

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č.12: tvorba učebních pomůcek, didaktická technologie

ALGORITMIZACE S MICRO:BIT

ALGORITHMIZATION WITH MICRO:BIT

Autoři: Jan Smutný, Michaela Hübnerová

Škola: Střední průmyslová škola strojní a elektrotechnická a Vyšší odborná škola, Liberec 1, Masarykova 3, příspěvková organizace; Masarykova 3, 460 01 Liberec 1

Kraj: Liberecký kraj

Konzultant: Ing. Tomáš KAZDA, DiS.

Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupnění této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Liberci dne 15. 3. 2022

.....

Jan Smutný, Michaela Hübnerová

Liberec 2022

Anotace

Práce se zabývá vytvořením úloh z algoritmizace pro žáky základních škol prvního a druhého stupně. Obsahuje jak naprogramované pomocné bloky, tak didaktickou část obsahující metodické listy pro učitele a pracovní listy pro žáky. Práce obsahuje 18 úloh, které rozšíří možnosti informatiky podle nového RVP.

Cíle úloh vychází z revize rámcového vzdělávacího programu, který se od roku 2023 musí aplikovat na všech základních školách.

Přináší další materiály a podklady pro lepší a zábavnější výuku informatiky na základních školách.

Klíčová slova

Micro:bit; dekompozice problému; grafický zápis algoritmu; vzdělávání

Annotation

The work deals with creating programming assignments for primary school students. It contains both pre-programmed blocks of code and a didactic part containing methodical sheets for teachers and worksheets for students. The work altogether consists of 18 exercises, thanks to which there will be a better possibility of teaching computer science according to the new FEP.

It is based on the revision of the framework educational program, which must be applied to all primary schools from 2023.

It brings more materials for better and more engaging teaching of computer science in primary schools.

Keywords

Micro:bit; decomposition; graphical notation of the algorithm; education

Obsah

Úvod.....	1
1 Nový rámcově vzdělávací program.....	2
1.1 Reforma informatiky v RVP.....	2
1.1 iMyšlení.....	3
2 Seznámení s Micro:bitem.....	4
2.1 Hardware.....	4
2.2 Software.....	5
2.3 Vytváření knihoven (rozšíření).....	6
3 Definování možných cílů dle klíčových kompetencí.....	8
3.1 Caesarova šifra (1/2).....	8
3.1.1 Úloha pro 1. stupeň.....	9
3.1.2 Úloha pro 2. stupeň.....	10
3.2 Caesarova šifra (2/2).....	10
3.2.1 Úloha pro 1. stupeň.....	12
3.2.2 Úloha pro 2. stupeň.....	12
3.3 LED mozaika.....	13
3.3.1 Úloha pro 1. stupeň.....	13
3.3.2 Úloha pro 2. stupeň.....	14
3.4 Světelná závora.....	15
3.4.1 Úloha pro 1. stupeň.....	16
3.4.2 Úloha pro 2. stupeň.....	17
3.5 Poplach, poplach!.....	17
3.5.1 Úloha pro 1. stupeň.....	19
3.5.2 Úloha pro 2. stupeň.....	20
3.6 Zeptáme se publika 1/2.....	20
3.6.1 Úloha pro 1. stupeň.....	21
3.6.2 Úloha pro 2. stupeň.....	22
3.7 Zeptáme se publika 2/2.....	22

3.7.1	Úloha pro 1. stupeň	22
3.7.2	Úloha pro 2. stupeň	23
3.8	Pozor, vejce, nespadni!.....	25
3.8.1	Úloha pro 1. stupeň	26
3.8.2	Úloha pro 2. stupeň	26
3.9	Tik tok bác!	27
3.9.1	Úloha pro 2. stupeň (lehčí varianta)	28
3.9.2	Úloha pro 2. stupeň (těžší varianta)	28
4	Návrh metodických a pracovních listů	30
4.1	Metodické listy	30
4.2	Pracovní listy	31
5	Zpětná vazba od žáků	33
	Závěr (Jan Smutný)	34
	Závěr (Michaela Hübnerová)	35
	Seznam zkratk a odborných výrazů.....	36
	Seznam obrázků.....	38
	Použité zdroje	39
A.	Seznam příložených souborů	I

Úvod

Celá práce je o návrhu a zpracování úloh z programování pro žáky základních škol tak, aby vyhovovaly novému Rámcově vzdělávacímu programu. Úlohy jsou realizovány za pomoci mikropočítače Micro:bit, který je pro tyto účely vhodný.

Pro zhotovení celé práce bylo potřeba seznámit se se změnami v Rámcově vzdělávacím programu, vybrat cílové kompetence, podle nich navrhnout a poté konzultantem nechat schválit vhodná zadání. Z nich vypracovat metodické listy pro učitele a pracovní listy pro žáky, naprogramovat knihovny pro každou úlohu a vymyslet vzorové řešení každé úlohy. Všechny potřebné materiály jsou k dispozici na URL: <https://github.com/microbit-cz/>

Téma této práce jsme si vybrali, protože rezonuje v pedagogických kruzích, a může být prakticky použitelná pro učitele na ZŠ. Má také potenciál zvýšit atraktivitu výuky informatiky. Dalším důvodem je, že jsme v tomto projektu viděli „vyšší“ smysl. Líbila se nám myšlenka, že náš projekt dopomůže k výuce programování na základních školách a třeba někoho namotivuje k pokračování ve studiu programování.

1 Nový rámcově vzdělávací program

Příležitost k vytvoření této práce vznikla na základě nově zrevidovaného Rámcově vzdělávacího programu pro základní vzdělávání vydaným Ministerstvem školství, mládeže a tělovýchovy v lednu 2021. Změny v RVP si kladly za cíl pokročit hlavně v oboru informatiky, jejíž výuka byla vedena podle starého RVP. Mělo by tedy dojít k rozvoji digitální gramotnosti žáků základních škol. Od 1. září 2021 je školám umožněno upravit své školní vzdělávací programy podle nového RVP, nejzazší termín pro výuku podle nového RVP je 1. září 2023 pro všechny ročníky 1. stupně a 1. září 2024 pro všechny ročníky 2. stupně základních škol. (1) (2)

Výuka podle nového pojetí informatiky dle RVP se dá realizovat mimo jiné i za pomoci Micro:bitu. Na internetu jsou již ke stažení zdarma dostupné učebnice nebo pracovní listy, avšak s konzultantem práce jsme se shodli na tom, že úlohy by mohly být rozmanitější, a právě z tohoto důvodu v úlohách cílíme na praktičnost, zábavu a propojení s jinými předměty. Rádi bychom tímto docílili rozšíření možností, které učitelé budou mít (nebo již mají) k dispozici. (3)

1.1 Reforma informatiky v RVP

V novém RVP se informatika značně rozšířila. Vzdělávací obsah je rozdělen na učivo prvního a druhého stupně. U každého je pak látka rozvrhnutá do čtyř sekcí.

- Data, informace a modelování
- Algoritmizace a programování
- Informační systémy
- Digitální technologie

V samotné sekci jsou pak rozepsány očekávané výstupy, aneb to, co by žák po dokončení výuky měl umět. Z těchto cílů jsme vybrali ty nejdůležitější a nejlépe proveditelné ve spojitosti se zbytkem projektu. Přesné znění výstupu je vždy uvedeno u každé úlohy ve třetí kapitole tohoto dokumentu.

V rámci našich zadání jsme nejvíce čerpali ze sekce *Algoritmizace a programování*, jež je zaměřena na vytváření kódu, opravování chyb v něm, ověřování správnosti nebo vymýšlení postupů (algoritmů).

Vybrané výstupy jsme pak propojili pomocí ŠVP ze stránek *imysleni.cz* s určitým ročníkem, ve kterém by konečná úloha mohla být vyučována.

1.1 iMyšlení

Webová stránka *imysleni.cz* byla vytvořena Jihočeskou univerzitou v Českých Budějovicích za účelem informování dětí, rodičů, pedagogů a další širší publikum o rozvoji a vůbec smyslu informatického myšlení. Na jejich webu můžeme kromě seznámení se smyslem zapojování informatického myšlení do výuky najít také různé články, rozhovory a novinky.

Nabízí i online kurzy pro lektory a učitele, učebnice a výukové materiály na témata, jako je programovací jazyk Scratch, Python, základy informatiky a i samotný Micro:bit s MakeCode nebo Pythonem. Tyto materiály byly vytvořeny v rámci projektu PRIM, do kterého se zapojilo nespočet pedagogů základních, středních a vysokých škol. Po jejich dokončení byly více než dva roky testovány na desítkách žáků.

Co nás ale nejvíc zajímalo bylo to, že vytvořili školní vzdělávací program (ŠVP), kde podle nového RVP navrhli, jakou látku a témata zařadit do jakého ročníku. Tuto část jsme poté využili, když jsme rozhodovali, pro kterou třídu každou úlohu určit. Nutno ale podotknout, že jsme se od dokumentu pouze odráželi, jelikož samotná výuka s Micro:bitem je v něm uvedena až od 8. ročníku, což nám pro naše účely přišlo pozdě. Vzali jsme tedy některé výstupy a upravili je, aby seděly více přímo na Micro:bit s MakeCode.

2 Seznámení s Micro:bitem

Micro:bit je programovatelný mikropočítač od společnosti BBC, který byl původně navržen pro výuku programování ve Velké Británii. Svou velikostí nepřesáhne platební kartu (43×52 mm). Poprvé byl představen během kampaně BBC „Make It Digital“ 12. března 2015. Micro:bit je vhodným mikropočítačem pro vzdělávání, a proto ho k výuce programování podle nového RVP lze použít i u nás v České republice. (4) (5)

2.1 Hardware

Doposud existují dvě verze Micro:bitu. První verze, které se říká „v1.5“ a druhá verze, vydaná 13. října 2020, která nese název „v2“. Já jsem po celou dobu práce používal druhou verzi, která má všechny funkce první verze, a navíc je obohacena dalšími funkcemi.

Verze 2 obsahuje:

- 5×5 LED matici
- Akcelerometr
- Kompas
- Snímání teploty
- Senzor intenzity světla
- Dvě tlačítka
- Dotykové logo
- Tři dotykové piny (pokud se tak nakonfigurují)
- Bluetooth LE 5.0 nebo 2,4Ghz Micro:bit Radio
- Mikrofon
- Reproduktor

První verze neobsahuje Bluetooth verzi 5.0, ale 4.0, dále nedisponuje mikrofonem, reproduktorem a dotykovým logem. Druhá verze se také může chlubit i celkově lepším hardwarem.

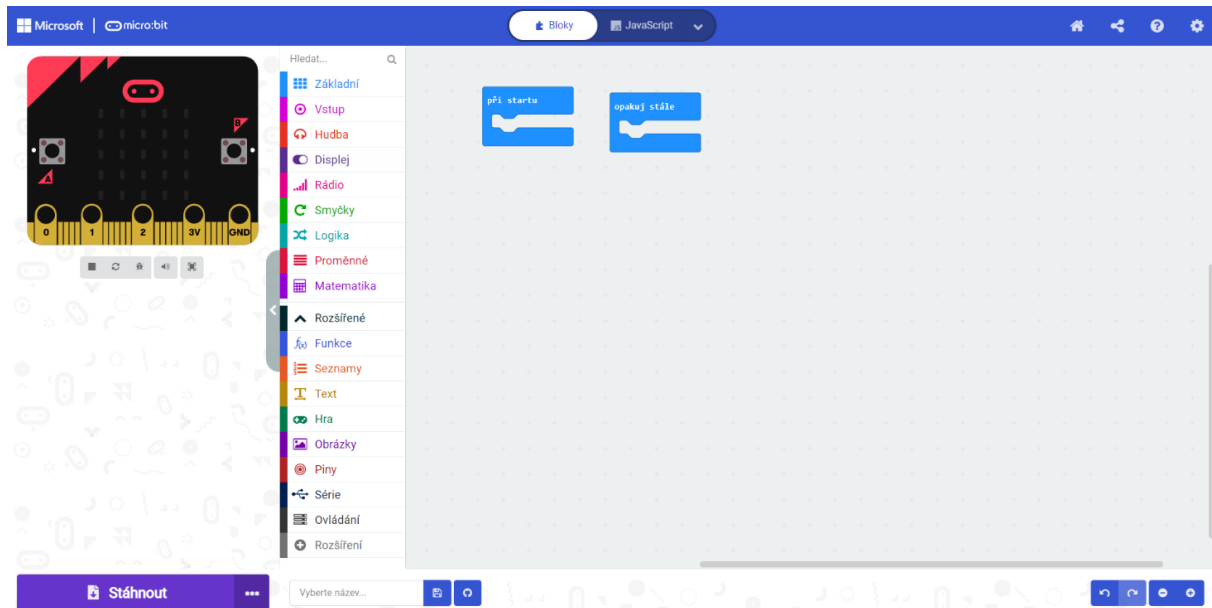
Micro:bit se dá programovat ve třech editorech: MakeCode, MicroPython a Scratch. Já jsem si vybral editor MakeCode a to hlavně kvůli možnosti využít emulátor, který stránka nabízí, jednoduchému debugování, webovému rozhraní a přímému stahování programu do Micro:bitu. (6) (4)

2.2 Software

Micro:bit se dá programovat vícero způsoby. Žáci na základních školách budou podle RVP programovat v tzv. „blokovém kódu“. Další dvě možnosti jsou programovací jazyky Javascript a Python, což pro mě jsou bližší varianty. Pro účel práce byl zvolen Javascript, protože se v něm dají dělat rozšíření pro Micro:bit, které v Pythonu dělat nelze. MakeCode ale všechny tři možnosti mezi sebou překládá. Je tedy možné chvíli programovat v blocích, chvíli v Javascriptu, chvíli v Pythonu a vždy navázat na to, co bylo napsáno v přechozím jazyce. MakeCode také plně zvládá práci s GitHubem. Po přihlášení do GitHubu zvládne vytvořit repositář, naklonovat ho, aplikovat změny a přijímat je.

Takto komplexní možnosti samozřejmě nejsou určeny pro žáka základní školy, zde se předpokládá používání blokového návrhu programu, stejně jako tomu je ve známé platformě Scratch. Pro účely jednotlivých úloh navržené bloky byly vytvořeny ve zmíněném jazyce JavaScript.

Po naprogramování kódu stačí připojit Micro:bit k zařízení přes USB kabel (USB na straně PC/notebooku, MicroUSB na straně Micro:bitu) a ve webovém rozhraní MakeCode přenést vytvořený program stisknutím jediného tlačítka do Micro:bitu.



Obrázek 1 Prostředí MakeCode (7)

Základní dva bloky, se kterými se žáci setkají, jsou bloky „při startu“ a „opakuji stále“. Z názvu vyplývá, že blok „při startu“ slouží k provedení něčeho, co se má stát jen jednou při spuštění programu a víckrát již ne. Smyčka „opakuji stále“ slouží k provedení instrukcí, které se mají provádět opakovaně. (7)

2.3 Vytváření knihoven (rozšíření)

V prostředí MakeCode je možné přidat do projektu další soubory. Stačí tedy přidat soubor s názvem vytvářené knihovny (který se nutně nemusí shodovat s tím, co žáci uvidí v makecode), vytvořit pro ni unikátní namespace a můžeme začít psát kód. Na další straně je vidět příklad, jak může knihovna vypadat. Všechny možnosti definovaných bloků jsou vkládány ve formě specificky formátovaného komentáře jazyka JavaScript. Na úvod je možné si nadefinovat například jakou prioritu bude knihovna mít (jak bude řazena v seznamu možných příkazů prostředí Makecode; vlastnost „weight“), jakou barvu bude mít (vlastnost „color“), jakou ikonu bude mít („icon“) nebo to, jak se bude knihovna ve výsledku jmenovat (vlastnost „block“). Knihovně přiřazujeme jmenný prostor proto, aby její kód byl izolovaný od programu uživatele i dalších připojených knihoven (rozšíření). Na příkladu jsou vidět některé možnosti, které lze u knihovny nadefinovat. Na příkladu je funkce „calibrate“, u které je přidán popis funkce („Spustí kalibraci a nastaví toleranci“) a také to, jak bude blok funkce vypadat („block="Zkalibruj a nastav toleranci %tol"). Dále je možné mimo jiné nastavit omezení hodnot parametru. Zde je například parametr „tol“ omezen na čísla od 0 do 255.

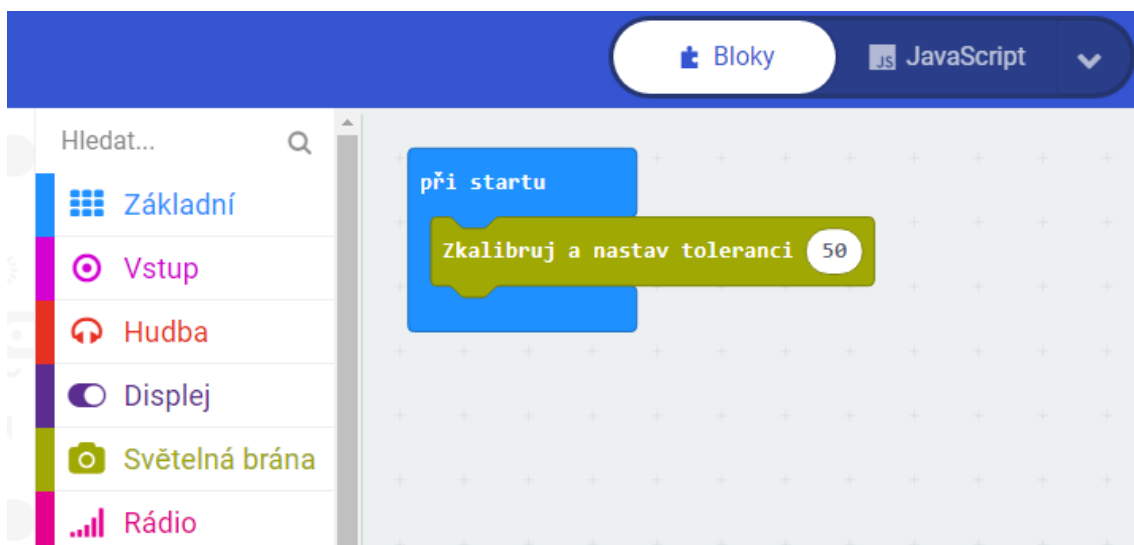
```

//% weight=100 color=#a0a803 icon="\uf030" block="Světelná brána"
namespace lightGate {
  let toleration = 0
  let lightLevel = 0
  let methodLock = false
  /**
   * Spustí kalibraci a nastaví toleranci
   * @tol Tolerance hladiny světla (0-255)
   */
  //% block="Zkalibruj a nastav toleranci %tol"
  //% tol.min=0 tol.max=255

  export function calibrate(tol: number): void {
    let sumOfMeasurements = 0;
    for (let i = 0; i < 10; i++) {
      sumOfMeasurements += input.lightLevel()
    }
    lightLevel = Math.round(sumOfMeasurements / 10)
    toleration = tol;
  }
}

```

Funkce v prostředí MakeCode poté může vypadat třeba takto:



Obrázek 2 Příklad bloku z knihovny (7)

3 Definování možných cílů dle klíčových kompetencí

Po zvolení vhodných výstupů z RVP a ŠVP následovalo vymyšlení samotných úloh. Společně jsme dali dohromady kolem 30 možných úloh, ze kterých jsme po poradě s konzultantem vybrali devět finálních:

- Caesarova šifra 1/2
- Caesarova šifra 2/2
- LED mozaika
- Světelná závora
- Poplach, poplach!
- Zeptáme se publika 1/2
- Zeptáme se publika 2/2
- Pozor, vejce, nespadni!
- Tik tok bác!

3.1 Caesarova šifra (1/2)

První část úlohy Caesarova šifra je zaměřena na její základní princip. Jelikož se v RVP objevuje celá sekce, kde se řeší šifry, rozhodli jsme se vybrat jednu z těch známějších a vytvořit úlohu, ve které si žáci zkusí, jak šifrování a dešifrování textu funguje.

Dělí se na dvě části (nejedná se o části pro první a druhý stupeň). Žáci by měli po zhotovení a vyzkoušení pochopit význam šifrování (v tomto případě jde o šifrování symetrické), což splňuje jeden z výstupů RVP. Stěžejní pro úlohu je, aby žáci pochopili, jak Caesarova šifra funguje, není to ale nic složitého. Šifra spočívá v tom, že si nejdříve zvolíme nějaký klíč, který bude reprezentovat posun písmen. Pak postačí každé písmeno o daný klíč posunout v abecedě a je zašifrováno. Dešifrování probíhá tak, že se posun od písmena odečte.

Takto by se dala poměrně rychle vysvětlit Caesarova šifra. Samozřejmě ale nemusíme šifrovat pouze písmena, ale i čísla a jiné znaky. Domníváme se, že pro žáky bude zprvu jednodušší na pochopení, když budou šifrovat jen text a nikoli čísla nebo jiné znaky. Ale Javascript samozřejmě nabízí možnost konverze znaku na číslo z ASCII (znaková sada) tabulky, takže je tu i tato možnost. V rozšíření ale děláme to, že jiné znaky, než písmena nešifrujeme/nedešifrujeme a necháváme je být tak, jak jsou.

Dalším úskalím je to, že Micro:bit neumí spoustu znaků zobrazit (největší omezení dle mého představuje nemožnost zobrazení diakritiky) a pokud na nějaký takový znak narazí, tak jednoduše nechá LEDky zhasnuté. Proto v rozšíření pracujeme pouze s anglickou abecedou, se znaky A B C D E F G H I J K L M N O P Q R S T U V W X Y Z. Nestane se tedy, že by rozšíření „vyprodukovalo“ znak s diakritikou. Pokud ale rozšíření takový znak dostane, tak ho nejdříve převede na znak bez diakritiky.

Na úlohu není potřeba nic jiného, než samotný Micro:bit, PC s přístupem k internetu, abychom mohli na stránky MakeCode a kabel s micro USB koncovkou, abychom mohli Micro:bit připojit do počítače při nahrávání programu a při jeho napájení.

Očekávané vstupní znalosti vyučujícího jsou orientace v prostředí Microsoft MakeCode, schopnost práce s blokovým kódem a celkové pochopení a schopnost vysvětlení principu Caesarovy šifry (v metodickém listu je ale pro jistotu vysvětlena).

Jak již bylo zmíněno, v RVP se toto zadání váže na sekci Data, informace a modelování, ve kterém je uvedeno jako součást učiva kapitoly „kódování a přenos dat“.

3.1.1 Úloha pro 1. stupeň

Verze pro nižší ročník obsahuje již připravené bloky na zašifrování a dešifrování. Pro žáky je vlastně již vše připraveno. První úlohu jsme chtěli udělat takovou rozjezdovou (jak pro nás, tak pro žáky) a z programátorského hlediska na ní zatím není nic moc zajímavého.

Třídu, pro níž je úloha určena, jsme vybrali podle ŠVP z imysleni.cz. V tom je téma Úvod do kódování a šifrování zařazen do učiva 4. třídy.

Žák by tímto zadáním měl splnit výstupy prvního stupně níže uvedené.

- ŠVP
 - Žákyně/žák zakóduje/zašifruje a dekáduje/dešifruje text
- RVP
 - **I-5-1-02** - Žákyně/žák popíše konkrétní situaci, určí, co k ní již ví, a znázorní ji

3.1.1.1 Rozšíření

Rozšíření obsahuje dva bloky. Jeden pro zašifrování a jeden pro dešifrování. Do obou je potřeba vložit dva parametry. Text k zašifrování nebo dešifrování a posun.

3.1.2 Úloha pro 2. stupeň

Žáci na druhém stupni základních škol již budou mít zadání trochu složitější. V rozšíření totiž nebudou mít k dispozici blok na zašifrování/dešifrování celého textu, ale jen na zašifrování/dešifrování jednoho znaku. Jelikož ale Javascript nezná datový typ „char“, není způsob, jak si vynutit jako vstup do bloku pouze jeden znak. Programově to tedy řešíme tak, že si vždy vezmeme jen první znak z textu, který byl do funkce předán. Pokud tedy uživatel zadá do bloku text A, vezme se znak A, a vše funguje bez problému. Pokud by ale uživatel zadal nějaký delší text (například „pes“), vezme se pouze znak P.

Verze zadání pro druhý stupeň je v ŠVP zařazen do látky 6. třídy. Tentokrát je více zaměřena na téma „Algoritmizace a programování“.

Žák by tímto zadáním měl splnit výstupy druhého stupně níže uvedené.

- ŠVP
 - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program, dbá na jeho čitelnost a přehlednost
- RVP
 - I-9-2-05 – Žákyně/žák v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za ně; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení programu, proměnné

3.1.2.1 Rozšíření

Obsahem rozšíření jsou opět dva bloky: zašifrování a dešifrování jednotlivých znaků. Úkolem žáků tedy je, aby text „rozsekali“ na jednotlivé znaky, poté je po jednom zašifrovali/dešifrovali a opět poskládali.

3.2 Caesarova šifra (2/2)

Druhá část úlohy s Caesarovou šifrou bude o posílání zašifrovaného textu a jeho následném dešifrování. Rozšíření bude úplně stejné, jako u první části a žáci budou ke splnění druhé částí potřebovat funkční zašifrování a dešifrování. Samotné posílání zpráv bude fungovat přes rádio, kterým Micro:bit disponuje. Úloha naplňuje téma „kódování a přenos dat“, tentokrát se ale více zaměříme na část „přenos dat“.

Nastavení rádia je velice jednoduché. Stačí si jen nastavit skupinu rádia (číslo od 1 do 255). Ta pak určuje, jaké Micro:bity spolu mohou komunikovat. Micro:bity ve stejné skupině budou navzájem přijímat zprávy. Pokud skupinu nenastavíme, nastaví se náhodně. Micro:bit má tři možnosti, jak poslat přes rádio data: odeslat číslo, odeslat text anebo odeslat text s číslem. K tomu má příslušné eventy, když Micro:bit nějakou zprávu přijme. V eventu je možné si o přijatém paketu („zprávě“) zjistit i další informace, jako třeba sílu paketu, čas, kdy byl odeslán nebo sériové číslo Micro:bitu, který paket odeslal, pokud to odesílatel povolí.

Posílání paketů probíhá na bázi „broadcastu“. To znamená, že Micro:bit zprávu pošle všem Micro:bitům v okolí (které jsou ve stejné skupině). Nelze tedy ze strany odesílatele docílit toho, aby zprávu dostal jen jeden konkrétní Micro:bit. Každý Micro:bit si tedy může zobrazit, co jeden Micro:bit odeslal. A právě o tom je tato úloha.

Žáci by na ní mohli pochopit, jak funguje symetrické šifrování a proč je vlastně důležité data šifrovat. Princip úlohy spočívá v tom, že žáci nejdříve pošlou zprávu nezašifrovanou a tím pádem si ji každý Micro:bit ve stejné skupině může přečíst. Poté přijde na řadu zhotovené zašifrování. Žáci zprávu před odesláním zašifrují a po přijetí zase dešifrují. To, aby z dešifrování vzešel původní text, závisí na správném klíči, který si musí žáci domluvit. Řekněme například, že budeme mít skupinu tří žáků. Jeden z nich bude vysílat nějaký zašifrovaný text. Zbylí dva žáci se budou snažit si text přečíst, ale jen s jedním z nich se vysílající žák tajně domluví na posunu. Ten bude tím pádem schopen si text přečíst. Třetí žák, který se také snaží přečíst text, bude muset posun uhádnout. Šance, že se ale trefí do správného čísla, je minimální a taktéž vyzkoušet nějaký větší počet kombinací v rozumném čase, není možné.

Varianty pro nižší a vyšší ročník jsou prakticky stejné až na to, že se zašifrování a dešifrování liší podle stupně obtížnosti. Ale princip je stejný. Nejdříve se nastaví skupina rádia, poté se při nějaké události (například stisknutí tlačítka) odešle text. U přijímače se také nastaví skupina a když je přijat přes rádio nějaký text, tak se jeho dešifrovaná verze zobrazí. Jeden Micro:bit by samozřejmě mohl dělat jak zobrazování dešifrovaného textu, tak odesílání zašifrované podoby.

Požadované pomůcky na tuto úlohu jsou alespoň 2 Micro:bity (nejlepší by ale bylo, kdyby měl každý žák svůj vlastní, nebo kdyby se děti mohly rozdělit na menší skupinky se sudým počtem žáků), PC s internetovým připojením a alespoň dva micro USB kabely.

Očekávané vstupní znalosti vyučujícího jsou orientace v prostředí Microsoft MakeCode, schopnost pracovat s blokovým kódem a již z předešlé úlohy chápání Caesarovy šifry.

3.2.1 Úloha pro 1. stupeň

Stejně jako první část je zadání určeno pro děti 4. třídy.

Žák by tímto zadáním měl splnit výstupy prvního stupně níže uvedené.

- ŠVP
 - Žákyně/žák zakóduje/zašifruje a dekáduje/dešifruje text
 - Žákyně/žák předá informaci zakódovanou pomocí textu či čísel
- RVP
 - **I-5-1-02** - Žákyně/žák popíše konkrétní situaci, určí, co k ní již ví, a znázorní ji
 - **I-5-4-01** - Žákyně/žák najde a spustí aplikaci, pracuje s daty různého typu

3.2.2 Úloha pro 2. stupeň

I druhá část úloha spadá na druhém stupni pod učivo 6. třídy. Tentokrát je ale více zaměřena na téma „Algoritmizace a programování“.

Žák by tímto zadáním měl splnit výstupy druhého stupně níže uvedené.

- ŠVP
 - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program, dbá na jeho čitelnost a přehlednost
- RVP
 - **I-9-1-02** - Žákyně/žák navrhuje a porovnává různé způsoby kódování dat s cílem jejich uložení a přenosu
 - **I-9-2-05** - Žákyně/žák v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za ně; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení programu, proměnné

3.3 LED mozaika

V této úloze si žáci vyzkouší za pomoci Micro:bitu „malovat“ na LED 5×5 pole, které na základním Micro:bitu je. Součástí úlohy bude hýbání kurzorem, překreslení LEDky (rozsvícení, zhasnutí), smazání celé kresby a vypnutí/zapnutí blikání kurzoru. Malováním se v tomto kontextu rozumí rozsvěcování LED.

Pro tuto úlohu je zapotřebí pouze samotný Micro:bit, micro USB kabel a PC s připojením k internetu.

Do očekávaných vstupních znalostí vyučujícího patří orientace v prostředí Microsoft MakeCode a schopnost pracovat s blokovým kódem.

3.3.1 Úloha pro 1. stupeň

Varianta pro nižší ročník obsahuje již předpřipravené bloky, které musí žáci umístit do správných (smysluplných) vstupů. Tato úloha je poměrně jednoduchého rázu. Žáci si při ní vyzkouší různé události, na které Micro:bit dokáže reagovat (tlačítka, zatřesení, naklápění...)

Tato úloha je tentokrát zařazena do učiva 5. třídy.

Žák by tímto zadáním měl splnit výstupy prvního stupně níže uvedené.

- ŠVP
 - Žákyně/žák pomocí obrázku znázorní jev
 - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program řídící chování Micro:bitu
- RVP
 - **I-5-2-03** - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program; rozpozná opakující se vzory, používá opakování a připravené podprogramy

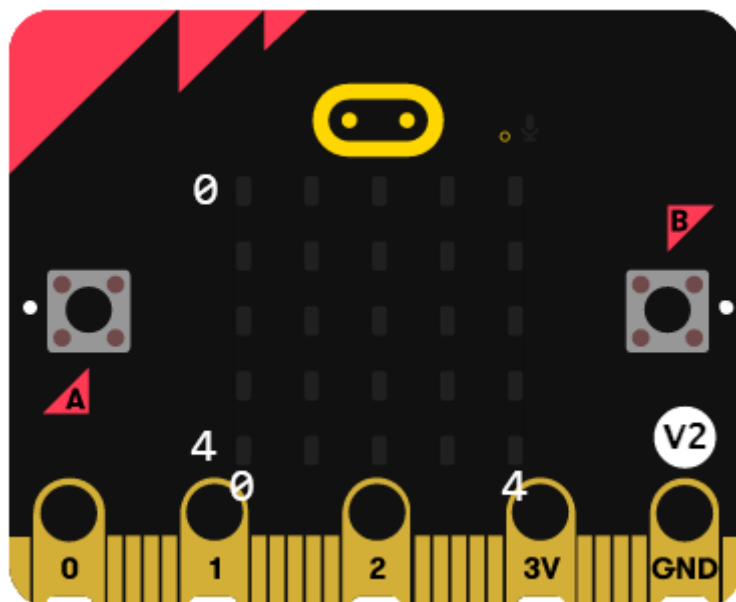
3.3.1.1 Rozšíření

Samotné rozšíření si ukládá stav mozaiky a tím pádem ví, kde se zrovna kurzor nachází. Žáci tedy musí jen umístit bloky do libovolných vstupů (vstupy by měly alespoň trochu dávat smysl, ale z funkčního hlediska je to úplně jedno). Rozšíření nabízí bloky pro překreslení bodu (aktuální bod se buď rozsvítí, jestliže nesvítil, nebo zhasne, pokud přes

provedením bloku svítit), vymazání kresby (zhasne všechny LEDky), přepnutí kurzoru (dočasně vypne kurzor, aby nepřekážel ve chvíli, kdy si bude chtít uživatel prohlédnout kresbu, anebo ho znovu zapne), blikání kurzoru a hýbání kurzorem (tento blok si jako parametr bere výčet, který udává, jakým směrem se chceme vydat). Žáci se tedy nemusí vůbec starat o souřadnice. Vše za ně dělá rozšíření.

3.3.2 Úloha pro 2. stupeň

Ve variantě pro vyšší ročník mají žáci souřadnice již ve své vlastní režii. Nejzajímavější je to, jakým způsobem zajistit, aby se kurzor přesunul například na začátek řádku, pokud z řádku vyjede. Žáci se budou tedy muset postarat o „přetékání“ kurzoru. K tomu je potřeba pochopit souřadnice, se kterými se v této úloze pracuje a které jsem se pokusil vysvětlit na následujícím obrázku:

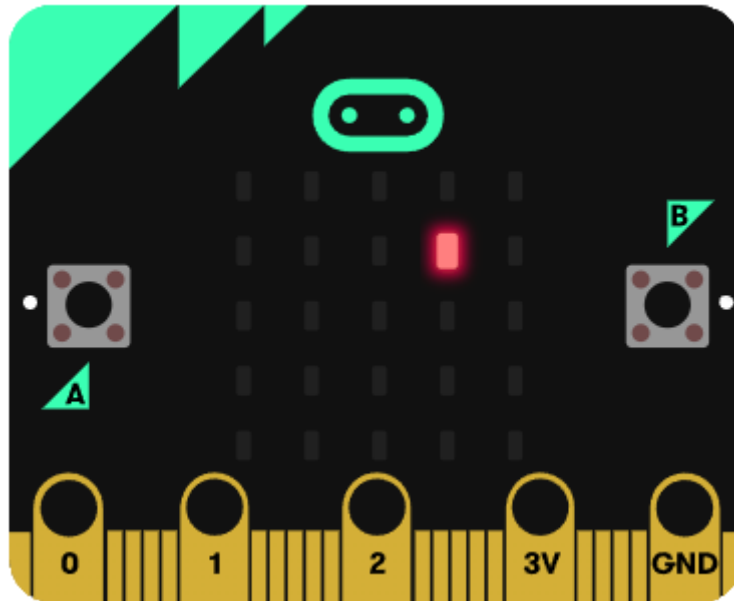


Obrázek 3 Souřadnice na Micro:bitu (7)

System souřadnic na Micro:bitu je pěkně vidět na následujícím kódu a jeho výsledku:

```
led.plot(3, 1)
```

Kód rozsvítí LED na určitých souřadnicích. V tomto případě čtvrtou LED zleva a druhou LED seshora (čísluje se od nuly). Výsledek pak vypadá takto:



Obrázek 4 Rozsvícení LED na určitých souřadnicích (7)

Úloha LED mozaika pro 2. stupeň je vytvořena pro 7. ročník.

Žák by tímto zadáním měl splnit výstupy druhého stupně níže uvedené.

- ŠVP
 - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví přehledný program k vyřešení problému
 - Žákyně/žák používá podmínky pro ukončení opakování, rozezná, kdy je podmínka splněna
- RVP
 - **I-9-2-05** - Žákyně/žák v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za ně; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení programu, proměnné
 - **I-9-2-06** - Žákyně/žák ověří správnost postupu, najde a opraví v něm případnou chybu

3.4 Světelná závora

Jak již bylo zmíněno, Micro:bit dokáže měřit hladinu světla, která je hlavním předmětem v této úloze. Světelná závora je kombinace světelného zdroje a senzoru světla, s jehož pomocí dokážeme reagovat na změnu intenzity světla (přerušeni/vypnutí

světelného zdroje). V praxi to funguje jako „laserový paprsek“, jehož přerušení může mít za následek nějakou událost (například zahoukání). K této úloze budou žáci mimo Micro:bitu potřebovat i nějaký zdroj světla, který se dá jednoduše úmyslně přerušit, ale zároveň ho nepřerušíme omylem (stropní světlo například není vhodné, protože senzor zaznamená i drobnější stíny, které nemusíme dělat úmyslně). Ideální je jakákoli baterka, protože s ní lze dobře manipulovat.

Do očekávaných vstupních znalostí vyučujícího patří orientace v prostředí Microsoft MakeCode a schopnost pracovat s blokovým kódem.

3.4.1 Úloha pro 1. stupeň

Žáci si ve verzi pro nižší stupeň budou moci zvolit, co se má stát, když dojde k porušení světelné závory. Zbytek je pro ně již vytvořen.

Úloha světelná závora je v rámci prvního stupně určena pro 4. ročník.

Žák by tímto zadáním měl splnit výstupy prvního stupně níže uvedené.

- ŠVP
 - Žákyně/žák sestaví program pro Micro:bit
 - Žákyně/žák pomocí programu ovládá senzor
 - Žákyně/žák události ke spouštění programu
- RVP
 - **I-5-2-03** - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program; rozpozná opakující se vzory, používá opakování a připravené podprogramy
 - **I-5-2-02** - Žákyně/žák popíše jednoduchý problém, navrhne a popíše jednotlivé kroky jeho řešení

3.4.1.1 Rozšíření

Rozšíření obsahuje blok, který desetkrát změří hladinu světla, kterou poté zprůměruje (vydělí deseti) a výslednou hodnotu uloží jako referenční hodnotu, která se poté bude porovnávat s aktuálně naměřenou hladinou světla. Dále je potřeba tomuto bloku jako parametr předat toleranci (číslo od 0 do 255), což značí, jak moc se může hladina světla vychýlit od referenční hodnoty, aniž by se provedl kód, který je možné

vložit do druhého bloku. Ten vyžaduje bloky (tzv. delegát) podle volby uživatele, které se provedou, pokud dojde k porušení hladiny světla.

3.4.2 Úloha pro 2. stupeň

Verze pro vyšší ročník bude obsahovat úplně to stejné rozšíření, ale úkolem žáků bude naimplementovat přerušení hry (zmáčkne se tlačítko a Micro:bit přestane snímat intenzitu světla; zmáčkne se znovu a Micro:bit opět začne se snímáním).

Verze pro druhý stupeň je určena 8. třídě.

Žák by tímto zadáním měl splnit výstupy druhého stupně níže uvedené.

- ŠVP
 - Žákyně/žák vytvoří program pro micro:bit a otestuje jeho funkčnost
 - Žákyně/žák vyřeší problém tím, že naprogramuje micro:bit
- RVP
 - **I-9-2-05** - Žákyně/žák v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za ně; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení programu, proměnné

3.5 Poplach, poplach!

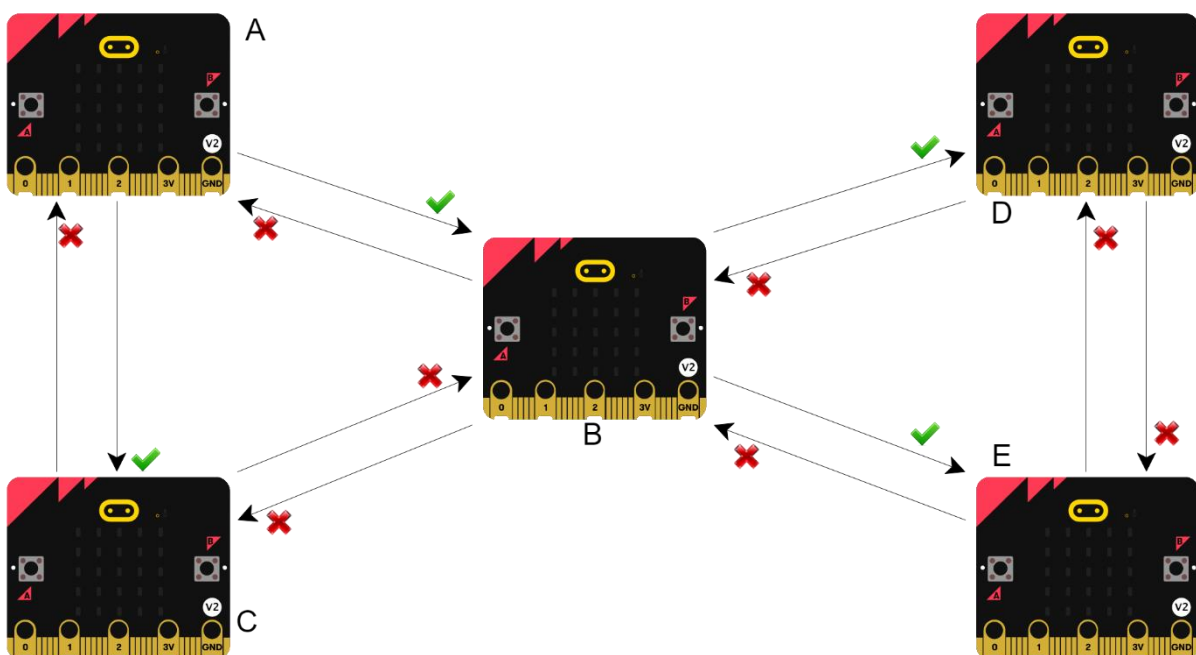
Tato úloha funguje jako alarm, který se spustí po zmačknutí tlačítka (nebo provedení jakéhokoli jiného vstupu). Alarm funguje cyklicky, což znamená, že pokud máme v dosahu Micro:bity se správným programem (kódem), tak zapnutí alarmu pošle zároveň přes rádio zprávu Micro:bitům v okolí, že mají také spustit alarm (začít houkat). Tyto Micro:bity zprávu opět pošlou dál. Tímto způsobem se dá vytvořit takový „řetěz“. Alarm lze poté i vypnout, a to z jakéhokoli Micro:bitu.

Takže například na Micro:bitu A zmáčkne tlačítko. Micro:bit začne houkat a tím pošle Micro:bitům v okolí zprávu, ať taky začnou houkat. Ty to opět pošlou dál. Zprávu dostane i Micro:bit Z, na kterém zmáčkne zase jiné tlačítko a tím alarm vypne. Micro:bit Z tedy přestane houkat a pošle zprávu s pokynem Micro:bitům v okolí, aby taky přestaly houkat. Zpráva se nakonec dostane zpět až k Micro:bitu A, který původně alarm zapnul.

Rozšíření je samozřejmě pojištěno i proti cyklení paketů pořád dokola. Pokud Micro:bit A pošle zprávu Micro:bitu B, zpráva se mu poté vrátí, protože Micro:bit B zprávu jednoduše pošle všem v okolí a neumí zprávu neposlat tomu Micro:bitu, ze kterého zpráva přišla.

Zabránění zacyklení je vyřešeno tak, že Micro:bit nepřeosílá dál zprávy, které „říkají“ to, co Micro:bit již dělá. Pokud tedy Micro:bit A pošle paket, který značí spuštění alarmu a sám alarm spustí, bude ignorovat paket, který mu poté přijde opět z Micro:bitu B s obsahem, který říká „začni houkat“, protože Micro:bit A již houká.

Celý systém posílání paketů se pokusím více objasnit na této kresbě:



Obrázek 5 Posílání paketů

Komunikaci zahájil Micro:bit A (řekněme, že uživatel spustil na Micro:bitu A alarm). Ten tedy začne houkat a pošle zprávu Micro:bitům, které má v dosahu, aby taky začaly houkat. Micro:bity B a C taky začnou houkat a pošlou zprávu Micro:bitům v okolí. Oba pošlou zprávu Micro:bitu A, který ale jejich zprávu bude ignorovat, protože již houká a nebude tím pádem ani nic víc posílat. Zároveň si Micro:bity B a C pošlou zprávy navzájem, ale protože oba již houkají, tak svoje zprávy také budou ignorovat. Micro:bit B má ještě v dosahu Micro:bity D a E. Oba přijmou zprávu od Micro:bitu B a začnou houkat, ale Micro:bit B jejich zprávy již nepřijme, protože houká a ze stejného důvodu Micro:bity E a D nepřijmou zprávy, které si vymění mezi sebou.

Do požadovaných pomůcek je jako vždy zahrnut Micro:bit, micro USB kabel a PC s připojením k internetu, navíc je zde vhodné mít ke každému Micro:bitu k dispozici držák na baterie (obvykle dvě AAA) s odpovídajícím kabelem, abychom mohli zařízení zapnuté přenášet.

Do očekávaných vstupních znalostí vyučujícího patří orientace v prostředí Microsoft MakeCode a schopnost pracovat s blokovým kódem.

3.5.1 Úloha pro 1. stupeň

Ve verzi pro nižší ročník se budou muset žáci vypořádat se zapínáním a vypínáním alarmu a poté se zpracováním pokynu, který může přijít od jiného Micro:bitu.

Zadání je vytvořeno pro 5. třídu.

Žák by tímto zadáním měl splnit výstupy prvního stupně níže uvedené.

- ŠVP
 - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program řídící chování Micro:bitu
- RVP
 - **I-5-2-03** - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program; rozpozná opakující se vzory, používá opakování a připravené podprogramy

3.5.1.1 Rozšíření

Rozšíření se skládá ze tří bloků. Jeden z bloků dokáže spustit alarm a zároveň odeslat zprávu ostatním Micro:bitům, aby alarm taky spustily. Poté je k dispozici blok, který vypne alarm a „poručí“ Micro:bitům, aby udělaly to stejné. Jako poslední blok je „Přijmout pokyn“, který přijme a zpracuje pokyn od jiného Micro:bitu, nebo ho odmítne. Jako parametr si tento blok bere pokyn, což je nějaký text. K tomu je samozřejmě potřeba blok ze sekce rádio, který je uzpůsoben na přijímání textu (blok je jmenuje „když je přijat text“).

3.5.2 Úloha pro 2. stupeň

V těžší verzi budou mít žáci k dispozici pouze dva bloky. Jeden pro spuštění alarmu a odeslání zprávy ostatním a jeden pro vypnutí alarmu a poslání paketu ostatním Micro:bitům v okolí.

Úloha je vytvořena pro 6. třídu.

Žák by tímto zadáním měl splnit výstupy druhého stupně níže uvedené.

- ŠVP
 - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program pro ovládání micro:bitu, v programu najde a opraví chyby
- RVP
 - **I-9-2-05** - Žákyně/žák v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za ně; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení programu, proměnné

3.5.2.1 Rozšíření

K dispozici budou pouze dva bloky a žáci si budou muset poradit bez bloku, který přijímal pokyny. O to se budou muset totiž postarat žáci. Hlavní myšlenka spočívá v tom, že si žáci zvolí nějaké dva texty (klíče, řekněme) pro zapínání a vypínání alarmu, na které bude Micro:bit reagovat (například „alarm_zapnout“ a „alarm_vypnout“). Poté stačí do bloku, který přijímá všechny rádiové zprávy („když je přijat text“), vložit správnou logiku (měl by se zkontrolovat obsah přijatého paketu a podle toho se rozhodnout, co provést).

3.6 Zeptáme se publika 1/2

Tyto dvě úlohy dohromady zrealizují sběr jakýchkoli dat (hlasování). Na to bude potřeba jeden server („výběrčí dat“) a několik hlasujících zařízení. Tato úloha je, jak nadpis napovídá, rozdělena na dvě části. V první části bude tedy úkol dělat hlasovací zařízení. Jelikož je první část velice jednoduchá, tak bude jen jedna varianta pro oba ročníky a bude bez jakéhokoli rozšíření.

Jediný problém, kterému budou v této úloze žáci čelit, je odeslání textu (hlasovací možnost) společně se sériovým číslem zařízení (unikátní číslo, které

jednoznačně identifikuje Micro:bit). Žáci totiž budou hlasovat jen systémem A, B, C D a to, co samotné možnosti budou reprezentovat, si nastaví až server (druhá část). Sériové číslo žáci musí poslat z toho důvodu, aby si server mohl přiřadit jejich hlas právě k sériovému číslu a v případě, že by žáci v rámci jednoho hlasování hlasovali vícekrát, změní se pouze jejich hlas, protože bude jejich sériové číslo uloženo na serveru.

Odesílání sériového čísla Micro:bit je možné po povolení od uživatele. MakeCode obsahuje totiž přímo blok, ve kterém se dá nastavit hodnota true/false podle toho, jestli se má sériové číslo společně s paketem posílat. Na straně serveru je možno pak sériové číslo z paketu „vypreparovat“. Pokud by žáci odesílání sériového čísla nepovolili, na server by z paketu vytáhl místo sériového čísla číslo 0. V tomto případě rozšíření hlas nezaznamená.

Požadované pomůcky na tuto část úlohy jsou co největší počet Micro:bitů (nejlepší by bylo, kdyby měl každý žák svůj vlastní, nebo kdyby se děti mohly rozdělit na menší skupinky), PC s internetovým připojením a ke každému Micro:bitu kabel s micro USB konektorem. Vhodné je také mít dopředu vymyšlené téma hlasování, aby se na konci dala zpracovaná úloha vyzkoušet.

Do očekávaných vstupních znalostí vyučujícího patří orientace v prostředí Microsoft MakeCode a schopnost pracovat s blokovým kódem.

Tato úloha díky své jednoduchosti nemá dvě varianty a je určena jak pro první stupeň, tak pro druhý.

3.6.1 Úloha pro 1. stupeň

V rámci prvního stupně je úloha vhodná pro 4. třídu.

Žák by tímto zadáním měl splnit výstupy prvního stupně níže uvedené.

- ŠVP
 - Žákyně/žák
- RVP
 - **I-5-2-03** - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program; rozpozná opakující se vzory, používá opakování a připravené podprogramy

3.6.2 Úloha pro 2. stupeň

Pro druhý stupeň bych úlohu zařadila do 6. třídy.

Žák by tímto zadáním měl splnit výstupy druhého stupně níže uvedené.

- ŠVP
 - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program, dbá na jeho čitelnost a přehlednost
- RVP
 - **I-9-2-05** - Žákyně/žák v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za ně; program vyzkouší a opraví v něm případné chyby;

3.7 Zeptáme se publika 2/2

V druhé části budou mít žáci za úkol vytvořit server, který bude shromažďovat všechny hlasy a vypisovat je.

Požadované pomůcky na tuto část úlohy jsou jeden Micro:bit, PC s internetovým připojením a kabel s micro USB konektorem. Vhodné je také mít dopředu vymyšlené téma hlasování, aby se na konci dala zpracovaná úloha vyzkoušet.

Do očekávaných vstupních znalostí vyučujícího patří orientace v prostředí Microsoft MakeCode a schopnost pracovat s blokovým kódem.

3.7.1 Úloha pro 1. stupeň

Verze pro nižší ročník opět nebude obsahovat mnoho programování. O veškerou funkcionalitu se totiž bude starat rozšíření.

Druhá část je na prvním stupni tak, jako předchozí část také pro žáky 4. třídy.

Žák by tímto zadáním měl splnit výstupy prvního stupně níže uvedené.

- ŠVP
 - Žákyně/žák sestaví program pro Micro:bit
 - Žákyně/žák oživí Micro:bit, otestuje jeho chování
 - Žákyně/žák používá opakování, události ke spouštění programu

- RVP
 - **I-5-2-03** - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program; rozpozná opakující se vzory, používá opakování a připravené podprogramy

3.7.1.1 Rozšíření

Rozšíření si bude pamatovat sériová čísla Micro:bitů, které již hlasovaly, společně s informací, jak hlasovaly. Rozšíření bude dále obsahovat celkem 4 bloky. První nastaví samotný obsah hlasování (jestli chceme hlasovat například o barvě, zvířeti, čísle...). Jako parametry si bere jednotlivé hodnoty (A, B, C D). Jak bylo zmíněno u „hlasovače“, server bude dostávat pouze pakety s obsahem A, B, C nebo D společně se sériovým číslem. Pokud tedy například budeme chtít hlasovat o tom, jakou by si žáci vybrali barvu z množiny černá, bílá, červená a modrá, předáme bloku právě tyto barvy jako vstupní argumenty.

Jako další blok máme k dispozici blok, která zaznamená hlas. Ten je potřeba umístit na místo, kam budou do Micro:bitu „přicházet“ zprávy – do bloku „když je přijat text“. Blok si poté porovná příchozí sériové číslo se sériovými čísly, které má uložené a pokud zjistí, že dané sériové číslo již hlasovalo, přepíše jeho dosavadní hlas.

Poslední dva bloky jsou již o něco jednodušší. Blok „Zobraz výsledky“ slouží k zobrazení dosavadních nasbíraných informací na LED pole, kterým Micro:bit disponuje. Vypsání probíhá tak, že se projde pole všech hlasovacích záznamů a spočítá se, kolik prvků od každé možnosti v poli je. Na LED poli se poté zobrazí text s názvem hlasovací možnosti (například „černá“) a společně s ní počet hlasů pro danou možnost.

Zbývající blok „Spust' nové hlasování“ jednoduše vymaže všechny záznamy o hlasech a sériových číslech, což vyústí v nové hlasování.

3.7.2 Úloha pro 2. stupeň

V těžší verzi budou k dispozici bloky pro zaznamenání hlasu a pro nové hlasování. Nebudou ale již přítomny bloky pro nastavení hlasovacích hodnot a pro zobrazení výsledků. Místo toho si budou muset žáci vystačit s blokem, který vrátí pole se zaznamenanými odpověďmi.

I druhá část hlasování je zařazena do 6. třídy.

Žák by tímto zadáním měl splnit výstupy druhého stupně níže uvedené.

- ŠVP
 - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program, dbá na jeho čitelnost a přehlednost
 - používá cyklus s pevným počtem opakování, rozezná, zda má být příkaz uvnitř nebo vně opakování,
- RVP
 - **I-9-2-05** - Žákyně/žák v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za něj; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení

3.7.2.1 Rozšíření

Rozšíření obsahuje celkem tři bloky. Bloky „Spust' nové hlasování“ a „Zaznamenej hlas se sériovým číslem“ fungují úplně stejně, jako u nižšího ročníku. Třetí blok bude vracet pole se zaznamenanými odpověďmi (například pokud dostaneme pole ["A", "A", "B", "D"], znamená to, že máme celkem 4 hlasy: 2×A, 1×B, 0×C, 1×D). Úkolem žáků poté bude, aby na základě pole spočítali, kolik hlasů od každé možnosti mají a poté výsledky vypsali na LED pole. V rozšíření úplně chybí definice hlasovacích možností. Ty budou totiž také v režii žáků.

Například takto by mohlo vypadat vypsání dosavadních výsledků hlasování:

```
input.onButtonPressed(Button.A, function () {
  odpovedi = server.currentAnswers()
  pocetD = 0
  pocetC = 0
  pocetB = 0
  pocetA = 0
  for (let hodnota of odpovedi) {
    if (hodnota == "A") {
      pocetA += 1
    } else if (hodnota == "B") {
      pocetB += 1
    } else if (hodnota == "C") {
      pocetC += 1
    } else if (hodnota == "D") {
```

```

        pocetD += 1
    }
}
basic.showString("ANO" + pocetA)
basic.showString("NE" + pocetB)
basic.showString("MOZNA" + pocetD)
basic.showString("NEVIM" + pocetC)
})

```

Jako první se do pole nahraje pole odpovědí. Poté se vynulují proměnné, které v sobě budou držet celkový počet hlasů pro každou možnost. Následně pole projde „for“ smyčkou a pro každý prvek se rozhodne, jakou ze čtyř proměnných zvýšit. Nakonec stačí jen ukázat výsledky na LED poli. Hlasovací možnosti jsou na tomto příkladu definovány až v samotném vypsání (ANO, NE, MOZNA, NEVIM).

3.8 Pozor, vejce, nespadni!

Úloha by měla simulovat hru „vajíčko na lžičce“. Představa je taková, že se žáci například rozdělí do dvou skupin, každá skupina bude mít jeden Micro:bit s úlohou a úkolem obou týmů bude přejít z bodu A do bodu B a opět se vrátit do bodu A, kde budou čekat ostatní žáci ze skupiny. Žák, který se vrátil, předá Micro:bit dalšímu žákovi, který opět půjde z bodu A do bodu B a zpět. Pokud se stane, že by při cestě Micro:bit detekoval „spadnutí vejce“, musí se žák vrátit do bodu A, a opět vyrazit do bodu B a zpět. Vyhrává ta skupina, která toto zvládne nejrychleji.

V praxi to tedy vypadá tak, že je na LED poli rozsvícena jedna LED, která značí pozici vejce. Když Micro:bit různě naklápíme, pozice rozsvícené LED se bude měnit v závislosti na tom, jakým směrem Micro:bit naklopíme. Pokud Micro:bit naklopíme moc (do stavu, kdy bude rozsvícená LED na okraji LED pole), „vejce spadne“. Změna polohy rozsvícené LED závisí na akcelerometru, který má v sobě Micro:bit zakomponovaný. Zde tedy dochází k vazbě na fyziku.

Požadované pomůcky pro tuto úlohu jsou optimálně alespoň dva micro:bity (abychom mohli ze hry udělat soutěž), PC s internetovým připojením a ke každému micro:bitu kabel s micro USB konektorem. Volitelně si vyučující nebo samotní studenti mohou přinést mechanické překážky či předměty, které jim fyzicky ztíží a zpestří toto zadání (např. balanční podložka, kužely na slalom apod.).

Do očekávaných vstupních znalostí vyučujícího patří orientace v prostředí Microsoft MakeCode, schopnost pracovat s blokovým kódem a dostačující kreativita pro ztížení konečného výstupu zadání.

3.8.1 Úloha pro 1. stupeň

Zadání pro nižší ročník bude poměrně jednoduché. Žáci si budou moci regulovat náročnost hry skrz parametr „obtížnost“, který obsahuje různé stupně obtížnosti a dále budou mít v rukou událost, která se provede, když „vejce spadne“.

Verzi pro 1. stupeň jsem zařadila do 5. třídy.

Žák by tímto zadáním měl splnit výstupy prvního stupně níže uvedené.

- ŠVP
 - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program pro ovládání Micro:bitu
- RVP
 - **I-5-2-03** - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program; rozpozná opakující se vzory, používá opakování a připravené podprogramy

3.8.1.1 Rozšíření

Rozšíření k této úloze bude obsahovat celkem tři bloky. První spustí hru a jako parametr si bere toleranci (čím větší, tím jednodušší hra je). Poté rozšíření obsahuje blok „aktualizuj“, který je vhodné umístit do smyčky „opakuj stále“, aby dokola aktualizoval pozici rozsvícené LED na LED poli. Jako poslední je k dispozici blok „Při pádu vejce“, který, jak název napovídá, provede sérii uživatelem zvolených bloků, když dojde k „pádu vejce“ (rozsvícená LED je na okraji LED pole).

3.8.2 Úloha pro 2. stupeň

V rozšíření bude blok na spuštění nové hry, ale nebude již k dispozici blok na aktualizaci. Místo toho bude ale přibude blok, který vrátí pole, které bude obsahovat aktuální souřadnice (například [2, 3]). Bude tedy potřeba udělat to, aby Micro:bit nějak zareagoval, když budou souřadnice v nepovolených hodnotách.

Tato úloha je zařazena do 9. třídy.

Žák by tímto zadáním měl splnit výstupy druhého stupně níže uvedené.

- ŠVP
 - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví přehledný program k vyřešení problému
- RVP
 - **I-9-2-05** - Žákyně/žák v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za něj; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení programu, proměnné
 - **I-9-2-06** - Žákyně/žák ověří správnost postupu, najde a opraví v něm případnou chybu

3.8.2.1 Rozšíření

Rozšíření bude obsahovat opět blok „Spust' hru s obtížností“, který bude umožňovat, stejně jako u verze pro nižší ročník, nastavení složitosti hry. Novým blokem bude blok „Souřadnice“, který vrátí aktuální souřadnice, na kterých se nachází rozsvícená LED. Souřadnice budou v poli o dvou prvcích, z nichž první bude obsahovat osu X a druhý osu Y.

3.9 Tik tok bác!

Poslední úloha je algoritmicky nejsložitější a měla by představovat simulaci hry podobnou mikádu nebo takového „hlídače“ věci. Příklad využití může třeba být, že se Micro:bit vloží do kelímku s tužkami a žáci budou tužky postupně vytahovat. Pokud Micro:bit zaznamená pohyb (změřená akcelerace bude příliš nízká nebo příliš vysoká), začne houkat (nebo přehraje jakýkoli jiný zvuk... to bude v režii žáků). Další příklad použití by bylo, že se Micro:bit položí na nějakou věc (například telefon) a úkolem žáků bude Micro:bit pomalu dát pryč, aby si věc pod ním mohli vzít.

Do požadovaných pomůcek je zahrnut Micro:bit, micro USB kabel a PC s připojením k internetu, navíc je zde vhodné mít ke každému Micro:bitu k dispozici držák na baterie (obvykle dvě AAA) s odpovídajícím kabelem, abychom mohli zařízení zapnuté pokládat i na místa dál od PC. Jako poslední věc je vhodné mít jakýkoliv předmět (kniha, penál, podložka), na kterém můžeme hlídač vyzkoušet.

Do očekávaných vstupních znalostí vyučujícího patří orientace v prostředí Microsoft MakeCode a schopnost pracovat s blokovým kódem.

3.9.1 Úloha pro 2. stupeň (lehčí varianta)

Nižší ročník bude mít k dispozici blok, který bude detekovat pohyb. Žáci si pak sami určí, co se má stát v moment, kdy k tomuto dojde. Bonusové vylepšení bude vypínání a zapínání hlídání, které si žáci budou muset ale vytvořit sami (není součástí rozšíření).

Zadání je vytvořeno pro 7. třídu.

Žák by tímto zadáním měl splnit výstupy prvního stupně níže uvedené.

- ŠVP
 - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví přehledný program k vyřešení problému
 - Žákyně/žák používá podmínky pro ukončení opakování, rozezná, kdy je podmínka splněna
- RVP
 - **I-9-2-05** - Žákyně/žák v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za ně; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení programu, proměnné

3.9.1.1 Rozšíření

Rozšíření u těchto dvou úloh bude velice krátké a jednoduché. Obsahuje totiž jen jeden blok, které představuje detekci pohybu. Bloku se předají bloky, které se mají provést v moment, kdy dojde k pohybu společně s obtížností hry, kterou si uživatel může zvolit.

3.9.2 Úloha pro 2. stupeň (těžší varianta)

Verze pro vyšší ročník je nejsložitější úloha, kterou tato sada nabídne. Žáci se v ní budou muset postarat o vše. O detekci zrychlení, počítání s odchylkou, zapínání a vypínání hlídání anebo zpětnou vazbu na displej v různých fázích hry.

Zadání je vytvořeno pro 9. třídu.

Žák by tímto zadáním měl splnit výstupy druhého stupně níže uvedené.

- ŠVP
 - Žákyně/žák řeší problém jeho rozdělením na části pomocí vlastních bloků
 - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví přehledný program k vyřešení problému
- RVP
 - **I-9-2-02** - Žákyně/žák rozdělí problém na jednotlivě řešitelné části a navrhne a popíše kroky k jejich řešení
 - **I-9-2-05** - Žákyně/žák v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za ně; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení

3.9.2.1 Rozšíření

Rozšíření obsahuje pouze jeden blok, který vrátí klidovou hodnotu akcelerometru, což je podle dokumentace 1023. Detekce příliš velké či malé akcelerace by mohla vypadat třeba takto:

```
function detekujPohyb(tolerance: number) {  
  akcelerace = input.acceleration(Dimension.Strength)  
  if (akcelerace + tolerance < mikado.normalValue()  
      || akcelerace - tolerance > mikado.normalValue()) {  
    return true  
  } else {  
    return false  
  }  
}
```

Micro:bit nabízí měření akcelerace ve třech dimenzích (x, y, z) anebo výslednou akceleraci ze všech třech směrů (Dimension.Strength).

4 Návrh metodických a pracovních listů

4.1 Metodické listy

Metodické listy jsou pomocné dokumenty pro učitele, ve kterých najdou potřebné informace o každé úloze, její postup, důležité poznámky k tématu zadání, případné otázky, na něž se mohou v průběhu hodiny zeptat žáků, aby ověřili, zda všemu rozumí.

Jako první je důležité uvést souhrnné informace o jednotlivé úloze. Proto jsem se rozhodla na začátek každého dokumentu vložit tabulku se základními informacemi jako jsou název úlohy, ročník, pro který je vytvořena, RVP a ŠVP požadavky jež splňuje, krátký popis cvičení, a nakonec odhadovaný čas, který by měla úloha optimálně zabrat.

Název úlohy	Caesarova šifra 1/2
Třída	4. třída
Úloha splňuje RVP rámce	<ul style="list-style-type: none"> • DATA, INFORMACE A MODELOVÁNÍ – kódování a přenos dat • ALGORITMIZACE A PROGRAMOVÁNÍ – řešení problému krokováním, programování, kontrola řešení
Propojení s RVP výstupy	<ul style="list-style-type: none"> • I-5-1-02 - Žákyně/žák popíše konkrétní situaci, určí, co k ní již ví, a znázorní ji
Propojení s ŠVP výstupy	<ul style="list-style-type: none"> • Žákyně/žák zakóduje/zašifruje a dekoduje/dešifruje text
Časová náročnost	45 minut (jedna vyučovací hodina)
Stručný popis úlohy	Žáci se seznámí s šifrováním a pomocí předpřipravených bloků v MakeCode si zkusí zašifrovat jejich vlastní zprávu.
Odkaz na rozšíření	https://github.com/microbit-cz/pxt-caesarova-sifra-rozsireni-nizsi-rocnik

Obrázek 6 Ukázka úvodní tabulky metodického listu

Jako další následuje podrobnější popis a cíl úlohy, vysvětlení principu a potřebné informace, které je nutné sdělit žákům na začátku.

Další částí je sekce „Budete potřebovat“, v níž je sepsán list potřebných pomůcek. Vždy se v něm objevuje samotný Micro:bit, micro USB kabel a PC s přístupem na stránky MakeCode. Jinak se do této části píšou jakékoliv jiné pomůcky nebo zařízení, které by mohly být vyžadovány pro úspěšné dokončení úlohy.

Následující část s názvem Předpokládané vstupní znalosti obsahuje všechny požadavky na učitele či lektora, které jsou nutné pro to, aby byli schopni cvičení s žáky dokončit.

Samotné vzorové řešení a postup je rozdělen do kroků. Popis každého z nich je vždy doprovázen obrázkem toho, jak má popsaná část na konci kroku vypadat.

Nechybí ani „tipy a rady“ pro vyučujícího, možné otázky, na které se u určité části postupu může zeptat a zapojit tak děti a přimět je k zamyšlení nad tím, co zrovna dělají.

4.2 Pracovní listy

Pracovní listy jsou určeny pro žáky. Jsou vytvořeny s myšlenkou, že je učitel před hodinou vytiskne a poté na začátku výuky rozdá. Obsahují potřebné doprovodné materiály k hodině a korespondují s metodickými listy vyučujícího (některé části mohou být totožné na obou dokumentech).

Samotné rozvržení a typ obsahu pracovního listu se může lišit od úlohy k úloze podle obtížnosti a tématu.

V lehčích, nebo více teoretických úlohách jsou zahrnuta doplňovací cvičení, spojovačky apod.

V zadání již více zaměřených na samotné programování v MakeCode, je list pojat spíše jako pomocník, obsahuje podrobnější postup, rady, tipy a pomocné otázky, které mohou přijít vhod, když se student zasekne a neví si dál rady.

Brala jsem taky v úvahu, že starší žáci jsou již mnohem více vedeni k samostatné práci, proto se pojetí listu pro vyšší ročník jako takový „návod“, jevil jako dobrý nápad.

U nižších ročníků se naopak snažím o to, aby se děti nenudily, proto jsou často cvičení propojená určitým příběhem, nebo jsou zahrnuty situace, ve kterých jejich vyřešením pomůžou např. některé z postav, které v listech vystupují.

2. Honzík a Anička si posílají tajné zprávy. Honzík pošle zašifrovanou větu, která vypadá takto:

VR MH BČDCYC.

Anička ví, že Honzík posunul každé písmenko zprávy o 3 místa doprava. Pomozte jí zjistit, co Honzík napsal. Znovu použijte tabulku s abecedou.

(Pozor!!! Snažíme se získat původní zprávu. Musíme tedy každé písmenko posunout o 3 místa doleva)

Obrázek 7 Ukázka cvičení v pracovním listu

5 Zpětná vazba od žáků

Díky zpětné vazbě žáků 8. a 9. tříd v rámci akce školy Soboty s technikou jsme provedli menší úpravy.

V alarmu se například neposílal jen text, ale i číslo, což ale žákům přišlo matoucí, proto jsme podle toho úlohu upravili. Dále jsme do úloh přidali programové události, protože se s nimi žákům lépe pracuje. Pokusili jsme se vylepšit popisky některých bloků, které nebyly žákům jasné. Dále jsme sjednotili pojmenování bloků, zjednodušili pojmenování repositářů, využili více funkce, které si žáci mohou přímo vytvořit v MakeCode a další menší úpravy.

Závěr (Jan Smutný)

S konzultantem práce jsme splnili osnovu – vybrali jsme úlohy, ty jsem nejdříve realizoval bez knihoven a poté jsem k nim knihovny dotvořil. Kdybych ale stál před touto prací znovu, určitě bych jí dával větší prioritu, neboť se až v průběhu práce ukázala být velmi časově náročnou. Bez předchozích zkušeností není jednoduché připravovat fragmenty kódu pro použití někým dalším. Zvláště, když se jedná o žáka základní školy, od kterého lze očekávat pouze základní orientaci v zápisu bloků kódu.

Na druhou stranu bylo pro mě zajímavé seznámit se s platformou Micro:bit do větší hloubky, a získat tak praktické srovnání s MCU Arduino, se kterým mám také praktické zkušenosti.

Dále bych určitě věnoval více času zpětné vazbě jak od učitelů, tak od žáků. Práci jsem dělal, jak nejlépe dovedu a jsem připraven na ni dále pracovat, aby plnila svůj účel a byl využit její plný potenciál.

Závěr (Michaela Hübnerová)

Vytvořila jsem set metodických listů pro vyučující na základní škole a k nim korespondující pracovní listy pro jejich žáky, za účelem rozšíření materiálů, které jsou v tuto chvíli k dispozici. Samotné úlohy se týkají programování jednodeskového počítače BBC Micro Bit (zjednodušeně Micro:bit).

Má nejoblíbenější část této práce bylo počáteční vymýšlení úloh. Diskutovali jsme to ve třech lidech, takže byl dostatek nápadů, ze kterých jsme pak mohli vybrat ty, co se nám zdáli nejunikátnější nebo něčím nejzajímavější. Pan učitel a zároveň náš konzultant práce nám pomohl vyřadit ty, jež neodpovídali naší myšlence udržet úlohy nízkonákladové.

Mezi některé z mých problémů patřil výběr samotného designu listů. Chtěla jsem je udělat něčím zajímavé, ale aby přitom bylo jasné, na co se má učitel nebo poté samotný student soustředit. Snažila jsem se každou úlohu barevně zkoordinovat tak, aby bylo vidět, že k sobě metodický a pracovní list patří. Musela jsem ale také mít na paměti, že se pracovní listy budou tisknout, stejně tak se k tomuto řešení mohou uchýlit i někteří učitelé s jejich průvodcem hodinou. Proto bylo nutné zajistit, aby i kdyby byl výsledný dokument vytisknut černobíle, kvalita čitelnosti textu a viditelnosti případných obrázků nebude ovlivněna.

Když se podívám zpátky na dobu, kdy jsem projekt začínala, přála bych si, abych ho již od počátku brala trochu vážněji a věnovala mu větší míru pozornosti. Tak by se třeba dalo vytvořit ještě více úloh a dále rozvinout ty současné.

Co se týká tvorby tohoto dokumentu, byla jsem ráda, že na chvíli mohu jenom psát a nepřemýšlet tolik nad jeho vizuální stránkou. Celý proces byl pro mě docela chaotický, často jsem přeskakovala z kapitoly na kapitolu podle toho, kam směřovaly moje myšlenky. Chvillemi mě psaní bavilo, chvillemi jsem si u něj trhala vlasy, ale to k tomu asi taky patří.

Ve finále můžu říct, že jsem se svou prací spokojena a věřím, že můj čas nad ní strávený nepřijde nazmar.

Seznam zkratk a odborných výrazů

RVP

Rámcově vzdělávací program

ZŠ

Základní škola

Arduino

Mikropočítač

Raspberry Pi

Mikropočítač

BBC

British Broadcasting Corporation

LED

Light-Emitting Diode

LE

Low Energy

Debuggování

Proces testování kódu a případné odstraňování chyb.

USB

Universal Serial Bus

Javascript

Programovací jazyk

Python

Programovací jazyk

GitHub

Bezplatný webový nástroj pro snadnou správu souborů využívající Git

Git

Verzovací systém

ASCII

American Standard Code for Information Interchange

Enum

Zkratka pro slovo enumerated („výčet“). V malování se jednalo o výčet směrů (nahoru, dolů, doprava, doleva).

String

Datový typ v programování reprezentující text

ŠVP

Školní vzdělávací program

PRIM

Podpora rozvíjení infromatického myšlení

Open source

Označení pro program nebo systém s otevřených zdrojovým kódem, což znamená, že ho (často za určitých podmínek) kdokoliv může upravovat

ARM architektura

Procesorová architektura, která se používá zejména v malých zařízením, jako jsou chytré telefony, tablety apod.

Seznam obrázků

Obrázek 1 Prostředí MakeCode (7)	6
Obrázek 2 Příklad bloku z knihovny (7).....	7
Obrázek 3 Souřadnice na Micro:bitu (7)	14
Obrázek 4 Rozsvícení LED na určitých souřadnicích (7)	15
Obrázek 5 Posílání paketů	18
Obrázek 6 Ukázka úvodní tabulky metodického listu.....	30
Obrázek 7 Ukázka cvičení v pracovním listu	32

Použité zdroje

1. **Ministerstvo školství, mládeže a tělovýchovy České republiky.** OPATŘENÍ MINISTRA ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY, KTERÝM SE MĚNÍ RÁMCOVÝ VZDĚLÁVACÍ PROGRAM PRO ZÁKLADNÍ VZDĚLÁVÁNÍ OD 1. 9. 2021. *MŠMT ČR*. [Online] 2022. [Citace: 12. Únor 2022.] <https://www.msmt.cz/vzdelavani/zakladni-vzdelavani/opatreni-ministra-skolstvi-mladeze-a-telovychovy-informatika>.
2. **Plaga, Robert.** Dopis ředitelům a ředitelkám ohledně úprav RVP. [Online] Leden 2021. [Citace: 12. Únor 2022.] https://www.msmt.cz/file/54867_1_1/.
3. **Jihočeská univerzita v Českých Budějovicích.** Učebnice a vzdělávací materiály pro školy. *Informatické myšlení*. [Online] 2018. [Citace: 12. Únor 2022.] <https://imysleni.cz/ucebnice>.
4. **Wikipedia foundation.** Micro Bit. *Wikipedia, The Free Encyclopedia*. [Online] 14. Únor 2022. [Citace: 23. Únor 2022.] https://en.wikipedia.org/wiki/Micro_Bit.
5. **Micro:bit Educational Foundation.** First steps. *BBC micro:bit*. [Online] 2016. [Citace: 20. Únor 2022.] <https://microbit.org/get-started/first-steps/introduction/>.
6. **Wostl, Marek.** blog zonepi. *beta blog zonepi*. [Online] 1. Prosinec 2020. [Citace: 24. Únor 2022.] <https://blog.zonepi.cz/microbit-v1-vs-microbit-v2/>.
7. **Microsoft.** Makecode editor. *Microsoft Makecode*. [Online] 2022. [Citace: 12. Březen 2022.] <https://makecode.microbit.org/#editor>.
8. **Wikipedia Foundation.** Diakritické znaménko. *Wikipedie otevřená encyklopedie*. [Online] 16. Září 2021. [Citace: 13. Březen 2022.] https://cs.wikipedia.org/wiki/Diakritick%C3%A9_znam%C3%A9nko.
9. **NÚV - Národní ústav pro vzdělání.** RVP ZV 2021 s vyznačenými změnami. *DOKUMENT KE STAŽENÍ*. [Online] 11. Únor 2021. [Citace: 31. Leden 2022.] <https://www.nuv.cz/file/4982>.
10. **Microsoft.** Microbit Documentation. [Online] 2021. [Citace: 11. Březen 2022.] <https://makecode.microbit.org/docs>.

A. Seznam příložených souborů

Na přiloženém datovém nosiči se nacházejí následující soubory a složky:

- **SOČ2021-2022-Smutný-Hübnerová-Algoritmizace-s-Micro-bit.docx** – editovatelná verze dokumentace odborné práce na SOČ
- **SOČ2021-2022-Smutný-Hübnerová-Algoritmizace-s-Micro-bit.pdf** – tisknutelná verze dokumentace odborné práce na SOČ
- **README.md** – dokumentace rozšíření v „markdown“ jazyce s odkazy na GitHub
- **./images** - složka s obrázky obsažených v README.md
- **./main soubory** -složka s vyřešenými úlohami
- **./repositáře** – složka s aktuální kopií repositářů
- **./Metodicke_listy** – vypracované metodické listy pro učitele
- **./Pracovni_listy** – vypracované pracovní listy pro žáky