



ČVUT

ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE

F3

**Fakulta elektrotechnická
Katedra řídicí techniky**

Bakalářská práce

Návrh a realizace části robota Eliška

Marek Feik
Kybernetika a robotika

Květen 2025

Vedoucí práce: Ing. Martin Hlinovský, Ph.D.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Feik** Jméno: **Marek** Osobní číslo: **508488**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra řídicí techniky**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Návrh a realizace části robota Eliška

Název bakalářské práce anglicky:

Design the part of the robot named Eliška

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Martin Hlinovský, Ph.D. katedra řídicí techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **29.01.2025**

Termín odevzdání bakalářské práce: **23.05.2025**

Platnost zadání bakalářské práce: **20.09.2026**

doc. Ing. Zdeněk Hurák, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis proděkana(ky) z pověření děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Feik** Jméno: **Marek** Osobní číslo: **508488**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra řídicí techniky**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Návrh a realizace části robota Eliška

Název bakalářské práce anglicky:

Design the part of the robot named Eliška

Pokyny pro vypracování:

1. Seznamte se s možnostmi stavebnice Lego Mindstorms Education EV3 a se stavebnicí Brian (současný stav, HW a SW vybavení) a s robotem Ludvíkem.
2. Návrhněte a realizujte hlavu robota „Eliška“ s možností rozpoznávání pohybů eventuálně i obličejů nebo s možností rozpoznávání řeči a následnou reakcí (např. pomocí ASR Whisper, Chat GPT, ElevenLabs, OpenCV nebo jiné)
4. Vytvořte dokumentaci a webové stránky k realizovanému projektu.

Seznam doporučené literatury:

- [1] Webová stránka projektu robot "Ludvik" - <https://robosoutez.fel.cvut.cz/robot-ludvik>
- [2] Webová stránka rozpoznávání řeči pomocí ASR Whisper a Chat GPT od společnosti OpenAI - <https://github.com/openai/whisper> a <https://openai.com/index/chatgpt/>
- [3] Webová stránka ElevenLabs pro generování zvuku z textu - <https://elevenlabs.io/>
- [4] Webová stránka OpenCV knihovny pro rozpoznávání objektů v obrazech a fotoaparátech - <https://blog.desdelinux.net/cs/>
- [5] Youtube "World's Best AI LEGO Robot!" - <https://www.youtube.com/watch?v=MghrMAWsf8>
- [6] Webová stránka projektu "CreativeMindstorms / AI-LEGO-HEAD" <https://github.com/CreativeMindstorms/AI-LEGO-HEAD>



PROHLÁŠENÍ

Já, níže podepsaný

Příjmení, jméno studenta: Feik Marek
Osobní číslo: 508488
Název programu: Kybernetika a robotika

prohlašuji, že jsem bakalářskou práci s názvem

Návrh a realizace části robota Eliška

vypracoval samostatně a uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací a Rámcovými pravidly používání umělé inteligence na ČVUT pro studijní a pedagogické účely v Bc a NM studiu.

Prohlašuji, že jsem v průběhu příprav a psaní závěrečné práce použil nástroje umělé inteligence. Vygenerovaný obsah jsem ověřil. Stvrzuji, že jsem si vědom, že za obsah závěrečné práce plně zodpovídám.

V Praze dne 21.05.2025

Marek Feik

.....
podpis studenta

Poděkování / Prohlášení

Rád bych poděkoval svému vedoucímu Ing. Martinu Hlinovskému Ph.D, za odborné vedení a cenné rady při tvorbě této práce.

Dále děkuji své rodině a blízkým za podporu a trpělivost během celého studia.

Mé poděkování patří také Fakultě elektrotechnické ČVUT, za poskytnutí technického zázemí při vývoji praktické části projektu.

V poslední řadě děkuji všem, kteří mi jakkoliv pomohli, inspirovali mě nebo mě v průběhu studia motivovali.

Prohlašuji, že jsem tuto bakalářskou práci na téma Návrh a realizace části robota Eliška vypracoval samostatně pod vedením Ing. Martina Hlinovského Ph.D. Veškeré zdroje, z nichž jsem čerpal, jsou řádně citovány a uvedeny v seznamu použité literatury.

Prohlašuji, že tato práce nebyla využita k získání jiného akademického titulu a že splňuje veškeré náležitosti stanovené pravidly vysoké školy.

V Praze, dne 23. 5. 2025

.....

Abstrakt / Abstract

Tato bakalářská práce se zabývá návrhem a realizací robotické hlavy Elišky, která je postavena z LEGO Mindstorms EV3. Hlava Elišky slouží pro propagaci fakulty. Eliška je schopná komunikovat s uživateli díky umělé inteligenci, což zahrnuje rozpoznávání a generování řeči. S pomocí počítačového vidění také dokáže sledovat okolní prostředí a reagovat na výzvy uživatele.

Hlava Elišky využívá pro pohyby dvě kostky EV3. Tyto pohyby zahrnují otáčení hlavy, pohyby očí, obočí, otevírání pusy a pohyby koutků. Komunikace mezi řídicím softwarem, napsaným v jazyce Python, a kostkami EV3 je realizována přes Bluetooth. Rozpoznávání hlasu je zajištěno systémy Whisper od OpenAI a Porcupine od Picovoice. Pro zpracování textu i obrazu je použit chatbot od společnosti Google – Gemini AI. Generování Eliščina hlasu obstarává ElevenLabs. Eliščíno vidění je realizováno knihovnou OpenCV, která umožňuje sledování obličejů, rukou.

Cílem práce bylo vytvořit interaktivní robotickou hlavu z kostek LEGO, která slouží pro účely reprezentace fakulty. Výsledkem je systém schopný verbální i neverbální komunikace, který slouží pro demonstraci možností umělé inteligence a robotiky.

Klíčová slova: Robotická hlava, LEGO Mindstorms, EV3, Chatbot, Umělá inteligence, Rozpoznávání řeči, Generování hlasu, Počítačové vidění, Python, Bluetooth komunikace, Gemini AI, OpenCV, Whisper, ElevenLabs

This bachelor's thesis focuses on the design and implementation of the robotic head Eliška, built using LEGO Mindstorms EV3. Eliška serves as a propagative project for the faculty. It is capable of communicating with users through artificial intelligence, including speech recognition and generation. With the help of computer vision, it can also observe its surroundings and respond to user prompts.

Eliška's movements are controlled by two EV3 bricks, enabling head rotation, eye movement and eyebrow movement, mouth opening, and mouth corner movements. Communication between the control software, written in Python, and the EV3 bricks is carried out via Bluetooth. Speech recognition is handled by Whisper from OpenAI and Porcupine from Picovoice. For processing both text and images, a chatbot powered by Google's Gemini AI is used. Eliška's voice is generated using ElevenLabs. Her vision is implemented using the OpenCV library, allowing facial and hand tracking.

The goal of this project was to create an interactive robotic head using LEGO bricks for faculty representation purposes. The result is a system capable of both verbal and non-verbal communication, demonstrating the possibilities of artificial intelligence and robotics.

Keywords: Robot head, LEGO Mindstorms, EV3, Chatbot, Artificial intelligence, Speech recognition, Speech generation, Computer vision, Python, Bluetooth communication, Gemini AI, OpenCV, Whisper, ElevenLabs

Obsah /

1 Úvod	1		
1.1 Představení tématu	1		
1.2 Cíle práce	1		
2 Teoretická část	2		
2.1 Robotické hlavy a jejich využití .	2		
2.2 Inspirace LEGO hlavou Dave . .	2		
2.2.1 Představení projektu LEGO Dave od Creative Mindstorms	2		
2.2.2 Převzaté koncepty a jejich úprava pro Elišku	3		
2.2.3 Rozdíly mezi hlavou Dave a Eliškou	3		
2.3 LEGO MINDSTORMS ja- ko platforma pro robotiku	3		
2.3.1 Možnosti a omezení LEGO MINDSTORMS	4		
2.3.2 Další projekty využíva- jící EV3 pro robotiku	5		
2.4 Umělá inteligence v chat- bot technologiích	6		
2.4.1 Jak fungují moderní chatboti	6		
2.4.2 Přehled AI chatbotů a jejich využití	7		
2.4.3 Gemini AI – principy fungování, zpracování textu a obrazu	8		
2.5 Počítačové vidění a rozpo- znávání objektů	9		
2.5.1 Základní principy po- čítačového vidění	9		
2.5.2 OpenCV ke zpracování obrazu	10		
2.5.3 Rozpoznávání obličejů a sledování pohybu	11		
2.6 Rozpoznávání řeči a gene- rování hlasu	11		
2.6.1 Whisper od OpenAI – princip fungování	11		
2.6.2 ElevenLabs – genero- vání hlasového výstupu z textu	12		
2.6.3 Synchronizace řeči s pohybem pusy	12		
2.7 The uncanny valley efekt	13		
3 Návrh robotické hlavy Eliška	14		
3.1 Požadavky na systém	14		
3.1.1 Jaké pohyby a funkce má robotická hlava mít	14		
3.1.2 Jaký má být způsob interakce s uživateli	14		
3.1.3 Přehled technologií nutných k realizaci	15		
3.2 Mechanická konstrukce	15		
3.2.1 Návrh mechanismů pro pohyb	15		
3.2.2 Horizontální a vertikál- ní pohyb očí	16		
3.2.3 Pohyb obočí	16		
3.2.4 Otevírání pusy	17		
3.2.5 Hýbání koutků	18		
3.2.6 Otáčení hlavy	18		
3.3 Elektronické vybavení	19		
3.3.1 Dvě EV3 kostky	19		
3.3.2 Připojené motory a je- jich funkce	20		
3.3.3 Telefon jako senzor – propojení mikrofону, kamery a reproduktorů	21		
3.4 Softwarová architektura	21		
3.4.1 Struktura kódu	21		
3.4.2 Hlavní řídicí smyčka a průběh interakce	22		
4 Realizace robotické hlavy	24		
4.1 Stavba fyzického modelu	24		
4.1.1 Konstrukce jednotli- vých pohybových me- chanismů	24		
4.1.2 Oči a obočí	25		
4.1.3 Pusa a koutky	25		
4.1.4 Podstava a detaily	27		
4.2 Ovládání motorů a realiza- ce pohybů	28		
4.2.1 Programování EV3 pro řízení pohybů	28		
4.2.2 Převod příkazů z Py- thonu do motorů EV3	28		
4.3 Implementace kódu v Pythonu	29		

4.3.1	Struktura celého systému a hlavní program . . .	29
4.3.2	Propojení EV3 s Pythonem přes Bluetooth . . .	30
4.3.3	Detekování a rozpoznání hlasu	31
4.3.4	Gemini AI jako chatbot a zpracování obrazu . . .	34
4.3.5	Detekce obličejů a rukou, sledování pohybů . . .	35
4.3.6	Využití ElevenLabs pro generování hlasu	36
4.3.7	Další funkce Elišky	37
4.3.8	Vlákna pro vizualizaci a řízení pohybu očí	38
4.4	Jak Eliška vypadá	39
5	Testování a experimenty	40
5.1	Funkčnost pohybových mechanismů	40
5.2	Odezva chatbotu a hlasového výstupu	40
5.3	Počítačové vidění	40
5.4	Uživatelské testování	41
6	Závěr	42
	Literatura	43

Tabulky / Obrázky

2.1 Přehled vlastností LEGO motorů	4
2.2 Parametry kostky LEGO EV3...	6
2.3 Přehled moderních chatbotů.....	8
2.4 Přehled Whisper modelů	11
3.1 Přehled potřebných technologií k realizaci	15
3.2 Přehled potřebných technologií k realizaci	21
4.1 Doplnkové funkce Elišky	38
2.1 Robotická LEGO hlava Dave od Creative Mindstorms	2
2.2 LEGO MINDSTORMS EV3 kostka	3
2.3 LEGO MINDSTORMS motor	4
2.4 LEGO Mindstorms EV3 Home Edition.....	5
2.5 LEGO solver Rubikovy kostky ..	6
2.6 Loga moderních chatbotů.....	7
2.7 Přehled rodiny modelu Gemini ..	9
2.8 Funkce počítačového vidění....	10
2.9 Popis fungování Whisperu	12
2.10 Efekt Uncanny valley	13
3.1 mechanismus pohybu očí	16
3.2 mechanismus pohybu obočí....	17
3.3 mechanismus pohybu dolní čelisti	17
3.4 Návrh mechanismu koutků	18
3.5 mechanismus krku.....	19
3.6 Kostka Brian	20
3.7 Zjednodušený diagram programu	22
4.1 Původní realizace skořápky hlavy	24
4.2 Postavený modul očí a obočí ..	25
4.3 Moduly pusy a koutků	26
4.4 Držák telefonu v podstavě	27
4.5 Závěrečné úpravy a vlasy	28
4.6 Okno vidění Elišky	36
4.7 Úsměv Elišky	39
4.8 Jak robot Eliška vypadá	39
5.1 Konverzace s Eliškou.....	41

Kapitola 1

Úvod

1.1 Představení tématu

V posledních letech, zejména od roku 2018, dochází k rychlému rozvoji umělé inteligence, robotiky a interaktivních technologií. Tyto obory nacházejí uplatnění nejen v průmyslu, ale čím dál více i ve vzdělávání, výzkumu a hlavně veřejné prezentaci. Robotické systémy a umělá inteligence schopná komunikace a vnímání okolí se stávají atraktivním nástrojem, jak přiblížit široké veřejnosti složité technologické koncepty srozumitelnou a zábavnou formou.

Robotická hlava Eliška vznikla jako interaktivní projekt pro reprezentaci fakulty, který má potenciál upoutat pozornost veřejnosti, zejména uchazečů o studium, a ukázat jim praktické využití poznatků z oboru Kybernetika a robotika. Kombinuje mechanickou konstrukci z LEGO kostek, řízení pomocí mikropočítačů, programování v jazyce Python a integraci moderních AI nástrojů, jako je rozpoznávání hlasu, generování řeči nebo počítačové vidění.

Práce tak spojuje více technických disciplín – od mechatroniky a řízení motorů přes softwarové inženýrství až po práci s cloudovými službami – a zároveň se zaměřuje i na uživatelskou přívětivost, interaktivitu a estetické ztvárnění robota.

1.2 Cíle práce

Cílem této bakalářské práce je vytvoření robotické hlavy Elišky z kostek LEGO. Hlava Elišky slouží k propagaci fakulty, aby poutavou a interaktivní formou ukázala možnosti robotiky. Zároveň ukazuje uchazečům a potenciálním budoucím studentům, co může být výsledkem studia oboru Kybernetika a robotika.

Interaktivními prvky se Eliška snaží přiblížit skutečnému člověku. Mezi ně patří schopnost konverzace. Eliška dokáže vést živou a přátelskou konverzaci s uživateli. Eliška slyší, co uživatel říká, a na základě toho odpoví. Pro účely propagace fakulty dokáže Eliška informovat uživatele o podrobnostech studia nebo jiných informacích o fakultě. Mimoto dokáže být Eliška nápomocná i jinými způsoby, například ví, kolik je hodin nebo jaké je počasí. Aby byla Eliščina schopnost konverzovat na vyšší úrovni, je konverzace obohacena o informace z okolního prostředí. Eliška dokáže popisovat objekty ve svém okolí. Mezi ty se řadí i samotný uživatel, kterého dokáže Eliška popsat. Dalším interaktivním prvkem Elišky je její neverbální komunikace. Ta je podpořena sledováním polohy obličeje nebo rukou uživatele očima, pokud to situace vyžaduje. Další schopnosti podporující neverbální komunikaci jsou vyjadřování emocí díky mimice obličeje a otevírání pusy, společně s otáčením hlavy.

Kapitola 2

Teoretická část

2.1 Robotické hlavy a jejich využití

Robotické hlavy představují důležitou součást humanoidních robotů, jejichž cílem je napodobit vzhled, pohyb a chování člověka. I když se na první pohled může zdát, že jde jen o estetický prvek, ve skutečnosti hraje hlava zásadní roli při interakci robota s lidmi. Umožňuje zprostředkovat emoce, vyjadřovat záměry a navozovat pocit přirozené komunikace. Robotické hlavy nacházejí uplatnění zejména v oblasti výuky, asistivních technologií, zábavy nebo výzkumu interakce člověka a stroje. Jejich schopnost neverbální komunikace výrazně zvyšuje důvěryhodnost a přijetí robota lidmi.

2.2 Inspirace LEGO hlavou Dave

Celá bakalářská práce vznikla inspirací robotickou LEGO hlavou jménem Dave [1], kterou vytvořil holandský youtuber Sten vystupující pod přezdívkou Creative Mindstorms. Je známý svými působivými LEGO projekty, mezi které kromě robotické hlavy Dave patří například LEGO automat na sladkosti, kreslicí robot nebo LEGO 3D tiskárna.

2.2.1 Představení projektu LEGO Dave od Creative Mindstorms

Robotická hlava Dave od Creative Mindstorms je známým open-source projektem. Dave je postaven z LEGO kostek Technic a ovládán přes kostky LEGO MINDSTORMS EV3. Dave vyniká, oproti jiným humanoidním robotům z kostek LEGO, svou mimikou a interaktivní komunikací s uživatelem pomocí umělé inteligence[1].



Obrázek 2.1. Obrázek robotické LEGO hlavy Dave od Creative Mindstorms[1].

2.2.2 Převzaté koncepty a jejich úprava pro Elišku

Jelikož Eliška vychází z projektu LEGO Dave, mají toho mnoho společného. Celý projekt robotické hlavy Elišky je obsáhlý a zahrnuje velké množství samostatných celků, které všechny musejí fungovat, aby fungovala i Eliška. Bylo tedy žádoucí načerpat co nejvíce inspirace a mít vyjasněná některá důležitá rozhodnutí a principy mechanismů, v čemž LEGO Dave přišel velmi vhod.

Mezi převzaté klíčové koncepty patří softwarová část celého projektu. Creative Minds využívá osvědčené metody pro propojení dvou kostek LEGO EV3 a programovacího jazyka Python. Další důležitou částí je využití telefonu jako senzorů. V neposlední řadě je podstatné zmínit, že pokud má člověk vizuální představu o tom, co se snaží postavit a jaké mechanismy pro to můžou být dobré, celý projekt se posouvá kupředu mnohem rychleji. Poslední věcí, kterou je dobré zmínit, která Elišce hodně pomohla, je použití paruky. Je to veselý detail, který člověka na první dobrou nenapadne.

2.2.3 Rozdíly mezi hlavou Dave a Eliškou

Ačkoliv mají hodně společného, je zde i řada věcí, ve kterých se tyto dvě hlavy liší. Nejvíce rozdílů je v softwarové části. Eliška využívá značné množství jiných knihoven než Dave. Je to zejména kvůli jazykové bariéře. Například rozpoznávání českého jazyka a generování mluveného slova v češtině kvůli úplně chybějící podpoře češtiny u některých služeb. Důvodem je ale i zastaralost a kompatibilita některých knihoven, nebo dokonce jejich nahrazení lepšími. Dave a Eliška se liší i v konstrukčních prvcích. Eliška má větší podstavu kvůli většímu ložisku držící krk s hlavou.

2.3 LEGO MINDSTORMS jako platforma pro robotiku

LEGO MINDSTORMS a LEGO Technic jsou skvělou platformou pro robotiku, protože jsou jednoduché na používání, ale zároveň přesné. Díky velkému množství rozmanitých dílků LEGO Technic – ozubených kol, hřídelí a ramen je lego snadným konstrukčním prvkem pro stavbu těl a základů robotů. Nízká vstupní hranice ale neznamená i nízký strop. Naopak, lego umožňuje stavbu i složitých konceptů, mechanismů a strojů. Díky intuitivní konstrukci lego a dobrému zapadání dílků do sebe není realizace nápadů nic složitějšího. Ze stejného důvodu je tak lego ideální i na stavbu prototypů a úprav již poskládaných celků.

Nejdůležitější částí celé práce je LEGO MINDSTORMS. To nabízí celou řadu aktuátorů a senzorů, které je jednoduché použít. Hlavním stavebním blokem je ale takzvaný „LEGO EV3 brick“ viditelný na obrázku 2.2.

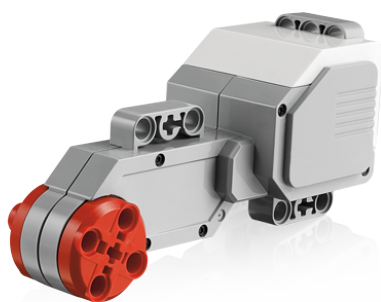


Obrázek 2.2. LEGO MINDSTORMS EV3 kostka. Obrázek je ze stránek Heureka.

Jde o malý počítač používající upravenou verzi Linuxu jako operační systém. Ten ovládá připojené motory a sbírá informace o okolním světě z připojených senzorů. Jde o třetí generaci systému MINDSTORMS (po RCX a NXT). Lego EV3 je snadné naprogramovat i pro naprosté začátečníky, ale podporuje i tvorbu složitějších skriptů přes Python[2].

2.3.1 Možnosti a omezení LEGO MINDSTORMS

LEGO MINDSTORMS EV3 umožňuje připojení až 4 motorů a 4 senzorů najednou. Existují dva druhy LEGO MINDSTORMS EV3 motorů – velký (Large Motor) a menší (Medium Motor) viz obrázek 2.3.



Large Motor



Medium Motor

Obrázek 2.3. LEGO MINDSTORMS motory. Obrázky jsou ze stránek imgbin a ruzovka.

Oba dva jsou servo motory, obsahující enkodér na snímání otáček. Díky tomu umožňují přesné řízení pohybu. Motory se ale liší ve své velikosti a výkonu. Konkrétní specifikace jsou vidět v tabulce 2.1. Hmotnost větších LEGO projektů se pohybuje v řádech kilogramů, což v kombinaci s omezeným prostorem a nutností rychlého pohybu motorů může být velmi omezujícím faktorem.

vlastnost	velký motor	malý Motor
hmotnost [g]	82	39
otáček za minutu	175	260
moment [N · cm]	43	15

Tabulka 2.1. Vlastnosti LEGO MINDSTORMS motorů.[3].

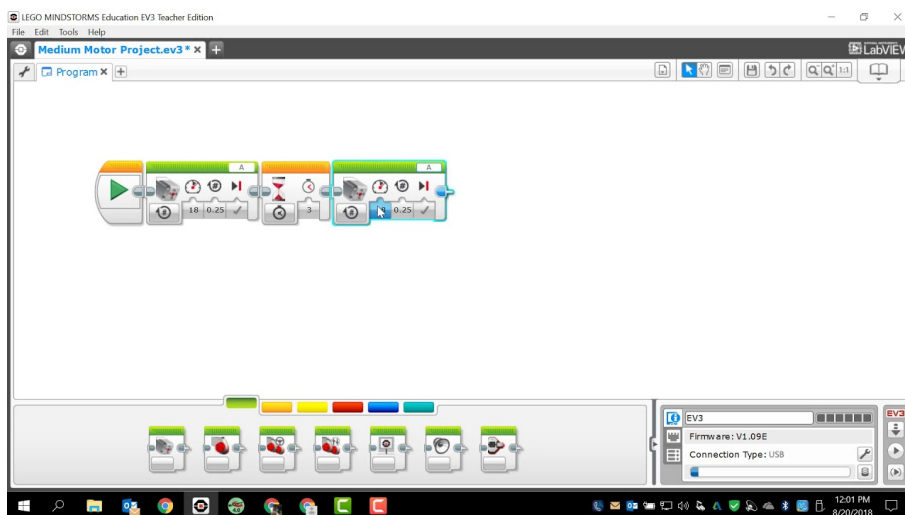
LEGO MINDSTORMS se pyšní celou řadou senzorů. Mezi ty nejpoužívanější patří například následující.

- Ultrazvukový senzor vzdálenosti
- Tlačítko jako dotykový senzor
- Senzor barev
- Gyroskop

Samotná EV3 kostka je vybavena displejem a tlačítky pro snadné ovládání. Dále je schopna se připojit k síti Wi-Fi nebo k jiným zařízením přes Bluetooth. Na kostce jsou dva USB porty. První USB port typu A pro připojení Wi-Fi donglu nebo třeba USB

flash disku a druhý Mini-USB pro připojení k počítači a nahrávání programů. Kostka též obsahuje malý vestavěný reproduktor na přehrávání tónů a nahraných zvuků.

Programování kostky EV3 lze provést mnoha způsoby. Nejjednodušším z nich je využít oficiální LEGO EV3 software LEGO Mindstorms EV3. Jde o vizuální drag & drop blokové schéma, viz obrázek 2.4, tedy funguje na stejném principu jako například Scratch. Jakmile se ale uživatel dostane ke složitějším a větším projektům, stává se vizuální prostředí nepřehledným a je velmi obtížné něco hromadně měnit.[4]



Obrázek 2.4. Ukázka oficiálního LEGO Mindstorms EV3 Home Edition programovacího softwaru.[5]

Druhým způsobem programování kostek EV3 je použití programovacího jazyka Python nebo MicroPython. Lze ale také využít i třeba programovacích jazyků Java nebo C++. To umožňuje použití opravdového programovacího jazyka, což může být vhodné pro zkušenější uživatele nebo pro výuku. Velkou výhodou jsou možnosti Pythonu. Python otevírá dveře k využití daleko složitějších a propracovanějších prvků. Například vzdálenému ovládní kostky kódem přes Bluetooth, to je popsáno v další části práce, konkrétně kapitole 4.3.2. Existuje i celá řada dalších způsobů, jak kostku EV3 ovládat, ale tyto dvě výše zmíněné jsou zdaleka nejpobulárnější[6].

Co se týče nevýhod LEGO MINDSTORMS, hned na první pohled jsou nějaké zřetelné. Kostky EV3 jsou poměrně velké a těžké, oproti novějším LEGO SPIKE Prime. Bez baterií váží zhruba 160 gramů. Hmotnost ani rozměry ale nelze srovnávat s dnešními běžnými mikroprocesory jako jsou třeba ESP32 nebo Arduino. Obdobně je na tom i výkon kostek. Ačkoliv hlavní procesor řady ARM se může pyšnit frekvencí až 300 MHz, pro náročné operace a programy to nemusí být dostatečný výkon. V tabulce 2.2 lze vidět podrobnosti parametrů kostky EV3.

Poslední nevýhodou kostek EV3 je jejich velmi pomalá rychlost zapínání a vypínání. Funkcionalitu robota to neomezuje, nicméně pro vývoj je to nepraktické[cite[7].

2.3.2 Další projekty využívající EV3 pro robotiku

Kostky LEGO MINDSTORMS EV3 a LEGO Technic jsou ideální pro jednoduchou a rychlou realizaci projektů v robotice. Není tedy divu, že existuje velké množství projektů, ve kterých se lze s EV3 setkat. Mezi nejznámější patří například Solvery Rubikovy kostky, viz obrázek 2.5.

Parametr EV3	
Datum vydání	září 2013
Displej	Monochromatický LCD displej s rozlišením 178 × 128 pixelů
Procesor	TI Sitara AM1808 s jádrem ARM926EJ-S @ 300 MHz
Paměť	16 MB
USB port	Ano
Wi-Fi	Volitelný dongl přes USB port
Bluetooth	Ano

Tabulka 2.2. Parametry kostky LEGO EV3. Tabulka je převzata z [7].



Obrázek 2.5. LEGO Mindstorms EV3 solver Rubikovy kostky. [8]

Mezi další známé, jednodušší, patří roboti sledující čáru. Ti by se dali považovat za Hello world robotiky s legem. Mezi zajímavější projekty se pak řadí třeba Sumoboti, kteří se vzájemně vytlačují z kruhu, nebo EV3 Pinball machine.

2.4 Umělá inteligence v chatbot technologiích

2.4.1 Jak fungují moderní chatboti

Základem dnešních moderních chatbotů jsou Velké jazykové modely (Large Language Models, dále už jen jako LLM). Zjednodušeně jsou LLM velmi komplexní funkce, které predikují jedno slovo¹ na základě vstupního textu. Chatboti jsou tedy determinističtí, ovšem náhodně vybírají i slova s nižší pravděpodobností, aby odpovědi působily více lidsky a přívětivěji. Od roku 2017 využívají LLM transformerovou architekturu, která umožňuje paralelní zpracování vstupního textu. Transformery jsou detailně popsány v článku Attention Is All You Need (Vaswani et al., 2017)[9].

Tyto modely jsou trénovány na ohromném množství dat, nejčastějším zdrojem je internet, knihy či rozhovory. Pro lepší představu objemu dat, ze kterých se model učí, je dále uveden příklad. Průměrnému člověku, který by četl nonstop, by trvalo přečíst data, na kterých byl trénován GPT-3 téměř 3000 let. Trénování modelů probíhá ve dvou různých fázích, které jsou krásně vysvětlené ve videu od 3Blue1Brown[10]:

¹ Ve skutečnosti přiřazují pravděpodobnosti ke všem slovům, ze kterých se pak jedno vybere.

- **Self-supervised learning** – jde o hlavní metodu předtrénování modelu. Ten získá na vstup text s chybějícím posledním slovem, které má doplnit. Na základě předpovězeného chybějícího slova se pak zpětnovazebně, pomocí metody backpropagation, upravují parametry modelu, kterých může být i několik miliard. Ty se upravují tak, aby slovo, které bylo původně na konci textu, získalo od modelu vyšší pravděpodobnost nadcházejícího slova.
- **Reinforcement learning z posílené lidské zpětné vazby (RLHF)** – tato část vyžaduje lidskou práci. Během této části lidé procházejí odpovědi a predikce modelu a hodnotí je. Nepravdivé nebo neužitečné odpovědi dostanou špatné hodnocení a model se na základě těchto hodnocení dále učí.

Níže je blíže rozebrán princip funkce LLM a transformerů[10].

- **Tokenizace vstupu** – prvním krokem je převedení textu na čísla. Vstupní text je rozdělen do menších bloků, obvykle slov – tokenů. Tokeny mohou být ale i části slov nebo znaky. Záleží na použitém tokenizeru.
- **Reprezentace tokenů vektory** – tokeny jsou dále převedeny na vektory, které vyjadřují významy tokenů a jejich souvislost s ostatními tokeny v prostoru o několika tisících dimenzí.
- **Zpracování a vrstvy transformeru** – transformer využívá dvou operací. První z nich je „attention“. Ta zajišťuje, aby se vektory tokenů mohly navzájem ovlivnit. Znamená to, že pokud by byl například tokenem „list“, jeho vektor by na začátku vypadal vždy stejně, ale dalšími tokeny jako třeba „strom“ nebo „papír“ by byl různě ovlivněn. Druhou operací je pak feedforward neurální síť. Ta aplikuje nelineární transformace na každý token zvlášť a přidává tím modelu větší kapacitu čerpat z jazykových vzorců naučených během trénování. Tyto dvě operace se střídají a mnohokrát opakují.

2.4.2 Přehled AI chatbotů a jejich využití

Díky předtrénovaným modelům lze vytvořit chatboty, kteří se specializují na nějakou specifickou oblast úkolů nebo jazyk. Je také možné ovlivnit jejich rychlost a přesnost. Vše závisí na trénovacích datech a nastavení parametrů.

V tabulce 2.3 jsou rozebrány silné stránky a použití moderních chatbotů od nejznámějších firem.



Obrázek 2.6. Loga nejznámějších chatbotů a firem.

Chatboti, jak je vidět i v tabulce 2.3, využívají nejrůznější techniky, aby co nejlépe odpovídali vystavenému úkolu. Zde jsou nějaké představené:

- **RAG (Retrieval-Augmented Generation)** – technika kombinující schopnosti generativního modelu se znalostmi předložené odpovídající databáze, což zlepšuje přesnost a spolehlivost a umožňuje vyhledávání v dokumentech[11].
- **Schopnosti agenta** – vybavuje chatbota schopností volat externí služby a funkce, díky kterým dokáže chatbot zvládat komplexnější úkoly, vracet lepší informace nebo dělat rozhodnutí. Mezi tyto funkce může patřit například volání API, využívání výpočetních nástrojů nebo přístup k externím databázím a znalostním systémům.
- **Multimodalita** – lepší a schopnější modely podporují multimodalitu. Ta znamená současné zpracování několika různých vstupních formátů, nejčastěji kombinaci textu s obrázkem. Mezi další formáty se řadí například zvuk nebo video.

Název	Vývojář a model	Zaměření	Silné stránky	Dostupnost
ChatGPT	GPT-4, GPT-4-turbo od OpenAI	Všeobecný asistent, výuka, programování, kreativita	Kvalita výstupu, nástroje, multimodalita	Web, API, Microsoft 365
Gemini	DeepMind a Gemini 1.5 od Google	Asistent pro vyhledávání, multimodální úkoly	Integrace s Google službami, analýza textu i obrazu	Web, API, Android, Google Workspace
Claude	Claude 3 od Anthropic	Výzkum, výuka, dokumenty	Bezpečný, jemný styl, dlouhý kontext (až 200k tokenů)	Web, API, Amazon Bedrock
Copilot	Microsoft s OpenAI GPT-4	Pomoc v MS Office, programování	Výborná integrace s Office, GitHub Copilot pro vývojáře	Microsoft 365, GitHub
Meta AI	LLaMA 2 a 3 od Meta	Open-source vývoj, výzkum, experimenty	Zdarma dostupné, komunitní vývoj	Messenger, Instagram, open-source GitHub

Tabulka 2.3. Tabulka s přehledem moderních chatbotů a jejich specifikací.[12–16]

Se znalostí výše zmíněného není překvapující využití těchto technologií. Chatboti jsou využíváni na mnoha místech, nejsou ale žádnou novinkou. Jednoho takového chatbota si každý člověk nosí v kapse. Hlasoví asistenti, jako jsou Siri, Alexa nebo Google Assistant, jsou nedílnou součástí mobilních zařízení[17]. Dále se už jistě každý setkal s nějakým chatbotem na zákaznické podpoře nebo jiném webchatu či helpdesku. Teprve poslední využití se nově objevuje ve velké míře. Tím je právě robotika a interakce robotů s lidmi.

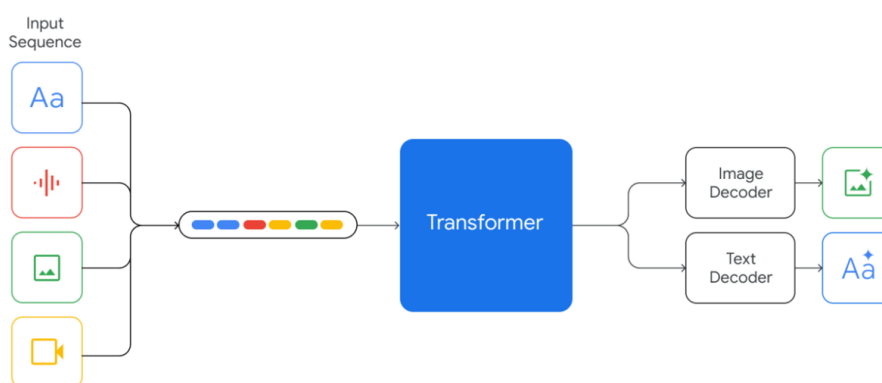
■ 2.4.3 Gemini AI – principy fungování, zpracování textu a obrazu

Gemini AI od společnosti Google DeepMind je esenciální částí hlavy robota Elišky. Tento pokročilý multimodální LLM je nástupcem modelu Bard a stará se o „inteligenci“

Elišky. Tento model má schopnosti pracovat nejenom s psaným textem, ale dokáže velmi dobře pracovat i s obrázky, zvukem, nebo dokonce i videem. Jeho cílem je přímo integrace porozumění a generování textu společně s obrazem.

Gemini využívá architekturu Transformer, která je běžným základem většiny LLM. Schopnost porozumět textu a obrázkům zároveň vzniká už v trénovací fázi modelu. Model byl už při trénování učen hledání souvislostí díky tomu, že trénovací data nebyla pouze text, ale obsahovala i obrázky, tedy multimodálním trénováním. Model se učil na obrovském množství dat pomocí self-supervised learningu a metody maskovaného prediktivního učení. To funguje na stejném principu jako doplňování odebraného slova z věty, jen s obrázky. Část obrázku je odebrána a model se ji pak snaží doplnit a porovnává své doplnění s původním obrázkem[18].

Gemini je schopný porozumět textu i obrazu a kontextu mezi nimi. Umí v nich i hledat souvislosti a dokáže zpracovávat otázky nebo výzvy. Gemini má schopnost řešit vizuální úkoly, rozpoznávat objekty, popisovat scénu a mnoho dalšího. Získané vizuální informace dokáže Gemini kombinovat s kontextem psaného textu. Například: „Popiš, co vidíš před sebou.“, „Jak vypadá člověk, který je před tebou?“ nebo „Co za předmět držím v ruce?“. Stejně jako jsou tokeny (představující slova) reprezentovány vektory, jsou i obrázky reprezentovány pomocí vektorů.



Obrázek 2.7. Přehled rodiny modelu Gemini od Google DeepMind. [19]

Gemini klade velký důraz na bezpečnost a škodlivé informace. Jak již bylo zmíněno, součástí trénování je i kontrola odpovědí lidskými pracovníky (RLHF). To ale není jediná filtrace dat. U Gemini je možné nastavit si vlastní filtry při generování výstupu.

Gemini má tři hlavní varianty. Jsou jimi Ultra, Pro a Nano. Každá z těchto variant je uzpůsobená k plnění jiného úkolu. Liší se v rychlosti a přesnosti odpovědí. V technickém světě je to vždy o balancu rychlosti a výkonu, není možné mít obojí[18].

2.5 Počítačové vidění a rozpoznávání objektů

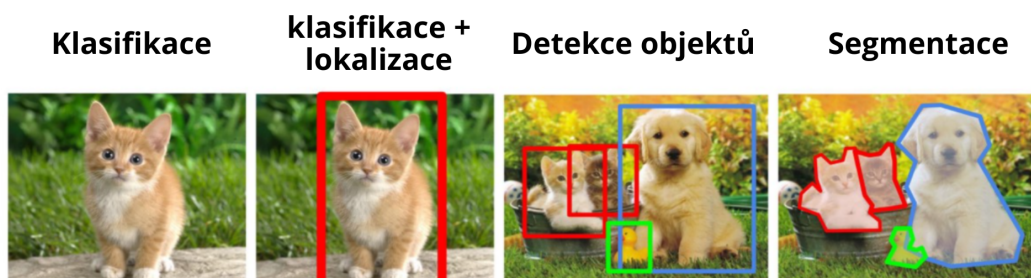
Počítačové vidění (computer vision) je oblast umělé inteligence a strojového učení, která se snaží přidat strojům schopnost „vidět“ stejně jako člověk svým okem. To zahrnuje i interpretaci těchto vizuálních dat.

2.5.1 Základní principy počítačového vidění

Mezi hlavní úkoly počítačového vidění se řadí detekce objektů, rozpoznávání obličejů, sledování pohybu, rekonstrukce 3D scén nebo segmentace obrazu. Moderní přístupy realizující počítačové vidění jsou založeny na metodách **rozpoznávání vzorů (pattern**

recognition). Tato metoda spočívá ve využití konvolučních neuronových sítí (convolutional neural network – CNN). Konvoluční neuronové sítě jsou typem hlubokých neuronových sítí optimalizovaných pro práci s obrazovými daty. Jejich klíčovou vlastností je schopnost automaticky extrahovat prostorové rysy z obrazu pomocí konvolučních vrstev, které aplikují sadu filtrů na vstupní obraz[20].

Tyto sítě jsou trénovány na velkých množstvích anotovaných obrazových dat, například miliony obrázků z datasetů jako ImageNet nebo COCO. Během trénování se síť učí optimalizovat své váhy tak, aby dokázala správně klasifikovat nebo lokalizovat objekty na základě vizuálních znaků. Díky těmto metodám je možné plnit zmíněné úlohy v reálném čase. Ukázka některých úloh je na obrázku 2.8.



Obrázek 2.8. Demonstrace funkcí počítačového vidění. [21]

Celý princip a bližší vysvětlení konvolučních neuronových sítí je podrobně popsán v odborném článku o studii Krizhevskyho a dalších[22]. Zjednodušeně vypadá princip funkce počítačového vidění při zpracovávání obrazu takto:

- **Získání obrazu** z kamery robota, webkamery nebo nahraného souboru. Může jít o video nebo samostatný obrázek.
- **Předzpracování** – často se vstupní data upravují, aby byla práce s obrazem jednodušší, rychlejší nebo vynikly nějaké detaily. Mezi takové úpravy se řadí například převod na stupně šedi nebo normalizace jasu.
- **Extrakce rysů** – hledání podstatných částí obrazu pomocí dalších algoritmů. Podstatné části obrazu záleží na okolnostech, nejčastějšími jsou však hrany v obrázku, tvary, podobné oblasti (barevně) a další.
- **Detekce a klasifikace** – rozpoznání z předem naučených dat, o co se jedná, kde na obrázku je to umístěné. Případně přesnost určení, s jakou model něco detekoval.

2.5.2 OpenCV ke zpracování obrazu

OpenCV (Open Source Computer Vision Library) je jedna z nejrozšířenějších open-source knihoven pro počítačové vidění a zpracování obrazu. Její vývoj byl zahájen již v roce 2000, tehdy pod společností Intel. Dnes je dostupná ve více programovacích jazycích, jako jsou C++, Python, a je aktivně využívána jak ve výzkumu, tak v průmyslových aplikacích.[23]

OpenCV poskytuje rozsáhlý soubor funkcí, které přesahují rámec pouhé detekce objektů. OpenCV obsahuje nástroje pro načítání a ukládání obrázků a videí podporující různé formáty, úpravu obrázků, jako jsou transformace, změny velikosti a barev a další. Součástí knihovny jsou i předtrénované klasifikátory pro detekci obličejů a dalších objektů, a to i včetně sledování pohybu ve videu. Knihovna podporuje i vytváření vlastních modelů, a tak jsou v ní dostupné i moduly pro klasifikaci nebo regresi. Kromě toho knihovna obsahuje nástroje, které pomohou při kalibraci kamery, a následnou práci s vytvářením 3D scén, k čemuž poslouží i odhadování hloubky.[23]

2.5.3 Rozpoznávání obličejů a sledování pohybu

Rozpoznávání obličejů a sledování pohybu (například rukou) je zásadní schopností pro interakci s uživatelem. Umožňuje robotům sledovat uživatele očima, zaměřit pohled na mluvčího nebo otáčet hlavu za sledovaným objektem. Knihovna OpenCV[23] pro tyto účely nabízí třídu `cv2.CascadeClassifier`, která umožňuje detekci obličejů pomocí tzv. Haarových kaskád[24]. Tato metoda využívá binární klasifikátory založené na jednoduchých vizuálních rysech, jako je kontrast mezi oblastmi, které se skládají do silnějšího klasifikátoru metodou AdaBoost. Přestože se tato technika řadí mezi starší, je díky své výpočetní nenáročnosti stále hojně využívána pro detekční úkoly.

Pro sledování rukou lze využít například modul `HandTrackingModule`[25], který je postaven nad knihovnou OpenCV a nástroji z MediaPipe[26]. Tento modul umožňuje detekovat pozici ruky v obraze, identifikovat prsty, sledovat jejich pohyb v čase a další. Takové funkce mohou být dále využity pro gestikulaci, ovládání robota nebo rozšíření konverzačního kontextu.

2.6 Rozpoznávání řeči a generování hlasu

Rozpoznávání řeči a generování hlasu jsou dvě klíčové vlastnosti, které umožňují přirozenou komunikaci člověka se strojem. **Rozpoznávání řeči (Automatic Speech Recognition, dále jako ASR)** převádí mluvené slovo na text, čímž napomáhá strojům porozumět hlasovému vstupu. Na druhé straně stojí **generování hlasu (Text-to-Speech, dále jako TTS)**, které plní funkci opačnou – ze vstupního textu vytváří zvukové stopy syntetizované řeči, kterou může uživatel slyšet.

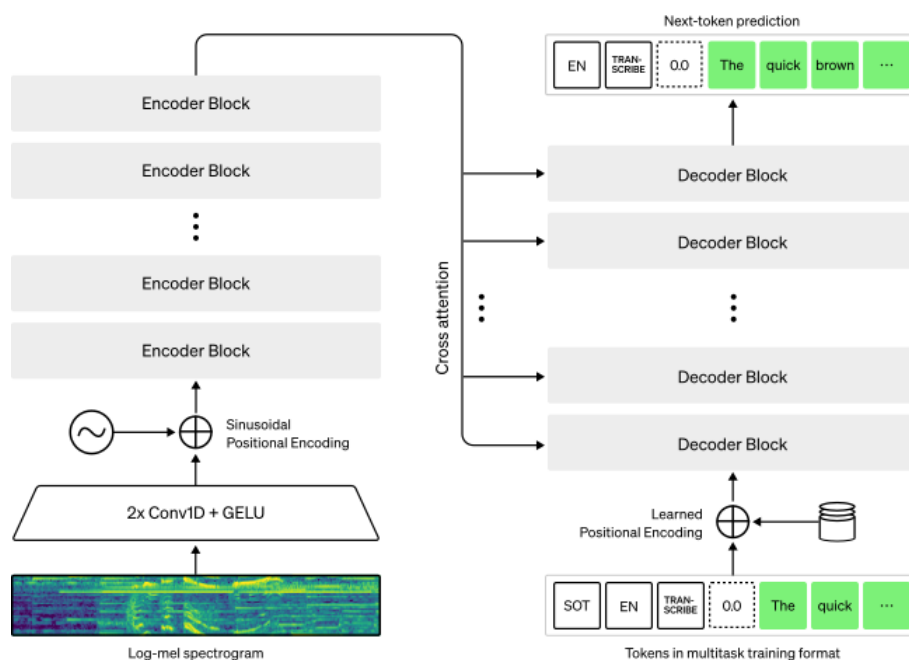
2.6.1 Whisper od OpenAI – princip fungování

Whisper je open-source model pro automatické rozpoznávání řeči vyvinutý společností OpenAI. Tento model byl trénován na až 680 000 hodinách multilingválních a multitaskových dat z internetu, která zahrnují rozhovory, podcasty, pohovory nebo třeba TED talk. Takto rozsáhlá a různorodá data zajišťují vysokou robustnost, schopnost rozpoznat až 90 jazyků a překlad.[27]

Název	Počet milionů parametrů	I čistě anglický model	Potřebná VRAM [GB]	Relativní rychlost
Tiny	39	ano	~1	~10x
Base	74	ano	~1	~7x
Small	244	ano	~2	~4x
Medium	769	ano	~5	~2x
Large	1550	ne	~10	~1x
Turbo	809	ne	~6	~8x

Tabulka 2.4. Tabulka s přehledem modelů Whisper a jejich parametry.[28]

Whisper využívá architekturu Transformer, ve které se střídají encoder a decoder bloky, využívající mechanismus attention, díky kterému je propagována informace mezi těmito bloky. Na obrázku 2.9 je vidět, jak model nejdříve převádí zvukovou nahrávku



Obrázek 2.9. Jak funguje Whisper od OpenAI. [29]

na mel spektrogram ², který použije jako vstup, ze kterého predikuje sekvenci textových tokenů. Tuto sekvenci navíc používá k přesnějšímu odhadu nadcházejícího tokenu.

Jako vždy se dostáváme k balancování rychlosti s přesností. Whisper nabízí několik modelů, které disponují různými rychlostmi v závislosti na velikosti modelu, jeho přesnosti a podporovaných jazycích. V tabulce 2.4 jsou uvedeny parametry těchto modelů.

■ 2.6.2 ElevenLabs – generování hlasového výstupu z textu

Systémy pro Text-to-Speech (TTS) vytvářejí z textového vstupu syntetizovaný mluvený výstup. Jednou z předních a velmi realistických služeb je dnes ElevenLabs. Jde o cloudovou platformu zakládající si na pokročilých modelech hlubokých neuronových sítí.

ElevenLabs nabízí moderní přístup generování lidského hlasu end-to-end, oproti staršímu skládání řeči ze segmentů nebo syntaktických pravidel. Modely jsou schopné intonace, zapojit do hlasu emoce a podobné vlastnosti, díky učení se na rozsáhlých datech z reálné řeči. ElevenLabs disponuje velkým množstvím, přes 5000, již předtrénovaných hlasů, které ovládají nejrůznější jazyky. Kromě toho ale podporuje i vytvoření vlastního virtuálního hlasu.[30]

■ 2.6.3 Synchronizace řeči s pohybem pusy

Pro systémy, které kombinují mluvený výstup s jejich vizuální stránkou, je důležitou součástí synchronizace pohybu úst a čelistí s řečí, jakou systém produkuje. Též známý jako lip-sync[31], je podstatný u například animovaných postav nebo robotických hlav. Existuje více metod a způsobů, jak zaručit, aby se ústa otvírala „do rytmu“ vydávaného hlasu. Níže jsou uvedeny některé z nich.

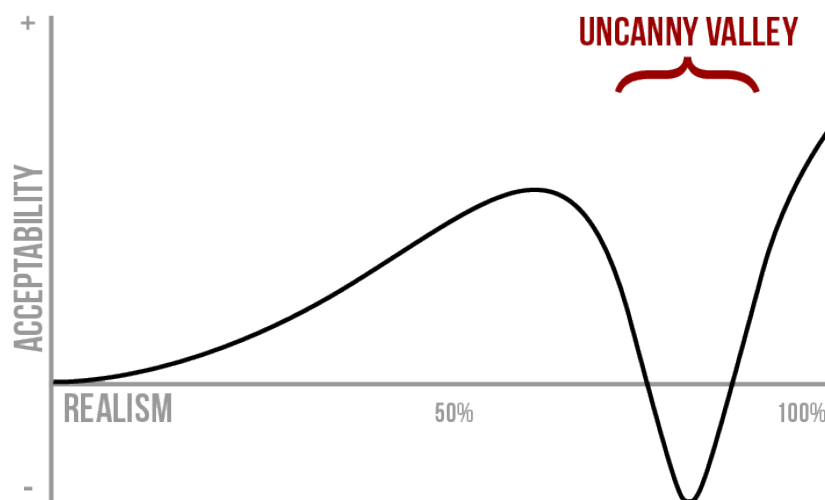
² Spektrální reprezentace zvuku, která zachycuje, jak se mění frekvence v čase. Stupnice je přizpůsobená lidskému sluchu.

- **Časování** – prakticky nejjednodušší metoda. Pohyb pusy je řízen (například cyklickým otevíráním a zavíráním) podle délky zvukového výstupu.
- **Signálový lip-sync** – otevírání úst je řízeno podle intenzity nebo frekvence signálu na výstupu. Například pomocí Fast Fourier transform (FFT), RMS energie nebo velikosti amplitudy.
- **Fonémový lip-sync** – lip-sync je pomocí této metody nejpřesnější. Výstupní text je převeden na fonémy, to jsou nejmenší základní zvukové jednotky řeči, které rozlišují význam slov. Každému z fonémů je přiřazena odpovídající poloha úst.

2.7 The uncanny valley efekt

Efekt Uncanny valley, v češtině jako „Údolí podivnosti“, je pojem popisující zvláštní až nepříjemný pocit, vyvolaný robotem nebo postavou, která se snaží vypadat lidsky, ale ne úplně dokonale.

Tento fenomén, objevující se při pohledu na věc, která téměř přesně vypadá jako člověk, ale něco na ní neseďí, pojmenoval japonský robotik Masahiro Mori[32]. Jde o jev, kdy se nám robot líbí čím dál víc, čím víc vypadá jako člověk – ale jen do určité chvíle. Pak se to náhle obrátí a začne lidem připadat nepříjemný nebo až děsivý. Tento fenomén je znázorněn grafem 2.10.



Obrázek 2.10. Propad vizuální přitažlivosti robota kvůli Uncanny valley efektu. [33]

Postavy nebo roboti, kteří jsou jasně rozlišitelní od člověka, například kreslení, animovaní nebo legračně vypadající, jsou vnímáni jako roztomilí. Stejně tak lidé, ti jsou přijati přirozeně. Problém ale nastane, když se robot nachází někde mezi. Stačí k tomu jen velmi málo, jako třeba zvláštní pohled, nepřirozené mrkání nebo nějaká jiná odchylka mimiky od běžného člověka.

Kapitola 3

Návrh robotické hlavy Eliška

Cílem bylo vytvořit co nejpřirozenější formu verbální i neverbální komunikace a vtáhnout tak uživatele do interakce, která připomíná rozhovor s člověkem. V této sekci je rozebráno, jak byly vybrány jednotlivé prvky Elišky a jak přispívají k naplnění tohoto cíle.

3.1 Požadavky na systém

3.1.1 Jaké pohyby a funkce má robotická hlava mít

Jak je uvedeno v sekci 2 cíle práce, interakce s Eliškou by měla probíhat i neverbální komunikací. V této sekci jsou představeny fyzické schopnosti hlavy Elišky. Při představě dvou konverzujících lidí člověka napadne hned několik znaků neverbální komunikace, které podvědomě děláme.

Prvním z nich je sledování druhé osoby, se kterou hovoříme. Je slušné dívat se do očí tomu, s kým vedeme konverzaci, a tak by se měla chovat i Eliška. Eliška by měla být zvědavá a přátelská, a tak by měla prozkoumávat okolí. Tyto dvě představy vedou k ovládnutí očí, a to jak horizontálně, tak vertikálně. Samotné pohyby očí ale nestačí, naopak. Kdyby se na hlavě pouze hýbaly oči, velmi rychle se propadneme do údolí podivnosti 2.10 a Eliška bude působit až děsivě. Při sledování nějakého objektu hýbe člověk celou hlavou. Proto by měla Eliška mít tuto schopnost také. Otáčení hlavy umožňuje Elišce větší rozhled, a i z dálky bude viditelné, že něco dělá.

Druhým, poměrně výrazným prvkem neverbální komunikace, je obočí. Ačkoliv si to nemusíme uvědomovat, lidské obočí hraje důležitou roli při vyjadřování emocí, a to nejen při konverzaci. Eliška by tedy měla umět hýbat svým obočím.

Nejdůležitějším prvkem, který je ale tak základní, že člověka ani nemusí napadnout, je otevírání pusy. Kdyby Eliška pouze vydávala zvuk a pusa se jí vůbec nehýbala, rozhodně by to nepůsobilo lidsky a ani dobře. Otevírání pusy správně, aby sedělo do mluveného slova, není lehké. Nicméně náhodné otevírání pusy robota neúprosně přesune směrem pryč od lidského vyjadřování.

3.1.2 Jaký má být způsob interakce s uživateli

V této podsekci jsou popsány různé způsoby interakcí s Eliškou, tedy jak může uživatel robota vnímat a komunikovat s ním, nebo jak naopak může robot reagovat na uživatele.

- **Hlasová komunikace** – základním a nejdůležitějším prvkem interakce je komunikace. Uživatel může na Elišku mluvit a Eliška dokáže řeč rozpoznat a rozumět jí. To se projeví tím, že Eliška dokáže smysluplně odpovědět a vést konverzaci. Eliška by neměla skákat lidem do řeči a měla by je poslouchat, když je to žádoucí. Mimo jiné Eliška rozumí i jiným jazykům než češtině (na výběr je z více než stovky dalších), je tedy možné na ni mluvit i například anglicky.

- **Zrakové vnímání** – robotická hlava Elišky rozpoznává v reálném čase lidské tváře nebo ruce a dokáže je sledovat. Stejně tak zvládá Eliška i rozpoznávat předměty v jejím okolí.
- **Neverbální odezva** – jak již bylo rozebráno v předchozí podsekcí 1, Eliška reaguje i neverbálně, to zahrnuje mimiku obličeje, sledování uživatele nebo otáčení hlavy.
- **Reprezentativní znalosti** – aby byla naplněna reprezentativní část projektu, Eliška má k dispozici vědomosti o univerzitě, fakultě a studiu. Tyto informace je Eliška schopná sdělit uživateli, pokud je na ně dotázána. Eliška má navíc postavené tričko (podstavu hlavy) v barvách fakulty.
- **Multimodální kontext** – Eliška zvládá kombinovat mluvené slovo a vidění. Projevem multimodálního kontextu je především schopnost reagovat na hlasové výzvy uživatele související se získáváním informací z okolního prostředí. Typicky jde o popis nějakého předmětu nebo odhadování jeho vlastností.

3.1.3 Přehled technologií nutných k realizaci

Z požadavků na systém a samotného zadání lze vyčíst řadu nutných technologií, které jsou potřebné k realizaci celého projektu. Všechny komponenty tvořící Elišku je možné rozdělit do jedné ze dvou skupin – buď hardwarové, nebo softwarové. Toto rozdělení a zároveň přehled je v tabulce 3.1.

Hardwarové komponenty	Softwarové komponenty
LEGO MINDSTORMS EV3 kostky	Python
LEGO servo motory	Bluetooth komunikace
LEGO TECHNIC dílky na tělo	Chat bot
Hardware, na kterém poběží kód	Rozpoznávání hlasu
Kamera	Generování hlasu
Mikrofon	Vidění
Reproduktor	

Tabulka 3.1. Tabulka s přehledem potřebných komponent tvořících robotickou hlavu Elišky.

Jednotlivé technologie jsou dále v této kapitole podrobně rozebrány. Pro každý požadavek jsou uvedeny konkrétní návrhy řešení s objasněním důvodů, proč byly zvoleny, včetně možných alternativ, porovnání a doplňujících úvah.

3.2 Mechanická konstrukce

3.2.1 Návrh mechanismů pro pohyb

Pro realizaci mimických a pohybových funkcí robotické hlavy Eliška bylo nutné navrhnout a sestavit několik nezávislých mechanismů, které jsou ovládány pomocí LEGO motorů. Při návrhu bylo třeba zohlednit omezený prostor uvnitř hlavy, požadovaný rozsah pohybů, mechanickou stabilitu a možnost přesného řízení jednotlivých částí.

Vývoj mechanismů byl usnadněn díky existenci projektu robotické hlavy Dave od Creative Mindstorms, který již řeší řadu technických problémů spojených s animací obličeje. Protože ale neexistují žádné oficiální instrukce na sestavení konstrukce hlavy,

sloužily dostupné fotografie a videa jako výchozí bod při její realizaci. Tyto funkční koncepty byly dále upraveny, rozšířeny a přizpůsobeny specifickým požadavkům projektu Eliška – například s důrazem na větší přesnost pohybu, kompaktnější a robustnější provedení nebo synchronizaci s řídicím softwarem.

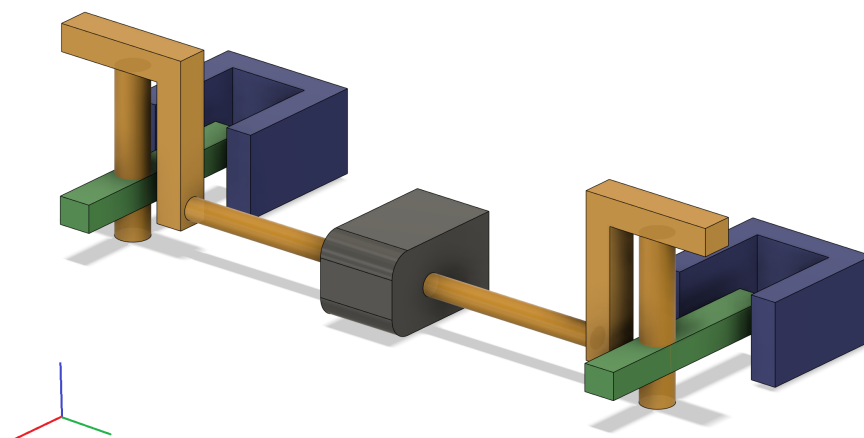
Každý z použitých mechanismů je navržen cíleně pro svou konkrétní funkci (např. zvednutí obočí, otevření pusy) a zajišťuje zároveň potřebnou robustnost při dlouhodobém provozu a náročnějším zacházením.

3.2.2 Horizontální a vertikální pohyb očí

Mechanismus pro ovládání pohybu očí je navržen tak, aby umožňoval nezávislý horizontální i vertikální pohyb obou očí současně. Jak funguje, lze vidět na obrázku 3.1. Základ celého systému, na kterém vše drží, tvoří pevná konstrukce (šedá část).

Vertikální pohyb je zajištěn pomocí žlutých ramen, která jsou spojena se zelenými nosníky a společně se otáčejí. Osa otáčení je zelená osa, přičemž ramena se opírají o pevnou konstrukci a otáčejí se dopředu a dozadu, čímž naklánějí oči podle požadovaného směru.

Horizontální pohyb zajišťují dvě modré vodící části, které posouvají zelené nosníky, nesoucí samotné oči, do stran. Tyto nosníky jsou mechanicky spojeny a ovládány pomocí jediné osy vedené skrz centrální konstrukci, což zajišťuje synchronizovaný pohyb obou očí do stran. Modré části se pohybují ve směru zelené osy.



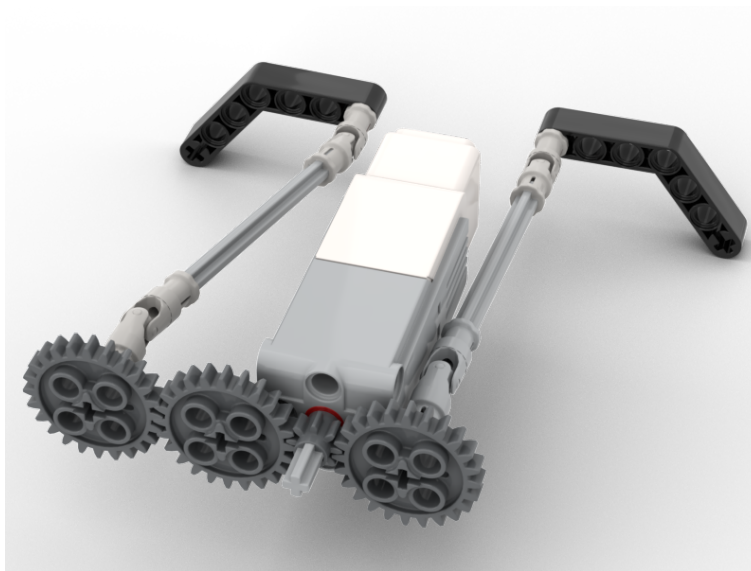
Obrázek 3.1. Návrh mechanismu vertikálního (žlutě) a horizontálního (modře) pohybu očí (zelená)

Celý mechanismus je kompaktní a navržený tak, aby se vešel do omezeného prostoru uvnitř hlavy, zároveň však poskytuje dostatečný rozsah pohybu a možnost přesného řízení pomocí LEGO motorů.

3.2.3 Pohyb obočí

Pro vyjádření emocí, jako je překvapení nebo zamračení, byl navržen jednoduchý zvedací mechanismus pro pohyb obočí. Obočí jsou ovládána prostřednictvím jednoho LEGO motoru, který řídí soustavu ozubených kol. Při návrhu tohoto mechanismu bylo klíčové efektivní rozmístění komponentů v omezeném prostoru uvnitř hlavy. Z důvodu nedostatku místa v přední části byl LEGO motor i s ozubenými kolečky umístěn do zadní části konstrukce.

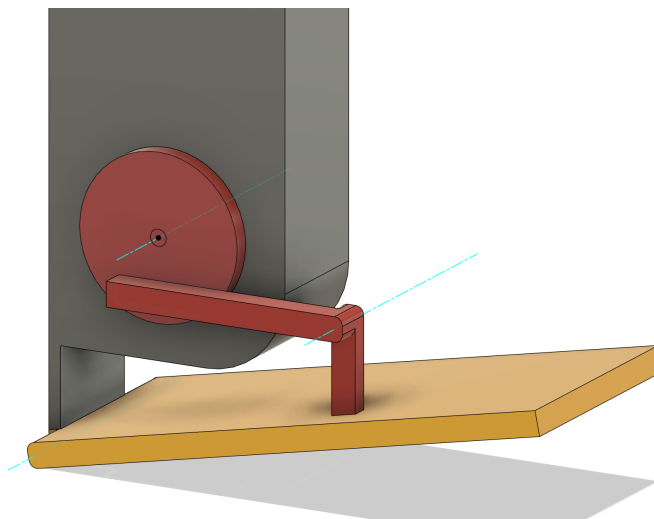
Otáčivý pohyb motoru je ze zadní části přenášen dopředu pomocí dlouhé hřídele. Ta je na obou koncích spojena s ohebnými točivými klouby (tzv. ‘universal joints‘), které umožňují přenos rotačního pohybu i v případě drobného vychýlení osy. Díky tomu je zajištěno plynulé a spolehlivé řízení ozubeného mechanismu ovládajícího obočí na přední straně hlavy. Tento mechanismus umožňuje souběžný pohyb obou obočí nahoru a dolů. Na obrázku 3.2 je vidět pohled na mechanismus obočí zezadu.



Obrázek 3.2. Pohled zezadu na návrh mechanismu obočí

■ 3.2.4 Otevírání pusy

Spodní čelist robota je ovládána pomocí klikového mechanismu, který převádí rotační pohyb motoru na vertikální pohyb čelisti. Základnu tvoří pevná šedá konstrukce, ke které jsou připevněny všechny části mechanismu.



Obrázek 3.3. Návrh mechanismu pohybu dolní čelisti

Červený kotouč (klikový disk) na obrázku 3.3 je otáčivá část motoru. K disku je excentricky uchyceno vodící rameno, které se při otáčení pohybuje nahoru a dolů. Druhý

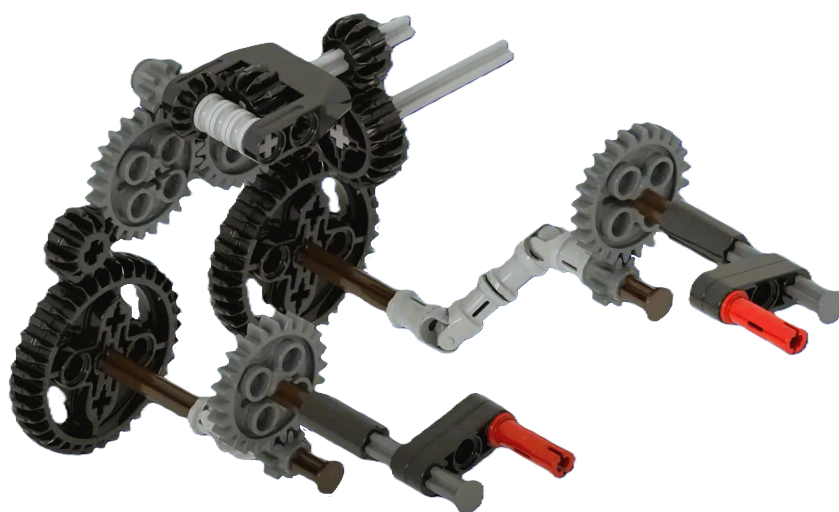
konec ramene je připojen k svislé části spojené se žlutou deskou představující spodní čelist. Ta je v zadní části uchycena otočným kloubem ke konstrukci, což jí umožňuje kývavý pohyb.

Díky tomuto uspořádání se při otáčení motoru čelist zvedá a klesá, čímž vzniká realistický pohyb otevírání a zavírání úst.

■ 3.2.5 Hýbání koutků

Pohyb koutků úst je realizován pomocí jednoho LEGO motoru, který ovládá dvě páky (vypadající jako ručičky u hodin umístěných v rozích úst) prostřednictvím soustavy ozubených kol a převodů. Koutky slouží ke změně tvaru rtů tvořených červenou gumičkou. Vzhledem k velmi omezenému prostoru v oblasti obličeje robotické hlavy Elišky a velikosti motoru nebylo možné motor umístit přímo k ústům. Z tohoto důvodu je motor umístěn v zadní části hlavy, odkud je pohyb přenášen až do oblasti koutků.

Návrh převodového systému byl výrazně ovlivněn prostorovými omezeními – v oblasti úst musel zůstat volný prostor pro spodní čelist, která se pohybuje dolů při otevírání pusy, a zároveň nesmělo dojít ke kolizi s mechanismy očí a obočí, které jsou umístěny za nosem a očima. Systém převodů je proto velmi složitý a vyžadoval pečlivé plánování i sestavení, aby bylo dosaženo funkčního přenosu pohybu v takto stísněných podmínkách. Jak je ale k dočtení níže v sekci 4.1.3, realizace tohoto mechanismu byla natolik složitá, že z původního návrhu se mnoho nezachovalo a výsledek vypadá jinak. Návrh i výsledné řešení mělo základ v mechanismu robotické LEGO hlavy DAVE, jak je vidět níže na obrázku 3.4.

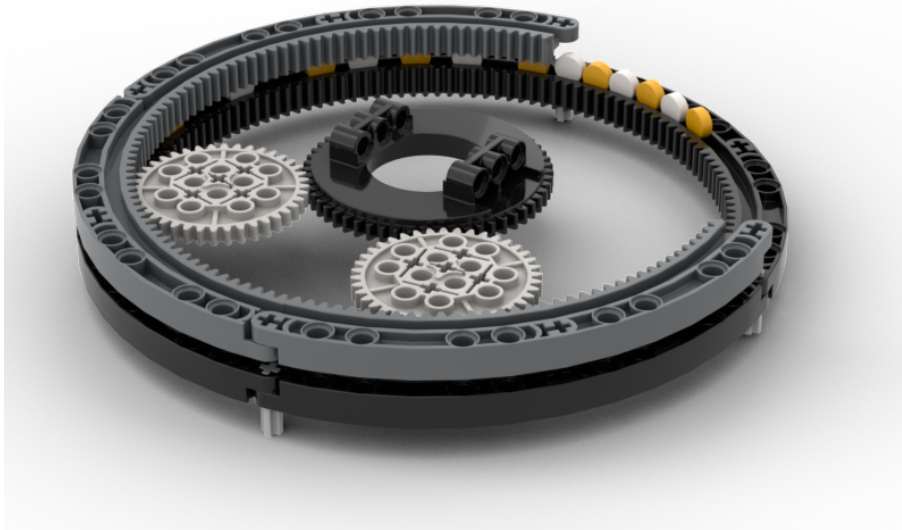


Obrázek 3.4. Návrh mechanismu koutků.[1]

■ 3.2.6 Otáčení hlavy

Otáčení celé robotické hlavy je zajištěno pomocí velkého LEGO ložiska, na kterém hlava přímo stojí. Toto ložisko, ačkoliv není oficiálním dílem LEGO, bylo sestaveno z jiných dílků LEGO poněkud netradičním způsobem, kvůli vysoké hmotnosti hlavy, neboť poskytuje dostatečnou stabilitu a zároveň umožňuje plynulý otočný pohyb.

Ložisko zároveň slouží jako základ pro planetovou převodovku, která je integrována do mechanismu otáčení. Planetová převodovka byla zvolena z několika důvodů – především zajišťuje vyšší přesnost polohování, větší odolnost vůči zatížení a také poskytuje vyšší



Obrázek 3.5. Návrh mechanismu krku – planetové převodovky a LEGO ložiska

točivý moment při nižších otáčkách motoru. Celý mechanismus tak umožňuje plynulé, stabilní a dostatečně silné otáčení hlavy i při její vyšší hmotnosti.

Na obrázku 3.5 je černé ozubené kolo jako prostřední část planetové převodovky, velká kola s ozubením uvnitř tvoří ložisko, svírající malinké mincovité dílky LEGO. Velké šedé kolo s ozubením uvnitř tvoří podstavu celé hlavy, naopak černé je pevnou součástí ramen a těla.

3.3 Elektronické vybavení

3.3.1 Dvě EV3 kostky

Kvůli požadavkům na co největší interaktivitu hlavy je zapotřebí ovládat nemalé množství motorů. Z předchozí podsekcce 3.2 je vidět, že i přes snahu použít co nejmenší počet motorů, jich je potřeba 6. Jak již bylo dříve zmíněno ve druhé kapitole v podsekcce 2.3.1 kostky EV3 mají každá pouze po 4 portech pro motory. Z toho důvodu je potřeba použít kostky dvě. V první kostce jsou zapojeny oba motory očí, motor ovládající obočí a motor krku. To dává druhé kostce na starosti řízení pusy a koutků. Ve stavebnici LEGO ale existují moduly pro rozšíření počtu výstupů, díky kterým by stačilo použít pouze jednu kostku. To by se projevilo na menší celkové kapacitě baterií, a tedy i kratší době provozu Elišky bez přímého napájení ze sítě, a proto tyto moduly nejsou použity.

Během návrhu celé robotické hlavy se naskytla možnost využít kostku Brian 3.6. Stavebnice Brian je obdoba LEGO MINDSTORMS EV3, která je vyvíjena na Fakultě elektrotechnické ČVUT. Kostka je modernější a mnohem výkonnější. Bohužel má ale jednu nevýhodu, která zamezila jejímu užití. Jelikož je Brian stále ve fázi vývoje, není s ním zatím možné komunikovat ani přes Wi-Fi, ani přes Bluetooth. To by znamenalo, že veškerý kód by musel běžet přímo na kostce. Vzhledem k velikosti projektu to vede k množství komplikací, které silně převažují pozitivita.



Obrázek 3.6. Ukázka kostky Brian vyvíjené na ČVUT.[34]

Umístění obou kostek v těle Elišky je nedílnou částí plánování a návrhu. Důvodů je hned několik. Prvním z nich je samotné zapojení motorů. Kostka musí být umístěna tak, aby se do ní daly zapojit všechny motory. To samo o sobě není složité, ale po uvedení dalších důvodů začne být problém jasný. Kostky musejí být v těle Elišky umístěny tak, aby byly stále přístupné. Opět z vícero důvodů – kostky je potřeba před používáním vždy zapnout, a tak musí být tlačítka na přední straně kostky dostupná. Ze stejného důvodu je potřeba mít viditelný i displej kostky. Ten je potřeba i pro párování přes Bluetooth. Je také nutné mít možnost kostky z Elišky vyndat. To z důvodů výměny baterií nebo případně celé kostky a také přístupu ke konektorům kostky. Vyměňovat stále baterie je ale nepraktické, protože kostka má poměrně velkou spotřebu, a tak existují kryty s dobíjecími bateriemi. EV3 kostky jsou v Elišce umístěny tak, aby k nim mohl za provozu vést napájecí kabel.

Jedním z klíčových omezení při umísťování EV3 kostek je jejich hmotnost a rozměry. Vzhledem k tomu, že se používají dvě kostky, není možné je obě umístit přímo do hlavy, protože vnitřní prostor je velmi omezený a již zaplněný jinými komponenty. Navíc je celá hlava pro LEGO motory poměrně těžká, a proto další zatížení v této oblasti není vhodné.

Dalším faktorem ovlivňujícím umístění kostek je vizuální stránka konstrukce. Aby působila robotická hlava co nejrealističtěji, je žádoucí, aby EV3 kostky nebyly na první pohled viditelné. Tento problém je řešen pomocí odnímatelných krytů, které kostky vizuálně zakrývají, ale zároveň umožňují snadný přístup pro údržbu nebo úpravy.

Všechny tyto omezení, faktory a potřeby vymezují jedno místo, kde mohou být kostky umístěny. Je jím zadní strana podstavy Elišky, tedy oblast lopatek u běžného člověka. Toto umístění má i tu výhodu, že hmotnost kostek pomáhá zatížit Eliščinu podstavu pro větší stabilitu.

■ 3.3.2 Připojené motory a jejich funkce

Robotická hlava Elišky využívá celkem 6 LEGO motorů. Z toho jsou 4 motory large a 2 medium. Ty jsou rozděleny mezi dvě kostky LEGO EV3 následujícím způsobem. Jedna kostka je zaměřená na všechny funkce týkající se pusu a mluvení. Druhá kostka se soustředí na oči a funkce simulování vidění. Kostky jsou na základě tohoto rozdělení pojmenovány jako EV3_sight a EV3_mouth a jejich rozdělení motorů i s pokrytím portů je popsáno v tabulce 3.2.

Port	EV3_sight	EV3_mouth
A	Medium motor – obočí	/
B	Large motor – oči vertikálně	/
C	Medium motor – oči horizontálně	Large motor – koutky
D	Large motor – krk	Large motor – čelist

Tabulka 3.2. Tabulka s obsazením portů jednotlivých kostek EV3.

3.3.3 Telefon jako senzor – propojení mikrofону, kamery a reproduktorů

Pro účely komunikace s Eliškou mluveným slovem jsou nezbytné dvě věci – mikrofón pro záznam řeči a reproduktor, aby mohla hovořit i Eliška. LEGO Technic nenabízí oficiální mikrofón ani reproduktor, proto bylo nutné zařídit je jinak. Kostka LEGO MINDSTORMS EV3 má integrovaný reproduktor, ten je však velmi slabý a pro potřeby Elišky nedostatečný. LEGO nabízí oficiální „dílek“ s kamerou, kterou je možné připojit k EV3 kostce. Tato kamera se již sice nevyrábí, ale na fakultě je k dispozici. Vzhledem k tomu, že kód, který obstarává veškerou funkčnost, neběží na LEGO kostce, stává se kostka pouze prostředníkem, který komunikuje s aktuátory. Díky tomu není nutné data ze senzorů (jako je třeba mikrofón nebo případná kamera) posílat do EV3 kostky, ale přímo do počítače. To umožňuje použít i jiné než LEGO senzory. Kvůli tomu by bylo sbírání vstupních dat do kostky zbytečným krokem.

Nejjednodušší cesta, která se nabízí, je využití mobilního telefonu. Ten obsahuje všechna tři potřebná zařízení. Má integrovaný mikrofón, reproduktor i kameru. Protože posbíraná data je potřeba dostat do počítače, na kterém běží hlavní program, použití mobilního telefonu se zdá být ještě atraktivnější. Existuje množství aplikací, které dokážou pohodlně streamovat zvuk (jako reproduktor i jako mikrofón) i video obraz přes síť Wi-Fi. Celkově telefon ulehčuje sběr dat a přehrávání hlasu Elišky do jednoho zařízení, se kterým je snadné komunikovat, a zároveň díky tomu dělá hlavu kompaktnější, protože v ní nemusí být více modulů s různými zařízeními.

3.4 Softwarová architektura

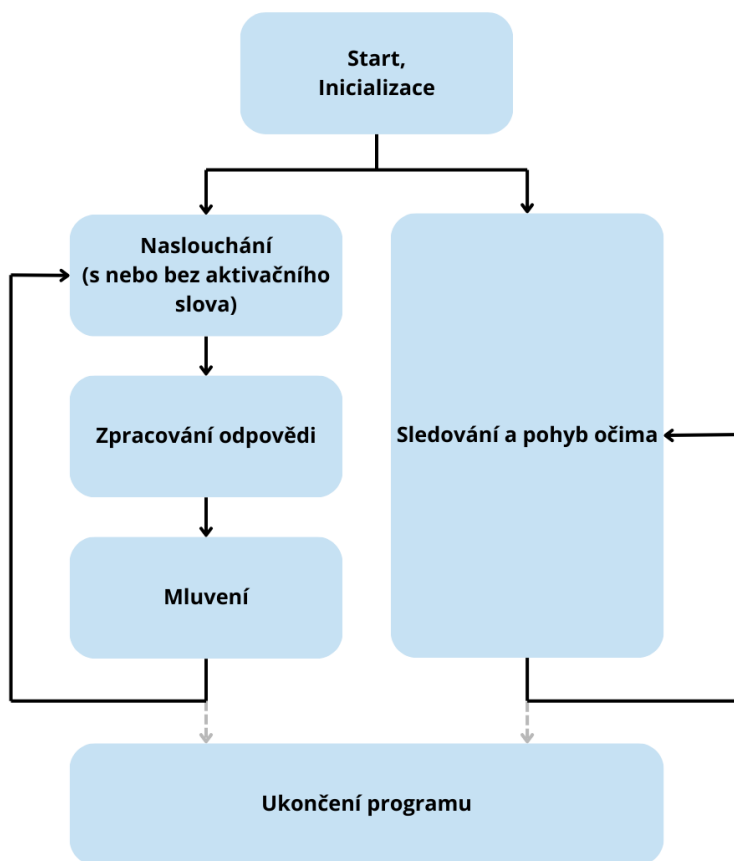
3.4.1 Struktura kódu

- `main.py` – hlavní spouštěcí program, který koordinuje fungování a interakci všech modulů.
- `ev3Functions_class.py` – `EV3Handler` – zajišťuje komunikaci s LEGO EV3 kostkami a řízení všech motorických prvků.
- `vision_class.py` – `visionHandler` – zajišťuje zpracování obrazu a detekci obličejů, rukou i polohy uživatele.
- `gemini_class.py` – `GeminiHandler` slouží k přímé komunikaci s jazykovým modelem Gemini AI a k vyhodnocení textového i vizuálního vstupu.
- `speak_class.py` – `speakHandler` – převádí odpověď modelu na řeč a synchronizuje pohyb čelisti s mluveným výstupem.
- `audio_class.py` – `Listening` – obsluhuje rozpoznávání lidské řeči a aktivačního slova. Jedná se o Eliščin sluch.

- `featureFunctions_class.py` – modul obsahující doplňkové funkce, které může AI chatbot zavolat pro získání dalších informací (počasí, informace o škole, ...).
- `threadHandler.py` – tato třída spravuje paralelní vlákna pro vizualizaci a sledování obličejů a rukou a také pohyby hlavy a očí s tím spojenými.

3.4.2 Hlavní řídicí smyčka a průběh interakce

Po spuštění programu se inicializují jednotlivé moduly a spouští se hlavní řídicí smyčka, která zajišťuje interakci mezi uživatelem a robotickou hlavou. Smyčka je navržena jako nekonečný cyklus, který poslouchá uživatele a odpovídá v reálném čase. Komunikace probíhá ve smyčce, která čeká na aktivační slovo (např. „hey eli“). Po jeho zachycení se systém dostane do fáze, kdy naslouchá a následně odpoví. Po odpovědi smyčka končí a vrací se zpět na začátek, tentokrát ale už nečeká na aktivační slovo a rovnou naslouchá dále. K pochopení pomůže zjednodušená vizualizace průběhu kódu na obrázku 3.7.



Obrázek 3.7. Zjednodušený diagram průběhu hlavního programu.

Průběh jedné iterace cyklu (a opakováním celá interakce) vypadá následovně. Při prvním probuzení či před zahájením nové konverzace systém čeká na aktivační slovo. Po jeho zaznění dostane uživatel zpětnou vazbu od robota změnou jeho výrazu – pozvednutí obočí. V ten moment začne robot poslouchat a detekovat řeč. Až uživatel domluví, vstup je zpracován přes Whisper. Pokud vyprodukovaný text byl rozeznán s vysokou přesností, použije se jako vstup pro chatbota Gemini AI. Odpověď se pak převede na zvukovou stopu a je přehrána. K tomu Eliška vyjádří emoci korespondující s odpovědí. Mluvení je doprovázeno pohybem čelistí. Až robot domluví, dojde i hlavní smyčka na

konec a začne znovu. Pokud je vstup od uživatele některou z klíčových frází, Eliška se vypne, nebo přejde do fáze čekání na klíčové slovo, tedy zahájení nové konverzace.

Kapitola 4

Realizace robotické hlavy

4.1 Stavba fyzického modelu

Po dokončení návrhu byly jednotlivé pohybové mechanismy robotické hlavy Eliška postupně sestaveny pomocí stavebnice LEGO. Hlavní důraz byl kladen na přesné dodržení rozměrů skutečné hlavy, mechanickou stabilitu, robustnost, prostorovou koordinaci všech částí a možnost hlavě udělat na konci hezčí design. Každý mechanismus byl postaven samostatně, otestován, a teprve poté integrován do finální konstrukce hlavy.

Složitější byla především stavba v oblasti obličeje, kde se nachází většina pohyblivých částí. Zde bylo nutné zajistit, aby mechanismy navzájem nezasahovaly do svého pohybového prostoru – zvláště u očí, obočí, pusy a nejvíce koutků. Vzhledem k omezenému prostoru bylo nutné některé komponenty modifikovat oproti původnímu návrhu, například změnit některé převody nebo předělat způsob uchycení.

Celá konstrukce byla navržena tak, aby bylo možné mechanismy demontovat pro účely oprav nebo ladění a také s důrazem na robustnost celé hlavy. Zároveň jsou všechny mechanismy konstrukce zabalené v celistvé vrstvě kostek LEGO, aby se paruka Elišky nezachytila do ozubených kol.

4.1.1 Konstrukce jednotlivých pohybových mechanismů

Při zahájení stavby robotické hlavy Elišky byl původně zvolen postup, kdy měla být nejprve postavena celá hlava jako uzavřená dutá skořápka, do které by byly postupně vkládány jednotlivé mechanismy. Tento přístup měl výhodu v tom, že bylo možné hned od počátku pracovat s vymezeným prostorem, a tím předejít případným kolizím mezi komponenty a mít jasně rozvržený tvar hlavy, jak je vidět na obrázku 4.1.



Obrázek 4.1. Původní realizace skořápky hlavy Elišky

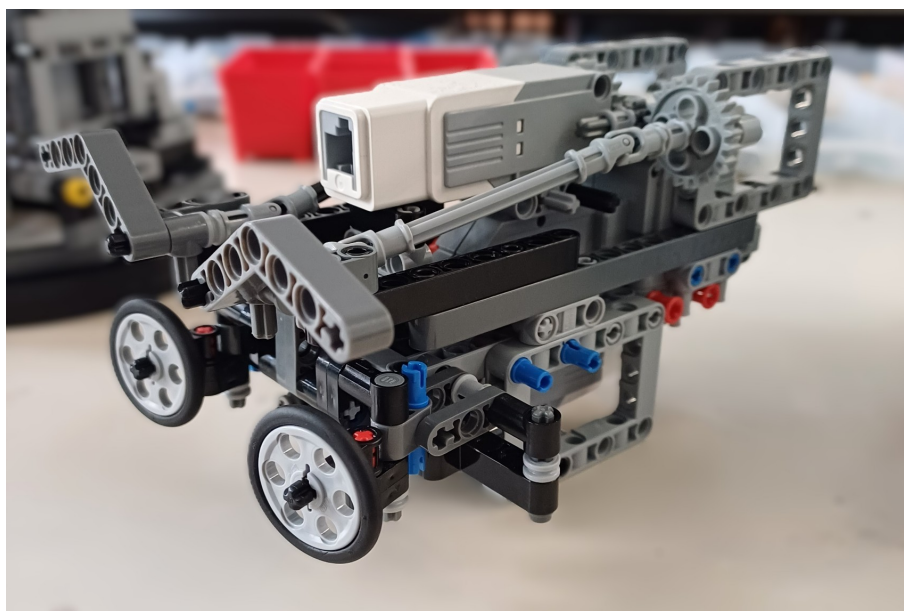
Během realizace se však ukázalo, že takovýto postup není vhodný. Uzavřená konstrukce výrazně omezovala přístup k vnitřním částem, a tím komplikovala jak návrh jednotlivých mechanismů, tak i jejich testování a případné úpravy. Z těchto důvodů byl přístup změněn: jednotlivé mechanismy byly nejprve navrhovány a stavěny samostatně jako funkční bloky, a teprve poté byly postupně integrovány do výsledné konstrukce hlavy. Tento postup se ukázal jako mnohem flexibilnější a umožnil lépe vyřešit prostorové nároky jednotlivých pohybových systémů. Zároveň bylo možné každý mechanismus otestovat a případně upravit ještě před jeho vestavbou do finální podoby. Na závěr byla celá hlava uhlazena, aby mechanismy nevyčnívaly ven a hlava vypadala lépe.

V následujících podsekcích jsou popsány jednotlivé části konstrukce, konkrétní řešení a úpravy, které byly během realizace provedeny. Pohybové mechanismy jsou doplněny ilustračními fotografiemi z výstavby.

4.1.2 Oči a obočí

Stavba očí byla poměrně snadná a proběhla bez větších obtíží. Mechanismus z robotické hlavy Dave byl skvělou inspirací. I přes to bylo nutné některé jeho části drobně upravit, aby se po dlouhém používání nepovolily a více držely. Také byly zvětšeny modré části z obrázku 3.1, aby se nestalo, že nosníky očí vypadnou z ohrádky a přestanou tak fungovat.

Mechanismus pro pohyb obočí byl konstrukčně velmi jednoduchý. Celý modul je složen pouze ze čtyř ozubených kol, jednoho motoru a univerzálních kloubů, což umožnilo jeho kompaktní propojení s modulem pohybu očí. Díky celkově úspornému a prostorově efektivnímu řešení bylo možné umístit všechny tři motory zajišťující mimiku v oblasti očí těsně vedle sebe. Moduly očí a obočí tak bylo možné spojit do jednoho celku, který se následně stal výchozím bodem pro další konstrukci celé hlavy Elišky.



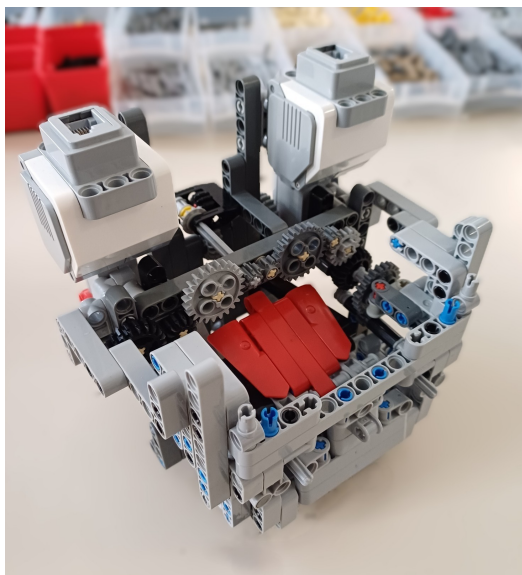
Obrázek 4.2. Postavené a spojené moduly očí a obočí s dočasnými očima

4.1.3 Pusa a koutky

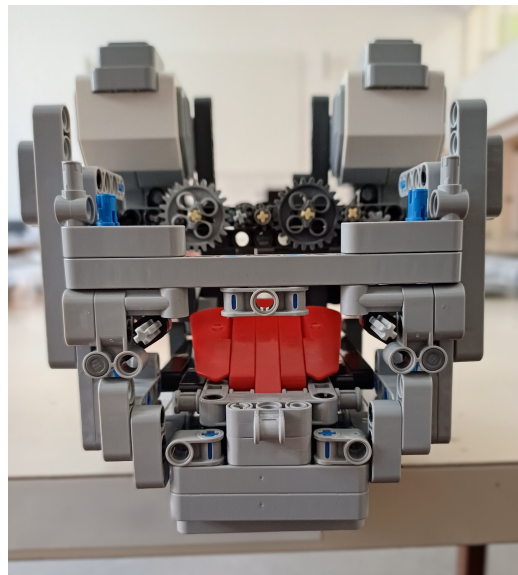
Konstrukce dolní čelisti přinesla několik technických výzev. Samotný mechanismus otevírání není složitý, avšak jeho realizace vyžadovala zvláštní pozornost kvůli mechanic-

kému namáhání. Aby mohl robot otevírat ústa rychle a plynule, a tím věrně napodobovat mluvení, musela být čelist dostatečně pevná a odolná. Tato požadovaná robustnost se ukázala jako problém zejména v oblasti brady, která se při prudkých pohybech čelisti začala po určité době uvolňovat nebo odpadávat.

Další komplikací byla nepřírozně vysoká pohyblivost čelisti, která působila rušivě. Tento problém byl naštěstí snadno řešitelný – pohyb čelisti byl mechanicky omezen fyzickým dorazem, který zabránil příliš velkému otevření úst. I přesto však zůstával klíčovým požadavkem dostatečně robustní návrh celé spodní čelisti, protože při vysokých rychlostech a silách docházelo k její deformaci nebo rozpojení jednotlivých částí.



Pohled ze shora



Pohled ze předu

Obrázek 4.3. Moduly pusy a koutků

I přes jasnou představu o konstrukci mechanismu pro pohyb koutků úst se nakonec původní návrh ukázal jako nevyhovující a musel být v průběhu stavby několikrát přepracován. Hlavním problémem byl extrémně omezený prostor v oblasti čelisti (viz sekce 3.2.5), který značně komplikoval vedení převodů a umístění mechanických spojů. Níže jsou uvedeny problémy, kvůli kterým bylo nutné systém koutků změnit a proč:

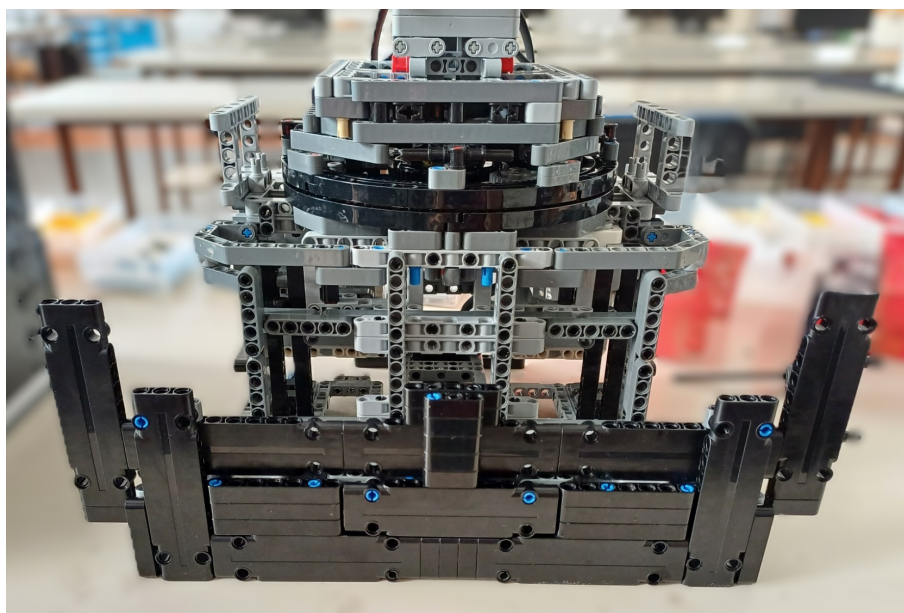
- **Nespolehlivost převodu** – při testování se opakovaně ukázalo, že větší množství LEGO ozubených kol v převodové soustavě způsobuje ztuhlou odezvu pohybu. V některých místech také docházelo k nežádoucímu prokluzování mezi spojenými ozubenými koly. Zvláště problematické bylo místo na kraji pusy, kde jsou koutky úst připojeny ke konstrukci – zde vzniká atypické mechanické spojení, které je citlivé na přesnost a stabilitu. V kombinaci s původní konstrukcí spodní čelisti, která v některých částech nedržela dostatečně pevně, docházelo k protáčení kol, drhnutí mechanismu a zaostávání jednoho z koutků za druhým.
- **Opačný směr koutků** – během testování se opakovaně stávalo, že každý z koutků úst se pohyboval opačným směrem. Tento problém byl způsoben nesprávným počtem ozubených kol v převodovém systému. Najít správný počet kol, jejich umístění a navrhnout jejich vedení tak, aby nezabíralo příliš prostoru a zároveň bylo mechanicky stabilní, se ukázalo být podstatně náročnější, než se původně očekávalo.

- **Nejednotný převod** – další komplikací byl nesynchronní pohyb obou koutků. Kvůli velmi omezenému prostoru uvnitř hlavy bylo obtížné najít vhodné místo pro vedení převodů tak, aby byla zachována jejich symetrie. V některých případech se podařilo najít funkční řešení propojující motor s oběma koutky, ale po sestavení se ukázalo, že rychlost pohybu každého koutku je jiná. To vedlo k nepřírozenému a nesouměrnému výrazu obličeje.

■ 4.1.4 Podstava a detaily

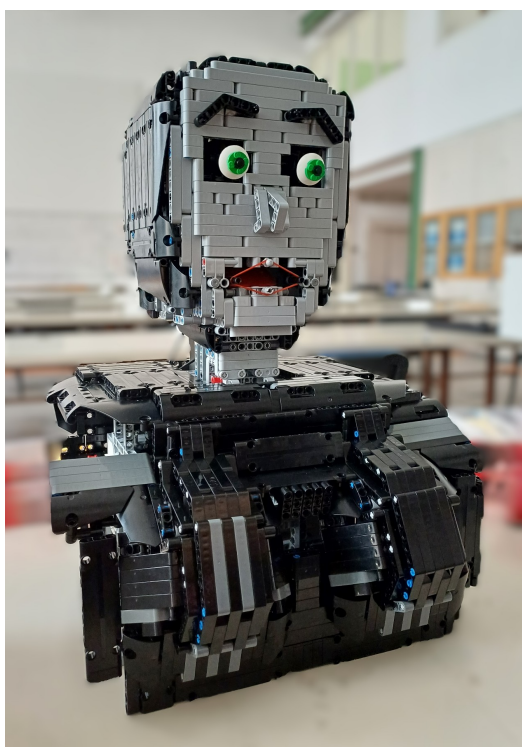
I po sestavení všech jednotlivých pohybových modulů nebyla konstrukce hlavy Elišky kompletní. Pro její plnou funkčnost bylo nutné navrhnout a postavit i podstavec s rameny, který slouží jako nosná základna a zároveň ukrývá klíčové komponenty systému. Uvnitř podstavce je umístěn motor pro otáčení krku, mobilní telefon (sloužící jako kamera, mikrofon i reproduktor) a obě LEGO EV3 kostky, které zajišťují řízení všech motorů.

Součástí finální úpravy bylo také vizuální „zaobalení“ celé konstrukce, které jí dodalo kompaktní a přirozenější vzhled. Cílem bylo, aby výsledek nepůsobil jako otevřený stroj s viditelnými komponenty, ale připomínal skutečnou hlavu. Mobilní telefon je umístěn v přední části podstavce tak, aby jeho kamera směřovala vpřed a měla co nejlepší výhled do okolí, ale zároveň nebyla rušivým elementem – optika je zarovnána s malým otvorem ve středu těla, čímž je zajištěno nenápadné „vidění“ Elišky. Umístění EV3 kostek odpovídá návrhu popsanému v sekci 3.3.1.



Obrázek 4.4. Držák na telefon v podstavě Elišky

Posledním, ale vizuálně velmi důležitým detailem, se staly vlasy robotické hlavy. V počáteční fázi vývoje byla snaha vytvořit účes z LEGO dílů, avšak tento přístup nevedl k uspokojivému výsledku. Nakonec byla použita běžná paruka, která výsledné podobě Elišky dodala výrazně lidštější a přirozenější vzhled.



Dokončení podstavy a zakrytí děr



Paruka robotické hlavy Eliška

Obrázek 4.5. Poslední úpravy podstavy a celé hlavy

Použití paruky však přineslo i nové technické výzvy. Bylo nutné zakrýt všechny pohyblivé části a převody v horní části hlavy tak, aby nehrozilo zachycení vlasů do mechanismu. Výsledná úprava tedy musela kombinovat estetickou funkci s ochranou vnitřních komponent.

4.2 Ovládání motorů a realizace pohybů

4.2.1 Programování EV3 pro řízení pohybů

Pro řízení motorů LEGO EV3 není v tomto projektu nutné nahrávat žádný samostatný program přímo do kostek, jako tomu obvykle bývá. EV3 fungují pouze jako vykonávací jednotky, které přijímají příkazy ze zařízení, na kterém běží hlavní program v jazyce Python, přes Bluetooth. Jediným požadavkem je, aby byly kostky předem spárovány přes Bluetooth s počítačem, ze kterého jsou následně ovládány.

Díky tomuto přístupu je veškerá logika i řízení soustředěno v jednom hlavním programu, konkrétně v jednom samostatném modulu s třídou `ev3Handler`, což zjednodušuje ladění i správu kódu.

4.2.2 Převod příkazů z Pythonu do motorů EV3

Přenos příkazů z hlavního programu v Pythonu do LEGO EV3 kostek je realizován prostřednictvím knihovny `ev3_dc`[35], která umožňuje přímou komunikaci přes rozhraní Bluetooth. V projektu jsou použity dvě EV3 kostky, každá z nich má přidělené specifické motory: jedna řídí oblast očí (`ev3_sight`) a druhá oblast pusy (`ev3_mouth`) přesně tak, jak je uvedeno v tabulce 3.2.

Třída `ev3Handler` zapouzdřuje veškerou logiku spojenou s připojením, konfigurací a řízením motorů. Při spuštění se pomocí předem uložených adres v konfiguračním souboru (`ev3_config.json`) naváže spojení s oběma kostkami a zkonfiguruje se chování jednotlivých motorů včetně jejich rychlostí a rozsahů pohybu.

Každý motor je obsluhován samostatným objektem knihovny `ev3_dc.Motor`, přičemž jsou využívány různé metody pro jiné způsoby ovládání.

- `start_move()` – pro otáčení motoru ve zvoleném směru a rychlosti
- `start_move_to()` – pro přesný pohyb motoru do požadované polohy
- `start_move_for()` – pro otáčení motoru po stanovenou dobu
- `stop()` – pro zastavení motoru

Logika řízení motorů je navržena tak, aby se vždy kontrolovalo, zda už motor není blízko požadované pozice – tím se předchází zbytečnému pohybu. Tato kontrola probíhá přes metodu `within_tolerance()`, která porovnává aktuální a cílovou pozici s povolenou odchylkou.

Modul `ev3Handler` dále obsahuje funkce, které abstrahují vyšší úroveň chování, například:

- `move_to_emotion(emotion)`, která nastaví polohu obočí a koutků na základě emocí definovaných v konfiguračním souboru,
- `sync_jaw(indata)`, která ovládá pohyb čelisti v reálném čase podle intenzity zvuku během mluvení,
- `eye_tracking(position)`, která zajišťuje plynulé sledování objektu očima i hlavou.

Celý systém je navržen tak, aby bylo možné robotickou hlavu ovládat plně z Pythonu, bez potřeby nahrávání vlastního programu do EV3 kostek. To výrazně zjednodušuje vývoj i ladění, protože je možné v reálném čase upravovat chování bez nutnosti kompilace a nahrávání kódu do jednotlivých zařízení.

4.3 Implementace kódu v Pythonu

4.3.1 Struktura celého systému a hlavní program

Struktura celého systému byla navržena modulárně a její návrh i datový tok jsou podrobně popsány v části 3.4. V této části je prezentována přímo implementace v jazyce Python, která tyto návrhové principy převádí do praxe. Hlavní program `main.py` slouží jako centrální řídicí bod, kde jsou všechny komponenty systému propojeny a řízeny.

Inicializace modulů – na začátku programu jsou importovány jednotlivé třídy a moduly, které zajišťují specifické funkce systému.

```
from code.ev3Functions_class import ev3Handler as EV3
from code.gemini_class import GeminiHandler as Gemini
from code.vision_class import visionHandler as Vision
...
```

Každý modul je navržen jako samostatná třída. Jejich instance jsou vytvořeny v kontextu `with`, což zajišťuje správné spuštění a ukončení prostředků, zejména připojení k EV3 nebo běh paralelních vláken.

```
with EV3() as ev3:
    vision = Vision(verbose=2)
    gemini = Gemini(vision.set_mode, feature.get_weather_data, ...)
    speaking = Speak(sync_jaw=ev3.sync_jaw, ...)
    listening = listen(verbose=2)
```

Hlavní smyčka systému – po inicializaci se spustí hlavní smyčka programu, ve které dochází k rozpoznání vstupu a výstupu systému.

```
while True:
    if not justTalked:
        listening.wait_for_keyword("hey eli")
        justTalked = True
    ...
    prompt, audible = listening.speech_recognition()
    ...
    response, emotion = gemini.get_chat_response(prompt,
                                                  vision.get_frame())
    speaking.say(response, emotion)
```

Systém nejprve čeká na aktivační frázi. Po jejím zaznamenání dojde k rozpoznání vstupu uživatele, zpracování dotazu modelem Gemini AI a nakonec ke generování řeči a mimiky odpovídající obsahu a náladě odpovědi.

Celý systém je navíc navržen tak, aby bylo možné softwarovou část Elišky spustit i bez fyzické části, tedy bez robota z kostek LEGO. To velmi usnadňuje práci na úpravách kódu a částí, při kterých není fyzický model potřebný. Zároveň to dává za vznik možnosti vybrat si vstupní zařízení, ať už jde o kameru či mikrofon nebo výstupní zařízení v podobě reproduktoru.

4.3.2 Propojení EV3 s Pythonem přes Bluetooth

Propojení mezi Pythonem a LEGO EV3 je zajištěno pomocí knihovny `ev3_dc`, která umožňuje přímé ovládání motorů přes Bluetooth bez nutnosti programování přímo na straně EV3. Veškeré řízení motorů probíhá z hlavního počítače, což umožňuje centralizované řízení všech pohybů a snadnější ladění i údržbu systému. To už ale bylo zmíněno v sekci 4.2, proto je zbytek zaměřen na ukázky částí kódu.

Společnou částí každého souboru s nějakou třídou je získávání klíčových konfiguračních dat z „config“ souborů¹. Třídy při inicializaci otevřou tyto soubory a vyberou si z nich důležitá data nutná pro inicializaci. V případě třídy `ev3Handler` jsou to například Bluetooth adresy LEGO EV3 kostek, rychlosti a rozsahy motorů, nebo informace o kameře a předdefinované pozice motorů pro vyjádření emocí. Díky tomu je zajištěno snadné přepojení nebo výměna kostek bez nutnosti měnit zdrojový kód – stačí upravit konfigurační soubor.

```
with open('config/ev3_config.json', 'r', encoding="utf-8") as file:
    info = json.load(file)
    SIGHT_ADDRESS = info["EV3_SIGHT_ADDRESS"]
    MOUTH_ADDRESS = info["EV3_MOUTH_ADDRESS"]
    ...
    self.EMOTIONS = info["emotions"]
```

¹ To se bude opakovat v každém souboru, nicméně je zbytečné stále stejné vysvětlení opakovat.

Po navázání spojení se vytvoří instance jednotlivých motorů, které jsou připojeny ke konkrétním portům na kostkách a následně jsou motory nastaveny na příslušné rychlosti a další parametry z konfiguračního souboru.

```
self.m_eye_vertical = ev3.Motor(ev3.PORT_B, ev3_obj=self.ev3_sight)
self.m_neck = ev3.Motor(ev3.PORT_D, ev3_obj=self.ev3_sight)
...
self.m_jaw = ev3.Motor(ev3.PORT_D, ev3_obj=self.ev3_mouth)
```

Před samotným spuštěním systému je nutné inicializovat výchozí polohy všech motorů, aby bylo zajištěno správné chování jednotlivých pohybových mechanismů. K tomu slouží speciální metody, jako například `center_eyes(vertical_speed, horizontal_speed)`, která nastaví oči do centrální polohy na obou osách.

Každý motor nebo skupina motorů má v rámci třídy `ev3Handler` přiřazenou vlastní metodu, která zajišťuje konkrétní typ pohybu. Díky tomu je možné v hlavním programu nebo vyšších metodách volat pohyby v intuitivní podobě, jako například `open_jaw()`, `move_eyebrows(listening)` nebo `eye_tracking(position)`, aniž by bylo potřeba se starat o nízkoúrovňové ovládání motorů.

```
def move_to_emotion(self, emotion: str = "neutral") -> None:
    if self.use_Mindstorms:
        # Get the target positions
        pos_mouth, pos_eyebrow = self.get_motor_positions(emotion)

        # Move the mouth and eyebrows to the target positions
        self.move_mouth(pos_mouth)
        self.move_eyebrows(pos_eyebrow)
```

Podobně je řízen také pohyb čelisti během mluvení, a to prostřednictvím metody `sync_jaw(indata)`, která pracuje s daty o hlasitosti výstupu. Tato data jsou získávána z reproduktoru vybaveného funkcí `loopback`, která umožňuje současně přehrávat zvuk a zpětně měřit jeho intenzitu.

```
def sync_jaw(self, indata) -> None:
    if self.use_Mindstorms:
        # Normalize the volume
        volume_norm = np.linalg.norm(indata)*10
        # If the volume is below the threshold, move the jaw up,
        # otherwise move it down
        if volume_norm < self.JAW_SOUND_THRESHOLD:
            self.move_jaw(direction=1)
        else:
            self.move_jaw(direction=-1)
```

4.3.3 Detekování a rozpoznání hlasu

Pro ovládání robotické hlavy hlasem a „lidskou“ konverzaci je klíčová schopnost rozpoznat mluvený vstup uživatele. Tuto úlohu zajišťuje třída `Listening`, která je zodpovědná za detekci klíčového slova, záznam řeči a její následný přepis do textu. Celý modul byl navržen s důrazem na spolehlivost, nízkou latenci a odolnost vůči okolnímu šumu.

Při inicializaci třídy `Listening` dochází k načtení konfiguračních parametrů ze souboru (např. citlivost mikrofону, doba ticha, prahové hodnoty hlasitosti, vzorkovací frekvence nebo název souboru, kam se bude uložený zvuk zapisovat aj.). Kromě základního

nastavení mikrofону je zde také vytvořen buffer pro záznam zvuku a nastaveny kontrolní proměnné, které určují, kdy má začít ukládání hlasu a kdy má skončit. Zvláštností je, že systém dokáže upravit záznamovou hladinu a délku uvažovaného ticha podle toho, zda probíhá detekce klíčového slova nebo běžného hlasového vstupu. Mimo jiné modul inicializuje i následující.

- **Porcupine** pro detekci klíčového slova offline
- **PvRecorder** pro kontinuální poslech přes správný mikrofón
- **WebRTC VAD** pro detekci hlasové aktivity
- **Whisper** od OpenAI pro finální přepis hlasu do textu
- **interní Queue** pro zpracování streamovaného zvuku

Detekce aktivačního slova je řešena pomocí knihovny Porcupine, která umožňuje detekci předem definované fráze (např. „hey Eli“) přímo na zařízení bez nutnosti připojení k internetu. Funkce `wait_for_keyword()` spustí nepřetržité čtení zvukových rámců a předává je modelu Porcupine.

```
def wait_for_keyword(self, wanted_keyword: str) -> None:
    keyword_index = -1
    while keyword_index < 0:
        pcm = self.recorder.read()
        keyword_index = self.ppn.process(pcm)

    self.recorder.stop()

    if self.VERBOSE == 2:
        print(f"[DEBUG] [AUDIO_LISTEN] Keyword
              '{wanted_keyword.lower()}' detected.")
```

Jakmile je klíčové slovo detekováno, poslech se ukončí a systém přejde do režimu aktivního naslouchání uživatelského vstupu.

Detekce a záznam řeči – funkce `listen()` zajišťuje záznam mluvené řeči pomocí vícestupňové detekce. Prvním krokem je změření hladiny okolního hluku, která se využije k dynamickému nastavení prahové hodnoty.

Záznam probíhá v reálném čase pomocí `sounddevice.InputStream`, přičemž každá příchozí zvuková jednotka (frame) se zpracovává ve funkci `process_audio_frame()`. Ta využívá WebRTC VAD[36] ke klasifikaci, zda se jedná o řeč. Logika zajišťuje:

- zachování krátké historie zvuku před detekcí (tzv. pre-speech buffer),
- zahájení nahrávání po určitém počtu navazujících „řečových“ rámců,
- filtrování šumu,
- ukončení nahrávání při delším tichu,
- ořezání tichých segmentů.

```
def process_audio_frame(self, frame, vad, pre_speech_data,
                       speech_data, speech_frames, recording,
                       start_time):
    is_speech = vad.is_speech(frame, self.RATE)
    ...
    if is_speech:
        speech_frames += 1
    else:
```

```

        if speech_frames > 0:
            speech_frames -= 1

    # Start recording if continuous speech is detected
    if speech_frames >= self.SPEECH_THRESHOLD and not recording:

        recording = True
        speech_data = bytearray()
        start_time = time.time()

        # Include pre-speech buffer in recording
        while not pre_speech_data.empty():
            speech_data.extend(pre_speech_data.get())

    if recording:
        speech_data.extend(frame)

    return speech_frames, recording, start_time, speech_data

```

Po ukončení nahrávání (dostatečně dlouhém vstupu) je nahrávka uložena do WAV souboru a systém přechází k přepisu.

Přepis řeči pomocí Whisper – po uložení souboru probíhá přepis řeči pomocí modelu Whisper Turbo. Zvukový záznam je načten, upraven oříznutím ticha a převeden do mel spektrogramu, který je vstupem pro model. Výstupem metody je `string` detekované otázky uživatele a `boolean` `audible`, který nese informaci, zda byl vstup dostatečně kvalitní k dalšímu zpracování.

```

audio = self.preprocess_audio(audio)
audio = whisper.pad_or_trim(audio)

mel = whisper.log_mel_spectrogram(audio,
    n_mels=self.model.dims.n_mels).to(self.model.device)

options = whisper.DecodingOptions(language="cs", fp16=False)
result = whisper.decode(self.model, mel, options)
detectedQuestion = result.text

if result.text.strip() == "":
    print("[AUDIO_RECOG_WARNING] Nothing recognized.")
elif result.no_speech_prob > 0.8:
    print("[AUDIO_RECOG_WARNING] No words recognized
        in voice file.")
elif result.avg_logprob < -1.5:
    print("[AUDIO_RECOG_WARNING] Low probability, recognition
        is not accurate.")
else:
    audible = True
    ...
return detectedQuestion, audible

```

Jak je v kódu vidět, Eliška je nastavená na rozpoznávání českého jazyka. Tuto možnost lze vypnout, aby robot rozuměl všem jazykům, nicméně je potom rozpoznávání řeči pomalejší. Bohužel už tak je rozpoznávání textu z nahrávky nejpomalejším článkem, a tak byla snaha ho co nejvíce urychlit.

■ 4.3.4 Gemini AI jako chatbot a zpracování obrazu

Robotická hlava Eliška využívá pokročilý jazykový model Gemini od společnosti Google pro dvě hlavní funkce: vedení konverzace a zpracování vizuálních podnětů. Obě tyto schopnosti jsou implementovány ve třídě `GeminiHandler`, která spravuje komunikaci s modelem a zajišťuje předávání potřebných údajů ze systému, jako jsou vizuální vstupy, funkce nebo kontextové proměnné.

Při vytvoření instance `GeminiHandler` se načítá konfigurace z následujících souborů `config.json`, `prompt.txt` a `vision_prompt.txt`. Soubory `.txt` jsou zahájení konverzace s chatbotem, aby odpovídal pouze v požadovaném formátu² a s vědomím, že je robotická hlava z kostek LEGO. Další důležité věci, které jsou předány při inicializaci, jsou bezpečnostní pravidla (aby chatbot neodpovídal vulgárně apod.) a seznam funkcí, které může volat (například `get_weather_data`, `get_scene_description`). Je důležité zmínit, že třída vytvoří dvě různé instance modelu.

- **ELISKA_MODEL** – pro konverzaci s podporou automatického volání nástrojů (tzv. function calling).
- **VISION_MODEL** – specializovaný model pro zpracování obrazu a generování popisů scén.

Po inicializaci je vše připraveno ke generování odpovědi. K jejímu získání slouží metoda `get_chat_response(prompt, img)`. Metoda předá zadaný text (např. dotaz uživatele) do aktivní chatovací relace, která si uchovává kontext předchozí konverzace.

```
def get_scene_description(self, prompt: str) -> str:
    # Use the vision model to get a description of the scene
    response =
        self.VISION_MODEL.generate_content([self.img, prompt])
    scene_description = response.text.strip()

    return scene_description
```

Odpověď modelu může být rozšířena o tzv. funkční volání – v případě potřeby model automaticky zavolá některou z předdefinovaných funkcí (např. dotaz na počasí nebo zpracování obrazu). Odpověď má vždy na začátku textový tag s názvem emoce, který je poté oddělen a zpracován.

```
text_response = text_response.replace('\n', ' ').strip()

# Splitting the emotion from the response
prompt_words = text_response.split(maxsplit=1)

# Only keep the letters for the emotion and make them lowercase
prompt_words[0] = re.sub(r'[\[\]]', '',
                        prompt_words[0]).lower().strip()
```

² Chatbot odpovídá nejdříve jednoslovnou emoci, kterou by člověk měl při takové odpovědi, a až pak textem skutečné odpovědi.

Po každém použití se uchovává historie chatu, pokud je tato volba aktivována v konfiguraci. Třída využívá `atexit`, aby při ukončení programu automaticky spustila `save_memory()`, která uloží historii do souboru pomocí serializace `pickle`.

4.3.5 Detekce obličejů a rukou, sledování pohybů

Pro neverbální interakci robotické hlavy Elišky je zásadní schopnost vnímat a sledovat okolní objekty, zejména obličej uživatele a jeho ruce. Tyto funkce zajišťuje modul `visionHandler`, který zpracovává obrazový vstup z kamery a poskytuje informace o poloze sledovaných objektů. Třída zahrnuje detekci obličejů pomocí klasifikátoru Haar Cascade a detekci rukou přes knihovnu `cvzone`.

Po inicializaci a vytvoření okna se snímky z kamery začíná **detekce obličejů** nebo rukou. Pro detekci obličejů je použit klasifikátor Haar Cascade z knihovny OpenCV. Každý snímek je převeden do odstínů šedi a následně se na něm vyhledávají obličeje.

```
face_img = cv2.cvtColor(self.img.copy(), cv2.COLOR_BGR2GRAY)
faces_rect = self.haar_cascade.detectMultiScale(face_img,
                                                scaleFactor=1.1, minNeighbors=10)

# Check if any faces are detected
if len(faces_rect) > 0:

    # Update the track position
    track_position = (faces_rect[0][0], faces_rect[0][1],
                    faces_rect[0][2], faces_rect[0][3])
```

Pokud je obličej detekován, funkce `face_tracking()` vrátí jeho souřadnice ve formátu (x, y, w, h). Tyto údaje slouží k následnému pohybu očí a hlavy robota směrem k uživateli.

Detekce rukou je realizována pomocí třídy `HandDetector` z knihovny `cvzone`. Metoda `hand_tracking()` vrací souřadnice jedné nebo dvou detekovaných rukou v závislosti na zvoleném režimu (left hand, right hand, hands). Ruka je navíc klasifikována podle strany, což umožňuje např. sledovat jen levou ruku.

```
hands, _ = self.hand_detector.findHands(hand_img, draw=False,
                                       flipType=self.MIRROR_HANDS)
```

Pokud je detekována správná ruka, vrací funkce její pozici. Navíc je možné pomocí metody `get_fingers_up()` určit počet zdvižených prstů, což může někdy robotovi přijít vhod (například dokáže sčítat na prstech).

Třída podporuje více **sledovacích režimů**, které lze přepnout metodou `set_mode()`

- `face` – výchozí režim, sleduje obličej uživatele,
- `hands` – sleduje první nalezenou ruku,
- `left hand / right hand` – sleduje konkrétní ruku.

Podle zvoleného režimu metoda `face_hand_tracking()` automaticky vybere odpovídající detekční metodu a vrací aktuální pozici sledovaného objektu, za kterým se má robotická hlava Elišky otáčet a který má sledovat očima.

Získané pozice lze převést na relativní vynormované souřadnice pomocí metody `calculate_relative_coords()`, což umožňuje normalizované řízení motorů.

```
x, y, w, h = track_position
x = x / (self.img.shape[1]//2) - 1
```

```

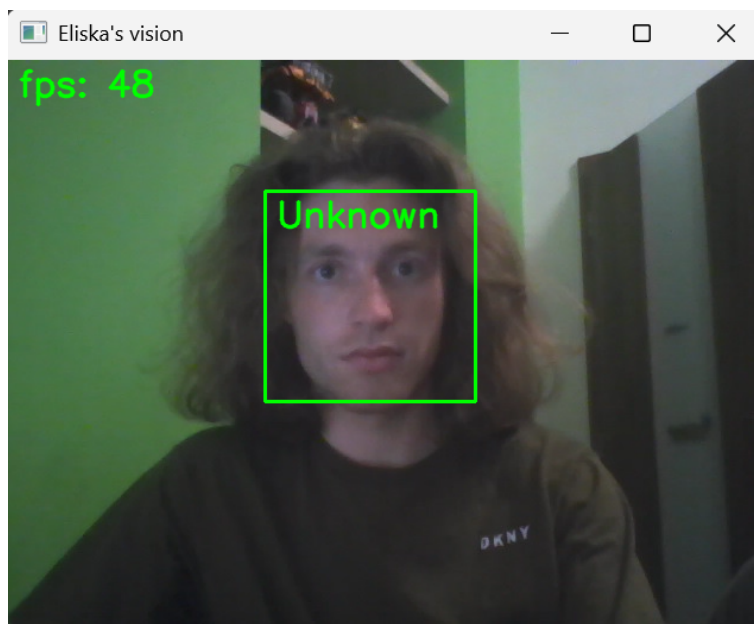
y = y / (self.img.shape[0]//2) - 1
w /= self.img.shape[1]
h /= self.img.shape[0]
track_position = (x, y, w, h)

# Make sure the tracking position is within the bounds
track_position = tuple(max(-1.0, min(1.0, i)) for i in track_position)

```

Tyto souřadnice se používají na přepočítání úhlů motorů očí a krku, aby se robot díval na detekované obličeje nebo ruce. Pokud detekovaný objekt není daleko od aktuálního pohledu očí, Eliška pohne pouze jimi. V opačném případě ale zapojí i krk.

Zobrazování výsledků probíhá v centrovaném okně, které aktualizuje metody `visualize()` a `create_centered_window()`. Jak to může vypadat, je zobrazeno na obrázku 4.6



Obrázek 4.6. Okno zachycující vidění Elišky a detekci obličeje

4.3.6 Využití ElevenLabs pro generování hlasu

Pro převod textových odpovědí robotické hlavy Elišky na přirozeně znějící řeč je využita služba ElevenLabs, která poskytuje realistickou syntézu hlasu včetně podpory emocí a vícejazyčnosti. Modul pro generování hlasu je implementován ve třídě `speakHandler`, která kromě samotného převodu řeči také synchronizuje pohyb čelisti a mimiku obličeje podle mluveného výstupu.

Při inicializaci třídy se nastaví výstupní reproduktor. Hlavní metodou pro převod textu na řeč je `say(text, emotion)`, která využívá oficiální knihovnu `elevenlabs`. Pomocí API klíče z konfigurace je při inicializaci navázáno spojení se serverem a požadovaný text je převeden do zvukové stopy.

```

audio = self.client.generate(
    text=text,
    voice="pMsXgVXv3BLzUgSXRp1E",
    model="eleven_multilingual_v2"
)

```

Pokud připojení nebo generování selže (z důvodů překročení limitů nebo jiných potíží), systém automaticky přejde na záložní řešení využívající knihovnu gTTS (Google Text-to-Speech) od Googlu.

Pro věrnější prezentaci mluveného projevu je **pohyb dolní čelisti synchronizován s hlasitostí výstupu**. To je realizováno pomocí tzv. loopback mikrofону, který snímá výstupní zvuk přímo z reproduktoru (nikoli z okolí). Hlasitost signálu je v reálném čase zpracována ve vlákne `sync_mouth()`, které průběžně volá funkci `sync_jaw()` s aktuálními daty.

```
def sync_mouth(self) -> None:
    with self.loopback_mic.recorder(samplerate=44100,
                                    blocksize=1024) as mic:
        while not self.stop_talking:
            data = mic.record(numframes=1024)
            self.sync_jaw(data)
            time.sleep(0.05)
        self.stop_talking = False
```

Synchronizace běží souběžně s přehráváním řeči ve vlákne, které se ukončí po skončení výstupu.

Součástí mluvení je také **řízení mimiky** robota. Při zahájení řeči se obočí a výraz tváře nastaví podle zjištěné emoce `move_to_emotion()`, a po skončení se navrátí do výchozího neutrálního stavu pomocí metody `move_to_neutral()`.

4.3.7 Další funkce Elišky

Kromě vedení běžného rozhovoru má Eliška k dispozici také několik podpůrných funkcí, které jí umožňují odpovídat na konkrétní dotazy uživatelů, poskytovat aktuální informace nebo přizpůsobit své chování podle uložených znalostí. Tyto funkce je možné propojit s jazykovým modelem prostřednictvím tzv. function calling. Tyto funkce jsou zároveň i ideálním místem pro customizaci Elišky a dodání jí nových možností.

- **Informace o počasí** – pomocí API služby OpenWeatherMap je možné získat aktuální počasí pro danou lokalitu. Funkce `get_weather_data()` se připojuje k serveru přes HTTP požadavek, zpracovává odpověď ve formátu JSON a vytváří pro Elišku výstup obsahující popis počasí, teplotu a rychlost větru.

```
params = {
    "id": CITY_ID,
    "units": "metric",
    "appid": WEATHER_API_KEY
}

# Get the response from the API
response = requests.get(WEATHERAPI_URL, params=params)
```

- **Aktuální čas a datum** – stejným způsobem jsou zjišťovány i informace o aktuálním čase a datu. Chatboti mohou mít se zjišťováním těchto druhů informací problémy, a tak je pro tyto potřeby implementována specifická funkce `get_weather_data()`.
- **Informace o škole** – funkce `retrieve_school_information()` načítá textový soubor `school_information.txt`, ve kterém lze uchovávat faktické informace o Fakultě elektrotechnické ČVUT. To umožňuje odpovědi na otázky typu „Co se na této fakultě

studuje?“ nebo „Jak se přihlásím ke studiu?“. Tyto informace se dají kdykoli upravit a aktualizovat podle aktuální potřeby prezentace školy.

- **Informace o uživateli** – podobně je připravena i další funkce, která rovněž bere informace z textového souboru – `retrieve_personal_information()`, která vrací individuální informace o uživateli (např. jméno, zájmy, oblíbené barvy apod.).

Funkce	Výstup	Účel
<code>get_weather_data()</code>	Popis počasí, teplota, rychlost větru	Poskytnutí aktuálního počasí pro konkrétní lokaci
<code>get_time_and_date()</code>	Aktuální čas a datum	Reakce na dotazy typu „Kolik je hodin?“ nebo „Jaký je dnes den?“
<code>retrieve_school_information()</code>	Obsah souboru <code>school_information.txt</code>	Odpovědi na otázky týkající se fakulty nebo studijního programu
<code>retrieve_personal_information()</code>	Obsah souboru <code>personal_information.txt</code>	Poskytnutí personalizovaných informací o uživateli

Tabulka 4.1. Tabulka s přehledem doplňkových funkcí Elišky a jejich účelu.

4.3.8 Vlákna pro vizualizaci a řízení pohybu očí

Pro správnou funkci sledování obličeje, zobrazování kamery a řízení pohybu očí běží v systému dvě samostatná vlákna, která spravuje třída `threadHandler`. Obě smyčky obsahují malý časový interval (`sleep(0.01)`), který snižuje zatížení procesoru a zabraňuje zahlcení systému.

- **Vizualizační vlákno** – metoda `vision_handler()` pravidelně zobrazuje snímky z kamery v okně uprostřed obrazovky. Do obrazu vkládá i grafické prvky, jako jsou ohraničení obličeje nebo ruky, případně další popisky.

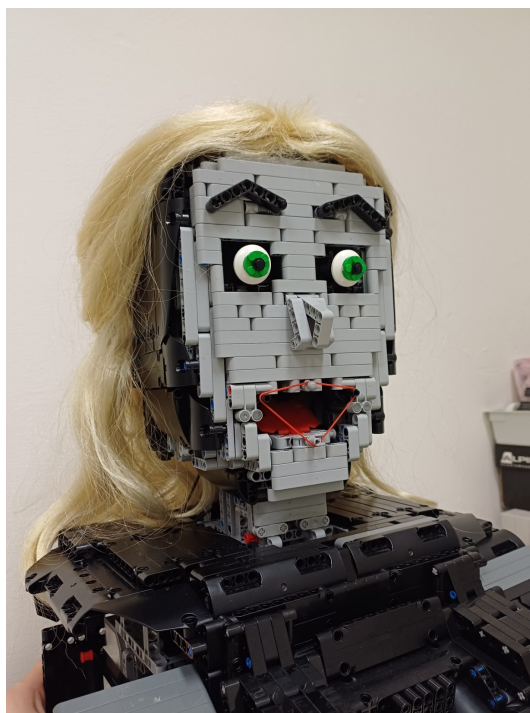
```
def vision_handler(self) -> None:
    self.create_centered_window()

    while not self.end_thread:
        self.visualize()
        sleep(0.01)
```

- **Sledovací vlákno** – vlákno `tracking_handler()` sleduje polohu uživatele (obličeje nebo rukou) a předává relativní souřadnice motorům očí. Pokud není uživatel detekován, dochází občas k náhodnému rozhlížení, které působí přirozeněji. Pro tyto občasné náhodné pohledy slouží funkce `set_random_tracking()`.

Obě vlákna se spouští metodou `start_threads()` a automaticky se ukončují při opuštění kontextu `with threadHandler(...)`. To zajišťuje stabilní a bezpečný běh celého vizuálního systému.

4.4 Jak Eliška vypadá



Obrázek 4.7. Obrázek zachycující obličej Elišky při úsměvu.



Obrázek 4.8. Obrázek celé busty robota Elišky

Kapitola 5

Testování a experimenty

Dokumentace a zdrojový kód je k dispozici na githubu¹.

5.1 Funkčnost pohybových mechanismů

Při testování různých pohybů obličejů a krku se objevilo několik výzev. Pro některé motory bylo nutné najít jejich limity, aby se nestalo, že Eliška nezačne například otáčet obočím dokola. Jak již bylo zmíněno v sekci 4.1.3, problémem pusu bylo najít kompromis mezi rychlostí otevírání pusu (a tedy i silou motoru) a fyzickou schopností konstrukce a kostek LEGO držet pohromadě. Největší problém ale působily koutky, neboť obsahují velké množství převodů, a to způsobuje zpomalení odezvy a možná protáčení. Úpravou a opravou těchto problémů byl stráven zhruba jeden týden. Kromě těchto zmíněných nedostatků fungují veškeré pohybové mechanismy přesně a bez problémů.

5.2 Odezva chatbotu a hlasového výstupu

Tato část systému představuje pravděpodobně nejslabší článek celého řešení. Přestože funguje spolehlivě, zpracování vstupu a generování odpovědi zabírá nejvíce času v celém procesu interakce s uživatelem. Klíčovým faktorem je zde použitý model pro rozpoznávání řeči – Whisper. Volbou jednoduššího modelu lze zkrátit dobu odezvy, ale za cenu mírně nižší přesnosti rozpoznání textu. V praxi to však nepředstavuje zásadní problém, protože chatbot je schopen drobné „překlepy“ interpretovat a správně porozumět smyslu věty. Kvalita odpovědi tedy většinou zůstává zachována, a to především díky pokročilému jazykovému modelu Gemini. Výsledkem mohou být jen mírně nepřesné odpovědi, případně doplňující otázka robota pro upřesnění záměru uživatele. Při komunikaci v angličtině se tyto problémy prakticky nevyskytují, což je dáno podstatně větším množstvím trénovacích dat oproti češtině, která je v tomto ohledu omezenější. Tedy ani snížení přesnosti proces dostatečně neurychlí, kvůli omezení v podobě češtiny.

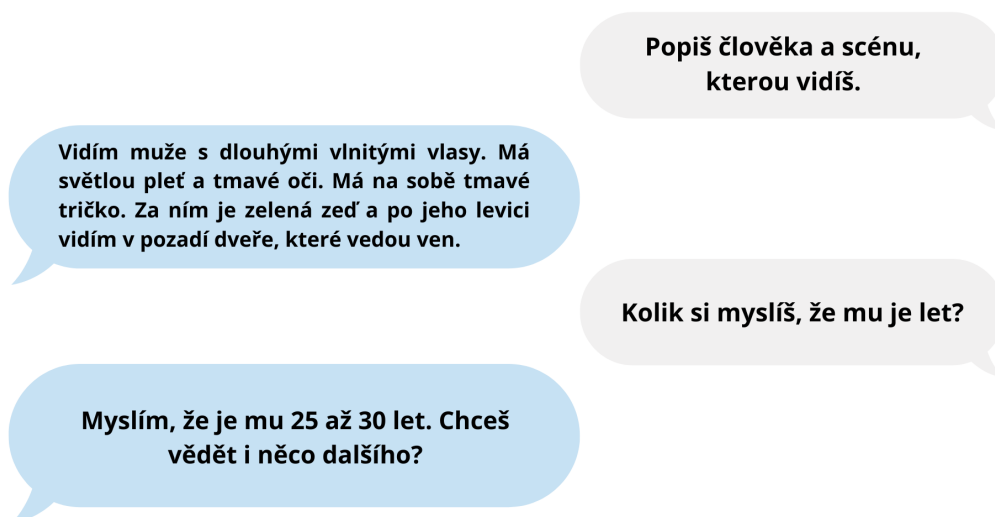
Generování hlasového výstupu probíhá bezproblémově – za předpokladu, že jsou dostupné kredity u služby ElevenLabs. Pokud nejsou, je hlas Elišky automaticky za chodu nahrazen generickým hlasem od Googlu, který sice není tak realistický, ale umožňuje pokračovat v komunikaci bez přerušování.

5.3 Počítačové vidění

Rozpoznávání objektů, zejména obličejů a rukou uživatele, je jednou z nejspolehlivějších částí celé práce. Eliška dokáže díky knihovně `OpenCV` bezpečně sledovat obličej, a to i pokud se hýbou, nebo nejsou zcela přímo natočeny směrem k ní. To samé se dá říci i o rukách.

¹ https://github.com/MarekFeik/Fel_Eliska

Zbytek počítačového vidění obstarává umělá inteligence Gemini AI od společnosti Google. Tento chatbot byl trénován multimodálně (na obrázcích a textu zároveň), a tak zvládá hledat souvislosti mezi zadanou otázkou a dodaným obrázkem z kamery s vysokou přesností. Mnohokrát si model všimne detailů, které ani sám uživatel nevnímá, nebo dokáže překvapit svou přesností a chápáním. Dobrým příkladem těchto jevů může být odhadování věku uživatele nebo popis okolního prostředí. Na ukázkou je na obrázku 5.1 vidět konverzace s Eliškou po dotázání na popis uživatele z obrázku 4.6 na straně 36.



Obrázek 5.1. Ukáзка konverzace s Eliškou, popis scény a uživatele.

5.4 Uživatelské testování

Z testování různými uživateli vzešla pozitivní zpětná vazba na to, jak je Eliška při konverzování milá a nápomocná. Jedinou nevýhodou při konverzaci může být hlučné okolní prostředí, tomu se ale velmi špatně zabraňuje. V hlučném prostředí je náročné rozlišit hlas od okolního šumu.

Dalším podnětem bylo sledování očima. Spoustě lidem to přijde jako velmi zajímavá součást projektu, ale zároveň to na někoho působí až děsivě. S tím může souviset i to, jak Eliška vypadá. Kostky LEGO dovolují pouze určitou úroveň lidskosti, která může být na hraně dříve zmíněného fenoménu „Uncanny valley“. Nápravou by mohla být snaha vytvořit Eliščin obličej a výraz veselejší a schválně umělý, odlišitelný od lidského.

S tématem vizáže přišla i velmi pozitivní zpětná vazba na vlasy Elišky. Při kontrastu s kostkami LEGO vypadají velmi dobře a dodávají jí osobitost.

Kapitola 6

Závěr

Cílem této práce bylo navrhnout a realizovat interaktivní robotickou hlavu z LEGO kostek, která dokáže reagovat s uživatelem přirozeným způsobem – vést s ním konverzaci a vnímat informace z okolí. Výsledkem je funkční busta robota jménem Eliška, která kombinuje prvky mechanického designu, řízení LEGO motorů a kostky EV3 přes Bluetooth, programování v jazyce Python a integraci pokročilých AI technologií pro rozpoznávání hlasu, generování odpovědí a syntézy přirozeného hlasu, jako jsou Whisper, Gemini nebo ElevenLabs.

Práce ukazuje, že i z jednoduchých a přístupných komponent a technologií, jako je LEGO, lze vytvořit systém, který nejen komunikuje, ale i zachycuje informace z okolního světa a dokáže na ně reagovat a pracovat s nimi. Největším přínosem projektu je spojení robotiky s moderními AI službami, což dává za vznik platformě vhodné nejen pro demonstraci možností umělé inteligence, ale i jako propagační nástroj pro fakultu. Klíčové součásti projektu, jako jsou interaktivita, schopnost sledovat uživatele nebo reagovat na otázky přirozeným hlasem, přispívají k zábavnější a osobitější popularizaci technologií a fakulty.

Vývoj robota probíhal iterativně. Nejprve vznikl návrh základního mechanismu inspirovaný existující LEGO hlavou, který byl následně upravován a přizpůsobován potřebám konkrétních funkcí. Paralelně s konstrukcí probíhal i vývoj softwarové části – především testování komunikace s EV3 přes Bluetooth, zajištění stability ovládání jednotlivých motorů, napojení na cloudové a AI služby a úprava nejrůznějších funkcí a možností kódu. Největší výzvou bylo sladění všech částí (jak softwarových, tak hardwarových) tak, aby výsledná odezva robota byla dostatečně rychlá a přirozená. Zásadní roli v tom hrálo ladění rozpoznávání řeči a převod mluveného slova na text. Celý vývoj tak vyžadoval kombinaci technických znalostí, kreativního přístupu i průběžného experimentování a testování.

Projekt nabízí dlouhou řadu možností budoucího rozšíření. Jednou z nich je plná a rychlejší obousměrná komunikace v češtině bez potřeby cloudových služeb, případně vylepšení schopností robota sledovat a interpretovat neverbální komunikaci nebo zapamatování si uživatelů a jejich obličejů. Možné je také rozšíření o další výrazy obličejů, gesta nebo schopnost samostatného pohybu v podobě dostavění zbytku těla a jeho implementace do Eliščina uvažování. V neposlední řadě lze systém využít jako základ pro pokročilejší humanoidní rozhraní nebo jako součást větších interaktivních instalací.

Tato práce ukázala, že i kombinací jednoduchých prostředků lze vytvořit poutavý a funkční systém, který propojuje fyzický svět s tím digitálním – a přibližuje tak techniku lidem.

Literatura

- [1] MINDSTORMS, Creative. *World's Best AI LEGO Robot!* [online]. Dostupné na <https://www.youtube.com/watch?v=MghrMAWsf18&t>.
- [2] GROUP, LEGO. *LEGO MINDSTORMS EV3* [online]. Dostupné na <https://www.lego.com/cs-cz/themes/mindstorms/ev3>.
- [3] "PHILO" HURBAIN, Philippe. *LEGO® 9V Technic Motors compared characteristics* [online]. Dostupné na <https://www.philohome.com/motors/motorcomp.htm>.
- [4] GROUP, LEGO. *LEGO MINDSTORMS EV3 Home Edition Software* [online]. Dostupné na <https://www.lego.com/en-us/themes/mindstorms/downloads>.
- [5] MAJORGEEKS. *LEGO Mindstorms EV3 1.4.5* [online]. Dostupné na https://www.majorgeeks.com/files/details/lego_mindstorms_ev3.html.
- [6] TEAM, Pybricks. *Pybricks – Python coding for smart LEGO hubs*. Dostupné na <https://pybricks.com/>. online.
- [7] , Wikipedia contributors. *Lego Mindstorms EV3 — Wikipedia, The Free Encyclopedia* [online]. Dostupné na https://en.wikipedia.org/w/index.php?title=Lego_Mindstorms_EV3&oldid=1274272978. Online; accessed 16-May-2025.
- [8] IET. *LEGO Robot breaks the Rubik's Cube World Record* [online]. Dostupné na <https://www.youtube.com/watch?v=c05DLbpp3-M>.
- [9] VASWANI, Ashish, Noam SHAZEER, Niki PARMAR, Jakob USZKOREIT, Llion JONES, Aidan N. GOMEZ, Lukasz KAISER a Illia POLOSUKHIN. Attention Is All You Need. *CoRR*. 2017, ročník abs/1706.03762. Dostupné na <http://arxiv.org/abs/1706.03762>.
- [10] (3BLUE1BROWN), Grant Sanderson. *Large Language Models explained briefly* [YouTube [online]]. Dostupné na <https://www.youtube.com/watch?v=LPZh9B0jkQs>.
- [11] LEWIS, Patrick, Ethan PEREZ, Aleksandra PIKTUS, Fabio PETRONI, Vladimir KARPUKHIN, Naman GOYAL, Heinrich KUTTLER, Mike LEWIS, Wen-tau YIH, Tim ROCKTASCHEL, Sebastian RIEDEL a Douwe KIELA. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *CoRR*. 2020, ročník abs/2005.11401. Dostupné na <https://arxiv.org/abs/2005.11401>.
- [12] OPENAI. *Text generation and prompting* [online]. Dostupné na <https://platform.openai.com/docs/guides/text?api-mode=responses>.
- [13] DEEPMIND. *Gemini 1 Report* [online]. Dostupné na https://storage.googleapis.com/deepmind-media/gemini/gemini_1_report.pdf.
- [14] ANTHROPIC. *Build with Claude* [online]. Dostupné na <https://docs.anthropic.com/en/home>.
- [15] GITHUB. *GitHub Copilot · Your AI pair programmer* [online]. Dostupné na <https://github.com/features/copilot>.

- [16] META-LLAMA. *GitHub - meta-llama/llama3: The official Meta Llama 3 GitHub site* [online]. Dostupné na <https://github.com/meta-llama/llama3>.
- [17] , Wikipedia contributors. *Virtual assistant — Wikipedia, The Free Encyclopedia* [Online]. Dostupné na https://en.wikipedia.org/w/index.php?title=Virtual_assistant&oldid=1287147412.
- [18] DEEPMIND, Google. *Gemini 1 Technical Report* [online]. Dostupné na https://storage.googleapis.com/deepmind-media/gemini/gemini_1_report.pdf.
- [19] INCUBITY. *Gemini LLM: Deep Dive with Hands-on Implementations* [online]. Dostupné na <https://incubity.ambilio.com/gemini-llm-deep-dive-with-hands-on-implementations/>.
- [20] FEI-FEI LI, Serena Yeung, Justin Johnson. *CS231n: Convolutional Neural Networks for Visual Recognition* [online]. Dostupné na <https://cs231n.github.io/convolutional-networks/>.
- [21] MEDIUM. *Google to Offer Tensorflow Training at AI Frontiers Conference* [online]. Dostupné na <https://medium.com/aifrontiers/google-to-offer-tensorflow-training-at-ai-frontiers-conference-b46fceb54189>.
- [22] KRIZHEVSKY, Alex, Ilya SUTSKEVER a Geoffrey E. HINTON. ImageNet Classification with Deep Convolutional Neural Networks. In: *Advances in Neural Information Processing Systems*. 2012. s. 1097–1105. Dostupné na https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.
- [23] , OpenCV team. *OpenCV: Open Source Computer Vision Library* [online]. Dostupné na <https://opencv.org/>.
- [24] VIOLA, Paul a Michael JONES. Rapid Object Detection using a Boosted Cascade of Simple Features. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2001. s. 511–518.
- [25] HASSAN, Murtaza. *Hand Tracking Module – CVZone* [online]. Dostupné na <https://github.com/cvzone/cvzone>.
- [26] RESEARCH, Google. *MediaPipe Hands: Real-time Hand Tracking* [online].
- [27] RADFORD, Alec, Jong Wook KIM, Tao XU, Greg BROCKMAN, Christine MCLEAVEY a Ilya SUTSKEVER. Robust Speech Recognition via Large-Scale Weak Supervision. *arXiv preprint arXiv:2212.04356* [online]. 2022. Dostupné na <https://arxiv.org/abs/2212.04356>.
- [28] OPENAI. *GitHub - openai/whisper: Robust Speech Recognition via Large-Scale Weak Supervision* [online]. Dostupné na <https://github.com/openai/whisper?tab=readme-ov-file>.
- [29] RADFORD, Alec, Jong Wook KIM, Tao XU, Greg BROCKMAN, Christine MCLEAVEY a Ilya SUTSKEVER. Robust Speech Recognition via Large-Scale Weak Supervision. In: Andreas KRAUSE, Emma BRUNSKILL, Kyunghyun CHO, Barbara ENGELHARDT, Sivan SABATO a Jonathan SCARLETT, editoři. *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 2023. s. 28492–28518. Proceedings of Machine Learning Research. Dostupné na <https://proceedings.mlr.press/v202/radford23a.html>.
- [30] ELEVEN LABS INC. *ElevenLabs - AI Voice Generation Platform* [online]. Dostupné na <https://www.elevenlabs.io>.

-
- [31] TAYLOR, Paul. Text-to-Speech Synthesis. *Cambridge University Press* [online]. 2009, s. 245–247. Dostupné na <https://www.cambridge.org/core/books/texttospeech-synthesis/D2C567CEF939C7D15B2F1232992C7836>.
- [32] MORI, Masahiro. The Uncanny Valley. *Energy* [online]. 1970, ročník 7, č. 4, s. 33–35. Dostupné na <https://spectrum.ieee.org/the-uncanny-valley>. Přeložil Karl F. MacDorman a Norri Kageki.
- [33] CASSIDYJAMES. *The “Uncanny Valley” Curve* [online]. Dostupné na <https://cassidyjames.com/blog/uncanny-valley-curve/>.
- [34] ROBOSOUTĚŽ, Fakulta elektrotechnická ČVUT. *Stavebnice Brian* [online]. Dostupné na <https://robosoutez.fel.cvut.cz/stavebnice-brian>.
- [35] ev3dev developers. *ev3dev — Linux-based operating system for LEGO MIND-STORMS EV3* [online]. Dostupné na <https://www.ev3dev.org/>.
- [36] WISEMAN, John. *py-webrtcvad: Python interface to the WebRTC Voice Activity Detector* [online]. Dostupné na <https://github.com/wiseman/py-webrtcvad>.