu Python skript v jazyce Python verze 2.7. V tomto boxu je již předepsáno několik metod označených slovem `def` a názvem metody. Jedním z těchto metod je `onInput_onStart`, která se spustí při inicializaci vstupu `onStart` nebo metoda `onStop`, která zastaví probíhající funkce uvnitř boxu a poté aktivuje výstup `onStopped`, který inicializuje následný box. Také je možné upravovat již existující funkce nebo spojit dvě funkce do jedné.

5 Návrh programu

V praktické části se zabývám tvorbou návrhu mého programu, který je cílem mé práce. Popisuji svůj postup při tvorbě programu, svá řešená úskalí i skripty použité v prostředí Choregraphe. Nejprve shrnuji požadavky na program a možnost tyto požadavky realizovat. Později ukazují postup při tvorbě programu.

5.1 Analýza požadavků

Při návrhu programu jsem nejprve zmapovala cílovou skupinu, pro kterou je program určen. Cílovou skupinou měli být žáci prvního stupně základní školy. Respektive žáci, kteří zvládají číselný rozsah, který robot procvičuje.

Dle osnov se u žáků na konci 3. ročníku ZŠ předpokládá znalost čísel do tisíce. Žáci dokáží sčítat a odčítat trojčíferná čísla, ve kterých jsou dvě čísla různá od nuly. (např. $205 + 350$) Ohledně násobení žáci automaticky užívají násobilku a násobí z paměti dvojčíferné číslo jednocíferným v jednoduchých případech. Co se týká dělení, zvládají dělit dvojčíferné číslo jednocíferným mimo obor násobitek a beze zbytku (příkladem $48 : 4$ a $56 : 4$) [49]. Bylo by tedy vhodné, aby uživatel mohl rozsah čísel jednoduše rozšířit až do tisíce, procvičování by tedy bylo přínosné i pro žáky 4. a 5. třídy.

Ohledně učiva probíraného ve 2. třídě základní školy. Žáci se učí sčítat a odčítat v oboru do 20 s přechodem přes desítku (příkladem $7 + 9$ a $16 - 8$) a také zvládají násobilku 2, 3, 4, 5 [50]. Pro toto použití bude v programu možné upravit rozsah čísel, ze kterých jsou příklady tvořeny.

Pro žáky 1. tříd je v osnovách uvedeno spíše procvičování orientace v číslech do 10 a 20 než vypočítávání příkladů [51], proto na ně nebude program cílen.

Robot by měl k žákům promlouvat, odpovídat jim, reagovat na jejich odpovědi a vytvářet tak dojem vědomého dialogu. Zároveň by měl pokládat příklady konkrétním žákům a tyto žáky střídat. Také by měl nabízet možnost zábavy, která žáky pobaví a upoutá jejich pozornost, aby procvičování nebylo příliš monotónní.

Program by měl být postaven tak, aby nebyl potřebný téměř žádný zásah ze strany učitele nebo vychovatele, který program spouští. Zároveň by robot měl

žáky zaujmout a procvičování by je mělo bavit, aby žáci robota neopustili příliš brzo.

5.2 Návrh implementace stanovených požadavků

Před samotnou implementací jsem zjišťovala, jak by robot pomocí svého hlasu pokládal matematické příklady a dokázal vyhodnotit správnost odpovědí z pohledu programovacího jazyka.

Příklady, které robot pokládá, jsou generovány pomocí funkce `Random` v jazyce Python. Vygenerují se dvě celočíselné hodnoty ve zvoleném rozsahu. Tento rozsah lze upravit podle znalostí žáků. Mezi tyto dvě hodnoty je doplněno znaménko dle procvičované operace a poté je příklad vysloven a spočítán.

Příklad, který je žákovi zadán, robot spočítá a uchová si výsledek. Tento výsledek poté porovnává s odpovědí žáka. V závislosti na tom, zda se tyto hodnoty shodují, robot reaguje buď pochvalou žáka, nebo naopak upozorněním na chybnou odpověď a sdělením správné odpovědi, kterou spočítal robot.

Také jsem brala v úvahu, že žáky procvičování příkladů po čase omrzí. Proto jsem zakomponovala i šest tanečků a zábavných prezentací, které mají žáky zaujmout a pobavit, aby robotovo působení nebylo příliš monotónní.

Při implementaci jsem narazila na možný problém v případě použití ve větší skupině žáků. Robotův mikrofón není příliš přesný při rozpoznávání řeči. Nejen, že hlasový projev je nutné omezit na pouhá slova, ale i odlišný akcent či intonace robotovi činí problémy. Také v případě, že by ve třídě byl přílišný hluk nebo velké množství rušivých zvuků, by se robot mohl při rozpoznání zmýlit v tom, co žák odpovídá, nebo by nemusel rozumět vůbec. Toto může nastat zejména u mladších žáků, kteří jsou často neklidní a nesoustředění. Stejný problém by mohl nastat, pokud by robot byl příliš vzdálený od žáka, nebo pokud by v místnosti byla přílišná ozvěna.

Těmto vlivům je nutné přizpůsobit uspořádání ve třídě, případně zmenšit velikost skupiny, která se procvičování účastní. Robot by měl být schopen spolupracovat s menší skupinou, např. 5–10 žáků. Je tedy vhodnější spíše do školní družiny než do běžné třídy při výuce. Použití ve vyučovací hodině se předpokládá

spíše při rozdělení do skupin, aby žáci nebyli od robota příliš vzdáleni a mohli si s robotem navzájem zřetelně rozumět.

Dále robot nabízí velké množství gest a pohybů díky mnoha pohyblivým kloubním spojením s motory. To umožňuje provádět téměř totožné pohyby a gesta jako člověk, kterými doprovází svůj hlasový projev. Součástí pohybů robota je i chůze. Robot je schopen se pohybovat do libovolné polohy dle nastavení vzdálenosti [52]. V rámci mého programu bude chůze použita pro výběr dalšího žáka, který bude zodpovídat příklad.

Při pohybu jsem použila funkci rozpoznávání obličeje. Robot dokáže rozpoznat obličej, spočítat, kolik obličejů je v jeho zorném poli [53]. Tuto funkci využiji k vyhledání dalšího žáka, kterému budou pokládány příklady.

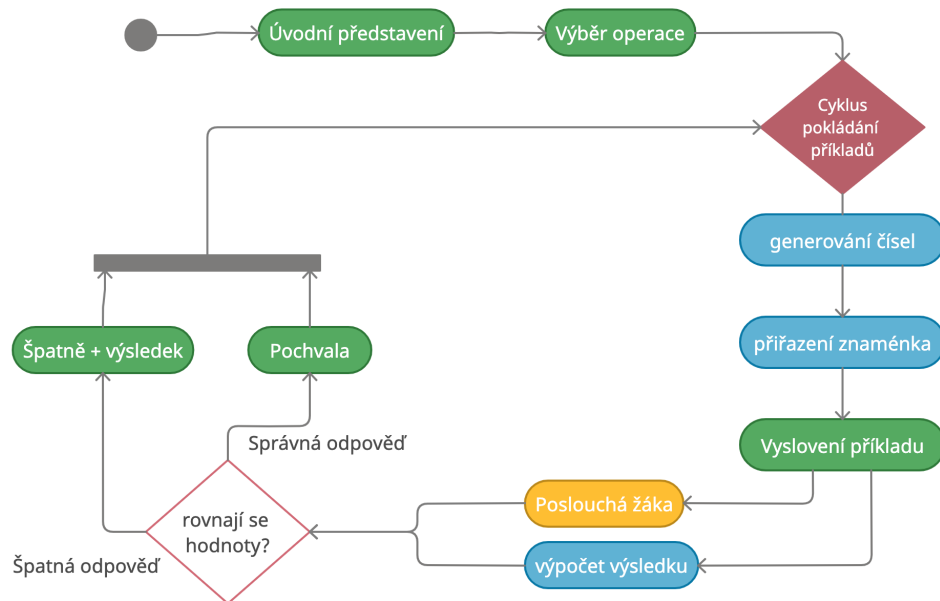
5.3 Diagram programu

5.3.1 Základní schéma

V prvotním návrhu programu je důležité stanovit základní body, kterými je úvodní představení robota, výběr matematické operace, kterou chceme procvičovat, vygenerování početního příkladu a jeho vyslovení směrem k dětem, poslouchání odpovědi žáka, vyhodnocení správnosti odpovědi a vyjádření odpovídající reakce podle toho, zda je odpověď správná nebo nesprávná.

Tyto stěžejní body programu je nutné nejen seřadit tak, aby časově odpovídaly a dávaly smysl jako celek, ale také je nutné je uskupit tak, aby robot pokládal příklady opakovaně. K tomu je vhodné použít cyklus, který bude probíhat nepřetržitě dle potřeby. Na začátku cyklu vygenerujeme hodnoty pro příklad, který robot následně vysloví, poslouchá odpovědi, posléze odpověď vyhodnotí a dle správnosti reaguje. Ostatní úkony budou umístěny na začátek programu před cyklus. Základní uspořádání je na Obrázku 16.

Robot provede výpočet příkladu sám v rámci skriptu, zároveň poslouchá a čeká na odpověď žáka. Pokud se obě tyto operace provedou úspěšně, výsledek výpočtu a odpověď žáka se porovnají. Pokud se tyto číselné hodnoty sobě rovnají, příklad je považován za správně zodpovězený a žák je robotem pochválen. V opačném případě je předpokládáno, že chybná hodnota je odpověď žáka. V tomto případě je žákovi sděleno, že jeho odpověď nebyla správná a dozvídá se i správný výsle-



Obrázek 16: Základní diagram programu – cyklus příkladů

dek tohoto příkladu. V této podobě je program schopen plnohodnotně fungovat a pokrývá většinu stěžejních částí s výjimkou ukončení programu a několika doplňujících funkcí.

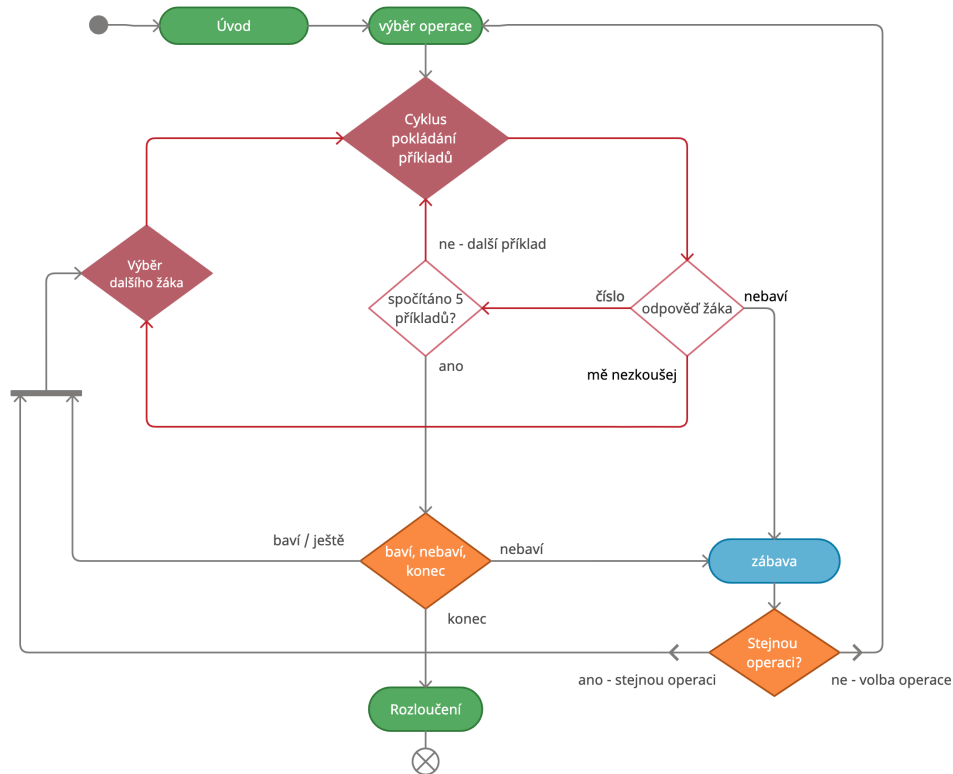
5.3.2 Úplné schéma

V dalším kroku je potřeba začlenit mimo jiné například funkci pro rozpoznávání tváří žáků, aby robot byl schopen po sérii pěti příkladů vyhledat jiného žáka. Robot se několikrát pomocí chůze otáčí v kruhu a detekuje tváře v okolí. Pokud rozpozná dostatečné množství tváří, jednu si vybere a té položí příklad. Mezi vyslovením příkladu a odpovědí žáka robot sám udržuje s žákem oční kontakt.

V posledním kroku jsem do programu přidala již zmíněné ukončení programu. Žák zodpoví 5 příkladů. Tuto hodnotu je možno libovolně změnit dle potřeby. Robot se zeptá, zda chce procvičovat dál a žák buď odpoví „ano/ještě“ a procvičování pokračuje, nebo odpoví „stačilo/ukončit“, robot se rozloučí a program tím skončí. Další možná odpověď je „nudím se“ nebo „už mě to nebaví“. V tomto případě program pokračuje na box se zábavou.

Box se zábavou obsahuje 6 zábavných funkcí, kdy robot tancuje, hraje na kytaru apod. Při tvorbě této zábavy jsem použila převážně již připravené pohyby

a funkce, které jsou obsaženy v programu Choregraphe, mírně jsem je upravila a doplnila o zvukový podklad.



Obrázek 17: Kompletní diagram programu

Také jsem přidala reakci na odpověď žáka „mě už to nebaví“ do místa, kde robot poslouchá a čeká na výsledek příkladu od žáka, aby bylo možné přerušit cyklus pokládání 5 příkladů, pro případ, že by se žáci začali nudit např. u 3. příkladu. V tomto případě se znovu náhodně vybere jedna ze zábavných prezentací.

Program také obsahuje reakci na odpověď „mě nezkoušej“, když robot čeká na zodpovězení příkladu, pro případ, že zvolenou tváří by byl vyučující. Při takovéto odpovědi nečeká na zodpovězení příkladu a vyhledá další obličej.

Zároveň se robot vždy po zábavě zeptá, zda žák chce procvičovat tutéž matematickou operaci nebo ji chce změnit. Podle toho buď robot pokládá další příklad, nebo se zeptá, jakou operaci chce žák procvičovat nyní, jako to bylo v úvodu programu, kdy robot vyjmenuje možnosti, které jsou na výběr a čeká na odpověď žáků.

6 Tvorba programu

Při tvorbě programu jsem postupovala od stěžejních částí jako je úvodní představení, výběr matematické operace, uložení operace i jiných dat do proměnné, vytvoření příkladu, vyslovení příkladu a vyhodnocení správnosti odpovědi. Poté jsem program postupně rozšiřovala stejně tak, jak je znázorněno v diagramech v předchozí kapitole.

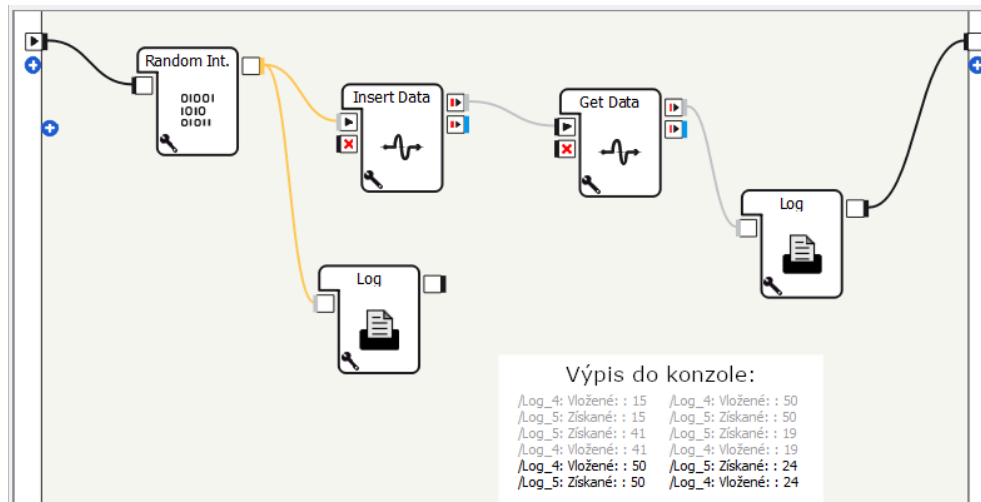
6.1 Tvorba dílčích částí

Program jsem vytvářela po dílčích částech. Tyto části jsem vytvářela každou zvlášť a testovala jejich možnosti a chování. Po vytvoření těchto několika dílů jsem vše spojila do celistvého programu.

6.1.1 ALMemory

Jednou ze zásadních funkcí je ukládání dat v průběhu programu. Data jsou ukládána do paměti robota a zůstávají zde uložena po dobu běhu programu. O ukládání dat se stará funkce ALMemory. Tuto funkci jsem využila zejména pro uchování identifikátoru operace, která je právě procvičována, k počítání zodpovězených příkladů, ale i v mnoha dalších situacích.

Pro účel testování jsem použila box, který vygeneruje náhodnou celočíselnou hodnotu. Tuto hodnotu funkce Log vypíše do okna konzole. Zároveň se tato hodnota zapíše do paměti podle klíče, který slouží jako identifikátor uložených dat. Pokud uložení dat proběhlo bez chyb, v dalším boxu je hodnota načtena znovu podle klíče na principu key – value. Pokud byla data nalezena a přečtena, hodnota se znovu vypíše do konzole a tím program končí (viz. Obrázek 18).



Obrázek 18: Testování ALMemory

6.1.2 Funkce podmínky

Funkce podmínky pracuje se dvěma číselnými nebo textovými hodnotami a jejich vztahem mezi nimi. První hodnotu získává ze vstupu do boxu. Druhou hodnotu je nutné nastavit jako statickou v parametrech boxu. Lze také nastavit operátor, který se má při porovnání použít. Pokud vstupní hodnota odpovídá předpokladu, program pokračuje horním výstupem z boxu. Pokud podmínka není splněna, je inicializován spodní výstup.

Toto nastavení jedné statické hodnoty je sice pro uživatele snadné, ale neumožňuje porovnávat dvě dynamicky se měnící hodnoty. Pro porovnávání dvou dynamických hodnot je nutno přidat další vstup do boxu a upravit vnitřní skript této funkce. Tímto se budu zabývat v dalších částech.

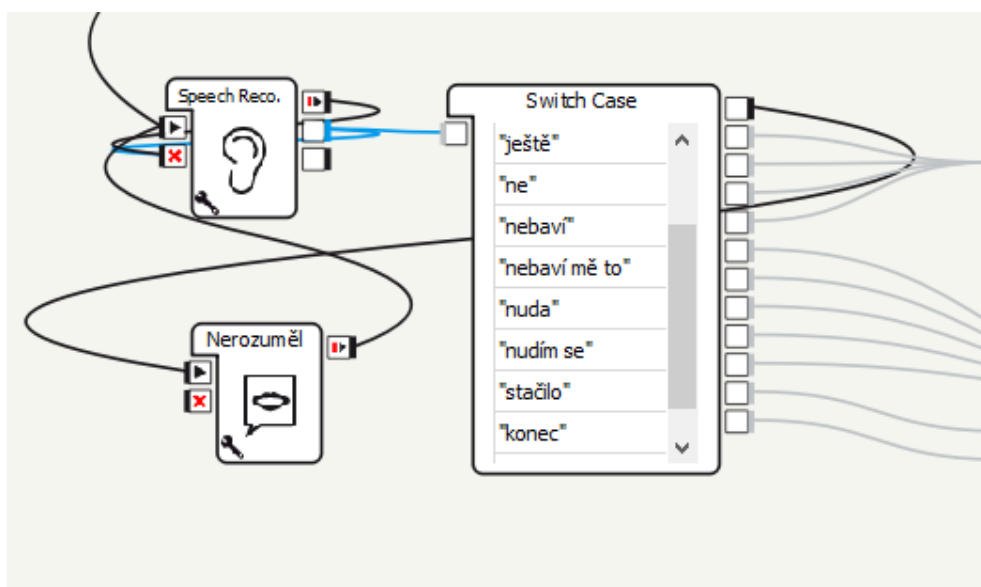
6.1.3 Převod řeči na text

Tato funkce je realizována pomocí dvou samostatných boxů. Prvním je Speech recognition, který obsahuje seznam slov a výrazů, které jsou očekávány. Při spuštění této funkce robot poslouchá zvuk v jeho okolí a dle zvoleného jazyka hledá shodu ve své slovní zásobě. Pokud je nějaké slovo rozpoznáno, zobrazí se jeho textový přepis v dialogovém okně spolu s procentuální shodou. Pokud je takové slovo na seznamu a shoda je dostatečná, je ve tvaru String předáno dále. Zároveň je nutné zmínit, že při rozpoznání slova robot nepřestane poslouchat, a proto je potřeba poslouchání zastavit pomocí modré spojnice, která směřuje

z výstupu rozpoznávaného slova do vstupu onStop označeného červeným křížkem (viz. Obrázek 19).

Nyní rozpoznané slovo pokračuje do druhého boxu, který je v této souvislosti často používán, tím je Switch Case. Tento box přijímá textový řetězec s rozpoznávaným výrazem od předchozího boxu a pro větší přehlednost je až zde program větven podle obsahu rozpoznávaného slova. Box Switch Case obsahuje libovolný počet řádků, do kterých je možné vepsat text nebo číslice. Obsah těchto řádků by měl být shodný s očekávanými slovy v předchozím boxu. Pokud přijatý text bude shodný s textem některého z řádků, aktivuje jen konkrétní výstup patřící k tomuto řádku a tím se program rozvětví. Pokud v tomto boxu nebude přijaté slovo nalezeno, aktivuje se horní výstup. Právě tento výstup je vhodné propojit s inicializačním vstupem v předchozím boxu Speech Reco. jak je znázorněno na obrázku 19 a díky této spojnici začne proces rozpoznávání znovu.

V mém projektu jsem tuto funkci použila pro volbu matematické operace, která bude procvičována, i k další hlasovým odpovědím jako například sdělení výsledku zadaného příkladu nebo veškerým rozhodováním o následných činnostech. Takové odpovědi byly například ne, nebaví, nebaví mě to, nuda, nudím se nebo ano, jo, chci, ještě, po zodpovězení pěti příkladů. V případě, že Nao zvolil pro další příklad tvář vyučujícího, byla odpověď „mě nezkoušej, já jsem učitel“. Pro ukončení je očekávána odpověď „stačilo“ nebo „ukončit“.



Obrázek 19: Ukázka rozpoznávání hlasu

Aby byla větší šance na shodu při rozpoznávání slov v hlučnějším kolektivu, volila jsem spíše kratší odpovědi, případně delší a kratší variantu (nebaví mě to, nebaví). Také je možné nastavit požadovanou procentuální shodu na nižší hodnotu tak, aby robot na jednu stranu nedetekoval jiná slova jako ta očekávaná, na druhou stranu aby rozuměl i mladším žákům se slabším hlasem a odlišnou výslovností. Toto nastavení je nutné vyzkoušet vzhledem k akustickým podmínkám prostoru i dalším okolnostem.

Dalším důležitým aspektem bylo jasně stanovit, jaké odpovědi jsou očekávány. Například na začátku programu robot vyjmenuje čtyři matematické operace, které lze procvičovat a tyto operaci jsou v očekávaných slovech. Nao také zmíní, že lze procvičovat variantu „všechno“, kdy mezi očekávanými slovy je „všechno“ i „vše“. Stejně je to i v případě věty „Pro ukončení stačí říct konec.“, kdy je jasně dáno, jaké slovo program očekává. Tyto veškeré úpravy jsem aplikovala nejen z důvodu, aby robot lépe rozuměl, ale především, aby robot měl vůbec šanci řeč rozpoznat.

6.1.4 Vyslovení textu

Choregraphe nabízí více způsobů, jak převést text na řeč. V mém projektu jsem použila Say, Say text a Dialog. První dva se od sebe příliš neliší. Je možné u nich nastavit výšku hlasu a rychlost projevu. Liší se ale v použití.

Funkce Say je schopna fungovat samostatně bez vstupních dat. K jejímu spuštění stačí pouze inicializace a poté přehraje text, který je staticky zadáný uvnitř boxu. Tato funkce je vhodná pro neměnný text, například úvodní představení, rozloučení na konci programu nebo uvedení následující činnosti. V programu jsem tento box použila mimo jiné pro výčet matematických operací, které je možné zvolit a pro dotázání žáka, zda ho to ještě baví a chce procvičovat dále.

Druhou možností, kterou nyní zmíním je Say text. Tato funkce neumožňuje vložit statický text, ale očekává text ze vstupu. Použití je vhodné pro vyslovení proměnlivého textu, se kterým se pracuje v jiném boxu. Vstupem může být text i číslo. Já jsem tuto metodu využila pro vyslovení příkladu, kdy jsem vytvořila box, který očekával dva vstupy v podobě textových řetězců. Mezi tyto hodnoty se vloží matematické znaménko a příklad vysloví.

6.1.5 Tvorba dialogu

Toto je další způsob, jakým robot vysloví text. Obsahuje box Dialog, který slouží k zahájení předem vytvořeného dialogu a vkládá se do požadovaného místa v diagramu. Obsahuje možnost nastavení jazyka a volby tématu.

Druhou nezbytnou částí pro fungování je téma. Tvoří ho samostatný soubor, který se v diagramu programu nezobrazuje, odkazuje na něj název tématu v boxu dialogu. Toto téma má vlastní pravidla i syntaxi a vzniká v QiChat. Umožňuje vytvořit dojem vědomého dialogu pomocí předem nastavených otázek a odpovědí (viz. kapitola 3.2.1 Dialog).

V tomto programu jsem pro takovýto dialog nenašla využití, ale tuto funkci jsem několikrát použila, aby Nao zvolil v opakujících se situacích různé reakce. Příkladem je moment, kdy žák zodpoví příklad a robot má reagovat dle správnosti odpovědi. Dialog umožňuje tyto reakce volit náhodně.

```
topic: ~Spravne ()
language: czc

concept:(pochvala) ^rand["Dobre" "Skvele" "Spravne"
                        "Vyborne" "Velmi dobre"]
concept:(pridavnejmeno) ^rand[Dobra Skvela Spravna]

concept:(druhaveta) ^rand["~pridavnejmeno odpoved."
                          "Odpovedel jsi spravne."
                          "Jsi sikovny poctar." "Jde ti to."]

u: (e:onStart) ~pochvala ~druhaveta $onStopped=1
```

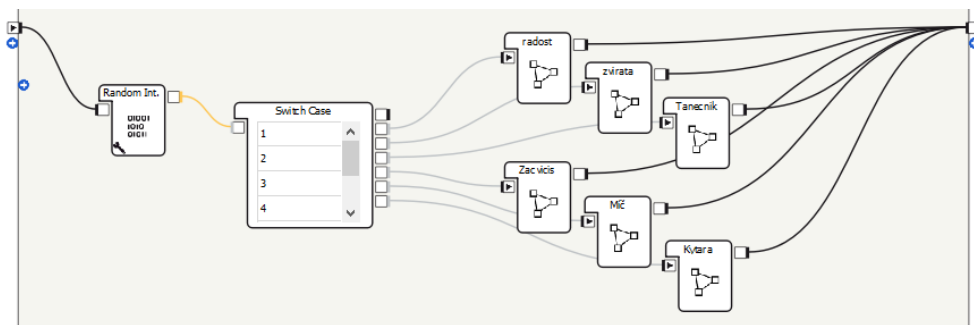
Příklad 3: Zdrojový kód dialogu

Robot: Správně! Odpověděl jsi správně.
 Robot: Správně! Správná odpověď.
 Robot: Skvělé! Jsi šikovný počítač.
 Robot: Skvělé! Dobrá odpověď.
 Robot: Dobře! Jde ti to.
 Robot: Dobře! Skvělé odpověď.
 Robot: Velmi dobře! Dobrá odpověď.

Obrázek 20: Výsledný dialog

6.1.6 Generování náhodné hodnoty

Choregraphe nabízí v základu generování dvou různých číselných hodnot, celočíselného Integeru a desetinného Float v rozsahu 0 – 1 000 a v případě Integeru lze zapnout možnost Shuffle, která ošetří případné opakování se vygenerovaných čísel. V mém programu jsem použila generování celočíselných hodnot pro tvorbu příkladu a také pro náhodný výběr zábavy. V boxu zábavy se náhodně generuje celočíselná hodnota v rozmezí od 1 do 6 a podle toho je zvolena jedna ze 6 zábavných aktivit (na Obrázku 21).



Obrázek 21: Náhodné generování čísel

6.1.7 Matematické operace

Další funkcí, kterou jsem použila byly matematické operace. Choregraphe nabízí již vytvořené násobení a dělení. Oba boxy očekávají dva číselné vstupy, vykonají svůj výpočet a výsledek vrátí do výstupu.

Násobení přijme vstup, převede číslo na Float a pokud již přijalo oba vstupy, vynásobí je a vrátí. Dělení pracuje totožně, akorát ještě kontroluje, zda se nedělí nulou. V takovém případě proces zastaví a vypíše upozornění do konzole. Tyto již vytvořené funkce mi posloužili jako inspirace při tvorbě skriptu pro sestavení a výpočet příkladu. Tyto skripty jsou popsány v následujících částech.

6.2 Výsledné funkce

V této kapitole jsou popsány celé funkce obsažené v programu včetně jejich zdrojových kódů. V těchto funkcích jsou mimo jiné použity funkcionality z předchozí podkapitoly.

6.2.1 Výpočet příkladu

Skript výpočtu má několik znaků shodných s boxy matematických operací v podkapitole 6.1.7 Matematické operace.

Hlavním znakem je nejen očekávání dvou vstupů v podobě číselných hodnot, ale především kontrola, zda oba vstupy přijaly data, a zda jsou obě hodnoty již převedeny do potřebného datového typu a připraveny na výpočet. Tato kontrola je realizována tak, že spolu s proměnnými, které uchovávají vstupní hodnoty, jsou vytvořeny ještě další dvě proměnné typu Boolean. Pokud vstup přijme číselnou hodnotu, uloží ji do int. proměnné a zároveň Boolean proměnnou nastaví na True. V případě, že mají hodnotu True proměnné od obou vstupů, program pokračuje dále k výpočtu. Současně se obě hodnoty nastaví na False, aby byly připraveny na další vstup při následujícím příkladu (viz. Příklad 4).

```
def onInput_Multiplicand(self, rVal):
    self.input1 = int (rVal)
    self.bool1 = True
    self.process()

def onInput_Multiplier(self, rVal):
    self.input2 = int (rVal)
    self.bool2 = True
    self.process()

def process (self):
    i1 = self.input1
    i2 = self.input2

    #pokud jsou zadany oba vstupy True, zavola operaci
    if self.bool1 and self.bool2:
        self.getOperation()
        self.bool1 = False
        self.bool2 = False
```

Příklad 4: Zdrojový kód kontroly vstupů

Identifikace operace Další důležitou částí je načtení matematické operace, která je nyní procvičována. Id operace je uloženo v paměti pomocí ALMemory. Operace jsou označeny jako 1 – sčítání, 2 – odčítání, 3 – násobení, 4 – dělení a 5 – náhodně všechny čtyři operace. Tento box pracuje v tuto chvíli se dvěma proměnnými v paměti. Jedna slouží pro uchovávání již zmíněného id operace. Pokud by tato hodnota byla 5, očekává se, že procvičovaná operace bude s každým

příkladem volena náhodně pomocí náhodného generování čísla v rozmezí 1–4 (1 ze 4 operací). Pro následující boxy je nutné tuto hodnotu znát, například pro vyslovení příkladu, proto se uloží do druhé proměnné.

Sčítání a odčítání Známe tedy již id operace, která má být v příkladu. Pomocí klasické podmínky je dle id vybrán odpovídající výpočet. Do tohoto výpočtu jsou dosazeny hodnoty ze vstupu a příklad je spočítán.

V případě sčítání nedochází k žádné nestandardní situaci, jak je vidět ve skriptu Příkladu 5. Pokud se podíváme na odčítání, je nutné ošetřit situaci, kdy menší je větší než menšenec a jako výsledek by vyšla záporná hodnota. Takovou hodnotu by jednak žák nemusel být schopen spočítat, ale záporná hodnota výsledku by mohla činit obtíže i v průběhu programu.

```
#scitani
if (self.operace == 1):
    rRes = self.input1 + self.input2

#odcitani
elif (self.operace == 2):

    if (self.input1 < self.input 2):
        temp = self.input1
        self.input1 = self.input
        self.input2 = temp

    rRes = self.input1 - self.input2
```

Příklad 5: Zdrojový kód sčítání a odčítání

Pro ošetření této situace jsem implementovala kontrolu v podobě porovnání hodnot. Pokud je menší větší než menšenec, jsou tyto dvě hodnoty navzájem vyměněny pomocí třetí proměnné. Až po této kontrole a případné výměně je možné provést výpočet příkladu.

Násobení a dělení Průběh násobení nevyžaduje žádnou zásadní úpravu. Stejně jako v případě sčítání se mezi první a druhou hodnotu dosadí znaménko a příklad se spočítá.

```

#nasobeni
elif (self.operace == 3):
    rRes = self.input1 * self.input2

#deleni
elif (self.operace == 4):
    nasob = self.input1 * self.input2
    rRes = nasob / self.input2
    self.input1 = nasob

```

Příklad 6: Zdrojový kód násobení a dělení

Rozdílný je průchod dělením. Aby bylo možné příklad spočítat z paměti, je vhodné zvolit dělení beze zbytku. V tomto případě nejde o dělení právě těch vygenerovaných čísel, protože šance, že by jejich podíl vyšel beze zbytku je mizivá. Zvolila jsem postup, kdy se tyto hodnoty navzájem vynásobí a výsledek je poté dělen jednou ze vstupních hodnot a druhá tvoří výsledek. Máme-li hodnoty 3 a 9, tak jejich součin je 27. V druhém kroku vezmeme 27 jako dělenec a libovolné z vygenerovaných čísel jak dělitel. Zbývající hodnota bude tvořit výsledek ($27 / 9 = 3$ nebo $27 / 3 = 9$).

Procvičování náhodné operace V případě, že id operace je 5, pokračuje program do části „procvičování všeho“. Jak už jsem zmínila, zde se vygeneruje náhodně volená hodnota v rozmezí 1–4 a dle této hodnoty je vybrán jeden z výpočtů. Nesmí být opomenuto zapsání nově vygenerovaného id operace do proměnné, aby robot v dalším boxu věděl, s jakým znamínkem příklad vyslovit.

```

#nahodne
elif (self.operace == 5):
    randInt = random.randint( 1, 4 )
    self.memory.insertData(self.getParameter("randKey"), randInt)

#scitani
if (self.operace == 1):
    rRes = self.input1 + self.input2

#odcitani
...

```

Příklad 7: Zdrojový kód procvičování náhodné operace

Závěr tvorby příkladu V závěru tohoto skriptu je výsledek příkladu uložen do ALMemory, zároveň je odeslán výstupem do dalšího boxu společně s hodnotami, které tomuto výsledku odpovídají s ohledem na zvolenou operaci.

6.2.2 Vyslovení příkladu

Tento box pracuje se dvěma vstupy a proměnnými ALMemory. Je zde stejná funkce jako v předchozím boxu, která ověřuje, zda byly získány obě vstupní hodnoty. Pokud byla získána alespoň jedna hodnota, je zavolána funkce pro získání id procvičované operace.

Získání id procvičované operace Toto id je získáno z paměti robota a pokud je jeho hodnota 5, znamená to, že jsou procvičované náhodné operace a je nutné načíst data i z druhé proměnné, která uchovává id operace pro aktuální příklad. Dále se pracuje již jen s touto hodnotou. Nyní se ověří, zda jsou dostupné obě vstupní hodnoty a je možné přejít k textové podobě id operace.

```
def getOp(self):
    self.memory = self.session().service("ALMemory")
    op = self.memory.getData(self.getParameter("key"))

    if (op == 5):
        rOp = self.memory.getData(self.getParameter("randKey"))
        op = rOp

    if self.bool1 and self.fbool2:
        self.operace = op
        self.getword()
    pass
```

Příklad 8: Získání id procvičované operace

Textová podoba operace V této funkci jsou definovány čtyři varianty dle id operace. Robot si tento text operace uloží do proměnné (např.: slovo = „děleno“). Následně volá funkci příklad, kde se do textového řetězce vloží první číselná hodnota ze vstupu, poté slovo pro operaci a nakonec druhá vstupní hodnota. Při vkládání do textového řetězce musí být všechny hodnoty převedeny na String, zejména číselné hodnoty ze vstupu.

Dále jsem v textovém řetězci přidala mezery, aby robot vyslovoval každé slovo zvlášť. Na konci řetězce najdeme otazník, který v robotově intonaci není příliš znát, ale z pohledu gramatiky je zde nutný.

V posledním kroku je před tento textový řetězec doplněno „Kolik je“ a příklad může být vysloven.

```
def getWord(self):
    if (self. == 1):
        self.opword = "plus"
    elif (self. == 2):
        self.opword = "minus"
    elif (self. == 3):
        self.opword = "krat"
    elif (self. == 4):
        self.opword = "deleno"
    self.priklad()
```

Příklad 9: Textová podoba operace

6.2.3 Poslouchání výsledku od žáka

Nyní robot vysloví příklad ve tvaru „Kolik je deset mínus šest?“ a čeká na odpověď od žáka. Poslech je realizován klasicky pomocí Speech recognition, kdy jsou definována očekávaná slova a pokud je hlasový projev shodný s některým ze slov, je toto slovo předáno dále. Při tomto použití jsou definovány číslíce v rozmezí 0 – 100. Pokud budou rozpoznány, předá je program dále jako výsledek od žáka, a jeho správnost bude vyhodnocena.

Další možné odpovědi, které program očekává, jsou pro případ, že by žáky procvičování omrzelo nebo pro případ, že robot zkouší vyučujícího. V prvním případě robot očekává výrazy: nebaví, už mě to nebaví, nudím a nudím se. Poté přejde do boxu se zábavou a příklad zanechá nezodpovězený. Stejně tak je tomu při odpovědi „mě nezkoušej“ a „já jsem učitel“. Robot nečeká na odpověď a přejde k vyhledávání tváří, kde najde nového žáka.

6.2.4 Kontrola výsledku

Porovnání výsledku od žáka a výsledku výpočtu programu je jednoduché. Tato funkce opět počká, až bude mít data z obou vstupů a poté je porovná. Předpokládá se, že výpočet příkladu v rámci skriptu je správný a pokud žák odpoví jinak, jeho odpověď je považována za chybnou.

Nejen pro účel testování, ale také pro ověření rozpoznání odpovědi, jsou v kódu umístěny dva příkazy pro výpis do konzole v případě, že program je spuštěn z počítače. První výpis obsahuje výsledek příkladu, který vypočítal program a slouží jako referenční pro ověření správnosti odpovědi žáka. Druhý výpis obsahuje číselnou podobu odpovědi žáka, aby i vyučující mohl vidět, zda robot odpověď žáka převedl správně, a zda se odpověď žáka a programu shoduje.

```
def compare(self):

    if (self.bool1 and self.bool2):
        self.logger.info("Vysledek: " + str(self.input1))
        self.logger.info("Odpoved zaka: " + str(self.input2))

        self.bool1 = False
        self.bool2 = False

    if (self.input1 == self.input2):
        self.output_then()
    elif (self.input1 != self.input2):
        self.output_else()
```

Příklad 10: Porovnání výsledků

Po porovnání odpovědí se dle správnosti aktivuje jeden ze dvou výstupů. Pro správnou odpověď je připraven box s pochvalou. V opačném případě je žákovi řečeno, že jeho odpověď nebyla správná a poté je přečten správný výsledek, který je uložen v ALMemory. Následně program pokračuje k dalšímu příkladu.

6.2.5 Zábava

Box se zábavou obsahuje generátor náhodných čísel v rozmezí od 1 do 6 a dle hodnoty je vybrána jedna z aktivit. Aktivity jsou tvořeny mnoha boxy, především pomocí gest, pohybů a aktivit, které Choregraphe již nabízí. Součástí je také přehrávání zvuku.

Hudba použitá v aktivitách Tanečník, Rozcvička a Kytara je získána z knihoven, v rámci předplatného je možné dostupné materiály použít pro komerční i nekomerční použití [55, 56, 57]. Zvuky přehrávané při předvádění zvířat jsou součástí těchto pohybů, které jsou obsaženy v Choregraphe.

Ohledně pohybů jsem se inspirovala a několik pohybů převzala z webu Univerzity Notre Dame [1]. Zde je obsažených 18 pohybových aktivit. Všechny jsou volně ke stažení ve formátu .crg a doprovázené videem umístěným na youtube. Z pohledu licence je bezplatně dovoleno jakékoliv užívání, úpravy, publikování a distribuce, ale pouze pod touto licencí.

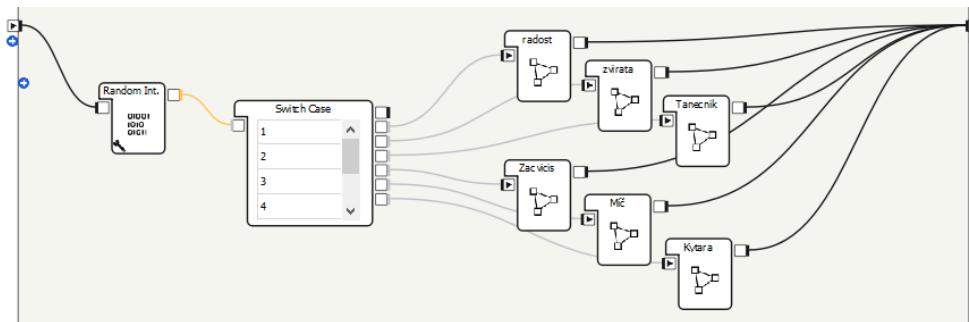
Přesto, že 18 aktivit je poměrně velký počet, většina z nich trvá méně než 10 vteřin. Obvykle se jedná jen o jednoduchá gesta nebo několik pohybů, které jsem spojila do smysluplného celku s určitým tématem. Jedná se o aktivity Tanečník a Rozcvička.

Ze zmíněného zdroje jsem testovala téměř všechny pohyby, ale většina z nich nebylo možné použít především z důvodu, že robot při některých z nich byl velmi nestabilní.

Dále se v mnoha pohybech vyskytl problém, kdy oproti vzorovému videu robot ve skutečnosti narážel například pažemi o sebe. Celkově byly výsledné pohyby výrazně odlišné, proto bylo nutné testovat pohyby výrazně zpomalené. V souvislosti s rychlostí mě dále překvapilo především u tanečních pohybů, že robot ve videu provádí pohyby poměrně rychle a přitom plynule. Při mém testování pohyby působily trhaným a uspěchaným dojmem. Proto jsem pohyby zpomalila a mnoho pozic úplně vynechala.

Pohyby pro aktivity Radost, Zvířata, Míč a Kytara jsou převážně z pohybů obsažených v Choregraphe včetně zvuků.

V prvním boxu robot ukazuje své emoce a poté rozpoznává náladu v obličeji žáka. Je schopen detekovat hodnoty positive, negative a neutral. Druhý box dává žákům hádat, jaká zvířata robot předvádí. Třetí a čtvrtý box přehrávají hudbu při které se robot pohybuje. Ve třetím boxu robot předvádí svůj tanec a ve čtvrtém boxu robot předvádí rozcvičku a poté vyzve žáky, aby se přidali, pustí hudbu a pohyby opakuje. Pátá zábava je kratší, zde Nao hodí žákům virtuální míč a pochválí je, že míč chytili. Poslední aktivitou je hra na kytaru. Robot přehrává



Obrázek 22: Box obsahující zábavné aktivity

zvuk kytary a předvádí, že hraje. Pohyby nejsou příliš autentické, ale dětskému publiku to snad postačí.

6.3 Shrnutí poznatků z praktické části

Zde jsem shrnula mé získané znalosti jak z tvorby programu, tak s nestandardních situací, které se při mé práci vyskytly.

6.3.1 Práce s robotem

V první chvíli mě robot zaujal svým vzhledem a pohybovými možnostmi. Především jeho pohyby hlavou, udržování očního kontaktu a gestikulace působí opravdu autentickým dojmem.

Stabilita a pohyb Ačkoliv na mnoha videích na internetu vypadá robot jako dokonalý stroj bez jediné chybičky, ve skutečnosti tomu tak není. Přesto, že robot má vnitřní detekci náklonu, občas se ocitá na hraně své rovnováhy a zřetelně vrávorá. Dle pozorování jsem dospěla k závěru, že záleží především na rovnosti a typu povrchu. Nejen, že robota dokáže výrazně rozhodit i běžný prach, který se usazuje na stole, ale prach nalepený především na chodidlech robota. Také záleží na typu povrchu zvoleného stolu, pravděpodobně dle materiálu a přilnavosti povrchu.

V souvislosti s pohybem jsou poměrně hlučné jeho motory, což kazí celkový dojem robota. Tento zvuk je nejen nepříjemný a způsobuje robotův projev méně srozumitelný, ale také není možné poslouchat projev člověka, když se robot pohy-

buje. V případě, že se robot během poslechu pohne v rámci samovolných pohybů, zachytí svůj zvuk a uživateli sdělí, že jeho odpovědi nerozuměl.

Rekognice řeči Během testování jsem také řešila, že robot řeč rozpozná ne-správně, s nízkou procentuální shodou nebo slovo neslyší vůbec. Záleží nejen na hluku, šumu a akustických vlastnostech prostoru, jak jsem předpokládala na počátku. Významnou roli hraje především výška hlasu, intonace, hlasitost, ale také jakýsi specifický způsob, jakým robota oslovovat. Mně se osvědčilo krátké a důrazné „okřiknutí“ s důrazem na první slabiku. Na druhou stranu, srozumitelnost nebyla ovlivněna vzdáleností mezi mnou a robotem. Robot rozuměl bez problémů i na vzdálenost cca 2 metrů. Větší problém mu činilo, pokud jsem se nenacházela v robotově zorném poli.

Ve zkratce se dá říct, že robot má občas problém rozumět řeči člověka v závislosti na výslovnosti, intonaci, akcentu, šumu v okolí, zvukovým podmínkám místnosti, a především v závislosti na zvoleném slovu. Nezvučná slova jako třeba *osm*, *sedm*, *šest*, ale i třeba „*ne*“ bez správného důrazu na první slabiku mohou být pro robota téměř nerozpoznatelné. Obecně se dá říct, že je potřeba si na robotovo vnímání „zvyknout“ a najít si svůj způsob, jakým na něho mluvit.

Další zvláštností je, že robot vyslovuje ve slově *matematika* slabiku „*ti*“ jako samostatnou, proto vyslovuje písmeno *t* jako *ť*. S velkým sebezapřením jsem tedy použila „*matematyka*“, což jistě zaujme všechny uživatele, kteří uvidí textový přepis, ale nenapadl mě jiný způsob, jak zajistit správnou výslovnost.

Také jsem zpozorovala, že se program robota občas zacyklí nebo zasekne, nepokračuje dále, jak by měl a robot na nic nereaguje, dokud není program zastaven a spuštěn znovu. Toto se stává náhodně v různých částech programu, bez ohledu na vstup od uživatele, dokonce i v různých a triviálních programech o několika málo boxech. Naštěstí toto zamrznutí není příliš časté. Jedná se o každé desáté až patnácté spuštění.

6.3.2 Práce s Choregraphe

S vývojovým prostředím Choregraphe se mi pracovalo velmi příjemně. Překvapila mě jednak velká vybavenost z pohledu existujících funkcí a v souvislosti s tím

i grafické znázornění funkcí formou boxu s ikonou a názvem. Tento software je volně ke stažení na mnoha webech, a to jak aktuální, tak starší verze pro předchozí modely robota Nao.

Instalace softwaru Ohledně aktuálnosti mi činil problémy spíše samotný software, který podporuje pouze Linux Ubuntu 16.04 a dle doporučení pouze jeho 64bitovou distribuci. Stejně tak software podporuje pouze macOS Sierra 10.12 vydaný v roce 2016 [54].

Dále je podporována 64bitová verze Windows 10, což je také jediný podporovaný a zároveň aktuální operační systém. Zde je ale nutné doinstalovat Microsoft Visual Studio C++ 2010 x64 Redistributable Package, což může některým uživatelům činit problémy [54]. Také jsem na Windows řešila problém při prvním spuštění, kdy se program spustil a vypsal upozornění, že firewall brání jeho přístupu do sítě. Samotné pop-up okno o povolení firewallu se zobrazilo sice až po cca 5 minutách, ale za to jedno totožné okno mnohokrát přes sebe. Toto chování jsem pozorovala na 3 různých počítačích.

Náročnost na výkon počítače V rozsáhlejších projektu se s rostoucím počtem boxů zvyšují i nároky na výkon počítače, a to nejen při spuštění samotného programu, ale zejména při správě samotných boxů během editace. Zároveň se při větším množství boxů stával projekt poměrně nepřehledný a i možnost zoomu fungovala dost nevyzpytatelně.

Programování skriptů v jazyce Python Při tvorbě byla velkou výhodou možnost zobrazit a upravovat skripty obsažené v boxech. Použitý jazyk Python považuji za dobrou volbu právě z důvodu, že se jedná o moderní jazyk, který uživatele ušetří složité syntaxe a dalších komplikací. Zároveň mohou být někteří žáci s jazykem Python již seznámeni. V současnosti je tento jazyk vyučován na školách pro základy programování. Jedná se sice o verzi Python 3, robot používá verzi 2.7 a s Pythonem 3 přišlo mnoho změn a vylepšení, ale domnívám se, že tyto rozdíly nebudou natolik zásadní, aby uživatelům působily rozsáhlejší problémy při tvorbě na rozdíl od přechodu z Pythonu na jiný a možná i objektově orientovaný jazyk se složitější syntaxí. Zároveň pokud uživatel není seznámen konkrétně s jazykem

Python, lze již vytvořené skripty snadno pochopit a pro další tvorbu jednodušších funkcí se jimi inspirovat současně s použitím základních znalostí obecného programování.

Virtuální robot Velkou pomocí je virtuální robot, který postačí pro základní potřeby testování. Neumožňuje však testovat dialog v hlasovém provedení. Očekávala jsem, že by s virtuálním robotem bylo možné vyhodnotit hlasový vstup od uživatele například pomocí mikrofону počítače. Bohužel je toto možné pouze v textové podobě v panelu dialogu formou chatu Robot – Human.

Dále ve virtuálním robotovi nelze testovat funkce související s tváří člověka, jako například rozpoznávání obličeje, věku, pohlaví, nálady a další, což v pozdější fázi činilo problém při testování mého programu.

Funkce s obličejem člověka Tyto funkce na získávání dat o lidském obličejí jsou na jednu stranu velmi propracované a umožňují poměrně přesné údaje. Zhoršenou přesnost jsem pozorovala nejen v závislosti na světelných podmínkách jako například při snímání tváře proti světlu, v šeru nebo pokud se pozorovaná osoba výrazněji pohybuje. Problém způsobují také dioptrické brýle, se kterými má robot mizivé množství úspěšných rozpoznání. Dále výrazné vousy způsobují, že robot hádá vyšší věk.

S těmito funkcemi je nutné zmínit, že rozpoznávání je časově náročné a trvá v rozmezí několika desítek vteřin, a to ať už se jedná o určování pohlaví, věku nebo ukládání tváře do paměti. Jedinou funkcí, kterou jsem mohla zařadit do programu tak, aby žáci při dlouhém čekání neztratili pozornost, je detekce nálady, která proběhne obvykle do 5 vteřin. Právě kvůli prodlevě robot oslovuje žáky i žákyně „jsi šikovný počtář“ a nerozlišuje jejich pohlaví.

Návrh na doplňující funkce Pokud bych měla navrhnout doplnění funkcí, které jsem v Choregraphe postrádala, ocenila bych mimo jiné funkci, která by zajistila, že robot zanechá obličej, který aktuálně sleduje a bude udržovat oční kontakt s jinou tváří.

Dále mě sice překvapilo množství funkcí a nastavení v prostředí QiChat pro tvorbu dialogu, ale postrádala jsem zde možnost dosazení nějaké proměnlivé hod-

noty, a to buď z „globální proměnné“ robotovy ALMemory nebo jako výstup některého z boxů. Absence této funkce mi činila problém především když žák zodpoví příklad nesprávně a robot mu má sdělit, že se mýlil a k tomu správný výsledek. Musela jsem tedy rozdělit zvlášť dynamickou odpověď (To je špatně, To není správně, Takto je to nesprávně, ...) a zvlášť statickou větu „Správný výsledek je“ a do kódu jsem doplnila načtení výsledku z proměnné.

Také by bylo vhodné vytvořit slovník často používaných slov v daném jazyce (ano, ne, ukončit, ukaž, sedni, vypnout), která by robot buď rozpoznával automaticky, bez nutnosti je vypisovat dvakrát – poprvé do očekávaných slov v boxu Speech Recognition a podruhé do Switch case. Druhou možností by bylo tyto dva seznamy spojit, aby bylo možné slova zadat pouze jednou. Třetím možným řešením je nabídnout uživateli slovník, ze kterého pouze označí slova, která robot dokáže rozpoznat a zároveň je uživatel očekává.

7 Ověření použitelnosti

Ověření použitelnosti mělo proběhnout ve skupině žáků od 3. do 5. třídy. Z důvodu protiepidemických opatření proti šíření nemoci COVID-19 toto ověření neproběhlo v době, kdy jsem tuto práci zpracovávala.

Nejen, že mnoho základních škol vidělo v mém pokusu epidemiologické riziko, ale také nechtěly komplikovat už tak narušenou výuku. Podařilo se mi domluvit spolupráci se zástupcem ředitele ZŠ J. Š. Baara, Mgr. Michalem Kadochem.

V zimních měsících roku 2020/21 probíhala distanční výuka všech žáků s výjimkou žáků 1. a 2. třídy ZŠ. Protože distanční výuka měla nejvýraznější dopad právě na tyto žáky spolu s faktem, že nebyla ještě ani polovina školního roku, nebyly znalosti těchto žáků dostatečné pro potřeby mého ověření. V médiích byl avizován návrat ostatních žáků 1. stupně do škol, ale naopak od 1.3. 2021 přešli na distanční výuku i žáci, kteří školu prezenčně navštěvovali. Proto jsem použitelnost programu pouze konzultovala s vedoucím mé práce.

Ve finální fázi tvorby mé práce byl schválen návrat do škol celého 1. stupně ZŠ. Pokusím se tedy provést ověření po odevzdání této práce, což by mělo být koncem dubna, případně v první polovině května 2021.

8 Závěr

Ve své bakalářské práci jsem se zabývala tvorbou programu k procvičování matematiky pro robota NAO se zaměřením na žáky 1. stupně základní školy. Abych mohla zmíněný program vytvořit, musela jsem si osvojit jazyk Python 2, vývojové prostředí Choregraphe a v neposlední řadě osnovy učiva matematiky 1. stupně základní školy, pro které je program určen. Součástí práce mělo být ověření mnou vytvořeného programu žáky 1. stupně základní školy, které se mi ovšem vlivem opatření proti šíření nemoci Covid-19 nepodařilo uskutečnit. Ověření jsem prováděla sama a z mého testování jsem vyvodila, že robot NAO je vhodný pro procvičování, dokáže žáky zaujmout svým vzhledem i vystupováním. Je všestranný a jeho použití je díky programovatelnosti široké a neomezuje se jen na školství.

Jako další možné použití mě napadá například porovnávání, které číslo je větší. Pro tento účel by bylo možné upravit tento program a výsledné procvičování by odpovídalo osnovám prvních tříd, pro které je stávající varianta příliš náročná. Dále Nao dokáže předvádět různá zvířata, ale není jich mnoho. V hodinách přírodopisu by žáky robot jistě zaujal, ale vytváření dalších zvířat by pro vyučujícího bylo časově náročnější.

Je ovšem potřeba na robota nahlížet jako na zařízení s mnoha nedostatky, které v práci zmiňuji a byly patrné již při tvorbě programu.

Se samotným robotem mě bavilo pracovat a pokud někdy budu mít tu možnost, ráda s ním budu znovu spolupracovat.

Seznam použité literatury a zdrojů

- [1] Special Movements. *The Nao Base* [online]. Notre Dame, Indiana: University of Notre Dame, 2014 [cit. 2021-04-15]. Dostupné z: <https://funlab.nd.edu/the-nao-base/special-movements/>
- [2] ACKERMAN, Evan. UBTECH Shows Off Massive Upgrades to Walker Humanoid Robot. *IEEE Spectrum* [online]. IEEE, 8. února 2019 [cit. 2021-04-15]. Dostupné z: <https://spectrum.ieee.org/automaton/robotics/humanoids/ubtech-upgrades-walker-humanoid-robot>
- [3] GEBHART, Andrew. Ubtech's soccer-playing Walker robot has a new pair of arms at CES 2019. *CNET* [online]. CNET, 7. ledna 2019 [cit. 2021-04-15]. Dostupné z: <https://www.cnet.com/home/smart-home/ubtechs-walker-now-has-arms-too-ces-2019>
- [4] UBTECH Robotics. *UBTECH Robotics, Inc.* [online]. UBTECH Robotics [cit. 2021-04-15]. Dostupné z: <https://www.ubtrobot.com/?ls=en>
- [5] Asimo - ROBOTS: Your Guide to the World of Robotics. *ROBOTS* [online]. IEEE [cit. 2021-04-15]. Dostupné z: <https://robots.ieee.org/robots/asimo>
- [6] SAMPLE, Ian. Honda unveils helmet that lets wearer control a robot by thought alone. *The Guardian* [online]. London, United Kingdom: The Guardian, 31. března 2009 [cit. 2021-04-15]. Dostupné z: <https://www.theguardian.com/science/2009/mar/31/mind-control-helmet-honda-asimo>
- [7] Pepper the humanoid and programmable robot. *SoftBank Robotics* [online]. SoftBank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: <https://www.softbankrobotics.com/emea/en/pepper>
- [8] Humanoidní robot Pepper. *Adastra.one a.s.* [online]. Adastra.one [cit. 2021-04-15]. Dostupné z: <https://adastra.one/products/pepper-robot>
- [9] MIKEŠOVÁ, Markéta. Nejznámější humanoidní robot Pepper se naučil novou funkci. Vynadá vám, když sundáte roušku. *Živě.cz* [online]. Živě.cz, 6.

- srpna 2020 [cit. 2021-04-15]. Dostupné z: <https://www.zive.cz/clanky/n-ejznamejsi-humanoidni-robot-pepper-se-naucil-novou-funkci-vynada-vam-kdyz-sundate-rousku/sc-3-a-205312/default.aspx>
- [10] BAUTER ENGEL, Jeff. SoftBank Taps Affectiva to Boost Pepper Robot's Emotional IQ. *Xconomy Boston* [online]. Xconomy, 28. srpna, 2018 [cit. 2021-04-15]. Dostupné z: <https://xconomy.com/boston/2018/08/28/softbank-taps-affectiva-to-boost-pepper-robots-emotional-iq>
- [11] ONLINE, Dqindia. India gets its first 3D printed humanoid robot at Rs. 1.5 lakh. *Dataquest* [online]. Gurgaon, India: Cyber Media (India), 8. ledna 2015 [cit. 2021-04-15]. Dostupné z: <https://www.dqindia.com/india-gets-first-3d-printed-humanoid-robot-rs-1-5-lakh>
- [12] HALTERMAN, Te. Soccer Playing, 3D Printed MANAV Robot Debuts at IIT Mumbai TechFest. *3dprint.com* [online]. 3DR HOLDINGS, 7. ledna 2015 [cit. 2021-04-15]. Dostupné z: <https://3dprint.com/35204/3d-printed-manav-robot>
- [13] ROBOTIS BIOLOID Series - Premium & GP. *ROBOTIS* [online]. Kalifornie, USA: ROBOTIS INC [cit. 2021-04-15]. Dostupné z: <https://www.robotis.us/bioloid-1/>
- [14] R+ Task 1.0. *ROBOTIS* [online]. ROBOTIS [cit. 2021-04-15]. Dostupné z: https://emanual.robotis.com/docs/en/software/rplus1/task/getting_started
- [15] ROBOTIS | PishRobot. *PishRobot* [online]. Tehran, Iran: PishRobot [cit. 2021-04-15]. Dostupné z: <https://www.pishrobot.com/eng/product-en/robotis>
- [16] ROBOTIS MINI Humanoid Robot. *ROBOTIS* [online]. Kalifornie, USA: ROBOTIS INC [cit. 2021-04-15]. Dostupné z: <https://www.robotis.us/robotis-mini>

- [17] Návodů na stavění - Zákaznický servis - LEGO.com CZ. *LEGO.com* [online]. Billund, Dánsko: The LEGO Group. [cit. 2021-04-15]. Dostupné z: <https://www.lego.com/cs-cz/service/buildinginstructions>
- [18] Informace | Mindstorms | Oficiálního LEGO® obchodu CZ. *LEGO.com* [online]. Billund, Dánsko: The LEGO Group. [cit. 2021-04-15]. Dostupné z: <https://www.lego.com/cs-cz/themes/mindstorms/ev3>
- [19] Technical overview — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/family/nao_technical/index_dev_naov6.html
- [20] Daily Use Nao — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: <https://www.softbankrobotics.com/emea/en/support/nao-6/2-daily-use>
- [21] Joint position sensors — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/family/nao_technical/mre_naov6.html
- [22] Motors — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/family/nao_technical/motors_naov6.html
- [23] CHERUBINI, Andrea. The Aldebaran NAO robot with its 21 degrees of freedom and reference frame. *ResearchGate* [online]. ResearchGate, červenec 2009 [cit. 2021-04-15]. Dostupné z: https://www.researchgate.net/figure/The-Aldebaran-NAO-robot-with-its-21-degrees-of-freedom-and-reference-frame-F-R-X-Y-Z_fig1_220143434
- [24] Loudspeakers — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/family/nao_technical/loudspeaker_naov6.html

- [25] Microphones — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/family/nao_technical/microphone_naov6.html
- [26] Video cameras — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/family/nao_technical/video_naov6.html
- [27] LEDs — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/family/nao_technical/leds_naov6.html
- [28] FSRs — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/family/nao_technical/fsr_naov6.html
- [29] Inertial unit — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/family/nao_technical/inertial_naov6.html
- [30] Sonars — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/family/nao_technical/sonar_naov6.html
- [31] Contact and tactile sensors — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/family/nao_technical/contact-sensors_naov6.html
- [32] Nao Humanoid robot. *PngEgg.com* [online]. Png Egg [cit. 2021-04-15]. Dostupné z: <https://www.pngegg.com/cs/png-dqrdf>
- [33] QiChat - Syntax — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné

z: http://doc.aldebaran.com/2-8/naoqi/interaction/dialog/dialog-g-syntax_full.html#dialog-rules

- [34] ALFaceCharacteristics — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: <http://doc.aldebaran.com/2-8/naoqi/peopleperception/alfacecharacteristics.html>
- [35] Face Characteristics - Get Expression — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-4/software/choregraphe/tutos/get_expression.html
- [36] ALMood Tutorials — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: <http://doc.aldebaran.com/2-5/naoqi/core/almoood-tutorials.html>
- [37] Face Characteristics - Get Age — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/software/choregraphe/tutos/get_age.html
- [38] ALMemory — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: <http://doc.aldebaran.com/2-8/naoqi/core/almemory.html>
- [39] What is Choregraphe — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/software/choregraphe/choregraphe_overview.html
- [40] NAO Tips Vol. 3: How to package and install applications with trigger sentences. *Brainary Interactive* [online]. Brainary Interactive, 14. srpen 2017 [cit. 2021-04-15]. Dostupné z: <https://www.brainaryinteractive.com/news/?offset=1505882085825>

- [41] Box — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: <http://doc.aldebaran.com/2-8/software/choregraphe/objects/box.html>
- [42] Scripting Python boxes — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/software/choregraphe/objects/python_script.html#python-script
- [43] Flow diagram panel — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/software/choregraphe/panels/flow_diagram_panel.html#choregraphe-reference-box-flow-diagram
- [44] Timeline — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/software/choregraphe/panels/timeline_panel.html#choregraphe-reference-box-timeline
- [45] Dialog box — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/software/choregraphe/objects/dialog_box.html#choregraphe-reference-box-type-dialog
- [46] Box - Optional components — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/software/choregraphe/objects/box_optional_components.html#choregraphe-reference-box-parameter
- [47] Box - Inputs/outputs — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-8/software/choregraphe/objects/box_input_output.html

- [48] Working with a simulated robot — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: <http://doc.aldebaran.com/2-8/dev/tools/robot-simulation.html>
- [49] Matematika pro 3. třídu. *Školík.cz* [online]. Školík.cz [cit. 2021-04-15]. Dostupné z: <http://skolik.cz/osnovy/matematika-pro-3-tridu/>
- [50] Matematika pro 2. třídu. *Školík.cz* [online]. Školík.cz [cit. 2021-04-15]. Dostupné z: <http://skolik.cz/osnovy/matematika-pro-2-tridu/>
- [51] Matematika pro 1. třídu. *Školík.cz* [online]. Školík.cz [cit. 2021-04-15]. Dostupné z: <http://skolik.cz/osnovy/matematika-pro-1-tridu/>
- [52] Your first steps in Choregraphe — Aldebaran 2.4.3.28-r2 documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: http://doc.aldebaran.com/2-4/software/choregraphe/choregraphe_first_steps.html
- [53] ALFaceDetection. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: <http://doc.aldebaran.com/2-4/naoqi/peopleperception/alfacedetection.html>
- [54] Choregraphe suite - Installation Guide — Aldebaran 2.8.6.23e documentation. *Softbank Robotics Documentation* [online]. Softbank Robotics, 2020 [cit. 2021-04-15]. Dostupné z: <http://doc.aldebaran.com/2-8/software/choregraphe/installing.html>
- [55] Happy by OrangeHead. *Audiojungle.net* [online]. Envato Pty. [cit. 2021-04-15]. Dostupné z: <https://audiojungle.net/item/happy/18457940>
- [56] Pool Party | Royalty Free Music | MorningLightMusic. *morninglightmusic.io* [online]. Morninglightmusic. [cit. 2021-04-15]. Dostupné z: <https://morninglightmusic.io/music/pool-party/>
- [57] Sad Guitar Loop by Music - ABC. *Audiojungle.net* [online]. Envato Pty. [cit. 2021-04-15]. Dostupné z: <https://audiojungle.net/item/sad-guitar-loop/16607280>

Seznam obrázků

1	Robot Walker [4]	14
2	Robot Asimo [5]	14
3	Robot Pepper [10]	15
4	Robot Manav [12]	16
5	Software Roboplus [14]	17
6	Kód v programu Roboplus [14]	17
7	Přestavby robota Bioloid [15]	18
8	Robotis Mini [16]	18
9	Lego Mindstorms [18]	19
10	Lego Mindstorms software [18]	20
11	Osy pohybu robota Nao [23]	22
12	Části robota Nao [32]	24
13	Software Choregraphe [40]	29
14	Grafické znázornění funkcí	30
15	Vstupy a výstupy boxů	32
16	Základní diagram programu – cyklus příkladů	38
17	Kompletní diagram programu	39
18	Testování ALMemory	41
19	Ukázka rozpoznávání hlasu	42
20	Výsledný dialog	44
21	Náhodné generování čísel	45
22	Box obsahující zábavné aktivity	53

Seznam příkazů

1	Zdrojový kód dialogu	26
2	Výsledný dialog	26
3	Zdrojový kód dialogu	44
4	Zdrojový kód kontroly vstupů	46
5	Zdrojový kód sčítání a odčítání	47
6	Zdrojový kód násobení a dělení	48
7	Zdrojový kód procvičování náhodné operace	48
8	Získání id procvičované operace	49
9	Textová podoba operace	50
10	Porovnání výsledků	51

A Přílohy

1. CD obsahující elektronickou podobu práce, zdrojový kód programu a video popisující fungování programu.