

## Projekt: HRA BLUDIŠTĚ

**ČASOVÁ NÁROČNOST:** 3 až 4 vyučovací hodiny

### VÝSTUPY V RVP:

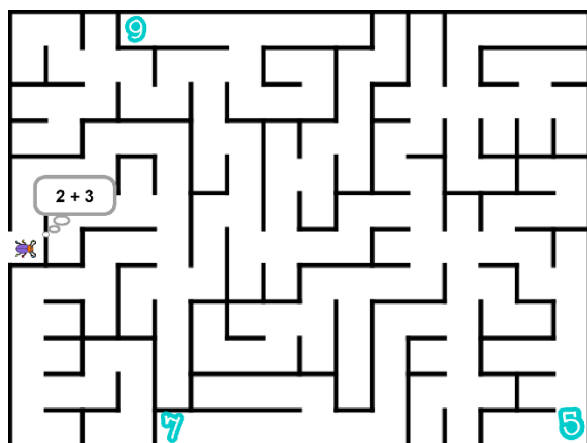
Data, informace a modelování: 1-3, 1-4

Algoritmizace a programování: 2-5, 2-6, 2-7, 2-8, 2-9, 2-10

Informační systémy: 3-12, 3-13, 3-14, 3-15, 3-16

### PŘEDSTAVENÍ PROJEKTU:

Tento projekt vychází z klasické hry „Bludiště“ a současně ji kombinuje s jednoduchými matematickými úlohami, které svou obtížností spadají do náplně matematiky 1. a 2. třídy ZŠ. Žáci druhého stupně tedy budou vlastně vytvářet „výukovou hru“ pro mladší děti.



Hlavní postavou hry je Brouk, který se po spuštění hry objevuje ve vchodu do bludiště. Úkolem hráče je Brouka dostat ven. Bludiště ale nemá klasický východ, jen je v něm umístěno několik číslic. Za jednou z nich je však východ z bludiště ukryt. Jedinou informací, kterou má Brouk k dispozici je jednoduchý matematický příklad (výraz), jehož řešení (výsledná hodnota) odpovídá číslici v bludišti, která východ skrývá. Hráč tak musí vypočítat příklad (určit hodnotu výrazu) a dovést Brouka

k dané číslici. Pokud se hráč zmýlí a dojde k nesprávné číslici, odečte se mu 1 bod. Když hráč dojde ke správné číslici, obdrží 2 body a pokračuje do dalšího levelu hry (nové bludiště, příklady i číslice). Po úspěšném zvládnutí druhého levelu hra končí a objevuje se obrazovka s nápisem „Victory“ a výsledným počtem bodů, které hráč získal.

Kompletní verze hry je v souboru *BLUDISTE-F06-Body.sb3*. Hra se spouští praporkem a pohyb Brouka se ovládá šipkami.

### STRUČNÁ CHARAKTERISTIKA FÁZÍ PROJEKTU:

Projekt je opět rozdělen do několika fází. Už druhá fáze projektu představuje funkční verzi jednoduché hry. Fáze následující pak postupně přinášejí („nabalují“) různá rozšíření, ale tak, že na konci každé fáze je vždy plně funkční hra. Tato koncepce dovoluje použití jen části projektu (např. v případě menší časové dotace) a současně podporuje i individualizaci výuky, resp. umožňuje, aby každý žák postupoval vlastním tempem a ukončil vývoj hry ve fázi, kterou stihne nebo zvládne.

V níže uvedeném seznamu fází je řada konkrétních údajů (počty, hodnoty apod.). Ty zde slouží jen jako příklad a nemusí být během realizace projektu s žáky respektovány. Texty, počty číslic, přidělované body a další parametry si mohou žáci určit sami.

1. **Pohyb v bludišti** – Brouk (postava) prochází bludištěm a nesmí projít stěnou.
2. **Matematický příklad** – V bludišti jsou umístěny 3 Číslice (postavy). Brouk si na začátku hry myslí jednoduchý matematický příklad. Hráč musí dovést Brouka bludištěm ke správnému číslu – výsledku příkladu. Každá Číslice reaguje na dotyk Brouka. Pokud Brouk dorazí k Číslici, která je správným řešením příkladu, řekne Číslice *Správně*, jinak řekne *Špatně*.
3. **Seznam příkladů** – Brouk na začátku vybere z připraveného seznamu 3 příkladů náhodně jeden z nich. Při dotyku Brouka s Číslicí musí Číslice nejprve vyhodnotit, zda výsledek zadaného příkladu odpovídá hodnotě, kterou Číslice představuje a podle toho řekne *Správně* nebo *Špatně*.
4. **Druhý level** – Když se Brouk dostane k Číslici představující správný výsledek příkladu, postoupí do dalšího levelu. Změní se bludiště, startovní pozice Brouka, příklady i číslice.
5. **Vítězství** – Když Brouk ve druhém levelu dojde k číslici, která je řešením příkladu, hra končí. Bludiště i Číslice zmizí a objeví se nápis *Victory*.
6. **Body** – Pokud hráč dovede Brouka ke správné číslici, obdrží 2 body. Pokud Brouka dovede k číslici, která nepředstavuje výsledek zadaného příkladu, je hráči odečten 1 bod. Konečný bodový stav se pak zobrazí na konci hry (u nápisu *Victory*; fáze 5. Vítězství).

Některé z fází projektu jsou navrženy tak, že je lze přeskochit. Nejsou na nich tedy závislé fáze následující. Není např. nutné vytvářet dva levely hry, počítat body, či na konci hry zobrazovat nápis *Victory*. Na druhou stranu, nic nebrání tomu, aby žáci vytvořili např. více než 2 levely, rozšířili hru o další prvky, postavy apod.

Poslední fáze – **Body** může být z logiky věci zařazena v podstatě kamkoli za fázi 2 – **Matematický příklad**. Poslední fáze ale k záměru projektu (viz níže *Co se žáci naučí nebo procvičí*) přispívá jen okrajově. Jde vlastně jen o jednoduchou práci s proměnnou. Proto byla po úvaze zařazena na závěr jako „bonusová“ pro rychlejší žáky.

Řešení jednotlivých fází je k dispozici pro učitele ke stažení nebo ke spuštění v online prostředí Scratch pod názvem **PROJEKT: Bludiště**. Materiál po žáky není členěn podle jednotlivých snímků, jako je tomu u předprojektů, ale je rozdělen na popis fází projektu, základní nastavení hry a rozbor jednotlivých fází.

#### CO BUDEME DĚLAT:

- ovládání Brouka s omezením na cestu danou bludištěm (nesmí projít stěnou bludiště)
- chování dalších postav při interakci s Broukem
- vybírat náhodně z připravené sady příkladů
- vytvořit další level hry
- vytvoříme bodování
- vytvoříme obrazovku „Vítězství“ s informací o získaných bodech

#### CO SE ŽÁCI NAUČÍ NEBO PROCVIČÍ:

- rozložit hru na menší funkční celky
- pracovat s více seznamy

- využívat seznamy, mezi kterými je vztah (relace)
- rozesílat zprávy v klíčových momentech hry (přechod do dalšího levelu, konec hry)
- ve vyšších fázích: modifikovat stávající program (uvědomit si, kde všude se změny promítnou)
- řešit projekt po etapách (fázích), které umožňují kontrolu funkčnosti a teprve pak postoupit do další etapy (fáze projektu)

### POUŽITÝ PROJEKT:

Všechny podklady (soubory) jsou ve složce **projekt – BLUDIŠTĚ**.

- Ve složce pro učitele jsou soubory pro jednotlivé fáze projektu a metodika.
- Ve složce pro žáky je k dispozici výchozí projekt se třemi připravenými bludišti (3 pozadí scény) a pracovní listy ve formě prezentace.

### Pracovní list pro žáky – HRA BLUDIŠTĚ

## Hra Bludiště

Vytvoříme jednoduchou hru pro děti:  
**bludiště**, ve kterém **brouk** hledá cestu ven

- ven se dostane, pokud v bludišti najde správný číselný kód
- v bludišti bude rozmístěno několik čísel – kódů, ale jen jeden správný
- úkolem brouka bude dojít ke správnému kódu (k tomu dostane nápovědu – zadaný příklad, který musí vyřešit, aby určil správný kód)

Potom budeme hru postupně vylepšovat





### METODICKÉ POZNÁMKY a ŘEŠENÍ ÚLOHY:

Nejprve žákům promítneme úvodní slide, velmi stručně představíme projekt a následně ukážeme hotovou hru. Aby bylo zřejmé, že příklady jsou vybírány náhodně, je třeba hru před žáky zahrát aspoň dvakrát, popř. při druhé hře stačí jen začátek, aby bylo vidět, že byl zadán jiný příklad (bude-li náhodně vybrán stejný, spustě hru znovu praporkem). Následně s žáky rozebere prvky a situace, které ve hře nastávají. Vhodné je začít s viditelnými prvky (Brouk, Číslice, Bludiště). Prodiskutujeme s žáky, jak hra probíhá, co postavy dělají a jak to můžeme naprogramovat.

Následně probereme s žáky možný postup (fáze), jak hru postupně vytvářet. K tomu lze přistoupit v zásadě třemi způsoby:

- a) Během diskuze o fázích projektu je možné zaznamenávat žáky navrhované fáze (postup tvorby) a následně je společně zpřesňovat a upravovat jejich pořadí.
- b) Předložíme žákům přeházený seznam fází a necháme je rozmyslet, jak nejlépe půjdou fáze postupně za sebou, které fáze na sobě závisejí a zda jedna nepředbíhá jinou (funkčními prvky)
- c) Po diskuzi o jednotlivých prvcích hry žákům dáme k dispozici hotový seznam fází projektu a diskutujeme s nimi nad jejich řešením a dalšími možnostmi zpracování hry.

V případě, že uznáte za vhodné, nechte každého žáka postupovat vlastním způsobem.

#### **OTÁZKY UČITELE:**

- Každá hra se dá rozdělit na menší části, které společně zajišťují všechny funkce ve hře.
  - o Jaká je funkce jednotlivých postav hry a jak ji naprogramujeme?
  - o Kde začneme u této hry?
  - o Jaké informace budou postavy sdílet a jaké si budou předávat?
- Můžeme hru nějak zjednodušit? Které fáze můžeme vynechat, aniž by se to promítlo do jiných?
- Jak můžeme hru naopak doplnit nebo rozšířit?

#### **MOŽNÉ POTÍŽE:**

Zvolíme-li v metodice postup dle bodu a), může u některých žáků nastat situace:

- *žák nedokáže rozdělit hru na menší funkční celky* – Dejte žákovi k dispozici jednotlivé fáze projektu v přeházeném pořadí, ať se pokusí pořadí určit.

#### **ZÁVĚR:**

Žák by si měl v závislosti na rozboru hry uvědomit nutnost dělení celé hry na menší funkční celky, které může vždy otestovat před postupem do další fáze (to mj. závisí též na jeho předchozích zkušenostech; u méně zkušených k uvědomění dojde postupně až při samotné realizaci projektů, resp. jeho fází). Současně by měl žák chápat, že některé fáze hry jsou klíčové a nelze bez nich smysluplně pokračovat dále.

## Fáze 1 – Jednoduchá hra

### Základní informace:

- Stáhni si a otevři projekt **Bludiste** ([link](#)).
- Vyber si bludiště (pozadí) a přidej postavu **brouka**.
- Uprav velikost postavy, aby se vešly do bludiště.



### Co musíš vyřešit:

- Ovládání **brouka** klávesami (šipky nebo WSAD).
- Brouk nesmí projít stěnou bludiště!
- Při začátku další hry se **brouk** musí vrátit na start.

Využij bloky:

po stisku klávesy šipka vpravo

když tak

opakuj stále

natoč se směrem

dotýkáš se barvy ?

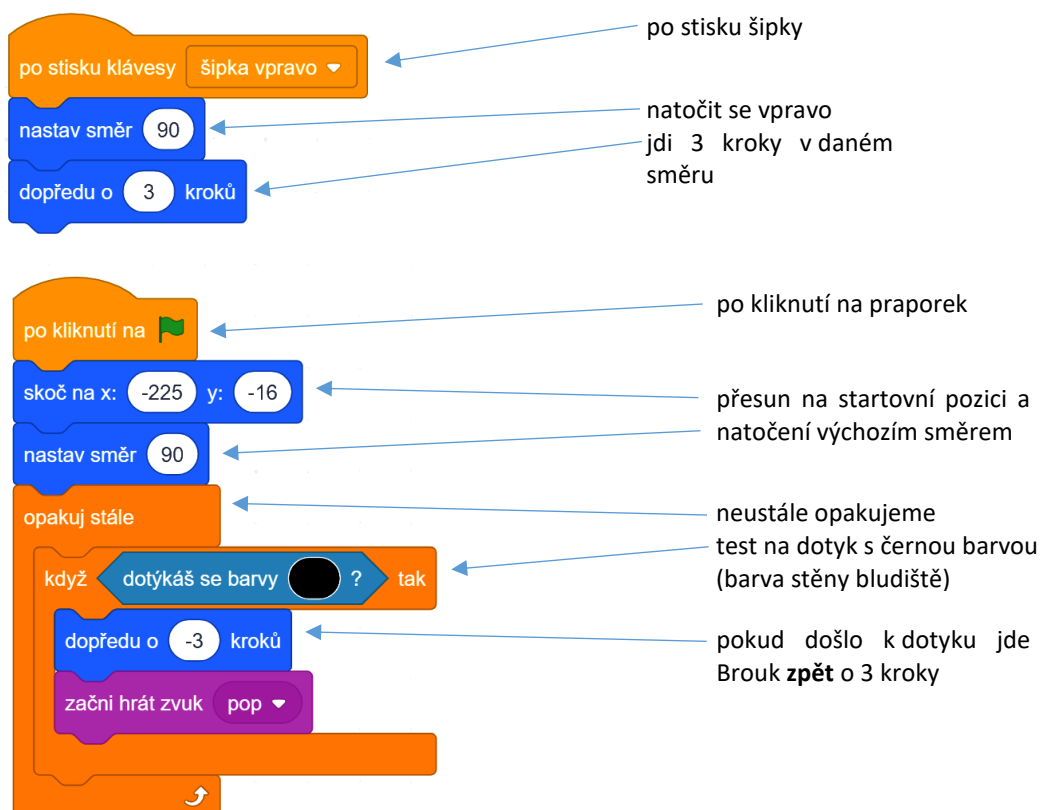
dopředu o kroků

skoč na pozici x: y:

**Nezapomeň průběžně ukládat !**

### ŘEŠENÍ ÚLOHY – varianta 1:

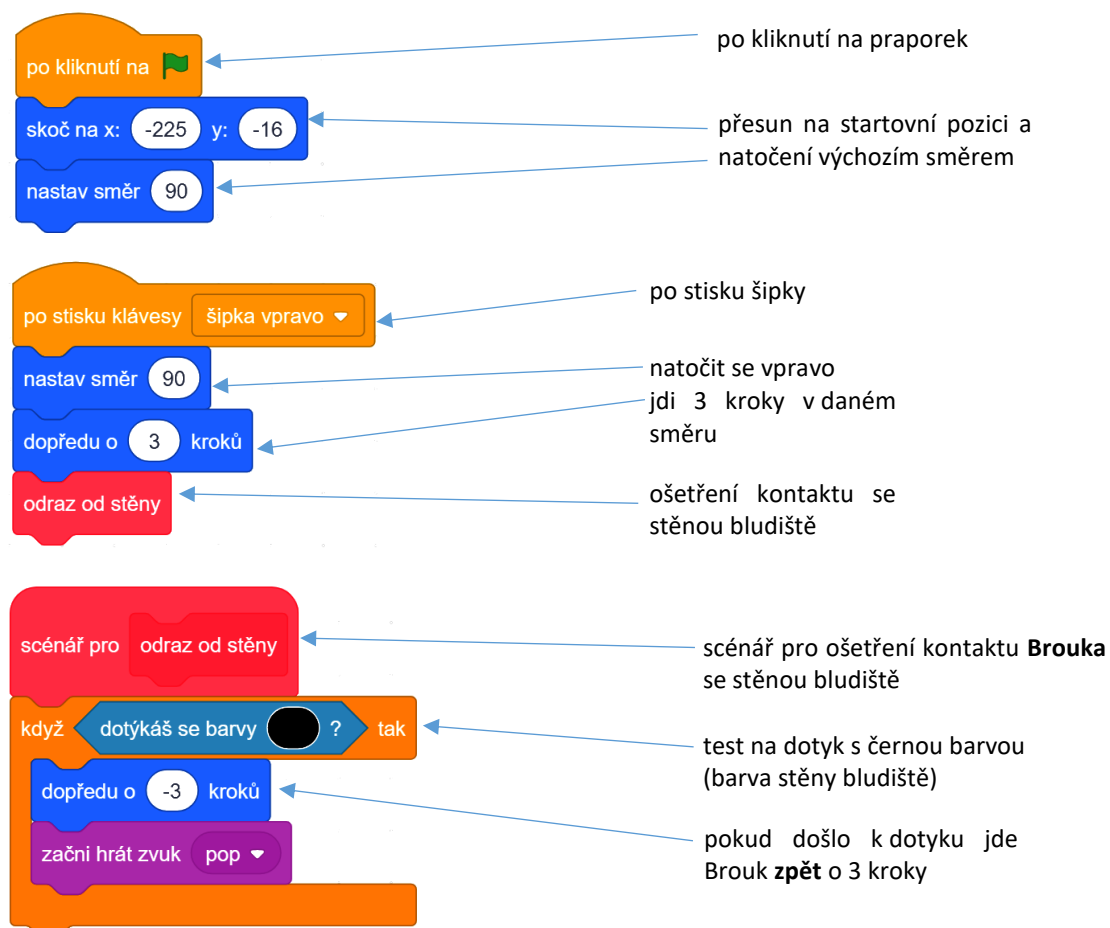
#### Brouk



## ŘEŠENÍ ÚLOHY – varianta 2:

Ve druhé variantě řešení je situace kontaktu Brouka se stěnou bludiště řešena s pomocí vytvořeného bloku „odraz od stěny“. V něm opět probíhá detekce barvy (černé stěny bludiště), ale neopakuje se v nekonečném cyklu („opakuj stále“). Blok „odraz od stěny“ je totiž prováděn po každém stisku ovládacích kláves (šipky).

### Brouk



## METODICKÉ POZNÁMKY:

Při přípravě projektu bylo uvažováno o tom, že si žáci nakreslí vlastní bludiště. Nicméně dosavadní zkušenosti ukazují, že to může způsobit řadu problémů. Většinou se v žákem vytvořených bludištích objevují úzké cesty, kterými postava neprojde. Další možné problémy souvisejí s tím, že Brouk je v této hře ovládán klávesami a pohybuje se tedy ve čtyřech základních směrech (0°, 90°, 180°, 270°), je tedy žádoucí, aby i stěny bludiště byly stejně orientovány. To většina žákovských výtvorů také nesplňuje.

Proto byl připraven výchozí projekt, který obsahuje 3 předem připravená bludiště. Tento projekt si žáci stáhnou a na něm mohou začít pracovat.

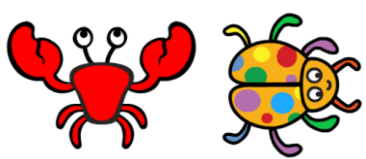
Pro tvorbu bludiště lze ale využít i specializované online nástroje (např.: [www.mazegenerator.net](http://www.mazegenerator.net)), které po zadání několika parametrů vygenerují bludiště, které netrpí problémy jako v případě volné

žakovské tvorby. Pokud se pro tuto variantu rozhodnete, je doporučeno si proces tvorby bludiště a zejm. jeho importu do Scratch vyzkoušet před tím, než se rozhodnete tuto možnost zařadit do výuky.

Pokud jde o postavu Brouka, lze žákům dovolit, aby si zvolili vlastní postavu, která se jim více líbí. Nicméně i výběr postavy nezřídka vyvolá otázky, zda její tvar je vhodný pro předpokládaný pohyb v bludišti. Po naprogramování pohybu vlastní postavy žáci během testování sami odhalí, zda je vhodná, či nikoli, resp. které tvarové aspekty jejich postavy jsou příčinou problémů při pohybu bludištěm.

Jako vhodné se jeví postavy, které jsou symetrické podle podélné osy a současně je jejich šířka srovnatelná s délkou. U postavy splňující tyto podmínky pak při pohybu bludištěm, dochází ke kontaktu se stěnou bludiště pouze při „čelním nárazu“, a nikoli za jiných okolností, např. při otočení během změny směru pohybu. Vhodné je také, aby vybrané postavy byly v pohledu shora.

Pokud budete mít pro výuku omezený čas, můžete, kromě Brouka doporučit žákům výběr např. z následujících postav:



Při pilotním ověřování byly do výběru zařazeny i postavy Kočka a Myš. Přestože jejich proporce do jisté míry vyhovují doporučení, docházelo při otáčení ke kontaktu ocásku se stěnou bludiště. Tím se splnila podmínka testující dotyk se stěnou a aktivoval se příkaz *dopředu o -3 kroků* (couvání o 3 kroky zpět) a protože ocásek je v zadní části postavy, došlo k tomu, že u prvního řešení (s nekonečným cyklem) se postava „procouvala“ stěnou bludiště a u druhého (s novým blokem) zůstala zaseklá ve stěně bludiště.



Žáci narazí na řadu menších problémů. U většinu z nich by však měli být schopni v rámci testování nalézt příčiny a ošetřit je.

Specifickým problémem, který pravděpodobně nevyřeší žáci sami, může být návrat postavy na startovní pozici po opětovném spuštění hry praporkem. Pokud žáci nemají s podobným problémem zkušenosti, umístí nejprve Brouka pomocí myši na startovní pozici a následně je seznámíme s tím, kde zjistí souřadnice, na kterých se Brouk nachází a směr, kterým je natočen (viz ŘEŠENÍ ÚLOHY).

#### OTÁZKY UČITELE:

- Jakou velikost kroku nastavíš pro Brouka?
- Jak vyřešíš, aby brouk neprošel stěnou bludiště?

#### MOŽNÉ POTÍŽE:

- *žák nebude vědět, kam si uložil hru* – předcházejte tomuto problém již na začátku upozorněním na konkrétní prostor, kam žáci své projekty ukládají; pokud již obtíže nastanou, hledejte v posledních spuštěných dokumentech Scratch.

- *žák nerozumí nekonečnému cyklu* – na konkrétním příkladu ze života ukažte žákovi nekonečný cyklus (např. střídání ročních období, pohyb sekundové ručičky u hodin)
- *žák nerozumí pozici – umístění objektu* – pomozte žákovi pomoci čtverečkovaného papíru a poloze bodu v kartézské soustavě souřadnic

## ZÁVĚR:

Žák má vytvořený svůj vlastní projekt s Broukem (či vlastní postavou), kterého ovládá pomocí kláves a který se pohybuje bludištěm. Pohyb Brouka je omezen stěnami bludiště.

## Pracovní list pro žáky – FÁZE 2

### Fáze 2 – Matematický příklad

**Základní informace:**

- **Brouk** si na začátku myslí příklad.
- Hráč musí brouka dovést ke správnému výsledku.
- Pokud hráč dovede brouka ke správnému výsledku, **číslice** řekne: *Správně!*
- Pokud hráč dovede brouka k nesprávnému výsledku, **číslice** řekne: *Špatně!*

**Co musíš vyřešit:**

- Když brouk dojde k číslici, dostane odpověď.

**Využij bloky:**

dotýká se Beetle ▼ ?

myšlenka ○ ○ sekund

čekej dokud nenastane

bublina ○ ○ sekund

## METODICKÉ POZNÁMKY:

Tato fáze je velmi jednoduchá a je přípravou pro následující 3. fázi. Žáci zde umístí 3 Číslice (jde o objekty, resp. postavy z nabídky Scratch) a vymyslí si příklad, který si Brouk na začátku hry myslí (příkaz *mysli si*) nebo řekne (příkaz *říkej*). Je samozřejmé, že alespoň jedna z Číslic musí být současně řešením příkladu.

Problémy, které by žáci nezvládli identifikovat a vyřešit samostatně, by se neměly v této fázi objevit.

## ŘEŠENÍ ÚLOHY (pro matematický příklad 2 + 3):

### Brouk (doplnění):

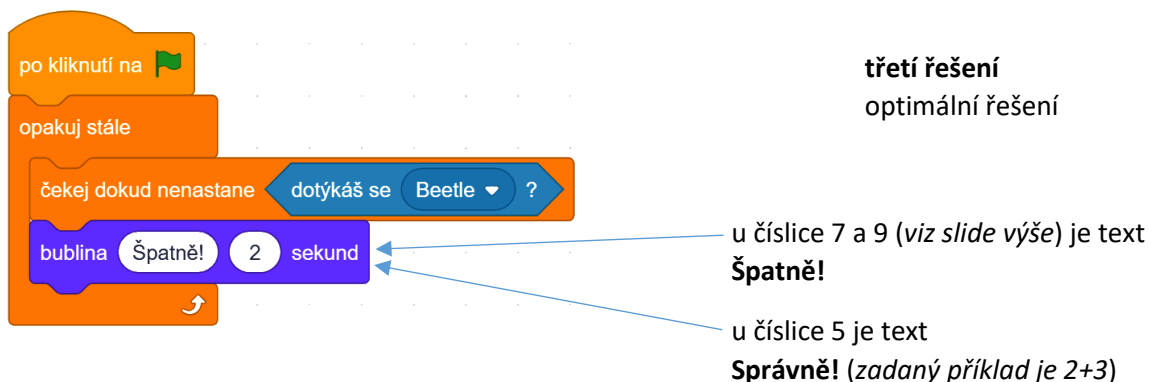
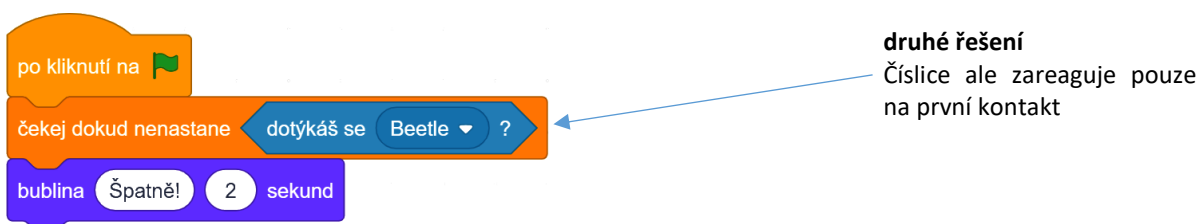
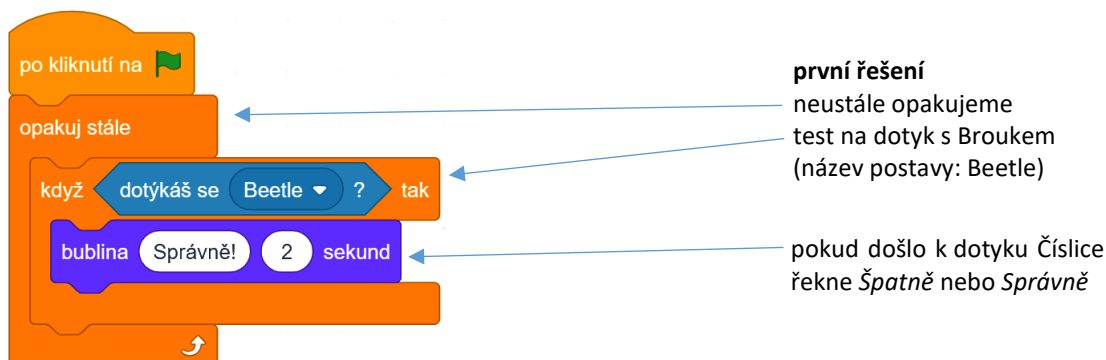
po kliknutí na

myšlenka 2 + 3 3 sekund

po startu hry Brouk řekne příklad

## Číslice:

V souboru BLUDISTE-F02-Priklady.sb3 je u každé číslice jiné řešení (viz níže).



## ZÁVĚR:

Žák má vytvořený svůj vlastní projekt, s Broukem (či vlastní postavou), kterého ovládá pomocí kláves a který se pohybuje bludištěm. Pohyb Brouka je omezen stěnami bludiště. Na začátku hry Brouk řekne zadání matematického příkladu. Číslice reagují na kontakt s Broukem. Pokud Číslice, které se Brouk dotkne, představuje správné řešení příkladu, řekne *Správně!*, jinak řekne *Špatně!*.

## Fáze 3 – Seznam příkladů

**Základní informace:**

- Brouk si myslí příklad, který náhodně vybere ze seznamu 3 příkladů.
- Před spuštěním hry proto nevíme, co si brouk vybere a jaké číslo bude správným výsledkem.

**Co musíš vyřešit:**

- Vytvořit seznam s příklady a seznam s výsledky.
- Brouk vybere náhodně číslo mezi 1 a 3 a uloží do **X**.
- Brouk si myslí příklad č. **X** ze seznamu příkladů.
- Až brouk dojde k číslici, musí číslice ze seznamu výsledků zjistit, jestli je správným výsledkem.

**Využij bloky:**

nastav CisloPříkladu na náhodné číslo od 1 do 3

když tak

prvek z tady vybereš seznam

jinak

CisloPříkladu

### METODICKÉ POZNÁMKY:

Třetí fáze hry je nejobtížnější, protože přichází s náhodným výběrem úlohy ze seznamu příkladů. K seznamu příkladů je pak navázán další seznam – seznam výsledků. Žák tedy musí pochopit vazbu mezi seznamy (viz níže).

Zásadní otázkou této fáze je, jak s žáky vlastně dojít k řešení s pomocí seznamů. Pokud ponecháme diskuzi o řešení náhodného zadávání příkladů zcela bez omezení, můžeme se dočkat řady pokusů žáků využívajících např. náhodných čísel, které v důsledku vyžadují velmi promyšlené a komplikované řešení, popř. jsou ve Scratchi neřešitelné (viz následující poznámka z ověřování projektu).

Pozn.: V prvních verzích tohoto projektu byly zadávané příklady koncipovány tak, aby hru mohli hrát i děti z první třídy. Šlo tedy o příklady na sčítání s jednociferným výsledkem. Při ověřování žáci typicky nejprve nabídli náhodné příklady na sčítání s náhodným výběrem dvou sčítanců. Sami pak přišli na to, že náhodný výběr sčítanců může dát výsledek, který neodpovídá žádné z číslic v bludišti. Jiným řešením, se kterým žáci následně přišli, byl náhodný výběr jednoho ze sčítanců a dopočítání druhého tak, aby vyšel předem určený výsledek (Číslo je v bludišti). V takovém případě byl při každém spuštění hry sice jiný příklad, ale vždy stejný výsledek. Objevili se i složitější návrhy – do bludiště umístíme např. devět číslic (0 – 8) a následně budeme náhodně vybírat dva sčítance z rozsahu 0 – 4.

Po zkušenostech z ověřování byl projekt upraven tak, že předpokládaný hráč již nebude prvňáček, ale žák 1. až 2. třídy (tím byly připsány další matematické operace) a upravena metodika (viz níže).

Vhodným způsobem, jak nasměrovat žáky k seznamům, je nejprve je nechat vypracovat alespoň deset příkladů vhodných pro děti z 1. a 2. třídy ZŠ (např. na sčítání, odčítání, jednoduché násobení a celočíselné dělení) s tím, že výsledky by měly odpovídat číslicím, které si v předchozí fázi vybrali do svého bludiště. Lze předpokládat, že v navržených příkladech budou zastoupeny různé matematické operace i počet činitelů ( $3+2$ ,  $2*3-1$ , ...). Příklady je pak třeba prodiskutovat zejména s ohledem na schopnosti dětí a v 1. a 2. třídě, zkorrigovat

extenzivní návrhy a zkontrolovat. Následně s žáky diskutujeme, jak jimi navržené příklady implementovat. Vědomí různorodosti příkladů povede žáky k úvahám o jiných řešeních, než je výše zmíněné náhodné generování sčítanců apod. Žáci by navíc měli mít vyřešen i předporojekt se seznamy a podle zkušeností z ověřování sami s řešením pomocí seznamů přijdou. Pokud se tak nestane, je na učiteli, aby je inspiroval (často stačí jen zmínit slovo „seznam“).

Při diskuzi implementace náhodných příkladů s pomocí seznamů je vhodné probrat všechny aspekty – dílčí kroky, které jsou k realizaci potřeba. Můžeme nejprve nechat žáky samostatně nebo v menších skupinkách analyzovat celý záměr, pak sepsat na tabuli výsledky a nápady všech skupin a následně v diskuzi pomocí otázek učitele doplnit chybějící kroky a jejich řešení. Jiným přístupem může být rovnou řízená diskuze a kladení otázek učitelem.

### **OTÁZKY UČITELE:**

Otázky, které musí být v každém případě nakonec zodpovězeny jsou v následujícím přehledu:

#### **1. Jak bude vybrán náhodný příklad se seznamu? Co budeme muset udělat?**

*Použijeme náhodné číslo v rozsahu odpovídajícímu délce seznamu (např. pro 10 příkladů – náhodné číslo z rozsahu 1 – 10). Následně vybereme položku seznamu – příklad na odpovídající pozici (seznamy mají číslované položky – viz obr. níže)*

##### **1.1. Kde výběr příkladu ze seznamu naprogramujeme?**

*Nejvhodnější je to provést u Brouka, ten má na počátku hry příklad zobrazit v bublině.*

#### **2. Jak rozhodneme, zda Brouk dorazil ke správné Číslici?**

*Musíme zkontrolovat, zda má Číslice hodnotu odpovídající výsledku příkladu. Číslice samozřejmě zná svou hodnotu.*

##### **2.1. Jak zjistíme výsledek příkladu?**

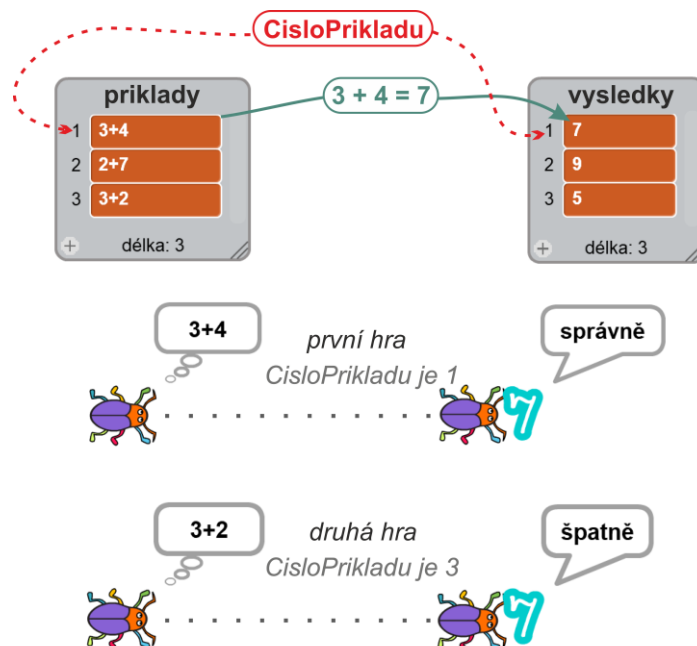
*Scratch nedokáže příklad zapsaný v seznamu spočítat. Musíme proto mít výsledky někde předem uloženy. Řešením je seznam výsledků. Budeme tedy mít dva seznamy, jeden se zadáním příkladů a jeden s výsledky, které budou spárované.*

##### **2.2. Budeme programovat u Brouka nebo u Číslice?**

*Programování u Brouka je problematické, brouk totiž může narazit do kterékoli Číslice v bludišti a bylo by tak nutné ošetřit všechny Číslice. Při změně Číslic nebo doplnění další do bludiště, by žáci museli upravovat i program Brouka. Naproti tomu programování u Číslice řeší jen kontakt s Broukem. Pokud Brouk do Číslice narazí lze rovnou vyhodnotit, zda je Číslice správným výsledkem.*

##### **2.3. Jak Číslice „najde“ odpovídající výsledek zadaného příkladu v seznamu výsledků?**

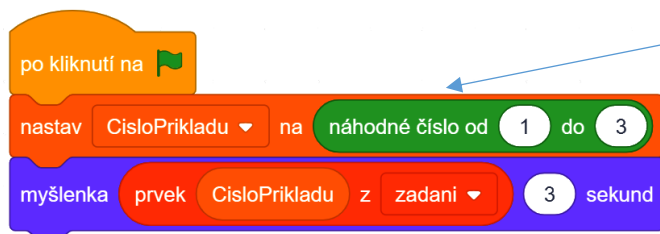
*Náhodné číslo, které bylo použito pro náhodný výběr příkladu ze seznamu, si musíme uložit do proměnné. Tu pak použijeme pro výběr čísla ze seznamu výsledků.*



Obr. 1. V horní části je ilustrována vazba mezi spárovanými seznamy. Náhodné číslo  $X$  pak slouží k výběru položek ze seznamu příkladů (který Brouk na začátku zobrazí v bublině) a následně ze seznamu výsledků (když Brouk narazí na číslici a ta porovná výsledek s vlastní hodnotou). Spodní část zobrazuje dva příklady. Jeden pro náhodné číslo 3 (uložené v  $X$ ) a druhý pro náhodné číslo 1 (uložené v  $X$ ).

## ŘEŠENÍ ÚLOHY:

**Brouk (přepracování části se zadáním příkladu):**

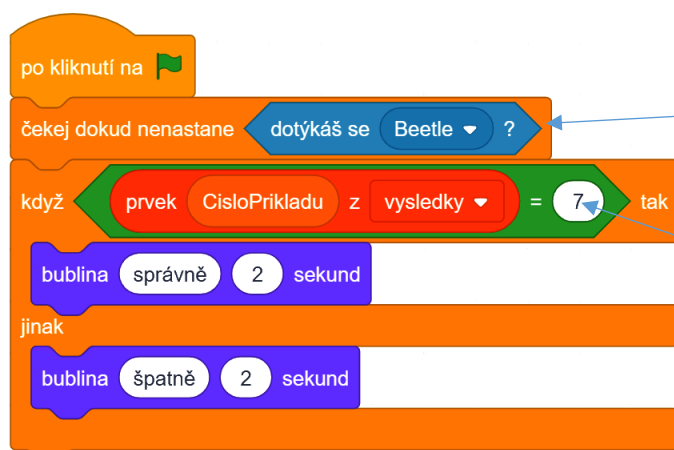


vybereme náhodné číslo v rozsahu dle počtu příkladů a uložíme do **CisloPříkladu**

proměnnou **CisloPříkladu** je třeba nejprve vytvořit na záložce Data

text, který si Brouk myslí vybereme pomocí **CisloPříkladu** ze seznamu příkladů

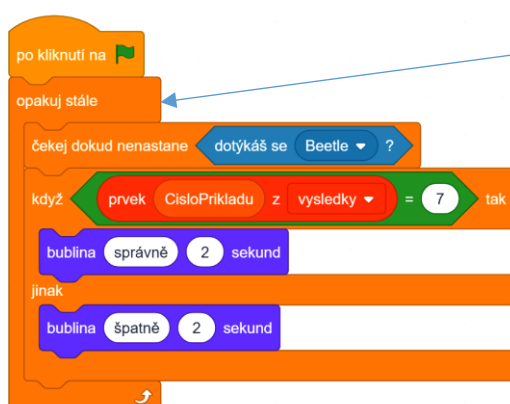
## Číslice (přepřacování části s kontrolou):



### Ukázka pro Číslici 7

čekáme, dokud nenastane kontakt Brouka s Číslicí

ze seznamu výsledků vybereme výsledek - prvek **CisloPříkladu** a porovnáme ho s hodnotou číslice



Celou sekvenci od detekce dotyku po porovnání výsledku s hodnotou číslice můžeme vložit do bloku **opakuji stále**. Číslice pak budou reagovat na opakovaný kontakt s Broukem.

## MOŽNÉ POTÍŽE:

- *žák nerozumí potřebě dvou seznamů* – Žák navrhuje využít jen jeden seznam, kde bude příklad i s řešením ( $2 + 3 = 5$ ). Nechte žáka zkusit nebo s ním proberte, jak potom bude vypadat zadání na začátku hry (příklad má zobrazit Brouk v bublině).
- *žák nerozumí vazbě mezi seznamy* – Ukažte žákovi ilustraci s vazbou mezi seznamy (viz obr. 1) nebo ji nakreslete na tabuli (kreslení na tabuli učitelem se zdá být vhodnější – žák může sledovat proces a sled úvah).

## ZÁVĚR:

Žák má vytvořený svůj vlastní projekt, s Broukem (či vlastní postavou), kterého ovládá pomocí kláves a který se pohybuje bludištěm. Pohyb Brouka je omezen stěnami bludiště. Na začátku hry je ze seznamu příkladů vybrán jeden příklad náhodně, který pak Brouk zobrazí v bublině. Číslo vybraného příkladu se uloží do proměnné X. Číslice, které se Brouk dotkne, porovná svou hodnotu s X-tým výsledkem se seznamu výsledků a na základě srovnání, řekne *Správně!*, nebo *Špatně!*.

## Fáze 4 – Druhý level

**Základní informace:**

- Když Brouk dojde ke správné Číslici, dostane se do dalšího levelu.
- Další level bude mít nové bludiště, nové příklady a nové Číslice.

**Co musíš vyřešit:**

- Vyměnit pozadí se starým bludištěm za nové.
- Skrýt staré Číslice a ukázat nové.
- Přesunout Brouka na start v novém bludišti.
- Přidat nové příklady a výsledky do seznamů.
- Brouk musí vybírat ze seznamu jen nové příklady.

**Využij bloky:**



The image shows several Scratch blocks. There are two orange 'when green flag clicked' blocks. The first one has a 'send message to b02' block and a 'go to level 2' block. The second one has a 'send message to b02' block and a 'go to level 2' block. There are also two purple 'when green flag clicked' blocks. The first one has a 'hide all sprites' block and a 'show all sprites' block. The second one has a 'hide all sprites' block and a 'show all sprites' block.

### METODICKÉ POZNÁMKY:

Přidání druhého levelu lze v zásadě rozdělit do dvou etap. První souvisí se skrytím prvků prvního levelu (Číslice), druhá s umístěním prvků nového levelu (nové Číslice, nové bludiště, nová startovní pozice Brouka). Přestože lze řešit obě etapy zároveň, doporučujeme, aby je žáci řešily odděleně. Postup ve dvou etapách umožní žákům spolehlivěji řešit všechny aspekty přechodu do dalšího levelu a uvědomit si pro žáky nečekané důsledky, které se projeví při znovuspuštění hry (např. skryté Číslice se po spuštění neobjeví).

**V první etapě** – skrytí Číslic by se žáci měli zorientovat zcela samostatně. Pokud to bude pro některé problém, lze s nimi individuálně prodiskutovat hlavní kroky řešení. K tomu se opět hodí otázky učitele:

- **Za jakých podmínek má nastat přechod do nového levelu?**  
*Když Brouk dojde ke správné Číslici.*
- **Jak zařídit schování všech starých Číslic? Kde se to naprogramuje?**  
*Skrytí objektu lze naprogramovat právě jen u daného objektu – objekt tedy může schovat sám sebe.*
- **Jak se Číslice „dozvědí“, že nastal přechod do vyššího levelu a že se mají schovat?**  
*Správná Číslice po oznámení „Správně!“ rozešle zprávu (v řešení níže jde o zprávu „postupDoLevel2“).*

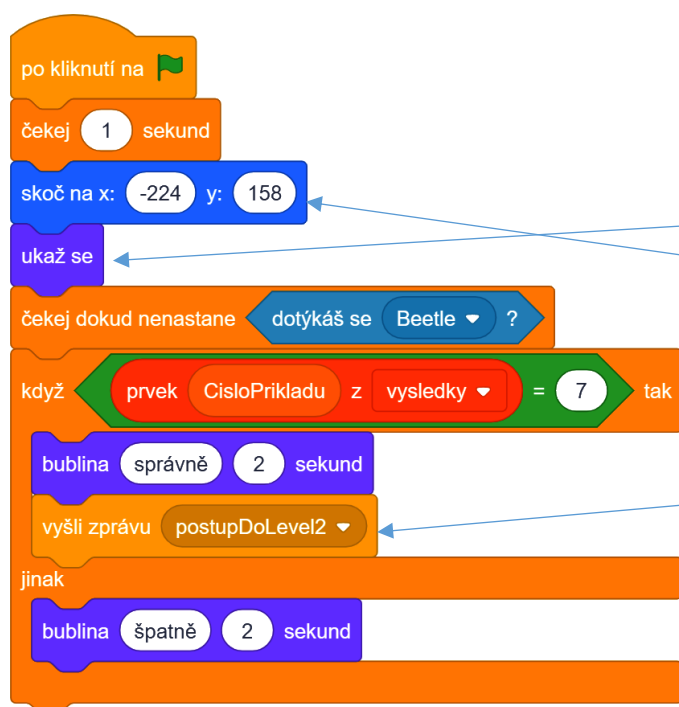
Žáci si obvykle neuvedomí, že Číslice zůstanou skryté i po opětovném spuštění hry. Proto je vhodné, aby si po dořešení první etapy a otestování, že Číslice zmizí, spustili hru znovu (než přistoupí k řešení druhé etapy). Pokud si neuvedomili, že číslice zůstanou skryté, uvidí to po spuštění hry.

**Ve druhé etapě** – (nové Číslice, nové Bludiště, nová pozice Brouka) jde o řešení podobných úkolů jako v první etapě. Po zvládnutí první etapy by tak ani s druhou neměli mít žáci problém. Hlavním rozdílem je, že Číslice druhé etapy jsou na počátku hry skryté a objevují se až po přijetí zprávy. Podobné je to i se změnou bludiště, která je realizována změnou pozadí. Nakonec je třeba ošetřit skrytí Číslic druhého levelu při opětovném spuštění hry.

Nové příklady pro druhý level mohou žáci řešit dvěma způsoby. Pravděpodobně přijdou s návrhem vytvořit nový seznam příkladů a nový seznam výsledků. Řešení je pak analogické jako v předchozí fázi projektu. Dalším způsobem je rozšíření seznamů o nové položky (např. doplnění 5 příkladů a 5 výsledků). Náhodné číslo (ukládáné do proměnné X) je pak třeba vybírat z intervalu, který odpovídá číslům (indexům) nově přidávaných položek.

## ŘEŠENÍ ÚLOHY:

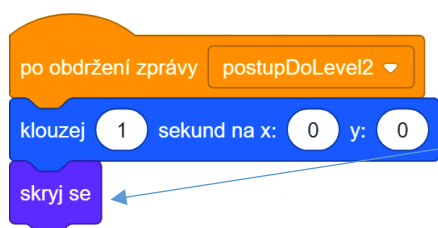
### Číslice prvního levelu (úprava pro přechod do druhého levelu)



po skrytí Číslice při přechodu do druhého levelu, je třeba ji při dalším spuštění hry znovu ukázat

návrat na výchozí pozici po dalším spuštění hry (důvod viz další blok)

V případě, že číslice je řešením zadaného příkladu, rozešle zprávu *postupDoLevel2*



po obdržení zprávy *postupDoLevel2* se číslice skryje

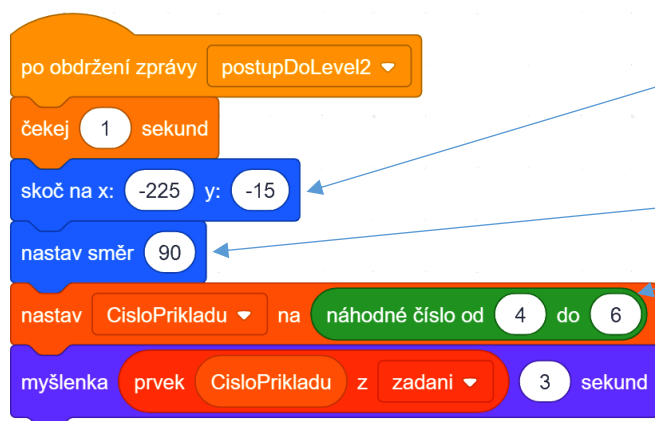
před skrytím lze použít efekt, zde např. klouzání číslic na střed bludiště (souvisí s předchozím blokem, kde musíme vrátit číslici na původní pozici při novém spuštění hry)

### Seznamy (rozšíření o další příklady a výsledky):

zadani	vysledky
1 3+4	1 7
2 2+7	2 9
3 3+2	3 5
4 8-2	4 6
5 6-3	5 3
6 9-1	6 8

ukázka rozšíření seznamů o trojici příkladů a výsledků

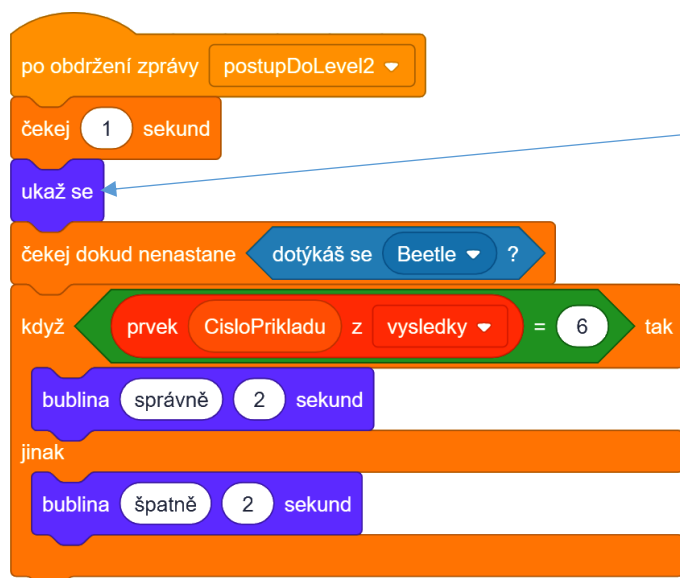
## Brouk (doplnění pro druhý level)



po obdržení zprávy postupDoLevel2 se Brouk musí přesunout na novou pozici – startovní pozici v novém bludišti (viz níže řešení Bludiště) a natočit se výchozím směrem

náhodný výběr příkladu pro druhý level (vybíráme jen nové příklady, tedy v rozsahu 4 až 6)

## nové Číslice (číslice pro druhý level)



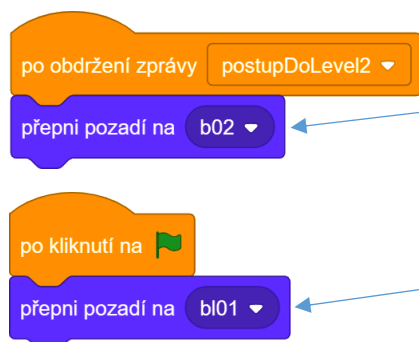
po obdržení zprávy postupDoLevel2 se nová Číslice poprvé objeví

ostatní příkazy jsou analogické jako u číslic prvního levelu



při spuštění hry se musí Číslice druhého levelu skrýt

## Pozadí



změna pozadí při přechodu do druhého levelu

při (dalším) spuštění hry se nastaví znovu první bludiště

## MOŽNÉ POTÍŽE:

- *žák má problémy s ošetřením všech situací, které souvisejí se skrýváním a objevováním objektů* – Probereme s žákem jednotlivé fáze hry – a) začátek hry, b) první level, c) přechod do druhého levelu, d) druhý level a sepíšeme, kdy je který objekt vidět, resp. kdy se má objevit a kdy (při jaké události) se má skrýt.

## ZÁVĚR:

Žák má vytvořený projekt s Broukem (či vlastní postavou), kterého ovládá pomocí kláves a který se pohybuje bludištěm. Pohyb Brouka je omezen stěnami bludiště. Na začátku hry je ze seznamu příkladů vybrán jeden příklad náhodně, který pak Brouk zobrazí v bublině. Číslo vybraného příkladu se uloží do proměnné X. Číslice, které se Brouk dotkne, porovná svou hodnotu s X-tým výsledkem se seznamu výsledků a na základě srovnání, řekne *Správně!*, nebo *Špatně!*. Pokud Brouk dorazí ke správné Číslici (řekne *Správně!*), dojde k postupu do dalšího levelu s novým bludištěm, novými Číslicemi a novou sadou příkladů, z nichž se bude vybírat opět náhodně.

## Pracovní list pro žáky – FÁZE 5

### Fáze 5 – Vítězství

**Základní informace:**

- Pro zapomnětlivé hráče je třeba přidat možnost znovu zobrazit příklad (jen ve druhém levelu).
- Až brouk dorazí v posledním levelu ke správné číslici, hra se ukončí oznámením *Victory*.

**Co musíš vyřešit:**

- Vymyslet způsob, jak znovu zobrazit zadaný příklad, např. po stisku mezerníku.
- Skrýt číslice, vyměnit pozadí.
- Zvětšit a přesunout brouka nad nápis *Victory*.
- Co se stane, když hru po skončení znovu spustíš?
- Jak to vyřešit?



Využij bloky:

nastav velikost na  %

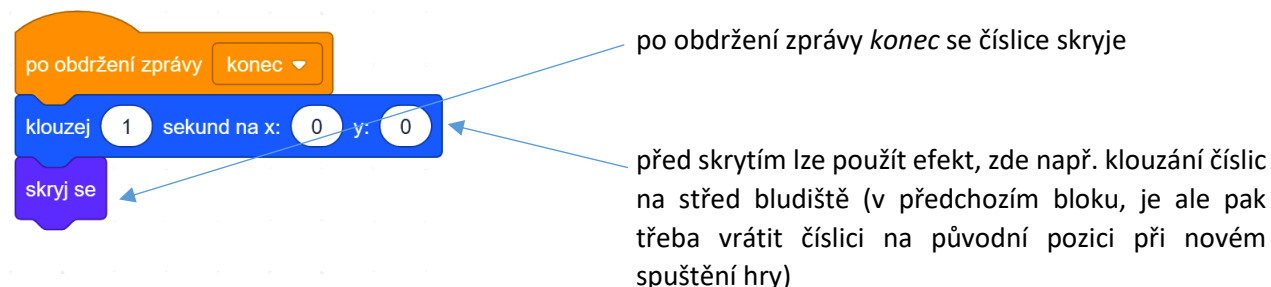
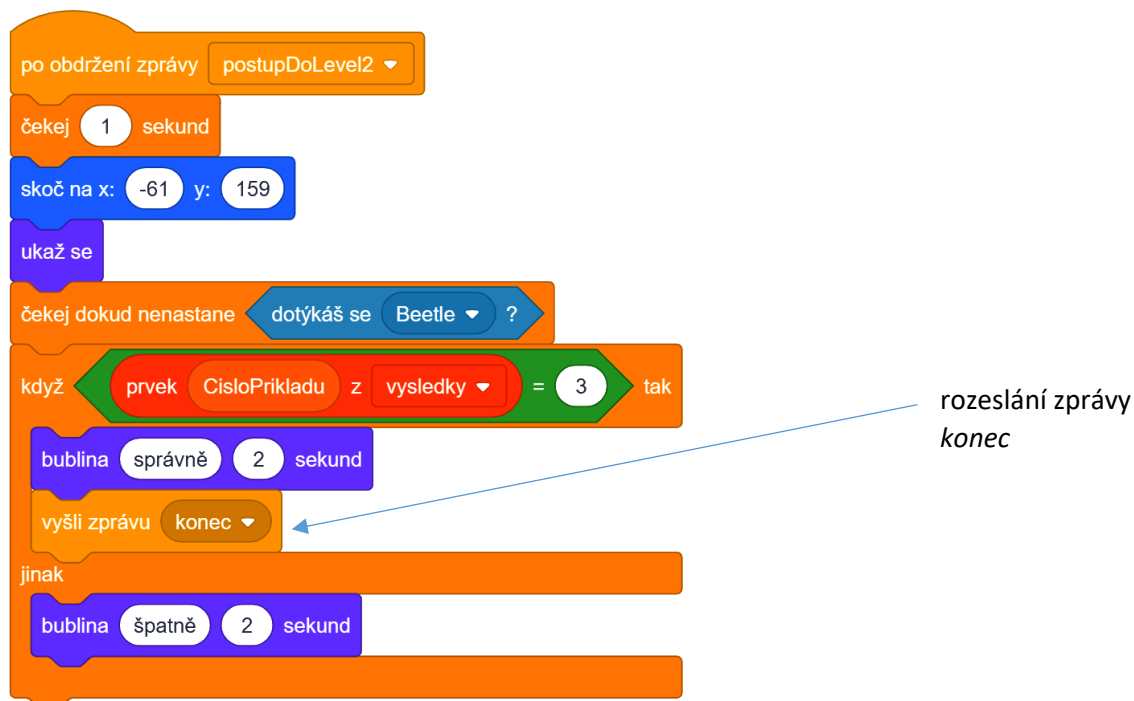
## METODICKÉ POZNÁMKY:

Pátá fáze by neměla být obtížná. K řešení se používají typově stejné konstrukce – rozeslání zprávy (v ukázce řešení zpráva „konec“), skrytí objektů a změna pozadí. Jak bude závěrečná obrazovka vypadat, můžeme ponechat zcela v režii žáků. Žáci mohou zařadit zvukové efekty, jednoduché animace apod. Pokud jde o nápisy, mohou se objevit problémy s češtinou, proto byl v ukázce zvolen text v angličtině.

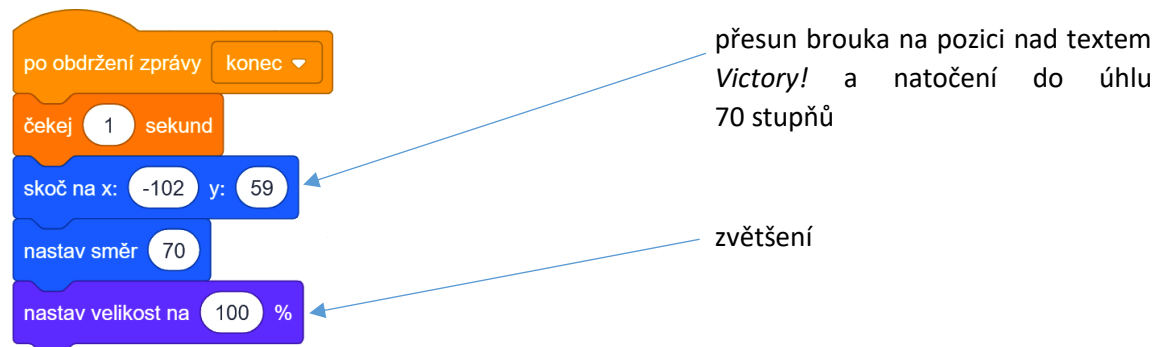
## ŘEŠENÍ ÚLOHY:

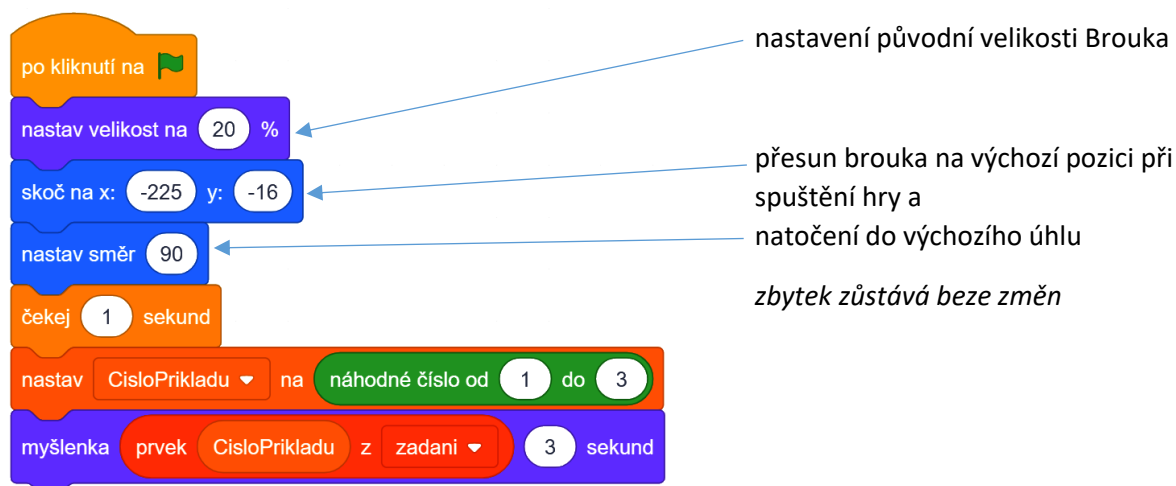
V ukázkovém řešení je vedle nápisu *Victory* umístěna postava Brouka, který se na danou pozici přesunul a zvětšil na 100 %.

### Číslice (doplnění konce hry)

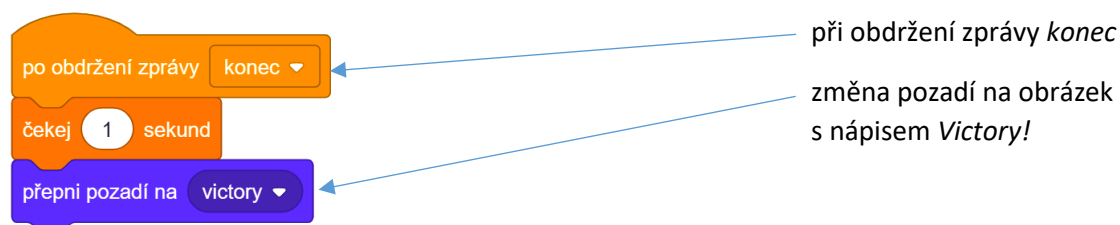


### Brouk (doplnění konce hry)





### Scéna (doplnění na konci hry)



### ZÁVĚR:

Žák má vytvořený projekt s Broukem (či vlastní postavou), kterého ovládá pomocí kláves a který se pohybuje bludištěm (je omezen stěnami bludiště). Na začátku hry je ze seznamu příkladů vybrán jeden příklad náhodně, který pak Brouk zobrazí v bublině. Číslo vybraného příkladu se uloží do proměnné X. Číslice, které se Brouk dotkne, porovná svou hodnotu s X-tým výsledkem se seznamu výsledků a na základě srovnání, řekne *Správně!*, nebo *Špatně!*. Pokud Brouk dorazí ke správné Číslici (řekne *Správně!*), dojde k postupu do dalšího levelu s novým bludištěm, novými Číslicemi a novou sadou příkladů, z nichž se bude vybírat opět náhodně. Když hráč dovede ve druhém levelu Brouka ke správné Číslici, zobrazí se obrazovka s nápisem *Victory!*

## Fáze 6 – Body

- **Základní informace:**
  - Aby mohli hráči soutěžit, přidáme bodování
  - Když brouk dojde ke správné číslici, dostane 2 body
  - Když dojde ke špatné, odečte se mu 1 bod
  - Výsledek oznámí brouk na konci hry větou *Získal si ... bodů.*
- **Co musíš vyřešit:**
  - Jak doplnit program, aby se body:
    - správně přičítaly
    - správně odčítaly
  - Kde všude bude potřeba program upravit?
  - Jak sestavit větu *Získáno bodů: ... ?*
  - Co se stane s body, když hru po skončení znovu spustíš?
  - Jak to vyřešit?



Využij bloky:

Body

změň Body o

nastav Body na

spoj ☐

### METODICKÉ POZNÁMKY:

Šestá fáze představuje doplnění typické součásti her – skóre. Jde v podstatě o základní práci s proměnnou a o rozhodnutí, kde a kolik bodů se bude přičítat či odčítat. Žáci by měli mít možnost si konkrétní hodnoty přidělovaných bodů určit sami. Mohou přidělovat více bodů za správnou odpověď, nemusejí body odčítat při špatné odpovědi a mohou zvážit i bodování jiných situací, než je kontakt Brouka s Číslicí.

Během ověřování se objevila i řešení, kdy za správnou odpověď bylo přiděleno 50 bodů, za špatnou odečteno 25 bodů a za kontakt Brouka se stěnou bludiště byl odečten 1 bod.

Žáci si opět musí promyslet, při jakých událostech chtějí pracovat s body a kde (u jaké postavy) je nutné příslušnou implementaci provést. Po zkušenostech z předchozích fází projektu by to pro žáky neměl být problém. Pokud se přesto objeví potíže, lze žákům pomoci ujasnit si problém a jeho řešení pomocí otázek (viz sekce Otázky učitele).

### OTÁZKY UČITELE:

#### 1. Za co chceme body přidělovat a za co strhávat?

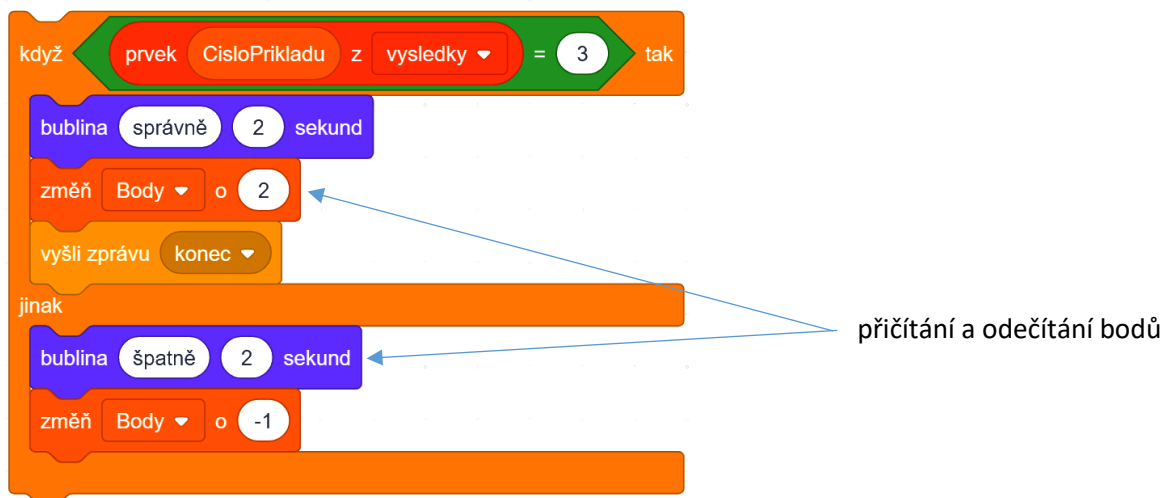
*Je vhodné, aby si každý žák nebo dvojice připravili soupis událostí např. na papír. V pravé části papíru by si měli nechat prostor (viz další otázky)*

#### 2. Kde budeme přidělování a strhávání bodů programovat?

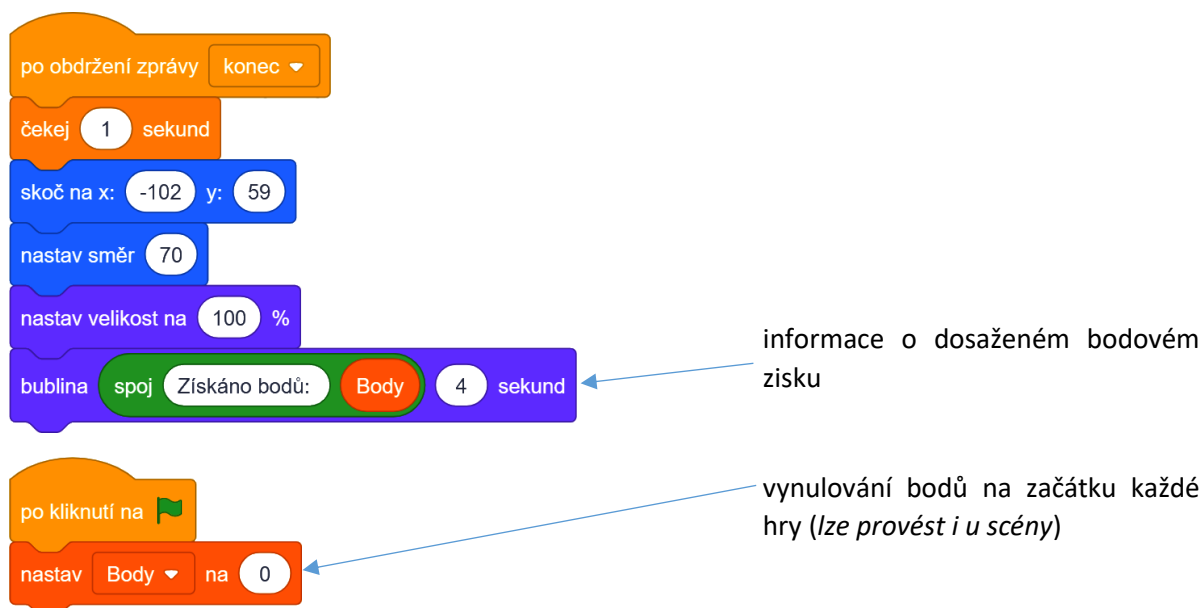
*Proměnná Body musí být globální (volba při vytváření proměnné: pro všechny postavy). Body typicky přidělujeme tam, kde je ošetřována příslušná událost (typicky u Číslic, kde přičítáme body za správnou odpověď a odčítáme za nesprávnou).*

## ŘEŠENÍ ÚLOHY:

### Číslice (doplnění bodů – ukázka pro Číslici druhého levelu)



### Brouk (doplnění bodů na konci hry)



## ZÁVĚR:

Žák má vytvořený projekt s Broukem (či vlastní postavou), kterého ovládá pomocí kláves a který se pohybuje bludištěm (je omezen stěnami bludiště). Na začátku hry je ze seznamu příkladů vybrán jeden příklad náhodně, který pak Brouk zobrazí v bublině. Číslo vybraného příkladu se uloží do proměnné X. Číslice, které se Brouk dotkne, porovná svou hodnotu s X-tým výsledkem se seznamu výsledků a na základě srovnání, řekne *Správně!*, nebo *Špatně!*. Pokud Brouk dorazí ke správné Číslici (řekne *Správně!*), dojde k postupu do dalšího levelu s novým bludištěm, novými Číslicemi a novou sadou příkladů, z nichž se bude vybírat opět náhodně. Když hráč dovede ve druhém levelu Brouka ke správné Číslici, zobrazí se obrazovka s nápisem *Victory!* a počtem získaných bodů.

Mentální aktivity žáků (Tabulka 3)
------------------------------------

<i>Vzory:</i>	předprojekt Nákupní seznam
<i>Abstrakce:</i>	náhodný výběr příkladů a vyhodnocování výsledků
<i>Dekompozice:</i>	rozložení hry do fází, ošetření přechodů mezi levely
<i>Logické myšlení:</i>	práce s dvěma seznamy (využití proměnné, která je indexem) a jejich využití ve více levelech hry
<i>Evaluce:</i>	nalezení funkčního řešení zadávání příkladů

Přístupy k řešení projektu (Tabulka 4)
--

<i>Vymýšlení, experimentování:</i>	jak „vyladit“ parametry pohybu postavy, aby bezproblémově procházel bludištěm (volba postavy, krok, velikost postavy, pohyb zpět)
<i>Navrhování, tvorba:</i>	návrhy příkladů (nebo jiných otázek), návrh systému bodování, návrh vlastního bludiště
<i>Ladění programu:</i>	při přechodu mezi levely a při znovuspuštění hry ošetřit stavy všech postav, pozadí a proměnných (viditelnost, pozice, velikost, hodnota)