



Pedagogická  
fakulta  
Faculty  
of Education

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

**Jihočeská univerzita v Českých Budějovicích**

Pedagogická fakulta  
Katedra informatiky

Bakalářská práce

# Rozvíjení algoritmického myšlení pomocí robotů Dash a Dot

Vypracoval: Štěpán Zahradka  
Vedoucí práce: Mgr. Jan Pršala

České Budějovice 2024

## **Prohlášení**

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

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

V Českých Budějovicích dne 13. dubna 2024.

.....  
Štěpán Zahrádka

# JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta  
Akademický rok: 2022/2023

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Štěpán ZAHŘÁDKA  
Osobní číslo: P21392  
Studijní program: B0114A300110 Oborové studium se zaměřením na vzdělávání na 2. stupni základní školy  
Specializace: Anglický jazyk se zaměřením na vzdělávání na 2. stupni ZŠ  
Informační technologie se zaměřením na vzdělávání na 2. stupni ZŠ  
Téma práce: Rozvíjení algoritmického myšlení pomocí robotů Dash a Dot  
Zadávací katedra: Katedra informatiky

### Zásady pro vypracování

Bakalářská práce se bude zabývat tvorbou sady úloh pro roboty Dash a Dot zaměřené na rozvíjení algoritmického myšlení žáků 2. stupně ZŠ. V práci budou nejdříve popsáni roboti Dash a Dot a jejich vliv na rozvoj algoritmického a informatického myšlení, bude zde podrobně rozebráno vybrané blokové prostředí a důvod, proč bylo vybrané, a proč je vhodné pro programování robotů Dash a Dot. Dále budou prezentovány vytvořené úlohy. Ty budou koncipovány tak, aby k jejich splnění bylo za potřebí pouze dvojice robotů Dash a Dot, případně předměty, které se často vyskytují ve školní třídě. Úlohy budou programovacího charakteru a budou využívat vybrané blokové prostředí. Úlohy budou rozděleny do kapitol podle toho, čím se zabývají. Například se může jednat o úlohy zaměřené na komunikaci mezi roboty pomocí jejich čidel. Sada úloh bude následně odzkoušena během výuky.

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

#### Seznam doporučené literatury:

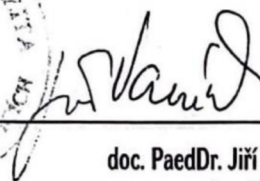
1. PAVLÍČEK, Antonín, Alexander GALBA a Michal HORA. Moderní informatika. Druhé, rozšířené vydání. [Praha]: Professional Publishing, 2017. ISBN 978-80-906594-6-9.
2. SARMA, Kanya. Dash and Dot. Ann Arbor, United States: Cherry Lake Pub, 2017. ISBN 1634727193.
3. DENNING, Peter J. a Matti TEDRE. Computational thinking. Cambridge, Massachusetts: The MIT Press, 2019. MIT Press essential knowledge series. ISBN 9780262536561.
4. Co je informatické myšlení? Informatické myšlení [online]. Jihočeská univerzita v Českých Budějovicích, 2018 [cit. 2023-04-18]. Dostupné z: <https://www.imysleni.cz/informaticke-mysleni/co-je-informaticke-mysleni>
5. Dash. Wonder Workshop [online]. 2019 [cit. 2023-04-18]. Dostupné z: <https://www.makewonder.com/getting-started/dash/>

Vedoucí bakalářské práce: Mgr. Jan Pršala  
Katedra informatiky

Datum zadání bakalářské práce: 19. dubna 2023  
Termín odevzdání bakalářské práce: 30. dubna 2024



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



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

V Českých Budějovicích dne 19. dubna 2023



## **Anotace**

Bakalářská práce se zabývá tvorbou sady úloh pro dvojici robotů Dash a Dot. Úlohy jsou vytvořeny za účelem rozvíjení algoritmického myšlení. Úlohy jsou programovacího charakteru a využívají blokově orientovaného prostředí Blockly.

V úvodu mé práce jsou popsány cíle a metody mé práce.

V teoretické části jsou rozebrány výukové cíle z pohledu informatiky, informatika na základních školách a nakonec informatické a algoritmické myšlení. Obě tato myšlení jsou následně porovnána. Dále jsou zde popsáni oba roboti spolu s blokovým prostředím. Nakonec jsou rozebrány postoje žáků k výuce robotiky a vliv robotických hraček na algoritmické myšlení.

V praktické části mé práce jsou vytvořené úlohy. Celkem se jedná o 14 úloh, které jsou následně podle podobnosti rozděleny do 3 částí. U každé úlohy jsou popsány její výukové cíle spolu s komentářem pro učitele. Každá úloha obsahuje také vzorové řešení.

## **Klíčová slova**

Dash a Dot, algoritmické myšlení, informatické myšlení, sada úloh, robotika

## **Abstract**

The bachelor thesis focuses on creating a set of exercises for the pair of robots Dash and Dot. The exercises are designed to develop algorithmic thinking. They are programming-oriented and utilize the block-based coding environment Blockly.

The introduction of my work describes the goals and methods of my work. In the theoretical part, educational objectives are discussed from the perspective of computer science, computer science in primary schools, and finally computational and algorithmic thinking. Both of these types of thinking are then compared. Additionally, both robots are described along with the block environment. Finally, student attitudes towards robotics education and the influence of robotic toys on algorithmic thinking are examined.

In the practical part of my work, tasks are created. There are a total of 14 tasks, which are subsequently divided into 3 parts based on similarity. With each exercise are described its educational objectives along with a commentary for teachers. Each exercise also includes a sample solution.

## **Keywords**

Dash and Dot, algorithmic thinking, computational thinking, task set, robotics

# Obsah

1	Úvod.....	9
1.1	Cíle práce .....	9
1.2	Metody práce.....	9
2	Výukové cíle .....	11
2.1	Obecné výukové cíle .....	11
2.2	Informatika na základních školách.....	12
2.3	Informatické myšlení .....	13
2.4	Algoritmické myšlení .....	15
3	Dash & Dot .....	18
3.1	Popis robotů .....	18
3.2	Blokové prostředí blockly .....	18
3.3	Nedokonalosti blokového prostředí .....	22
4	Postoje žáků k výuce robotiky .....	24
5	Vliv robotických hraček na algoritmické myšlení .....	26
6	Sada úloh.....	28
6.1	Pohybové úlohy.....	28
6.1.1	Začínáme programovat.....	29
6.1.2	První krůčky .....	30
6.1.3	Čtverec .....	32
6.1.4	Pozor! .....	34
6.1.5	Hra na schovávanou .....	36
6.1.6	Slalom .....	39
6.2	Spolupráce robotů .....	42
6.2.1	Reagujeme na Dota .....	43

6.2.2	Kapitán Dot .....	44
6.2.3	Zkušený trenér Dot.....	46
6.2.4	Předváděčka .....	48
6.2.5	Dot jako dirigent .....	50
6.3	Rozšíření .....	52
6.3.1	První výstřely .....	52
6.3.2	Střelba na cíl.....	54
6.3.3	Náhodný katapult .....	56
7	Testování úloh.....	58
7.1	Úvod.....	58
7.2	Průběh .....	58
7.3	Provedené změny .....	60
8	Závěr .....	61
9	Seznam použité literatury.....	62
10	Seznam obrázků .....	64

# 1 Úvod

Dash a Dot je dvojice velmi zajímavých interaktivních robotů, kteří se hodí nejen do výuky ale i na nejrůznější domácí procvičování. Toto byl jeden z hlavních důvodů, proč jsem se rozhodl s nimi ve své bakalářské práci pracovat a využít je pro zpestření každodenní výuky informatiky na školách. Samotní roboti se prodávají v sadě s několika rozšířeními čili se s nimi dá tvořit, učit se a v neposlední řadě procvičovat informatické potažmo algoritmické myšlení. Co se týče programování těchto robotů, tak nejčastější způsob je využití blokového programu s názvem Blockly. Kvůli jeho jednoduchosti a možnosti tvořit sofistikované úlohy vhodné pro žáky 2. stupně základní školy jsem se rozhodl v tomto programu pracovat. Dalším důvodem byla také podobnost s prostředím scratch, které se v dnešní době na základních školách k výuce programování používá velmi často.

## 1.1 Cíle práce

Hlavním cílem mé bakalářské práce je vytvořit sadu úloh pro rozvíjení algoritmického a informatického myšlení. Úlohy budou tvořeny pro žáky 2. stupně základních škol a budou programovatelné pomocí blokového prostředí Blockly. Další neméně důležitý cíl bude seznámení čtenářů se samotnými roboty a blokovým prostředím. Tímto bych také mohl docílit častějšího využití těchto robotů ve výuce a možná i robotických hraček obecně. Dalším cílem mé práce bude zpracovat text popisující historii informatického myšlení, jeho popis a následné porovnání s myšlením algoritmickým.

## 1.2 Metody práce

V teoretické části mé práce budu čerpat z ověřených zdrojů a s jejich pomocí vytvořím text, který bude podrobně popisovat oba roboty. Dále rozeberu vybrané blokové prostředí, kde se zaměřím také na jeho výhody a nevýhody, které objevím během své práce. Například, pokud nebudu moci provést některou úlohu z důvodu nedostatku instrukcí a podobně. Rozeberu také výukové cíle, podle kterých se následně budu snažit sestavit co možná nejlepší sadu úloh. Pokusím se své úlohy opřít i o znalosti z didaktiky informatiky, čili jak obecně vyučovat informatiku a jaké

vědomosti by si žáci na základní škole měli osvojovat. U blokového prostředí se zaměřím také na to, zda je vhodné pro mnou vybranou věkovou skupinu či nikoliv. Pokusím se také najít nějaký dotazník nebo studii jaký mají na využívání robotů ve vzdělávání názor samotní žáci a podpořit tím svou sadu úloh.

Co se týče praktické části, tak zde se pokusím práci rozdělit do několika systematických kroků, abych mohl vytvořit co nejkompexnější úlohy. Nejdříve si pečlivě prostuduji věkovou skupinu, pro kterou budou úlohy tvořeny, čímž se vyhnu tomu, že by byly úlohy příliš obtížné nebo naopak jednoduché. Naplánuji si obsah svých úloh tak, aby si žáci při vypracovávání rozvíjeli algoritmické myšlení postupně a pomalu. Také si předem definuji co konkrétně budu chtít žáky svými úlohami naučit. Zde přijde čas na tvorbu samotných úloh, čili úlohy pohybové, interaktivní úlohy s čidly a nakonec zaměřené na rozšíření obou robotů. Zaměřím se na to, aby byly úlohy správně strukturované, čili jasný začátek a konec. Dále aby byly dostatečně srozumitelné a názorné, ale na druhou stranu aby měli žáci dostatečnou volnost při tvorbě. Z tohoto důvodu budou některé úlohy obsahovat rozšíření, které bude vhodné například pro rychlejší skupiny žáků. Oproti základnímu řešení budou trochu obtížnější. U složitějších úloh také přidám tip, který bude fungovat jako jakási nápověda pro žáky. Ke konci mé práce věnuji čas také ladění a testování úloh na základě čeho se pokusím úlohy ještě upravit. U každé úlohy přidám srozumitelný popis pro učitele, kde bude vypsáno co je u dané úlohy důležité a jak by měli s žáky postupovat. K samotnému popisu přidám i konkrétní výukové cíle pro každou úlohu samostatně.

## 2 Výukové cíle

### 2.1 Obecné výukové cíle

Cíl vyučování je zamýšlený a očekávaný výsledek k němuž učitel v součinnosti se žáky směřuje. Výsledek je vyjádřen ve změnách, jichž se prostřednictvím vyučování dosahuje ve vědomostech, dovednostech a vlastnostech žáků. Existují cíle obecné jako je například cíl školy nebo cíl předmětu, dále existují cíle konkrétní jako jsou cíle samotné vyučovací hodiny nebo aktivity. [1]

Výukové cíle by měly být:

- komplexní – kognitivní, afektivní, psychomotorické
- konzistentní – neměly by se míchat cíle obecné a konkrétní
- kontrolovatelné – mělo by být jednoznačné jak jakožto učitel můžeme splnění cílů ověřit
- přiměřené úrovni žáků [1]

Aby se žáci dostali podle Bloomovy taxonomie cílů na úroveň osvojení, měli by projít všemi těmito procesy:

- Zapamatovat
- Porozumět
- Aplikovat
- Analyzovat
- Hodnotit
- Tvořit [1]

Z tohoto tvrzení je patrné, že se s žáky snažíme dosáhnout nejrůznějších výukových cílů. Vytvořená sada úloh se zabývá hlavně cíli konkrétními, a snaží se rozvíjet již získané vědomosti a dovednosti žáků. Nejdůležitějšími jsou tedy u sady úloh hlavně cíle kognitivní a psychomotorické. Sada úloh se ale dá zahrnout i do obecnějších cílů jako jsou například cíle předmětů nebo oblastí, čili v tomto případě informatika, rozvoj algoritmického myšlení a v neposlední řadě programování. Mnou vytvořené úlohy u žáků rozvíjí ve velké míře hlavně schopnosti porozumět,



aplikovat, analyzovat a tvořit. Tyto schopnosti jsou také součástí kognitivní oblasti podle bloomovy taxonomie cílů, ze které jsem se snažil ve své tvorbě vycházet.

## **2.2 Informatika na základních školách**

Informatika je ve společnosti často chápána jako výuka o počítačích, což je z mého pohledu velmi zjednodušené tvrzení. Samozřejmě, že hlavním důvodem k vzniku tohoto vědního oboru byl právě příchod nových technologií, ale to neznamená, že bychom se při výuce informatiky měli omezovat na pouhé vyučování praktických dovedností s počítačem. V dnešní době se tyto dovednosti řadí již spíše k základní lidské potřebě, protože se s technologiemi setkáváme v každodenním životě. [2]

Informatiku můžeme tedy chápat jako vědu o struktuře, počítačových jazycích, programování systémů pro zpracování dat. Dále se informatika zabývá například algoritmy, datovými strukturami a každodenními informačními procesy souvisejícími také s jevy v celé společnosti a přírodě. Na druhou stranu jsou v tomto odvětví stále důležité znalosti o zpracovávání, ukládání a následném přenosu či vyhodnocování dat. [3]

Díky těmto důležitým aspektům se předmět informatika stává stále více komplexním předmětem, který žáky připravuje na nejrůznější překážky během jejich studia ale také rozvíjí jejich kompetence v osobním životě. Dále toto velmi pozitivně přispělo k zvýšení počtu hodin informatiky v rámcovém vzdělávacím programu pro výuku informatiky na základních školách v roce 2021.

Žák druhého stupně základní školy by podle informací uvedených v rámcovém vzdělávacím programu z roku 2023 měl mít v oboru algoritmizace nejrůznější dovednosti. Co se týče očekávaných výstupů důležitých k mé bakalářské práci tak nejdůležitější je, že žák v odvětví algoritmizace ovládá právě tyto:

- Sestavuje a testuje symbolické zápisy postupů
- Popíše jednoduchý problém, navrhne a popíše jednotlivé kroky jeho řešení
- V blokově orientovaném programovacím jazyce sestaví program, rozpozná opakující se vzory, používá opakování a připravené podprogramy.

- Žák schopen ověřit správnost jím navrženého postupu či programu, najde a opraví v něm případnou chybu. [4]

Mnou vytvořené úlohy se z tohoto důvodu orientují právě na tyto oblasti a postupným průchodem všemi typy úloh by si žák měl tyto dovednosti dostatečně osvojit a procvičit. Úlohy jsou zadávány dostatečně obecně z toho důvodu, aby žáky nabádaly k tvořivosti a vytváření vlastních programů. Díky tomuto mají žáci větší volnost a zároveň to v nich vytváří pocit, že něco dokázali vytvořit, což pozitivně ovlivňuje jejich práci a obecně mají poté větší motivaci se naučit něco nového.

## 2.3 Informatické myšlení

Pokud se budeme bavit o informatickém myšlení, tak musíme nejdříve zdůraznit, že se jedná o poměrně nový pojem v oboru informatiky. První použití tohoto sousloví ve spojení s informatikou bylo v roce 1996 tehdejším matematikem a informatikem Seymourem Papertem, který přišel například i s programovacím jazykem LOGO. V tomto kontextu byl pojem využit spíše jako nějaká domněnka autora a dál nebyl nijak hlouběji popsán. [5]

O větší zájem o informatické myšlení se zasloužila až Jeanette Wing, která tehdy působila na Carnegie Mellon University. V roce 2006 vydala krátký článek, kde se pokusila stručně vymezit pojem informatické myšlení. V té době se jednalo spíše o vizi a informatické myšlení připodobnila k základním dovednostem jako je například čtení, psaní a počítání. Snažila se poukázat na to, že informatika se hodí profesionálům napříč obory, ale na druhou stranu také při řešení běžných každodenních problémů. Touto dobou bylo tedy informatické myšlení již definováno jako způsob přemýšlení, který si můžeme díky informatice osvojit. [6]

Tato autorka nakonec přišla s touto definicí:

*„ Informatické myšlení jsou myšlenkové postupy zapojené při takovém formulování problému a jejich řešení, které umožní tato řešení efektivně provést agentem zpracovávajícím informace “* [5]

Agentem může být například člověk, ale i stroj. Obecné informatické myšlení by podle Jeanette Wing mělo zahrnovat tyto schopnosti:

- Pochopit, které aspekty problému jsou řešitelné strojově

- Vyhodnotit shodu mezi inforatickými prostředky a problémem
- Porozumět možnostem a omezením inforatických prostředků
- Použít inforatické prostředky novým způsobem či v nové situaci
- Použít inforatické strategie v jakékoliv oblasti [5]

Z těchto informací je patrné, že inforatické myšlení je poměrně nový pojem. Tento pojem se také neustále vyvíjí a každý autor nebo vědec na něj nahlíží trochu odlišným způsobem. Každopádně ale můžeme říci, že se jedná o způsob myšlení využívaný jak při práci se stroji, tak v každodenním životě každého z nás.

Inforatické myšlení může být jednoduše definováno jako schopnost popsat, analyzovat a v neposlední řadě efektivně řešit předložený problém. Pro mnou vytvořenou sadu úloh je poměrně důležité to, že se inforatické myšlení snažíme rozvíjet postupně. Čili je důležité ho uplatňovat nejdříve na jednoduchých úlohách a postupně zvyšovat náročnost. Na druhou stranu pořád platí, že čím komplexnější daný problém je, tím užitečnější je následné využití nástrojů a přístupů inforatického myšlení. [7]

Pokud se chceme zaměřovat na rozvíjení inforatického myšlení, potažmo konkrétně algoritmickeho, tak je důležité žáky povzbuzovat k tomu, aby vymýšleli nová, vlastní řešení. Z tohoto důvodu jsem se rozhodl pojmut zadání úloh velmi obecně, čímž se snažím vytvořit vhodné prostředí pro žáky. Dále je také důležité si uvědomit, že chyba je přirozenou součástí procesu učení. Žáky bychom tedy měli učit s chybou pracovat. [8]

Jedním z cílů inforatického myšlení je naučit žáky důvěřovat vlastním schopnostem a úvaze. Jedná se o kreativní záležitost, kde je důležité zakomponovat schopnost přemýšlet nad problémem a kombinovat známé postupy s nově vymyšlenými nebo předanými učitelem. Jádrem výuky proto není vysvětlování učitele, ale samotná práce žáků. [8]

Dále je důležité u žáků rozvíjet schopnost spolupráce v týmu. Úlohy jsou proto vytvořeny pro alespoň 3 členné týmy, čímž žáky učíme kooperovat, kombinovat jejich silné stránky a v neposlední řadě vzájemně komunikovat. Snažím se tedy docílit toho, aby žáci porozuměli problému a následně své myšlenky dokázali správně a pochopitelně zformulovat ostatním. [7]

Všechny tyto vlastnosti infromatického myšlení jsem se pokusil do svých úloh zakomponovat. Úlohy jsou tvořeny tak, aby šli systematicky za sebou podle náročnosti a vždy se zabývaly určitou problematickou oblastí v odvětví programování. To, že mají žáci již splněné například používání cyklů neznamena, že se s nimi již v dalším programování nesetkají. V dalších úlohách bude zakomponované také, jen se již nebude jednat o hlavní cíl dané úlohy, ale bude k tomu přidána nějaká nová znalost. Tímto si žáci vědomosti lépe zopakují a zapamatují.

## 2.4 Algoritmické myšlení

Na první pohled by se mohlo zdát, že pojmy algoritmické a infromatické myšlení jsou podobné nebo dokonce totožné. Přesto, že je to lidmi často vnímáno stejně, v teorii tomu tak není. Co se týče podobnosti, tak zde bychom určitě některá rysy našli. Například se může jednat o to, že oba pojmy souvisí s hledáním řešení předloženého problému. Algoritmické myšlení je často uváděno jako oblast, ve které se již zmiňované infromatické myšlení využívá. [9]

Algoritmizaci vnímáme jako proces tvorby algoritmu, který řeší daný problém. Co se týče základních fází algoritmizace tak jsou následující: [10]

- Formulace problému
- Analýza úlohy
- Vytvoření algoritmu
- Odladění algoritmu

Z těchto tvrzení by se dalo stručně říci, že algoritmické myšlení je poměrně významná součást infromatického myšlení.

Při tvorbě svých úloh jsem se tedy snažil, aby každá úloha nerozvíjela algoritmické myšlení pouze obecně, ale aby se zaměřila na jednotlivé fáze potřebné k dostatečnému rozvoji této oblasti.

Co se týče první části, čili formulace problému, tak zde je nejdůležitější správně a srozumitelně formulovat zadání, aby nedocházelo například ke špatnému pochopení ze strany žáků. Zadání by také mělo tvořit motivační funkci, čili buďto reflektovat zájmy a koníčky mládeže, nebo alespoň přiblížit samotnou úlohu. O toto se u mých

úloh může částečně postarat i samotné využití robotů, kteří bývají pro žáky často velmi fascinující a zábavní, čímž je mohou snadno motivovat k práci. [10]

Fáze analýzy úlohy je stále zaměřena na formulaci zadání. Tato metodika vyžaduje, aby zadání obsahovalo jasné pokyny pro postup při žakovském řešení. Pokyny by měly sloužit pouze jako orientační body, nikoliv jako návod nebo dokonce přesný postup. Na druhou stranu by neměly žáka omezovat na hledání pouze jediného řešení ale spíše podceňovat jeho přirozenou kreativitu. [10]

Fáze vytvoření algoritmu mluví sama za sebe. Vede k samotnému návrhu algoritmu a následnému vytvoření programu. [10]

Ve fázi odladění je důležité vytvořit vhodné prostředí pro ladění a optimalizaci programu. Tomu se ve svých úlohách věnuji nejčastěji pomocí doprovodných otázek v zadáních, které žáky nabádají k pozorování a odzkoušení jimi vytvořeného programu. [10]

Podle další dostupné literatury by se dalo algoritmické myšlení popsat jako klíčový koncept v oblasti informatiky, který zahrnuje hlavně rozkládání složitějších problémů na několik menších, což se dá definovat jako dekompozice. Dekompozice patří spolu s abstrakcí mezi jedním z nejdůležitějších složek algoritmického myšlení. Mezi další velmi důležité části algoritmického myšlení můžeme zahrnout například navrhování postupu krok za krokem pro splnění konkrétního úkolu nebo dosažení požadovaného cíle. Neméně důležitou součástí je také ladění, neboli rozpoznávání a následné opravování chyb. V informatice je algoritmické myšlení základem pro programování a výpočetní řešení nejrůznějších problémů. [11]

V tomto článku jsem se dočetl i o pozitivním vlivu využití robotických hraček při výuce. Článek se zabývá konkrétně roboty Dash a Dot a jejich využití. Využití robotů společně s programovací aplikací Blockly bylo velmi účinným prostředkem k rozvoji algoritmického myšlení u žáků základní školy. Dále zkoumaný rámec výuky algoritmického myšlení ukázal pozitivní efekty na rozvoj dovedností učitelů a žáků v oblasti výpočetního myšlení. U podobných úloh je také důležité jakožto učitel poskytovat dostatečnou oporu a diferencovat vyučování pro úspěšné zapojení žáků a podporu jejich učení. Roboti Dash a Dot nabízejí poměrně zajímavý pohled do budoucnosti díky jejich dynamické interakci. Aby byl využit jejich plný potenciál tak je velmi důležitý strukturovaný přístup, vše musí být přizpůsobeno potřebám dětí a

jejich učebním trajektoriím. Algoritmické myšlení se tedy stává konkrétní dovedností, kterou si žáci mohou postupně osvojovat a případně přenášet dále. [11]

Skutečnost, že jsem našel článek o využití robotů Dash a Dot a jejich pozitivním vlivu na algoritmické myšlení mi velmi pomohla si před samotnou tvorbou úloh utřídit myšlenky a rozhodnout se, kterým směrem se budu při samotné tvorbě úloh ubírat. Snažil jsem se tedy o to, aby úlohy rozvíjeli pokud možno všechny aspekty již zmiňovaného algoritmického myšlení. Konkrétně tedy, aby zadání žáky nabádala k dekompozici a mohli si každou úlohu rozdělit na několik menších problémů. Z tohoto důvodu jsou také samotné úlohy řazeny podle náročnosti kdy nejdříve jsou jednodušší úlohy zaměřené vždy na jeden hlavní problém a následují úlohy složitější, kde se již jedná o kombinaci různých problematických částí programování.

## **3 Dash & Dot**

Dash a Dot jsou robotické hračky převážně modré a oranžové barvy. Svou povahou jsou velmi vhodné ke vzdělávání a vyučování na druhém stupni základní školy. Oba roboti komunikují v anglickém jazyce. Disponují nejrůznějšími čidly a senzory, čímž se pro žáky stávají více interaktivní a zábavní. Jedná se o roboty vyvinuté společností Wonder Workshop. [12]

### **3.1 Popis robotů**

Dash je schopen se pomocí svých koleček pohybovat po různých typech povrchů, čili k práci s ním není nutná žádná speciální podlaha ani zázemí. Může se pohybovat dopředu, dozadu ale je schopný se i otáčet, což zpřístupňuje zajímavé pohyby a manévry, které se dají vhodně využít při výuce programování. Je vybaven několika senzory, například infračervené senzory pro detekci překážek nebo senzory pro detekci zvuku, což mu umožňuje aktivně reagovat na okolní podněty. Disponuje také několika LED světly a reproduktory, čímž je schopen vytvářet zvuky a světelné efekty. Díky těmto vlastnostem můžou žáci vytvářet nejrůznější multimediální projekty. Mnou popisovaný robot je také modulární, což znamená, že k němu lze připojit rozšiřující moduly, jako například katapult nebo xylofon. Je navržen pro programování pomocí blokových programovacích jazyků, které jsou intuitivní a lehce pochopitelné pro žáky základních škol. Konkrétně využívá prostředí Blockly, kterému se budeme věnovat v pozdějších fázích práce. [13]

Dot je jakýmsi partnerem robota Dash, který byl vyvinut stejnou společností. Jedná se o menší verzi určenou hlavně jako doplněk k Dashovi. Spolu roboti tvoří kompletní vzdělávací sadu pro rozvoj algoritmického myšlení. Dot není opatřen kolečky, čili se nemůže pohybovat. Na druhou stranu ale disponuje několika tlačítky a senzory náklonu, které z něj dělají důležitou interaktivní část celé sady. Což je vhodné pro sadu úloh. Dot rozšiřuje možnosti využití obou robotů ve vzdělávacím prostředí. [13]

### **3.2 Blokové prostředí blockly**

Blockly je grafické programovací prostředí vyvinuté konkrétně pro práci s robotickými hračkami Dash a Dot. Jedná se o blokové prostředí, čili žáci nejsou



nuceni kód psát ručně, ale pouze pomocí přetahování a spojování různých bloků vytvářejí souvislé programy. Díky tomuto je prostředí vhodné pro žáky základních škol. Blockly má ale jednu velkou nevýhodu a to, že není dostupné v českém jazyce, čímž se na území České Republiky stává vhodné až pro žáky 2. stupně základních škol, kteří už se nějakou dobu výuce anglického jazyka věnují. Na druhou stranu jsem během tvorby úloh v prostředí nenarazil na žádná slovíčka, která by byla problematická a u výuky bude vždy učitel, který by se měl předem připravit na možné dotazy ohledně cizího jazyka. [14]

Blokové prostředí je velmi snadno pochopitelné také díky svému barevnému rozdělení do několika kategorií, kdy se každá kategorie zabývá jinými funkcemi. Dohromady se v prostředí vyskytuje 9 částí, které se dají různě kombinovat pro tvorbu nápaditých a zajímavých programů. [14]

Hlavním důvodem proč jsem si pro svou práci vybral právě prostředí blockly byla jednoznačně dostupnost. Jedná se o aplikaci, která je volně ke stažení pro všechna mobilní zařízení. Ať už se systémem android nebo IOS. Kvůli těmto možnostem jsem mohl úlohy tvořit bez problému z tabletu nebo telefonu. Dále se jedná o aplikaci vyvinutou přímo pro programování těchto robotů a proto jsem se chtěl vyhnout případným problémům při zvolení jiného způsobu programování. V neposlední řadě je aplikace pomocí bloků poměrně jednoduše ovladatelná a zároveň velmi podobná prostředí scratch, se kterým se žáci na základních školách velmi často setkávají. Dalším důvodem byla také možnost pracovat na mobilních zařízeních, protože tím vznikne zajímavá pestrost, kdy žáci při výuce nemusí pracovat pouze na stolních počítačích.

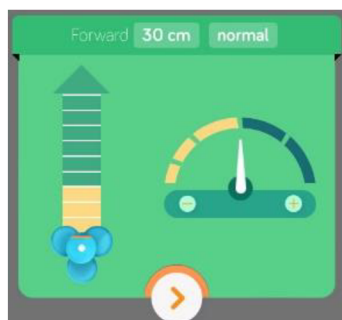
## 1. Start

Tato část se věnuje blokům, díky kterým žáci mohou vytvářet funkce a následně je volat, což se hodí pro organizaci bloků při tvorbě delších programů. Dále obsahuje blok **When**, kde se dá vybrat ze škály několika podmínek pro oba roboty, například pohyb Dota, stisknutí tlačítka nebo zda má Dash někde překážku.

## 2. Drive

Tato část má na starosti hlavně Dashův pohyb. Jsou zde bloky jako například **Forward**, **Backward**, **Turn Left** nebo **Turn Right**. Jedná se o jednu z nejdůležitějších částí. U bloků se dá vybírat z nejrůznějších parametrů jako

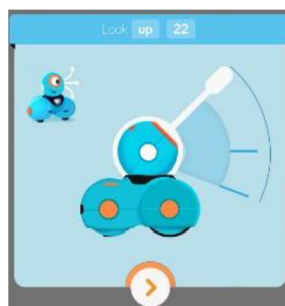
například jakou rychlostí má Dash jet, jak daleko má jet nebo o kolik stupňů se má otočit. Co se týče vybírání konkrétních hodnot tak se mi zdá velmi zábavné a vhodné pro žáky základních škol, protože nemusí nikde vypisovat čísla, ale pouze hodnoty určí pomocí grafického výběru, jako je tomu na následujícím obrázku. Tímto lze také zamezit syntaktickým chybám během tvorby programu.



Obrázek 1 nastavování rychlosti

### 3. Look

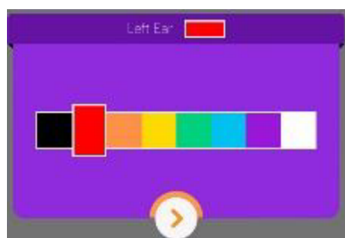
Tato část má pouze jednu důležitou činnost a to že pomocí bloků zajišťuje pohyb Dashovy hlavy. Dash může díky těmto blokům hlavu otáčet až o 120 stupňů doleva nebo doprava. Hlavu lze natáčet i směrem dolů nebo vzhůru. Nachází se zde také blok, díky kterému může Dash natočit hlavu směrem ke zvuku. Hodnoty jsou znovu nastavovány pouze graficky, podobně jako na obrázku.



Obrázek 2 natáčení hlavy

### 4. Light

V této části mohou žáci najít bloky určené k zapínání nebo vypínání LED světel na Dashovi. Mohou si vybrat ze škály několika barev. Jedná se spíše o zábavnou nastavbu, kterou ke své práci žáci tolik potřebovat nebudou. Tuto část mohou ale využít například během úlohy předváděčka. Vzhledem k využití led světel si myslím, že zrovna tato kategorie by žáky mohla velmi bavit.



**Obrázek 3 ovládání světel**

## **5. Sound**

Tato část se zabývá zvukovými efekty pro oba roboty. Roboti mají již naučeno několik zvuků, které mohou žáci libovolně kombinovat a používat. Tato část je ale hůře využitelná při výuce, vzhledem k většímu počtu žáků ve třídách by se mohli vzájemně vyrušovat.

## **6. Animations**

Tuto část bloků budou žáci během své práce využívat minimálně, ale jedná se o různé předem nastavené animace obou robotů. Konkrétně jsou to nejrůznější kombinace světel, zvuků nebo různých pohybů. Vhodné by mohly být pro prvotní seznámení s roboty nebo se jimi žáci mohou inspirovat během své tvorby.

## **7. Control**

V této části se nachází velmi důležité bloky. Konkrétně se jedná o čekání, využívání cyklů a podmínky. Tyto bloky budou žáci využívat k tvoření velmi často. Ve vytvořené sadě se nachází úlohy, které jsou vytvořeny právě na procvičování těchto funkcí.

## **8. Variables**

Tato část se zaměřuje na proměnné, kterým se žáci budou věnovat v pozdějších fázích u tvorby složitějších programů. Žáci zde najdou základy, jako je nastavení nebo změna proměnné. Tu mohou dále využít například v předpřipraveném cyklu nebo vytvořit nějakou podmínku. Problém je, že tyto bloky se nedají kombinovat například s bloky v části Drive, ale mají zde specifický blok, díky kterému mohou být proměnné využity v programu.

## **9. Accessory**

Poslední část blokového prostředí se věnuje doplňkům jako je například katapult, kterým se budou žáci věnovat až ve 3. bloku úloh. Jsou zde k dispozici i výuková videa, která mohou být využita k vysvětlení toho jak s těmito pomůckami pracovat a

jak je k robotům připojit. Nachází se zde například bloky určené pro nabíjení nebo výstřely z katapultu. Dále mohou žáci k robotovi připojit rozšíření, díky kterému mohou například psát. Samotná sada, ale obsahuje i poměrně přehledný obrázkový návod, jak s rozšířeními pracovat, takže by to neměl být problém.

### **3.3 Nedokonalosti blokového prostředí**

Jedná se o blokové prostředí, což už samo o sobě znamená, že se bude jednat o poměrně zjednodušenou verzi některého programovacího jazyka. Co se týče největších překážek, se kterými jsem během své práce měl problém nebo by mohli mít samotní žáci, tak se jedná hlavně o využití anglického jazyka. Na druhou stranu jsme v tomto ohledu již poměrně daleko a žáci se s tímto cizím jazykem setkávají již od poměrně útlého věku a v blokovém prostředí se nevyskytují žádná odborná slova, která by mohla být problematická. U výuky bude vždy přítomný učitel, který se může postarat o hladký průběh a zajistit žákům s touto problematikou pomoc.

Během tvorby úloh jsem se několikrát setkal s tím, že jsem vytvořil nějaké zadání, ale nakonec jsem zjistil, že pomocí jednotlivých bloků ho následně nejsem schopen realizovat. Z tohoto důvodu jsem zadání spolu s vzorovým řešením vždy několikrát procházel a následně jsem se ho snažil koncipovat tak, aby bylo proveditelné i v tomto zjednodušeném prostředí. Konkrétně jsem měl problém s tím, že například podmínky nelze dávat vedle sebe, podobně jako při programování v textových programovacích jazycích díky použití funkce OR nebo AND. Nakonec jsem ale vždy došel k nějakému kompromisu a úlohy upravil tak, aby byly proveditelné.

Další problém, se kterým jsem se během svého tvoření setkal byl s reakcí na zvuk. Měl jsem vytvořeno několik úloh, které by se touto problematikou mohly zabývat, ale kvůli nedokonalosti senzorů jsem je nakonec bohužel nemohl využít. Hlavní problém je ten, že pokud chceme od robotů například nějakou reakci na zvuk, tak roboti nejsou schopni rozeznat o jaký zvuk se jedná. Já jsem úlohy tvořil a testoval v poměrně klidném domácím prostředí a přesto se roboti nechovali podle očekávání. Ke spuštění senzoru stačí například pouhé posunutí židle ve vedlejší místnosti. Z těchto důvodů si nedokáží představit, jak by se se zvukem dalo pracovat ve školním prostředí, kde je několik žáků současně.

Objevil jsem ještě jeden problém, který není úplně způsoben blokovým prostředím, ale s roboty pořád poměrně úzce souvisí. Jedná se o to, že pokud budou žáci využívat

úlohy z kategorie rozšíření, tak mohou mít problém se samotným nabíjením katapultu a jeho výstřely. I přes všechnu mou snahu se mi poměrně často stávalo, že ve chvíli, kdy měl Dash svůj katapult nabít, tak došlo k tomu, že se Dash otočil příliš velkou silou a míček tak nezapadl do správné jamky. Bohužel zde není možnost toho, abychom například nastavili rychlost nabíjení, čímž by se mohl daný problém vyřešit. U těchto úloh je ještě jeden menší problém a to, že jakmile není síla výstřelu z katapultu větší než alespoň 60%, tak Dash nevystřelí, ale míček se mu vrátí zpět do nabití fáze.

Dále jsem během tvorby objevil problém při práci s proměnnými. Konkrétně se jedná o to, že aby mohli žáci proměnou využít například k pohybu robota nebo jiné aktivitě, již nemůžou používat bloky z kategorie **Drive**, ale musí použít speciální blok schovaný pod názvem **Variables**. Toto může být pro žáky trochu matoucí, ale věřím, že s pomocí učitele vždy najdou správné řešení.

Během své práce jsem přišel na další poměrně závažný problém ohledně mnou vybraného blokového prostředí, který může mít negativní následky na samotnou práci se žáky. Samotné prostředí nabízí verzi pouze pro mobilní zařízení, což může při výuce znamenat zásadní problém, se kterým by se mělo vždy předem počítat. Žáci by proto měli mít k dispozici například tablety nebo jiná mobilní zařízení, která budou s daným prostředím kompatibilní.

Všechny výše zmíněné nedokonalosti jsem posbíral buďto během své tvorby nebo následného testování úloh se žáky. Nakonec jsem došel k závěru, že se nejedná o zásadní problémy, díky kterým bych nebyl schopný práci realizovat. Některé z uvedených nedokonalostí mohou být ve výsledku v jiném kontextu brány i jako lehké výhody. Například využití mobilních zařízení se mi nakonec během testování osvědčilo jakožto výhoda, kdy žáci nemuseli pracovat na stolních počítačích a měli díky tomuto alespoň změnu oproti klasické výuce programování.

## 4 Postoje žáků k výuce robotiky

Studie provedena v rámci dokumentu *Robots in Elementary School* v roce 2016 zkoumala postoje a perspektivy žáků základních škol z Polska a Ukrajiny na téma výuky robotiky. V této době nebyla robotika ještě zdaleka tak rozšířena jako dnes, ale na druhou stranu se nejedná o tak dávnou minulost, aby se získaná data nedala použít i v současnosti. [15]

Z výsledků této studie bylo patrné, že žáci projevují velký zájem o učení základů robotiky. Na druhou stranu zde stále byla menšina žáků, která robotiku vnímala spíše jako proces mimoškolního vyučování. Tyto výsledky naznačují, že je potřeba výuku robotiky a programování diferencovat, aby byly brány v potaz různé preference a zkušenosti žáků. Většina respondentů ale vyjádřila podporu výuky robotiky ve školních třídách samozřejmě pod podmínkou, že škola poskytne vhodné podmínky a vybavení. Z tohoto postoje je patrné, že žáci jsou otevřeni k novým technologiím a je zde potenciál pro rozvoj dovedností u žáků. [15]

Co se týče začlenění robotiky do školní osnovy tak většina respondentů preferovala začlenění s rozsahem 1-2 hodin týdně. Žáci si tedy uvědomují hodnotu a potenciál výuky robotiky a rádi by ji začlenili do vzdělávacích plánů základních škol. Z dotazníku vyšly tyto konkrétní hodnoty, kdy 72,5% polských respondentů a 76,6% ukrajinských respondentů projevilo zájem o učení robotiky. Naopak pouze 19,3% polských respondentů a 18% ukrajinských respondentů vyjádřilo názor, že je výuka robotiky nevhodná. Tento postoj může být ovlivněn například obavami z nových technologií. Zbylí respondenti uvedli nejednoznačné odpovědi, na základě kterých by se mohly následně navrhovat efektivní vzdělávací programy. [15]

Důležitou roli při výuce robotiky hraje také adekvátní vybavení, jako jsou stavebnice LEGO Robotics, LEGO MINDSTORMS, WeDo 2.0 nebo právě roboti Dash a Dot. Tyto technologie umožňují zábavný způsob výuky robotiky u dětí. Práce s roboty může mít také velmi pozitivní vliv na týmovou spolupráci a soutěživost mezi žáky, což může vést k efektivnějšímu učení a rozvoji dovedností v rámci vzdělávacího procesu [15]

Na základě výše zmíněného článku a výsledků zkoumaného dotazníku je patrné, že je zájem žáků o výuku robotiky významný a existuje potenciál pro rozvoj

dovedností v této oblasti, konkrétně tedy i algoritmického myšlení. Díky tomu, že většina respondentů vyjádřila podporu výuce robotiky na školách je mnou vytvořená sada úloh velmi vhodná pro využití ve vzdělávání na základních školách. Výsledky studie také naznačují důležitost diferencovat výuku, což se snažím ve své sadě řešit použitím základního a rozšířeného zadání.



## 5 Vliv robotických hraček na algoritmické myšlení

Cílem výuky informatiky, potažmo algoritmicizace není pouze naučit žáky konkrétní programovací jazyk, ale spíše rozvíjet jejich algoritmické a infromatické myšlení. V článku Vliv robotických hraček na algoritmické myšlení jsem se dočetl několik zajímavých informací o tom, jak práce s roboty přispívá k rozvoji právě těchto kompetencí. [16]

Podle výše zmíněného článku je důležitý praktický přístup k výuce s využitím fyzických objektů, jako mohou být například robotické hračky. Díky tomu mají žáci možnost s objekty interaktivně pracovat, čímž poskytujeme názornější způsob porozumění principům programování a algoritmického myšlení. Hlavním přínosem práce s robotickými hračkami je jednoznačně názornost a interaktivita. Na rozdíl od tradičních metod výuky programování, které se často omezují na pouhé psaní kódu na obrazovku počítače. Tyto hračky umožňují žákům pracovat s fyzickými objekty a vidět okamžitý výsledek svého programování.[16]

Na trhu je k dispozici široká škála robotických hraček s různými vlastnostmi a funkcemi. Každá z těchto hraček má své pedagogické výhody a omezení, které je důležité vzít v úvahu při výběru nástroje pro vzdělávací účely. Do své práce jsem tedy vybral roboty, kteří se nemusejí nijak sestavovat. Žáci tak neztrácejí čas mechanickými dovednostmi a mohou začít ihned programovat.[16]

Dále můžeme pomocí práce s roboty velmi efektivně využívat a podporovat práci ve skupinách. Díky tomu se žáci aktivně zapojí do procesu učení a můžeme u nich rozvíjet také schopnost vzájemně spolupracovat s ostatními, sdílet své nápady, myšlenky a komunikovat. [16]

Práce s robotickými hračkami samozřejmě není jediným způsobem, jak rozvíjet algoritmické myšlení. Co možná nejlepším způsobem je každopádně kombinace různých metod jako je například klasické programování nebo modelování. Robotické hračky přináší do výuky programování a rozvoje algoritmického myšlení nové možnosti a přístupy. Díky tomu se mohou žáci naučit programování a v neposlední řadě i systematické řešení problémů. Důležité je, aby výuka programování byla přizpůsobena potřebám a schopnostem jednotlivců a poskytovala prostor pro jejich tvořivost a experimentování. [16]

Díky tomuto článku jsem zjistil, že práce s roboty má u žáků na základní škole určitě velmi velký potenciál. Roboti mají opravdu pozitivní vliv na rozvoj algoritmického myšlení u dětí. Bohužel jsem nenašel žádný článek, který by se věnoval přímo robotům Dash a Dot, ale věřím, že programování je u většiny těchto hraček velmi podobné. Rozhodl jsem se tedy na základě zjištěných skutečností vytvořit sadu úloh pro rozvíjení právě těchto kompetencí.

## 6 Sada úloh

V následující části mé práce se nachází vytvořená sada úloh, která je rozdělena do několika kategorií na základě toho, jak spolu úlohy souvisí. Sada obsahuje celkem 14 úloh rozdělených do 3 částí. Úlohy jsou očíslovány tak, jak by měly být za sebou s žáky procházeny. Podle možností a zkušeností mohou žáci pracovat jak samostatně tak po skupinách po přibližně 3-4 žácích. Záleží na podmínkách a možnostech každé školy a třídy. Každá úloha je rozdělena stejným způsobem. Nejdříve je zadání, které by se mělo předat žákům před samotnou prací. Následují výukové cíle dané úlohy spolu s krátkým komentářem pro učitele, kde je zdůrazněno co je u dané úlohy důležité. U některých úloh se dále vyskytuje rozšíření, které je vhodné pro diferenciatní výuku. Může být využito například pro rychlejší nebo nadanější žáky. Složitější úlohy obsahují také tip, což může fungovat jako lehká nápověda pro žáky. Většinou se zde dozví jaké bloky jsou pro řešení nejvhodnější a podobně. Nakonec je u každé úlohy přiloženo vzorové řešení formou obrázku. Toto řešení je následně doplněno o krátký komentář.

Při tvorbě této sady úloh jsem se snažil opřít o výukové cíle informatiky, didaktiku informatiky a v neposlední řadě o aspekty algoritmického a inforatického myšlení.

### 6.1 Pohybové úlohy

Během práce na pohybových úlohách se žáci postupně seznámí s nejrůznějšími dovednostmi v oboru programování. 1. úloha funguje jako návod pro samotné připojení robotů a prostředí blockly. Žáci začnou na jednoduchých úlohách, kde budou využívat hlavně robota Dashe a jeho pohyb. Postupně se k pohybovým blokům přidá využívání cyklů, podmínek a nakonec v rozšířeních u některých úloh i využití proměnných. Ve stejné posloupnosti, jak jsou úlohy v práci číslovány by měly být i následně využívány. Úlohy postupují podle obtížnosti a postupně se nabalují nové vědomosti. Všechny pohybové úlohy se zaměřují hlavně na Dashe a jeho pohybové dovednosti. Dot je zde využíván minimálně.

## 6.1.1 Začínáme programovat

### Zadání:

Nejdříve začneme tím, že pomocí podržení tlačítka oba roboty zapneš. Následně můžeš otevřít aplikaci blockly, kde pomocí oranžového plus v pravém horním rohu připojíš oba roboty. Roboti se k zařízení připojují pomocí Bluetooth, tak si dej pozor abys ho měl zapnuté a povolené. Úspěšné připojení poznáš tak, že ikony obou robotů zezelenají stejně jako na obrázku.



Obrázek 4 ukázka správného připojení robotů

Další důležitý krok, než začneme programovat je vytvoření nového projektu. Pomocí **Menu** -> **My Projects** -> **Create New** -> **Blank Project** si vytvoříš nový projekt. Dbej zvýšenou pozornost tomu, jak projekt pojmenuješ, ať v budoucnu nemáš problém mezi sebou projekty rozeznat. Pokud si vše udělal správně, měl by si na ploše vidět pouze blok **When Start**.

### Cíle a poznámky k úloze:

Co se týče úlohy **Začínáme programovat** tak jejím cílem není cílem dostat od žáků nějaký jimi vytvořený projekt, ale jedná se spíše o prvotní seznámení s roboty a s prostředím ve kterém budou pracovat. Měla by být doplněna o lehký výklad k samotnému blokovému prostředí kde by měly být popsány jeho nejdůležitější bloky a funkce. V případě, že učitel ví o věcech, které jeho žákům dělají problémy není od věci jim tyto problematiky znovu dovysvětlit, aby se předešlo pozdějším žakovským problémům s ostatními úlohami.

## 6.1.2 První krůčky

### Zadání:

Pokud se ti povedlo úspěšně dokončit připojení robotů a vytvořit si svůj vlastní projekt, můžeme začít s 1. úlohou. Budeme se věnovat základnímu pohybu, který může připadat jednoduchý, ale jedná se o základní stavební kámen pro všechny ostatní úlohy.

V této úloze bude tvým úkolem vyzkoušet několik bloků z kategorie Drive, konkrétně se bude jednat o bloky **Forward**, **Backward**, **Turn Left** a **Turn Right** a následně tyto bloky využít k vytvoření Dashova pohybu. Postupně využívej všechny bloky a pečlivě sleduj, jak se Dash chová. Jistě sis všiml, že jsou zde defaultně nastaveny nějaké hodnoty. Co když tyto hodnoty změníš? Jak to ovlivní chování Dashe?

Nakresli pomocí robota písmeno L.

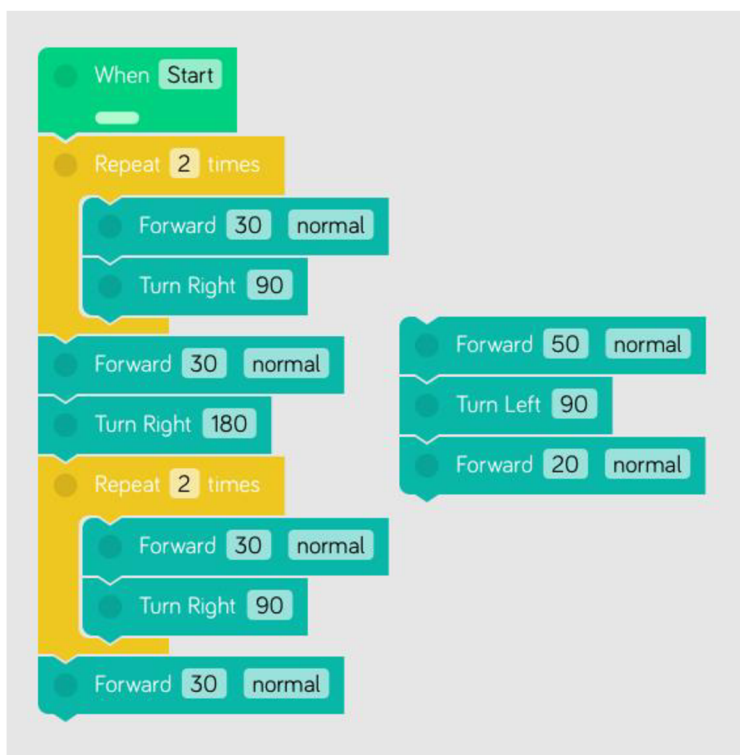
**Rozšíření:** Vytvoř program, díky kterému Dash nakreslí číslo 3. Číslo může být hranaté, můžeš se inspirovat například sedmi-segmentovým displejem, který se využívá například u digitálních hodin. Robot může začínat na kterémkoli konci.

### Cíle a poznámky k úloze:

V úloze s názvem **První krůčky** se žáci seznámí s konkrétními pohybovými bloky a jejich výstupem bude 1. smysluplný program. Žáci si osvojí pohybové bloky a pokusí se robotem nakreslit číslo L. Zadání není nijak více specifikováno, takže hodnotit musíme hlavně to jak žáci k problému přistoupili a zda po spuštění programu robot udělá požadovaný pohyb. Pomocí doprovodných otázek v zadání se v žácích snažím vzbudit větší zájem a to, aby se o svém programu pokusili více přemýšlet a ne pouze splnit zadání.

Co se týče rozšíření, tak zde již žáci pouze aplikují vědomosti získané díky pozorování robotů. U této úlohy není důležité využití cyklů, proto není problém pokud žáci vytvoří dlouhý program bez opakování. Nejdůležitější je, aby si dostatečně osvojili bloky ovládající pohyb robota, protože se jedná o základní stavební kámen pro všechny ostatní úlohy. Je pouze na nich, zda se rozhodnou využít například couvání, nebo robota otočí o 180 stupňů.

### Vzorové řešení:



Obrázek 5 vzorové řešení: První krůčky

U této úlohy neexistuje jediné správné řešení. Tím, že v úloze není definováno, zda má Dash například couvat nebo podobně mohou žáci využít téměř libovolný způsob. Musí ale samozřejmě dosáhnout požadovaného výsledku. Kontrolujeme tedy hlavně samotný pohyb robota, zda vytvořený pohyb opravdu připomíná písmeno L případně číslo 3. V zadání není nikde definován rozměr požadovaných útvarů, takže velikost je v tomto případě čistě na žácích.

### 6.1.3 Čtverec

#### Zadání:

Super, základní pohyb máme za sebou a tak můžeme začít tvořit trochu komplexnější programy.

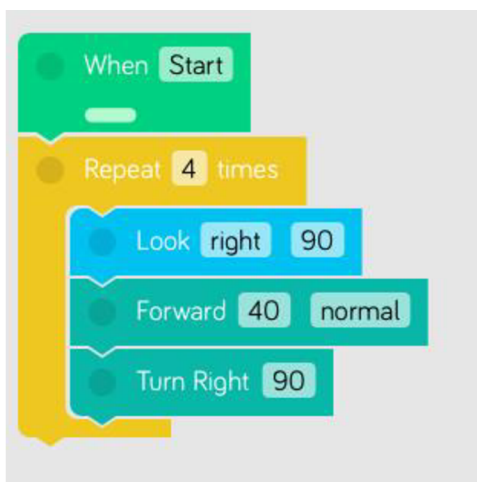
V této úloze bude tvým úkolem vytvořit program, díky kterému Dash udělá pohyb do čtverce. Strana čtverce by měla být dlouhá 40 cm. Dej si pozor na to aby Dash skončil ve stejném bodě jako začínal. Jestli již Dash umí čtverec, pomocí dalších bloků zajisti, aby měl hlavu při jízdě vždy otočenou do středu čtverce.

**Rozšíření:** Pokud se ti v programu bloky hodně opakují, nešlo by program zkrátit? Vylepši program tak, aby byl každý blok vždy použit maximálně jednou. K tomu by ti mohly být užitečné některé bloky z kategorie **Control**.

#### Cíle a poznámky k úloze:

Úloha **Čtverec** přímo navazuje na předchozí úlohu a jejím hlavním cílem je procvičit doposud získané dovednosti ohledně pohybu a navázat na ni novou dovedností v podobě použití cyklů. Žáci mají za úkol vytvořit program díky kterému se robot na podlaze bude pohybovat do čtverce a poté přidat několik nových bloků pro náklon jeho hlavy. Následně jsou v rozšíření úlohy vybízeni k tomu, aby využili nové bloky a jimi vytvořený program tím hodně zkrátili. V této úloze je již velmi důležité aby učitel při kontrole dal větší pozornost tomu, zda žáci opravdu cykly použili. Úloha u studentů prohlubuje schopnost analyzovat problém a strategicky přemýšlet o tom, jak nejlépe dosáhnout požadovaného řešení.

#### Vzorové řešení:



Obrázek 6 vzorové řešení: Čtverec



U této úlohy je již důležité dbát na to, aby žáci správně využili cykly. Jako správné řešení tedy nemůžeme považovat příliš se opakující bloky i přesto, že se Dash bude pohybovat správně. Také je zde přidáno otáčení hlavy. Pokud budeme kontrolovat správnost řešení, musíme tedy dbát na to, zda žáci splnili opravdu všechny části zadání.

## 6.1.4 Pozor!

### Zadání:

Jak název napovídá, podstatou této úlohy bude zjišťování, zda nemá Dash kolem sebe nějakou překážku do které by mohl nabourat.

Tvým úkolem bude Dashe rozpohybovat tak, aby se rozjel rovně dopředu. V případě, že před sebou uvidí nějakou překážku, tak se zastaví a následně začne couvat. Pokud během couvání Dash detekuje překážku za sebou, opět se zastaví a pojedede vpřed.

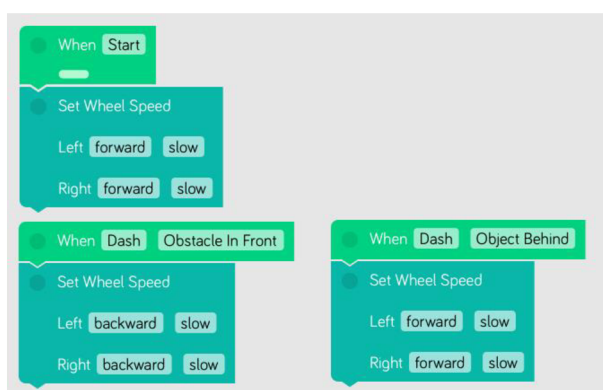
Nastav některou z pomalejších rychlostí pohybu, aby se dalo fungování programu pořádně vyzkoušet. Dej si pozor na to, aby byl program po spuštění funkční až do chvíle než jej někdo přeruší nebo vypne.

K vyzkoušení, zda program správně funguje můžeš využít například sešit ale klidně i Dashova pomocníka Dota.

### Cíle a poznámky k úloze:

Hlavním cílem úlohy s názvem **Pozor!** je žáky seznámit se skutečností, že roboti disponují svými senzory a naučit je s nimi pracovat. Žáci budou tvořit takový program, aby Dash během svého pohybu monitoroval okolí a vždy před překážkou začal buďto couvat nebo naopak jel dopředu. Bez splnění této úlohy budou mít žáci velký problém s ostatními úlohami v tomto bloku, protože nebudou mít procvičenou práci se senzory. Jedná se tedy o prvotní seznámení se s čidly, které žákům usnadní práci v následujících úlohách.

### Vzorové řešení:



Obrázek 7 vzorové řešení: Pozor!

U této úlohy by žáci s řešením neměli mít nějaký velký problém. Jedná se sice pro první využití senzorů, ale pokud využijí správný blok, samotné sestavení do funkčního celku by již mělo jít bez problému. Tato úloha by se samozřejmě dala splnit i pomocí podmínek **if**. Využil jsem ale raději bloku **When**, místo použití podmínky **if**, protože se robot v reálném prostředí a čase choval přirozeněji a přesněji.

## 6.1.5 Hra na schovávanou

### Zadání:

V této úloze navážeme na znalosti a dovednosti získané v předchozí úloze a využijeme dota jako hledanou osobu.

Nejdříve si vymeze prostor, ve kterém budou oba roboti hru hrát. Mělo by se jednat o obdélník nebo čtverec. Prostor si označ nějakými předměty a hlavně si změň délky stran, abys s nimi mohl v programu pracovat. Samotná hra bude probíhat tak, že kdekoliv ve vymezeném prostoru postavíš Dota. Dash postavíš do pravého dolního rohu vymezeného prostoru a vytvoříš program, díky kterému Dash Dota v prostoru vždy najde. Ve chvíli kdy se k němu přiblíží tak pomocí zvukového znamení vyjádří, že ho našel. Mysli na to, že v prostoru se smí nacházet pouze Dot. Dash nesmí nikdy vyjet ven z označeného prostoru ani narazit do dota.

**Rozšíření:** Pokud již máš řešení hotové, vyzkoušej zda opravdu funguje pro každý prostor, který tě napadne. Nešlo by pomocí jedné hodnoty určit velikost prostoru? Abys nemusel vždy při změně přepisovat celý program? Zajisti aby program šel změnit pro jiné rozměry přepsáním pouze jedné hodnoty.

**Tip:** Prostor si nejdříve nakresli na papír a zamysli se, jak by to bylo možné udělat. K tomu, abys vždy při změně prostoru musel přepsat pouze jednu hodnotu ti pomůže proměnná, kterou najdeš v kategorii **Variables**.

### Cíle a poznámky k úloze:

Úloha s názvem **Hra na schovávanou** navazuje na již získané dovednosti. Konkrétně rozvíjí zkušenosti s pohybem Dashe, použití cyklů, zjištění zda se někde nenachází překážka a k tomuto přidává díky svému rozšíření novou znalost a to v podobě proměnné. Hlavním cílem je rozšířit u žáků znalosti ohledně cyklů tím, že jsou nuceni použít jiný cyklus než pouze s konkrétním počtem opakování. Dále se úloha zaměřuje na rozvoj prostorové orientace a schopnost rozvrhnout si práci před samotným výkonem. Úloha vyžaduje vytvoření programu pro navigaci robota v předem určeném prostoru tak, aby pokryl všechny jeho části a našel svého parťáka Dota. Žáci dále získávají praktické zkušenosti v oblasti robotického programování a logického myšlení. V rozšíření se znovu pomocí otázek snažím v žácích vytvořit zájem a podněcovat je k tomu, aby o programu přemýšleli více obecně a ne pouze

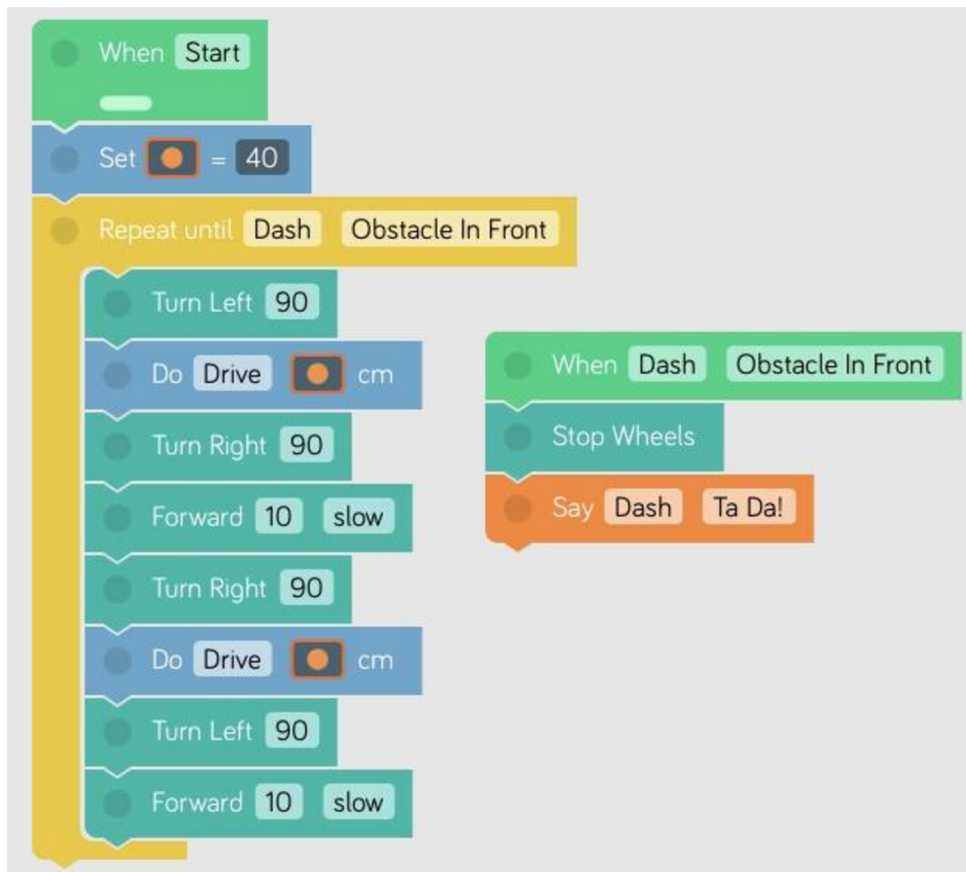
tak, aby splnili zadání. Tímto žáky motivuji k využití proměnné, aby bylo zadání funkční i v případě změny rozměrů prostoru. Důležité je také to, aby si žáci uvědomili, že jediný rozměr, který potřebují k úspěšnému zvládnutí je délka spodní strany jimi vybraného tvaru, protože Dash se bude pohybovat pouze do chvíle, než najde Dota.

### Vzorové řešení:



Obrázek 8 vzorové řešení: Hra na schovávánou

Toto řešení je pro základní část úlohy. Jedná se pouze o to, aby žáci dokázali správně vymyslet systém, kterým se má robot po vytvořené oblasti pohybovat. Následně z této myšlenky vytvořit program, kde využijí již nabitě zkušenosti. Robota musíme ještě pomoci bloku **When** zastavit, jinak program dobíhá až do konce, čímž by se oba roboti mohli srazit.



**Obrázek 9 vzorové řešení: Hra na schovávanou - rozšíření**

Zde je složitější řešení a to s použitím proměnné, díky které můžou žáci libovolně měnit rozměry daného útvaru. Kvůli své náročnosti by mohlo sloužit například jako zábava pro rychlejší žáky, zatímco ostatní budou pracovat na základní části této úlohy. Pokud víme, že pracujeme s šikovnými žáky, tak není problém rovnou od začátku vyžadovat rozšířené řešení.

## 6.1.6 Slalom

### **Zadání:**

Jistě už si někdy při nějakém sportu běhal slalom, i kdyby ne tak nezdoufej. V této úloze budeš mít příležitost si to vyzkoušet s robotem.

Vytvoř alespoň ze 4 stejně velkých předmětů pravidelný slalom. Použít můžeš například lahve od pití, důležité je aby byly překážky přibližně stejně velké a pro robota dostatečně viditelné. Tvým hlavním úkolem bude vymyslet program, díky kterému Dash slalom projede, aniž by zapomněl na nějakou překážku nebo něco shodil. Využij minimálně 4 překážky. Vzdálenost mezi předměty může být stejná.

**Rozšíření:** Pokud již Dash slalom projede bez problémů, můžeme postoupit trochu dále. Vyndež ze slalomu některou z překážek, nebo mezi nimi například udělej různé mezery. Pokud Dash naráží, zajisti, aby slalom projel vždy, nehledě na to jak vzdálené od sebe překážky budou

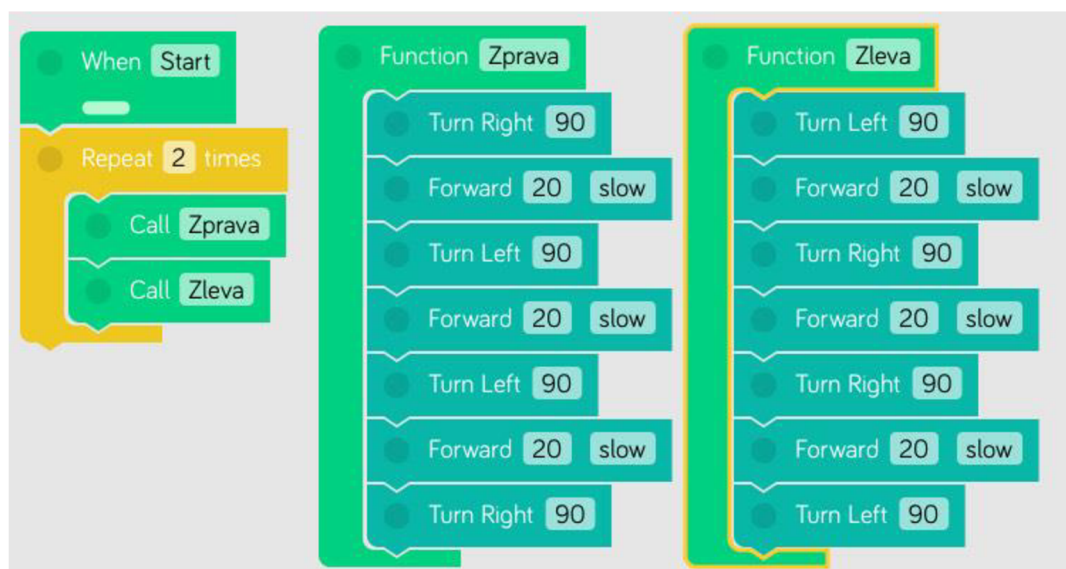
**Tip:** Nejdříve naprogramuj Dashův pohyb kolem překážky a ve finálním programu ho dále používej jako blok **Call Function**. Tímto se budeš v programu lépe orientovat. Podívej se tedy do kategorie Start, kde najdeš vše potřebné. K tomu, aby Dash střídal objíždění z levé a pravé strany využij proměnou.

### **Cíle a poznámky k úloze:**

**Slalom** je poměrně prostorově náročná úloha, čili není vhodná pokud máte hodně žáků na ne moc velkém prostoru. Kdyžtak je možnost využít místo této úlohy zadání z úloh 2C a 2D, které se zaměřuje na podobnou znalost nebo můžete pro toto řešení žáky rozdělit do větších skupin. Co se týče překážek tak já jsem pro vzorové řešení použil obyčejné ruličky od toaletního papíru a fungovaly bez problému. Žáci mají za úkol nejdříve vytvořit slalom mezi překážkami, které jsou od sebe stejně daleko. Hlavním cílem této úlohy je to aby si žáci ozkoušeli, zda jimi ovládaný robot překážky objíždí aniž by do některé narazil. Dbejte pozornost tomu, aby žáci opravdu využili blok funkce z kategorie start, čímž se program pochopitelněji a budou se v něm dále moci lépe orientovat. Poté co žáci vytvoří funkční pohyb robota, pomocí pohybových bloků a cyklů, tak je čeká aby se zamysleli jak by mohli program pozměnit pro více obecné řešení. Jedná se znovu o využití proměnné, ale tentokrát je již program trochu složitější.

Cílem této úlohy je v žácích rozvíjet schopnost tvořivosti, ale také vytrvalosti pokud se jim na první pokus program nepovede vytvořit. Dále tato úloha zlepšuje jejich algoritmické myšlení a schopnost orientace v prostoru. Neméně důležitým aspektem této úlohy je první seznámení žáků s blokem **IF** a **ELSE**, proto žáci musí být s podmínkami již z dřívějších seznámení, abychom zamezili možným problémům, kdy žáci nebudou vědět jak s blokem pracovat. Případně můžeme před touto úlohou jakožto učitel žákům tento blok dodatečně vysvětlit.

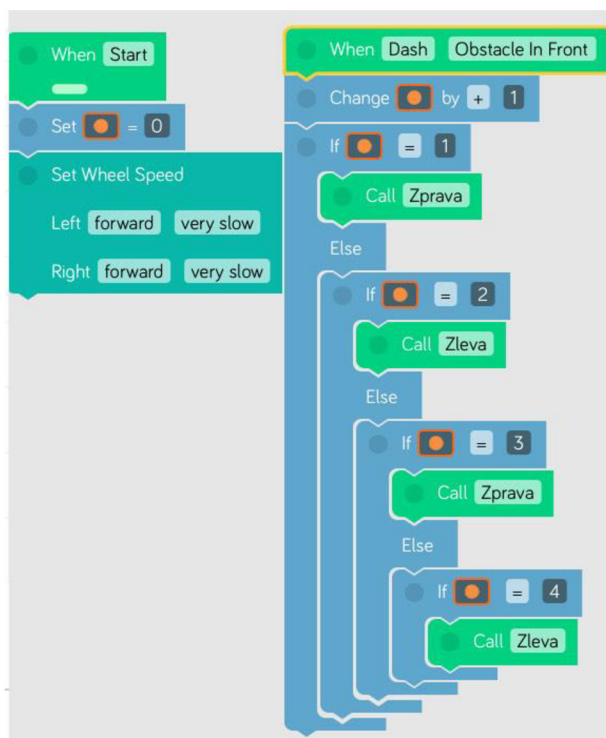
### Vzorové řešení:



Obrázek 10 vzorové řešení: Slalom

V základní části této úlohy si žáci vyzkouší vytvoření vlastní funkce, se kterou mohou následně lépe pracovat. Důležité je tedy správné užití bloku **Function** a následné zasazení do programu. Také by si měli všimnout různých opakování a program tedy pomocí cyklu zkrátit. Funkcím by bylo vhodné věnovat více času, protože se mohou hodit i v dalších úlohách a práci poměrně značně zjednoduší.





**Obrázek 11** vzorové řešení: Slalom - rozšíření

V rozšíření této úlohy si žáci znovu vyzkouší využití proměnných a vytvořit jednoduché rozhodování robota. Znovu bych rozšíření volil spíše pro rychlejší žáky, kteří již budou mít správně vyřešenou základní část úlohy. V této úloze se již využívají současně podmínky a proměnné, proto může být řešení poměrně problematické.

## 6.2 Spolupráce robotů

Následující úlohy jsou zaměřeny hlavně na vzájemnou spolupráci obou robotů. Je zde již více využíván Dot a jeho nejrůznější vstupy jako jsou například tlačítka nebo čidla náklonu. Úlohy jsou znovu seřazeny podle náročnosti a postupně jsou přidávány nebo opakovány nejrůznější znalosti jako například využití opakování, podmínek a v neposlední řadě i proměnných. Před vstupem do této části úloh je důležité, aby za sebou žáci již měli správně splněnou alespoň většinu z předchozího bloku úloh.

## 6.2.1 Reagujeme na Dota

### Zadání:

V těchto úkolech se již zaměříme více na spolupráci obou robotů, čili budeme Dota využívat více aktivněji, než v minulém bloku úloh. Nejdříve si vyzkoušíš jak může Dash reagovat na Dota.

Tvým hlavním úkolem bude vytvořit program, díky kterému se Dash začne pohybovat vpřed. Pomocí bloků zajisti, aby v případě, že bude zmáčknuto prostřední tlačítko na Dotovi, Dash se okamžitě zastaví.

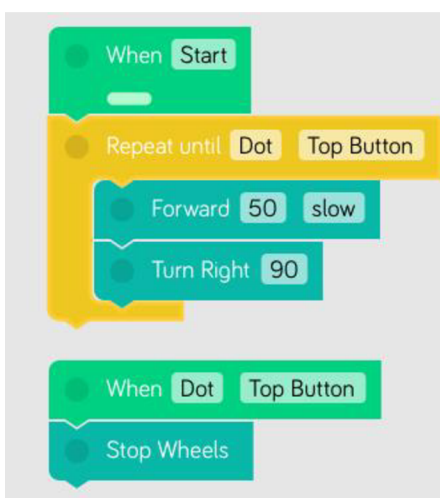
**Rozšíření:** Místo pohybu vpřed využij pohyb do čtverce.

### Cíle a poznámky k úloze:

**Reagujeme na Dota** je první úloha z části spolupráce robotů, žáci se naučí používat oba roboty současně a pomocí Dota ovládat Dashe. V této úloze mají žáci za úkol vytvořit program tak, aby mohli pomocí tlačítka na Dotovi Dashe zastavit. Cílem této úlohy je hlavně žákům ukázat, že mohou k dosažení stanoveného cíle využívat i interakce mezi roboty. Žáci se naučí využívat vstupy jako prostředek k řízení a chování robotů. Co se týče rozšíření, tak jde pouze o zopakování pohybových bloků pro budoucí programování.

### Vzorové řešení:

V této úloze se žáci začnou postupně seznamovat s Dotem. Důležité je aby správně užili cykly a zároveň si osvojili využití některého vstupu od Dashe.



Obrázek 12 vzorové řešení: Reagujeme na Dota

## 6.2.2 Kapitán Dot

### Zadání:

V této úloze bude Dot v roli šéfa a bude Dashovi udávat směr pohybu.

Hlavním úkolem bude vytvořit program, díky kterému budeš moci tím, že položíš Dota hlavou dolů rozpohybovat Dashe směrem vpřed.

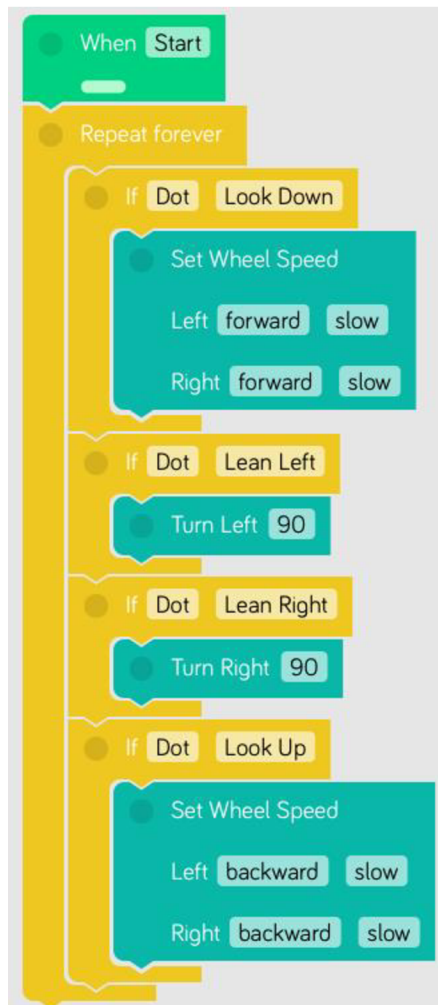
**Rozšíření:** Pokud již Dash zvládá pohyb vpřed, můžeš Dotovi přidat více funkcí. Konkrétně vytvoř takový program, aby se Dash pohyboval podle toho jak bude Dot zrovna nakloněn. Dash se bude pohybovat tímto způsobem:

- Dot má obličej směrem dolů – Dash se pohybuje dopředu
- Dot je na levém boku – Dash se otočí o 90° doleva
- Dot je na pravém boku – Dash se otočí o 90° doprava
- Dot je hlavou vzhůru – Dash couvá

### Cíle a poznámky k úloze:

Úloha s názvem **Kapitán Dot** se zaměřuje hlavně na využití funkce IF. Nejdříve žáci použijí pouze jednu podmínku, díky které rozpohybují Dashe. Následně jsou díky rozšíření nuceni řetězit několik podmínek za sebe tak, aby se robot choval podle zadání. Hlavním cílem je tedy u žáků rozvíjet znalost využití podmínek se kterou se již setkali v předchozích úlohách, ale tentokrát bez využití proměnných. Důležité je to, aby žáci nastavili nekonečné opakování, čímž zajistí, že bude program fungovat po celou dobu od spuštění a ne pouze na využití jedné podmínky. Dále je tato úloha velmi interaktivní, čímž by mohla být pro žáky velmi zábavná a poučná.

## Vzorové řešení:



Obrázek 13 vzorové řešení: Kapitán Dot

V tomto řešení je důležité, aby si žáci uvědomili, že musí využít nekonečný cyklus, do kterého následně vnoří podmínky určené zadáním. Není žádoucí, aby v této úloze žáci používali blok **When**.

### 6.2.3 Zkušený trenér Dot

#### Zadání:

Představ si, že je Dot trenérem a snaží se Dashe připravit na atletický závod. Tentokrát mají trénink běhu a Dot určuje rychlost.

Tentokrát budeme zase pracovat s tlačítky na Dotovi. Tvým úkolem bude zajistit, aby se Dash pohyboval dopředu nebo couval. Ovládat jeho pohyb ale bude Dot pomocí svých tlačítek. Dashův pohyb by se měl měnit tímto způsobem.

- stisk 1. tlačítka – Dash zrychlí
- stisk 2. tlačítka – Dash zpomalí
- stisk 3. tlačítka – Dash se úplně zastaví

Smyslem hry je tedy to, aby Dash jezdil a upravoval svou rychlost na základě pokynů z Dotových tlačítek.

Tuto úlohu řádně otestuj a pokus se zodpovědět tyto otázky:

- Mění se rychlost pomocí stejně velkých kroků?
- Pokud Dash stojí, po stisku 1. tlačítka se znovu rozjede pomalou rychlostí?
- Pokud Dash stojí, po stisku 2. tlačítka začne couvat?

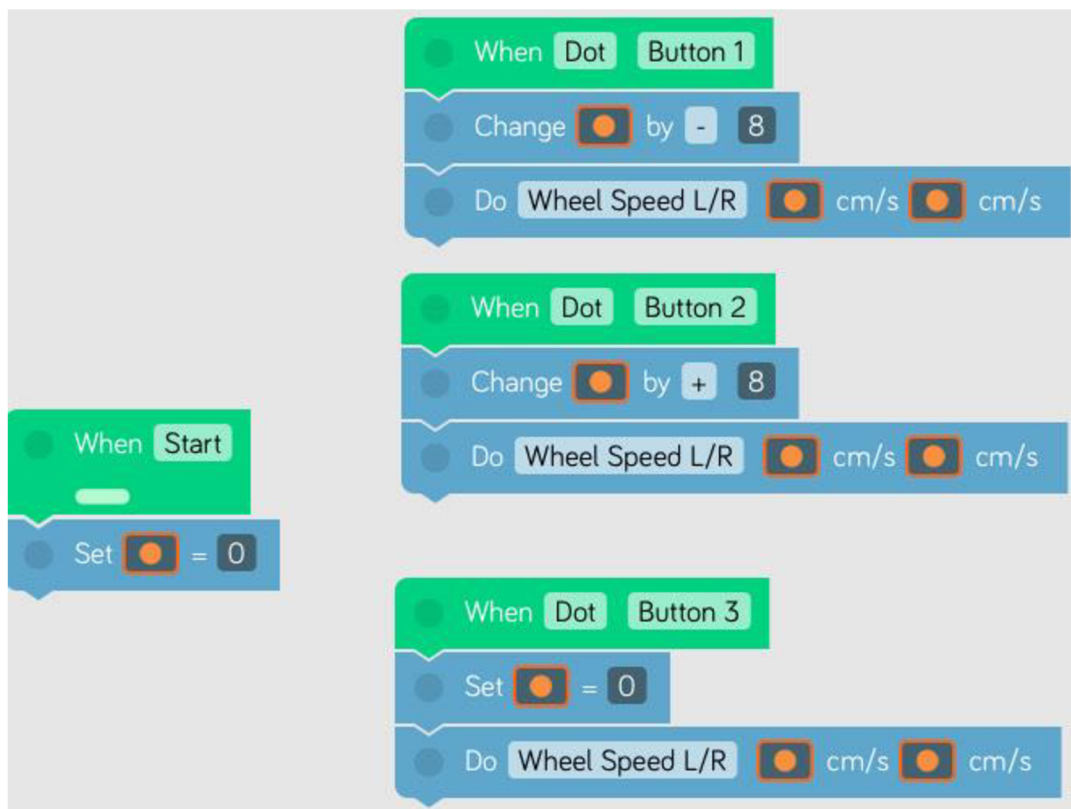
Měl by si být schopný si na všechny otázky zodpovědět pozitivně.

#### Cíle a poznámky k úloze:

Poměrně důležitým cílem úlohy **Zkušený trenér Dot** je u žáků procvičit práci s proměnnými. Jejich úkolem je vytvořit program díky kterému budou po celou dobu programu schopni tlačítka měnit její hodnotu. Konkrétně tak, aby dosáhli požadovaného cíle a měnili Dashovu rychlost. Proto si musí uvědomit co vlastně znamená rychlost v centimetrech za sekundu a že tuto rychlost mohou měnit pomocí čísel. Dále je také důležitá úloha dostatečně odzkoušet a využít správné hodnoty, aby bylo zrychlování co nejefektivnější. Dále se naučí identifikovat a řešit problémy vzniklé během jejich práce. Úloha poskytuje příležitost k praktickému testování a ladění programu v reálném prostředí. Budou rozvíjet schopnost definovat a aktualizovat proměnné podle potřeby a využívat je k uchování nejrůznějších stavů a informací potřebným k vyřešení zadaného problému. Tato úloha také pomocí

otázek u žáků rozvíjí schopnost sebereflexe a díky těmto otázkám si mohou sami zjistit, zda úlohu splnili správně či nikoliv. Následně tyto informace mohou využít k dalším úpravám svého programu.

### Vzorové řešení:



**Obrázek 14** vzorové řešení: Zkušený trenér Dot

Zde budou žáci pracovat čistě s proměnnými. Toto řešení je poměrně časově náročné, protože je důležité aby ho opravdu řádně otestovali. Musí sami vhodně zvolit hodnotu, o kterou budou robota buďto zrychlovat nebo zpomalovat.

## 6.2.4 Předváděčka

### Zadání:

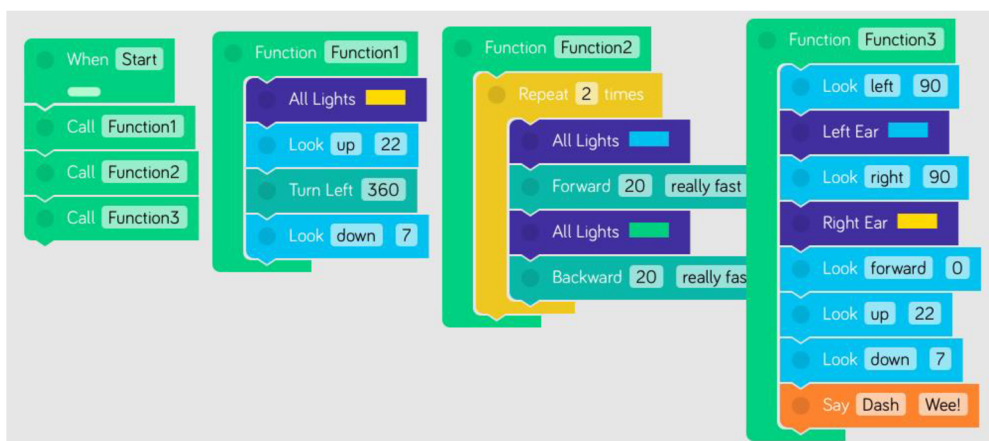
Dash se připravuje na talentové zkoušky a potřebuje oslnit porotu, proto tě poprosil jestli bys mu nepomohl naučit se nějaké nové triky.

Tvým hlavním úkolem bude vytvořit alespoň 3 Dashovy sestavy. Mělo by se jednat o nějaké zajímavé prvky, které si se během svého dosavadního programování naučil. Může jít například o otočky, různé pohyby hlavou nebo třeba zvukové a světelné efekty. Tvorbou sestav je libovolná, jen v každé sestavě použij vždy alespoň 2 z výše zmiňovaných interakcí. Každá reakce by měla být dostatečně originální. Důležité je, abys každou sestavu vytvořil jako funkci a v budoucnu s ní mohl pracovat jako s 1 blokem.

### Cíle a poznámky k úloze:

**Předváděčka** je ze všech doposud zmiňovaných úloh nejvíce zaměřena na rozvoj kreativity u žáků. Jejich úkolem bude vytvořit alespoň 3 Dashovy sestavy, konkrétně by se mělo jednat o různé otočky, práci s LED světly nebo například mohou využít i zvukové efekty. V této úloze mají žáci pouze tyto podmínky a to, aby byly sestavy dostatečně originální a využívaly alespoň 2 ze zmiňovaných interakcí současně. Z tohoto důvodu není hlavním cílem této úlohy naučit žáky nějaké nové programování, ale spíše jim dát prostor vytvořit něco svého díky doposud získaným vědomostem. Tato úloha je nezbytná pro absolvování následující úlohy, proto je také důležité aby žáci své výtvořily vytvořili jako funkce a v příští úloze s nimi mohli pracovat.

### Vzorové řešení:



Obrázek 15 vzorové řešení: Předváděčka



U tohoto řešení se nemusí kontrolovat nic ohledně využití barev nebo například pohybu. Důležité je, aby žáci splnili podmínku a to využít alespoň 2 interakce. Dále je důležité animace vytvářet jako funkce pro zjednodušení práce na další úloze.

## 6.2.5 Dot jako dirigent

### Zadání:

Představ si, že je Dot dirigentem a pomocí svých tlačítek bude spouštět některou z předem naučených sestav, které jsi vytvořil v předchozí úloze.

V této úloze budeš znovu pracovat s proměnnými. Tvým úkolem bude tebou vytvořené reakce použít a spouštět pomocí Dota. Reakce se budou spouštěny tímto způsobem:

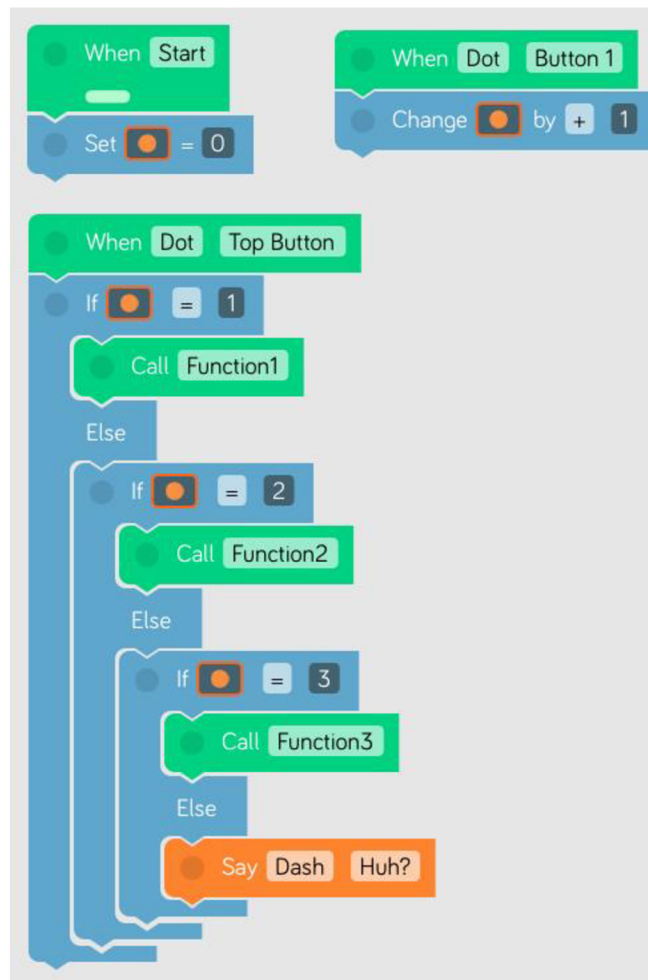
- 1 stisknutí Dotova tlačítka – reakce číslo 1
- 2 stisknutí Dotova tlačítka – reakce číslo 2
- 3 stisknutí Dotova tlačítka – reakce číslo 3

Dále bude potřeba využít ještě jednoho tlačítka, pomocí kterého budeš vybranou sestavu spouštět. Pokud tedy spustíš program, nejdříve naklikáš 1-3x pro výběr sestavy a následně dalším tlačítkem spustíš Dashovu reakci. Pokud bude počet stisknutí jiný než 1, 2 nebo 3, Dash pouze vydá nějaký zvuk a nic nepředvede.

### Cíle a poznámky k úloze:

Co se týče vzdělávacích cílů, tak je úloha **Dot jako Dirigent** velmi podobná zadání Slalomu. Jedná se také hlavně o práci s vnořenými podmínkami a proměnnými. Tentokrát žáci mají ale jimi vytvořené originální sestavy, což by mohlo vést k většímu zájmu ze strany žáků. Jedná se tedy o využití 2 tlačítek, kdy 1. bude jakési počítadlo, díky kterému žáci namačkají požadovaný počet stisknutí a pomocí 2. tlačítka následně reakci spustí. Pozor bychom si měli dát hlavně na to, aby žáci nevytvořili program, který bude každou reakci spouštět jiným tlačítkem. Tímto by nesplnili zadání a hlavně by se vyhnuli využití proměnné. Dále je u této úlohy důležité, aby se nespouštěla žádná z reakcí pokud bude hodnota proměnné jiná než je 1, 2 nebo 3. Toto by žáci měli vyřešit využitím samostatného zvukového efektu, který bude indikovat, že má proměnná jinou hodnotu.

## Vzorové řešení:



Obrázek 16 vzorové řešení: Dot jako dirigent

U tohoto řešení je důležité správně zkombinovat již naučené dovednosti a to konkrétně využití funkcí, rozhodovacích podmínek, nastavování a postupné měnění proměnných. Důležité je vědět, kdy využít blok **If** a kdy blok **When**.

## 6.3 Rozšíření

V následujících úlohách žáci využijí rozšíření. Konkrétně budou pracovat s katapultem, který je součástí robotické sady. Katapult je nutné nejdříve sestavit, proto by bylo vhodné tyto úlohy využívat současně. Všechny 3 úlohy v této části na sebe systematicky navazují a postupně rozvíjí algoritmické myšlení. Žáci si mohou osvojit například využití náhody, opakování, podmínek nebo práci s proměnnými.

### 6.3.1 První výstřely

#### **Zadání:**

Následující úloha se bude zabývat využitím katapultu a vytvářením programů pro jeho ovládání.

Nejdříve je důležité, abys správně zapojil všechny komponenty, které jsou k tomu potřeba. Použij návod a pokud si nejsi jistý, obrať se na učitele. Dále řádně odzkoušej všechny funkce katapultu, zda funguje správně nabíjení a podobně.

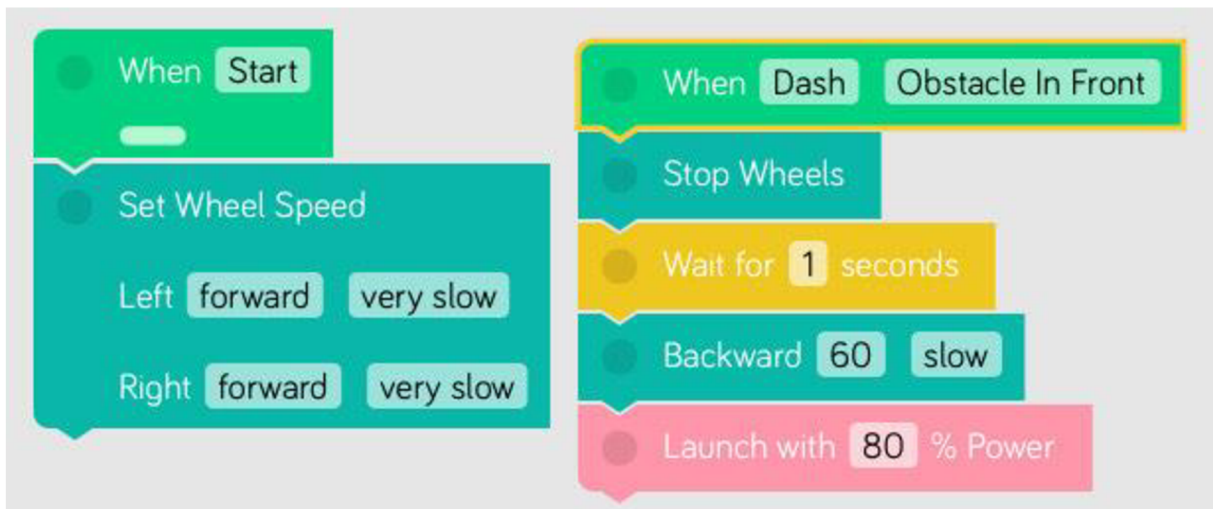
Tvým úkolem bude vytvořit program, díky kterému se Dash bude pohybovat směrem vpřed, dokud před sebou nebude mít nějakou překážku jako je například zeď. Ve chvíli kdy Dash uvidí překážku, zastaví se a couvne o 60 cm. Následně Dash vystřelí z katapultu směrem k překážce. Nabíjení katapultu v této úloze řešit nemusíš, stačí pokud vždy na začátku Dashovi nabiješ katapult ručně.

#### **Cíle a poznámky k úloze:**

Úloha s názvem **První výstřely** je první úloha zaměřující se na využití rozšíření, konkrétně katapultu. Žáci mají za úkol s pomocí učitele správně zapojit všechny důležité části, které jsou nezbytné pro správný chod katapultu. Následně vytvoří program, kde využijí své dosavadní vědomosti ohledně detekování překážek a vyzkouší si samotný katapult.

Tato úloha má několik výukových cílů, konkrétně se zaměřuje na zopakování některých dovedností z minulých úloh a následně na ně navazuje nutností využití nových bloků. U této úlohy a také u všech ostatních úloh je důležité, aby si žáci uvědomili nutnost využití bloku pro čekání, aby vše bezpečně fungovalo. Dále tato úloha pomocí sestavování katapultu rozvíjí motorické schopnosti u žáků.

### Vzorové řešení:



Obrázek 17 vzorové řešení: První výstřely

V tomto případě by měla být věnována zvýšená pozornost tomu, zda žáci správně využívají blok **Wait**, kterým zajistí plynulý chod programu. Dále je také důležité nastavit dostatečnou sílu pro výstřely z katapultu. Záleží pouze na učiteli, zda nechá žáky na problém přijít samotné nebo jim hned na začátku s řešením poradí. Tato úloha se samozřejmě dá řešit i za použití podmínky **If**, ale toto řešení se mi zdá jednodušší a přehlednější.

### 6.3.2 Střelba na cíl

#### **Zadání:**

Samotné střelení máš za sebou, tak ho můžeš zakomponovat do složitějších řešení. Do tohoto úkolu se již zapojí i Dot a využiješ nabíjení katapultu.

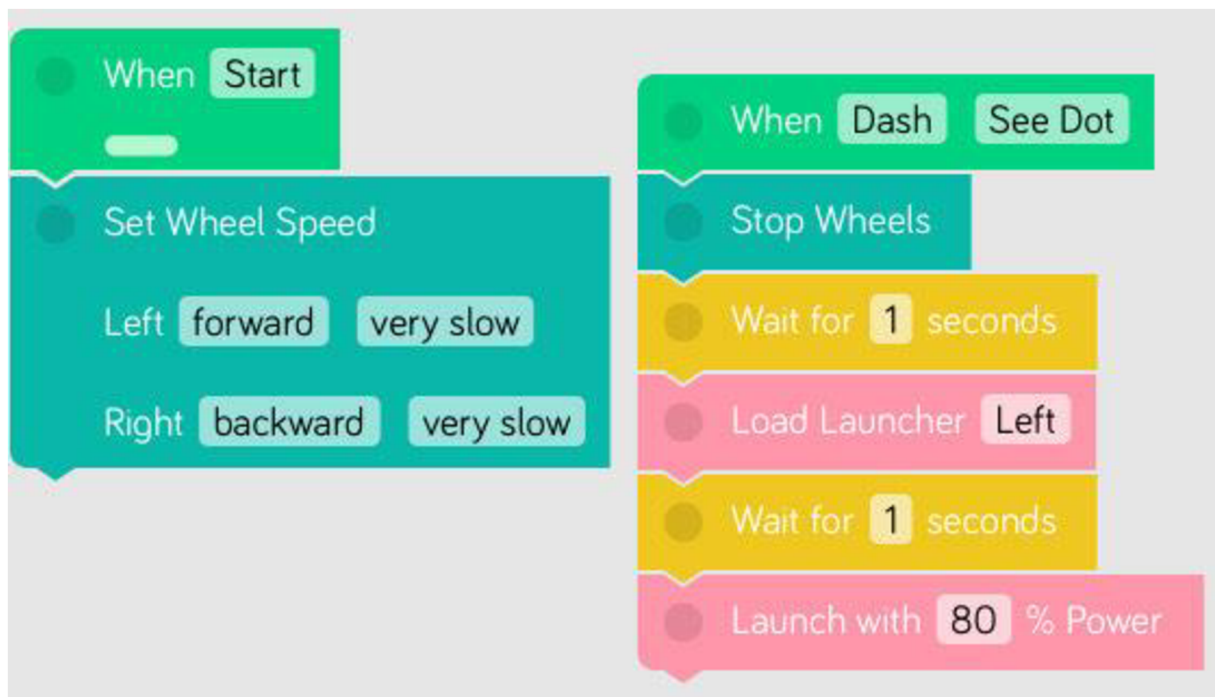
Tvým úkolem bude vytvořit takový program, díky kterému se Dash začne otáčet na jakoukoli stranu. Během chodu programu kdykoliv do prostoru kolem Dashe postav Dota. Ve chvíli, kdy ho Dash uvidí se přestane otáčet. Následně nabije katapult a vystřelí tímto směrem.

#### **Cíle a poznámky k úloze:**

Co se týče úlohy **Střelba na cíl**, tak zde žáci vytvoří takový program, aby se Dash otáčel do chvíle, kdy uvidí Dota. Tato úloha se znovu snaží rozvíjet již získané vědomosti ohledně programování pomocí blokového prostředí, ale přidává další rozšíření a to v podobě nabíjení katapultu. Velmi důležitá část je nastavení samotného otáčení robota. Žáci si musí uvědomit, že mohou robota otáčet i pomocí jiných bloků, než těch určených k otáčení. Konkrétně pokud nastaví jedno kolo na směr dopředu a druhé dozadu. Toto je nezbytné k úspěšnému vyřešení úlohy.

Tato úloha cílí na rozvoj představivosti a kreativity. Což je způsobeno hlavně využitím jiného bloku pro otáčení, protože jsou žáci nuceni přijít s jinou alternativou, než na kterou byli doposud zvyklí. V úlohách není schválně uvedenou jakou sílu by měli žáci pro střelbu použít, což je naučí tomu, aby si dokázali s některými problémy poradit samostatně. Konkrétně je problém v tom, že pokud nastaví menší sílu než je 50 %, tak nedojde k úplnému výstřelu.

### Vzorové řešení:



Obrázek 18 vzorové řešení: Střelba na cíl

U této úlohy musí žáci sami přijít na to, že mohou robota otáčet i jiným způsobem, než bloky k tomu určenými. Konkrétně tedy tak, že nastaví jedno kolo na otáčení vpřed a druhé kolo do druhého směru, čímž zajistí otáčení do nekonečna.

### 6.3.3 Náhodný katapult

#### Zadání:

V této úloze si zopakujeme cykly a využití proměnné. Z Dashe se v tuto chvíli stane náhodný generátor výstřelů.

Tvým úkolem bude, abys pomocí blockly vytvořil takovýto program. Dash se otočí o náhodný počet stupňů do náhodné strany, následně pomocí některého ze zvukových efektů dá najevo, že je připraven. V tuto chvíli mu do jednoho z jeho zásobníků připravíš míček. Poté si Dash musí nabít a vystřelit připravený míček tímto směrem. Program se bude opakovat stále dokola, dokud ho sám nevypneš.

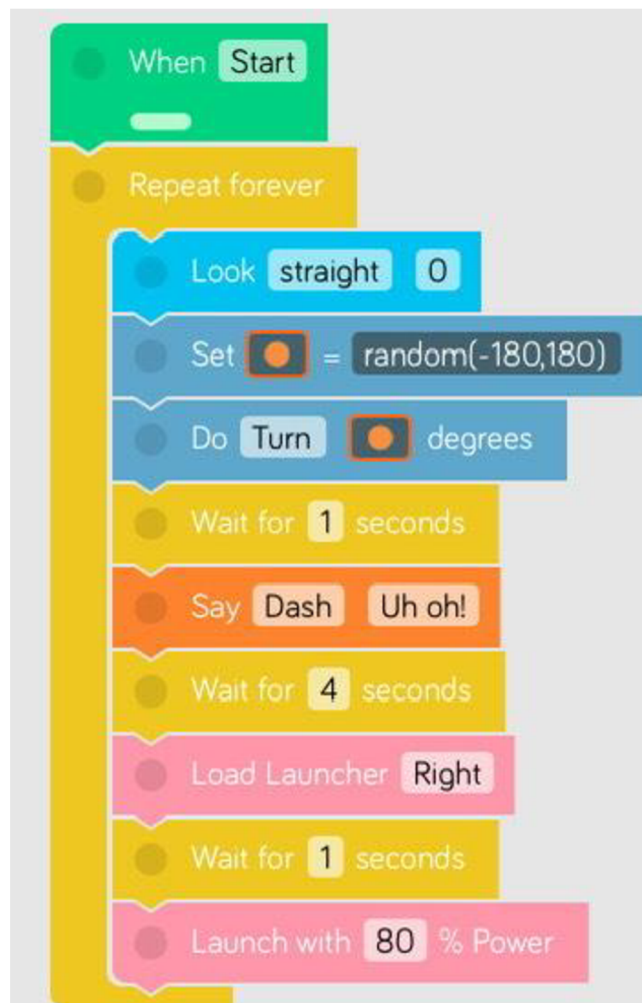
#### Cíle a poznámky k úloze:

**Náhodný katapult** se již podle svého názvu zaměřuje na využití náhodných hodnot. Žáci jsou nuceni robota vždy otočit o náhodný počet stupňů na kteroukoli stranu. Důležité je, aby si uvědomili, že stranu otočení nemusí nikde nastavovat, ale zajistí to správným rozmezím pro náhodnou hodnotu. Konkrétně tak, že v případě záporných hodnot se bude robot otáčet na jinou stranu než v případě hodnot kladných.

Hlavní cíl této úlohy je zopakování několika znalostí. Například o využití cyklu, kdy je v tomto případě nastavit nekonečné opakování tak, aby robot daný pohyb vykonával po celou dobu chodu programu. Dále si žáci zopakují to, že pro správný chod programu musí využít alespoň některá čekání a naposledy si zopakují samotné střelení z katapultu. Na tyto vědomosti následně navazuje využití proměnné a správné dosazení hodnot.



### Vzorové řešení:



Obrázek 19 vzorové řešení: Náhodný katapult

U tohoto řešení se znovu jedná o kombinaci několika již známých bloků. Žáci využijí cykly nebo například proměnné a zakomponují to do funkčního programu ovládajícího Dashův katapult. Jedná se o poslední úlohu, proto kombinuje téměř všechny důležité vědomosti z oboru programování.

## 7 Testování úloh

### 7.1 Úvod

K otestování mnou vytvořené sady úloh jsem se rozhodl využít přímou výuku informatiky na základních školách. Kvůli úspoře času a možnosti otestovat úlohy na co nejvíce skupinách popřípadě třídách jsem se nakonec rozhodl ze všech vytvořených úloh vybrat pouze několik vzorových, které jsem následně využil při testování. Otestoval jsem tedy konkrétně tyto úlohy: **Začínáme programovat**, **Čtverec**, **Pozor!** a **Kapitán Dot**. Z důvodu toho, abych měl co možná nejlepší a nejobjektivnější zpětnou vazbu jsem si testování domluvil na 2 základních školách. Jednalo se o Základní školu Kubatova v Českých Budějovicích, kterou jsem si domluvil díky realizaci své asistentské praxe. Dále jsem úlohy testoval na Základní škole Zborovská v Táboře, kde mi s domluvou pomohl vedoucí mé bakalářské práce. Na obou školách jsem testoval kvůli nedostatku robotů formou skupinové práce po 3 – 4 žácích. U obou škol se jednalo o smíšené třídy 7. nebo 8. ročníků čítajících kolem 15 žáků. Bylo velmi zajímavé, že na obou školách byl poměrně velký rozdíl ve znalostech programování zrovna mezi těmito 2 třídami. Kdy 7. třídy byly oproti 8. v programování velmi napřed.

### 7.2 Průběh

Nejdříve jsem úlohy testoval na základní škole Kubatova v Českých Budějovicích, kde jsem měl k dispozici s každou třídou 1 vyučovací hodinu. Bohužel na této škole neměli k dispozici žádné tablety, na kterých by bylo možné stáhnout požadované blokové prostředí. Nakonec jsme ale s vedoucím učitelem došli k závěru, že necháme žáky pracovat na jejich mobilních telefonech. Stažení aplikace zabralo pár minut, takže to nijak neovlivnilo průběh celé vyučovací hodiny. Zde jsem byl ze zkušenosti velmi příjemně překvapen, protože jsem se obával, aby úloh nebylo na 1 vyučovací hodinu příliš. Nakonec se ale našlo několik skupinek, které zvládly všechny 4 předložené úlohy během hodiny vyřešit. Co se týče anglického jazyka, tak s tím žáci neměli vůbec žádný problém. Naopak bych řekl, že se tím pro ně úlohy staly trochu zajímavější. S panem učitelem jsme se následně shodli, že bylo využití anglického jazyka naopak přínosem, protože žáci tolik nevnímali podobnost se scratchem, se

kterým měli podle jeho slov v minulosti poměrně velké problémy. V samotných zadáních jsem žádný problém nevypozoroval, žáci vždy vše bez problému pochopili a ani se na nic ze zadání nedoptávali. Věc, která dělala žákům největší problém bylo bezpochyby rozlišení zda mají použít pro pohyb blok, kde nastaví požadovanou ujetou vzdálenost nebo pouze nastavit směr a rychlost pohybu. Vždy jim stačila pouze lehká nápověda a vysvětlení rozdílu mezi těmito bloky, aby nakonec došli k požadovanému řešení. Mezi těmito třídami byl velmi znatelný rozdíl co se týče zkušenostmi s programováním, kdy žáci ze 7. třídy byli mnohem rychlejší a šikovnější než starší žáci z 8. třídy. Toto mi ale nakonec pan učitel vysvětlil tím, že se měnil vzdělávací plán a proto žáci 7. třídy již mají programování ve scratchi za sebou a žáci 8. třídy nikoliv. I přes jejich dosavadní neznalost programování žáky úlohy poměrně bavily. Z tohoto bylo patrné, že úlohy lze řešit i bez předchozí znalosti programování pouze díky testování a zkoušení nových zatím neznámých bloků.

Jak jsem již dříve zmiňoval, úlohy jsem testoval i na základní škole v Táboře, kde jsem měl tentokrát k dispozici celkem 4 třídy. Znovu se jednalo o 7. a 8. ročníky. Co se týče práce s úlohami, tak žáci byli na podobné úrovni jako na předešlé škole. Bylo zde ale několik chlapců, kteří byli velmi šikovní. Například jsem byl velmi příjemně překvapen, když bez mého vstupu automaticky používali v programu opakování. U některých úloh dokonce přišli s jiným řešením, než jsem já uvedl jako vzorové. Během pozorování žákovské práce jsem si všiml, co je na programování bavilo nejvíce a jednalo se hlavně o využívání barev a zvuků, takže by určitě bylo vhodné tyto části do úloh v budoucnu více zakomponovat. Dále jsem byl poměrně překvapen, že žáci měli problém u otáčení robota určovat o kolik stupňů se má otočit. Pletli si hlavně rozdíl mezi 180 a 90 stupni, což je ale problematika spíše z oboru matematiky.

Během testování jsem narazil také na několik nových problémů jak se samotnými roboty tak s blokovým prostředím. V blokovém prostředí nelze naprogramovat to, aby se Dash točil a ve stejnou chvíli například také ovládal světla nebo vydával nějaký zvuk. S roboty byl poměrně zásadní problém ve výdrži baterie. Všechny sady jsem před prací plně nabil. Při prvním testování to bylo bez problému. Ve chvíli, kdy jsem ale pracoval se 4 skupinami žáků po sobě, tak u poslední skupiny žáků již došlo k vybití několika robotů, což poměrně znesnadňovalo práci. K tomuto problému

naštěstí docházelo až v pozdějších minutách vyučovací hodiny, takže samotnou výuku to tolik nenarušilo, ale ve chvíli kdy by se mělo jednat například o celý vyučovací den, tak by baterie nevydržela.

Během testování jsem také přišel na několik zajímavých zjištění. Skupiny byly většinou smíšené, ale ve chvíli kdy se měly děti rozdělit do skupin, tak byly vždy vytvořeny buďto skupiny samých dívek nebo samých chlapců. Na základě mého pozorování mohu říci, že chlapci byli vždy do práce s roboty mnohem více zapálení než dívky. Bylo vidět, že je práce více baví a díky tomu jim plnění úkolů šlo mnohem rychleji. Dívky byly většinou lehce pomalejší, ale zase pokud požádaly o nějakou radu, tak vždy na správné řešení přišly.

Na druhou stranu chlapci hodně často vytvářeli úplně jiné programy než měli v zadání, takže jsem je musel neustále obcházet a kontrolovat zda pracují. Dívky většinou opravdu řešili zadanou úlohu.

### **7.3 Provedené změny**

Na základě zkušeností, které jsem svým testováním získal jsem se rozhodl některé úlohy ještě trochu upravit. Nejednalo se ale o žádné velké změny, většinou jsem pouze lehce pozměnil zadání nebo ho trochu zkrátil. Všiml jsem si například nesrovnalosti v úloze čtverec, kdy jsem po žácích v první verzi zadání chtěl čtverec nakreslit, ale vlastně šlo pouze o to, aby se robot do čtverce pohyboval. Musel jsem tedy pozměnit znění zadání z nakreslí čtverec na zajisti, aby se robot pohyboval do čtverce. Dále například u úlohy Kapitán Dot jsem měl podmínku pokud je Dot hlavou dolů, což bylo trochu nejednoznačné a žáci nevěděli jak přesně jsem to myslel. Musel jsem tedy zadání přepsat tak, aby bylo jasné, že jsem myslel chvíli, kdy má robot obličej položen na podložce. U některých úloh jsem trochu pozměnil i jejich název, aby byly názvy zejména pro žáky zábavnější a zajímavější. Takovéto nepřesnosti a samozřejmě ještě různé interpunkční nebo gramatické chyby jsem nakonec na základě zpětné vazby od žáků a učitelů ještě opravil nebo změnil.

## 8 Závěr

Během své práce jsem vytvořil sadu úloh vhodnou pro rozvoj algoritmického myšlení u žáků 2. stupně základní školy. K práci je potřeba pouze dvojice robotů a vhodné prostředí, například prostředí školní třídy. U některých úloh je potřeba nějakých předmětů, které se ale vždy v prostředí školy nacházejí. Celkem se jedná o 14 úloh, které jsou systematicky rozděleny do 3 kategorií. 1. kategorie s názvem Pohybové obsahuje úlohy založené na pohybu Dashe. Další kategorie má název spolupráce robotů, kde žáci již pracují s oběma roboty a využívají různé interaktivní vstupy. Poslední kategorie nese název rozšíření. Zde si žáci mohou vyzkoušet práci s katapultem.

Dále jsem ve své práci popsal výukové cíle v oblasti informatiky, abych během tvorby vycházel z konkrétních výukových cílů. U každé úlohy je tedy shrnuto jakým aspektům informatiky a programování se věnuje. Zaměřil jsem se také na didaktiku informatiky abych se inspiroval a zjistil co by měl žák na úrovni 2. stupně základní školy v oboru programování ovládat.

V další části mé práce jsem se zaměřil na myšlení, která jsou pro mou sadu úloh klíčová. Konkrétně jsem podrobně popsal jak myšlení informatické tak myšlení algoritmické a obě vzájemně porovnal.

Další důležitou částí mé práce je detailní popsání dvojice interaktivních robotů Dash a Dot.

Zaměřil jsem se také na popis vybraného blokového prostředí Blockly. Popsal jsem všechny jeho části a také jsem vytvořil text o jeho nevýhodách, na které jsem v průběhu své práce narazil.

Našel jsem výzkum zabývající se postojem žáků k výuce robotiky, kterým jsem podpořil mnou vytvořenou sadu úloh.

Úlohy jsem nakonec otestoval na 2 základních školách. Konkrétně na základní škole Kubatova a na základní škole Zborovská. Na základě zkušeností a vědomostí získaných při testování úloh a také díky možnosti obhajovat na nečisto jsem úlohy následně ještě lehce upravil.

## 9 Seznam použité literatury

- [1] VALIŠOVÁ, Alena a KOVAŘÍKOVÁ, Miroslava. Obecná didaktika a její širší pedagogické souvislosti v úkolech a cvičeních. Pedagogika (Grada). Praha: Grada, 2021. ISBN 978-80-271-3249-2.
- [2] PAVLÍČEK, Antonín; GALBA, Alexander a HORA, Michal. Moderní informatika. Druhé, rozšířené vydání. Praha: Professional Publishing, 2017. ISBN 978-80-906594-6-9.
- [3] VANÍČEK, J., & M. ČERNOCHOVÁ. Didaktika informatiky na startu. In: STUHLÍKOVÁ, Iva; JANÍK, Tomáš; BENEŠ, Zdeněk; BÍLEK, Martin; BRÜCKNEROVÁ, Karla et al. Oborové didaktiky: vývoj, stav, perspektivy. Syntézy výzkumu vzdělávání. Brno: Masarykova univerzita, 2015. ISBN 978-80-210-7769-0.
- [4] MŠMT. Rámcový vzdělávací program pro základní vzdělávání. Online. 2023. Dostupné z: <https://www.edu.cz/rvp-ramcove-vzdelavaci-programy/ramcovy-vzdelavacici-program-pro-zakladni-vzdelavani-rvp-zv/>. [cit. 2024-04-11].
- [5] LESSNER, Daniel. Učíme informatiku. Online. 2014. Dostupné z: <https://ucime-informatiku.blogspot.com/2014/09/informaticke-mysleni-2-ruzna-vymezeni.html>. [cit. 2024-04-11].
- [6] WING, Jeannette. Computational thinking. Online. 2006. Dostupné z: <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>. [cit. 2024-04-11]. Začátek formuláře
- [7] JIHOČESKÁ UNIVERZITA. Informatické myšlení. Co je informatické myšlení. Online. 2018 [cit. 2024-04-11]. Dostupné z: <https://www.imysleni.cz/informaticke-mysleni/co-je-informaticke-mysleni>
- [8] SELBY, Cynthia C. a John WOOLLARD. Refining an Understanding of Computational Thinking. Southampton: University of Southampton. Online. 2014. Dostupné z: <https://eprints.soton.ac.uk/372410/>. [cit. 2021-03-22].
- [9] ZVĚŘINA, Lukáš. Sada úloh pro výuku algoritmizace na základních školách. Diplomová práce. Plzeň: Západočeská univerzita, Fakulta pedagogická, Katedra výpočetní a didaktické techniky, 2020.

- [10] PŠENČÍKOVÁ, Jana. Algoritmizace. Vydání druhé. Prostějov: Computer Media, 2021. ISBN 978-80-7402-414-6.
- [11] ANGELI, Charoula; DIAKOU, Panayiota a ANASTASIOU, Vaso. THE EFFECTS OF AGE AND LEARNING WITH EDUCATIONAL ROBOTIC DEVICES ON CHILDREN'S ALGORITHMIC THINKING. Online. 2023. Dostupné z: <https://eric.ed.gov/?id=ED636331>. [cit. 2024-04-11].
- [12] SHARMA, Kamyia. Dash and Dot. Cherry Lake Pub, 2017. ISBN 1634727193.
- [13] WONDER WORKSHOP. Makewonder. Online. Get started with Dash. 2023. Dostupné z: <https://www.makewonder.com/getting-started/dash/>. [cit. 2024-04-11].
- [14] WONDER WORKSHOP. Makewonder. Online. Blockly. 2024. Dostupné z: <https://home.makewonder.com/apps/blockly>. [cit. 2024-04-11].
- [15] SMYRNOVA-TRYBULSKA, Eugenia; MORZE, Natalia a ZUZIAK, Wojciech. Robots in elementary school : some educational, legal and technical aspects. Online. 2016. Dostupné z: [https://rebus.us.edu.pl/bitstream/20.500.12128/7888/1/Smyrnova\\_Trybulska\\_Robots\\_in\\_elementary\\_school.pdf](https://rebus.us.edu.pl/bitstream/20.500.12128/7888/1/Smyrnova_Trybulska_Robots_in_elementary_school.pdf). [cit. 2024-04-11].
- [16] ČERNÝ, Michal. Výukoví roboti: nástroj pro rozvoj algoritmického myšlení. Online. 2015. Dostupné z: <https://clanky.rvp.cz/clanek/c/g/19905/VYUKOVI-ROBOTI-NASTROJ-PRO-ROZVOJ-ALGORITMICKEHO-MYSLENI.html>. [cit. 2024-04-11].

## 10 Seznam obrázků

Obrázek 1 nastavování rychlosti .....	20
Obrázek 2 natáčení hlavy .....	20
Obrázek 3 ovládání světel .....	21
Obrázek 4 ukázka správného připojení robotů .....	29
Obrázek 5 vzorové řešení: První krůčky .....	31
Obrázek 6 vzorové řešení: Čtverec .....	32
Obrázek 7 vzorové řešení: Pozor! .....	34
Obrázek 8 vzorové řešení: Hra na schovávanou .....	37
Obrázek 9 vzorové řešení: Hra na schovávanou - rozšíření .....	38
Obrázek 10 vzorové řešení: Slalom.....	40
Obrázek 11 vzorové řešení: Slalom - rozšíření .....	41
Obrázek 12 vzorové řešení: Reagujeme na Dota .....	43
Obrázek 13 vzorové řešení: Kapitán Dot .....	45
Obrázek 14 vzorové řešení: Zkušený trenér Dot.....	47
Obrázek 15 vzorové řešení: Předváděčka .....	48
Obrázek 16 vzorové řešení: Dot jako dirigent.....	51
Obrázek 17 vzorové řešení: První výstřely .....	53
Obrázek 18 vzorové řešení: Střelba na cíl.....	55
Obrázek 19 vzorové řešení: Náhodný katapult .....	57