

INOVATÍVNE METODIKY PRE VYUČOVANIE PROGRAMOVANIA V SCRATCHI PRE ZÁKLADNÉ ŠKOLY

MONIKA TOMCSÁNYIOVÁ, ANGELIKA HANESZ, ZUZANA TKÁČOVÁ



EURÓPSKA ÚNIA
Európsky sociálny fond
Európsky fond regionálneho rozvoja



OPERAČNÝ PROGRAM
ĽUDSKÉ ZDROJE



*Tento projekt sa realizuje vďaka podpore z Európskeho sociálneho fondu
v rámci Operačného programu Ľudské zdroje*

www.minedu.sk www.employment.gov.sk/sk/esf/ www.itakademia.sk

Inovatívne metodiky pre vyučovanie programovania v Scratchi pre základné školy

Spracované v rámci národného projektu IT Akadémia – vzdelávanie pre 21.
storočie

Bratislava 2020

Inovatívne metodiky pre vyučovanie programovania v Scratchi pre základné školy

Spracované s finančnou podporou národného projektu IT Akadémia – vzdelávanie pre 21. storočie

Autori: Monika Tomcsányiová, Angelika Hanesz, Zuzana Tkáčová

Recenzenti: doc. RNDr. Gabriela Lovászová, PhD., PaedDr. Iveta Demková

Neprešlo jazykovou úpravou.

Vydavateľ: Centrum vedecko-technických informácií SR, Bratislava

Rok vydania: 2020

Vydanie : 1. vydanie

ISBN:

Bratislava 2020

Obsah podlieha licenci Creative Commons BY 4.0

Tento projekt sa realizuje vďaka podpore z Európskeho sociálneho fondu v rámci Operačného programu Ľudské zdroje.

Obsah

ÚVOD	5
1 Preskúmame Scratch	8
2 Animácie	18
3 Udalosti myši a klávesnice	28
4 Zadávame vstupy, vytvárame premenné	35
5 Posielame správy	41
6A Použitie kamery	47
6B Použitie mikrofónu	53
7 Nové bloky	58
8 Náhoda: pečiatkujeme a presúvame	65
9 Súradnice a vstupy	74
10 Hra NIM 13	83
11 Bludisko	91
12 Štvorčeková sieť	99
13 Preskakuj	108
14 Posielanie správ a reagovanie na ne	114
15 Duplikovanie a klonovanie objektov	119
16 Programujeme grafický editor	126
17 Kreslíme zvuk	132
18 STE(A)M: Matematika	141
19 Animujeme chemické reakcie	154
20 STE(A)M: Projekty	162
PRÍLOHY	172

ÚVOD

V nasledujúcej zbierke je pripravená séria 20 metodík, teda veľmi podrobne zapísaných **postupov pre učiteľa**, ako viesť vyučovaciu hodinu. Tieto metodiky sú určené pre základný kurz programovania v prostredí **Scratch**, séria pokrýva oblasť Algoritmické riešenie problémov.

Metodiky sú určené pre 5. až 8. ročník základnej školy, konkrétny ročník je uvedený v každej metodike. Jednotlivé metodiky na seba nadväzujú. Je vhodné, ak ich učiteľ zaradí do jednotlivých ročníkov tak, že **nasledujú za sebou** v priebehu niekoľkých vyučovacích hodín. Zvyšok školského roka sa potom učiteľ môže venovať iným oblastiam informatiky, ktoré sú určené v ŠVP. Pokiaľ to nie je uvedené priamo v niektorej metodike, **neodporúčame meniť navrhnuté poradie metodík**. Učiteľ by si mal, pri rozhodnutí nasadiť tieto metodiky do vyučovania, prezrieť dopredu všetky metodiky. Tým získa prehľad o celkovej koncepcii celej série metodík.

Pri práci budeme používať **online** prostredie **Scratch** dostupné na adrese <https://scratch.mit.edu/>. Ak škola nemá možnosť zabezpečiť pre žiakov dostatočne rýchly prístup na internet, odporúčame stiahnuť si a používať **offline** verziu. V princípe sa tieto verzie nelíšia.

Na každú, alebo takmer na každú, vyučovaciu hodinu majú žiaci k dispozícii **pracovný list**, ktorý obsahuje zadania úloh a aj miesto pre zapisovanie žiackeho riešenia. Odporúčame, aby učiteľ žiakom pri každej fáze vyučovacej hodiny uviedol zoznam úloh z pracovného listu, ktoré majú vyriešiť. Pracovný list môže učiteľ žiakom vytlačiť a žiaci si môžu svoje vypracované zadania vkladať do zošitov a vytvárať si tak svoje portfólio o prostredí Scratch. Učiteľ môže dať žiakom k dispozícii aj priamo elektronický dokument (v tvare docx), do ktorého zapisujú svoje riešenia. Vypracované dokumenty si potom žiaci ukladajú napr. do svojho elektronického portfólia (priečinka).

Súčasťou metodiky na každú vyučovaciu hodinu sú aj jednotlivé úlohy, ktoré žiak nájde v pracovnom liste. Metodika potom obsahuje okrem zadaní úloh **vložené aj odpovede**, o ktorých predpokladáme, že sa budú objavovať v žiackych riešeniach pracovného listu. Treba si však uvedomiť, že postup na riešenie jednotlivých úloh, hlavne v zložitejších zadaniach, **nie je jednoznačný** a že skutočné žiacke riešenia sa môžu od predpokladu odpovedí v metodike často (aj veľmi podstatne) líšiť a pritom žiacke riešenie môže byť správne. Pod zadaním jednotlivých úloh z pracovného listu je v metodike poznámka pre učiteľa, ako úlohu v pracovnom liste **uviesť** alebo ako počas jej riešenia **usmerňovať žiakov** a pomáhať im. Učiteľ tak získa jediným náhľadom do metodiky prehľad o tom, aké úlohy sú v pracovnom liste a aké problémy by mohli mať žiaci pri ich riešení.

Väčšina metodík obsahuje aj Scratch súbory. Niektoré z nich sú určené **iba pre učiteľa** (ukážkové projekty, riešené projekty) a iné sú určené na to, aby ich učiteľ nejakým spôsobom **sprístupnil** pre žiakov (rozpracované projekty, projekty na žiacke skúmanie). Tieto súbory sú uložené **výhradne lokálne v priečinkoch zvlášť pre učiteľa a osobitne pre žiaka**. Je na učiteľovi, ktoré z nich, kedy a akým spôsobom dá žiakom k dispozícii. Učiteľ môže vytvoriť napr. vo svojej počítačovej sieti priečinok, do ktorého bude vkladať žiacke súbory. Pre svoje potreby ukážkových projektov si môže učiteľ vytvoriť svoje **vlastné** (verejne neprístupné) **Scratch Štúdio**. Do neho môže vkladať projekty, tak to bude potrebovať v priebehu školského roka. Je potrebné uvedomiť si, že **žiacke súbory nie je vhodné** dávať ako voľne dostupné na internet. Ide často o projekty, ktoré sú na začiatku vyučovacej hodiny **nefunkčné**, pretože úlohou žiaka je dokončiť ich v rámci práce na hodine a dotvoriť ich podľa pracovného listu, ktorý má k dispozícii, resp. dokončiť ich podľa pokynov učiteľa z vyučovacej hodiny. Zverejnenia žiackych projektov na internete bez materiálov a pracovných listov, sa **obávame** preto, že ak by si tieto súbory žiaci otvorili z internetu a spustili ich **bez** toho, aby mali k dispozícii **pracovný list**, mohli by nadobudnúť dojem, že tieto **projekty sú „zlé“**, že sa v nich nič „nedeje“ a nevedeli by, čo s nimi robiť ďalej. **Až pracovný list a invencia učiteľa na vyučovaní dávajú projektom zmysel**. Preto by mal učiteľ **projekty zverejňovať iba svojim žiakom a až v čase, keď s nimi budú žiaci na hodine pracovať**. Žiaci si rozpracovaný projekt **otvorí, dokončia** ho podľa pokynov v pracovnom liste a podľa inštrukcií učiteľa a môžu si ho **uložiť** do počítača alebo do štúdia všetkých projektov danej skupiny žiakov alebo, v prípade, ak si každý žiak vytvorí svoje vlastné konto, priamo na web programu **Scratch**.

Každá metodika obsahuje všetky potrebné informácie a materiály tak, aby sa učiteľ nahliadnutím do nej dozvedel postup, aký má na vyučovaní použiť. Súčasťou metodiky je aj riešenie pracovného listu (modrým písmom) a väčšinou aj riešenie jednotlivých úloh (väčšinou celý program, ak je program príliš dlhý, musí si ho učiteľ pozrieť priamo otvorením projektu v prostredí Scratch). Takto môže učiteľ prehliadnutím metodiky ľahko vidieť, aké úlohy budú žiaci na hodine riešiť, dočíta sa, ako má žiakov počas hodiny viesť, aké im dávať otázky a dozvie sa aj ďalšie podrobnosti, ktoré súvisia s vyučovacou hodinou. Učiteľ v metodike vidí aj to, ako bude, resp. môže, vyzeráť hotový program, ktorý majú žiaci vytvoriť, aké programátorské konštrukcie sa v programe vyskytnú, či je riešenie úlohy dlhé, nakoľko sú použité programátorské konštrukcie náročné, atď.

V nasledujúcej tabuľke uvádzame názvy metodík a naše odporúčania pre ich nasadenie do jednotlivých ročníkov na 2. stupni ZŠ.

metodika	zaradenie do ročníka
1 Preskúmame Scratch	5. ročník
2 Animácie	5. ročník
3 Udalosti myši a klávesnice	5. ročník
4 Zadávame vstupy, vytvárame premenné	5. ročník
5 Posielame správy	5. ročník
6A Použitie kamery	6. ročník
6B Použitie mikrofónu	6. ročník
7 Nové bloky	6. ročník
8 Náhoda: pečiatkujeme a presúvame	6. ročník
9 Súradnice a vstupy	6. ročník
10 Hra NIM 13	7. ročník
11 Bludisko	7. ročník
12 Štvorčeková sieť	7. ročník
13 Preskakuj	7. ročník
14 Posielanie správ a reagovanie na ne	7. ročník
15 Duplikovanie a klonovanie objektov	7. ročník
16 Programujeme grafický editor	8. ročník
17 Kreslíme zvuk	8. ročník
18 STE(A)M: Matematika	8. ročník
19 STE(A)M: Animujeme chemické reakcie	8. ročník
20 STE(A)M: Projekty	8. ročník

Ak učiteľ nestihne metodiku použiť v uvedenom ročníku, môže ju presunúť a zaradiť ako prvú v nasledujúcom ročníku na hodinách, ktoré plánuje venovať programovaniu v prostredí Scratch. Metodiky, ktoré sú navrhované pre 8. ročník **nie je možné** zaradiť do nižšieho ročníka, pretože žiaci nemajú dostatočné znalosti z predmetu, ktorý sa v STE(A)M projektoch používa ako predmet, pre ktorý žiaci vytvárajú program. Pre metodiky 18 až 20 dávame na zváženie učiteľa, či ich zaradiť pre všetkých žiakov alebo len pre tých, ktorí majú zvýšený záujem o programovanie.

1 PRESKÚMAJME SCRATCH

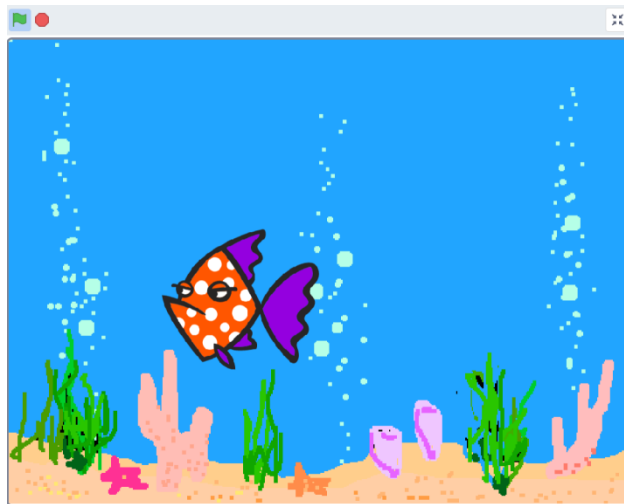
Tematický celok / Téma	Stupeň školy / Odporúčaný ročník /
Algoritmické riešenie problémov: <ul style="list-style-type: none"> • analýza problému • jazyk na zápis riešenia • pomocou postupnosti príkazov 	ZŠ / 5. - 6. ročník / 1 vyučovací hodina
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> • Riešenie jednoduchých algoritmických úloh bez počítača 	
Ciele	
Žiakom osvojované vedomosti a	Žiakom rozvíjané spôsobilosti
Prostredie Scratch <ul style="list-style-type: none"> • orientovať sa v prostredí Scratch, • zmeniť pozadie scény výberom z knižnice, • vybrať postavu z knižnice a pridať ju do scény, • vytvoriť pohyb postavy s nastavením štýlu otáčania. 	Informatické myslenie <ul style="list-style-type: none"> • (ALG8) zapísať algoritmy v konkrétnom programovacom jazyku (pohyb postavy) • (DEK1) lineárna dekompozícia – lineárne rozdeliť problémy na menšie časti tak, aby sa dali využiť pre dosiahnutie cieľa (rozdelenie výsledného programu na čiastkové zložky)
<p>Moderné detské programovacie prostredia ako Scratch prinášajú do výučby mnohé benefity, ako je cloudové programovanie, blokové prostredie Blockly, sociálnu interakciu v rámci komunity, zdieľanie a remixovanie projektov, čím významne prispievajú k zmene doterajších didaktických prístupov vo vyučovaní programovania. Ponúkajú žiakom možnosť vytvárania komplexných multimediálnych projektov, čo si vyžaduje od žiakov postupný systematický návrh riešenia. Nakoľko je to spravidla prvý kontakt žiakov s takýmto komplexným programovacím prostredím, nemajú ešte vytvorené a osvojené elementárne zručnosti pre riešenie problémov dekompozíciou na čiastkové podproblémy súvisiace s riešením scény, postavami, ako aj návrhom scenárov. Preto je nevyhnutné zamerať sa na rozvoj týchto zručností hneď od začiatku práce v tomto novom prostredí, aby mohol žiak postupne samostatne začať vytvárať vlastné programátorské projekty.</p>	
Dominantné vyučovacie metódy a formy	
Bádateľská metóda (model 5E). Frontálna forma. Práca vo dvojiciach.	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> • prístup na internet https://scratch.mit.edu/ alebo off-line práca v nainštalovanom programe Scratch 3 • pracovný list pre žiaka • projekt pre učiteľa akvarium a pre žiaka pokus1 	
Diagnostika splnenia vzdelávacích cieľov	
Diagnostika pomocou sebahodnotiacej rubriky a na základe analýzy práce žiakov.	

ÚVOD

Toto je prvá metodika zo série 20 metodík, ktoré sú určené pre základný kurz programovania v prostredí **Scratch**. Pri práci budeme využívať online prostredie **Scratch** dostupné na adrese <https://scratch.mit.edu/> s editorom, ku ktorému sa dostaneme výberom **Vytvor** v hlavnej ponuke.

ZAPOJENIE – CCA 6 MIN.

Na úvod motivujeme žiakov frontálnou ukážkou projektu **akvárium** (využitím dataprojektora alebo interaktívnej tabule) a riadenou analýzou projektu, na ktorom budú počas hodiny pracovať. Výsledkom tejto analýzy bude základná špecifikácia projektu **akvárium** a formulácia cieľov tejto vyučovacej hodiny. Žiakov vyzveme, aby chvíľu sledovali hotový projekt akvária a následne sa pokúsili vlastnými slovami porozprávať, čo v ňom vidia a čo sa v ňom deje.



Hotový projekt akvárium

Poznámka

Postupne oslovujeme žiakov a nabádame ich, aby slovné vyjadrili svoje pozorovanie – najlepšie je, ak jeden žiak pridá len jeden prvok, aby sa v tejto fáze hodiny mohlo zapojiť čo najviac žiakov. Žiaka neopravujeme, len sa ho pomocnými otázkami snažíme usmerniť, prípadne osloviť ďalších žiakov, ktorí môžu pozorovanie upresniť.

Žiaci by mali byť schopní správne identifikovať:

- pozadie/scénu
- pohybujúci sa objekt (rybu) a dokázať popísať charakter jeho pohybu

a navrhnúť tak špecifikáciu výsledného softvérového projektu akvária. Výsledkom pozorovania môže byť schéma, pozri obrázok, alebo môže vzniknúť pojmová mapa a pod. Tú môže učiteľ, spolu so žiakmi, postupne zapisovať na tabuľu, aby počas celej hodiny bolo možné sledovať progres riešenia. Žiaci si schému zapisujú aj do dopracovaného listu.



Dekompozícia projektu akvária na menšie časti

Na základe navrhnutej dekompozície predstavíme žiakom špecifické ciele hodiny:

- orientovať sa v prostredí **Scratch**
- zmeniť pozadie scény výberom z knižnice pozadí
- vybrať postavu z knižnice a pridať ju do scény
- vytvoriť pohyb postavy s ošetrením otáčania

SKÚMANIE – CCA 10 MIN.

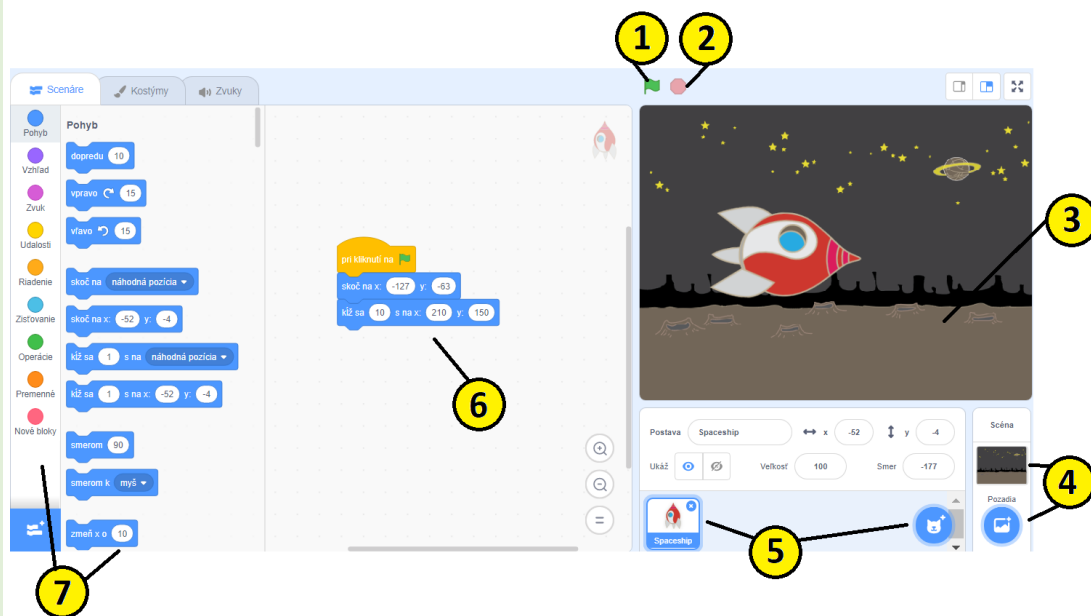
Pre úspešné vyriešenie projektu je potrebné najprv sa zorientovať v prostredí **Scratch**. V tejto fáze hodiny žiaci pracujú vo dvojiciach na počítačoch s použitím pracovných listov, do ktorých si zaznačujú riešenia jednotlivých úloh. Pri práci používajú pripravený projekt **pokus1**. Na tomto mieste odporúčame upozorniť žiakov, že toto nie je test a ich odpovede nebudeme hodnotiť ako správne, resp. nesprávne (podobne to tak bude aj v nasledujúcich metodikách). V tejto fáze nám ide len o skúmanie. K výsledkom žiackeho skúmania sa vrátíme v časti **Vysvetlenie**. V tejto etape záznamy žiakov nevyhodnocujeme ani nekomentujeme.

Poznámka

Napriek tomu, že sa jedná o prácu vo dvojiciach, pracovné listy môžu dostať všetci žiaci a zapisovaním riešení si tvoria poznámky, ktoré si po skončení hodiny založia do svojich zošitov.

Túto časť môžu žiaci realizovať vo dvojiciach. Žiaci vo dvojiciach môžu vzájomne komunikovať a lepšie tak pochopiť skúmané javy. Aj diskusia medzi skupinami je vhodná.

Úloha 1 Klikajte a skúmajte postupne jednotlivé oblasti podľa čísiel na obrázku a priradte čísla k správnej funkcii skúmanej oblasti podľa tabuľky:



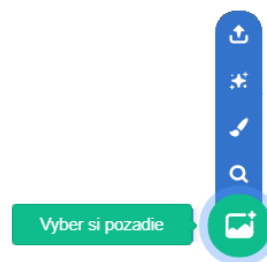
3	okno bežiaceho programu
7	skupiny príkazov pre programovanie (pre pohyb postáv, ich vzhľad, zvuky a pod...)
5	práca s postavami (výber, kreslenie a pod...)
1	spustenie programu
4	práca s pozadím scény (výber, kreslenie a pod...)
6	program, t.j. čo má robiť zvolená postava (=scenár)
2	zastavenie programu

Poznámka

Úloha 1 sa zameriava na základnú orientáciu v programovom prostredí **Scratch**. Na uľahčenie postupu sú jednotlivé oblasti číslované tak, ako má cez ne žiak postupne prechádzať. V prvej časti riešenia ich skúma a pozoruje, v druhej časti priradzuje k jednotlivým oblastiam pripravené popisy ich účelu. Modrou farbou sú pre učiteľa zobrazené správne riešenia. V žiackom pracovnom liste riešenia, samozrejme, uvedené nie sú.


Úloha 2

- Preskúmaj oblasť pre prácu s pozadím scény a zisti a zapíš, akými spôsobmi sa dá vytvoriť nové pozadie. Spôsoby vytvorenia pozadia zistíš pomocou nápovede – stačí priblížiť myšku k ikonke a neklikať.
prezretím všetkých pozadí Scratch, nakreslením, náhodne, zo súboru
- Preskúmaj knižnicu s pozadiami a vyber si niektoré. Dokážeš pridať nové pozadie? **ÁNO NIE**






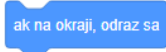
Poznámka

Úloha 2 sa zameriava na skúmanie možností zmeny pozadia využitím nápovede. Do pracovného listu žiak zaznamenáva svoje zistenia, na základe ktorých by mal byť schopný doplniť do skúmaného programu nové pozadie z knižnice.

Úloha 3	a) Preskúmaj oblasť pre prácu s postavami a zisti, akými spôsobmi vieš vytvoriť novú postavu:	
	výberom zo všetkých postáv Scratch, nakreslením, náhodne zo súboru	
	b) Preskúmaj knižnicu s postavami a vyber si niektorú postavu. Postava sa objaví na scéne aj v okne postáv. Dokážeš ju na scéne presunúť na iné miesto? ÁNO NIE	
	c) Vyskúšaj kliknúť pravým tlačidlom myši na zvolenú postavu v okne postáv. Vieš ju zmazať? ÁNO NIE	

Poznámka

Úloha 3 sa zameriava na skúmanie možností pridania nových postáv využitím nápovede. Do pracovného listu žiak zaznamenáva svoje zistenia, na základe ktorých by mal byť schopný doplniť do skúmaného programu novú postavu z knižnice. Ďalej žiak experimentuje s posúvaním postáv na scéne a s využitím kontextového menu (dostupného po kliknutí pravým tlačidlom myši na postavu) s cieľom mazania postavy.

Úloha 4	Preskúmajte bližšie skupiny príkazov – nájdite, v ktorých skupinách sa nachádzajú nasledujúce bloky:		
			 
	Skupina blokov: UDALOSTI	Skupina blokov:	Skupina blokov:

Poznámka

Úloha 4 sa zameriava na skúmanie skupín príkazov a hľadanie potrebných blokov. Žiaci doplnia do pracovného listu názvy skupín blokov. Pri vyhľadávaní blokov by si mali zároveň všimnúť farebnú príslušnosť bloku k danej skupine blokov.

VYSVETLENIE – CCA 12 MIN.

Na základe svojich poznámok v pracovných listoch žiaci vysvetlia, čo zistili. Učivo nevysvetľujeme, len usmerňujeme žiakov pri vysvetľovaní. V prípade nesprávnej alebo nie úplne jasnej formulácii, môžu úlohu vysvetliť ďalší žiaci. Do fázy vysvetľovania by sa malo zapojiť čo najviac žiakov.

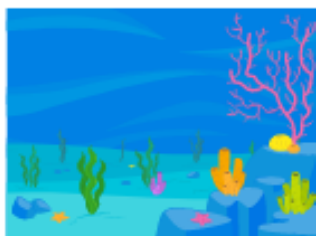
V druhej časti vysvetľovania nadviážeme na prezentované žiacke zistenia a **spoločne** budeme tvoriť vzorový projekt akvárium. Postup je stanovený navrhnutou schémou na obrázku vyššie. Žiakov by sme mali vyzývať, aby podľa schémy a svojich doterajších zistení sami navrhovali jednotlivé kroky riešenia; v prípade niektorých postupov (napr. otáčanie postavy, vytváranie skriptu pre pohyb postavy) im môžeme ukázať čiastkové riešenie, ktoré žiaci spoločne ďalej rozširujú a doladujú.

V nasledujúcej časti metodiky uvádzame pre učiteľa návod, ako vytvoriť projekt. Podľa neho môže postupovať v prípade, ak si zvolí frontálny spôsob priebehu vyučovacej hodiny a projekt bude vytvárať na učiteľskom počítači, pričom bude vysvetľovať žiakom jednotlivé kroky a ukazovať im postupy pre vytvorenie projektu.

AKVÁRIUM – NÁVOD

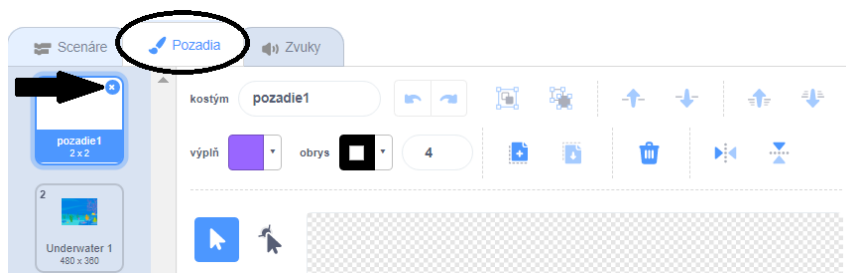
Krok 1

Začneme nastavením pozadia scény, ktoré si vyberieme z knižnice – zvolíme si napr. pozadie **UNDERWATER1**, ktoré nájdeme v téme **POD VODOU**:

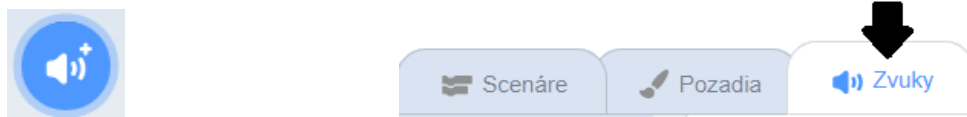


Krok 2

Po pridaní pozadia pribudne nová karta **Pozadia**. Pôvodné biele pozadie z karty vymažeme – najprv naň klikneme a potom ho môžeme vymazať krížikom v pravom hornom rohu:



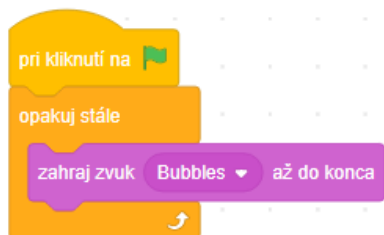
K pozadiu pridáme prehrávanie zvuku – najprv klikneme na kartu **Zvuky**. Tu môžeme vybrať zvuk z knižnice kliknutím na ikonku:



Pre akvárium je vhodný zvuk **BUBBLES** (v kategórii **Všetko** sú zvuky usporiadané abecedne). Na prehrávanie zvuku vytvoríme scenár na karte **Scenáre** – prehrávanie sa má spustiť po kliknutí na zelenú vlajku a je ho potrebné stále opakovať, preto potrebujeme vybrať potrebné príkazy na vytvorenie výsledného programu:

zo skupiny príkazov Udalosti	zo skupiny príkazov Riadenie	zo skupiny príkazov Zvuk

Výsledný scenár:



Krok 3

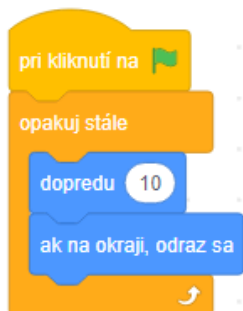
Projekt má zatiaľ len pôvodnú postavu mačky, ktorú zmažeme a miesto nej doplníme z knižnice postáv novú postavu – vhodnú rybu nájdeme v kategórii **Zvieratá**.



Potrebujeme ešte doplniť vhodný scenár pre našu rybu – po kliknutí na zelenú vlajku má ryba stále opakovať posunutie, o napr. 10 krokov, a ak narazí na okraj, tak sa odrazí. Všetky potrebné bloky vidíme v nasledujúcej tabuľke:

zo skupiny príkazov Udalosti	zo skupiny príkazov Riadenie	zo skupiny príkazov Pohyb

Výsledný scenár

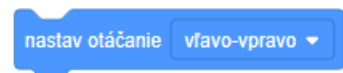


Krok 4

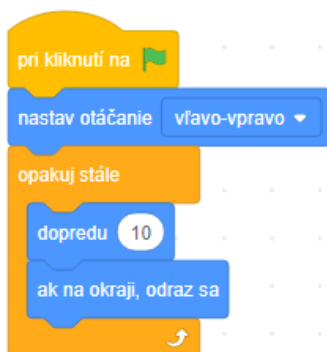
Projekt otestujeme. Pohyb ryby má dva problémy. Jednak ryba pri odraze od vnútorného okraja akvária pláva hlavou dole:



Túto neprirodzenú situáciu vyriešime nastavením štýlu otáčania postavy pri jej pohybe príkazom zo skupiny príkazov **Pohyb**:

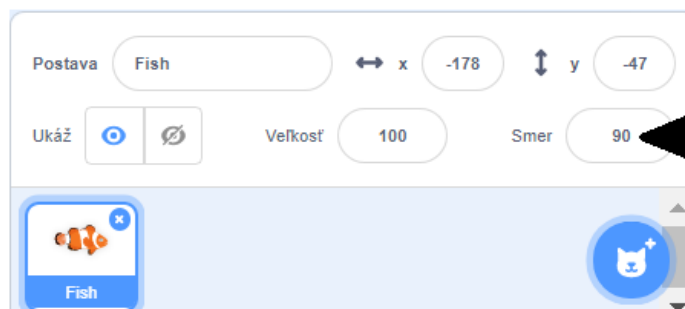


Tento príkaz vložíme na úplný začiatok nášho programu, teda ešte pred samotný blok opakovania pohybu:



Poznámka

*Pokiaľ ideme meniť štýl otáčania postavy a naša postava je aktuálne v nesprávnom natočení (t.j. ryba pláva hlavou dole), musíme aj manuálne nastaviť aktuálne natočenie postavy v políčku **Smer**, ktoré*



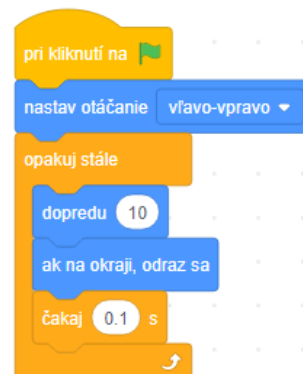
nájdeme pri vlastnostiach postáv:

Ďalším problémom je, že ryba pláva príliš rýchlo – toto vyriešime doplnením “čakacieho” bloku zo skupiny blokov **Riadenie**:



Tento príkaz vložíme pred alebo za pohyb ryby a nastavíme vhodný čas, napr. 0.1 sekundy – používame desatinnú bodku (na rozdiel od matematiky, kde používame desatinnú čiarku).

Výsledný scenár pre rybu v našom akváriu bude napokon vyzerať takto:



Poznámka

Vyššie vidíme návod, ako vytvoriť projekt akvárium. Návod môže slúžiť pre učiteľa ako tutoriál, podľa ktorého je vhodné rámcovo postupovať, avšak jednotlivé kroky riešenia by mali zvládnuť navrhnuť aj samotní žiaci, na základe svojich doterajších poznatkov z fázy **Skúmania**. Počas vývoja projektu sa snažíme prenechať iniciatívu na žiakov, ktorých postupne vyzývame, aby formulovali ďalšie kroky riešenia, ktoré potom spoločne implementujeme – učiteľ na svojom učiteľskom počítači frontálne a žiaci na svojich počítačoch (1 žiak/1 počítač).

ROZPRACOVANIE – CCA 8 MIN.

Po vytvorení základného funkčného modelu akvária vyzveme žiakov, aby navrhli ďalšie vylepšenia/rozšírenia programu. Svoje návrhy zapisujú do pracovného listu.

Úloha 5 Do hotového projektu **akvária** navrhni a doplň vlastné vylepšenia alebo rozšírenie.

Svoje návrhy zapíš sem:

Medzi žiackymi odpoveďami by sa mohli objaviť napr.

- pridať do akvária viac rybičiek/živočíchov,
- rybičky by sa pohybovali rôznou rýchlosťou,
- po nejakom čase sa môže pozadie zmeniť na iné (deň/noc),
- rybičky by mohli reagovať na kliknutie myšou (napr. zmenou smeru plávania, vydaním nejakého zvuku),
- rybičky by neplávali len systematicky sprava doľava, ale niekedy aj náhodne,
- aplikovať scenár na iné pozadie a iné postavy (napr. lúka a let motýľov)...

Žiaci môžu tiež zhodnotiť, ktoré z rozšírení by už boli schopní so svojimi doterajšími vedomosťami zrealizovať, čo môžu aj samostatne prakticky vyskúšať. Sledujeme individuálnu prácu žiakov, podporujeme a povzbudzujeme ich.


HODNOTENIE – CCA 4 MIN.


Na evalváciu slúži sebahodnotiaca rubrika, pomocou ktorej žiaci zakrúžkovaním sami zhodnotia úroveň osvojenia kľúčových vedomostí a zručností, ako aj splnenie cieľov hodiny. Zároveň rubrika slúži na zhrnutie základných poznatkov a zručností, ktoré si mali žiaci na hodine osvojiť.

ČO SOM SA NAUČIL/NAUČILA...	
Viem vysvetliť, na čo slúžia základné časti prostredia Scratch	VIEM / VIEM S POMOCOUCOU / NEVIEM
Viem vybrať pozadie z knižnice	VIEM / VIEM S POMOCOUCOU / NEVIEM
Viem vybrať postavu z knižnice	VIEM / VIEM S POMOCOUCOU / NEVIEM
Viem vytvoriť jednoduchý pohyb postavy	VIEM / VIEM S POMOCOUCOU / NEVIEM





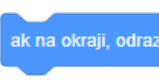
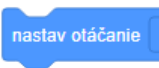
Na poslednej strane pracovného listu majú žiaci pripravené zhrnutie vedomostí

ČO SME SA NAUČILI

Postavy môžeme meniť alebo vytvárať kliknutím na ikonu  v pravom dolnom rohu prostredia Scratch.

Pozadia scény môžeme meniť alebo vytvárať kliknutím na ikonu .

Pri programovaní sme použili nasledovné bloky:

-  nastavenie začiatku programu (spustenie kliknutím na zelenú vlajku)
-  neustále opakovanie (blokov, ktoré sú vo vnútri tohto bloku)
-  pozastavenie programu na určitý čas (napr. pre spomalenie postavy)
-  pohyb postavy o 10 krokov dopredu
-  odrazenie postavy od okraja okna scény
-  nastavenie spôsobu otočenia postavy po odraze

2 ANIMÁCIE

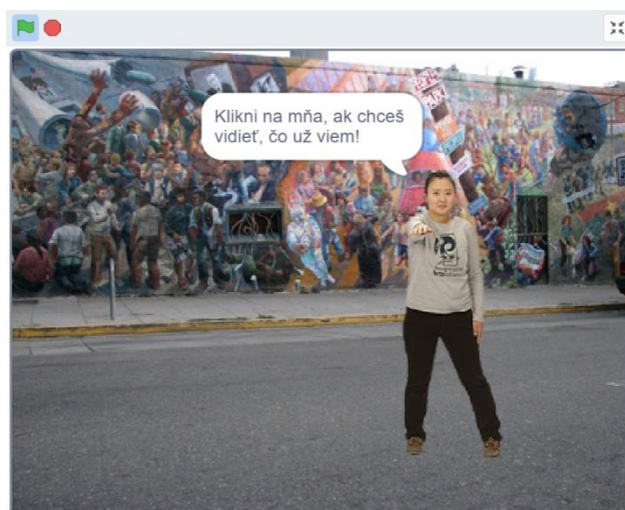
<i>Tematický celok / Téma</i>	<i>Stupeň školy / Odporúčaný ročník / Rozsah</i>
Algoritmické riešenie problémov: <ul style="list-style-type: none"> analýza problému jazyk na zápis riešenia pomocou postupnosti príkazov 	ZŠ / 5. - 6. ročník / 1 vyučovací hodina
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> Riešenie jednoduchých algoritmických úloh bez počítača Základy práce v prostredí Scratch Princípy tvorby jednoduchých animácií 	
Ciele	
Žiakom osvojované vedomosti a zručnosti	Žiakom rozvíjané spôsobilosti
Prostredie Scratch <ul style="list-style-type: none"> zmeniť kostým postavy, vytvoriť animáciu striedaním kostýmov, vložiť prehrávanie zvuku, vytvoriť udalosť riadenú vstupmi z klávesnice, zastaviť vykonávanie scenára. 	Informatické myslenie <ul style="list-style-type: none"> (LOG2) využitím logických zdôvodnení predpokladať správanie sa jednoduchých programov (akcie, ktoré sa majú vykonať po konkrétnych udalostiach) (ALG8) zapísať algoritmy v konkrétnom programovacom jazyku (vytváranie interaktívnych multimedialných animácií) (DEK2) hierarchická dekompozícia – hierarchicky rozdeliť problémy na menšie časti tak, aby sa dali využiť pre dosiahnutie cieľa (rozdelenie výsledného programu na čiastkové zložky)
<p>Prostredie Scratch ponúka žiakom možnosť vytvárania komplexných interaktívnych multimedialných projektov, čo si vyžaduje od žiakov postupný systematický návrh riešenia. Žiaci nemajú ešte spravidla vytvorené a osvojené elementárne zručnosti pre riešenie problémov dekompozíciou na čiastkové podproblémy súvisiace s riešením scény, postavami, ako aj návrhom scenárov. Pri práci s postavami je možné využívať prvky interaktivity, ktoré ponúka prostredie Scratch, ktoré však vnášajú do vykonávania algoritmu istú nelinearitu súvisiacu s tzv. udalosťami riadeným programovaním (angl. event driven programming). Preto je potrebné zamerať sa na precvičovanie detekcie a programátorského riešenia stavov, ktoré vznikajú pri interaktívnych vstupoch od používateľa.</p>	
Dominantné vyučovacie metódy a formy	
Bádateľská metóda (model 5E). Frontálna forma. Práca vo dvojiciach.	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> lokálna verzia Scratch 3 alebo prístup na internet s online editorom pracovný list pre žiaka pracovné projekty online pre učiteľa interaktívna postava a pre žiaka pokus2 	
Diagnostika splnenia vzdelávacích cieľov	
Diagnostika pomocou sebahodnotiacej rubriky a na základe analýzy práce žiakov.	

ÚVOD

Metodika je zameraná na živé experimentovanie a objavovanie prostredia **Scratch** ako nástroja pre tvorbu interaktívnych multimediálnych animácií. Je pravdepodobné, že žiaci sa už s tvorbou jednoduchých animácií stretli – jednak v podobe kreslených animácií (napr. v programe **Logo Motion**), alebo v podobe udalosťami riadených animácií v niektorom z prezentačných softvérov, alebo aj pri programovaní v niektorom z detských programovacích prostredí (napr. **Baltík** alebo **ScratchJr**). Interaktivita programu bude realizovaná stlačením rôznych klávesov na klávesnici, ktoré môžu žiaci tvorivo využívať vo svojich programoch.

ZAPOJENIE – CCA 5 MIN.

Na úvod motivujeme žiakov **frontálnou** ukážkou (využitím dataprojektora alebo interaktívnej tabule) a riadenou analýzou projektu, na ktorom budú pracovať, výstupom čoho bude základná špecifikácia softvérového projektu interaktívnej postavy a formulácia cieľov hodiny. Žiakov vyzveme, aby chvíľu sledovali hotový projekt **interaktívna postava** a následne sa pokúsili vlastnými slovami opísať, čo v ňom vidia a čo sa v ňom deje.



Hotový projekt **Interaktívna postava**

Poznámka:

Postupne oslovujeme žiakov a nabádame ich, aby slovné vyjadrili svoje pozorovanie – najlepšie je, ak jeden žiak pridá len jeden prvok, aby sa v tejto fáze hodiny mohlo zapojiť čo najviac žiakov. Žiaka neopravujeme, len pomocnými otázkami sa ho snažíme usmerniť, prípadne osloviť ďalších žiakov, ktorí môžu pozorovanie upresniť.

Žiaci by mali byť schopní správne identifikovať:

- pozadie/scénu,
- sprievodný zvuk,
- postavu a charakter jej správania

a navrhnuť tak špecifikáciu výsledného softvérového projektu. Výsledkom pozorovania môže byť schéma, pozri obrázok, alebo môže vzniknúť pojmová mapa a pod. Tú môže učiteľ, spolu so žiakmi zapísať na tabuľu, aby počas celej hodiny bolo možné sledovať progres riešenia. Žiaci si schému zapíšu aj do dopracovaného listu.



Dekompozícia projektu interaktívnej postavy na menšie časti

Na základe navrhnutej dekompozície sa spýtame žiakov, ktoré časti projektu by už teraz vedeli samostatne vyriešiť. Žiaci by mali byť schopní identifikovať **nový** problém – animáciu postavy. Položme im otázku, či sa už niekde s tvorbou animácie stretli a ako by sme podľa podobných princípov boli schopní realizovať animáciu našej postavy. Následne predstavíme žiakom ciele hodiny:

- zmeniť kostým postavy,
- vytvoriť animáciu striedaním kostýmov,
- vložiť prehrávanie zvuku,
- vytvoriť udalosť riadenú vstupmi z klávesnice,
- zastaviť vykonávanie scenára.

SKÚMANIE – CCA 10 MIN.

V tejto fáze hodiny žiaci pracujú vo dvojiciach na počítačoch s použitím pracovných listov, do ktorých si zaznačujú riešenia jednotlivých úloh. Pri práci používajú pripravený projekt **pokus2**. Na tomto mieste odporúčame upozorniť žiakov, že toto nie je test a ich predpovede nehodnotíme ako správne, resp. nesprávne (podobne aj v nasledujúcich metodikách). V tejto fáze nám ide len o skúmanie. K výsledkom žiackeho skúmania sa vrátíme v časti **Vysvetlenie**. V tejto etape záznamy žiakov nevyhodnocujeme ani nekomentujeme.

Úloha 1 Kliknutím na zelenú vlajku spustíte program, otestujte ho a doplňte tabuľku:

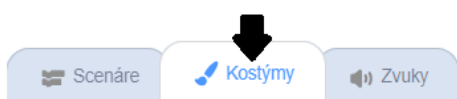


Udalosť	Akcia (čo sa deje?)
Po kliknutí na zelenú vlajku	<i>Pozdraví sa a povie, aby sme naňho klikli alebo stlačili medzerník</i>
Po kliknutí na smajlíka	<i>Zmení sa na prekvapeného</i>
Po stlačení medzerníka	<i>Žmurkne, postupne zmení výraz na smutný a potom naspäť na veselý</i>

Úloha 2 Sledujte bližšie správanie sa smajlíka po stlačení medzerníka a usporiadajte správne jednotlivé fázy animácie – doplňte poradové čísla:

1.	3.	7.	4.	6.	5.	2.

Úloha 3 Kliknite na záložku **KOSTÝMY**:



Preskúmajte v programe kostýmy smajlíka a dokreslite ich podľa názvov:

smajlik1	smajlik2	smajlik3	smajlik4	smajlik5

Riešenie **úlohy 3** vidíme na obrázku. Samozrejme žiaci budú smajlíka kresliť do pracovného zošita rukou.

smajlik1	smajlik2	smajlik3	smajlik4	smajlik5

Úloha 4 Preskúmajte bližšie skupiny príkazov – nájdite, v ktorých skupinách sa nachádzajú nasledujúce bloky:

Skupina príkazov: Udalosti	Skupina príkazov: Riadenie	Skupina príkazov: Vzhľad
Skupina príkazov: Riadenie	Skupina príkazov: Vzhľad	Skupina príkazov: Udalosti

Úloha 5 Porovnaj v programe príkazy, ktoré sa vykonajú po kliknutí na zelenú vlajku, po kliknutí na smajlíka a po stlačení medzerníka a pokúste sa pridať nové správanie smajlíka tak, aby

- po stlačení klávesu **a** smajlík krátko žmurkol a chvíľu hovoril „Ahoj!“
- po stlačení klávesu **b** smajlík najprv povedal „Škoda, že už bude koniec...“ a potom postupne zosmutnel

Podarilo sa Vám to? **ÁNO - NIE**

VYSVETLENIE – CCA 10 MIN.

Na základe svojich poznámok v pracovných listoch žiaci vysvetlia, čo zistili. Učivo nevysvetľujeme, len usmerňujeme žiakov pri ich vysvetľovaní. V prípade nesprávnej, alebo nie úplne jasnej formulácie, môžu úlohu vysvetliť ďalší žiaci. Do fázy vysvetľovania by sa malo zapojiť čo najviac žiakov.

V druhej časti vysvetľovania nadviažeme na prezentované žiacke zistenia a spoločne budeme tvoriť vzorový projekt interaktívnej postavy. Postup je stanovený navrhnutou schémou. Žiakov by sme mali vyzývať, aby podľa schémy a svojich doterajších zistení sami navrhovali jednotlivé kroky riešenia.

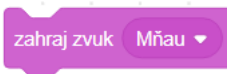
S prehrávaním zvukov sa žiaci stretli už v rámci predošlej hodiny/témy, kde sa použil zvuk len na vylepšenie pozadia akvária. Žiakom pripomenieme použitie karty **Zvuky**:

Tu môžeme vybrať zvuk z knižnice kliknutím na ikonku:



Na prehrávanie zvuku máme v skupine príkazov **Zvuk** k dispozícii dva podobné bloky:

Príkaz **zahraj zvuk** spustí prehrávanie, a pokračuje sa vo vykonávaní



dalších príkazov. Príkaz **zahraj zvuk...až do konca** najprv prehrá celú zvukovú nahrávku a až potom sa pokračuje ďalším príkazom podľa scenára (toto však môže spôsobiť komplikácie pri dlhých zvukových súboroch).

V projekte **interaktívna postava** je potrebné žiakov upozorniť, aby pri prehrávaní rozlišovali, či zvuk reprezentuje zvuk postavy (napríklad nejaký náraz alebo jej smiech) a vtedy ho použijeme ako **súčasť scenáru postavy**. Ak budeme používať zvuk len ako sprievodný prvok v programe (podobne ako to bolo v projekte **akvárium** alebo aj pri postave je zvuk len na doplnenie atmosféry potrebnej k tancu alebo cvičeniu), potom je **súčasťou scenára pre pozadie** našej scény.

Pri navrhovaní scenára postavy si môžeme pomôcť vytvorením tabuľky udalostí a akcií, pozri tab. 1, v ktorej s pomocou žiakov špecifikujeme akcie, ktoré je potrebné zahrnúť do jednotlivých blokov.

Tab.1 Tabuľka udalostí a akcií (príklad)

Udalosť	Akcia
<i>Po kliknutí na zelenú vlajku</i>	<i>zobrazí na 2 sekundy bublinu „Ahoj, volám sa Anina. Klikni na mňa, ak chceš vedieť, čo už viem!“</i>
<i>Po kliknutí na postavu (t.j. na „mňa“)</i>	<i>zobrazí na 2 sekundy bublinu „Ukážem Ti, ako tancujem! Stlač kláves a alebo b! Zastaviť ma môžeš medzerníkom!“</i>
<i>Po stlačení klávesu a</i>	<i>stále opakuje zmenu kostýmu (verzia 1) a krátke čakanie</i>
<i>Po stlačení klávesu b</i>	<i>stále opakuje zmenu kostýmu (verzia 2) a krátke čakanie</i>
<i>Po stlačení medzerníka</i>	<i>zastaví všetko</i>

Poznámka

Nižšie vidíme návod pre učiteľa ako postupovať pri vytvorení projektu **animácia**. Tento návod môže učiteľ použiť ako tutoriál, podľa ktorého bude rámcovo postupovať počas **frontálneho vývoja** projektu spolu so žiakmi. Avšak jednotlivé kroky riešenia by mali zvládnuť navrhnuť aj samotní žiaci, na základe svojich doterajších poznatkov z fázy SKÚMANIE. Počas vývoja projektu sa snažíme prenechať iniciatívu na žiakov, ktorých postupne vyzývame, aby formulovali ďalšie kroky riešenia, ktoré potom spoločne implementujeme – učiteľ na svojom učiteľskom počítači frontálne a žiaci na svojich počítačoch (1 žiak/1 počítač).

INTERAKTÍVNA POSTAVA – NÁVOD

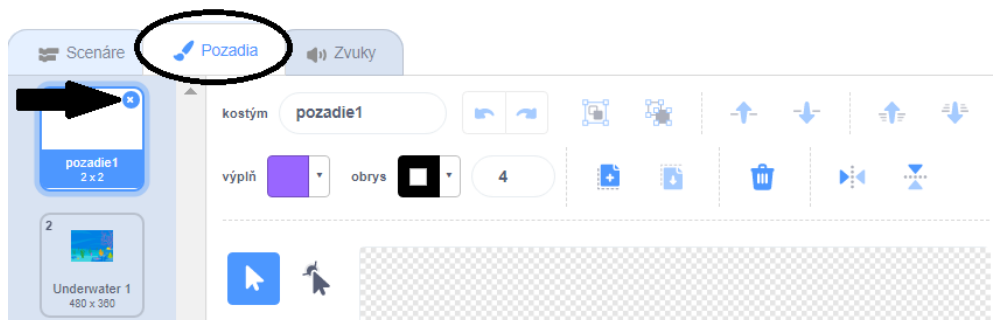
Krok 1

Začneme nastavením pozadia scény, ktoré si vyberieme z knižnice – zvolíme si napr. pozadie **WALL 2**, ktoré nájdeme v téme **VONKU**:

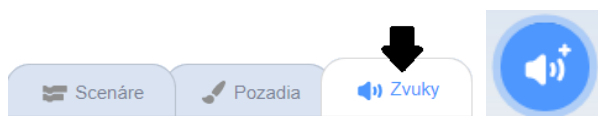


Krok 2

Po pridaní pozadia pribudne v časti **Pozadia** nové pozadie. Pôvodné biele pozadie z karty vymažeme – najprv naň klikneme a potom ho môžeme vymazať krížikom v pravom hornom rohu:



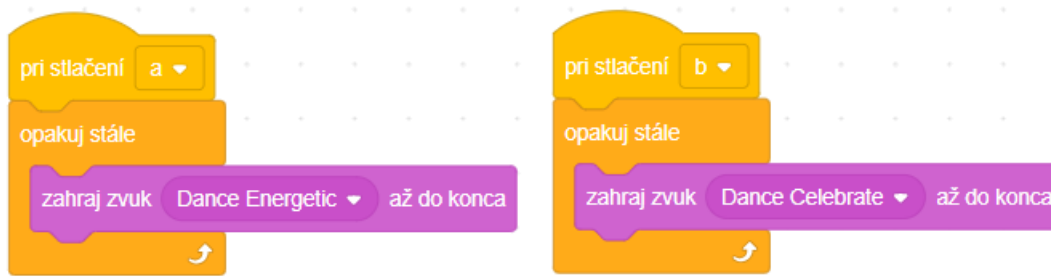
K pozadiu pridáme prehrávanie zvuku – najprv klikneme na kartu **Zvuky**, kde môžeme vybrať zvuk z knižnice kliknutím na ikonku:



Pre tanečnú zostavu sú vhodné napr. zvuky **DANCE ENERGETIC** a **DANCE CELEBRATE** (v kategórii **Všetko** sú zvuky usporiadané abecedne). Na prehrávanie zvukov vytvoríme dva scenáre na karte **Scenáre** – prehrávanie jedného zvuku sa má spustiť po stlačení klávesu **a** prehrávanie druhého zvuku po stlačení klávesu **b**, čo zabezpečíme príkazom zo skupiny príkazov **Udalosti**, v ktorej stlačením šípky dolu vyberieme namiesto medzerníka príslušné písmeno:



Prehrávanie je potrebné stále opakovať, preto potrebujeme vybrať potrebné príkazy na vytvorenie výsledných dvoch scenárov:



Ktorýkoľvek zvuk je potrebné vedieť kedykoľvek zastaviť po stlačení medzerníka, čo zabezpečíme príkazom na zastavenie všetkých scenárov (nájdeme ho v skupine **Riadenie**) a potrebný scenár bude potom vyzeráť nasledovne:



Krok 3

Projekt má zatiaľ len pôvodnú postavu mačky, ktorú zmažeme a miesto nej doplníme z knižnice postáv novú postavu – vhodnú ľudskú postavu (napr. Aninu) nájdeme v kategórii **Tanec**:



Po vložení postavy si môžeme prezrieť jej kostýmy kliknutím na kartu **Kostýmy**, kde zistíme, že táto postava má 13 rôznych kostýmov, ktoré použijeme na vytvorenie jej cvičebných zostáv. Potrebujeme ešte doplniť vhodný scenár pre našu Aninu – pomôžeme si tabuľkou udalostí:

Udalosť	Akcie
Po kliknutí na zelenú vlajku	- zobrazí na 2 sekundy bublinu „Ahoj, volám sa Anina. Klikni na mňa, ak chceš vedieť, čo už viem!“
Po kliknutí na postavu (t.j.na	- zobrazí na 2 sekundy bublinu „Ukážem Ti, ako tancujem!“

„mňa“)	Stlač kláves a alebo b! Zastaviť ma môžeš medzerníkom!“
Po stlačení klávesu a	- stále opakuje zmenu kostýmu (verzia 1) a krátke čakanie
Po stlačení klávesu b	- stále opakuje zmenu kostýmu (verzia 2) a krátke čakanie
Po stlačení klávesu medzerník	zastaví všetko

Všetky potrebné bloky si môžeme pripraviť:

zo skupiny príkazov Udalosti	zo skupiny príkazov Riadenie	zo skupiny príkazov Vzhľad

Výsledné scenáre pre nami zvolené udalosti môžu vyzeráť napríklad takto:

ROZPRACOVANIE – CCA 10 MIN.

Po vytvorení funkčného modelu interaktívnej postavy vyzveme žiakov, aby navrhli ďalšie vylepšenia/rozšírenia programu. Môžu uvádzať napr. vylepšenia, ktoré uvádzame v riešení úlohy 6.

Úloha 6 Do hotového projektu **Interaktívna postava** navrhnite a doplňte vlastné vylepšenia alebo rozšírenie!

- rozšíriť postave existujúce cvičebné zostavy o ďalšie cviky,
- zmeniť rýchlosť vykonávania cvičebných zostáv,
- doplniť ďalšie texty postave,
- pridať postave ďalšie cvičebné zostavy po stlačení ďalších klávesov,
- pridať do projektu ďalšiu postavu s inými zostavami,
- aplikovať scenár na iné pozadie a iné postavy (napr. lúka a let motýľov, akvárium, ulica...)

Žiaci slovné zhodnotia, ktoré z navrhnutých rozšírení by už boli schopní so svojimi doterajšími vedomosťami zrealizovať, čo môžu aj samostatne prakticky vyskúšať. Cieľom tejto časti hodiny je vytvoriť vlastný interaktívny multimediálny program s využitím zvukov, animácií, textu a ovládania kliknutím myšou a klávesnicou. Sledujeme individuálnu prácu žiakov, podporujeme a povzbudzujeme ich.

HODNOTENIE – CCA 5 MIN.

Na evalváciu slúži sebahodnotiaca rubrika, pomocou ktorej žiaci zaškrtnutím sami zhodnotia úroveň osvojenia kľúčových vedomostí a zručností, ako aj splnenie cieľov hodiny. Zároveň rubrika slúži na zhrnutie základných poznatkov a zručností, ktoré si žiaci na hodine mali osvojiť.

ČO SOM SA NAUČIL/NAUČILA...	
Viem zmeniť postave kostým	VIEM / VIEM S POMOCOU / NEVIEM
Viem vytvoriť animáciu striedaním niekoľkých kostýmov	VIEM / VIEM S POMOCOU / NEVIEM
Viem pridať do programu zvuk	VIEM / VIEM S POMOCOU / NEVIEM
Viem zobraziť postave bublinu na rozprávanie	VIEM / VIEM S POMOCOU / NEVIEM
Viem nastaviť, aby postava reagovala na stlačenie niektorého klávesu na klávesnici	VIEM / VIEM S POMOCOU / NEVIEM
Viem zastaviť vykonávanie scenára (napr. činnosť postáv, prehrávanie zvukov...)	VIEM / VIEM S POMOCOU / NEVIEM

3 UDALOSTI MYŠI A KLÁVESNICE

<i>Tematický celok / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov: <ul style="list-style-type: none"> • analýza problému, • jazyk na zápis riešenia, • pomocou postupnosti príkazov, • pomocou cyklu • pomocou vetvenia Reprezentácie a nástroje: <ul style="list-style-type: none"> • práca s grafikou. 	ZŠ / 5. - 6. ročník / 1 vyučovacia hodina
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> • základy práce v prostredí SCRATCH • riešenie jednoduchých algoritmických úloh bez počítača 	
Ciele	
Žiakom osvojované vedomosti	Žiakom rozvíjané zručnosti a spôsobilosti
Prostredie Scratch <ul style="list-style-type: none"> • ovládať pohyb postavy pomocou myši, • pomocou ťahania postavy vykresľovať čiaru, • vytvoriť udalosť riadenú vstupmi z klávesnice, • zastaviť vykonávanie scenára, • nastavovať zmenu farby a hrúbky pera. 	Informatické myslenie <ul style="list-style-type: none"> • (LOG1) využitím logických zdôvodnení predpokladať správanie sa algoritmov • (LOG6) logicky zdôvodniť zmenu algoritmu/programu • (ALG6) dotvárať nekompletné algoritmy (doplň kód, dokonči program)
<p>Ovládanie projektu myšou alebo klávesnicou je jednou zo základných vlastností interaktivity projektu, zvyšuje záujem žiakov o programovanie a o tvorbu programov, ktoré budú atraktívne pre spolužiakov a kamarátov. Učiteľ musí zvoliť takú úlohu, ktorá je dostatočne a primerane náročná pre žiakov. Prostredie Scratch môže byť stále pre žiakov máťúce, učiteľ musí dbať na dôslednú dekompozíciu úlohy na čiastkové problémy, ktoré riešia scenáre jednotlivých postáv na scéne.</p>	
Dominantné vyučovacie metódy a formy	Príprava učiteľa a pomôcky
Bádateľská metóda (5E). Frontálna forma. Práca vo dvojiciach.	<ul style="list-style-type: none"> • lokálna inštalácia prostredia Scratch alebo prístup na internet s online editorom • pracovný list • pracovné súbory chyt mysicku, kreslenie, mapa
Diagnostika splnenia vzdelávacích cieľov	
Diagnostika pomocou sebahodnotiaceho testu.	

ÚVOD

Žiaci väčšinou bez problémov ovládajú počítačové hry, v ktorých sa na riadenie používa myš alebo klávesy. Na tejto vyučovacej hodine sa rovnakým spôsobom naučia programovať postavy v **Scratch**, ktoré budú reagovať na stlačenie klávesu alebo tlačidla myši.

ZAPOJENIE – CCA 15 MIN.

Učiteľ vedie so žiakmi motivačný rozhovor o počítačových hrách. Kládie im nasledujúce alebo podobné otázky, nabáda ich, aby povedali, aké hry hrajú a ako sa tieto hry ovládajú.

- Aké hry sú pre teba najzaujímavejšie? – predpokladané odpovede žiakov: športové, strategické, akčné, pretekárske a pod.
- Ako sa ovládajú počítačové hry, ktoré hrávaš? – predpokladané odpovede žiakov: myšou, stláčaním kláves, šípkami, medzerníkom a pod.

Ukážeme na dataprojektore projekt **chyt' myšičku**. Jeden žiak hru môže demonštrovať aj na učiteľskom počítači, pričom spolu diskutujeme o jej ovládaní.

Projekt je potrebné spustiť zelenou zástavkou, potom sa postava, ktorá má obrázok myši pohybuje spolu s kurzorom. Mačka sa k nej natáča a pohybuje sa smerom k nej.

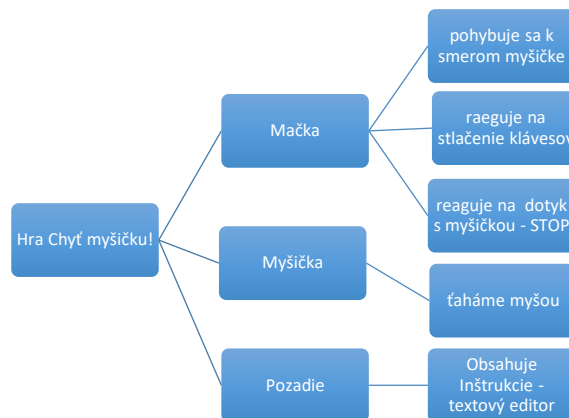
Úloha 1 Otvorte projekt **chyt' mysicku** a zahrajte sa hru na základe inštrukcií v pozadí.



Žiaci by mali identifikovať:

- pozadie/scénu,
- ovládanie hry klávesmi a kurzorom myši,
- neustály pohyb mačky k postave - myšičke.

Následne by mali vedieť rozdeliť úlohu na menšie časti – urobiť dekompozíciu úlohy. Dekompozíciu úlohy robíme spoločne, kreslíme návrhy žiakov na tabuľu alebo žiaci svoje návrhy nakreslia na tabuľu sami. Ukážka dekompozície je na schéme. So žiakmi diskutujeme, ktoré časti by už boli schopní naprogramovať sami.



Sformulujeme ciele hodiny – programovanie **udalostí**: ovládanie postavy myšou, ovládanie programu klávesmi, nastavenie zmien v príkazoch.

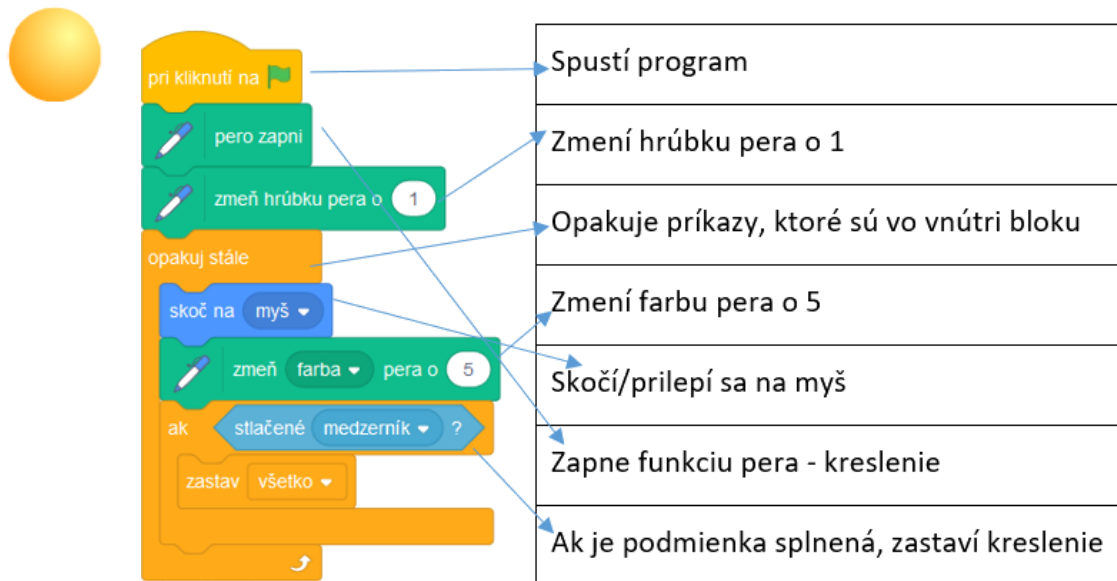
Poznámka

Počas skúšania hry môžete diskutovať aj o tom, ako by sa dala mačka zrýchliť, ako by bolo možné sťažiť naháňanie pre mačku (napr. zmenšením myšičky a pod.)

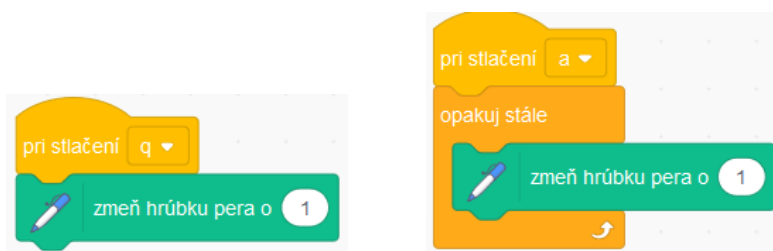
SKÚMANIE - CCA 10 MIN.

V tejto časti hodiny si žiaci otvoria projekt **kreslenie**. V projekte sa kreslí ťahaním postavy – guľôčky. Po stlačení medzerníka sa kreslenie ukončí. Grafická plocha sa zmaže stlačením tlačidla.

Úloha 2 Skúmajte program **kreslenie**. Objavte na čo slúžia nasledujúce príkazy a spojte ich s popismi:



Doplňte do projektu **zmenu hrúbky pera** ovládaním pomocou klávesov:



Zistite rozdiel medzi oboma spôsobmi.

Pridajte ďalšie ovládanie kreslenia pomocou klávesnice.

Žiaci majú v tejto úlohe skúmať a objavovať fungovanie jednotlivých príkazov, ktoré obsahuje daný projekt. Mali by v ňom rozpoznať príkaz, ktorý zabezpečí skok postavy na miesto, na ktorom sa nachádza kurzor **myši**, mali by odhaliť, prečo guľôčka kreslí čiary a tiež zistiť, aký kláves v programe funguje a čo urobí. V pracovnom liste žiaci počas pozorovania spoja príkazy s ich popismi. Pracujú buď samostatne alebo vo dvojiciach. Spolupráca žiakov je vítaná.

Následne žiaci pokračujú v riešení pracovného listu a doplnia príkazy na zmenu hrúbky pera, pričom majú zistiť rozdiel medzi dvomi navrhnutými spôsobmi. V ďalšej fáze **Vysvetlenie** by mali byť schopní slovne sformulovať rozdiel medzi oboma spôsobmi. Šikovnejší žiaci určite navrhnu aj ďalšie doplnenia alebo nastavenia pre tento projekt.

VYSVETLENIE – CCA 5 MIN.

Na základe svojich poznámok v pracovných listoch žiaci vysvetlia, čo zistili a ako fungovali jednotlivé bloky. Učivo nevysvetľujeme, len usmerňujeme žiakov pri vysvetľovaní. V prípade nesprávnej alebo nie úplne jasnej formulácie môžu úlohu vysvetliť ďalší žiaci. Do fázy vysvetľovania by sa malo zapojiť čo najviac žiakov.

Žiaci by mali byť schopní zistiť funkčnosť dvoch blokov: **smenom k myši** a **skoč na myš**. Obidva bloky musia byť v cykloch, aby sa vykonávali neustále, alebo dlhodobo. Predpokladáme nasledujúce formulácie žiakov pri popisovaní rozdielov

- blok **skoč na myš** spôsobuje okamžitý presun postavy na kurzor myši.
- blok **smenom k myši** iba natočí postavu k ukazovateľu myši a až ďalšími blokmi ju pohybujeme.

Úlohy sú zamerané na rozšírenie pochopenia ovládania myšou. Žiaci by si mali spomenúť na nekonečný cyklus – opakuj stále, s ktorým sa už stretli.

V druhej úlohe mali žiaci doprogramovať pre kresliacu postavu (gulôčku) ďalšie funkcie, alebo nastavenia. Vyzveme žiakov, aby slovne sformulovali svoje nápady a riešenia.

ROZPRACOVANIE – CCA 10 MIN.

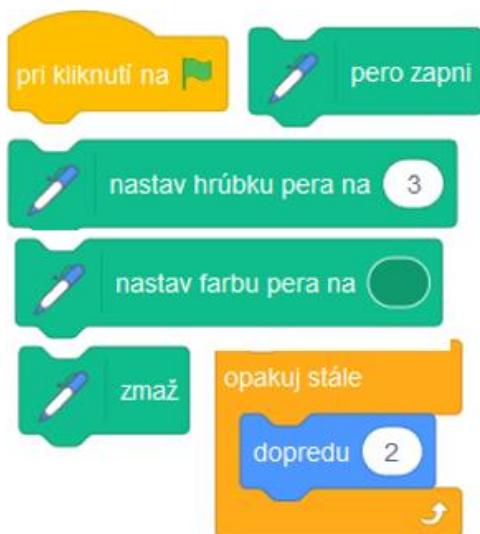
Necháme žiakov, aby sami naprogramovali projekt **mapa**. V pracovnom liste im ponúkame základné programové bloky, podľa ktorých by, aj na základe vedomostí z dnešnej hodiny, mali byť schopní viac-menej samostatne programovať. Učiteľ pozoruje žiakov a pomáha im.

Úloha 3 Vytvorte vlastný projekt – mapa

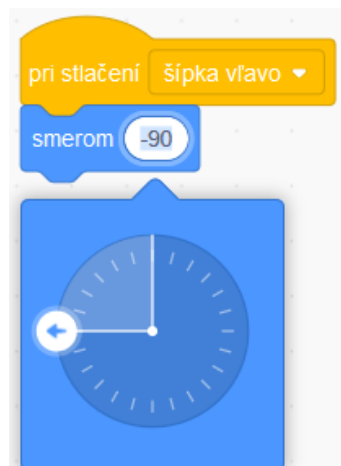
Vytvorte projekt na kreslenie mapy ulíc mesta, kde všetky ulice sú na seba kolmé.

Bloky na ovládanie pera pridáte výberom rozšírenia v ľavom dolnom rohu – **Rozšírenie Pero**.

1. Zvoľte postavičku na kreslenie.
2. Vytvorte program, kde po kliknutí na zelenú zástavu sa bude postava neustále pohybovať, ťahať za sebou čiaru a meniť smer podľa stlačených šípok na klávesnici. Použite nasledovné bloky:

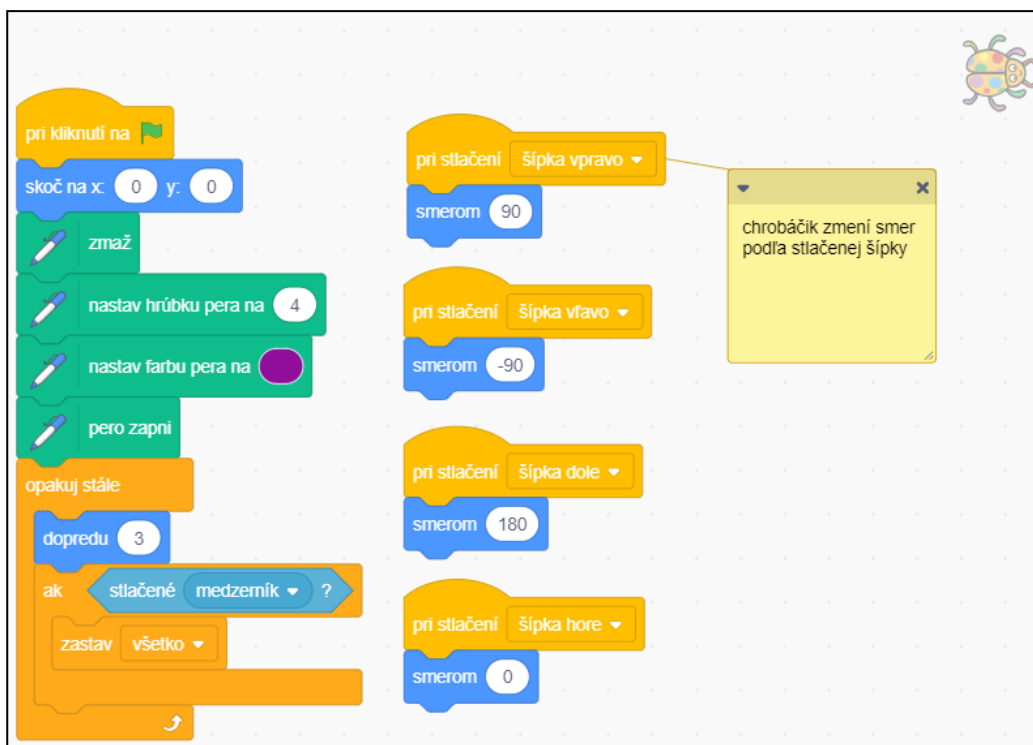


Smerovanie postavy môžete meniť napríklad takto:



3. Doprogramujte projekt tak, aby sa postava pohybovala v smere stlačených šípok HORE, DOLE, VPRAVO a VĽAVO.
4. Skúste zmeniť farbu, hrúbku pera, rýchlosť postavy.
5. Navrhňte vhodné zastavenie kreslenia.

Možné riešenie úlohy:



Úloha je zameraná na naprogramovanie nového projektu tak, aby sa postava z projektu pohybovala neustále primeranou rýchlosťou a aby menila smer podľa toho, ktorá šípka bola stlačená. Je to rozšírenie predchádzajúcich úlohy o ovládanie postavy šípkami na klávesnici.

Ak je ešte zostávajúci čas, žiaci môžu niektoré navrhnuté zlepšenia realizovať, hľadajú riešenia v zoznamoch blokov. Cieľom je vytvoriť samostatne, bez pomoci učiteľa vlastnú aplikáciu s využitím ovládania klávesnicou a myšou. Ďalšie riešenie je uvedené aj v projekte **mapa2**.

HODNOTENIE – CCA 5 MIN.

Žiaci vyplnia sebahodnotiaci test, ktorý slúži aj na zhrnutie nových poznatkov. Ak si žiaci z pracovných listov vytvoria portfóliovú zbierku napr. v rýchloviazači, tieto poznámky im môžu slúžiť ako zhrnutie vedomostí z každej vyučovacej hodiny. Je na rozhodnutí učiteľa, či test, alebo najlepšie projekty ohodnotí aj klasifikačnou známku.

Spojte čiarami jednotlivé popisy ovládania s príslušným blokom príkazu .		
<i>Postava sa stále otáča k postave s menom myšička.</i>		
<i>Postava kreslí čiaru.</i>		
<i>Hra sa zastaví stlačením klávesu medzerník.</i>		
<i>Postava sa pohybuje spolu s kurzorom myši.</i>		
<i>Zmení smer postavy po stlačení klávesu.</i>		

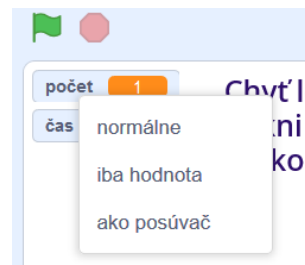
4 ZADÁVAME VSTUPY, VYTVÁRAME PREMENNÉ

<i>Tematický celok / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov: <ul style="list-style-type: none"> analýza problému, jazyk na zápis riešenia, postupnosť príkazov Reprezentácie a nástroje: <ul style="list-style-type: none"> práca s grafikou 	ZŠ / 5. - 6. ročník / 1 vyučovacia hodina
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> základy práce v prostredí Scratch riešenie algoritmických úloh bez počítača ovládanie prostredia Scratch, orientácia v skupinách príkazov 	
Ciele	
Žiakom osvojované vedomosti	Žiakom rozvíjané zručnosti a spôsobilosti
Prostredie Scratch <ul style="list-style-type: none"> pochopiť, čo je to premenná, na Scéne zobraziť premennú rôznymi spôsobmi, je schopný vytvárať bubliny, v ktorých je potrebné spojiť hodnoty premenných, využiť bloky zo skupiny Operácie na výpočet hodnoty premennej, navrhnuť bloky, ktoré umožnia používateľovi zadať vstupy programu. 	Informatické myslenie <ul style="list-style-type: none"> (LOG1) využitím logických zdôvodnení predpokladať správanie sa algoritmov (ALG6) dotvárať nekompletné algoritmy (doplň kód, dokonči program) (ALG7) vylepšovať existujúce algoritmy (zlepšenie efektívnosti, rozšírenie algoritmu na väčšiu množinu vstupov, rozšírenie funkcionality) (DEK1) lineárna dekompozícia – lineárne rozdeliť problémy/procesy na menšie časti tak, aby sa dali využiť pre dosiahnutie cieľa (určenie objektov/správania sa v nejakej hre – pozadie/postava/udalosti)
Podobne, ako v zložitejších programovacích prostrediach, aj prostredie Scratch umožňuje používať premenné a počítať z nich rôzne hodnoty. Už v jednoduchých počítačových hrách sa pripočítavajú body, mení sa smer pohybu, alebo sa spočítavajú hodnoty premenných. V tejto metodike ponúkame žiakom projekty, ktoré využívajú zobrazovanie premenných, výpis spájaných hodnôt, resp. matematických operácií s premennými. Upozorňujeme na dôvod, prečo je potrebné vstupné hodnoty uložiť do premennej.	
Dominantné vyučovacie metódy a formy	
Bádateľská metóda (5E) Frontálna forma, práca vo dvojiciach	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> lokálna inštalácia prostredia Scratch alebo prístup na internet pracovný list pre žiaka pracovné súbory pre učiteľa a žiaka chyt lopticku, nasobime, nakup 	
Diagnostika splnenia vzdelávacích cieľov	
Diagnostika pomocou sebahodnotiaceho testu a na základe analýzy práce žiakov.	

ÚVOD

Hlavným cieľom hodiny je ukázať žiakom **vytváranie a používanie premenných**. Využijeme na to projekty na výpočet násobku čísla a výpočet ceny nákupu.

Premennú pridávame v skupine príkazov **Premenné**, výberom **Nová premenná**. Zmenu zobrazenia premennej meníme kliknutím pravým tlačidlom myši.



Scratch ponúka štyri typy zobrazenia: normálne, iba hodnota alebo posúvač. Ak máme premennú zobrazenú ako posúvač, potom opäť kliknutím pravým tlačidlo myši nastavíme minimum a maximum hodnôt posúvača.

Žiaci pracujú na základe pracovných listov, blokové príkazy sú uvedené farebne. Ak nemáte možnosť žiakom vytlačiť pracovné listy farebne, odporúčame pracovné listy premietiť projektorom, aby žiaci sa mohli lepšie zorientovať medzi skupinami príkazov.

ZAPOJENIE – CCA 5 MIN.

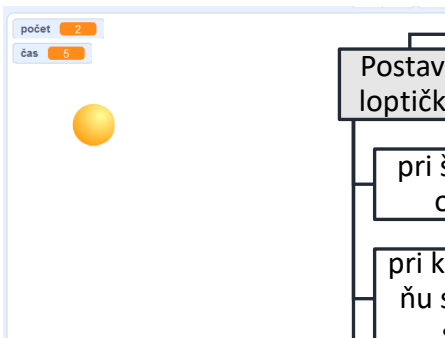
Hodinu začneme motivačným rozhovorom o počítačových hrách.

Z čoho sa skladajú bežne počítačové hry? – očakávame napríklad takéto odpovede žiakov: V hrách sa zbierajú body a zisťuje sa skóre hráčov. Pomocou myši alebo klávesov ovládame postavy/objekty.

Vyzveme žiakov, aby si zahrali hru, ktorá je v projekte **chyt loptičku** Ich úlohou je počas 20 sekúnd klikať na pohybujúcu sa loptičku a získať čo najlepšie skóre.

Úloha 1 Otvorte si projekt **chyt loptičku** a zahrajte sa hru.

Zakrúžkujte udalosti, ktoré už by ste vedeli naprogramovať.



```
graph TD
    Root[Chyt loptičku] --> Postava[Postava loptička]
    Root --> Zmeny[zmeny]
    Postava --> P1[pri štarte sa objaví]
    Postava --> P2[pri kliknutí na ňu sa zmení skóre]
    Postava --> P3[po časovom limite sa skryje]
    Zmeny --> Počet[počet - skóre]
    Zmeny --> Uplynutý[uplynutý čas]
    Počet --> P4[vynuluje sa pri štarte]
    Počet --> P5[pri kliknutí sa zmení]
    Uplynutý --> P6[začína s nastaveným časom]
    Uplynutý --> P7[pravidelne sa zmení]
```

Ak sa žiaci zatiaľ nestretli s príkazmi ukáť a skry, ktoré slúžia na objavenie a zmiznutie postavy, ukážeme tieto príkazy. Sformulujeme cieľ hodiny: naučiť sa ovládať **meniace sa** údaje počas behu programu, zavádzame premenné.

SKÚMANIE – CCA 15 MIN.


V tejto fáze vyučovacej hodiny necháme žiakov pracovať s pracovnými listami. V projekte **nasobíme**, sú použité vstupy a výstupy programu z kategórií **Zisťovanie** a **Vzhľad**. V projekte je postava dievčatka, ktorá vyzve žiaka, aby zadal nejaké číslo a dievčatko mu povie násobok 10. Úlohou žiakov je pochopiť, ako program funguje, čo znamenajú jednotlivé príkazy.

Žiaci pracujú sami alebo vo dvojiciach, ale pracovný list má každý žiak vlastný. Svoje riešenia budú žiaci kontrolovať spoločne vo fáze **Vysvetlenie**.

Úloha 2 Preskúmajte projekt Násobíme a čiarami spojte jednotlivé príkazy a popisy.		
		Spojenie 2 textov (naraz sa vypíše text a hodnota odpovede na otázku).
		Blok otázka, teda výzva na zadanie odpovede a jej uloženie.
		Uložená odpoveď.
		Postava si myslí text počas sekundy.
		Zadané číslo sa vynásobí číslom 10.
		Postava hovorí text počas 2 sekúnd.

Úloha 3 je zameraná na dôsledné pochopenie vstupnej hodnoty. Nabádajte žiakov ku opätovnému spúšťaniu programu **nasobime** a zadávaniu čísel kladných, záporných, desatinných, rôznych mien. Žiaci by si tu mohli uvedomiť, že hodnota odpovede z 1 polohy je v druhej polohe už zabudnutá.

Úloha 3 Aká je hodnota **odpoveď** v jednotlivých fázach vykonávania programu?



	Hodnota odpoveď na mieste označenom 1.	Hodnota odpoveď na mieste označenom 2.
1. pokus		
2. pokus		
3. pokus		

Upravte program tak, aby zadané číslo vynásobil iným, napríklad dvojciferným. Program otestujte.

Necháme žiakov **experimentovať**, aby upravili program na násobenie dvojmiestnym, alebo väčším číslom. Svoje riešenia si môžu vyskúšať aj vo dvojiciach. Pri zmene činiteľa v násobení by mali zmeniť aj výzvu na odpoveď: Mysli si číslo....

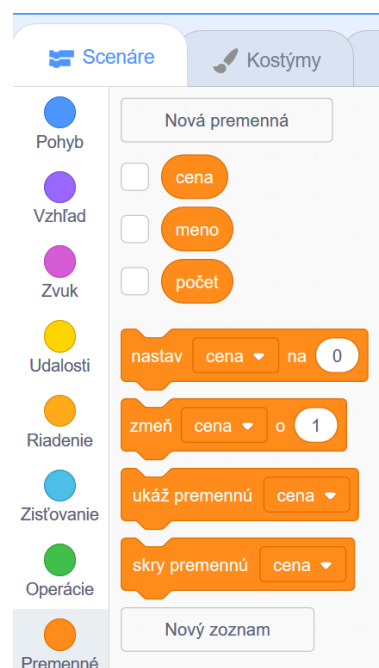
VYSVETLENIE – CCA 5 MIN.

Na základe svojich poznámok v pracovných listoch žiaci vysvetlia, čo zistili, ako fungovali jednotlivé bloky. Učivo nevysvetľujeme, len usmerňujeme žiakov pri vysvetľovaní. V prípade nesprávnej alebo nie úplne jasnej formulácie môžu úlohu vysvetliť ďalší žiaci. Do fázy vysvetľovania by sa malo zapojiť čo najviac žiakov.

Zhrnutie: Ak chceme zadané odpovede uchovať, musíme pre nich vytvoriť úložisko – premennú.

Frontálnym spôsobom na dataprojektore ukážeme žiakom spôsob vytvorenia premennej. Pri vytvorení novej premennej sa objavia 4 ďalšie programové bloky, ktoré umožnia nastaviť, zmeniť, ukázať a skryť príslušnú premennú.

To, čo chceme nastaviť alebo zmeniť sa nastaví pomocou výberu v bloku.



ROZPRACOVANIE

V tejto fáze žiaci vytvoria svoj vlastný projekt na základe predchádzajúcich skúseností.

Úloha 4 Navrhňte projekt s názvom **nakup**, v ktorom sa predávajúci opýta **Ako sa voláš?**

Potom sa program opýta, za akú cenu si chcem kúpiť koláčik a koľko kusov koláčikov chcem kúpiť.

Program potom vypočíta koľko mám zaplatiť.

Vytvorte nové premenné: **meno** **cena** **počet**

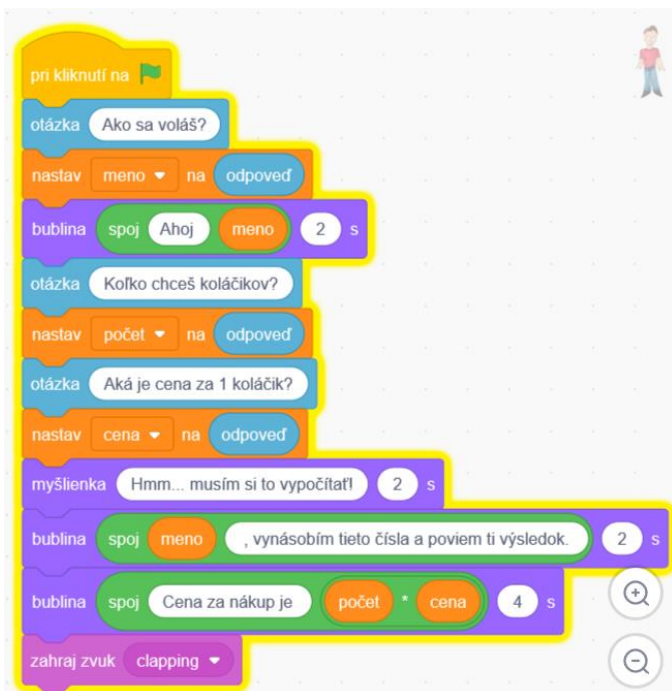
Po výzve na odpoveď uložte odpoveď do príslušnej premennej:



Žiaci majú pochopiť dôvod pre používanie premenných.

Ak je to potrebné, môžeme nakresliť na tabuľu krabičky *odpoveď*, *cena*, *počet* a vysvetliť žiakom, ako sa počas behu programu prepisuje hodnota odpovedí.

Možné riešenie úlohy tu uvádzame ako obrázok, učiteľ ho nájde v projekte **nakup**.



Žiaci celý čas pracujú vo dvojiciach, podľa možnosti vedľa seba a každý na vlastnom počítači, aby získali zručnosti v prostredí **Scratch** a aby sa naučili pracovať s blokmi prostredia Scratch.

Poznámka

Ak sú niektorí žiaci skôr hotoví, môžeme im navrhnúť, aby vylepšili prvú úlohu chyt lopticku.

Žiaci môžu navrhnúť napr. že si hráč zvolí, ako dlho chce hrať. Alebo, že hru zrýchlia. Učiteľ nájde takéto riešenie v projekte **chyt lopticku2**.

HODNOTENIE – CCA 3 MIN.

Žiaci vyplnia sebahodnotiaci test, ktorý zároveň slúži aj na zhrnutie nových poznatkov. Ak si žiaci z pracovných listov vytvoria portfóliovú zbierku napr. v rýchloviazači, tieto poznámky im môžu slúžiť ako zhrnutie vedomostí z každej vyučovacej hodiny. Je na rozhodnutí učiteľa, či test, alebo najlepšie projekty ohodnotí aj klasifikačnou známku.

Spojte čiarami jednotlivé popisy ovládania s príslušným blokom príkazu .		
Odpoveď na otázku sa uloží do premennej.		otázka Koľko chceš koláčikov?
Postava zadá otázku.		počet * cena
Matematická operácia, ktorá vynásobí hodnotu dvoch premenných.		nastav počet na odpoveď
Zmení sa hodnota premennej o 1.		skoč na náhodná pozícia
Postava skočí na náhodnú pozíciu.		zmeň počet o 1

5 POSIELAME SPRÁVY

<i>Tematický celok / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov: <ul style="list-style-type: none"> • analýza problému, • jazyk na zápis riešenia, • postupnosť príkazov Reprezentácie a nástroje: <ul style="list-style-type: none"> • práca s grafikou 	ZŠ / 5. - 6. ročník / 1 vyučovacia hodina
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> • základy práce v prostredí Scratch • riešenie algoritmických úloh bez počítača • ovládanie prostredia Scratch, orientácia v skupinách príkazov 	
Ciele	
Žiakom osvojované vedomosti	Žiakom rozvíjané zručnosti a spôsobilosti
Prostredie Scratch <ul style="list-style-type: none"> • pochopiť, akým spôsobom sa vytvára správa, ako sa správy posielajú a ako postavy reagujú na správy, • dokáže navrhnuť scenár, ktorý prehrá zvuky z knižnice, • vybrať si alebo nakresliť a zmeniť pozadie pre Scénu, • vybrať pre postavu viac kostýmov a v scenári meniť postave kostýmy. 	Informatické myslenie <ul style="list-style-type: none"> • (LOG1) využitím logických zdôvodnení predpokladať správanie sa algoritmov • (ALG3) vytvárať vlastné algoritmy riešiace problém/časti problému • (DEK1) lineárna dekompozícia – lineárne rozdeliť objekty/problémy/procesy na menšie časti tak, aby sa dali využiť pre dosiahnutie cieľa • (VZO1) rozpoznať časti problému, ktoré majú podobné vlastnosti
Dominantné vyučovacie metódy a formy	
Bádateľská metóda (5E) Frontálna forma. Práca vo dvojiciach.	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> • lokálna inštalácia prostredia Scratch3 alebo prístup na internet s online editorom • pracovný list pre žiaka • projekty pre učiteľa a žiaka: rozhovor, rozhovor2, tanec, tanec2, klavir • slúchadlá alebo reproduktor 	
Diagnostika splnenia vzdelávacích cieľov	
Diagnostika pomocou sebahodnotiacej rubriky a na základe analýzy práce žiakov.	

ÚVOD

V tejto metodike chceme predstaviť **Scratch** ako nástroj na efektné a užitočné programy, ktorými sa môžu žiaci prezentovať pred spolužiakmi a ktoré podporujú ich kreativitu a sú zaujímavé.

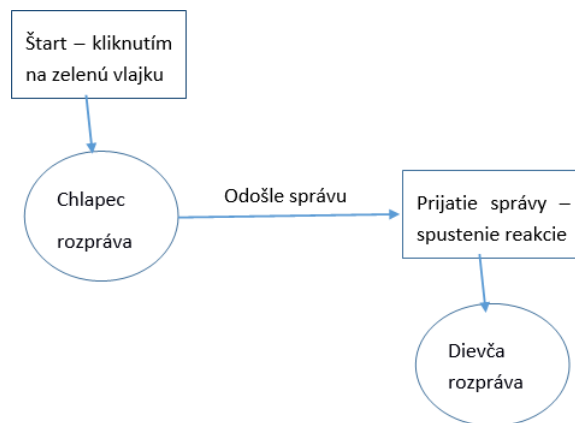
ZAPOJENIE – CCA 5 MIN.

Aj túto hodinu začneme **motivačným rozhovorom** a otázkami typu: čo je na počítačových hrách najzaujímavejšie, čo žiakov najviac baví pri hraní. Predpokladáme, že medzi odpoveďami sa vyskytnú aj tieto:

- pohyb postáv, zmena pozadia, udalosti, prechod do iných levelov, príbeh v hre
- ak žiaci povedia aj iné vlastnosti hier, tak ich informujeme, že tieto ďalšie vlastnosti budú obsahom ďalších hodín

Vyzveme žiakov, aby sa pozreli na projekt **rozhovor**, v ktorom nájdú jednoduchý rozhovor dievčaťa s chlapcom. Jednotlivé akcie sú riadené správami. Keď chlapec dohovori, vyšle správu. Keď dievča prijme správu, začína hovoriť.

Žiaci by mali byť schopní všimnúť si, že projekt je riadený – pomocou správ. Z predchádzajúcich hodín by už mali vedieť dekompozíciu úlohy. Kreslenie schémy robíme spoločne, tak že učiteľ, alebo žiaci sami zakresľujú časti schémy na tabuľu. Upozorňujeme žiakov, aby si všímali udalosti, ktoré **riadia** vykonávanie programu.



SKÚMANIE – CCA 20 MIN.

V **úlohe 1** žiaci vyplnia samostatne tabuľku. **Modrou farbou** uvádzame riešenie pre učiteľa.

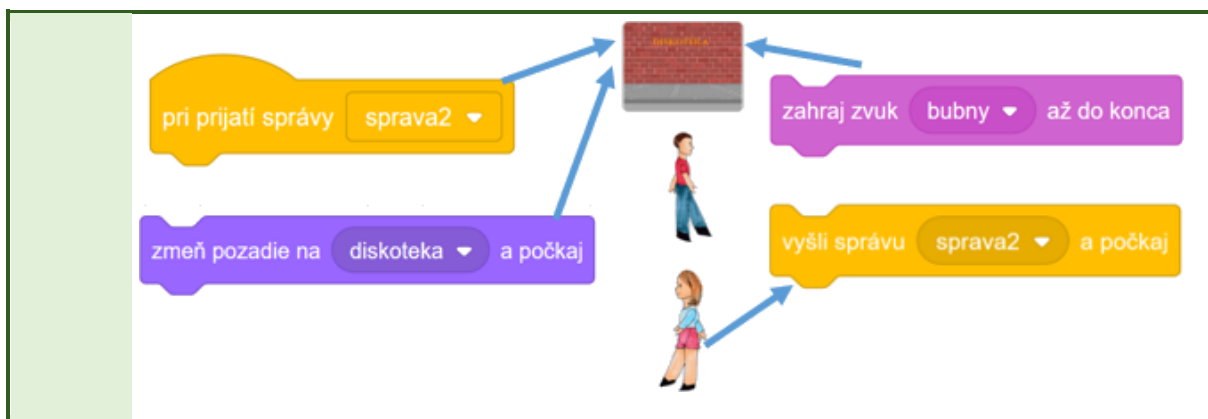
V žiackych pracovných listoch samozrejme toto riešenie nie je uvedené.

Úloha 1	Preskúmajte projekt rozhovor . Prezrite si všetky postavy, scénu a odpovedzte:	
Otázka	Odpoveď	
Koľko pozadí je použitých v projekte a ako sa volajú?	2: stena, diskotéka	
Koľko kostýmov má chlapec?	6	
Koľko kostýmov má dievča?	6	
Kto posiela správu „otazka“ ?	chlapec	
Kto na správu reaguje?	dievča	

Úloha 1 je zameraná na preskúmanie posielania správ medzi postavami a počtu kostýmov pozadia a postáv. Žiaci by si mali uvedomiť, že medzi postavami sa posiela iba jedna správa, ktorá má svoje meno „otazka“. Celý program spúšťa zelená zástavka, po nej chlapec osloví dievča a to, keď dostane správu, zamyslí sa, zasmieje sa (spustí sa zvuk) a odpovie chlapcovi. Tým sa projekt končí.

V **úlohe 2** žiaci samostatne doprogramujú projekt podľa zadania. Na obrázku vidíme, ako by mali žiaci nakresliť čiary medzi postavami. Hotový program nájde učiteľ v projekte **rozhovor2**.

Úloha 2	Doprogramujte projekt rozhovor tak, aby sa po skončení rozprávania dievčatka preplo pozadie scény na diskoteka a aby sa potom spustila hudba. Spojte čiarami bloky , ktoré ste doprogramovali s postavami, ku ktorým ste bloky pridali.
----------------	--



V **úlohe 3** je rozhovor medzi dvoma chlapcami, ktorí sa predvádzajú pri tancovaní. Je to Adam a Ivan. Projekt sa spúšťa zelenou zástavkou a riadi sa posielaním správ.

Úloha 3 Preskúmajte scenáre postavičiek a pozadia v projekte **tanec**.
Ktorá postava alebo pozadie vykonáva tieto príkazy?

Príkaz	Ktorá postava/pozadie vykonáva príkaz?
vyšli správu tanec2	Adam
pri prijatí správy tanec2	Ivan, Scéna
zmeň pozadie na chodník	Scéna
zahraj zvuk drum machine až do konca	Scéna
d'alší kostým čakaj 0.5 s	Adam, Ivan

V akom poradí vykonávajú príkazy jednotlivé postavy a **Scéna**?


1.	Adam, Scéna
2.	Ivan, Scéna
3.	

V **úlohe 3** žiaci pravdepodobne prídu na to, ktorá postava začína vykonávať svoj program po stlačení zelenej zástavky. Hoci tabuľka obsahuje tri riadky, budú nám stačiť dva. Pri spustení programu totiž súčasne spustíme aj Adama aj **Scénu**, a potom, po odoslaní správy, opäť zareaguje na správu aj Ivan aj **Scéna** zmenou svojho pozadia.

Úloha 4 je samostatné programovanie s pridaním ďalšej postavy.

Úloha 4 Do projektu **tanec** vložte postavičku kráčajúceho dievčaťa Avery Walking.

Jej pohyb naprogramujte tak, aby sa začala a skončila animovať súčasne s Ivanom. **Animácia** postavy je postupná **zmena kostýmu**.



Avery Walk...

Pri riešení tejto úlohy by mali žiaci dokázať pridať ďalšiu postavu. Ak je to potrebné, žiakov privedieme k riešeniu úlohy diskusiou, počas ktorej vedieme žiakov k tomu, aby si uvedomili, aký scenár, resp. scenáre, by mala postava vykonávať:

- Pri spustení projektu zelenou vlajkou postava Avery ešte nemá byť na scéne, preto je vhodné skryť ju.
- Postavu Avery je vhodné umiestniť na konkrétnu pozíciu a po prijatí správy ju ukázať.
- Pohyb Avery zabezpečíme zmenami jej kostýmu a vhodnou rýchlosťou bloku **dopredu**.

Riešenie úlohy nájde učiteľ v projekte **tanec2**.

VYSVETLENIE – CCA 5 MIN.

Vyzveme žiakov, aby nám predviedli svoje riešenia. Či už na dataprojektore alebo tým, že ostatní žiaci prídu k ich monitoru.

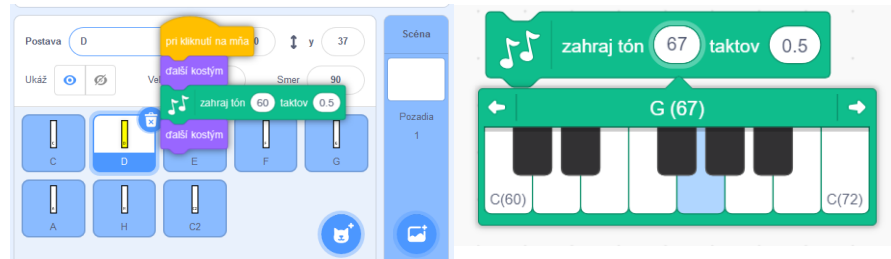
ROZPRACOVANIE – CCA 10 MIN.

Teraz by mali žiaci pracovať samostatne, každý z nich by mali byť sústredený. Prípadný rozhovor medzi žiakmi o projekte alebo problémoch, na ktoré narazia, im však povolíme.

Úloha 5 Preskúmajte projekt **klavir**.

Ako by sa na klavíri dalo hrať? Poskladajte jednotlivé bloky tak, aby sa pri kliknutí myšou na kláves na obrázku zahral príslušný zvuk. Naprogramujte aj zmenu kostýmu počas hrania tónu, napr. kláves na chvíľu zafarbite.

V **úlohe 5** majú žiaci naprogramovať každú postavu – jednotlivé klávesy klavíra. Úloha je zameraná na prehrávanie tónov pri stláčaní klávesov.



Aby hra na klavíri bola čo najrealistickejšia, budeme meniť aj kostýmy jednotlivých klávesov na obrázku klavíra.

Žiaci majú v projekte už pripravené postavy, ich úlohou je naprogramovať správanie, teda program pre jeden kláves a potom len skupinu blokov prekopírovať pre druhý kláves.

Kopírovanie sa robí **presunutím skupiny blokov na ikonu postavy** v zozname postáv. Tento postup je vhodné žiakom ukázať a potom ich nechať zrealizovať ho pre ďalšie postavy.

Po skopírovaní príkazov pre nový kláves, treba každému z nich **nastaviť správny tón**. Žiaci by mohli poznať tóny stupnice z hudobnej výchovy, ale aj v prípade, že nepoznajú, môžu tón správne nastaviť, pretože pri kliknutí do políčka pre zmenu tónu sa objaví meno tónu a do políčka sa následne vpiše pre vybraný tón správne číslo.

Úloha je určená na samostatnú prácu žiakov, na rozvoj ich tvorivosti a na podporu spolupráce medzi nimi. Súčasne podporuje zapojenie umeleckých predpokladov, špeciálne hudobné nadanie žiakov. Riešenie úlohy nájde učiteľ v projekte **klavir2**.

Úloha 6 je určená na samostatnú prácu žiakov, zrejme ju vyriešia len tí rýchlejší. Žiaci by mali vedieť navrhnúť ďalšie vylepšenia programu klavír, svoje nápady prípadne zaznamenať a aj zrealizovať. Cieľom je vytvoriť samostatne, bez pomoci učiteľa, vlastnú aplikáciu.

Úloha 6	Nepovinná úloha. Pridajte do projektu klavir iné funkcionality, napr. pri stlačení niektorého tlačidla na klávesnici nech sa zahrá celá melódia. Vymyslite a nižšie zapíšte aj ďalšie vylepšenia projektu klavir . Svoje nápady zrealizujte.
----------------	---

HODNOTENIE – CCA 5 MIN.

Na evalváciu slúži sebahodnotiaca rubrika, pomocou ktorej žiaci zaškrtnutím sami zhodnotia úroveň osvojenia kľúčových vedomostí a zručností, ako aj splnenie cieľov hodiny.

?	Viem? <input checked="" type="checkbox"/>
Posielať správy.	<input type="checkbox"/>
Spúšťať animáciu postáv po prijatí správy.	<input type="checkbox"/>
Opakovane animovať postavu.	<input type="checkbox"/>
Meniť pozadie projektu po prijatí správy.	<input type="checkbox"/>
Prehrávať hudbu a spúšťať ju udalosťou.	<input type="checkbox"/>

6A POUŽITIE KAMERY

<i>Tematický celok / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov: <ul style="list-style-type: none"> • analýza problému, • jazyk na zápis riešenia, • postupnosť príkazov Reprezentácie a nástroje: <ul style="list-style-type: none"> • práca s grafikou 	ISCED 2 / 5. - 6. ročník
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> • základy práce v prostredí Scratch • riešenie algoritmických úloh bez počítača • ovládanie prostredia Scratch, orientácia v skupinách príkazov, • pohyb postáv, zmena kostýmov a pozadia 	
Ciele	
Žiakom osvojované vedomosti	Žiakom rozvíjané zručnosti a spôsobilosti
Prostredie Scratch <ul style="list-style-type: none"> • pracovať s kamerou – ovládanie snímania, • dokáže využiť nastavenia a hodnoty zo skupiny Zisťovanie, • je schopný pripraviť scenár na ovládanie postáv pomocou snímania kamerou. 	Informatické myslenie <ul style="list-style-type: none"> • (ALG3) vytvárať vlastné algoritmy riešiace problém/časti problému (postupnosti krokov na realizáciu nejakej činnosti vedúcej k cieľu) • (DEK1) lineárna dekompozícia – lineárne rozdeliť objekty/problémy/procesy na menšie časti tak, aby sa dali využiť pre dosiahnutie cieľa (detekcia prvkov/určenie objektov/správania sa v nejakej hre – pozadie/postava/udalosti) • (VZO1) rozpoznať časti objektu/problému, ktoré majú rovnaké/podobné vlastnosti/pravidlá správania sa • (VZO5) preniesť/použiť vzory/myšlienky/riešenia z jedného problému na druhý problém
Metodika rieši problém používania externých vstupov počítača na ovládanie programu. Scratch umožňuje fyzické ovládanie programu, ktoré nie je bežné v klasických programovacích prostrediach pre žiakov tejto vekovej skupiny. Kamera je nástroj na fyzické programovanie, ide o rozšírenú realitu, ktorá riadi prepínanie pozadí a spúšťanie animácií a pohybu postáv, prehrávanie zvukov a pod. Ak používate online prostredie, v prehliadači je potrebné povolíť používanie kamery .	
Dominantné vyučovacie metódy a formy	
Bádateľská metóda (5E). Frontálna forma. Práca vo dvojiciach.	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> • lokálna inštalácia prostredia Scratch alebo prístup na internet s online editorom • interná alebo externá kamera počítača • pracovný list pre každého žiaka • súbory pre učiteľa a žiaka: futbal, klavir, fotoaparát, odbijanie, orchester 	
Diagnostika splnenia vzdelávacích cieľov	
Diagnostika pomocou sebahodnotiacej rubriky a na základe analýzy práce žiakov.	

ÚVOD

Počas tejto vyučovacej hodiny chceme žiakom predstaviť **Scratch** ako nástroj na **fyzické programovanie**. Ukážeme im, že **vstupmi z kamery** vieme riadiť beh programu. Takto koncipované programy sú nástrojom na uvedenie žiakov do ovládania programu **reálnymi senzormi** a podporujú aj ich kreativitu. V úvode hodiny žiaci pracujú s programami, ktoré ich majú zaujať a vtiahnuť do programovania projektov ovládaných kamerou. Metodika s použitím kamery je alternatívou metodiky **6B Použitie mikrofónu**. Učiteľ si vyberie **jednu** z týchto dvoch metodík podľa technického vybavenia učebne. Ak má učiteľ k dispozícii aj kamery aj mikrofóny, môže použiť obe alternatívy na **dvoch samostatných** vyučovacích hodinách.

ZAPOJENIE – CCA 10 MIN.

Hodinu začíname diskusiou, v ktorej môžeme využiť nasledujúce otázky: Má niekto doma poplašné zariadenie? Viete popísať jeho funkciu? Ako sa spúšťa domáci alarm? Čo musí byť v dome namontované, aby sa alarm spustil?

Možné odpovede: Alarm sa spúšťa pohybom osoby alebo zvieratá pred senzorom. Keď senzor zaregistruje pohyb, tak spustí sirénu – zvuk.

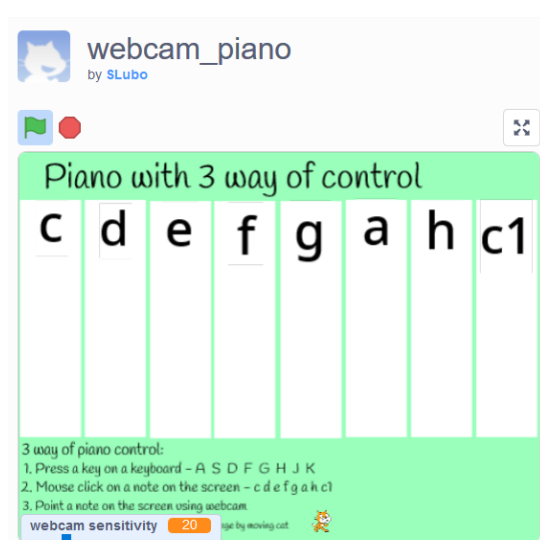
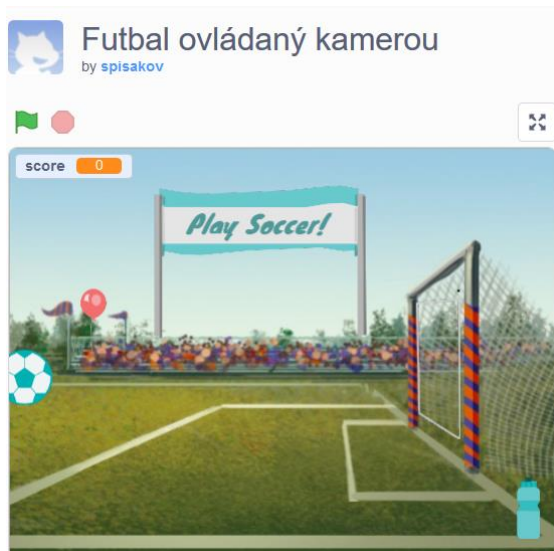
Žiakom predstavíme ciele hodiny – snímanie kamerou, riadenie programu pomocou snímania - pohyb postáv, prehrávanie zvukov a zmena pozadí.

Úloha 1

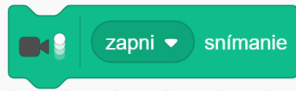
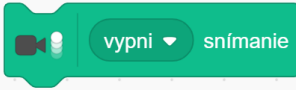
Vyskúšajte projekty **Futbal ovládaný kamerou** a **klavír**.

Zahrajte si na **klavíri**, ktorý sa dá ovládať tromi spôsobmi:

- Kliknutím na klávesy myšou.
- Pomocou tlačidiel na klávesnici A, S, D, F, G, H, J.
- Ukazovaním na klávesy na obrazovke pomocou kamery. Nezabudnite si zapnúť kameru!

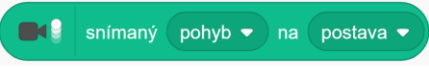
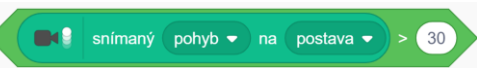


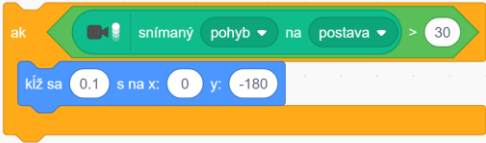
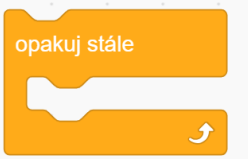
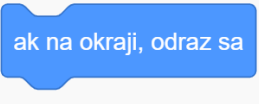
SKÚMANIE – CCA 20 MIN.

Úloha 2	Preskúmajte projekt fotoaparát . Experimentujte s nastaveniami pre snímanie a odpovedajte na otázky v tabuľke:	
	Otázka	Odpoveď
	Akým príkazom sa zapína zobrazovanie na scéne?	Zapni snímanie (Video snímanie) 
	Čo znamená príkaz nastav snímanie na 100 % ?	Zobrazenie nepriehľadného pozadia.
	Čo znamená príkaz nastav snímanie na 0 % ?	Zobrazenie pozadia s najväčšou priehľadnosťou.
Akým príkazom sa vypína zobrazovanie na scéne?	Vypni snímanie (Video snímanie) 	

Žiacke odpovede sme do metodiky zapísali **modrým písmom**. Žiaci by mali experimentovať s nastavením parametrov pre priehľadnosť, časový interval, a pod.

Samozrejme tento fotoaparát neurobí fotografiu, len imituje fotografovanie so zvukovým signálom, aj zapnutie a vypnutie.

Úloha 3	Preskúmajte projekt odbijanie_dribling . Experimentujte s použitými príkazmi a meňte jednotlivé parametre, aby ste pochopili, ako príkazy fungujú. Na základe svojich pozorovaní popíšte príkazy.	
	Príkaz	Vysvetlenie príkazu
	Postava bude reagovať na pohyb zachytený kamerou, ak sa bude objekt pred kamerou dotýkať postavy	
	Určenie citlivosti snímania (senzibility),	

	kedy začne postava reagovať na pohyb.
	Rozhodovanie, ak postava zaregistruje pohyb, tak sa bude kĺžať smerom k bodu so súradnicami 0 a -180
	Ako je zabezpečené neustále opakovanie pohybu lopty.
	Ako je zabezpečené odrážanie lopty.

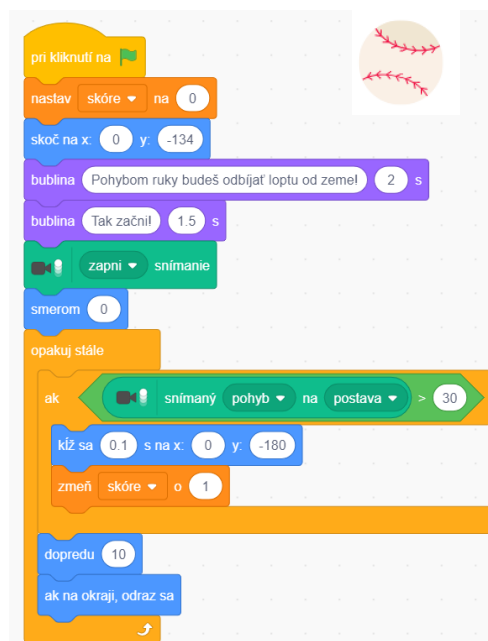
Na základe experimentovania s použitými príkazmi a zmenou jednotlivých parametrov, by žiaci pri riešení **úlohy 3** mali pochopiť, ako príkazy fungujú. Na základe svojich pozorovaní majú popísať použité príkazy v projekte. Riešenie, ktoré by mohli zapísať žiaci do pracovného zošita, sme do metodiky zapísali **modrým písmom**. Úlohu skontrolujeme v nasledujúcej fáze hodiny.

Úloha 4 Do projektu **odbijanie_dribling** doplňte inú loptu a naprogramujte ju tak, aby sa pohybom ruky odrážala od horného okraja plochy.

Pridajte prehratie zvuku a počítanie skóre – počet odbití rukou. Skóre sa vždy pri spustení programu musí vynulovať!

Možné riešenie úlohy nájde učiteľ v projekte **odbijanie_dribling_riesenie**.

Žiaci môžu projekt vylepšiť aj podľa svojich ďalších návrhov.



VYSVETLENIE – CCA 5 MIN.


Na základe svojich poznámok v pracovných listoch žiaci vysvetlia, čo zistili, ako fungovali jednotlivé bloky. Učivo nevysvetľujeme, len usmerňujeme žiakov pri vysvetľovaní. V prípade nesprávnej alebo nie úplne jasnej formulácii môžu úlohu vysvetliť ďalší žiaci. Do fázy vysvetľovania by sa malo zapojiť čo najviac žiakov.

Žiakom na usmernenie vysvetľovania môžeme žiakom klásť napr. aj takéto otázky:

- Kedy sa spúšťa snímanie kamerou a ktorým objektom? (Odpoveď: Na začiatku programu, väčšinou ako program pre Scénu.)
- Aký dôvod má nastavenie priehľadnosti snímania? (Odpoveď: Aby sme videli na scéne seba aj loptu, resp. klávesy.)
- Vedeli by ste iným spôsobom naprogramovať pohyb lopty smerom nadol? (Odpoveď: Kízaním k určitému bodu na podlahe.)

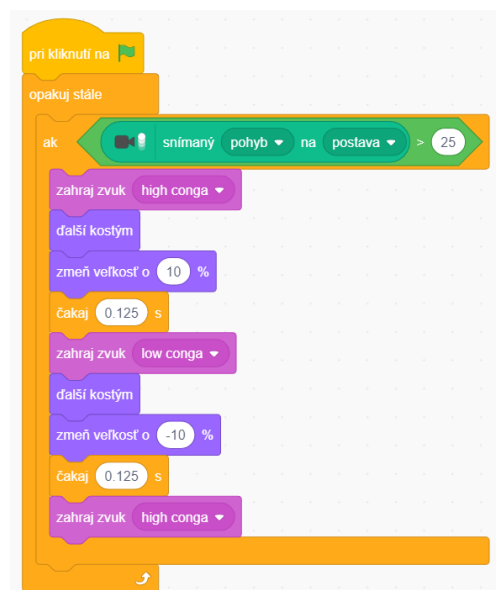
Počas tejto fázy učiteľ môže načrtnúť aj dekompozíciu úloh a zhrnúť jednotlivé položky projektov.

ROZPRACOVANIE – CCA 10 MIN.

Úloha 5	<p>Preskúmajte projekt orchester-doplnit.</p> <p>Doprogramujte ho tak, aby ste mohli snímaním pohybu rúk zahrať pripravené melódie pre saxofón a bicie.</p> <p>Navrhňte aj ďalšie vylepšenia, napr. aby sa nástroje zväčšili a následne zmenšili a pod.</p>	
----------------	--	--

Možné riešenie úlohy vidíme na obrázku a je uvedené aj v projekte **orchester_riesenie**.

Ak ostáva časový priestor, nabádajte žiakov, aby vylepšili projekt ďalšími doplneniami, napr. aby sa nástroje zväčšili a následne zmenšili, pridaním ďalších postáv, môžu pridať tanečníkov, alebo hudobníkov a pod.



HODNOTENIE – CCA 5 MIN.

Zaškrtnutím zhodnoťte svoje vedomosti.	Viem	S pomocou viem	Neviem
Zapnúť snímanie, zastaviť snímanie.			
Nastaviť priehľadnosť snímania.			
Ovládať pohyb postavy pomocou pohybu pred			
Nastaviť citlivosť (senzitivitu) snímania.			
Meniť kostým postave pomocou pohybu pred			

Na evalváciu slúži rubrika, pomocou ktorej žiaci zaškrtnutím sami zhodnotia úroveň osvojenia si kľúčových vedomostí a zručností, ako aj splnenie cieľov hodiny.

6B POUŽITIE MIKROFÓNU

<i>Tematický celok / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov: <ul style="list-style-type: none"> • analýza problému, • jazyk na zápis riešenia, • postupnosť príkazov Reprezentácie a nástroje: <ul style="list-style-type: none"> • práca s grafikou 	ISCED 2 / 5. - 6. ročník
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> • základy práce v prostredí Scratch • riešenie algoritmických úloh bez počítača • ovládanie prostredia Scratch, orientácia v skupinách príkazov, ovládanie grafických efektov 	
Ciele	
<i>Žiakom osvojované vedomosti</i>	<i>Žiakom rozvíjané zručnosti a spôsobilosti</i>
Prostredie Scratch <ul style="list-style-type: none"> • pracovať s mikrofónom, • dokáže využiť nastavenia zo skupiny Zisťovanie, • v programe využiť ovládanie postáv pomocou hlasitosti snímania mikrofónom. 	Informatické myslenie <ul style="list-style-type: none"> • ALG3 vytvárať vlastné algoritmy riešiace problém/časti problému (postupnosti krokov na realizáciu nejakej činnosti vedúcej k cieľu) • (DEK1) lineárna dekompozícia – lineárne rozdeliť objekty/problémy/procesy na menšie časti tak, aby sa dali využiť pre dosiahnutie cieľa (detekcia prvkov/určenie objektov/správania sa v nejakej hre – pozadie/postava/udalosti) • (VZ01) rozpoznať časti objektu/problému, ktoré majú rovnaké/podobné vlastnosti/pravidlá správania sa • (VZ05) preniesť/použiť vzory/myšlienky/riešenia z jedného problému na druhý problému
Metodika rieši problém používania externých vstupov počítača na ovládanie programu. Prostredie Scratch umožňuje fyzické ovládanie programu, ktoré nie je bežné v klasických programovacích prostrediach. Mikrofón je nástroj na spúšťanie akcie v programe, slúži ako senzor na ovládanie programov. Ak používate online prostredie, v prehliadači je potrebné povolit používanie mikrofónu .	
Dominantné vyučovacie metódy a formy	
Bádateľská metóda (5E) Frontálna forma, práca vo dvojiciach	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> • mikrofón • lokálna inštalácia prostredia Scratch3 alebo prístup na internet s online editorom • pracovný list pre každého žiaka • pracovné súbory pre žiakov a pre učiteľa: sviecky, stromy, balony, farebna hudba 	
Diagnostika splnenia vzdelávacích cieľov	
Diagnostika pomocou sebahodnotiacej rubriky a na základe analýzy práce žiakov.	

ÚVOD

Počas tejto vyučovacej hodiny chceme žiakom predstaviť prostredie Scratch ako nástroj na **fyzické programovanie**, t.j. ukážeme, že externými vstupmi z mikrofónu vieme riadiť program.

Takto koncipované programy sú nástrojmi na uvedenie žiakov do ovládania programu reálnymi senzormi a podporujú ich kreativitu. V úvode hodiny žiaci pracujú s programom, ktorý ich má zaujať a vtiahnuť do programovania projektov ovládaných zvukom. Metodika s použitím mikrofónu je alternatívou metodiky **6A Použitie kamery**. Učiteľ si vyberie jednu z týchto dvoch metodík podľa technického vybavenia učebne. Ak má učiteľ k dispozícii aj kamery aj mikrofóny, môže zrealizovať obe vyučovacie hodiny, pretože metodiky obsahujú rôzne úlohy.

ZAPOJENIE – CCA 10 MIN.

Vyučovaciu hodinu začneme **motivačným rozhovorom** s otázkami ako: „Viete, že mikrofón môže slúžiť aj ako senzor na spúšťanie nejakých dejov? Napríklad pri odpočúvaní osôb (políciou, alebo tajnou službou), sa nahrávanie začne až vtedy, keď zazvoní telefón. Povedzte ďalšie príklady!“ Žiaci by mali spomenúť viacero možností, kedy sa mikrofón používa ako senzor.

V prvej úlohe majú žiaci preskúmať **hotový** projekt so sviečkami.

Úloha 1

Preskúmajte projekt **sviecky**.

Experimentujte s použitými príkazmi a meňte jednotlivé parametre, aby ste pochopili, ako fungujú jednotlivé príkazy.



Úloha je zameraná na preskúmanie príkazov, ktorými sa ovládajú sviečky. Ako sa menia kostýmy sviečok pri zmene hlasitosti? Žiaci majú skúšať intenzitu svojho hlasu. Ak budú dobré zvukové podmienky, tak sviečky budú pokojne „horieť“, ak budú žiaci kričať, alebo fúkať do mikrofónu počítača, sviečky budú „horieť“ silným plameňom. Táto úloha ukazuje žiakom ovládanie programu pomocou mikrofónu.

Mikrofón (aj kamera) sú štandardnými zariadeniami a sú bežne vstavané do notebookov. Ak však používate stolné počítače, je potrebné využiť externé zariadenia. Využitie týchto zariadení priamo v programovaní rozširuje žiakom obzor a pripravuje ich na pochopenie programovania pomocou **senzorov**.

SKÚMANIE – CCA 20 MIN.

Úloha 2

Otvorte si projekt **stromy**.

Z ponúknutých blokov **zostavte program** tak, aby sa na určitú hlasitosť stromy kývali. Podľa toho, ako hlasno zafúkate, by sa stromy mali ohýbať. Využite zmenu kostýmov pre postavu stromu.



Možné riešenia úlohy sú pre učiteľa uvedené aj v projekte **stromy_riesenie**:



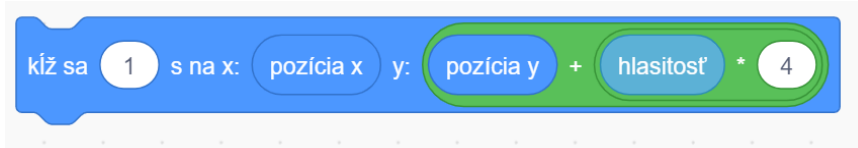
Žiaci by mali experimentovať s nastavením parametrov pre hlasitosť, časový interval medzi zmenou kostýmu, a pod.

Úloha 3

Otvorte projekt **balony**.

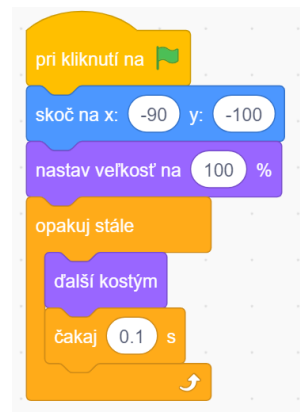
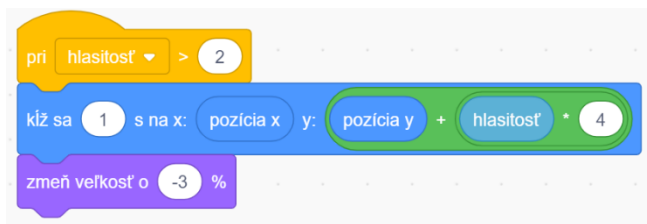
Doprogramujte pohyb balónov tak, aby sa pri zmene hlasitosti pohybovali smerom nahor. Napr. ak zafúkate, alebo zatlieskate.

Môžete ovládať balóny aj tak, že čím väčší zvuk vydáte, tým budú stúpať rýchlejšie. Napr. takto.



Doprogramujte aj to, aby sa počas letu balóny súčasne aj zmenšovali, ako keby sa vzdďaľovali.

Jedno z možných riešení vidíme na obrázku a je uvedené v projekte **balony_riesenie**.



Projekt je určený na opakovanie poznatkov zo zostavovania matematických výrazov s použitím aritmetických operácií. Žiaci majú skúšaním a experimentovaním zostaviť postupnosť príkazov tak, aby balóny stúpali smerom hore a súčasne sa aj zmenšovali. Podľa možnosti nechajte žiakov pracovať s projektom dlhšie, aby mali šancu nájsť riešenie, resp. vstupy tak, aby boli s výsledkom stúpania a zmenšovania spokojní. Podľa vlastných návrhov môžu vylepšiť projekt ešte aj ďalšími doplneniami.

VYSVETLENIE – CCA 5 MIN.

Táto fáza je dôležitá, aby sme zistili, či každý žiak správne pochopil úlohy a či ich vedel vyriešiť. Na základe svojich poznámok v pracovných listoch si žiaci vysvetlia, aké premenné používali – **hlasitosť** v skupine príkazov **Zisťovanie** (svetlomodré).

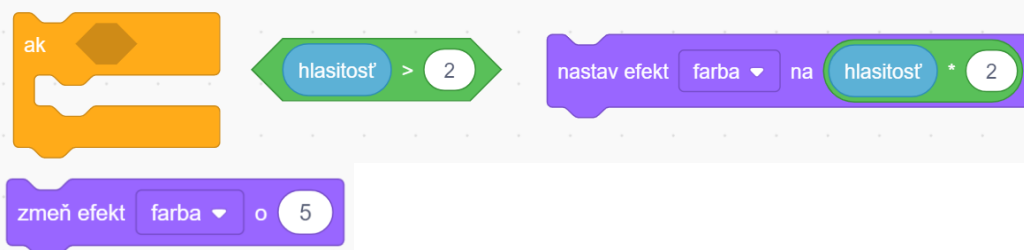
Je dôležité zdôrazniť, že existuje aj premenná hlasitosť v skupine **Zvuk** (fialové), ktorá obsahuje hodnotu hlasitosti prehrávaného zvuku danou postavou. Učiteľ by mal diskusiu iba moderovať tým spôsobom, že do nej nezasahuje, ale dbá na to, aby žiaci, ktorí majú program vyriešený, správne vysvetlili ostatným, ako postupovali.

ROZPRACOVANIE – CCA 10 MIN.

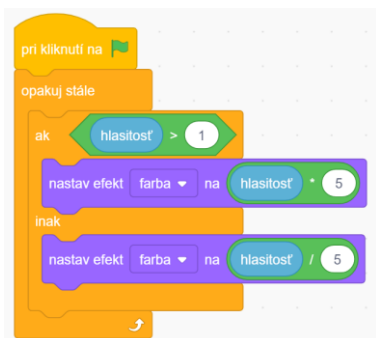
- Úloha 4** Otvorte projekt **farebna_hudba**.
Doprogramujte ho tak, aby sa podľa hlasitosti prehrávanej hudby menili na postavách svetiel farebné efekty.
Vyberte, kde je intenzita hlasitosti hudby nastavená:
- pri **Postave1**
 - pri každej postave
 - pri **Scéne**



Možné riešenie môžete vyskladať aj pomocou takých blokov:



Žiaci si majú zistiť, že intenzita hlasitosti nastavenej hudby sa nastavuje pre **Scénu**. Ďalšia senzorová hlasitosť, na ktorú sa svetelné efekty majú spúšťať, sa zisťuje pri postavách pomocou svetlomodrých blokov **Zisťovanie**. Možné riešenie úlohy vidíme na obrázku, učiteľ ho nájde v projekte **farebna_hudba_riesenie**.



Ak ostáva časový priestor, nabádajte žiakov, aby vylepšili projekt napr. ďalšími postavami, môžu pridať tanečníkov, alebo hudobníkov a pod.

VYHODNOTENIE – CCA 5 MIN.

Zaškrtnutím zhodnoťte svoje vedomosti.	Viem	S pomocou viem	Neviem
Spúšťať scenáre pomocou zvukov.			
Zistiť hlasitosť snímaného zvuku.			
Poznám rozdiel medzi hlasitosťou prehrávaného zvuku a hlasitosťou zvuku snímaného mikrofónom.			
Zmeniť pozíciu (súradnice) postavy v závislosti od			
Meniť farebné efekty v závislosti od hlasitosti.			

Na evalváciu slúži rubrika, v ktorej žiaci zaškrtnutím sami zhodnotia úroveň osvojenia kľúčových vedomostí a zručností, ako aj splnenie cieľov hodiny.

7 NOVÉ BLOKY

<i>Tematický celok / Téma</i>	<i>Odporúčaná ročník</i>
Algoritmické riešenie problémov: <ul style="list-style-type: none"> analýza problému, jazyk na zápis riešenia, postupnosť príkazov Reprezentácie a nástroje: <ul style="list-style-type: none"> práca s grafikou 	ZŠ / 5. - 6. ročník / 1 vyučovací hodina
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> základy práce v prostredí Scratch riešenie algoritmických úloh bez počítača ovládanie prostredia Scratch, orientácia v skupinách príkazov, ovládanie grafických efektov, ovládanie klávesnicou 	
Ciele	
Žiakom osvojované vedomosti	Žiakom rozvíjané zručnosti a spôsobilosti
Prostredie Scratch <ul style="list-style-type: none"> pochozí vytváranie nových blokov a ich používanie pri zostavovaní scenára, dokáže zmeniť farbu kostýmu postavy pomocou blokov zo skupiny Vzhľad, je schopný pridať rozšírenie Pero a využiť z neho blok opečiatkuj, pri vytváraní programu využíva cyklus. 	Informatické myslenie <ul style="list-style-type: none"> (ALG3) vytvárať vlastné algoritmy riešiace problém/časti problému (postupnosti krokov na realizáciu nejakej činnosti vedúcej k cieľu, (DEK1) lineárna dekompozícia – lineárne rozdeliť objekty/problémy/procesy na menšie časti tak, aby sa dali využiť pre dosiahnutie cieľa (VZO1) rozpoznať časti objektu/problému/procesu, ktoré majú rovnaké/podobné vlastnosti/pravidlá správania sa (VZO2) určiť rovnaké/podobné vlastnosti/pravidlá správania sa častí objektu
Dekompozícia a rozdelenie úlohy na menšie úlohy je náročná téma pre žiakov základnej školy. V metodike použité úlohy umožnia lepšie pochopiť a precvičiť rozpoznanie častí programu, ktoré majú rovnaké vlastnosti a pravidelne sa opakujú. Menšie podúlohy je vhodné riešiť pomocou vlastných príkazov – v Scratch sa dajú vytvárať v skupine Nové bloky .	
Dominantné vyučovacie metódy a formy	
Bádateľská metóda (5E). Frontálna forma. Práca vo dvojiciach.	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> lokálna inštalácia prostredia Scratch3 alebo prístup na internet s online editorom pracovný list pre každého žiaka projekty pre učiteľa a žiaka: veterník, mozaika, naramok, naramok2, dlazba 	
Diagnostika splnenia vzdelávacích cieľov	
Diagnostika pomocou sebahodnotiaceho testu a na základe analýzy práce žiakov.	

ÚVOD

Metodika sa venuje **cyklom** a vytváraniu a používaniu **vlastných blokov**. V časti SKÚMANIE žiaci pracujú s hotovým projektom **veterník**, v ktorom sú začlenené všetky programátorské jazy, ktoré budú žiaci na tejto vyučovacej hodine objavovať.

ZAPOJENIE - CCA 15 MIN.

Hodinu začneme motivačným rozhovorom, žiakom kladieme napr. nasledujúce otázky:

- Čo je podľa vás mozaika? Možné odpovede: Obrazy, motívy vyskladané z malých kameňkov, sklíčok, niekedy opakované geometrické tvary, kvietky a pod.
- Kde sa všade ste sa stretli s mozaikami? Predpokladané odpovede: na budovách, na stenách alebo podlahách.
- Ukážeme žiakom rôzne reálne mozaiky. Pre učiteľa ich uvádzame v prezentácii mozaiky.



1



2



3



4



5

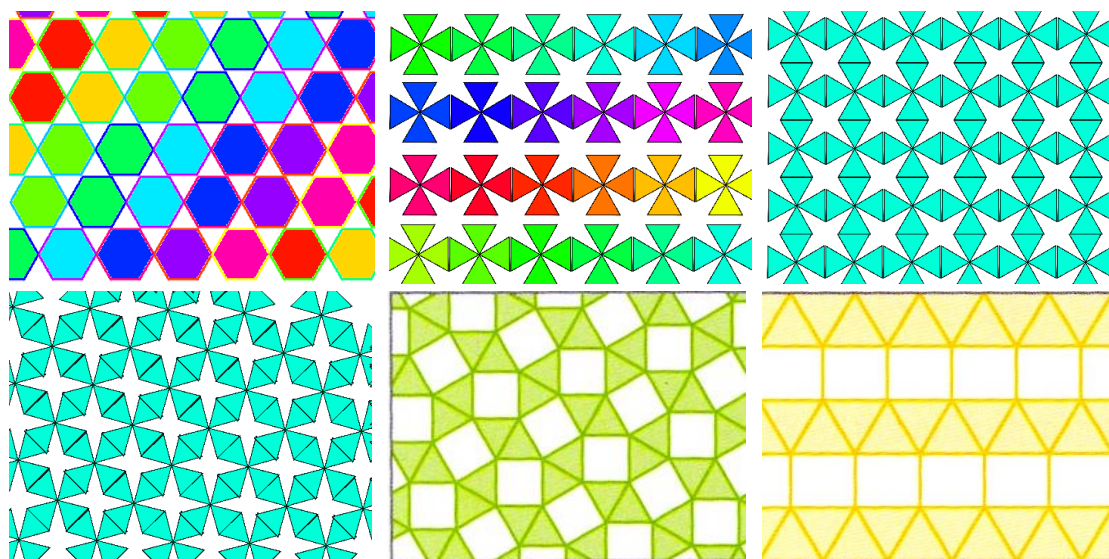
Poznámka

Mozaika (tal. Mosaico) je plošná výzdoba, ornamentálna alebo figurálna, zostavená z drobných kociek, hranolčekov alebo kolíčkov, kamenných, neskôr sklenených, upevňovaných do mäkkej malty alebo tmelu. Mozaika je dvojdimenzionálny obraz, kde ukladaním kúskov ku sebe, sa má vyplniť čo najväčšia plocha bez medzier. Holandský umelec M.C. Escher v roku 1922 navštívil palác Alhambra v španielskej Granade. Maurské ornamente tvoriace mozaiky na stenách a dlažbe ho nadchli natoľko, že sa začal zaujímať o to, ako pravidelne rozdeliť rovinu, pozri <https://sk.wikipedia.org/wiki/Mozaika>

- Môžeme použiť programovanie na vykresľovanie pravidelných vzorov? Napríklad na mozaiku v kúpeľni alebo na podlahe.

Úloha 1

Pozorujte jednotlivé mozaiky a vyznačte v nich opakujúce sa vzory.



Žiaci by mali byť schopní nájsť opakujúce sa vzory na obrázkoch. Úlohu môžu žiaci riešiť aj spoločne tak, že učiteľ premietne obrázky cez dataprojektor. Žiakov povzbudzujeme otázkami Čo sa opakuje? Biele, alebo farebné vzory? Zároveň sa ich snažíme viesť k tomu, aby našli čo najmenšiu časť, napr. trojuholník, z ktorého sa vyskladá veterník a pod. Je viac dobrých riešení, stačí, ak žiaci naznačia svoje nápady. Môžu si všimnúť, že tretí a štvrtý obrázok je ten istý vzor, len otočený.

Táto úloha rozvíja informatické myslenie, konkrétne hľadanie vzorov a pripravuje žiaka na nakreslenie, resp. použitie, takej postavy, aby pomocou jej **pečiatkovanja** vznikla podobná mozaika.

SKÚMANIE – CCA 10 MIN.

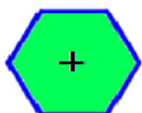
V tejto fáze uvedieme žiakom cieľ hodiny: vytvárať vlastné bloky a používať cyklus na opakovanie rovnakej činnosti.

Úloha 2 Preskúmajte projekt **veterník**, vyhľadajte všetky programové bloky, s ktorými ste sa doteraz ešte nestretli.

Zapíšte názvy blokov a aj to, v ktorej kategórii blokov ste ich našli:

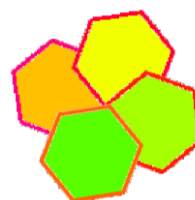
Kategória	Blok
	zmaž
Nové bloky	

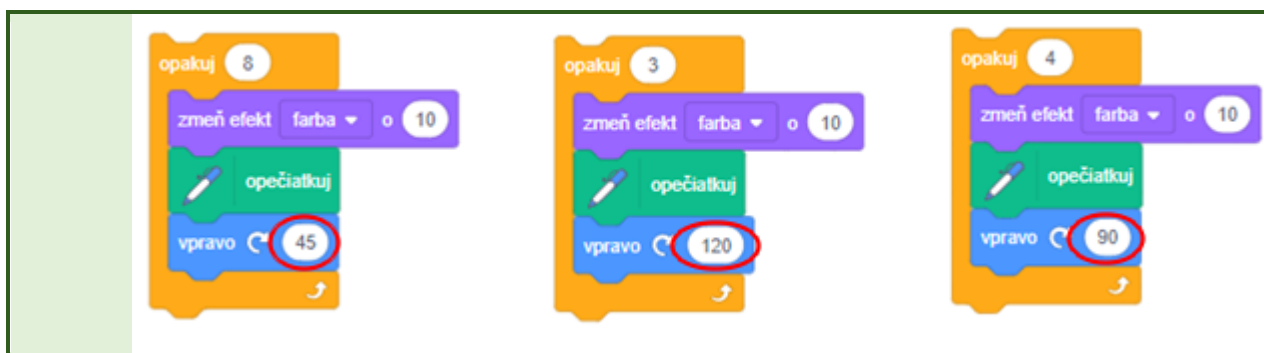
Úloha 3 Otvorte projekt **mozaika**, vyskúšajte príkazy, ktoré sú naprogramované pre jednotlivé postavy. Prejdite do záložky **Kostýmy** a zistite a do pracovného listu zakreslite bod, okolo ktorého sa otáča každý zo šesťuholníkov.



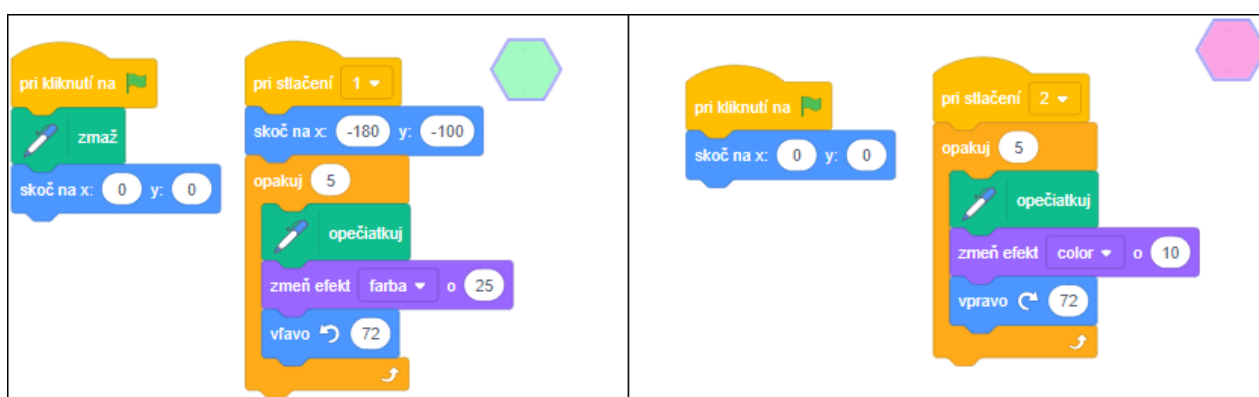
Žiaci zakreslia krížik do obrázkov tak, ako vidíme na obrázku.

Pozrite si obrázky a ku nim prislúchajúce programy. Uvažujte o tom, pre ktorú z dvoch postáv v projekte sú nasledujúce skupiny blokov. Doplňte chýbajúce údaje do blokov.

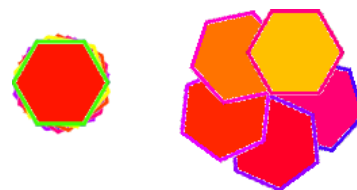




V projekte **mozaika** sú pripravené dve rovnaké postavy – šesťuholníky, ktoré majú pripravené aj podobné scenáre. Ich obrázky sa líšia v tom, **kde majú stred**, teda bod, okolo ktorého sa obrázok otáča pri využívaní základného bloku **vľavo** alebo **vpravo**. Na obrázku v riešení pre učiteľa sme chýbajúce čísla doplnili a zakružkovali.

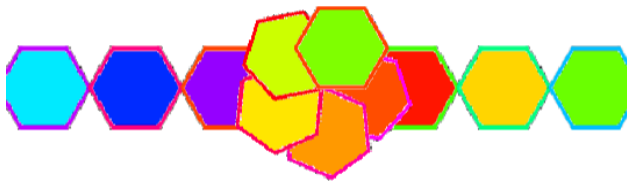


Žiaci by si mali prezrieť program a zistiť, ako sa spúšťajú bloky pre každý šesťuholník. Zrejme uvidia, že zelený šesťuholník reaguje na stlačenie **číslice 1**, ružový reaguje na stlačenie **číslice 2**. Ak si pozrú kostýmy zistia, že bod, okolo ktorého sa bude obrázok otáčať je v strede zeleného šesťuholníka, ale v ľavom rohu ružového. Takže hoci sú programy rovnaké, zelený pečiatkovaním vytvára obrázok, v ktorom vidíme len obrysy šesťuholníka, ale ružový vytvorí kvet, pozri obrázok:



Učiteľ si môže pripraviť aj na papieri nakreslený a vystrihnutý šesťuholník a priamo demonštrovať otočenie šesťuholníka okolo stredu a okolo jedného rohu, aby žiaci lepšie pochopili rozdiely.

Úloha 4 Doprogramujte projekt **náramok** tak, aby ste nakreslili náramok, ktorý sa skladá z reťaze farebných korálikov a z ozdobného kvetu tak, ako vidíte na obrázku.



Vytvorte vlastné bloky **retaz** a **kvet**.

Môžete si nakresliť aj vlastné koráliky. Pri ich kreslení vždy uvažujte, okolo ktorého bodu obrázka sa má korálik otáčať.

Úloha 4 je zameraná na **vlastnú tvorbu**, pričom žiaci využívajú poznatky z predchádzajúcich úloh. Možné riešenie úlohy nájde učiteľ v projekte **naramok2**.

Žiaci môžu nakresliť aj vlastné koráliky, z ktorých si vytvorí zázračný náramok.

VYSVETLENIE – CCA 5 MIN.

V tejto fáze vyučovacej hodiny si spoločne skontrolujeme riešenie **úlohy 2** formou riadeného rozhovoru. Žiaci sa prvýkrát stretli s príkazom **opečiatkuj**. Mali by vedieť **sformulovať**, že to je príkaz, pomocou ktorého postava zanechá svoju kópiu na mieste, na ktorom sa práve nachádza. Aby boli obrázky graficky pútavejšie, v programe sme využili základný blok **zmeň efekt farba**. Tento blok zabezpečí, že postava si zmení farbu, farba sa vyberá v poradí farieb dúhy (oranžová, žltá, zelená, modrá, fialová, červená), pričom celá škála je rozdelená do 200 odtieňov. Ak dosiahneme 200, pokračuje opäť od 0. Teda, ak by sme dali zmenu o 200, stále by sme dostali tú istú farbu. Čím menšiu hodnotu zmeny dáme, tým bude zmena odtieňa menšia.

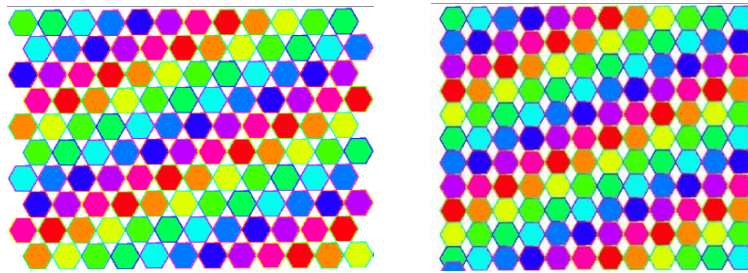
Ďalej by sme mali umožniť žiakom ukázať svoje vlastné zázračné náramky. Komentujte a vyžadujte, aby aj ich programy obsahovali nové bloky.

ROZPRACOVANIE – CCA 10 MIN.

Úloha 5 je zameraná na samostatnú tvorbu vlastného programu. Postava je daná v projekte, ale žiaci si môžu navrhnuť aj iný motív.

Možné riešenia úlohy nájde učiteľ v projektoch **dlazba2** a **dlazba3**.

Úloha 5 Naprogramujte návrh farebnej dlažby do svojej kúpeľne pomocou šesťuholníkových dlaždíc. Dlaždicu máte pripravenú v projekte **dlažba**. Vydláždenie môže vyzeráť napr. takto:



Navrhňte vlastný blok **riadok**.

HODNOTENIE – CCA 5 MIN.

Sebahodnotiaci test slúži na zopakovanie, ale aj zhrnutie získaných nových poznatkov.

Spojte čiarami jednotlivé **popisy ovládania** s príslušným **blokom príkazu**.

<p>Skupina blokov, v ktorej je volanie nového bloku.</p>	<p>Vlastný blok, ktorý opečiatkuje postavu pootočenú okolo bodu, ktorý je stredom obrázka.</p>	<p>Vlastný blok, ktorý vedľa seba 10-krát opečiatkuje postavu.</p>

8 NÁHODA: PEČIATKUJEME A PRESÚVAME

<i>Tematický celok / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov: <ul style="list-style-type: none"> • analýza problému • jazyk na zápis riešenia • postupnosť príkazov 	ISCED 2 / 5. - 6. ročník
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> • základy práce v prostredí Scratch • ovládanie prostredia Scratch, orientácia v skupinách príkazov, ovládanie grafických efektov, ovládanie klávesnicou 	
Ciele	
<i>Žiakom osvojované vedomosti</i>	<i>Žiakom rozvíjané zručnosti a spôsobilosti</i>
Prostredie Scratch: <ul style="list-style-type: none"> • pripraví scenár, ktorý generuje náhodné čísla z nejakého intervalu, • navrhne scenár, ktorý presúva postavu na náhodné súradnice, • vytvorí vlastný program, v ktorom využije pečiatkovanie kostýmu postavy, • navrhuje a používa vlastné bloky. 	Informatické myslenie <ul style="list-style-type: none"> • (LOG2) využitím logických zdôvodnení predpokladať správanie sa jednoduchých programov • (ALG3) vytvárať vlastné algoritmy riešiace problém/časti problému (postupnosti krokov na realizáciu nejakej činnosti vedúcej k cieľu) • (VZO1) rozpoznať časti procesu, ktoré majú podobné vlastnosti/pravidlá správania sa (premenná v cykle, člen postupnosti) • (VZO5) použiť vzory z jedného problému na druhý (analógia, použitie cyklov, podprogramov, modulov, skrátenie zápisu dát obsahujúcich vzory, nastaviť štartovací stav, nechať bežať podľa pravidiel)
Metodika rieši problém, ako sa v prostredí generujú náhodné čísla a ako sa používajú napr. na náhodné umiestnenie postavy na scéne. Správne sa orientovať v dvojrozmernom priestore – súradniciach scény, v ktorých sa používajú aj kladné a aj záporné súradnice môže predstavovať pre žiakov problém. V metodike uvedené projekty a úlohy im pomôžu pochopiť a precvičiť využívanie súradníc. Súčasťou vyučovacej hodiny je aj precvičovanie vytvárania a používania nových blokov.	
Dominantné vyučovacie metódy a formy	
Projektová metóda. Frontálna forma. Práca vo dvojiciach.	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> • lokálna inštalácia prostredia Scratch3 alebo prístup na internet s online editorom • pracovný list pre každého žiaka • pracovné súbory pre učiteľa a žiaka: hviezdy, kvety, kacicky, kacka a pes 	
Diagnostika splnenia vzdelávacích cieľov	
Diagnostika pomocou sebahodnotiaceho testu vo fáze hodnotenia a na základe analýzy práce žiakov.	

ÚVOD

Táto vyučovací hodina sa venuje **cyklom** a **náhodným číslam**, rozpracúva aj problematiku nových blokov.

ZAPOJENIE – CCA 15 MIN.

Hodinu začneme **motivačným rozhovorom**, uvedieme cieľ hodiny: používanie náhody v programoch.

Otázky pre žiakov:

- V prírode sú často predmety umiestnené nepravidelne – porozhadzovane. Poznáte nejaké?
Možné odpovede žiakov: kvety na lúke, kamene na poli, hviezdy na oblohe, vysypané kocky z krabice a pod.
- Ako je v programovaní takéto nepravidelné umiestnenie urobené?
Očakávame odpoveď, ktorú by mohli niektorí žiaci poznať: Pomocou náhodného čísla.

Úloha 1 Preskúmajte projekt **hviezdy**.

Otvorte na učiteľskom počítači projekt **hviezdy**, spustite ho. Cieľom hry je nájsť postavu – hviezdu.

V projekte sa hviezda 10 krát presunie na náhodné miesto a urobí pečiatku. Úlohou hráča je kliknúť na hviezdu, teda ju nájsť medzi ostatnými opečiatkovanými kópiami. Ak hráč trafi postavu hviezdy, tak sa zmaže celá plocha. Vyzvite žiakov, aby sa s projektom zahrali na svojich počítačoch.

SKÚMANIE – CCA 20 MIN.

Nechajte žiakov, aby preskúmali projekt **hviezdy** a vypracovali **úlohy 2 až 5**. V projekte nie je naprogramované spúšťanie pohybu hviezdy pomocou premennej pohyb. Je vytvorená premenná, v scenári sú príslušné bloky, ale žiaci majú v **úlohe 4** sami určiť poradie blokov tak, aby sa hviezda pohybovala o toľko krokov dopredu, akú hodnotu má premenná pohyb. Učiteľ má k dispozícii vyriešený program v projekte **hviezda_riesenie**.

Žiaci môžu pracovať vo dvojiciach, avšak pracovný list by mal mať každý žiak vlastný.

Udalosti, ktoré by mali žiaci zakrúžkovať vidíme na obrázku.

Úloha 2

Zakrúžkujte, ktoré udalosti sa používajú v projekte.



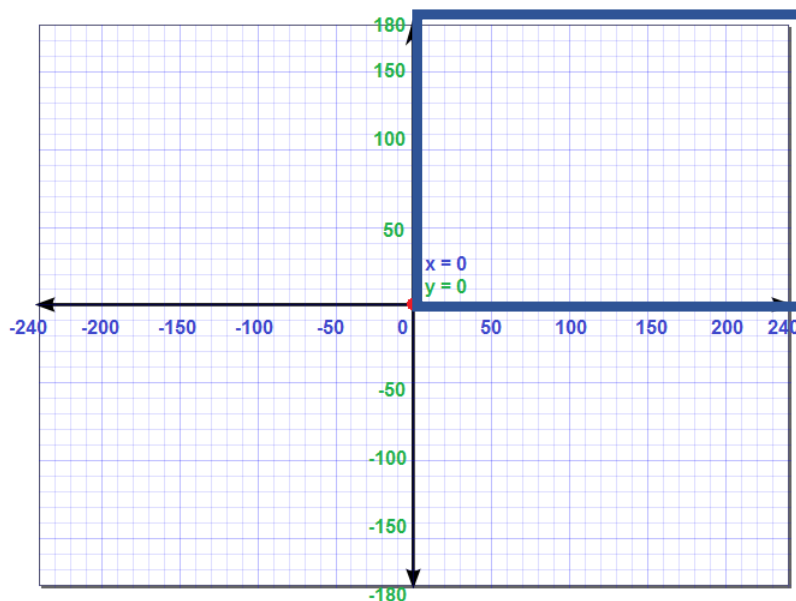
Úloha 3

Preskúmajte blok náhodne. Navrhnite, aké hodnoty má blok náhodne generovať, pre blok **súradnice x, y** ale tak, aby sa generovali hviezdy **len v hornej pravej** časti scény.

Dopíšte do obrázka!



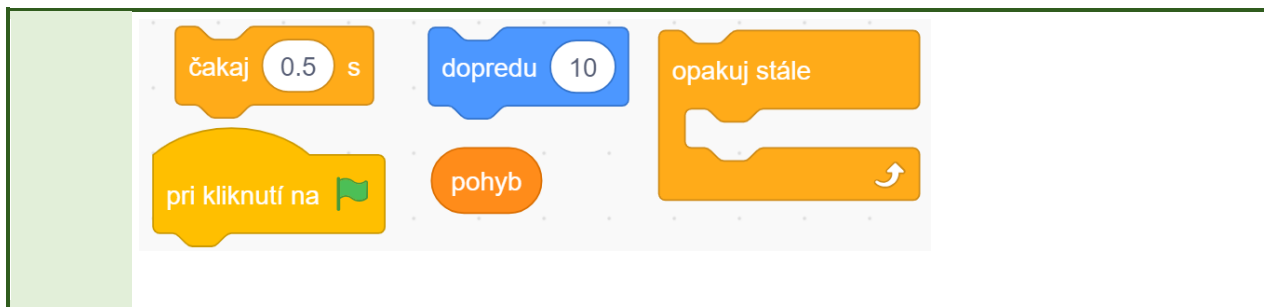
Označte časť scény, na ktorej sa objavili hviezdy.



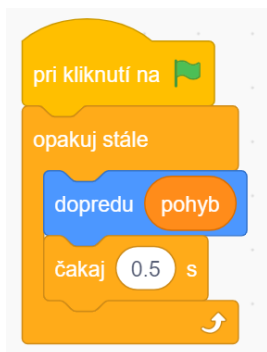
Na obrázku vidíme riešenie úlohy, keď majú žiaci označiť pravú hornú štvrtinu Scény.

Úloha 4

Doprogramujte projekt tak, aby sa pri zmene hodnoty premennej **pohyb** postava hviezdy pohybovala dopredu o príslušný počet krokov. Usporiadajte príslušné bloky.



Riešenie úlohy vidíme na obrázku:

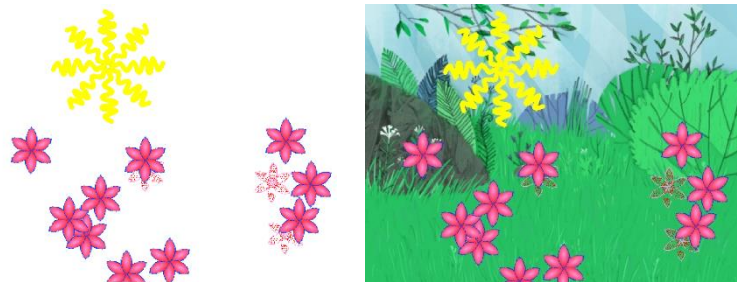


Žiaci v tejto úlohe objavujú, aký má byť krok hviezdy v bloku dopredu, aby sa hviezda pohybovala podľa hodnoty premennej **pohyb**, ktorá je zobrazená ako posúvač.

Úloha 5 Otvorte si projekt **kvety**.


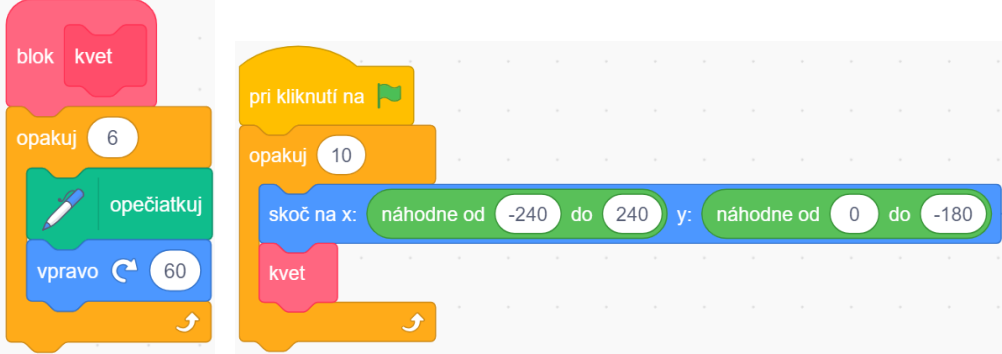

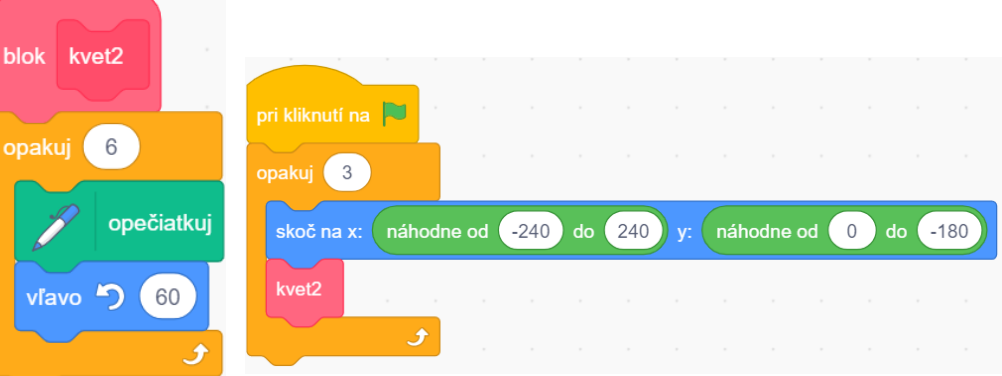


Nachádzajú sa v ňom 3 postavy – 2 lupene a 1 lúč.

- Pre postavu **lupeň1** je vytvorený nový blok **kvet**. Doprogramujte projekt tak, aby sa vykresľoval kvet z lupeňov pomocou bloku kvet na 10 náhodných miestach v dolnej časti scény.
- Pre postavu **lupeň2** doprogramujte nový blok **kvet2**. Tento nový blok použite v programe, ktorý bude vykresľovať na náhodných miestach 3-krát kvet, ale iba v dolnej časti scény.
- Pre postavu **luc** vytvorte nový blok **slnko** a použite ho v programe tak, aby sa vykreslilo slnko na náhodnom mieste v hornej časti scény.



Môžete do svojho projektu pridať aj pozadie, alebo si nakresliť vlastné pozadie.

Riešenie

 <p>lupeň1</p>	
 <p>lupeň2</p>	
 <p>lúč</p>	

VYSVETLENIE – CCA 5 MIN.

Na základe svojich poznámok v pracovných listoch si žiaci počas nasledujúcej diskusie vysvetlia, čo zistili vo dvojiciach o tom, ako fungovali jednotlivé bloky. Tu je vhodné, aby učiteľ zobrazil na tabuli **súradnicovú sústavu s hodnotami x a y** (ďalšie pozadie scény v projekte **kvety**) a aby žiaci vysvetľovali a **vykresľovali**, aké majú byť **parametre** pre blok **náhodne**, ktorý generuje pozície x a y. Učivo nevysvetľujeme, len usmerňujeme žiakov pri ich vysvetľovaní. V prípade nesprávnej, alebo nie úplne jasnej formulácii niektorého žiaka, môžu úlohu vysvetliť ďalší žiaci. Do fázy vysvetľovania by sa malo zapojiť čo najviac žiakov.

Učiteľ by mal dbať na to, aby sa počas vysvetľovania objavili nasledujúce informácie a aby väčšina žiakov pochopila hodnoty pre x-ovú a y-ové súradnice v rôznych častiach scény:

- ak chceme, aby sa bola súradnica **v ľavej časti** scény, tak hodnoty pre blok **náhodne** v časti pre generovanie **x-ovej** súradnice majú byť z intervalu od -220 do 0. Ak berieme do úvahy šírku objektu, tak pravú hodnotu treba zmenšiť o šírku obrázka objektu. Ak žiaci veľkosť obrázka nevedia presne, stačí túto veľkosť odhadnúť.
- ak chceme, aby sa bola súradnica **v pravej časti** scény, tak hodnoty pre blok **náhodne** majú byť pre **x-ovú** súradnicu z intervalu 0+šírka postavy do 220
- ak chceme, aby bola súradnica v **hornej** polovici scény, tak hodnoty pre blok **náhodne** pre **y-ovú** súradnicu majú byť z intervalu od 0 + výška postavy do 180
- a ak sa má blok náhodne vygenerovať súradnicu v **dolnej** časti scény, tak hodnoty pre **y-ovú** súradnicu majú byť od -180 do 0-výška postavy

Spôsob volania **nových blokov** žiaci už videli na predchádzajúcej hodine, ale pravdepodobne bude potrebné zopakovať, čo znamená volanie bloky v inom bloku. Môžeme to urobiť tým, že vysvetľovať bude žiak, o ktorom vieme, že má správne vyriešenú **úlohu 5**, v ktorej vytvoril nové bloky a aj ich **vhodne pomenoval**.

Poznámka

Vieme, že je pomerne náročné **systematicky** viesť žiakov k tomu, aby si bloky **pomenovali** vhodnými menami. Je však veľmi dôležité, aby ich k tomu učiteľ viedol a upozorňoval ich, ak majú názov bloku, ktorý nenesie žiadnu informáciu o tom, čo blok robí, napr. DDD, NIECO... Treba si uvedomiť, že žiakom správne pomenovávanie vlastných blokov **výrazne pomôže** pri vývoji náročnejších projektov.

ROZPRACOVANIE – CCA 10 MIN.

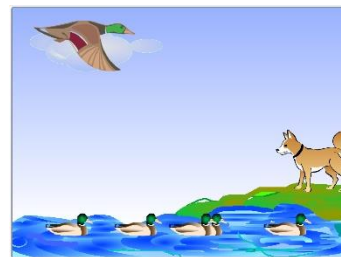
V tejto časti vyučovacej hodiny žiaci pracujú s dvoma programami – **kacicky** a **kacka_a_pes**. Pre učiteľa sme pripravili riešenia oboch programov, ktoré nájde v rovnako pomenovaných projektoch v učiteľskom priečinku. Učiteľ si môže pred hodinou tieto vzorové riešenia pozrieť. Ak je to potrebné, môže ich počas tejto časti hodiny ukázať žiakom, ale len v **celostránkovom** režime, aby žiaci nevideli bloky, ktoré majú sami vytvoriť.

Úloha 6


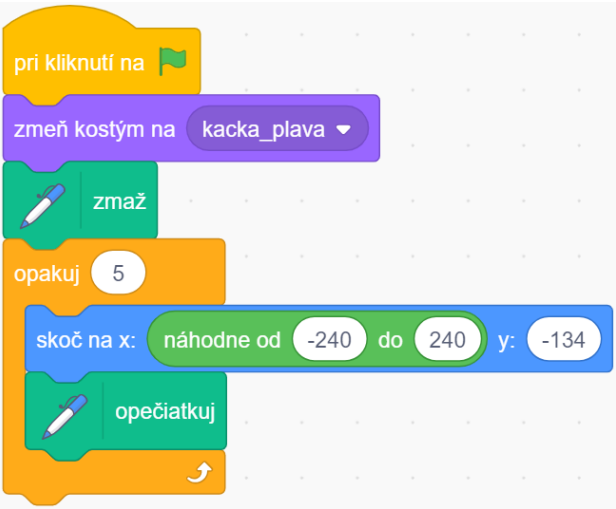


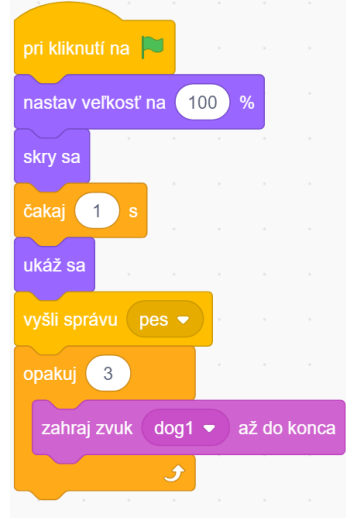
Otvorte si pripravený projekt **kacicky**.

Naprogramujte pohyb postáv podľa nasledujúceho scenára:

- na náhodných miestach v dolnej časti scény (na vode) opečiatkujte 5-krát kačku
- nech sa postava pes objaví 2 sekundy po začiatku programu a zabreše
- po zabrechaní psa kačka odletí na náhodné miesto na oblohe




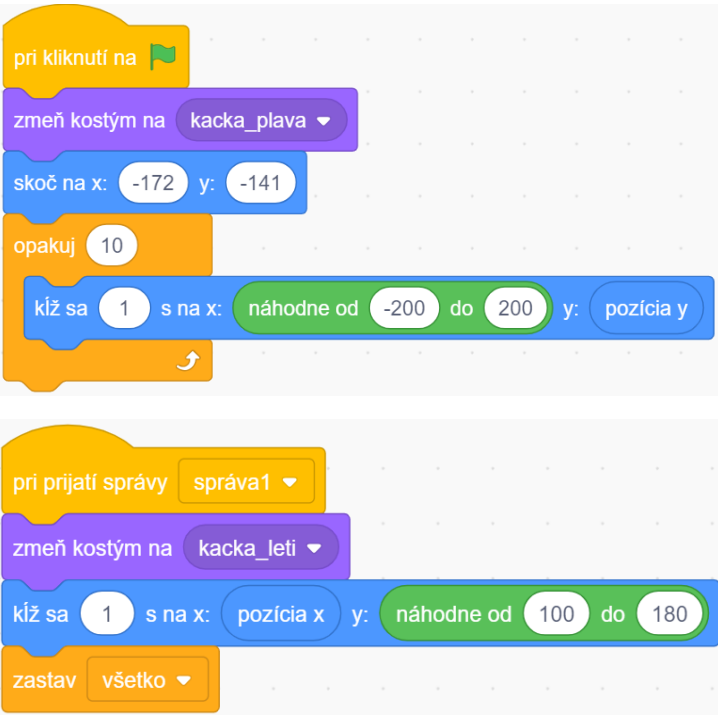

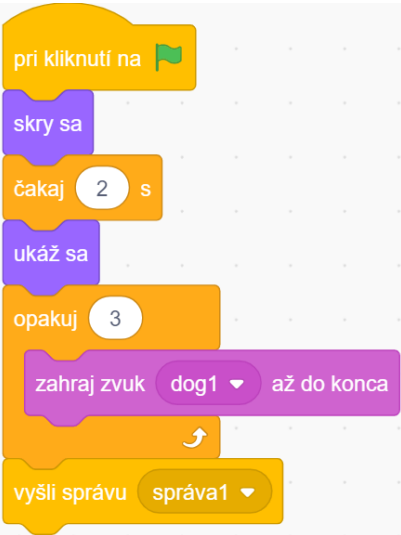
Možné riešenie:

 kacka	
	
 pes	

Úloha 7 Otvorte si pripravený projekt **kacicky_a_pes**.

Doprogramujte ho tak, aby náhodne po jazierku **plávala** len 1 kačička. Keď sa zjaví pes, vyleť hore nad jazierko.



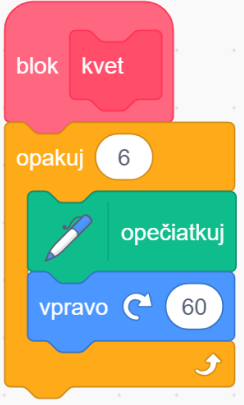
Možné riešenie:

Odporúčame, aby vytvorené projekty žiaci odovzdali učiteľovi. Niektoré z nich môže učiteľ ukázať ostatným žiakom a ohodnotiť známku. Projekty je tiež možné zverejniť v **Scratch** štúdiu, ktoré učiteľ vytvorí pre svojich žiakov.

HODNOTENIE – CCA 5 MIN.

Na evalváciu slúži krátky test, v ktorom žiaci odpovedajú na pripravené otázky.

Vyberte správny popis ovládania pre príslušné bloky príkazu.	
	
Postava skočí náhodne do	
<ul style="list-style-type: none">• ľavej časti scény• pravej časti scény• hornej časti scény	
	
Postava, ktorá vykonáva tieto bloky a prijme správu:	
<ul style="list-style-type: none">• zmení svoju farbu a premiestni sa na náhodnú pozíciu• zmení svoj kostým a bude kľžať na náhodnú pozíciu• zmení svoj kostým a bude kľžať na pozíciu v hornej časti scény	
	<p>Vytvorili sme nový blok kvet.</p> <p>Čo robí?</p> <ul style="list-style-type: none">• pomocou postavy nakreslí kvet tak, že skočí na náhodné miesto a opečiatkuje sa• niekde na scéne opečiatkuje 6 kvetov• pomocou postavy nakreslí kvet tak, že sa 6-krát opakovane opečiatkuje a otočí sa nejaký uhol

9 SÚRADNICE A VSTUPY

<i>Tematický celok / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov: <ul style="list-style-type: none"> analýza problému, jazyk na zápis riešenia, postupnosť príkazov Reprezentácie a nástroje: <ul style="list-style-type: none"> práca s grafikou 	ISCED 2 / 5. - 6. ročník
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> základy práce v prostredí Scratch riešenie algoritmických úloh bez počítača ovládanie prostredia Scratch, orientácia v skupinách príkazov, ovládanie grafických efektov, ovládanie klávesnicou 	
Ciele	
Žiakom osvojované vedomosti	Žiakom rozvíjané zručnosti a spôsobilosti
Prostredie Scratch: <ul style="list-style-type: none"> pochozí, ako je navrhnutá súradnicová sústava v prostredí, využíva premenné – zobrazuje premenné na scéne, mení hodnotu premennej, pripraví scenár, ktorý zabezpečí ovládanie postavy klávesmi - šípkami. 	Informatické myslenie <ul style="list-style-type: none"> (ALG3) vytvárať vlastné algoritmy riešiace problém (postupnosti krokov na realizáciu nejakej činnosti vedúcej k cieľu, vytvárať scenáre a storyboardy) (DEK2) hierarchická dekompozícia – hierarchicky rozdeliť objekty /procesy na menšie časti tak, aby sa dali využiť pre dosiahnutie cieľa (podproblémy rozdeliť na menšie podproblémy) (VZO1) rozpoznať časti procesu, ktoré majú podobné vlastnosti (člen postupnosti, časti programu) (ALG3) vytvárať vlastné algoritmy riešiace problém (postupnosti krokov na realizáciu nejakej činnosti vedúcej k cieľu, vytvárať scenáre)
Programovanie väčších projektov si okrem dekompozície vyžaduje aj následnú syntézu a schopnosť žiakov pochopiť spoluprácu a súčinnosť jednotlivých častí programu. Postupom, ktorý majú zobrazený v pracovnom liste, a ktorý budú podľa tohto pracovného listu realizovať, privedie žiakov k naprogramovaniu svojej vlastnej hry.	
Dominantné vyučovacie metódy a formy	
Frontálna forma, práca vo dvojiciach.	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> lokálna inštalácia prostredia Scratch3 alebo prístup na internet s online editorom pracovný list pre každého žiaka pracovné súbory: Choppy Knight (https://scratch.mit.edu/projects/353873212/), Lov na syr 	
Diagnostika splnenia vzdelávacích cieľov	
Preskúšanie vytvorených projektov, žiaci si zahrajú svoje hry.	

ÚVOD

Metodika sa venuje súradniciam a ovládaniu postavy pomocou tlačidiel. V úvode žiaci pracujú s hotovým projektom, hrou **Choppy Knight**. V tejto hre sa pohybuje postava, ktorá zbiera mince (poklad) a má sa pritom vyhýbať prekážkam. Následne majú žiaci **naplánovať** svoju vlastnú hru a na základe úloh v pracovnom liste vytvoria svoj vlastný projekt.

SKÚMANIE – CCA 10 MIN.

V úvode hodiny vyzveme žiakov, aby si zahrali počítačovú hru **Choppy Knight**, ktorá je vytvorená v prostredí **Scratch**, pozri <https://scratch.mit.edu/projects/353873212/>

Úloha 1 Zahrajte hru **Choppy Knight**.

Zapíšte, koľko postáv je v projekte.

Kde je definované pozadie jednotlivých úrovní hry?

Ktorá postava reaguje na zelenú zástavku?

14 postáv

postava LEVEL obsahuje 12 kostýmov

po stlačení zelenej zástavky sa ukáže postava TB



V hre sa pohybuje postava, ktorá má zbierať mince (poklad) a vyhýbať sa prekážkam, ktoré sú zobrazené ako ostré hroty. Ak postava spadne na prekážku, vráti sa na začiatok aktuálnej úrovne. Pomocou stlačenia medzery vie odhodiť zbraň, aby zneškodnil strážcov. Postavička sa ovláda šípkami hore, vpravo, vľavo (alebo aj klávesmi a, d, w). Keď sa žiaci zoznámia s hrou, spoločne urobte rozbor projektu podľa otázok v pracovnom liste.

Zistenia, ktoré by mali urobiť žiaci vidíme zapísané **modrou farbou**. Pri skúmaní projektu by si mali žiaci všimnúť, ako sú realizované jednotlivé úrovne. Mohli by sa tiež zaujímať o to, akým spôsobom postava „padá“, aby bol jej pohyb realistický.

REALIZÁCIA - CCA 30 MIN.

Vyzvite žiakov, aby naplánovali a vytvorili svoju vlastnú hru s témou **Lov na syr**.

Žiaci môžu pracovať aj vo dvojiciach a spoločne tak vytvárať svoj projekt. Táto metodika je **odlišná** od predchádzajúcich a nie je navrhnutá formou 5E. Žiaci v nej budú na celej hodine vytvárať jeden dlhší projekt. V pracovnom liste uvádzame návod na riešenie jednotlivých častí projektu. Úlohou žiakov je tieto kroky zreplikovať, prípadne v niektorých častiach pracovného listu ich dotvoriť podľa vlastného námetu.

Úloha 2 Vytvorte svoju vlastnú hru s témou **Lov na syr**, v ktorom sa nachádza:

- myška, ktorú má hráč ovládať pomocou šípok a doviešť ju k syru, ktorý sa objaví na náhodnom mieste
- v labyrinte sa budú pohybovať aj chrobáčky, keď sa s nimi myška stretne, hra sa ukončí.

V pracovnom liste je navrhnuté, ako si postupne **rozvrhnúť programovanie** projektu a tiež jeden z možných postupov riešenia. Ak majú žiaci iné nápady, nechajte ich pracovať podľa vlastného plánu.

Na začiatku programovania projektu je vhodné zaradiť diskusiu o tom, ako si vlastný projekt rozvrhnúť, aké postavy bude obsahovať a tiež aké bloky zrejme použijeme pri jeho realizácii.

- Postavy: myška, labyrint, syr, chrobáčky.
- Scenáre: myška ovládaná z klávesnice, náhodné objavenie sa syra, náhodný pohyb chrobákov na scéne.

Navrhnutý postup riešenia:

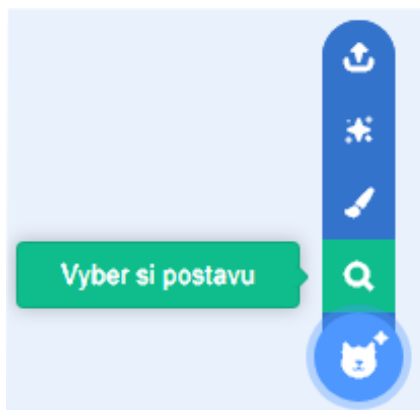
1. Ovládanie postavy myšky pomocou šípok.
2. Príprava hracej plochy – labyrintu.
3. Pohyb myšky medzi stenami labyrintu.
4. Syr a jeho náhodné objavenie sa.
5. Vytvorenie chrobákov a ich pohyb.

Vytvorte hru **Lov na syr**, v ktorej:

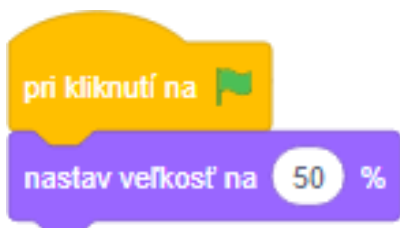
- bude postava myšky, ktorú budeme ovládať pomocou klávesov – šípok
- bude labyrint, myška nemôže prejsť cez stenu
- na náhodnom mieste scény sa objaví syr
- hra skončí, keď sa myška dostane k syru

1. Ovládanie postavy myšky pomocou šípok

Vytvorte nový projekt, zrušte postavu mačky a vytvorte novú postavu **Mouse1** výberom z ponuky postáv Scratch



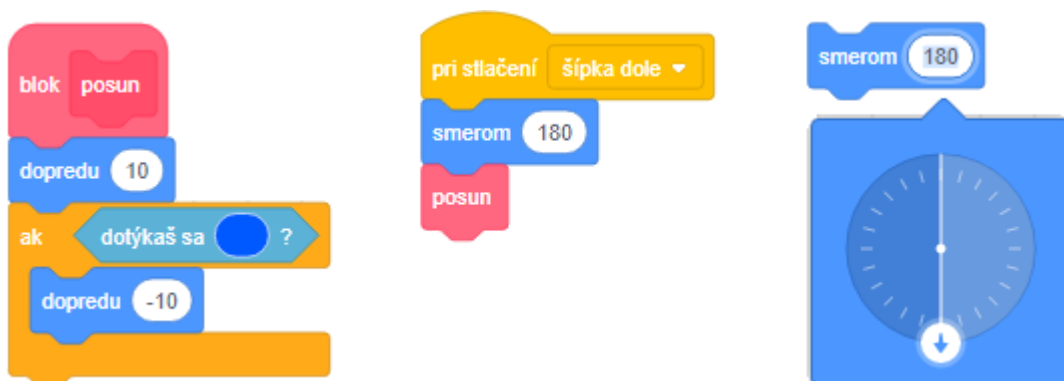
Pri štarte programu môžete zmenšiť veľkosť myšky:



Najprv žiaci majú vybrať postavu, ktorú bude hráč ovládať pomocou šípok. Možno si všimnú, že myška z galérie má dlhší chvost, ktorý by mohol prekážať pri pohybe myšky v labyrinte. Je dobré potom ho v editore skrátiť.

Pomocou známeho rozhodovacieho bloku vieme naprogramovať ovládanie šípkami. Smer natočenia postavy sa nastaví pomocou grafického uhlomeru.

Pre každý smer pohybu postupne naprogramujte ovládanie myšky pomocou klávesov – šípok. Vytvorte nový blok posun, v ktorom sa bude myška posúvať o nejaký krok. Na obrázku vidíte, bloky, ktoré zabezpečia pohyb myšky smerom dolu (smerom 180).



Podobne doprogramujte aj pohyb pre ostatné klávesy – šípky hore, vpravo a vľavo.

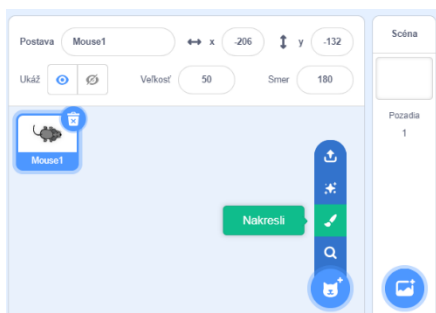
Nezabudnite pri stlačení ďalších šípok využiť už hotový blok **posun**.

V návode je postup pre naprogramovanie šípky dolu. Žiaci majú doprogramovať ovládanie pre ďalšie šípky. Zrejme zmenia udalosť pri stlačení, v ktorej vyberú inú šípku a následne zmenia smer, ktorým sa má myška natočiť.

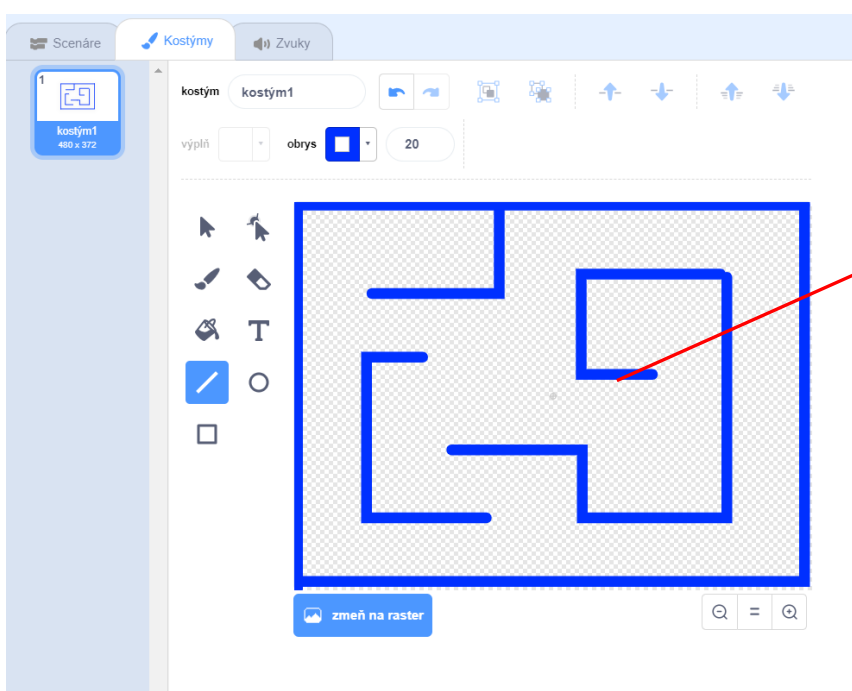
Žiaci by mali pripraviť takéto udalosti:



2. Príprava hracej plochy: nakreslite postavu, ktorá bude mať obrázok labyrintu. Otvorte kresliaci editor:



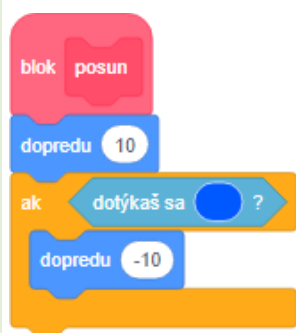
Nakreslite modré čiary labyrintu:



ak kreslíte so stlačeným klávesom Shift, budú sa kresliť rovné čiary

Labyrint navrhujeme nakresliť ako novú postavu. Na kreslenie využijeme kresliaci editor, ktorý je súčasťou prostredia Scratch. Takýto postup je výhodný aj preto, že ak budú chcieť žiaci vytvárať ďalšie úrovne hry, stačí ak nakreslia postave nový kostým.

3. Doplňte podmienku pre pohyb myšky tak, aby myška neprešla cez stenu labyrintu:



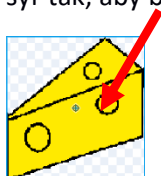
Všimnite si, že podmienku sme doplnili do bloku **posun**. Keďže sa tento blok využíva pri pohybe **pre všetky šípky**, myška reaguje na stenu vždy.

V tejto časti doprogramujú žiaci pohyb myšky tak, aby pri náraze na stenu labyrintu zacúvala, čo zabezpečí, že myška neprechádza cez stenu labyrintu.

Najčastejšia chyba, ktorá sa žiakom stáva je, že ak sa myška otočí príliš blízko pri stene jej chvost sa bude dotýkať steny, a tak myška zostane stáť. Ak sa to žiakom stane, navrhneme im, aby myške skrátili chvost.

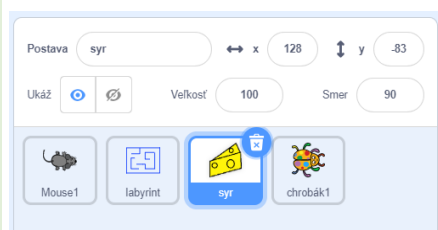
4. Syr, jeho náhodné objavenie sa, presun pri dotyku s myškou a zmena skóre.

Vytvorte novú postavu, nakreslite jej obrázok – syr. Pri kreslení dávajte pozor, aby ste jej stred umiestnili do stredu kresliacej plochy. V kresliacom editore si všimni terčik a umiestni syr tak, aby bol v strede syra.



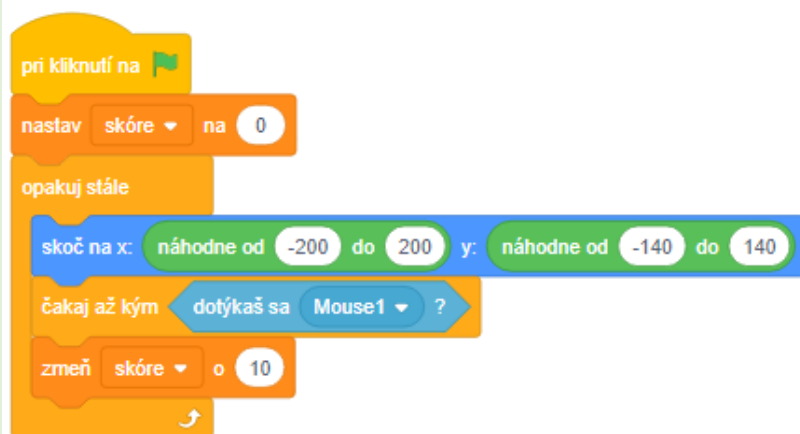
Naprogramuj, aby sa postava **syr** náhodne objavila na scéne.

Poloha postavy na scéne je určená pomocou súradníc. Aktuálnu polohu postavy vidíme v informáciách o postave. Preneste syr na ľavý okraj a pozrite si hodnoty pre x a y.



Náhodné umiestnenie syra naprogramujeme pri stlačení zelenej zástavky a čakanie na to, kým sa ho dotkne myška. Aby sme mohli počítať skóre, vytvorte premennú **skóre** a doprogramujte jej zmenu pri stretnutí myšky a syra.

Celý program pre syr môže vyzerat' napríklad takto:



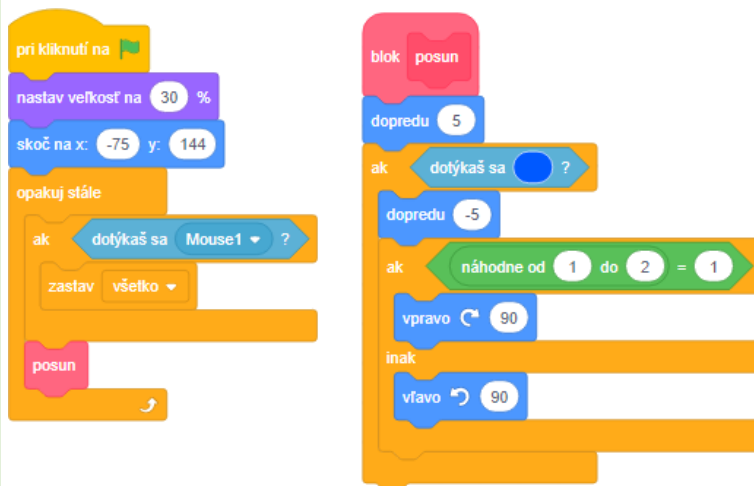
Syr si žiaci nakreslia v kresliacom editore prostredia Scratch. Treba ich upozorniť, aby jeho veľkosť zmenili tak, aby sa syr zmestil medzi steny labyrintu.

5. Invázia chrobákov

Vyberte postavu chrobáčka a umiestnite ho na konkrétne miesto scény.

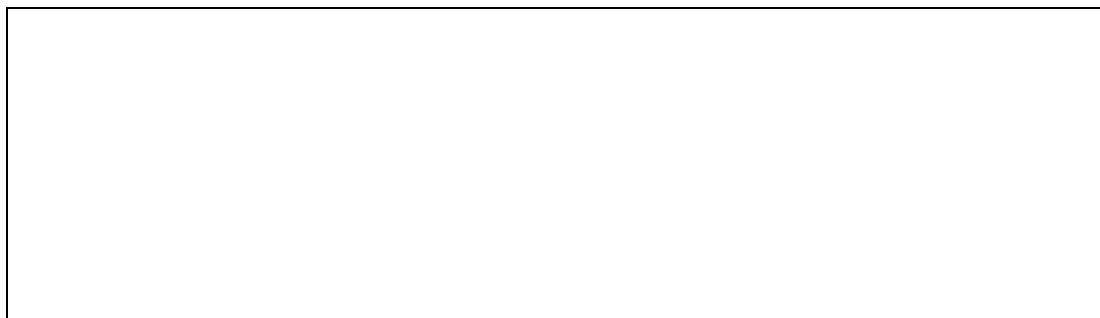
Naprogramujte jeho správanie napr. tak, aby sa pohyboval a po narazení na stenu aby sa náhodne otočil, buď doprava alebo doľava. Ak sa dotkne s myšky, hra sa skončí.

Program pre **jedného chrobáčka** môže vyzerat' aj takto:



Ďalších chrobáčikov vytvorte kopírovaním postavy chrobáka. Nezabudnite ich pri štarte programu umiestniť na iné miesta scény.

Napište, aké vylepšenia by ste chceli do svojej hry naprogramovať:



Žiak vloží na scénu postavu jedného chrobáka a naprogramuje tak, ako vidí v pracovnom liste.

Keďže bude zaujímavejšie, aby sa v labyrinte pohybovalo viac chrobáčikov, týchto získa kopírovaním už vytvoreného chrobáčka.

Žiak by nemal zabudnúť presunúť nových chrobáčikov na iné miesto scény.

Na konci pracovného listu majú žiaci možnosť zapísať svoje nápady na vylepšenie hry. Môžu sa objaviť napríklad nasledujúce návrhy:

- Pridať postavu, ktorá dobije myšku energiou tak, že pri kolízii pošle správu pre všetkých chrobákov, aby sa na určitý čas skryli, takže sa myška bude môcť voľne pohybovať.
- Pridať postavu, napríklad ducha, ktorý môže prechádzať aj cez steny a tiež zastaví celú hru.
- Doplniť do labyrintu „skratky“, teda miesta, do ktorých keď sa myška dostane preskočí na iné miesto v labyrinte.

PREZENTÁCIA – CCA 10 MIN.

Žiaci by si mali zahrať vytvorené hry, vzájomne preskúšať a otestovať svoje projekty. Na základe testovania by mali navrhnúť ďalšie vylepšenia a doplnenia do projektov.

10 HRA NIM 13

<i>Oblasť informatiky / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov: <ul style="list-style-type: none"> • analýza problému, • jazyk na zápis riešenia, • postupnosť príkazov Reprezentácie a nástroje: práca s grafikou	ISCED 2 / 7. - 8. ročník
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> • príkaz bublina, spájanie slov pomocou bloku spoj • podmienený príkaz ak • vytváranie vlastných blokov • posielanie správ • premenné, kostýmy a zmena kostýmu pomocou hodnoty premennej 	
Ciele	
Žiakom osvojované vedomosti a zručnosti	Žiakom rozvíjané spôsobilosti
Hra NIM <ul style="list-style-type: none"> • skúmať a analyzovať hru NIM 13 • objaviť stratégiu hry NIM 13 pre hráča, ktorý ťahá ako prvý, resp. ako druhý v poradí • rozhodnúť sa, či je hra vhodná na naprogramovanie v prostredí Scratch Prostredie Scratch <ul style="list-style-type: none"> • definovať vlastnosti hry a to, ako ich realizovať v scenári • zistiť, ktoré programátorské koncepty sa v programe vyskytnú • naplánovať postup riešenia (identifikovať a definovať údaje, ich inicializáciu, zmeny, ich vzájomný vzťah) 	Informatické myslenie <ul style="list-style-type: none"> • (ALG3) vytvárať vlastné algoritmy na riešenie problému • (ALG6) využívať existujúce algoritmy v návrhu vlastných algoritmov • (ALG8) zapísať algoritmy v konkrétnom programovacom jazyku Bádateľské spôsobilosti: <ul style="list-style-type: none"> • zaznamenávať výsledky pozorovania • konštruovať model a manipulovať s ním pomocou softvéru • zdieľať a prezentovať výsledky pred spolužiakmi
Dominantné vyučovacie metódy a formy	
Bádateľská metóda (5E). Frontálna forma.	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> • pracovný list 10_Hra_NIM13_PL • dokument – článok: Badateľsky orientovaný prístup ve výuce programování na 2. stupni ZŠ (str. 41) v publikácii: <i>Pech: Badateľsky orientovaná výuka matematiky a informatiky s podporou technológií (bovmi.pdf)</i>, o pravidlách a objavovaní výhernej stratégie hry NIM • prázdny projekt pre žiakov s jednou postavou s obrázkom zápaliek 10_Hra_NIM13 • riešené projekty pre učiteľa 10_Hra_NIM13_bez_obrazku, 10_Hra_NIM13_s_obrazkom, 10_Hra_NIM13_jednoducha 	
Diagnostika splnenia vzdelávacích cieľov	
Žiacke odpovede vo fáze Vysvetlenie. Pozorovanie činnosti žiakov vo fáze Skúmanie a Rozpracovanie . Spoločné hodnotenie vytvoreného projektu spolužiakmi.	

ÚVOD

Prostredie Scratch sa dá použiť aj na tvorbu a vývoj malých hier, v ktorých je **počítač súperom hráča**. Pre počítač je potrebné **navrhnuť algoritmus** a následne ho **naprogramovať** tak, aby dokázal byť v hre protihráčom s rozumnou **stratégiou**. Niekedy je vhodné upraviť stratégiu, v ktorej počítač vždy vyhrá tak, aby mal aj hráč šancu na víťazstvo.

Pri takto koncipovaných hodinách informatiky musia žiaci teda **nielen dokázať navrhnuť**, ako preniesť samotnú hru do počítača, pripraviť jej dizajn a ovládanie, ale aj **objaviť víťaznú stratégiu** hry, navrhnuť **algoritmus** víťaznej stratégie a následne ho naprogramovať.

Strategická hra NIM má veľa variácií, my budeme hrať s 13 zápalkami. Hra NIM 13 má pomerne jednoduché pravidlá a podľa dostupných výskumov a článku *Badatelsky orientovaný prístup ve výuce programování na 2. stupni ZŠ* (Pech: Badatelsky orientovaná výuka matematiky a informatiky s podporou technológií, str. 41-50,) vieme, že žiaci dokážu objaviť stratégiu hry NIM 13 približne za **desať** minút. **Pravidlá hry** sú jednoduché: dvaja hráči striedavo odoberajú 1, 2 alebo 3 zápalky z kopy 13 zápaliek. Ten, kto odoberie poslednú zápalku, prehráva.

Metodika tejto vyučovacej hodiny má charakteristiku **bádateľského** vyučovania. Žiaci najprv preskúmajú hru **NIM 13** tým, že budú vidieť jednu alebo niekoľko verejných partií učiteľa s niektorým žiakom alebo nejakú verziu hry na počítači, a následne budú hru hrať vo dvojiciach. V rámci niekoľkých partií by mali dokázať **sformulovať** víťaznú stratégiu hry **NIM 13**. Potom budú samostatne alebo spolu s učiteľom rozhodovať, či je hra vhodná na naprogramovanie v prostredí Scratch, ktoré poznajú a v ktorom programovali. Ďalej budú **analyzovať**, aké programátorské konštrukcie budú potrebné pri realizácii tejto hry do počítačovej podoby a aká bude programátorská náročnosť ich programu. Nakoniec každý z nich pripraví svoj vlastný projekt, v ktorom bude počítač protihráčom. Niektorým žiakom môžeme navrhnúť, aby pripravili menej náročnú hru bez stratégie pre počítač, viac pozri v časti **Rozpracovanie**.

V rámci programovania strategickej hry môžeme pre našich žiakov vyhľadať **aj inú strategickú hru**. Musíme si však uvedomiť, aké schopnosti žiakov sú potrebné na objavenie víťaznej stratégie hry a tiež to, aké zručnosti a programátorské skúsenosti musia mať žiaci, aby hru a jej stratégiu dokázali implementovať v prostredí Scratch.

ZAPOJENIE – CCA 8 MIN.

Na hodinu prinesieme reálne zápalky alebo špáradlá a na začiatku hodiny ukážeme vzorovú partiu hry **NIM 13**. Môžeme hrať s niektorým žiakom alebo aj s celou skupinou

žiakov. Počas hrania ukážkovej partie hry vysvetľujeme pravidlá hry tak, aby ich žiaci pochopili. Na papier, na tabuľu (alebo v tabuľkovom editore alebo v textovom editore, ktorý premietame pomocou dataprojektora) zapisujeme priebeh hry. Môžeme sa rozhodnúť, či pri tejto vzorovej hre budeme vždy **využívať víťaznú stratégiu** pre druhého hráča (počet odobratých zápalky je potrebné dopĺňať tak, aby prvý hráč a druhý hráč odobrili v jednom ťahu **spolu 4 zápalky**). Túto víťaznú stratégiu by mali žiaci postupne odhaliť. Učiteľ sa môže rozhodnúť aj tak, že v niektorej zo vzorových partii nechá vyhrať žiaka, ktorý ťahá ako prvý.

Zápis hry môžeme robiť „do stĺpca“ alebo „do riadku“ tak, ako na sú na obrázku ukázané dve vzorové hry. Môžeme sa tiež rozhodnúť, či budeme k odobratým zápalkám písať znamienko „-“ (mínus):

	vzorová hra	počet	žiaci	učiteľ	počet	žiaci	učiteľ	počet	žiaci	učiteľ	počet	žiaci	učiteľ	počet	žiaci	učiteľ	počet
počet zápalky	13	13	2	2	9	1	3	5	2	2	1	1	vyhral	0			
žiaci	-2																
učiteľ	-2																
počet zápalky	9																
žiaci	-1																
učiteľ	-3																
počet zápalky	5																
žiaci	-3																
učiteľ	-1																
počet zápalky	1																
žiaci	-1																
učiteľ	vyhral																
počet zápalky	0																

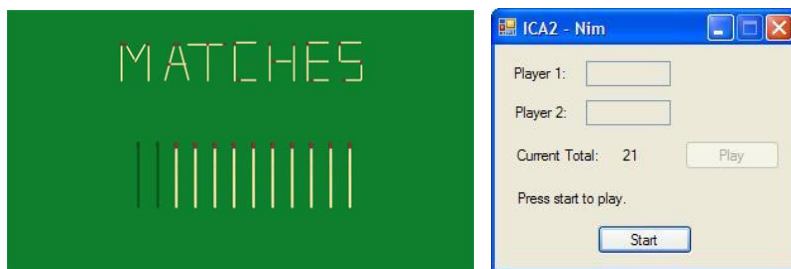
Ak chceme využívať iba víťaznú stratégiu, nemali by sme hrať príliš veľa vzorových partii, pretože je možné, že už pri ich hraní by šikovnejší žiaci dokázali zistiť víťaznú stratégiu (predovšetkým v prípade, ak je počet zostávajúcich zápalky vyznačený a tak je na prvý pohľad vidieť, aké zostávajúce počty zápalky sú významné).

Potom necháme žiakov hrať reálnu hru NIM 13 vo dvojiciach alebo v skupinách. Nabádame ich k tomu, aby si priebeh partii zapisovali. Počas niekoľkých reálnych partii medzi sebou sa žiaci snažia objaviť víťaznú stratégiu tejto hry.

Ďalším, resp. iným, spôsobom priebehu fázy **Zapojenie** môže byť to, že použijeme **hotovú hru v Scratch 10_Hra_NIM13_bez_obrazka**, ktorú dávame učiteľovi k dispozícii v tejto metodike. Hru ukazujeme vo forme **Na celú obrazovku** (ikona v Scratch vpravo hore



). Pomocou dataprojektora hru ukážeme žiakom tým, že učiteľ hrá partiu s počítačom, ktorú premieta cez dataprojektor a počas nej objasňuje pravidlá hry. Po tejto časti hodiny by mali byť **žiaci motivovaní** nielen k hraní hry, ale aj k tomu, aby začali uvažovať o víťaznej stratégii pre niektorého hráča aj k jej naprogramovaniu.



Rôzne varianty hry NIM, zdroj internet

SKÚMANIE – CCA 10 MIN.

Žiaci počas tejto časti hodiny hrajú hru NIM vo dvojiciach, pričom si robia záznam o tom, ako kto ťahal a kto vyhral. Žiaci si môžu ťahy hry zaznamenávať do tabuľky pripravenej v pracovnom liste alebo spôsob zaznamenávania ťahov môžeme nechať na žiakoch samotných. Tí majú z predchádzajúcich hodín, ktoré boli vedené bádateľským prístupom, už pomerne veľa skúseností so zapisovaním výsledkov bádania a preto veríme, že **si sami dokážu zvoliť** správnu metódu zápisu priebehu jednotlivých partii hry NIM.

Tu uvádzame **časť pracovného listu**. Riešenia pre **úlohu 1** neuvádzame – každý žiak bude mať svoje vlastné zápisky z hier. Víťaznú stratégiu sme zapísali jednoducho, ovládanie sme zvolili podľa toho, ako sme navrhovali v diskusii so žiakmi.

1. Do pripravenej tabuľky zapíš mená hráčov a počty zápaliek, ktoré v každom ťahu odobrali:

	hra1	hra2	hra3	hra4
počet zápaliek	13			

2. Aká je víťazná stratégia hry NIM 13 pre hráča, ktorý ťahá ako druhý v poradí?

ak hráč, ktorý ťahal zoberie 1 zápalku, druhý hráč zoberie 3 zápalky
 ak hráč, ktorý ťahal zoberie 2 zápalky, druhý hráč zoberie 2 zápalky
 ak hráč, ktorý ťahal zoberie 3 zápalky, druhý hráč zoberie 1 zápalku
 všeobecne: ak hráč zobral K zápaliek, druhý hráč zoberie 4 - K zápaliek

3. Ako sa bude dať ovládať Tvoja hra?

hráč zadá číslo 1, 2 alebo 3

potom zobrazím o toľko menej zápaliek

počítač bude ťahať tak, aby vždy vyhral

4. Naprogramuj hru NIM 13 s víťaznou stratégiou pre počítač.

VYSVETLENIE – CCA 5 MIN.

V tejto fáze vyučovacej hodiny by mali byť žiaci schopní, podľa svojich konkrétnych herných situácií a toho, ktorý hráč vyhral, vlastnými slovami **sformulovať hypotézu** o tom, či existuje spôsob a situácia, v ktorej niektorý z hráčov hru **vždy vyhrá**. Necháme žiakov, aby uviedli svoje hypotézy a vysvetlili ich svojim spolužiakom. Kladením otázok môžeme žiakom pomôcť precizovať formuláciu výhernej stratégie tak, aby ju pochopili všetci žiaci.

Ďalšia časť tejto fázy hodiny je venovaná tomu, ako hru naprogramovať v prostredí Scratch. Začíname **spoločnou diskusiou** k implementácii projektu do počítačovej podoby. Vedeť žiakov k tomu, aby navrhli ovládanie ich budúcej hry NIM a objasnili, aké programátorské konštrukcie budú pri jej programovaní potrebovať.

Na začiatku ich vedieme k tomu, aby sa rozhodli, či budú v hre zobrazené obrázky zápaliek alebo nie. Väčšina žiakov bude zrejme chcieť zobrazovať zápalky tak, ako to bolo v reálnej hre. Preto môžeme dať žiakom k dispozícii obrázok **zapalky.sprite3**, v ktorom je pripravený obrázok zápaliek.

Kolko zápaliek chces odobrať?
Stlač 1, 2 alebo 3.

pocet 13



Dva rôzne dizajny hry NIM v prostredí Scratch. Vľavo projekt s obrázkami zápaliek. Vpravo bez obrázkov.

Žiaci by mali byť schopní navrhnuť do projektu:

Obrázok zápaliek: Žiaci môžu navrhnuť, že budú využívať jednu postavu s 13 kostýmami alebo 13 postáv, kostým každej z nich tvorí jedna zápalka. V ďalšej diskusii vedieme žiakov k tomu, aby dokázali sformulovať, **aké akcie** budú postavy-zápalky robiť (zobrazovať sa a skrývať sa) a aj k tomu, aby si uvedomili, ako programátorsky náročné by bolo posielanie správ v prípade, že by sa postava mala ukázať alebo skrýť, ak by na každej z nich bola len jedna zápalka. Táto diskusia by mala viesť k tomu, že bude zrejme oveľa jednoduchšie mať **len jednu postavu**, ktorej budeme meniť kostým podľa toho, koľko

zápaliek je ešte v hre. Niektorým žiakom môže napadnúť, že nie je potrebné zobrazovať obrázok, stačí vedieť počet zápaliek, ktoré ostali v hre a tento počet môže byť vypísaný ako hodnota premennej alebo ho môže povedať napr. nejaká postava, ktorá bude v hre.

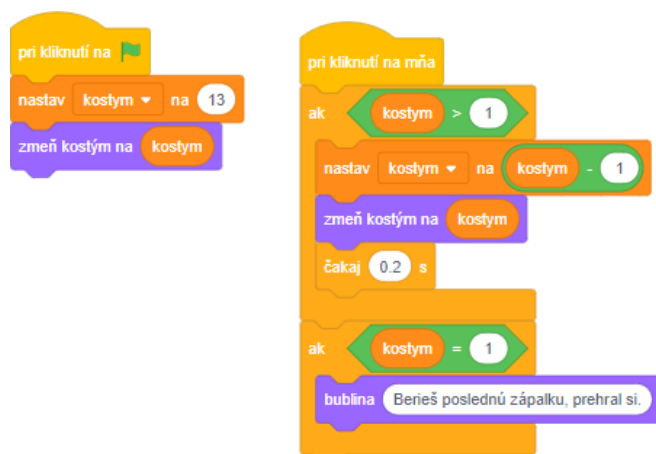
Ovládanie hry: Navrhnuť jednoduché ovládanie hry nemusí byť pre žiakov intuitívne a môžu mať tendenciu napr. vkladať do hry ďalšie postavy, ktoré budú zobrazovať číslce 1, 2, 3, podľa toho, koľko zápaliek chce hráč odobrať. Niektorí žiaci zrejme toto ovládanie dokážu bez problémov naprogramovať, takže spolu s nimi rozoberieme aj túto možnosť ovládania hry. Diskusiu môžeme viesť aj tak, aby žiaci pochopili, že pre jednoduchosť stačí (skryte, bez zobrazenia na scéne) reagovať na tri klávesy – čísla 1, 2, 3 a podľa nich zmeniť postavu kostým. Podobným spôsobom potom zabezpečíme ťah počítača.

ROZPRACOVANIE – CCA 20 MIN.

Keďže táto metodika priamo nadväzuje na sériu 9 predchádzajúcich metodík pre prostredie Scratch a programovanie v ňom, veríme, že žiaci už na tejto vyučovacej hodine **majú dostatok vedomostí** a aj **programátorských skúseností** a **zručností**, ktoré potrebujú pri programovaní tejto hry.

V tejto časti hodiny učiteľ **konzultuje** individuálne s jednotlivými žiakmi pri ich programovaní, radí a pomáha im, napr. pri definovaní premenných a rozdelenia kompetencií pre jednotlivé postavy. Žiaci pracujú samostatne na riešeniach svojich projektov, mali by dokázať tvorivo riešiť problémy, ktoré sa objavia počas programovania.

Dôležité Žiaci, ktorí **nemajú záujem** o programovanie môžu mať problémy s implementáciou takejto komplexnej hry. Učiteľ sa v takom prípade môže rozhodnúť, aby naprogramovali jej zjednodušenú verziu. V nej **nebudú** programovať stratégiu pre počítač, ale namiesto toho vytvoria **hru pre dvoch** hráčov. Ovládanie hry bude spočívať iba v klikaní na postavu zápaliek, ktorá si pri kliknutí zmení **kostým o 1**. Celú logiku hry, teda striedanie hráčov pri ťahaní a zabezpečenie toho, aby žiadny hráč neodobral viac zápaliek, ako je v pravidlách dovolené, si budú „strážiť“ samotní hráči pri klikaní na zápalky. Takto navrhnutá hra je programátorsky oveľa jednoduchšia, pozri obrázok s riešením **jednoduchého variantu** hry NIM13:



Riešenie s **výhernou stratégiou** s využitím jednej postavy s **kostýmami** zápalek od 1 do 13:

```
pri kliknutí na [praporek]
nastav kostym na 13
zmeň kostým na kostym
```

```
pri stlačení [1]
odober 1
nastav odobrať na 1
vyšli správu message1 a počkaj
```

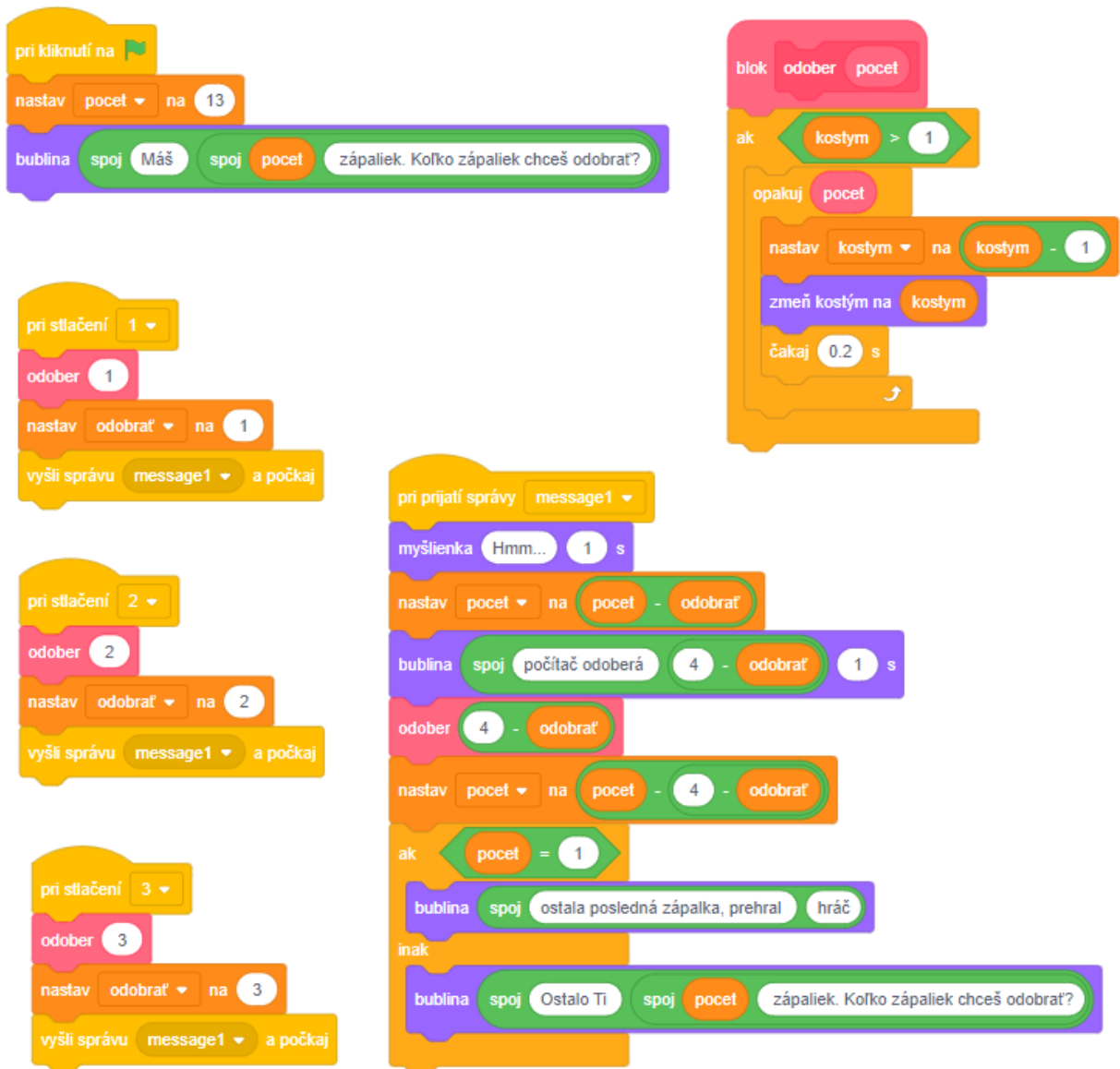
```
pri stlačení [2]
odober 2
nastav odobrať na 2
vyšli správu message1 a počkaj
```

```
pri stlačení [3]
odober 3
nastav odobrať na 3
vyšli správu message1 a počkaj
```

```
blok odober pocet
ak kostym > 1
opakuje pocet
nastav kostym na kostym - 1
zmeň kostým na kostym
čakaj 0.2 s
```

```
pri prijatí správy message1
myšlienka Hmm... 1 s
bublina spoj počítač odoberá 4 - odobrať 1 s
odober 4 - odobrať
ak kostym = 1
bublina spoj ostala posledná zápalka, prehral hráč
```

Riešenie s **výhernou stratégiou** s postavou hráča, ktorý **hovorí**, koľko zápaliek ešte máme:



Obe riešenie nájde učiteľ aj v príslušných projektoch.

HODNOTENIE – CCA 2 MIN.

Môžeme osloviť niektorého žiaka, aby prezentoval svoj projekt pred spolužiakmi. Hodnotenie môžeme urobiť komentovaním jeho projektu ostatnými žiakmi. Hodnotenie všetkých projektov urobí učiteľ po ukončení hodiny, resp. neskôr v prípade, ak žiaci dokončovali svoje projekty doma.

11 BLUDISKO

<i>Oblasť informatiky / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov	ISCED 2 / 7. - 8. ročník
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> • postava, kostýmy • scéna, pozadie • vetvenie • premenné • posielanie správ • práca s klávesmi, riadenie postavy pomocou šípok 	
Ciele	
Žiakom osvojované vedomosti a zručnosti	Žiakom rozvíjané spôsobilosti
Prostredie Scratch <ul style="list-style-type: none"> • v rámci tejto aktivity žiaci nezískavajú nové vedomosti a zručnosti, ale učia sa vhodne využívať tie, ktoré nadobudli na predchádzajúcich hodinách. 	Informatické myslenie <ul style="list-style-type: none"> • (ALG3) vytvárať vlastné algoritmy na riešenie problému • (ALG6) využívať existujúce algoritmy v návrhu vlastných algoritmov • (ALG8) zapísať algoritmy v konkrétnom programovacom jazyku Bádateľské spôsobilosti: <ul style="list-style-type: none"> • zaznamenávať výsledky pozorovania • konštruovať model a manipulovať s ním pomocou softvéru • zdieľať a prezentovať výsledky pred spolužiakmi
Dominantné vyučovacie metódy a formy	
Bádateľská metóda (5E). Frontálna forma.	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> • pracovný list pre žiaka 11_Bludisko_PL • projekt pre učiteľa 11_Bludisko_final (hotový projekt) • projekt pre učiteľa 11_Bludisko_ukazka 	
Diagnostika splnenia vzdelávacích cieľov	
Žiacke odpovede vo fáze vysvetlenia. Pozorovanie činnosti žiakov vo fáze skúmania a rozpracovania. Spoločné hodnotenie spolužiakmi, posúdenie správnosti navrhnutých postupov a programátorských riešení.	

ÚVOD

Pojem bludisko (alebo labyrint) by mohol byť pre žiakov pomerne známy. Bludiská by mohli poznať z gréckych báji, napr. o Ariadne a Théseovi, ktorý hľadal v labyrinte Minotaura, ale aj z rozprávok, či z detských časopisov, v ktorých sa často nachádzajú úlohy o hľadaní cesty v bludisku či labyrinte. Pre starších žiakov existuje pomerne veľa úloh, v ktorých sa často rieši problém nájdania najkratšej cesty v labyrinte.

V našej aktivite budeme uvažovať o jednoduchých labyrintoch a bludiskách, v ktorých sa bude dať pomocou šípok pohybovať postava (napr. guľôčka, lopta). Ďalej budeme spomínať pojem **stena** (postava nedokáže prechádzať cez stenu) a pojem **voľné políčko** (postava sa pohybuje po voľných políčkach).

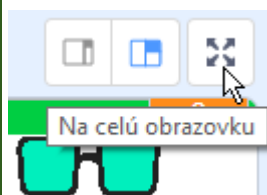
Nasledujúca metodika obsahuje sériu otázok, ktoré vedú žiakov od preskúmania, ako nakresliť svoje vlastné bludisko a ako ho zobraziť v pozadí **Scény** až po naprogramovanie prechádzania postavy bludiskom pomocou šípok.

ZAPOJENIE – CCA 5 MINÚT

V úvode vyučovacej hodiny ukazujeme žiakom pomocou **frontálnej ukážky** (využitím dataprojektora alebo interaktívnej tabule) hotový projekt **11_Bludisko_ukazka**. Dôležitou súčasťou tejto časti hodiny je **spoločná diskusia učiteľa a žiakov** a následná **dôsledná analýza** ukážkového projektu.

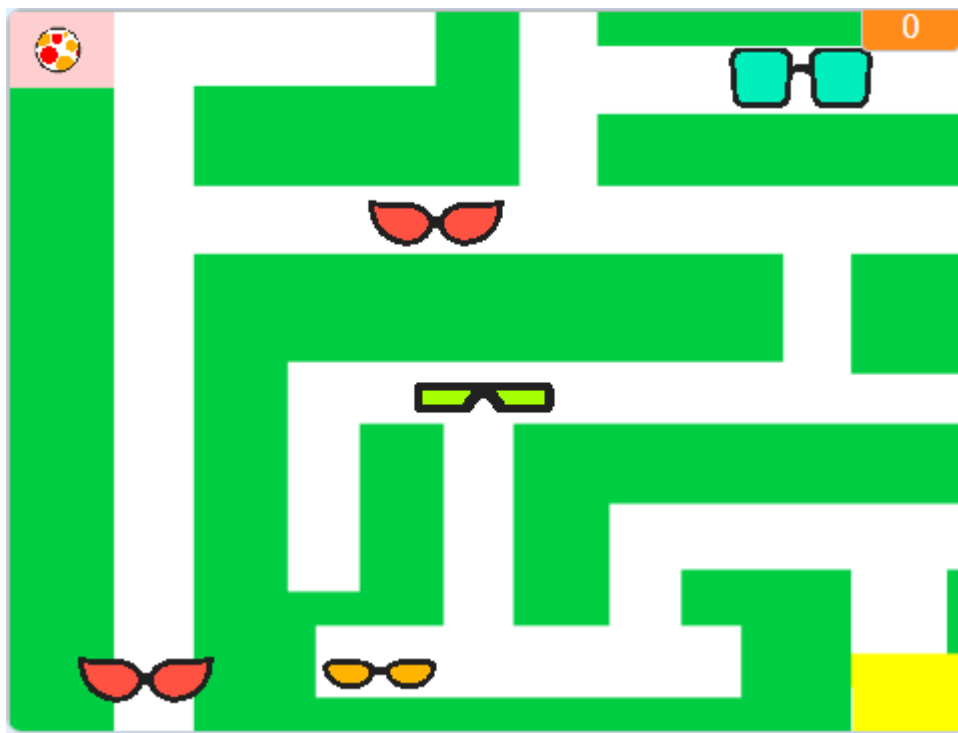
Aktivita 1

Otvoríme projekt **11_Bludisko_ukazka**. Projekt ukazujeme v **celostránkovom** režime, do ktorého sa prepne stlačením tlačidla v pravej hornej časti webovej stránky projektu:



Žiaci by mali byť schopní v projekte identifikovať niekoľko základných častí:

- Nakreslené bludisko ako súčasť **Pozadia Scény**.
- Zelené časti rozpoznať a pomenovať ako steny, biele časti ako tie, po ktorých sa môže postava pohybovať.
- Pohyb postavy s obrázkom lopty pomocou šípok.
- Niekoľko **statických** postáv, okuliare.
- Zobrazenú hodnotu premennej v pravom hornom rohu.
- Počas ukážky si majú žiaci všimnúť, že pri prechádzaní postavy **okuliare zmiznú** a pripočíta sa počet predmetov, ktoré postava pozbierala. Žiaci by si tiež mohli uvedomiť, aká programátorská konštrukcia je spojená s touto akciou – zrejme čakanie postavy s obrázkom okuliarov na to, kým sa stretne s loptou.



Projekt 11_Bludisko_ukazka

Učiteľ počas demonštrácie prechádza postavou, upozorňuje na to, ako sa postava správa pri stene, pri prekážke, ako reaguje na stretnutie s okuliarmi. Po ukončení ukážky by mali byť žiaci schopní identifikovať jednotlivé časti projektu a **slovne vyjadriť** základné informácie o štruktúre projektu. Okrem postáv, ktoré sa nachádzajú na scéne by mali identifikovať premennú, ktorá je zobrazená v pravom hornom rohu scény. Žiaci by mali dokázať zistiť, čo premenná zobrazuje, kedy sa vynuluje a kedy sa zvýši jej hodnota. Pro programovaní by potom mali vhodne zvoliť jej **meno**, napr. **pocetVeci**, **pocetNajdenych**, **pocetNajdenychVeci**.

Žiaci by mali prejaviť záujem preskúmať záložku **Pozadie** pre **Scénu**. Žiaci by si mali dokázať **spomenúť** na viacero projektov, v ktorých sa vyskytoval pohyb postavy pomocou šípok a mali by **chápať analógiu** s prezeraným projektom. Žiaci by mali poznať dva spôsoby, ktorými sa pohyb pomocou klávesov šípok dá v prostredí Scratch naprogramovať (relatívne otáčanie vľavo, vpravo a pohyb postavy a absolútny pohyb zmenou súradníc x a y) a rozhodnúť sa, ktorý z nich použijú (zrejme ten, s ktorým majú viac skúseností, resp. ktorý používali v doterajších projektoch).

SKÚMANIE – CCA 15 MINÚT

Žiaci riešia **úlohu 1** z pracovného listu. Pracovný list, ktorý je pripravený ako sada otázok, ktoré majú pomôcť žiakom **lepšie identifikovať** programátorské **podproblémy** pri programovaní projektu bludisko a hľadať a **navrhovať riešenia** pre ne.

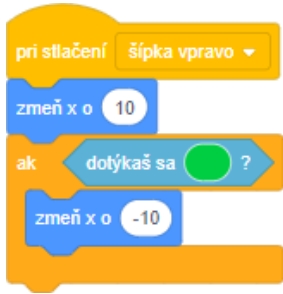
Niektoré programátorské postupy by mali žiaci poznať z predchádzajúcich projektov, napr. že ak chceme, aby postava **nedokázala** prechádzať cez nejakú farbu,

navrhne také riešenie, že postava sa pri svojom pohybe **posunie do vnútra** tejto farby (steny), ale hneď sa **vráti späť**.

Ak sú žiaci **šikovní**, pracovný list s otázkami vlastne **nepotrebujú** a môžeme s nimi hneď prejsť k časti **Vysvetlenie**.

Úloha 1

Podľa ukážky a diskusie so spolužiakmi k projektu **11_Bludisko_ukazka** odpovedz na otázky:

otázka	odpoveď
Kde je nakreslené bludisko?	bludisko je Pozadie , ktoré sme vytvorili pre Scénu
Aký kostým má postava, ktorá sa v bludisku pohybuje?	lopta
Po stlačení ktorých klávesov sa postava pohybuje?	po stlačení šípok
Cez akú farbu lopta nedokáže prejsť?	cez zelenú
Na obrázku je časť programu, ktorá zabezpečí, že pri stlačení šípky vpravo postava nedokáže prejsť cez zelenú farbu. Napiš, čo sa v programe deje.	 <p>Postava sa najprv posunie v smere x o 10 krokov.</p> <p>Potom pomocou príkazu ak zisťujeme, či sa dotýka zelenej farby.</p> <p>Ak sa dotýka zelenej, postava sa posunie v smere x o -10 krokov.</p>
Aký tvar majú postavy, ktoré po stretnutí s loptou zmiznú?	okuliare
Aká hodnota sa zobrazuje vpravo hore?	koľko okuliarov už lopta pozbierala

Kedy hra skončí?	keď postava - lopta príde na žlté políčko vpravo dolu
------------------	--

VYSVETLENIE – CCA 10 MINÚT

Po ukončení fázy **Skúmanie**, resp. odpovedaní na otázky spojené so správaním sa jednotlivých postáv, **vyvoláme** niektorého žiaka, ktorý ostatným vysvetlí, ako pripraviť bludisko, ako navrhnuť pohyb lopty pomocou šípok a **neprechádzanie** postavy stenami a tiež to, ako zistiť, či je postava v cieli. Ostatní žiaci sa vyvolaného žiaka pýtajú, vyjadrujú svoje nápady na riešenie predostretého problému, zapájajú sa do diskusie a snažia sa pochopiť navrhované postupy.

V tejto fáze vysvetľovania, a vlastne aj tvorby návrhu riešenia programu bludiska, **riadime diskusiu** tak, aby **všetci žiaci chápali**, čo a akým spôsobom je potrebné v bludisku naprogramovať. Snažíme sa docieľiť, aby žiaci dokázali **slovami sformulovať podmienku** pre obrázok labyrintu a pohyb postavy tak, aby sa postava pred stenou labyrintu "zastavila". Ak sa žiaci s takýmto postupom doteraz nestretli, musíme viesť diskusiu veľmi starostlivo, aby všetci žiaci pochopili "návrat" postavy pri dotyku s rovnakou farbou, akú má stena labyrintu a aj to, že **iné postavy, cez ktoré bude lopta prechádzať nemôžu** mať rovnakú farbu vo svojom obrázku.

ROZPRACOVANIE – CCA 10 MINÚT

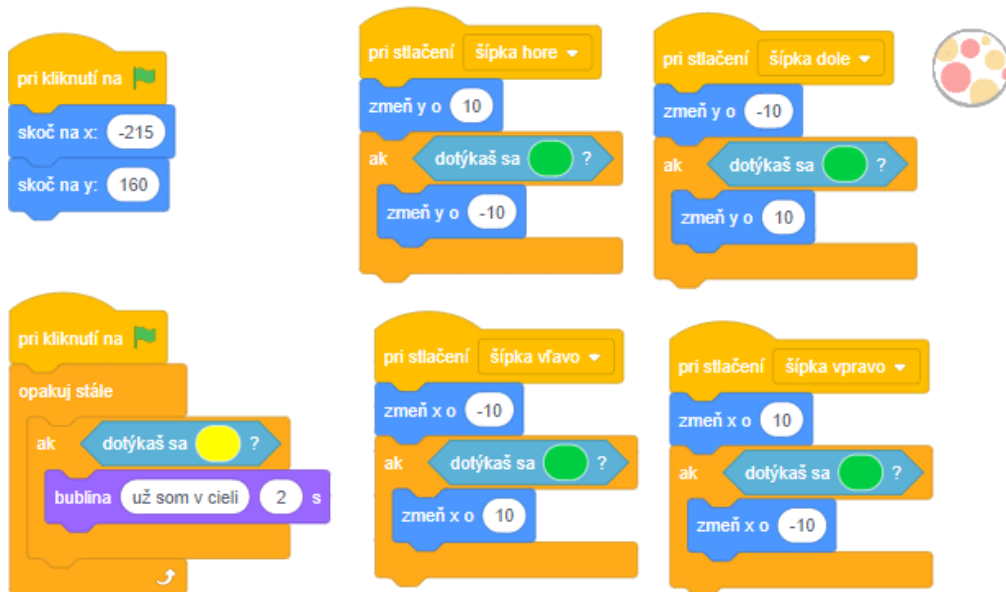
V tejto časti hodiny žiaci riešia **úlohy 2 až 4**, v ktorých majú naprogramovať svoje vlastné bludisko, navrhnuť kostým postavy, pripraviť jej pohyb pomocou šípok.

Vyzveme žiakov, aby si každý **nakreslil veľmi jednoduché vlastné bludisko (napr. len niekoľko obdĺžnikov) ako pozadie scény**, a aby naprogramovali riešenia **úloh 2 až 4**.

Úloha 2

Priprav svoj projekt, nakreslili si vlastné bludisko, navrhni svoju vlastnú postavu a naprogramuj jej pohyb tak, aby neprechádzala stenami. Naprogramuj aj to, aby postava povedala, že našla východ z bludiska.

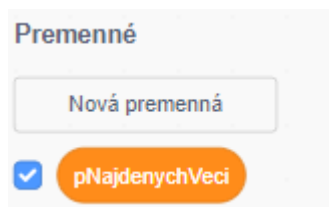
Riešenie **úlohy 2**, ktoré využíva pohyb postavy **Lopta** pomocou zmeny súradníc môžeme vidieť na nasledujúcom obrázku:



Úloha 3

Doplň do bludiska ďalšie postavy, ktoré musí pohyblivá postava "pozbiarať". Vpravo hore zobrazuj číslo, ktoré určuje, koľko predmetov už postava pozbierala.

Pre riešenie **úlohy 3** potrebujeme najprv zabezpečiť, aby sme mali premennú, do ktorej budeme počítat počet nájdených vecí na ceste lopty bludiskom. Vytvoríme premennú **pNajdenychVeci** – v časti **Premenné** stlačíme **Nová premenná**, premennú vhodne pomenujeme a necháme si ju zobrazit':

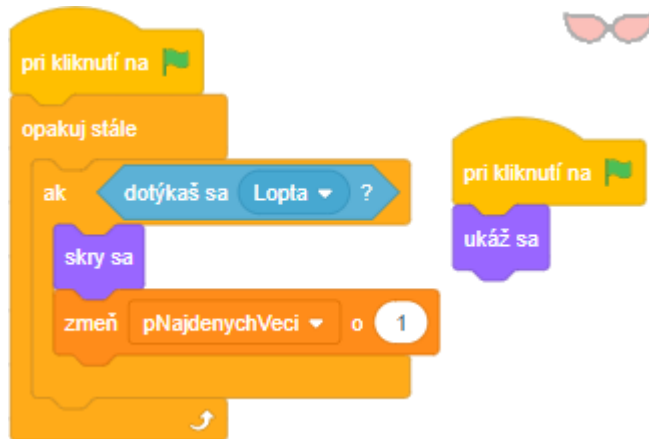


Pre postavu **Lopta** ešte upravíme časť programu pri kliknutí na zelenú zástavku, v ktorej premennú **pNajdenychVeci** vynulujeme. Vynulovanie premennej by sme mohli naprogramovať aj v scenári inej postavy, ale je lepšie, ak je inicializácia hodnôt na jednom mieste programu, t.j. v scenári jednej postavy.

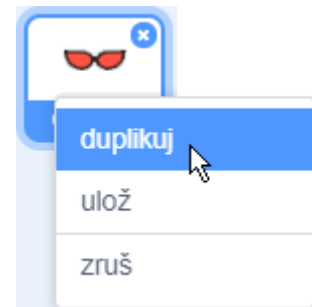


Teraz môžeme vytvoriť novú postavu, nazvime ju Okuliare1 a vyberme jej kostým z kostýmov Scratch. Postava sa **má skryť** pri stretnutí s pohybujucou sa postavou s menom **Lopta** – postava bude po štarte programu stále zisťovať, či sa dotýka postavy s menom

Lopta. V prípade ak áno, skryje sa a zvýšime o 1 premennú **pNajdenychVeci** . Pri kliknutí na zelenú zástavku sa postava ukáže. Žiakom treba pri vytváraní a programovaní scenárov pre postavu okuliare zdôrazňovať, aby do všímali, či naozaj pripravujú scenár tej postave, ktorej chcú (vidieť to v pravom hornom rohu plochy scenárov)



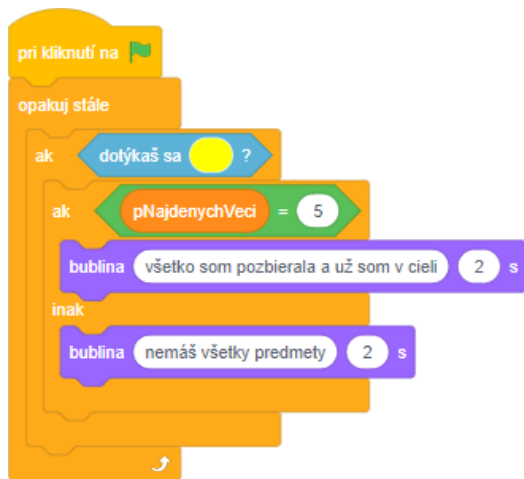
Rovnaký program ako sme práve pripravili pre postavu **Okuliare1**, ktorá má reagovať na stretnutie s postavou **Lopta**, budú mať aj ostatné postavy, ktoré sa majú pri stretnutí s postavou **Lopta** ukryť. Preto môžeme tieto ďalšie postavy – spolu so scenárom, ktorý sme práve pripravili – získať **duplikovaním** postavy s menom **Okuliare1**. Novým postavám, ktoré vytvoríme duplikovaním vhodne zmeníme tvar a ťahaním ju umiestnime na niektoré miesto v bludisku. Duplikovanie postáv by si mali žiaci **vyskúšať viackrát**, je to jedna z metód, ako ľahko získať postavy s rovnakým správaním. V budúcnosti budeme tento prístup používať aj pri generovaní postáv.



Úloha 4

Uprav správu na konci hry tak, aby postava povedala, či pozbierala všetky predmety.

Najprv si potrebujeme uvedomiť, do ktorej časti programu vložíme správu o počte predmetov. Koniec hry zisťuje postava **Lopta** v prípade, ak sa dotýka žltej farby. Ak chceme zistiť, či má všetky predmety, do scenára pre postavu **Lopta** doplníme testovanie a reakciu na prípad, ak je hodnota premennej **pNajdenychVeci** rovná 5 alebo nie je:



Rozširujúce úlohy

5. Dopln' do bludiska postavy – tenké zelené steny, ktoré sa budú v nejakom čase zobrazovať a skrývať.
6. Dopln' na vhodné miesta bludiska postavy, pri stretnutí s ktorými sa presunie lopta znovu na začiatok.
7. Keď sa lopta dostane do cieľa a pozbiera všetky predmety, zobraz ďalší labyrint. Obrázok vytvor ako ďalšie pozadie scény.

Riešenia rozširujúcich úloh neuvádzame v tejto metodike v textovej podobe. Učiteľ ich nájde v súbore **11_Bludisko_final**.

Rozširujúce úlohy sú určené pre žiakov, ktorí majú záujem o programovanie a ktorí ich riešia buď na hodine, ak im ostal čas, alebo si ich vyriešia v rámci domácej prípravy.

Pri ich riešení by žiaci mali prísť na to, že hoci navrhnu nové pozadie, je dôležité dodržať farbu stien. Postavy, ktoré sa v bludisku nachádzajú treba pri zobrazení ďalšieho pozadia tiež zobraziť (pretože pri prechádzaní bludiskom sa skrývali), prípadne ich presunúť na iné miesta (toto je však náročné na precíznosť a prácnosť riešenia).

HODNOTENIE – CCA 5 MINÚT

Keďže každý žiak bude mať svoje vlastné bludisko, svoje riešenia môžu prezentovať viacerí žiaci, resp. učiteľ môže vytvoriť **Scratch Štúdio**, v ktorom zverejní žiacke riešenia. V Scratch Štúdiu budú môcť všetci žiaci komentovať riešenia spolužiakov. Takýto spôsob **vzájomného hodnotenia** môže byť pre žiakov atraktívny a môže ich motivovať dokončiť svoj projekt aj doma.

Učiteľ **hodnotí** žiakov podľa **funkčnosti projektu** a správnosti riešenia jednotlivých jeho častí. Je potrebné uvedomiť si, že v takto koncipovaných projektoch je správnych riešení viacero. Pri hodnotení môže učiteľ brať ohľad nielen na funkčnosť riešenia, ale napr. aj efektívnosť, či kreativitu žiaka pri navrhovaní rôznych komponentov labyrintu.

12 ŠTVORČEKOVÁ SIETĚ

<i>Oblasť informatiky / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov	ISCED 2 / 7. - 8. ročník
<i>Požiadavky na vstupné vedomosti a zručnosti</i>	
<ul style="list-style-type: none"> Vedomosti a zručnosti zo všetkých predchádzajúcich aktivít tejto zbierky. 	
<i>Ciele</i>	
<i>Žiakom osvojované vedomosti a zručnosti</i>	<i>Žiakom rozvíjané spôsobilosti</i>
Prostredie Scratch <ul style="list-style-type: none"> v rámci tejto aktivity žiaci nezískavajú nové vedomosti a zručnosti, ale učia sa vhodne využívať tie, ktoré nadobudli na predchádzajúcich hodinách. 	Informatické myslenie <ul style="list-style-type: none"> (ALG3) vytvárať vlastné algoritmy na riešenie problému (ALG6) využívať existujúce algoritmy v návrhu vlastných algoritmov (ALG8) zapísať algoritmy v konkrétnom programovacom jazyku Bádateľské spôsobilosti: <ul style="list-style-type: none"> zaznamenávať výsledky pozorovania konštruovať model a manipulovať s ním pomocou softvéru zdieľať a prezentovať výsledky pred spolužiakmi
<i>Dominantné vyučovacie metódy a formy</i>	
Bádateľská metóda (5E). Frontálna forma.	
<i>Príprava učiteľa a pomôcky</i>	
<ul style="list-style-type: none"> pracovný list 12_Stvorcekova_siet_PL Scratch projekt pre žiakov 12_Stvorcekova_siet Scratch projekt 12_Stvorcekova_siet_ukazka s ukážkou pre žiakov a zároveň s riešeniami úloh 1 až 6, projekt 12_Stvorcekova_siet_rovnaky_pocet s riešeniami Rozširujúcich úloh 	
<i>Diagnostika splnenia vzdelávacích cieľov</i>	
Žiacke odpovede vo fáze vysvetlenia. Pozorovanie činnosti žiakov vo fáze skúmania a rozpracovania. Spoločné hodnotenie spolužiakmi, posúdenie správnosti navrhnutých postupov a programátorských riešení.	

ÚVOD

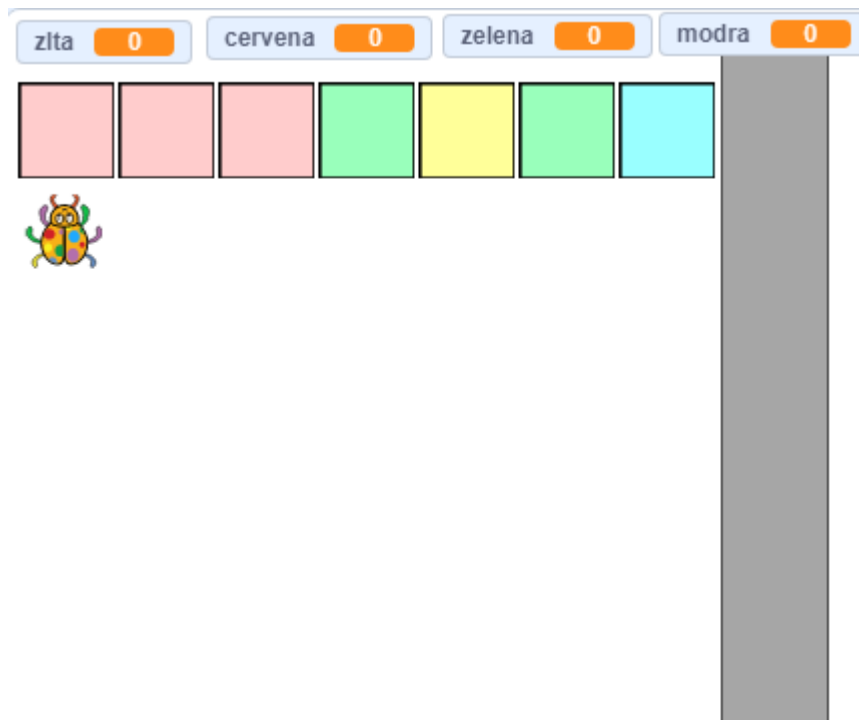
Prostredie **Scratch** je pomerne silný nástroj aj na vývoj hier. Môže ísť o komplexné multimediálne projekty alebo aj o programovanie jednoduchších hier, pomocou ktorých si žiaci osvoja, resp. rozvinú a hlbšie pochopia niektoré programátorské koncepty.

V oboch prípadoch si práca na týchto projektoch vyžaduje od žiakov **systematický** prístup k analýze problému a následnému návrhu riešenia. Z predchádzajúcich aktivít by mali mať žiaci pomerne **dobre osvojenú** dekompozíciu na čiastkové podproblémy súvisiace s riešením scény, postavami, ako aj návrhom scenárov, pozri aktivity 1 až 11.

Nasledujúca metodika hodiny obsahuje **sériu gradovaných úloh**, ktoré vedú žiakov od preskúmania, akým spôsobom sa vykresľujú štvorce do jedného riadku, k vygenerovaniu štvorčekovej siete (rozširujúce úlohy pre žiakov so záujmom o programovanie). Následne môžu žiaci naprogramovať hru, v ktorej postava prechádza po hracej ploche a zisťuje farbu štvorčeka, na ktorý prišla.

Aktivita 1

*V úvode vyučovacej hodiny učiteľ ukazuje žiakom pomocou frontálnej ukážky (využitím dataprojektora alebo interaktívnej tabule) hotový projekt **12_Stvorcekova_siet_ukazka**. Dôležitou súčasťou tejto časti hodiny je **spoločná diskusia** učiteľa a žiakov a následná **dôsledná** a podrobná analýza projektu.*



Projekt 12_Stvorcekova_siet_ukazka

Otvoríme projekt a ukazujeme ho v celostránkovom režime, aby žiaci priamo nevideli scenáre s riešeniami.

Projekt spúšťame **viackrát** a šípkami pohybujeme lienkou po jednotlivých štvorcoch. Žiakov upozorňujeme na hodnoty premenných **cervena**, **zelena**, **modra** a **zlta** a na to, že hracia plocha z farebných štvorčiek je vždy iná, teda štvorčiek majú rôzne farby.

Žiaci by mali byť schopní v projekte identifikovať niekoľko základných častí:

- vykreslenie riadku s rôznofarebnými štvorčkami
- pohyb lienky pomocou klávesov šípok
- počítanie počtu farebných štvorcov podľa jednotlivých farieb

Žiaci by mali byť schopní povedať, že štvorce vznikajú **pečiatkovaním** skrytej postavy.

Žiaci by mali prejaviť záujem o preskúmanie záložky *Kostýmy* tejto postavy, aby videli, ako vyzerá a utvrdili sa v tom, koľko má kostýmov. Počas ukážky môžeme žiakov vyzvať k tomu, aby **vysvetlili**, aké kostýmy musí mať skrytá postava.

Pri analýze pohybu lienky by si mali žiaci **spomenúť** na viacero projektov, v ktorých sa vyskytoval pohyb postavy pomocou šípok a mali by chápať ich analógiu s prezeraným projektom – napr. na minulej hodine pohybovali postavou v projekte **Bludisko**.

SKÚMANIE – CCA 15 MINÚT

Žiaci dostanú pracovné listy a pri ich vyplňaní skúmajú rozpracovaný projekt **12_Stvorcekova_siet**, v ktorom sú dve postavy (štvorček a lienka) a je v ňom pripravený blok **riadok** na vykreslenie jedného riadku štvorcov.

Úloha 1

Otvor projekt **12_Stvorcekova_siet**. Preskúmaj v ňom blok **riadok**, ktorý vykreslí vedľa seba do jedného riadku 7 štvorcov. Odpovedz na otázky.

otázka	odpoveď
Aký kostým má postava, ktorá vytvára štvorce? Koľko má kostýmov?	postava má tvar farebného štvorca, má 4 kostýmy
Ktorý príkaz zabezpečí, že sa štvorce vykreslia?	príkaz opečiatkuj
Na akom mieste Scény začína postava pri kreslení riadku?	začína v ľavom hornom rohu, na x: -210, y: 120
Aký smer má postava? Je viditeľná?	postava je natočená doprava, nie je viditeľná

Mení sa smer postavy počas vykresľovania riadku?	smer postavy sa nemení
Kde skončí postava po vykreslení štvorcov v riadku?	bude na konci za posledným štvorcom, ale nevidieť ju
Zapiš príkaz, ktorým sa postava pohybuje po opečiatkovaní jedného štvorca.	dopredu 50

VYSVETLENIE – CCA 10 MINÚT

Po ukončení skúmania v **úlohe 1** vyvoláme niektorého žiaka, ktorý vysvetlí, akým spôsobom postava kreslí riadok štvorcov. Ostatní žiaci mu pomáhajú, resp. zapájajú sa do diskusie a objasňovania riešenia, ktoré **dostali pripravené** v projekte.

V tejto fáze riešenia projektu je dôležité, aby všetci žiaci chápali postup, ktorý je v riešení navrhnutý pre vykreslenie jedného riadka štvorcov.

V rámci **spoločnej diskusie** následne necháme žiakov vyriešiť **úlohu 2**, v ktorej je navrhnutý pohyb lienky pomocou šípok

Úloha 2

Navrhni pohyb lienky pomocou klávesov – šípok. Na pohyb lienky dopredu navrhni vlastný blok **posun**.



Po stlačení zelenej zástavky zobraz lienku **50 krokov pod riadkom** štvorčekov. Vyskúšaj a pohybuju lienku po jednotlivých štvorcoch alebo aj mimo nich.

Žiaci by si mali spomenúť na riadenie postavy pomocou šípok. Ak je to potrebné, učiteľ pripomenie, že stláčanie šípok budeme hľadať v časti *Udalosti*, pričom primárne je

pripravená udalosť:

pri stlačení medzerník ▾

v ktorej sa dajú vybrať udalosti pri stlačení jednotlivých šípok:

pri stlačení šípka hore ▾

pri stlačení šípka dole ▾

pri stlačení šípka vpravo ▾

pri stlačení šípka vľavo ▾

Žiaci by si mali spomenúť, že po stlačení šípky je potrebné nastaviť postavu príslušným smerom a posunúť ju o niekoľko krokov, napr. 50.

Nový blok **posun** obsahuje zatiaľ iba jeden príkaz na pohyb postavy, žiakov však vedieme k tomu, aby tento nový blok vytvorili, pretože ho budeme využívať v ďalšej časti riešenia:



Úloha 3

Doplň program tak, aby lienka do premennej **zlta** spočítavala, koľkokrát pri svojom pohybe stúpila na žltý štvorček.

Ak je to potrebné, učiteľ žiakom na dataprojektore pripomenie prácu s premennými – vytvorenie premennej, nastavenie jej hodnoty na 0. Spolu so žiakmi potom vyhľadáva miesto v programe, kde je potrebné zvýšiť hodnotu premennej. Učiteľ pripomenie základnú konštrukciu **ak**. Tiež upozorní žiakov na to, že ak lienka stúpi na ten istý žltý štvorec viackrát, do premennej **zlta** sa **zakaždým** pripočíta 1.

Pri **spoločnom** riešení tejto úlohy by mal učiteľ žiakov **vyzývať**, aby sami postupne navrhovali jednotlivé kroky riešenia a uvádzali, ktoré bloky budeme v riešení potrebovať.

Žiaci by mali

- dokázať nájsť miesto, kde sú zobrazené premenné, identifikovať premennú **zlta** a spomenúť si, aké operácie sa s premennou dajú robiť – nastaviť jej hodnotu, zmeniť hodnotu
- určiť, na ktorom mieste programu je potrebné nastaviť hodnotu premennej **zlta** na 0, zrejme pri spustení projektu pomocou zelenej zástavky, a vložiť tam príslušný blok
- správne určiť, kedy treba zvýšiť hodnotu premennej **zlta** o 1, zrejme vždy, keď sa lienka posunie na nové miesto, zrejme v bloku **posun**
- určiť **podmienku**, ktorú je potrebné zadať do príkazu **ak**

Učiteľ sa pri vysvetľovaní uistí, že žiaci rozumejú vytvoreniu premennej, nastaveniu jej hodnoty na 0 a zvyšovaniu hodnoty v prípade, keď lienka stúpi na žltý štvorček.



ROZPRACOVANIE – CCA 10 MINÚT

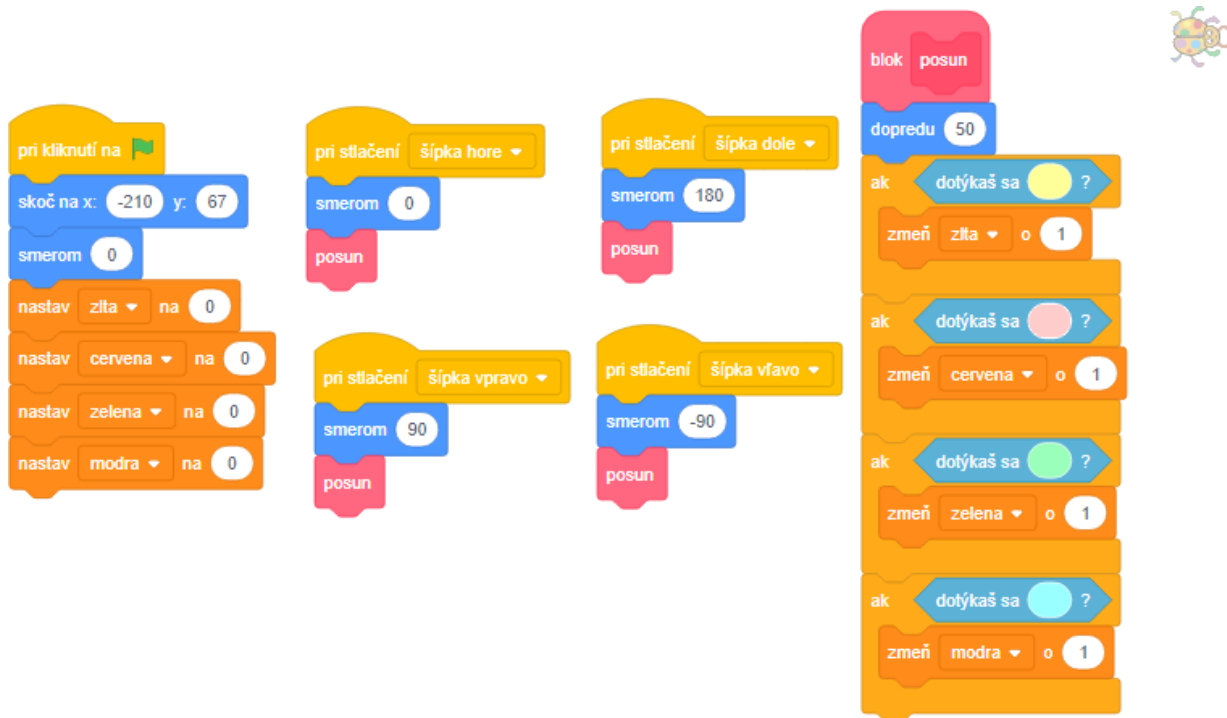
Na základe spoločného čiastočného riešenia **úlohy 3** by mali byť žiaci sami schopní doplniť odpovede do pracovného listu.

otázka	odpoveď
Kde je potrebné nastaviť hodnotu premennej na 0?	pri stlačení zelenej zástavky
Kedy sa má zvýšiť hodnota premennej zltá ?	ak postava stúpi na žltý štvorček

Úloha 4

Doplň program tak, aby lienka do premenných **cervena**, **zelena** a **modra** spočítavala, koľkokrát pri svojom pohybe stúpila aj na červený, zelený a modrý štvorček.

Na tejto úlohe si overujeme, či žiaci pochopili vytváranie a modifikáciu premenných a či dokážu sformulovať podmienky pre lienku a jednotlivé farby štvorčekov a správne ich umiestniť do skupiny už existujúcich blokov. Učiteľ žiakov pozoruje pri riešení a v prípade potreby pomáha tým žiakom, ktorí nevedia, ako postupovať ďalej. Žiakov navádza k tomu, aby vytvárali jednoduché podmienky pre každú farbu štvorčeka. Ak je to potrebné frontálne zobrazí, ako vybrať farbu zo Scény pre podmienku **dotýkaš sa**. Žiakov tiež navádza k tomu, aby niektoré časti blokov kopírovali, ale aby pritom nezabudli upraviť meno premennej, ktorú treba zvyšovať. Teda aby si jasne uvedomili, že pri dotyku s červenou farbou je potrebné zvýšiť o 1 hodnotu premennej **cervena**, pri dotyku s modrou farbou hodnotu premennej **modra**. Ak má učiteľ pocit, že kopírovanie takýchto častí robí žiakom problémy, viackrát im to zopakuje a predvedie. V programovaní je kopírovanie pomerne dôležitý postup, ktorý by mali mať žiaci čo najskôr zvládnutý.

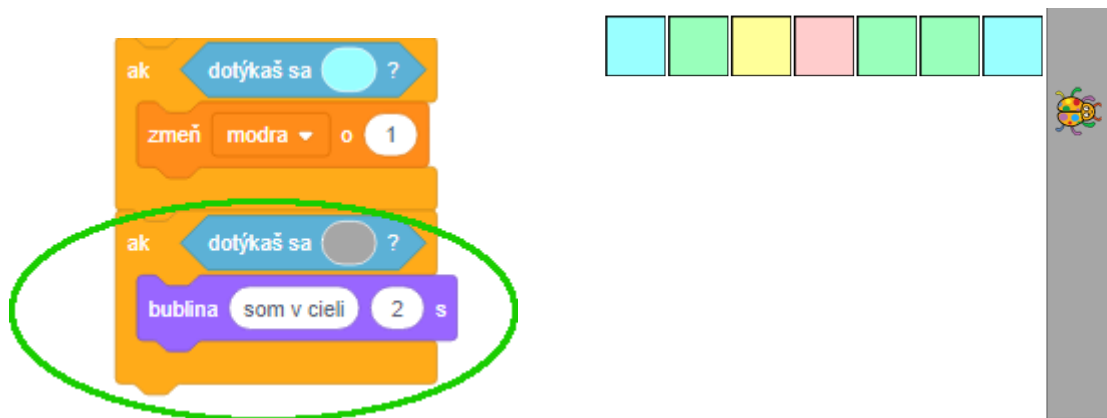


Úloha 5

Do pozadia Scény nakresli tmavosivú cieľovú čiaru. Doplň program tak, aby lienka povedala: "som v cieľi", keď sa pri svojom pohybe dostane na túto čiaru.

Spúščaj program niekoľkokrát a pohybuj lienkou tak, aby pri svojom pohybe chodila po štvorčekoch (a nie po bielej ploche), ale tak, aby počet zelených a modrých štvorčekov, na ktoré stúpi bol rovnaký.

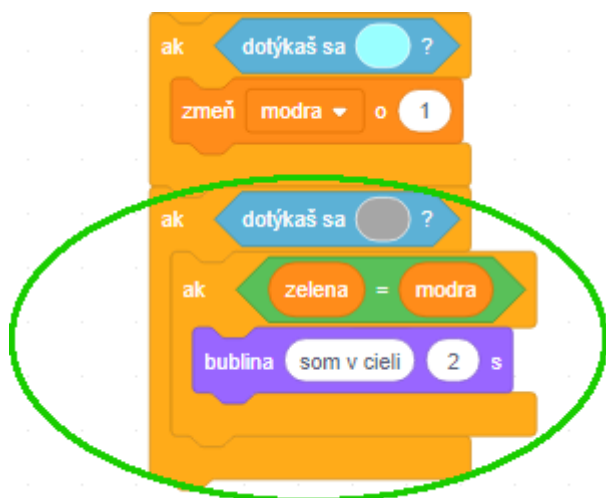
Pohybuj lienkou tak, aby na žiadny štvorček nestúpila viackrát.



Úloha 6

Doplň program tak, aby postava vypísala správu "som v cieľi", **iba vtedy**, ak sa rovná počet zelených a modrých políčok, na ktoré sa počas svojho pohybu dostala.

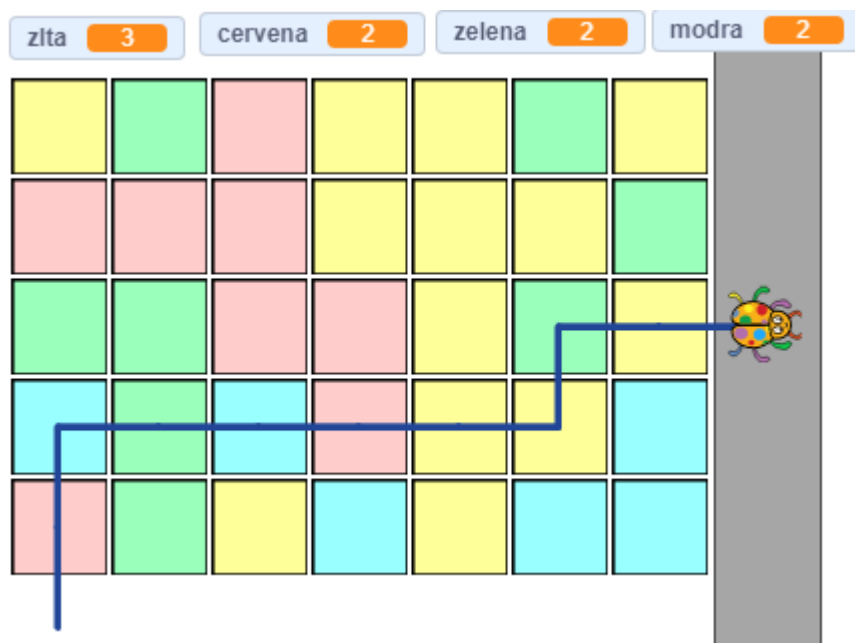
Žiaci by si mali uvedomiť, že ak sa postava nachádza na sivom políčku, musia ešte skontrolovať ďalšiu podmienku. Najjednoduchšie je vytvoriť novú podmienku a vložiť ju do podmienky, ak sa už postava dotýka sivej farby:



Teraz majú žiaci naprogramovanú najjednoduchšiu hru, v ktorej sa môžu snažiť pohybovať lienku pomocou šípok zo štartu do cieľa tak, aby navštívila rovnaký počet zelených aj modrých štvorčekov. Zatiaľ sa lienka môže pohybovať aj mimo štvorčekov, na bielej ploche alebo po žltých a červených štvorčekoch, ale aj napriek tomu sa **môžu snažiť hrať** túto hru tak, aby lienka vychádzala z hracej plochy farebných štvorčekov čo najmenej.

Rozširujúce úlohy

7. Doplň program tak, sa nakreslila hracia plocha s 5 riadkami.
8. Lienku zobraz na bielej ploche pod riadkami. Do bloku **posun** doplň podmienku, aby lienka počas svojho pohybu nemohla vyjsť do bielej plochy. Využi riešenie, pri ktorom sa lienka pôjde **najprv pozrieť** na miesto, na ktoré má prejsť a ak je tam biela plocha, vráti sa **naspäť**.
9. Zabezpeč, aby lienka **kreslila** za sebou tmavomodrú čiaru. Doplň program, aby sa lienka pri pohybe vždy vrátila, ak by sa mala dostať na políčko, na ktorom už bola. Pri riešení využi, že lienka pri pohybe najprv pôjde „pozrieť“ na políčko, na ktoré má prejsť. Ak na tom políčku modrá čiara **nie je** (teda na tomto políčku nikdy predtým nebola), vráti sa naspäť na svoje pôvodné políčko a na nové políčko prejde znovu, ale so zapnutým perom.



Riešenie rozširujúcich úloh v tejto metodike neuvádzame, učiteľ si ich môže pozrieť priamo v súbore s riešením **12_Stvorcekova_siet_rovnaky_pocet**.

Týmto vznikla kompletná hra na štvorčkovej sieti, ktorá môže byť vhodným základom aj pre ďalšie hry, v ktorých je hracia plocha zobrazená ako štvorčeková sieť, napr. Piškvorky alebo viaceré ďalšie, ktoré môžete nájsť na internete.

HODNOTENIE – CCA 5 MINÚT

Žiakov hodnotíme podľa naprogramovaných úloh a správnosti ich riešenia.

Na konci hodiny môžeme vyvolať niektorého žiaka, aby ukázal, kam sa v riešení úloh dostal. Je na zvážení učiteľa, či vyvolá žiaka, ktorý dokončil takmer všetky základné úlohy alebo takého, ktorý vypracoval iba časť riešenia. Vyvolaný žiak by mal byť schopný aj **slovami popísať** svoje riešenia.

13 PRESKAKUJ

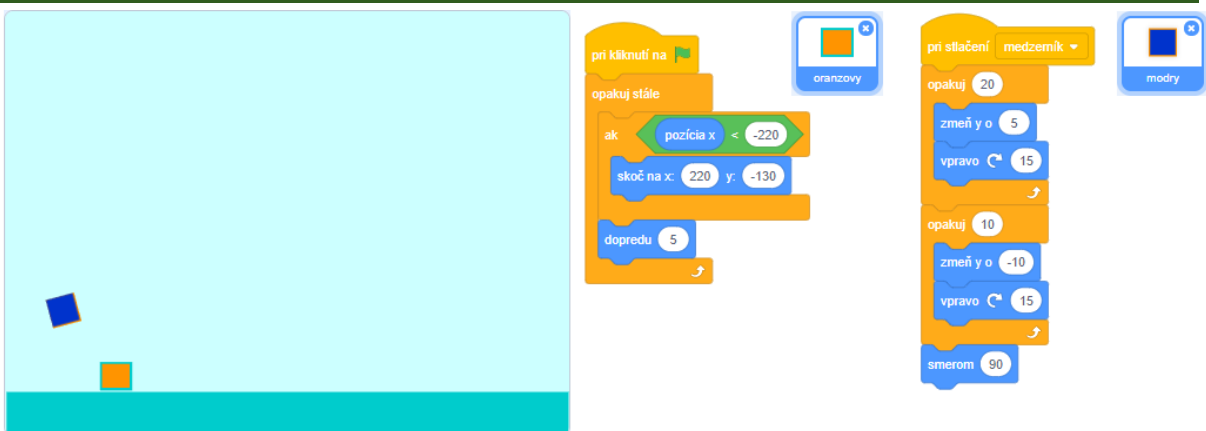
<i>Oblasť informatiky / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov / Skákacia hra	ISCED 2 / 7. - 8. ročník
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> Orientácia v prostredí Scratch, vedomosti a zručnosti zo všetkých predchádzajúcich aktivít tejto zbierky. 	
Ciele	
Žiakom osvojované vedomosti a zručnosti	Žiakom rozvíjané spôsobilosti
Prostredie Scratch <ul style="list-style-type: none"> v rámci tejto aktivity žiaci nezískavajú nové vedomosti a zručnosti, ale učia sa vhodne využívať tie, ktoré nadobudli na predchádzajúcich hodinách. 	Informatické myslenie <ul style="list-style-type: none"> (ALG3) vytvárať vlastné algoritmy na riešenie problému (ALG6) využívať existujúce algoritmy v návrhu vlastných algoritmov (ALG8) zapísať algoritmy v konkrétnom programovacom jazyku Bádateľské spôsobilosti: <ul style="list-style-type: none"> zaznamenávať výsledky pozorovania konštruovať model a manipulovať s ním pomocou softvéru zdieľať a prezentovať výsledky pred spolužiakmi
Dominantné vyučovacie metódy a formy	
Bádateľská metóda (5E). Frontálna forma.	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> pracovný list 13_Preskakuj_PL Scratch projekty 13_Preskakuj_ukazka, 13_Preskakuj_Uloha2 	
Diagnostika splnenia vzdelávacích cieľov	
Žiacke odpovede vo fáze vysvetlenia. Pozorovanie činnosti žiakov vo fáze skúmania a rozpracovania. Spoločné hodnotenie spolužiakmi, posúdenie správnosti navrhnutých postupov a programátorských riešení.	

ÚVOD

Pri programovaní hier je zaujímavé, ak sa postavy dokážu pohybovať pomocou klávesov a teda ak dokážu reagovať na stláčanie klávesov. S pohybom postavy pomocou klávesov – šípok hore, dolu, vpravo, vľavo – sme sa už stretli v predchádzajúcich aktivitách **11 Bludisko** alebo **12 Štvorčeková sieť**.

Aktivita 1

Vyvoláme niektorého žiaka a necháme ho skúšať funkčnosť pripraveného programu **13_Preskakuj_ukazka** tak, aby ho videli všetci žiaci na dataprojektore alebo interaktívnej tabuli. Súčasťou tejto časti hodiny je spoločná diskusia učiteľa so žiakmi a následná analýza ukážkového projektu.



Obr. 1 Projekt 13_Preskakuj_ukazka

Učiteľ prezentuje projekt v celostránkovom režime, aby žiaci **priamo nevideli** jednotlivé bloky.

Žiak sa "hrá" s projektom, pričom mu povieme, že projekt reaguje na kláves medzerník. Ostatní žiaci vidia správanie modrého štvorčeka, ktorý po stlačení medzerníka vyskočí a následne spadne na rovnaké miesto odkiaľ vyskočil. Všímajú si tiež pohyb oranžového štvorčeka. Program **nekontroluje**, či modrý štvorček narazil do oranžového.

Žiaci by mali byť schopní v projekte identifikovať niekoľko základných častí:

- Pozadie **Scény** – bledomodré s tmavším obdĺžnikom v dolnej časti.
- Postavy – modrý a oranžový štvorček.
- Pohyb modrého štvorčeka – vyskočenie a spadnutie – pri stlačení medzerníka.
- Nekonečnosť hry, žiaci by mali uvažovať, kedy by mali hru zastaviť.

SKÚMANIE – CCA 15 MINÚT

V časti **Skúmanie** riešia žiaci z pracovného listu **úlohy 1 aj 2. Úloha 1** im otázkami pomôže skúmať a bližšie popísať pohyb modrého štvorčeka a uvedomiť si, akými príkazmi sa dá takýto pohyb naprogramovať. Preskúmajú tiež pohyb oranžového štvorčeka.

Úloha 1

Podľa ukážky a diskusie so spolužiakmi k projektu **13_Preskakuj_ukazka** odpovedz na otázky a vypracuj nasledujúce úlohy.

Čo je nakreslené ako pozadie scény?	bledomodré pozadie a tmavšia "zem" v dolnej časti
Aké postavy sú v projekte?	modrý a oranžový štvorček
Po stlačení ktorého klávesu modrý štvorček vyskočí?	medzerník
Akými príkazmi sme zabezpečili vyskočenie a spadnutie modrého štvorčeka?	opakovaním zmeny y súradnice a otočenia
Ktorý príkaz zabezpečí, že sa modrý štvorček pri vyskočení otáča?	vpravo 15
Ako sme zabezpečili, že modrý štvorček stúpa pomalšie a padá rýchlejšie?	keď ide hore meníme y o 5 a keď padá meníme y o 10
Ktorým smerom je natočený oranžový štvorček?	k ľavému okraju obrazovky
Ktorý príkaz zabezpečuje, že sa oranžový štvorček pohybuje donekonečna?	príde naľavo a potom skočí znovu na pravú stranu obrazovky
Zapíš podmienku, pri splnení ktorej sa oranžový štvorček presunie znovu k pravému okraju obrazovky?	pozícia < -220

Kedy hra skončí?

nikdy

Úloha 2

Do tabuľky nižšie napíš, ako chceš projekt dokončiť.

Uvažuj napr. o tom, kedy má program skončiť, čo navrhnúť do programu, aby bola hra pre hráča náročnejšia, ako by sa hra zmenila, keby sa modrý štvorček pohyboval pomocou klávesov doprava/dolava, ako vypisovať dĺžku trvania hry...

Zapíš svoje návrhy na dokončenie hry

Riešenie úlohy neuvádzame, keďže každý žiak príde zrejme s rôznymi nápadi. Úvodné nápady sú uvedené priamo v zadaní **úlohy 2**.

Uveď, čo budeš v programovaní využívať

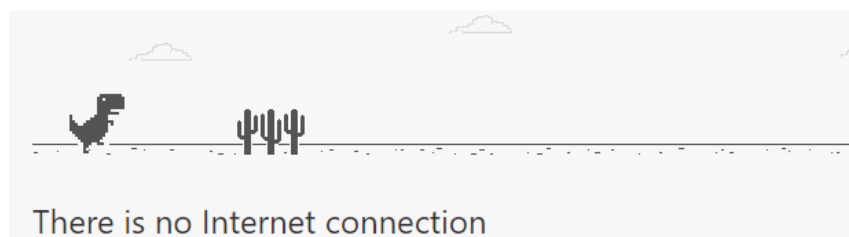
Riešenie úlohy neuvádzame.

Táto aktivita vedie žiakov k tomu, aby si **dokázali predstaviť**, čo by sa ešte v projekte mohlo nachádzať. Je dobré, ak žiaci v tejto fáze – môžeme ju nazvať príprava a návrh budúceho projektu – **ešte neuvažujú** o programátorskej náročnosti, ktorú bude implementácia v prostredí Scratch od nich vyžadovať. Žiaci by mohli prísť na niekoľko základných rozšírení projektu, ktoré sú naznačené v zadaní **úlohy 2**, ale môžu uvádzať aj svoje vlastné návrhy, na ktoré by mohli prísť vďaka tomu, že sa s podobným pohybom postáv zrejme stretli pri hraní hier na počítači, resp. mobilných zariadeniach.

Po ukončení fázy **Skúmania**, vyvoláme niektorého žiaka, ktorý spolužiakom vysvetlí, k akým záverom prišiel vo svojom skúmaní a aké námety na dokončenie, resp. rozšírenie projektu mu napadli.

Riadime diskusiu tak, aby všetci žiaci chápali jednotlivé časti úlohy a nabádame aj ostatných žiakov, aby povedali svoje námety na dokončenie úlohy, resp. spomenuli, s akými podobnými hrami sa stretli.

Napr. prehliadač **Chrome** má podobnú hru, ktorá sa nám zobrazí v prípade, že nemáme internetové pripojenie:

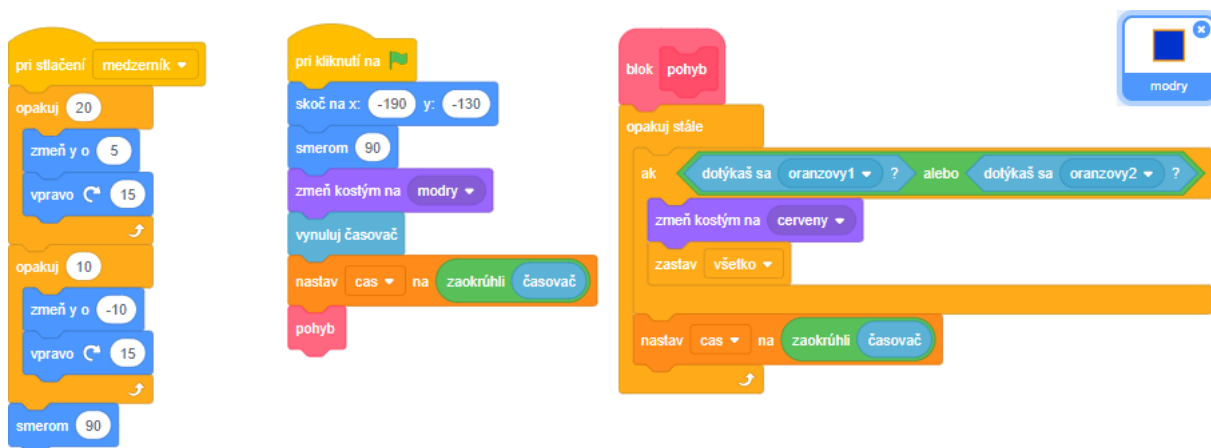


ROZPRACOVANIE – CCA 25 MINÚT

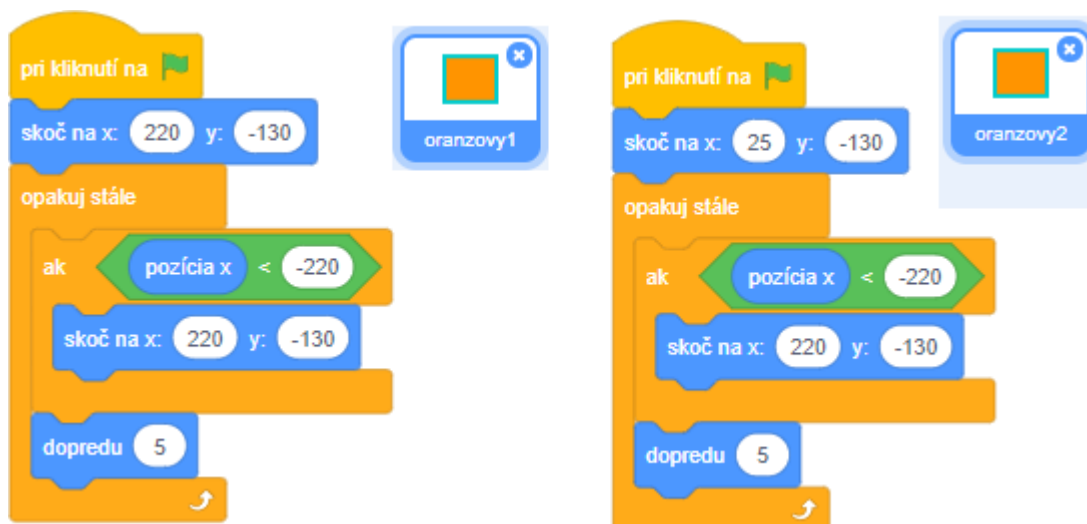
Vyzveme žiakov, aby každý naprogramoval niektoré zo svojich návrhov na dokončenie programu.

Tu uvádzame riešenie **niektorých** námetov z **úlohy 2**, ktoré nájdeme v projekte **13_Preskakuj_Uloha2**.

Na obrázkoch sú scenáre pre jednotlivé postavy, obrázok postavy je v pravej hornej časti scenára. Okrem postáv využívame časovač, ktorý sa vynuluje na začiatku hry a potom sa vkladá a zobrazuje pomocou premennej **cas**.



Všimnime si, že scenáre pre pohybujúce sa oranžové štvorčky sa líšia len v tom, že **oranzovy2** na začiatku hry skočí do stredu hracej plochy. Je to preto, aby oba oranžové štvorčky nezačínali na rovnakom mieste. Miesto kam má skočiť treba zistiť experimentovaním tak, aby sa hra dala hrať a nebola príliš náročná.



HODNOTENIE – CCA 5 MINÚT

Keďže sa žiaci mohli sami rozhodnúť, akým spôsobom budú program dokončovať, môžeme vyzvať viacerých žiakov, aby svoje programy prezentovali pomocou dataprojektora. Môžeme tiež vytvoriť **Scratch Štúdio**, v ktorom žiaci zverejnia svoje riešenia a môžu komentovať riešenia spolužiakov. Takýto spôsob **vzájomného hodnotenia** môže byť pre žiakov atraktívny a môže ich motivovať dokončiť projekt aj doma.

Žiakov hodnotíme podľa funkčnosti ich projektu a správnosti riešenia jednotlivých jeho častí. Je potrebné uvedomiť si, že v takto koncipovaných projektoch je **správnych riešení viacero**. Pri hodnotení môžeme brať ohľad nielen na funkčnosť riešenia, ale aj na kreativitu žiaka pri navrhovaní rôznych rozšírení programu.

14 POSIELANIE SPRÁV A REAGOVANIE NA NE

<i>Oblasť informatiky / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov / Posielanie správ a reagovanie na ne	ISCED 2 / 7. - 8. ročník
<i>Požiadavky na vstupné vedomosti a zručnosti</i>	
<ul style="list-style-type: none"> Orientácia v prostredí Scratch, vedomosti a zručnosti zo všetkých predchádzajúcich aktivít. 	
<i>Ciele</i>	
<i>Žiakom osvojované vedomosti a zručnosti</i>	<i>Žiakom rozvíjané spôsobilosti</i>
Prostredie Scratch <ul style="list-style-type: none"> v rámci tejto aktivity žiaci nezískavajú nové vedomosti a zručnosti, ale učia sa vhodne využívať tie, ktoré nadobudli na predchádzajúcich hodinách. 	Informatické myslenie <ul style="list-style-type: none"> (ALG3) vytvárať vlastné algoritmy na riešenie problému (ALG6) využívať existujúce algoritmy v návrhu vlastných algoritmov (ALG8) zapísať algoritmy v konkrétnom programovacom jazyku Bádateľské spôsobilosti: <ul style="list-style-type: none"> zaznamenávať výsledky pozorovania konštruovať model a manipulovať s ním pomocou softvéru zdieľať a prezentovať výsledky pred spolužiakmi
<i>Dominantné vyučovacie metódy a formy</i>	
Bádateľská metóda (5E). Frontálna forma.	
<i>Príprava učiteľa a pomôcky</i>	
<ul style="list-style-type: none"> pracovný list 14_Odomkni_PL Scratch projekty 14_Odomkni, 14_Preteky_riesenie 	
<i>Diagnostika splnenia vzdelávacích cieľov</i>	
Žiacke odpovede vo fáze vysvetlenia. Pozorovanie činnosti žiakov vo fáze skúmania a rozpracovania. Spoločné hodnotenie spolužiakmi, posúdenie správnosti navrhnutých postupov a programátorských riešení.	

ÚVOD

Pri programovaní hier a iných aktivít je často nevyhnutné, aby jedna z postáv reagovala na udalosť, ktorá sa stala inej postave. Napr. pri ak pri prechádzaní labyrintom nájde postava kľúč, môžu sa jej „odomknúť“ niektoré dvere (postava s kostýmom dvier) a tak si môže skrátiť cestu v labyrinte.

V tejto aktivite oboznámime žiakov s princípom posielania správ, reagovania na ne, ale aj s problémami, s ktorými sa môžeme pri tom stretnúť.

ZAPOJENIE – CCA 5 MIN.

Aktivita 1

Hotový projekt **14_Odomkni** zobrazujeme v celostránkovom režime na dataprojektore.

Vyvoláme niektorého žiaka a necháme ho skúšať funkčnosť pripraveného programu **14_Odomkni** tak, aby ho videli všetci žiaci. Súčasťou tejto časti hodiny je spoločná diskusia učiteľa so žiakmi a následná analýza ukážkového projektu.



Projekt 14_Odomkni

Žiak pracuje s projektom v celostránkovom režime. Povieme, že mačka reaguje na šípky – hore, dolu, vpravo, vľavo. Pri pohybe mačky sa zdanlivo nič nedeje, až dotedy, keď sa mačka nepriblíži ku krabici s darčekom. Vtedy sa zobrazí, že darček je „zamknutý“ a mačka sa presunie znovu do ľavého horného rohu. Žiaci zrejme pochopia, že mačka musí ísť najprv ku kľúčiku. Keď mačku dostanú ku kľúčiku, krabica zmizne a mačka bude mať voľnú cestu k darčekom. Po dotyku s darčekom mačka povie, že má darček.

Žiaci by mali byť schopní v projekte identifikovať niekoľko základných častí:

- Postavy – mačka, kľúčik, darček, krabica.
- Pohyb – mačky pomocou šípok.
- Reagovanie postáv na isté skutočnosti, ktoré nás zaujímajú – mačka sa dotýka krabice a reaguje **mačka aj krabica**.


- Reagovanie mačky, keď sa dotkne kľúča – reaguje kľúč, ale reaguje aj krabica (ukryje sa). Žiaci by si mohli uvedomiť aj to, že **tento typ reakcie sa nedá naprogramovať s tými vedomosťami**, ktoré majú doteraz, ale je nevyhnutné, aby sa dozvedeli niečo nové.

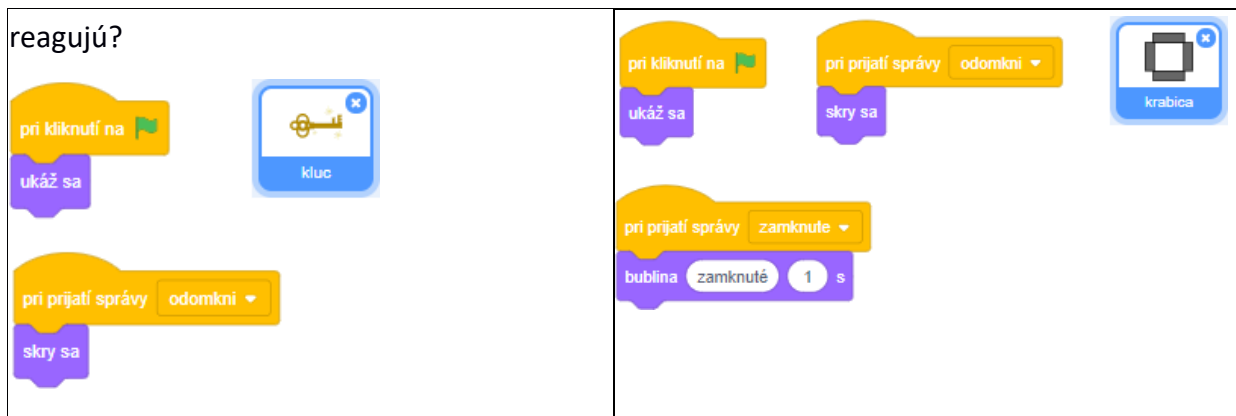
SKÚMANIE – CCA 15 MINÚT

V tejto časti vyučovacej hodiny riešia žiaci z pracovného listu **úlohy 1 aj 2**. **Úloha 1** im otázkami pomôže skúmať a bližšie popísať, čo sa deje v projekte, ale ukazuje im aj techniku posielania a reagovania postavy na správy a nutnosť pomenovať správu, aby na ňu mohla niektorá postava reagovať.

Úloha 1

Podľa ukážky a diskusie so spolužiakmi k projektu **14_Odomkni** odpovedz na otázky a vypracuj nasledujúce úlohy.

Aké postavy sú v projekte?	mačka, darček, kľúčik, krabica
Pomocou akých klávesov sa mačka hýbe?	pomocou šípok
Čo je cieľom hry?	aby sa mačka dostala k darčeku, ktorý je na začiatku uzamknutý v krabici
Ako sa môže mačka dostať k darčeku?	iba tak, že predtým pôjde ku kľúčiku a dotkne sa ho
Kedy sa krabica skryje?	keď sa mačka pri svojom pohybe dotkne kľúčika
Zapíš príkaz, ktorý umožňuje zistiť, či sa mačka dotýka kľúčika. Kedy by sa mal tento príkaz vykonávať?	dotýka sa kľúčik príkaz sa musí vykonávať stále
Keď sa mačka dotkne kľúčika, pošle správu s menom  . Na túto správu reaguje kľúčik a aj krabica. Napíš, akým spôsobom tieto postavy	kľúčik sa skryje krabica sa skryje



Úloha 2

Naprogramuj podobný projekt ako **14_Odomkni**.

Môžeš vymyslieť aj iný scenár, napr. kľúčik môže odomykať časť steny labyrintu alebo dvere...

ROZPRACOVANIE – CCA 20 MIN.

Vyzveme žiakov, aby každý naprogramoval hru alebo podobnú aktivitu, pri ktorej je nevyhnutné poslať správy, aby iná postava mohla na ne reagovať. Žiakom by mohol napadnúť napríklad projekt labyrint, do ktorého by mohli pridať kľúče a dvere, ktoré by sa odomkli, ak postava nájde kľúč.

Rozširujúce úlohy

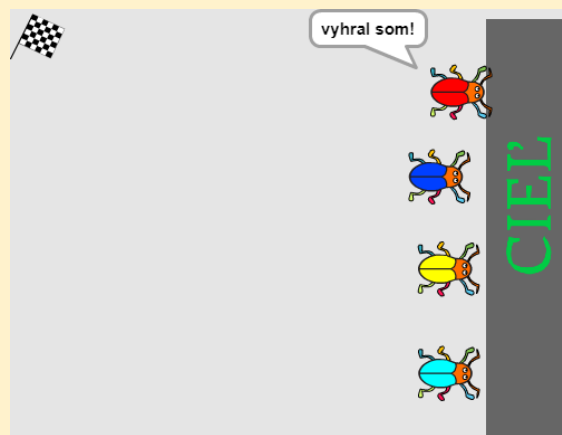
3. Naprogramuj hru **Tréning**, v ktorej dievčatko trénuje svojho psíka:

- po stlačení zelenej zástavky dievčatko hodí psíkovi loptu, ktorá sa presunie na náhodné miesto v hornej časti scény
- po kliknutí na dievčatko, dievčatko povie „Prines!“, pozri obr. vľavo
- keď psík začuje povel „Prines!“, natočí sa k lopte a prejde ku nej, zoberie ju, vráti sa k dievčatku, ktoré ho pochválí, pozri obr. vpravo



4. Priprav projekt **Preteky**, v ktorom

- sa po stlačení zelenej zástavky prostredia Scratch všetky chrobáčky presunú k ľavému okraju, teda sa pripravia na štart
- po stlačení čiernobielej **štartovacej** zástavky (postavy s tvarom zástavy) všetky chrobáčky naraz vyrazia smerom k cieľu, t. j. v každom kroku sa chrobáčikovi vygeneruje náhodná dĺžka kroku, o ktorú sa posunie
- ak sa hociktorý chrobáčik dostane do cieľa, povie správu „Vyhrál som!“ a on a aj všetky ostatné chrobáčky sa prestanú hýbať



Rozširujúce úlohy sú určené pre žiakov so záujmom o programovanie. Riešenia rozširujúcich úloh uvádzame v samostatných Scratch projektoch.

Pre veľmi šikovných žiakov je pripravené zadanie **Žiarovky**, ktoré nájdú v ppsx súbore, ktorý umožňuje vidieť animáciu. Učiteľ nájde riešenie v samostatnom projekte **14_ziarovky**.

HODNOTENIE – CCA 5 MINÚT

Keďže sa žiaci mohli sami rozhodnúť, ako upravia program v prvej úlohe, môžeme vyzvať viacerých žiakov, aby svoje programy prezentovali pomocou dataprojektora. Môžeme tiež vytvoriť **Scratch Štúdio**, v ktorom žiaci zverejnia svoje riešenia a môžu komentovať riešenia spolužiakov. Takýto spôsob **vzájomného hodnotenia** môže byť pre žiakov atraktívny a môže ich motivovať dokončiť svoje projekty aj doma, resp. navrhnúť iné, podobné aktivity.

Žiakov hodnotíme podľa funkčnosti ich projektov a správnosti riešenia jednotlivých jeho častí. Pri hodnotení môžeme brať ohľad nielen na funkčnosť riešenia, ale aj na kreativitu žiaka pri navrhovaní rôznych rozšírení programu.

15 DUPLIKOVANIE A KLONOVANIE OBJEKTOV

<i>Oblasť informatiky / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov / Klonovanie postáv aj s ich správaním	ISCED 2 / 7. - 8. ročník
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> Orientácia v prostredí Scratch, vedomosti a zručnosti zo všetkých predchádzajúcich aktivít. 	
Ciele	
Žiakom osvojované vedomosti a zručnosti	Žiakom rozvíjané spôsobilosti
Prostredie Scratch <ul style="list-style-type: none"> duplikovať postavu pomocou nástroja duplikuj z ponuky prostredia Scratch, využívať viacnásobné duplikovanie a rušenie postáv pri príprave rôznych scenárov, pre pokročilých: využiť blok naklonuj seba v programovom režime 	Informatické myslenie <ul style="list-style-type: none"> (ALG3) vytvárať vlastné algoritmy na riešenie problému (ALG6) využívať existujúce algoritmy v návrhu vlastných algoritmov (ALG8) zapísať algoritmy v konkrétnom programovacom jazyku Bádateľské spôsobilosti: <ul style="list-style-type: none"> zaznamenávať výsledky pozorovania konštruovať model a manipulovať s ním pomocou softvéru zdieľať a prezentovať výsledky pred spolužiakmi
Dominantné vyučovacie metódy a formy	
Bádateľská metóda (5E). Frontálna forma.	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> pracovný list 15_Duplikuj_PL Scratch projekty pre učiteľa 15_chytaj_muchy, 15_balony (ukazujeme ich v celostránkovom režime) 	
Diagnostika splnenia vzdelávacích cieľov	
Žiacke odpovede vo fáze vysvetlenia. Pozorovanie činnosti žiakov vo fáze skúmania a rozpracovania. Spoločné hodnotenie spolužiakmi, posúdenie správnosti navrhnutých postupov a programátorských riešení.	

ÚVOD

Pri programovaní hier sa pomerne často vyskytuje, že potrebujeme, aby na scéne boli, resp. vznikali a zanikali objekty, ktoré majú **rovnaké správanie**. Napr. po scéne plávajú rybičky, pohybujú sa muchy a úlohou hráča je klikáť na ne, padajú vločky, alebo kvapky, odlietajú balóny šťastia...

ZAPOJENIE A SKÚMANIE – CCA 5 MIN.

Aktivita 1

Ukážeme projekty **15_chytaj_muchy** a **15_balony**.



Projekty s klonovaním postáv

Oba projekty ukazujeme v celostránkovom režime. Pre projekt **15_chytaj_muchy** je to dôležité aj preto, že v inom režime sa žiakom pri hre môže stať, že prostredie nerozozná kliknutie na postavu, ale kliknutie rozpozná ako ťahanie a potom majú žiaci dojem, že projekt nefunguje správne. Projekty ukazujeme v celostránkovom režime, aj preto, aby žiaci priamo **nevideli** bloky s riešením.

So žiakmi podrobne diskutujeme o projektoch. Žiaci by mali byť schopní v projektoch identifikovať niekoľko základných častí:

- Postavy – v projekte **15_chytaj_muchy** majú postavy nielen rovnaké správanie ale aj tvar, v druhom projekte **15_balony** majú postavy „podobné“ obrázky a majú rovnaké správanie – putujú k oblohe.
- Pohyb – postáv, buď náhodný, alebo plynulý nejakým smerom.

V časti SKÚMANIE žiaci vypracujú **úlohu 1** a **úlohu 2** z pracovného listu. Mali by pochopiť, že v prípade, ak sa v projekte **nachádzajú postavy s rovnakým správaním**, je vhodné si toto správanie najprv pripraviť pre **jednu postavu**, ktorú potom **duplikujú**.

Úloha 1

Podľa ukážky a diskusie so spolužiakmi k projektu **15_chytaj_muchy** odpovedz na otázky a vypracuj nasledujúce úlohy.

Aké postavy sú v projekte?	v projekte je veľa rovnakých postáv – múch
Aký majú postavy kostým?	kostým muchy
Sú všetky postavy v projekte rovnaké?	áno, postavy sú rovnaké, majú rovnaký obrázok aj scenár
Akým spôsobom sa pohybuje každá mucha?	náhodne sa natočí a prejde o náhodný počet krokov, na okraji sa odrazí
Na akú udalosť reaguje každá mucha?	na udalosť pri kliknutí na mňa
Ako na túto udalosť mucha reaguje?	mucha sa skryje, počká 1 sekundu, presunie sa na náhodnú pozíciu a ukáže sa
Je správanie všetkých múch rovnaké?	áno, všetky muchy sa správajú rovnako
Kedy hra končí?	hra nikdy nekončí
Ako najrýchlejšie pridáme do hry ďalšiu muchu s rovnakým obrázkom aj s rovnakým správaním?	klikneme pravým tlačidlom na postavu v pravej dolnej časti a vyberieme duplikuj

V pracovnom liste sme modrým písmom zapísali, aké odpovede očakávame od žiakov. Pracovný list môžeme so žiakmi prejsť hneď po jeho vypracovaní. Žiaci by mali byť schopní vysvetliť svojimi slovami, čo sa deje v skúmanom projekte a formulovať aj ďalšie otázky, ktoré im napadnú pri práci s projektom.

ROZPRACOVANIE – CCA 35 MIN.

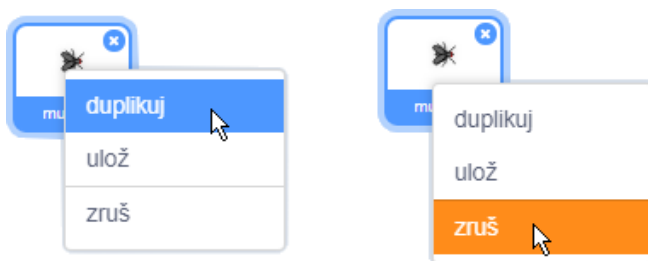
V tejto časti hodiny žiaci riešia ďalšie úlohy z pracovného listu.

Úloha 2

Naprogramuj svoj vlastný projekt **Chytaj muchy**, v ktorom bude úlohou hráča kliknutím na muchu chytať muchy.

Uvedom si, že najrýchlejšie takýto projekt vytvoriť je navrhnuť **najprv jednu muchu**, ktorej pripravíš kostým aj správanie a potom ju pomocou **duplikuj** „rozmnožiť“ toľkokrát, koľko múch chceme mať v hre.

Ak zistíš, že si postavil zabudol naprogramovať nejaké správanie, pomocou opakovaného mazania zruš okrem postavy **mucha1** všetky ostatné muchy, doprogramuj, čo potrebuješ a postavu znovu **duplikuj**.



Nástroje duplikuj a zruš.

Riešenie úlohy nájde učiteľ v súbore **15_chybaj_muchy**, ktorý slúžil ako ukázkový projekt v úvode hodiny.

Úloha 3

Navrhni projekt **Balóny šťastia**, v ktorom budú k oblohe letieť farebné balóny tak, ako to bolo v ukážke.

Učiteľovi dávame na zváženie, či dá žiakom vyriešiť **úlohu 3** aj **úlohu 4** alebo len jednu z nich. Úlohy sú si veľmi podobné, niektorí žiaci však môžu uvítať, že si duplikovanie precvičia aj na inej motivácii.

Na obrázku s riešením vidíme kostým a scenár **jedného** balóna a tým, že ostatné balóny sme vytvorili duplikovaním, tak vlastne aj všetkých balónov.

Aby boli balóny pri spustení letu rôzne, v programe vidíme, že balón si najprv vyberie náhodný kostým a nastaví si náhodnú veľkosť. Potom skočí na náhodnú pozíciu – x-ovú súradnicu si vyberá z celého rozsahu veľkosti scény, ale y-ovú súradnicu si vyberie iba v strede scény (aby nezačínal vo vode). Potom balón kľže na y-ovú súradnicu v hornej časti scény pričom si zachováva svoju x-ovú súradnicu. Týmto zabezpečíme, že balón letí kolmo hore.

Keď budeme mať hotový scenár **pre jeden** balón, vyskúšame ho a zistíme, že funguje, tak potom urobíme duplikáty tohto balóna **aj so správaním** – letením k hornej časti.



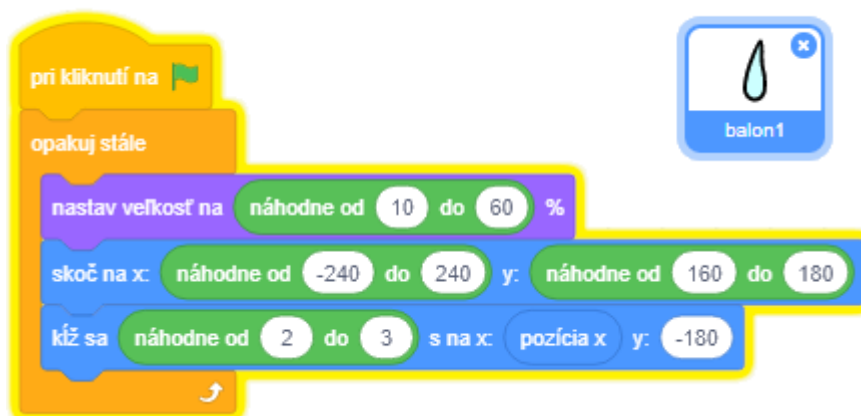
Let balóna smerom hore.

Úloha 4

Naprogramuj projekt **Dážď**, v ktorom bude „pršať“, t.j. z hornej časti scény budú padať kvapky. Duplikovaním kvapky dosiahni, aby bol dážď čo najprirodzenejší.

Ak si žiaci uvedomia, že dážď je vlastne kĺzanie „obrázku kvapky“ do dolnej časti scény, môžu pracovať tak, že nezačnú nový projekt, ale upravia predchádzajúci projekt **Balóny**.

Väčšina žiakov však zrejme vytvorí nový projekt, pretože im nenapadne, že stačí **prekresliť** balón na kvapku, zmeniť **smer** pohybu, skočiť na náhodnú pozíciu **v hornej časti** scény a kĺzať sa k dolnému okraju scény (všimnite si, že v našom riešení sme postavu nechali pôvodný názov **balon1**, hoci obrázok zobrazuje kvapku – počítaču je to jedno, človek však môže mať s takýmto prístupom problém. Ak na to žiaci neprídu sami, **nehovoríme** im o tejto možnosti.).



Padanie kvapky k dolnému okraju – y pozícia -180.

Pri riešení tohto projektu sa môže ukázať, že ak chceme, aby dážd' vyzeral prirodzene, je potrebné urobiť veľmi veľa duplikátov postavy.

Úloha 5

Navrhni vlastný projekt, v ktorom využiješ duplikovanie postáv. Do tabuľky zapíš, aký projekt budeš programovať a akú postavu budeš duplikovať.

Riešenie tejto úlohy neuvádzame. Žiakov by mohli napadnúť aktivity ako sneženie, plávanie rýb, lietanie lietadiel, lietanie viacerých rakiet vo vesmíre, lietanie kozmonautov vo vesmíre, pohyb áut a ľudí na ulici...

V niektorých návrhoch môžu mať postavy **iba podobné** (a nie úplne rovnaké) správanie a budú mať **rôzne** obrázky (rôzne vyzerajúce ryby, rôzne tvary vložiek...).

Počas individuálnej práce žiakov by mal učiteľ viesť žiakov k tomu, aby **vždy najprv vytvorili jednu postavu**, ktorú budú potom duplikovať.

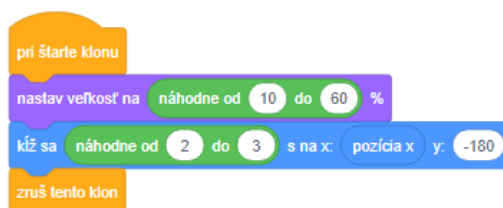
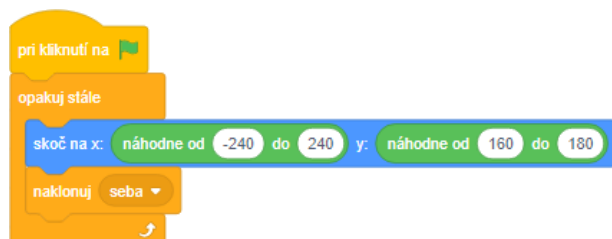
Z reálnych priebehov vyučovacích hodín vieme, že **tento spôsob uvažovania nie je jednoduchý**. Treba si však uvedomiť, že vytváranie „vzorov postáv“ je základom pre budúce objektové programovanie a navrhovanie **vlastných tried** v iných programovacích jazykoch, ako napr. Python.

Rozširujúca úloha

Úloha 6 (riešený príklad)

V projekte **Dážd'** využijeme namiesto duplikovania postáv klonovanie jednej postavy:

- v projekte vytvor **jednu** postavu s tvarom kvapky, so správnym natočením smerom nadol
- pri kliknutí na zelenú zástavku sa postava stále presúva na náhodnú pozíciu v hornej časti scény a tam sa naklonuje
- potom vyvolaj udalosť pri **štarte klonu**
- naprogramuj udalosť **štarte klonu**, v ktorej si klon zmení veľkosť a bude sa (tak ako v projekte **Dážd'**) kĺzať k dolnej časti scény
- po skončení kĺzania sa klon **zmaže** (toto je **dôležitá** časť programu, pretože v Scratch môže existovať len 400 klonov)
- uvedom si, že jediná postava kvapky (ktorá nie je klon) sa medzitým presunula niekoľkokrát na iné miesta, kde vytvorila ďalšie svoje klony



Riešený príklad ukazuje žiakom, **ktorí majú záujem o programovanie**, akým spôsobom môžeme **programovo** vytvárať klony a tak pripraviť „nekonečne“ veľa kvapiek bez toho, aby sme ich manuálne duplikovali. Scratch v súčasnosti dovoľuje vytvoriť len cca 400 klonov, preto je potrebné kvapky „zrušiť“. Žiaci by mali experimentovať s hodnotou v príkaze kľž, aby bol pohyb kvapiek realistický. Učiteľ môže nechať žiakov so záujmom o programovanie skúmať hotové riešenie a navrhnúť, aby rovnakým spôsobom upravili projekt **Balóny**, resp. svoj vlastný projekt, ktorý navrhli v **úlohe 5**.

V **prezentácii 15_jablka** sme pripravili ďalšie zadanie, ktoré môže dať učiteľ žiakom, ktorí javia výrazný záujem o programovanie. Obrázky nájdú žiaci v projekte **15_jablka_ziak**.

HODNOTENIE – CCA 5 MIN.

Žiakov hodnotíme podľa funkčnosti ich projektov a správnosti riešenia, ale aj dôslednosti dodržiavania postupu, pri vytváraní „vzoru“ postavy a jej duplikovaní, ktorý učiteľ pozoroval, keď žiaci riešili svoje projekty na hodine.

Je potrebné uvedomiť si, že v takto koncipovaných projektoch je **správnych riešení viacero**. Pri hodnotení môžeme brať ohľad nielen na funkčnosť riešenia, ale aj na kreativitu žiaka pri navrhovaní svojich vlastných programov, v ktorých je vhodné použiť duplikovanie postáv.

16 PROGRAMUJEME GRAFICKÝ EDITOR

<i>Oblasť informatiky / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov / Grafický editor – programujeme aplikáciu, ktorú poznáme	ISCED 2 / 7. - 8. ročník
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> Orientácia v prostredí Scratch, vedomosti a zručnosti zo všetkých predchádzajúcich aktivít tohto kurzu. 	
Ciele	
<i>Žiakom osvojované vedomosti a zručnosti</i>	<i>Žiakom rozvíjané spôsobilosti</i>
Prostredie Scratch <ul style="list-style-type: none"> v rámci tejto aktivity žiaci nezískavajú nové vedomosti a zručnosti, ale učia sa vhodne využívať tie, ktoré nadobudli na predchádzajúcich hodinách. 	Informatické myslenie <ul style="list-style-type: none"> (ALG3) vytvárať vlastné algoritmy na riešenie problému (ALG6) využívať existujúce algoritmy v návrhu vlastných algoritmov (ALG8) zapísať algoritmy v konkrétnom programovacom jazyku Bádateľské spôsobilosti: <ul style="list-style-type: none"> zaznamenávať výsledky pozorovania konštruovať model a manipulovať s ním pomocou softvéru zdieľať a prezentovať výsledky pred spolužiakmi
Dominantné vyučovacie metódy a formy	
Bádateľská metóda (5E). Frontálna forma.	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> projekt 16_graf_editor 	
Diagnostika splnenia vzdelávacích cieľov	
Žiacke odpovede vo fáze vysvetlenia. Pozorovanie činnosti žiakov vo fáze skúmania a rozpracovania. Spoločné hodnotenie spolužiakmi, posúdenie správnosti navrhnutých postupov a programátorských riešení. Vrstovnícke hodnotenie na základe testovania. Odporúčania na zlepšenie programu.	

ÚVOD

Nasledujúca metodika hodiny s názvom **Programujeme grafický editor** poskytuje učiteľovi možnosť prebrať so žiakmi tému grafika a práca s grafickými editormi z programátorského hľadiska. Ide o aktivitu, v ktorej môžu žiaci nahliadnuť a vyskúšať si, aké je to byť **tvorcami svojho vlastného grafického editora**.

ZAPOJENIE A SKÚMANIE – CCA 15 MIN.

V úvode vyučovacej hodiny učiteľ nechá žiakov otvoriť si a preskúmať nejaký grafický editor, v ktorom kreslili na iných hodinách informatiky alebo na hodinách výtvarnej výchovy. Odporúčame RNA alebo Skicár, podľa nainštalovaného operačného systému a softvéru, ktorý má škola k dispozícii.

Učiteľ položí žiakom niekoľko otázok a nechá ich, nech, skúmaním možností, ktoré poskytuje grafický editor, na ne získajú odpovede.

Žiaci by mali byť schopní po preskúmaní grafického editora zodpovedať na otázky:

- Načo slúži grafický editor?
- Aký je všeobecný postup, keď chcem v grafickom editore niečo nakresliť?
- Aké sú základné vlastnosti, resp. funkcie, grafického editora? Vymenujte niekoľko a popíšte ich vlastnými slovami.

Žiaci pracujú vo dvojiciach, pričom každá dvojica vytvorí svoj vlastný zoznam základných funkcií grafického editora, ktoré vyskúmala. Následne učiteľ postupne vyzve žiakov, aby prezentovali nájdené funkcie pred triedou. Potom si spoločne (celá trieda) vyšpecifikujú svoj vlastný zoznam základných funkcionalít, ktoré by mal mať grafický editor. Dôležitou súčasťou tejto časti hodiny je, aby žiaci naozaj spolu diskutovali prečo áno, prečo nie a čo zaradiť do spoločného zoznamu. Je to istá forma konfrontácie ich úsudku s názormi ostatných rovesníkov.

Výsledky skúmania

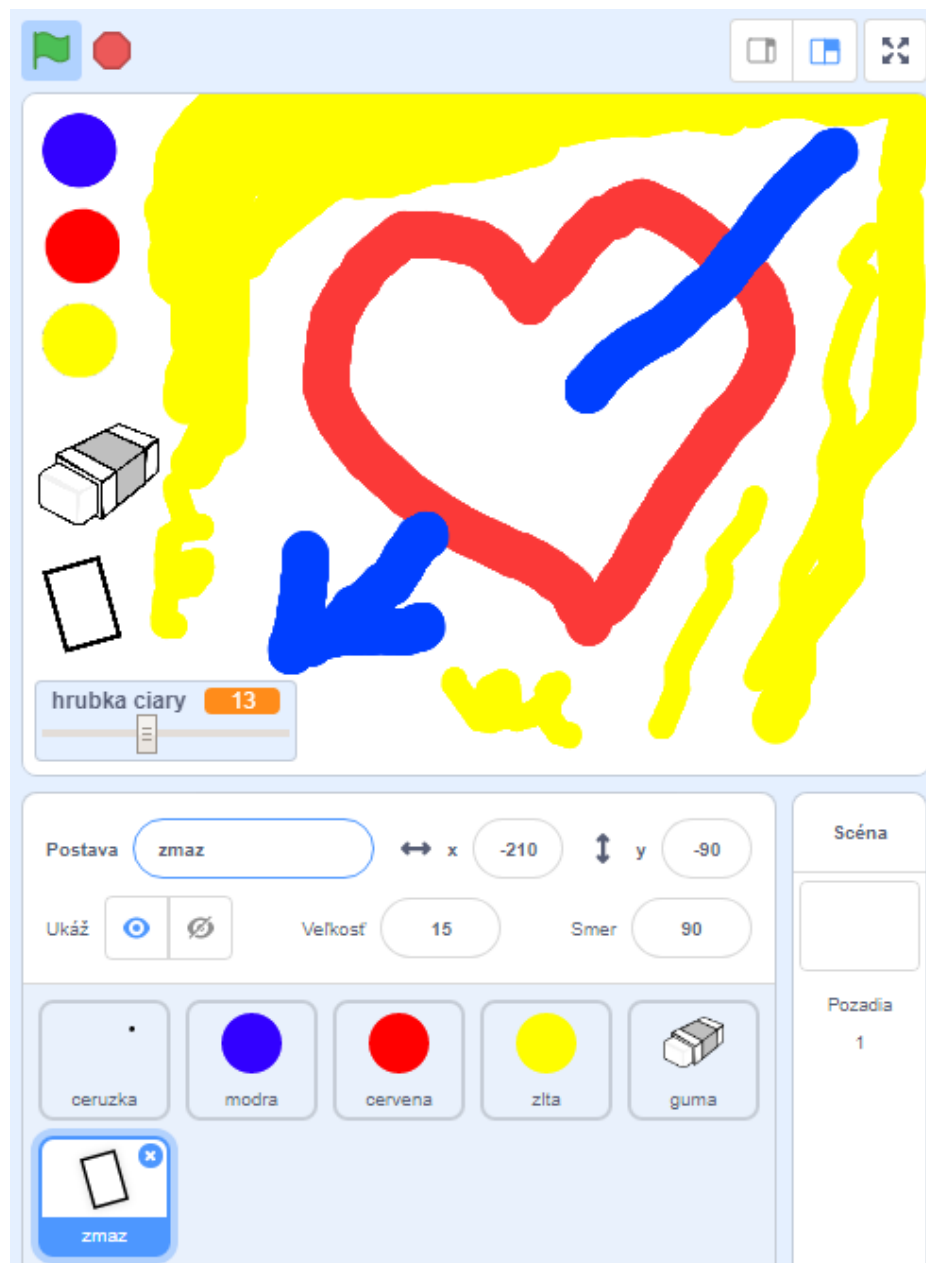
Žiaci by mali vytvoriť zoznam základných funkcií, ktoré má mať grafický editor a ktorý by mal obsahovať tieto funkcionality:

- kresliť do plochy (rôzne tvary, čiary, ...)
- meniť farbu (pera, objektu / plochy)
- vystrihovať a presúvať zvolené útvary
- gumovať
- nový papier

Po vytvorení tohto spoločného zoznamu učiteľ ukáže program **16_graficky_editor**, ktorý je pripravený v prostredí **Scratch** a ktorý má naprogramované len základné vlastnosti bežného grafického editora. Program učiteľ ukazuje v režime zobrazenia **na celú obrazovku**, aby žiaci priamo **nevideli bloky** s riešením. Učiteľ motivuje žiakov, že na konci

hodiny budú mať podobne funkčný program. **Pozor** ukážkový program funguje **správne** iba v režime zobrazenia **na celú obrazovku**, inak sa dajú po ploche presúvať aj tlačidlá na nastavenie farieb.

Pred začiatkom programovania učiteľ spolu so žiakmi prejde jednotlivé funkcionality, ktoré žiaci navrhli a snaží sa ich priviesť k tomu, aby **rozpoznali**, aké programátorské konštrukcie budú pri ich realizácii potrebovať. Učiteľ ich nedefinuje, skôr poukazuje na plusy a mínusy žiackych návrhov.



Hotový program s návrhom všetkých postáv

VYSVETLENIE – CCA 5 MINÚT

Aktivita 1

Po ukončení skúmania vyvoláme postupne niekoľko žiakov za sebou. Každý z nich sa sústreďí na niektorú časť predvážaného programu a na to, ako je zrealizovaná v prostredí Scratch. Určí objekt, ktorý je potrebné umiestniť do prostredia a uvedie, aké by mu navrhol správanie, resp. udalosti.

Žiaci by mali v projekte rozpoznať **postavy a premenné**:

- 6 postáv
 - ceruzku, ktorá kreslí (je skrytá)
 - nastavenia farby – modrú, červenú a žltú
 - gumu, vlastne bielu farbu
 - novú kresbu
- premennú **hrubka ciary**, ktorá určuje hrúbku pera, je zobrazená ako posúvač a ktorá je viditeľná pre všetky objekty

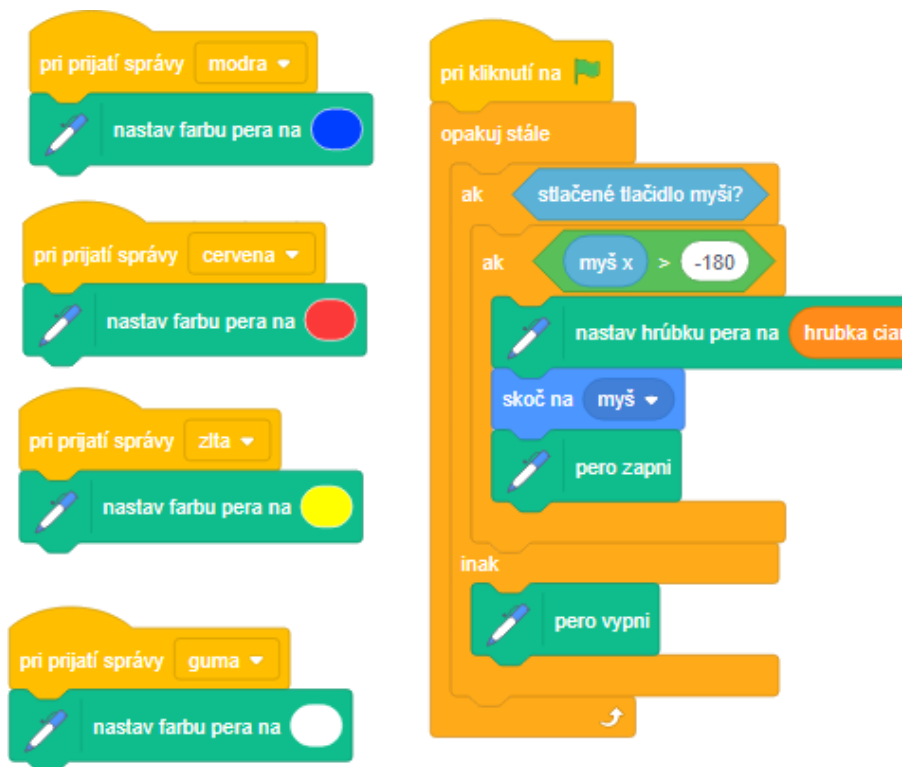
Žiaci by mali byť schopní popísať **funkčnosť** programu:

- farbu pera nastavujeme ceruzke, postavy (farbené kruhy), ktorými meníme farbu musia poslať ceruzke správu, akú farbu pera si má nastaviť



Bloky pre postavy, ktoré nastavujú farbu pera.

- zmenou posúvača meníme hrúbku pera ceruzky, pri každom kreslení si musí ceruzka nastaviť zvolenú hrúbku pera
- ak je stlačené ľavé tlačidlo myši, ceruzka sa presúva na pozíciu myši a zapne si pero



Bloky pre postavu, ktorá kreslí.

Riešenie úlohy nájde učiteľ v súbore **16_graficky_editor**.

Pre žiakov, ktorí nemajú záujem o programovanie, resp. nedokážu vytvárať komplexnejšie programy, sme do prezentácie **16_farebne_bodky** navrhli jednoduchší variant. Učiteľ nájde riešenie v rovnakom pomenovanom Scratch projekte.



Priprav projekt, ktorý bude kresliť na niektorom mieste scény pri stlačení tlačidla myši náhodné farebné bodky.

ROZPRACOVANIE – CCA 20 MIN.

Na základe skúmania, spoločnej diskusie, následného vysvetlenia a na základe vedomostí, ktoré majú žiaci z predošlých hodín, by mali byť schopní pre podobný program navrhnuť postavy, ich tvar a naprogramovať ich správanie.

HODNOTENIE – CCA 5 MIN.

Hodnotenie jednotlivých programov prebehne pomocou hodnotenia programu spolužiakom. Realizáciu hodnotenia navrhujeme tak, aby si žiaci pár minút pred koncom hodiny cyklicky presadli za počítač, na ktorom robil sused o dve miesta vpravo a budú testovať tento jeho program. Pri každom počítači bude pripravený papier, na ktorý tester zapíše programátorovi:

- **pozitíva**, ktoré sú podľa neho na programe spravené dobre

- **návrh na vylepšenie** jednej konkrétnej funkcie, ktorú podľa neho skúmaný program nespĺňa úplne dobre a dala by sa ešte vylepšiť

TABUĽKA NA ŽIACKE HODNOTENIE SPOLUŽIAKOVHO PROGRAMU

zapiš, ktoré funkcie grafického editora fungujú v programe správne	
napíš návrh na vylepšenie niektorej funkcie	

17 KRESLÍME ZVUK

<i>Oblasť informatiky / Téma</i>		<i>ISCED / Odporúčaný ročník</i>	
Algoritmické riešenie problémov / Programujeme kreslenie hlasitosti zvuku		ISCED 2 / 8. - 9. ročník	
<i>Požiadavky na vstupné vedomosti a zručnosti</i>			
<ul style="list-style-type: none"> Orientácia v prostredí Scratch, vedomosti a zručnosti zo všetkých predchádzajúcich aktivít. 			
<i>Ciele</i>			
<i>Žiakom osvojované vedomosti a zručnosti</i>		<i>Žiakom rozvíjané spôsobilosti</i>	
Prostredie Scratch <ul style="list-style-type: none"> v rámci tejto aktivity žiaci nezískavajú nové vedomosti a zručnosti, ale učia sa vhodne využívať tie, ktoré nadobudli na predchádzajúcich hodinách. 		Informatické myslenie <ul style="list-style-type: none"> (ALG3) vytvárať vlastné algoritmy na riešenie problému (ALG6) využívať existujúce algoritmy v návrhu vlastných algoritmov (ALG8) zapísať algoritmy v konkrétnom programovacom jazyku Bádateľské spôsobilosti: <ul style="list-style-type: none"> zaznamenávať výsledky pozorovania konštruovať model a manipulovať s ním pomocou softvéru zdieľať a prezentovať výsledky pred spolužiakmi 	
<i>Dominantné vyučovacie metódy a formy</i>			
Bádateľská metóda (5E). Motivačná hra. Diskusia. Frontálna forma.			
<i>Príprava učiteľa a pomôcky</i>			
<ul style="list-style-type: none"> pracovný list 17_Kresli_zvuk_PL mikrofón na učiteľskom aj žiackych počítačoch projekty 17_Kresli_zvuk, 17_Kresli_zvuk_vyfarbi, 17_Kresli_zvuk_1_az_8, 17_Kresli_zvuk_9_a_10 ak používate online prostredie, v prehliadači je potrebné povoliť používanie kamery 			
<i>Diagnostika splnenia vzdelávacích cieľov</i>			
Žiacke odpovede vo fáze vysvetlenia. Pozorovanie činnosti žiakov vo fáze skúmania a rozpracovania. Spoločné hodnotenie spolužiakmi, posúdenie správnosti navrhnutých postupov a programátorských riešení.			

ÚVOD

Nasledujúca metodika obsahuje aktivitu, ktorá vedie žiakov od uvažovania a skúmania zvuku, k vytvoreniu animácie, ktorá vykreslí hlasitosť zvuku, ktorý zaznamenal mikrofón v žiackom počítači, resp. notebooku.

Žiaci zvuky poznajú z rôznych aplikácií, na informatike sa stretli s rôznymi prehrávačmi zvuku, napr. Windows Media Player. Predkladaná aktivita využíva len poznatky, ktoré majú žiaci **z bežného života**, navrhuje, aby žiaci zvuk a jeho zaznamenávanie skúmali a pripravuje ich na lepšie na pochopenie témy o zvuku a jeho vlastnostiach. Téma zvuk sa vo fyzike podrobne preberá v osnovách na stredných školách.

ZAPOJENIE – CCA 5 MIN.

Aktivita 1 – Motivačná hra

V úvode vyučovacej hodiny si učiteľ zahrá so žiakmi **motivačnú hru** (netreba k tomu žiadne špeciálne pomôcky), bude stačiť, ak žiaci ostanú sedieť na svojich miestach a učiteľ im následne vysvetlí pravidlá: Všetci si zavrú oči a podľa toho, ktorý z nich bude vyvolaný, povie nahlas vetu: „Tichom sa šíril zvuk, ktorý zaplavoval mysle poslucháčov.“ (Alebo si učiteľ zvolí podľa vlastného uváženia nejakú inú vetu.) Následne bude učiteľ vyvolávať žiakov a ostatní budú potichu sedieť a vnímať, kto ako vetu povedal. Nič sa nemá komentovať. Potom žiaci môžu oči otvoriť a učiteľ začne diskusiu:

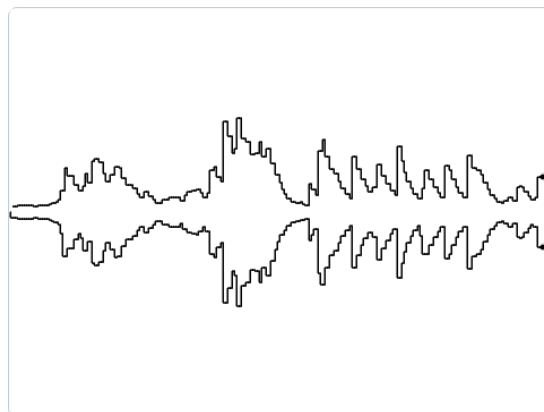
- **Opíšte ako spolužiak vyslovil vetu.**
 - V diskusii by mali žiaci dospieť k tomu, že niekto hovoril potichu, niekto nahlas, niekto (ne)jasne artikuloval. Niekto menil intonáciu, či mal rôznu farbu hlasu a pod.
- **Aké formy môže mať zvuk?**
 - Žiaci môžu začať rozprávať o formátoch (mp3, avi, ...) Ale učiteľ by mal nasmerovať diskusiu k zážitkovej skúsenosti.
- **Ako vieme vnímať zvuk?**
 - Napr. keď je hlasitosť zvuku veľmi vysoká, tak zvuk vie rozvibrovať okolité predmety, napr. reproduktory. Teda ho môžeme cítiť. Zvuk zvyčajne vnímame sluchom, lebo ho počujeme. Okrem toho môže mať zvuk aj vizuálnu podobu.

Dôležitou súčasťou tejto časti hodiny je **spoločná diskusia učiteľa a žiakov** pri ktorej bude učiteľ stavať na skúsenostiach, ktoré majú žiaci so zvukom a s hudbou. Následne začne spolu s nimi postupne budovať projekt kreslenia hlasitosti zvuku.

SKÚMANIE A ROZPRACOVANIE – CCA 35 MIN.

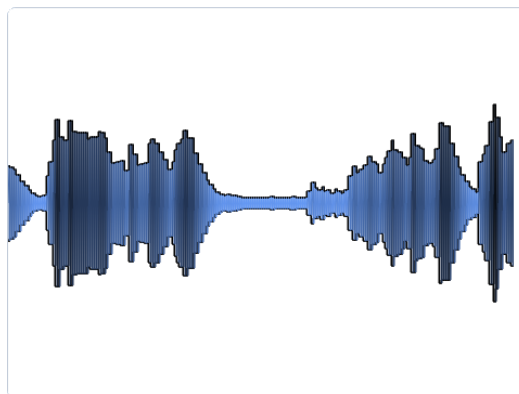
Pre túto aktivitu je potrebné mať **mikrofón**, ktorý je pripojený k počítaču, alebo ktorý je zabudovaný v notebooku. Potom, čo žiaci diskutovali o zvuku, a o tom, ako ho počuli, im predstavíme plán dnešnej hodiny: budeme sa hrať so zvukom tak, že budeme kresliť hlasitosť zvuku.

Predvedieme na projektore hotový projekt **17_Kresli_zvuk**. Žiaci môžu spoločne diskutovať, napr. o tom, koľko postáv je v projekte a aké scenáre zrejme majú.



Projekt 17_Kresli zvuk.

Alternatíva: Ak má učiteľ **žiacov so záujmom** o programovanie predvedie projekt **17_Kresli_zvuk_vyfarbi**, v ktorom je vnútro kresby vyfarbené podľa hlasitosti:



Projekt 17_Kresli zvuk_vyfarbi.

Ďalšiu časť hodiny môžeme viesť tak, že žiakom postupne zadávame úlohy, pomocou ktorých sami naprogramujú rovnaký projekt. Alebo môžeme dať žiakom pracovný list, v ktorom sa nachádzajú otázky a obrázky, ktoré ich privedú k naprogramovaniu kreslenia hlasitosti zvuku.

V ďalšej časti tejto metodiky uvádzame riešenia jednotlivých úloh z pracovného listu. Riešenie úloh nájde učiteľ v Scratch súboroch.

1. Začni nový projekt.

2. Nájdi premennú, ktorá obsahuje hlasitosť zvuku z mikrofónu. Zobraz premennú, rob hluč a sleduj hodnoty, ktoré premenná nadobúda.
Zapíš do tabuľky, akú najmenšiu a najväčšiu hodnotu mala premenná.

premenná	najmenej	najviac
hlasitosť	5	78

Premenná má meno **hlasitosť** a nachádza sa v skupine **Zisťovanie**.

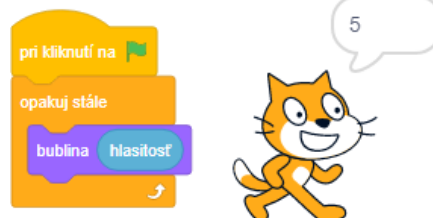
Stačí, ak žiaci zobrazia premennú na scéne, teda zapnú jej zobrazenie, a vidia, ako sa pri hluku mení jej hodnota:

hlasitosť 7



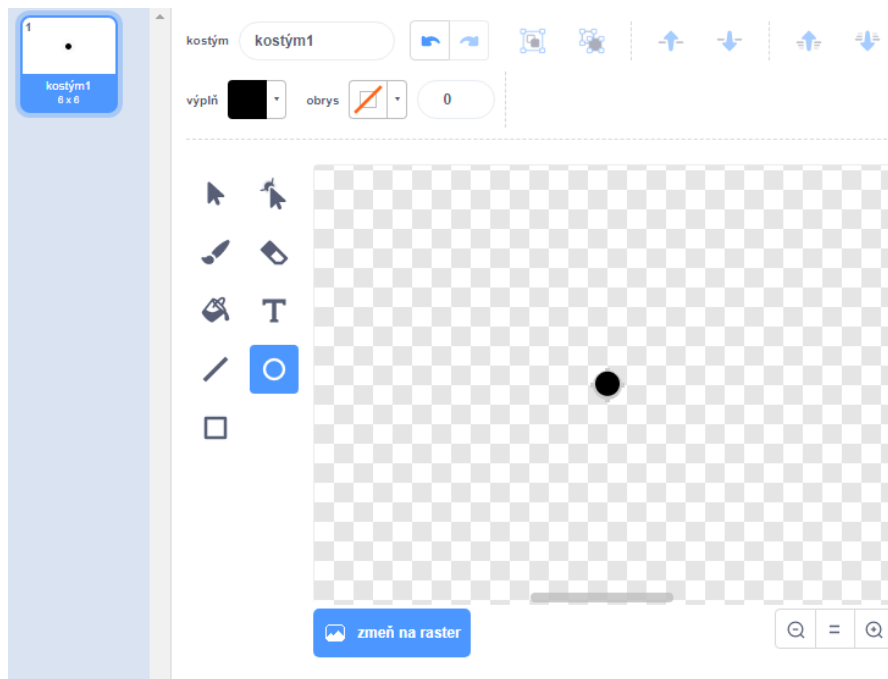
Hodnota premennej nemusí byť nikdy 0, tiež je možné, že žiakom sa ani veľkým hlukom nepodarí dosiahnuť hodnota 100.

Šikovnejší žiaci môžu vytvoriť program, v ktorom bude postava hovoriť v bubline hodnoty premennej **hlasitosť**:



Upozornenie: V skupine **Zvuk** existuje premenná s rovnakým menom **hlasitosť**, tá však **nezachytáva** hlasitosť z mikrofónu.

3. Zruš mačku a vytvor novú postavu. Postave nakresli kostým malej čiernej bodky.



Upozorňujme žiakov, aby bodku nakreslili naozaj malú, ideálne ako na obrázku, pretože neskôr pri programovaní môžu nastať problémy pri testovaní, či sa bodka dotýka okraja.

4. Zisti a zapíš do tabuľky, aká je x-ová súradnica ľavého a pravého okraja scény.

ľavý okraj	pravý okraj
-237	236

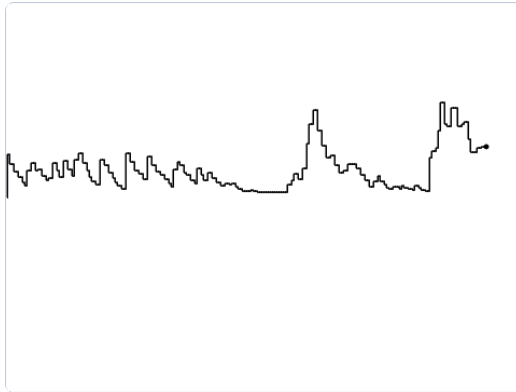
Žiaci by mohli tieto informácie vedieť z minulých hodín. Ak to chceme zistiť, najľahšie to urobíme tak, že postavu presunieme myšou na ľavý okraj a v časti **Pohyb**

pozrieme blok **skoč na x: -237**. Podobne ju presunieme na pravý okraj a pozrieme si číslo **skoč na x: 236** v bloku.

Žiaci nemusia odpozorovať rovnaké hodnoty ako máme v blokoch na obrázku, ale ak majú kostým bodky dostatočne malé, mali by dokázať posunúť postavu tak, aby ich hodnoty boli podobné.

5. Vytvor program, v ktorom budeš stále presúvať postavu na y-ovú súradnicu podľa aktuálnej hodnoty premennej **hlasitosť**. Súradnicu x meň o malé hodnoty, napr. o 1 alebo 2.

Vyskúšaj projekt, rob hluk tak, aby ti vyšiel podobný obrázok:



```

pri kliknutí na
skoč na x: -238 y: 0
zmaž
nastav farbu pera na
pero zapni
opakuj stále
  skoč na y: hlasitosť
  zmeň x o 2

```

V závislosti od toho, ako jemne vníma mikrofón zvuk, môžu byť hodnoty rôzne. V niektorých prípadoch treba prenásobiť hodnotu premennej **hlasitosť**, napr. číslom 10, aby vykresľované hodnoty neboli príliš mále.

6. Doprogramuj, aby postava po prejdení na pravý okraj scény zmazala doterajšiu kresbu a začala vykresľovanie znova od ľavého okraja scény.

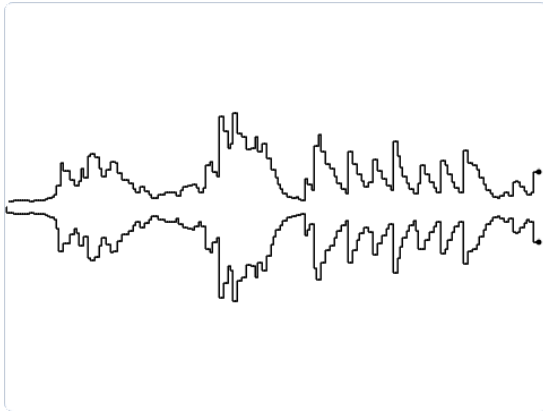
```

pri kliknutí na
skoč na x: -238 y: 0
zmaž
nastav farbu pera na
pero zapni
opakuj stále
  ak < pozícia x > 237
    skoč na x: -240 y: 0
    zmaž
  skoč na y: hlasitosť
  zmeň x o 2

```

V riešení pribudla podmienka, ktorou zistíme, či má postava väčšiu x-ovú súradnicu ako je pravý okraj. Ak áno, zmažeme scénu a presunieme postavu znova na ľavý okraj.

7. Duplikovaním postavy **kresli** vytvor novú postavu. Uprav novej postave program tak, aby kreslila presne to isté, čo postava **kresli**, len zvislo symetricky prevrátené tak, ako vidíš na obrázku:



```
pri kliknutí na [ ]
skoč na x: -238 y: 0
zmaž
nastav farbu pera na [ ]
pero zapni
opakuj stále
ak [ pozícia x > 237 ]
skoč na x: -240 y: 0
zmaž
skoč na y: 0 - hlasitosť
zmeň x o 2
```

Symetriu podľa vodorovnej osi zabezpečíme tým, že nová postava bude skákať na opačné y-ové súradnice.

Žiaľ, v Scratch môžeme opačnú hodnotu premennej získať iba tak, že vytvoríme výraz, v ktorom hodnotu odčítame od 0 (alebo prenásobíme hodnotou -1), t.j.

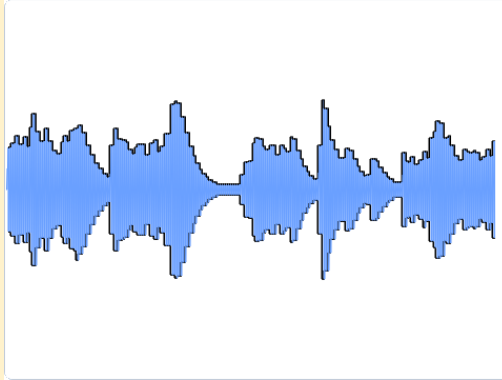
```
skoč na y: 0 - hlasitosť
```

8. Obe postavy ktoré kreslia skry, aby ich nebolo vidieť, keď budú kresliť a vhodne ich premenuj. Napr. *kresli hore* a *kresli dole*.

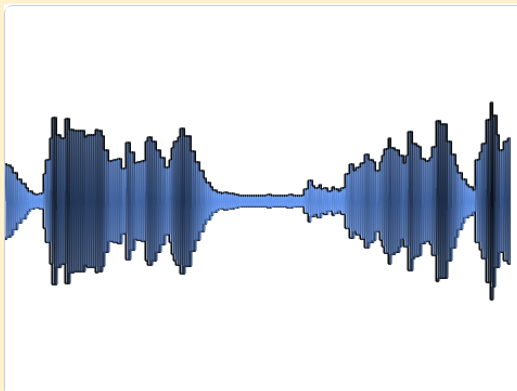
Rozširujúcu úlohy

9. Naprogramuj vyfarbovanie vnútra tak, ako vidíš na obrázku, napr. modrou farbou. Vyfarbovanie nech zabezpečí nová postava, ktorá bude robiť farebné čiary medzi postavami kresli hore a kresli dole.

Pomôcka: Pre novú postavu využij pozície, ktoré majú postavy, ktoré kreslia.



10. Zabezpeč, aby sa farby vykresľovali v závislosti od výšky hlasitosti. Keď je zvuk slabý, bude sa kresliť svetlomodrou a keď je silný, bude sa kresliť tmavomodrou.



Na to, aby sa mohla nová postava presúvať na pozície inej postavy využijeme blok

číslo pozadia ▾ z Scéna ▾

z časti **Zisťovanie**. V ňom sa dá vybrať namiesto Scény

niektorá postava a potom sa dá vybrať, ktorú vlastnosť postavy chceme použiť teda vyberieme:

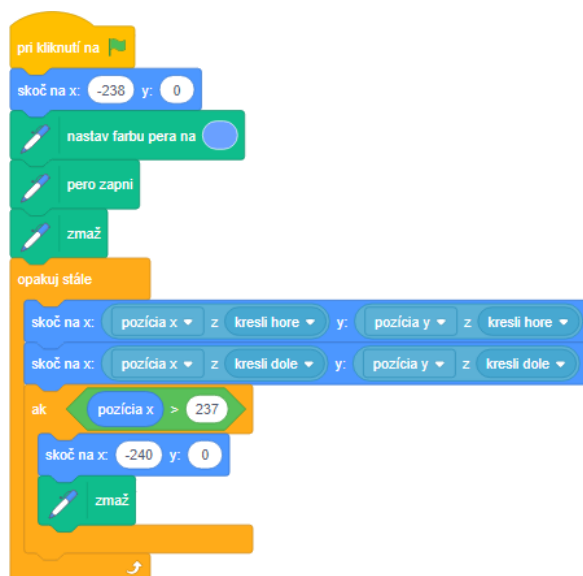
pozícia x ▾ z kresli dole ▾

Kreslenie a vyfarbovanie môže robiť aj jedna postava, ale program sa tým môže stať pre žiakov menej prehľadný.

Žiaci so záujmom o programovanie zrejme dokážu prísť sami na riešenie s jednou postavou. Riešenie úloh nájdete v Scratch projekte **17_Kresli_zvuk_9_a_10**.

Riešenie

9.



10.



Pre žiakov, ktorí **nemajú záujem o programovanie** sme v prezentácii **17_kresli_symetricky** pripravili jednoduchšie zadanie. Riešenie nájde učiteľ v rovnako pomenovanom projekte.

Prprav projekt, v ktorom budeme kresliť jednou postavou (rovnako ako v grafickom editore) a druhá postava bude kresliť symetrickú kresbu vodorovne prevrátenú.



HODNOTENIE – CCA 5 MINÚT

Žiakov hodnotíme podľa funkčnosti ich projektu a správnosti riešenia jednotlivých jeho častí. Je potrebné uvedomiť si, že v takto koncipovaných projektoch je **správnych riešení viacero**. Pri hodnotení môžeme brať ohľad nielen na funkčnosť riešenia, ale aj na kreativitu žiaka.

18 STE(A)M: MATEMATIKA

Tematický celok / Téma	ISCED / Odporúčaný ročník
Algoritmické riešenie problémov / Matematika – tvorba edukačný projektov	ISCED 2 / 8.-9. ročník
Požiadavky na vstupné vedomosti a zručnosti	
Orientácia v prostredí Scratch, vedomosti a zručnosti zo všetkých predchádzajúcich aktivít.	
Ciele	
Žiakom osvojované vedomosti a zručnosti	Žiakom rozvíjané spôsobilosti
Prostredie Scratch <ul style="list-style-type: none"> v rámci tejto aktivity žiaci nezískavajú nové vedomosti a zručnosti, ale učia sa vhodne využívať tie, ktoré nadobudli na predchádzajúcich hodinách. 	Informatické myslenie <ul style="list-style-type: none"> (ALG3) vytvárať vlastné algoritmy na riešenie problému (ALG6) využívať existujúce algoritmy v návrhu vlastných algoritmov (ALG8) zapísať algoritmy v konkrétnom programovacom jazyku Bádateľské spôsobilosti: <ul style="list-style-type: none"> zaznamenávať výsledky pozorovania konštruovať model a manipulovať s ním pomocou softvéru zdieľať a prezentovať výsledky pred spolužiakmi
<p>Cieľom tejto aktivity je navrhnúť projekt, ktorý by pomohol mladším žiakom pri učení sa niektorej oblasti matematiky. V metodike navrhujeme námety na niekoľko takýchto projektov. Úlohou žiaka v tejto aktivite je navrhnúť a naprogramovať komplexný program. To od neho vyžaduje, aby dokázal pripraviť vhodný dizajn a ovládanie programu a naprogramovať správanie jednotlivých postáv.</p> <p>Takto koncipované komplexné úlohy sú pre žiakov pomerne náročné. Učiteľ sa môže rozhodnúť upraviť metodiku napr. tak, že žiakom predvedie len jeden projekt a ten potom budú programovať všetci žiaci.</p>	
Dominantné vyučovacie metódy a formy	Príprava učiteľa a pomôcky
Projektová metóda. Bádateľská metóda. Diskusia.	<ul style="list-style-type: none"> súbory s riešením projektov (pre učiteľa)
Diagnostika splnenia vzdelávacích cieľov	
Spoločné hodnotenie spolužiakmi, posúdenie správnosti navrhnutých postupov a programátorských riešení. Vrstovnícke hodnotenie na základe testovania. Odporúčania na zlepšenie programu. Prezentácia a spoločné hodnotenie vytvorených matematických projektov.	

ÚVOD

Nasledujúca metodika v sérii programovania v prostredí **Scratch** pre žiakov základnej školy je venovaná témam a programovaniu takých projektov, ktoré by mohli byť užitočné pre **vyučovanie matematiky**. Počas programovania predchádzajúcich projektov si žiaci zrejme uvedomili, že sa len ťažko zaobídu bez matematických výpočtov. Najčastejšie potrebovali zistiť súradnice postáv, alebo uhol, ktorým sa má postava otočiť, aby nakreslila požadovaný N-uholník. Na matematike sa zrejme stretli s usporadúvaním čísel, pracovali s lineárnymi funkciami, zisťovali násobky a delitele čísel alebo sa venovali rôznym číselným sústavám. Informatika by sa bez matematiky určite nezaobišla. A naopak, veríme, že matematike môže pomôcť, keď využije možnosti informatiky a programovania.

Pre túto metodiku sme pripravili niekoľko tém, ktoré by žiaci mohli spracovať vo svojich projektoch. Veríme, že po ich prezretí aj samotní žiaci prídu so svojimi vlastnými námetmi na podobné projekty.

ZAPOJENIE A SKÚMANIE – CCA 5 MIN.

Aktivita 1

*Učiteľ ukáže na dataprojektore matematický projekt. Vyberie si taký, ktorý uzná za vhodný pre svojich žiakov tak, aby bol pre nich **primerane náročný**.*

Riešenia jednotlivých projektov nájde učiteľ v **Scratch** projektoch. Pred hodinou by si ich mal prezrieť a **rozhodnúť sa**, ktoré z nich ukáže svojim žiakom. Zrejme bude zvažovať hlavne to, aby ukážka **náročného projektu žiakov neodradila** od programovania.

Žiaci diskutujú o tom, aké postavy a funkčnosť projekt obsahuje. Učiteľ môže oslovovať niektorých žiakov a pýtať sa ich aj na tému matematiky, ktorej sa projekt týka.

ROZPRACOVANIE – CCA 35 MIN.

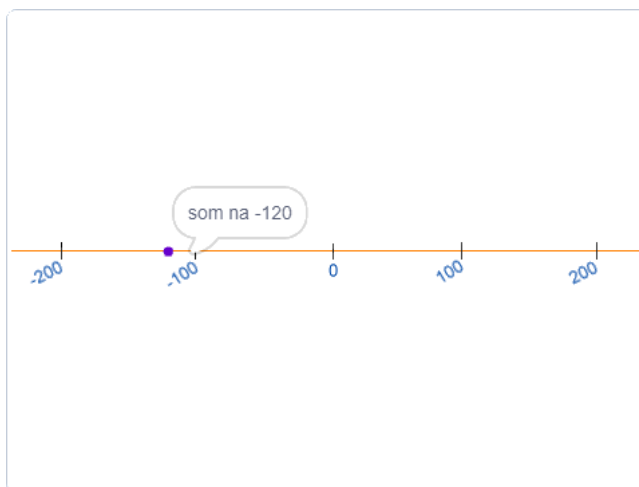
Vyzveme žiakov, aby každý naprogramoval podobný projekt, resp. navrhol vlastný projekt. Učiteľ **môže upraviť** metodiku tak, že **ukáže viacero projektov**, ale stále musí mať na zreteli, že náročné projekty môžu odradiť žiakov od programovania.

Viacere projekty si od návrhu po realizáciu vyžadujú pomerne **veľa času** a tak je pravdepodobné, že žiaci budú svoje projekty dokončovať doma.

V ďalšej časti metodiky predstavujeme projekty a uvádzame návody a čiastočné riešenia niektorých projektov. Projekty uvádzame v poradí **od jednoduchších po zložitejšie** a náročnejšie. Posledný projekt si dokonca vyžaduje oboznámenie sa žiakov s údajovou štruktúrou **zoznam** (list). Veríme, že niektorí žiaci budú mať o programovanie taký záujem, že budú ochotní a schopní naučiť sa pracovať s touto údajovou štruktúrou.

ODHAD NA ČÍSELNEJ OSI

Úlohu sme čerpali z obsahu predmetu matematika v 5 ročníku základnej školy.

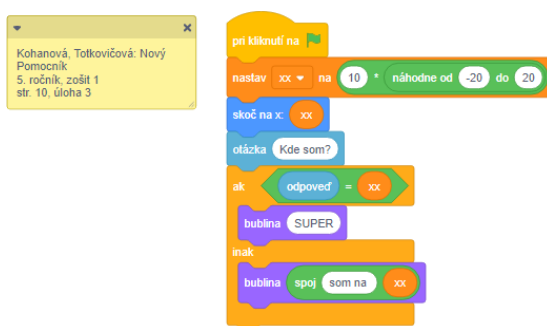


Projekt Odhad na číselnej osi

Riešenie

Projekt obsahuje iba pozadie scény s číselnou osou a jednu postavu, ktorá skočí na náhodné miesto na číselnej osi (násobok 10). Úlohou hráča je **odhadnúť a zapísať** súradnicu, na ktorej sa postava nachádza. Celé riešenie vidíme na obrázku a nájdeme ho v Scratch projekte **odhad_na_ciselnej_osi**.

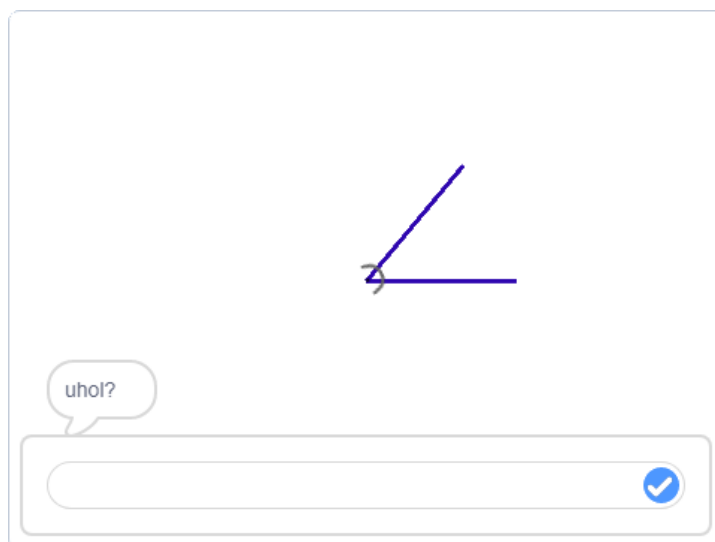
Žiakom je vhodné dať k dispozícii obrázok číselnej osi – **cis_os.png**, ktorý sa dá priamo načítať ako pozadie scény.



Celé riešenie projektu odhad na číselnej osi

ODHAD UHLA

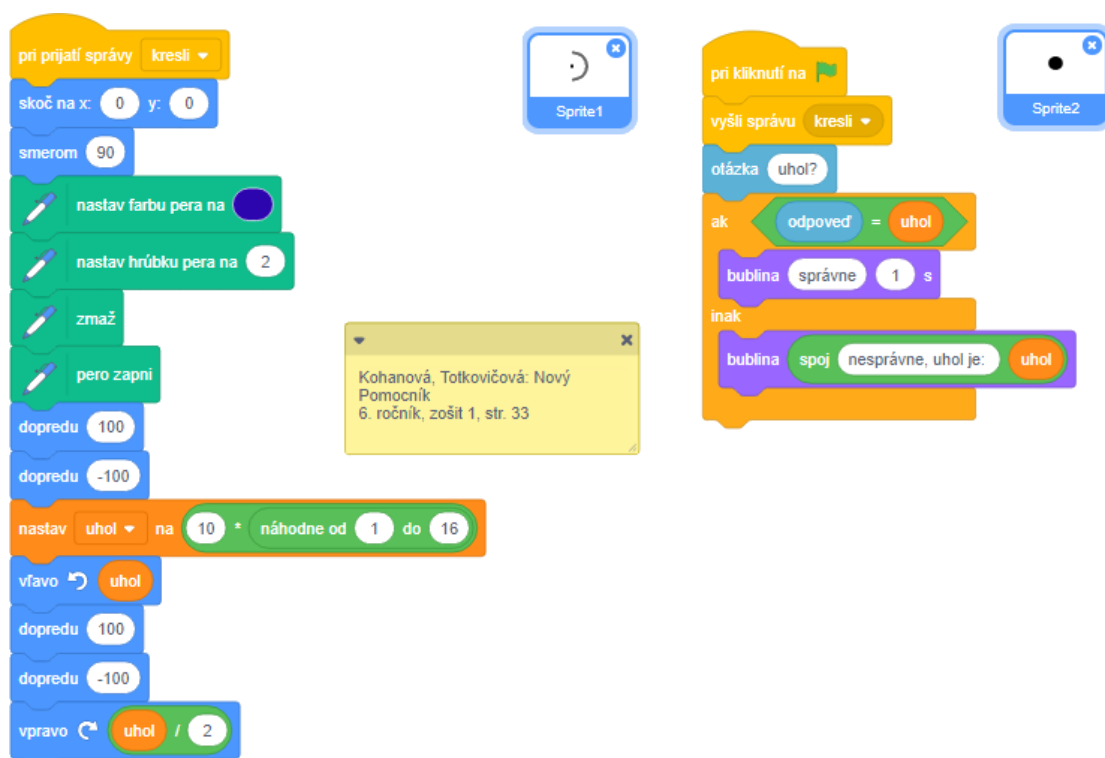
S témou uhol sa žiaci stretávajú na matematike už v 6. ročníku. Program bude generovať a kresliť rôzne veľké uhly. Úlohou hráča je **odhadnúť a zapísať** veľkosť nakresleného uhla. Ukážkový projekt generuje len uhly, ktoré sú násobkom 10 a sú menšie ako priamy uhol.



Projekt Odhad uhla

Riešenie

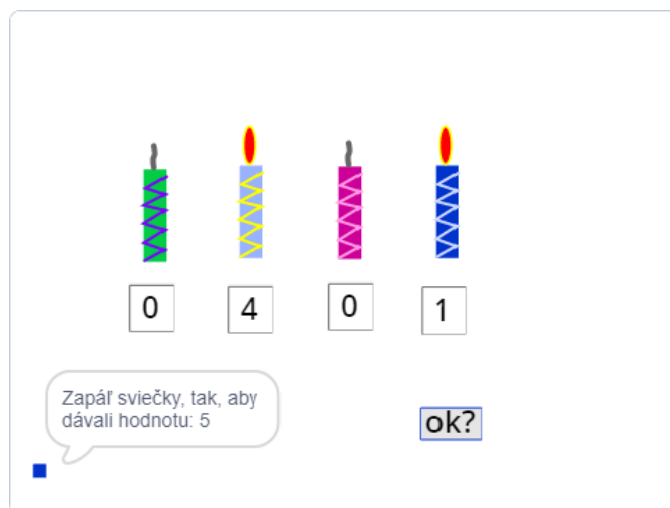
Projekt obsahuje dve postavy. Jedna kreslí uhol a druhá zobrazuje otázku pre hráča. Riešenie je v Scratch projekte **odhad_uhla**.



Riešenie projektu Odhad uhla

DVOJKOVÁ SÚSTAVA

Program bude zapálenou, resp. zhasnutou, sviečkou na príslušnom mieste simulovať dvojkový zápis čísla. Počítač vygeneruje náhodnú hodnotu a úlohou žiaka je kliknutím zapáliť sviečky tak, aby zobrazovali dvojkový zápis daného čísla.



Dvojková sústava trochu inak

Riešenie

Úloha obsahuje 4 postavy – sviečky. Každá má dva kostýmy – zhasnutú a zapálenú sviečku. Kartačky pod sviečkami s číslami sú tiež postavy, ktoré tiež reagujú, keď hráč klikne na "ich" sviečku. Pri kontrole správnosti žiackeho riešenia sme využili to, že jeden kostým kartačky má meno 0 a druhý kostým má také meno, aké číslo je na kartačke.

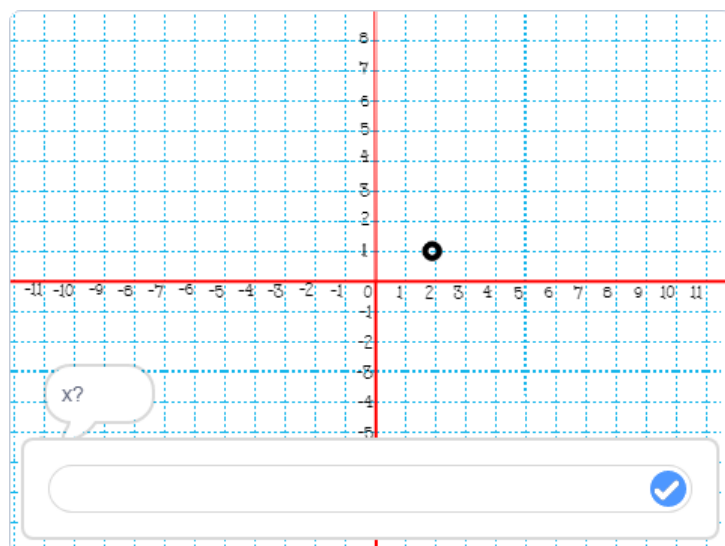


Žiaci môžu navrhnúť rôzne úpravy programu: pridať ďalšie postavy, aby bolo vidieť naozajstný dvojkový zápis čísla, vložiť medzi kartačky znak +, aby bolo zrejmé, že je potrebné hodnoty sčítať, vložiť kartačku s výsledkom...

Celé riešenie je už dlhšie a učiteľ ho nájde v Scratch projekte **dvojkova_sustava**.

ZISTI, KDE SOM V SÚRADNICOVEJ SÚSTAVE

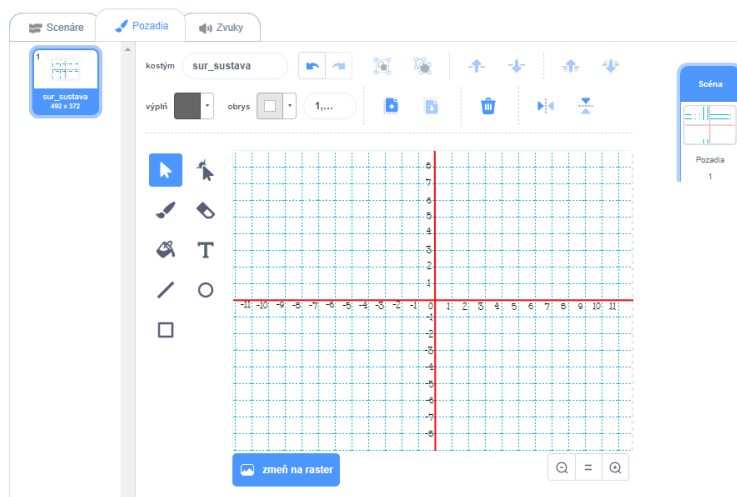
Súradnicová sústava sa využíva ako v matematike, tak aj v informatike. Je preto pomerne dôležité, aby sa v nej žiaci dokázali orientovať. Projekt presúva objekt – čiernu bodku, na náhodné miesto. Úlohou hráča je zadať jeho x-ovú a y-ovú súradnicu.



Projekt Zisti, kde som a zadávanie x-ovej súradnice

Riešenie

Vzorové riešenie sme naprogramovali tak, aby žiaci nemuseli násobiť vygenerované súradnice, ale aby sa hľadaný bod dostal na miesto tým, že bude niekoľkokrát meniť svoju x-ovú a y-ovú súradnicu o kladné hodnoty. Pri hľadaní riešenia pre umiestnenie bodu bude zrejme najdôležitejšie práve to, ako zabezpečiť, aby sa postava náhodným generovaním nedostala mimo zobrazenej súradnicovej sústavy. Pre riešenie je vhodné dať žiakom k dispozícii obrázok súradnicovej sústavy – súbor **sur_sustava.svg**, ktorý potom priamo načítajú ako pozadie scény.



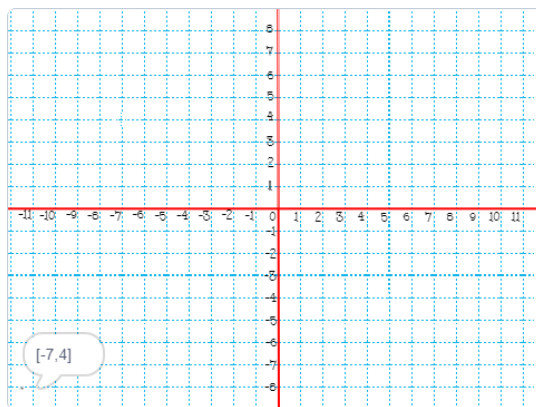
Obrázok pozadia pre scénu

Scratch projekt s riešením je v súbore **zisti_kde_som**.

KDE JE BOD V SÚRADNICOVEJ SÚSTAVE

Projekt vyžaduje od používateľa, aby podľa zobrazených súradníc klikol do súradnicovej sústavy na miesto, kde sa nachádza bod, ktorý vygeneroval počítač.

Je vhodné dať žiakom obrázok súradnicovej sústavy v súbore **sur_sustava.svg**, ktorý sa dá načítať ako pozadie scény. Tiež budú potrebovať obrázok, ktorý zobrazuje „skrytý“ bod, na ktorý sa dá kliknúť. Využili sme, že tento bod „vyzerá“ ako keby bol súčasťou štvorčekovej siete, je v súbore **p2.sprite3**.



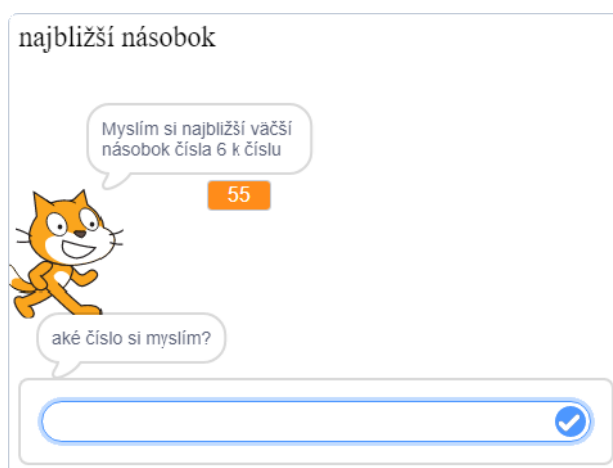
Na mieste [-7, 4] sa nachádza postava, ktorú vlastne nevidieť, ale na ktorú sa dá kliknúť

Riešenie

Program s riešením nie je rozsiahly, ale vyžaduje si využitie **viacerých náročnejších konceptov** – umiestnenie bodu na správne miesto na scéne, prácu s premennými, kontrolu správnosti. Učiteľ ho nájde v Scratch projekte **klikni_kde_je_bod**.

NAJBLIŽŠÍ VÄČŠÍ NÁSOBOK

Násobky čísel žiaci uplatnia pri vyhľadávaní najmenšieho spoločného násobku, podobné úlohy sa nachádzajú v obsahu matematiky pre 6. ročník základnej školy.



Projekt Zisti najbližší väčší násobok

Riešenie

Projekt obsahuje viacero postáv, z ktorých väčšina sa využíva pri zobrazení vysvetlenia a matematickej úvahy, ktorú sme našli v literatúre.

najbližší násobok

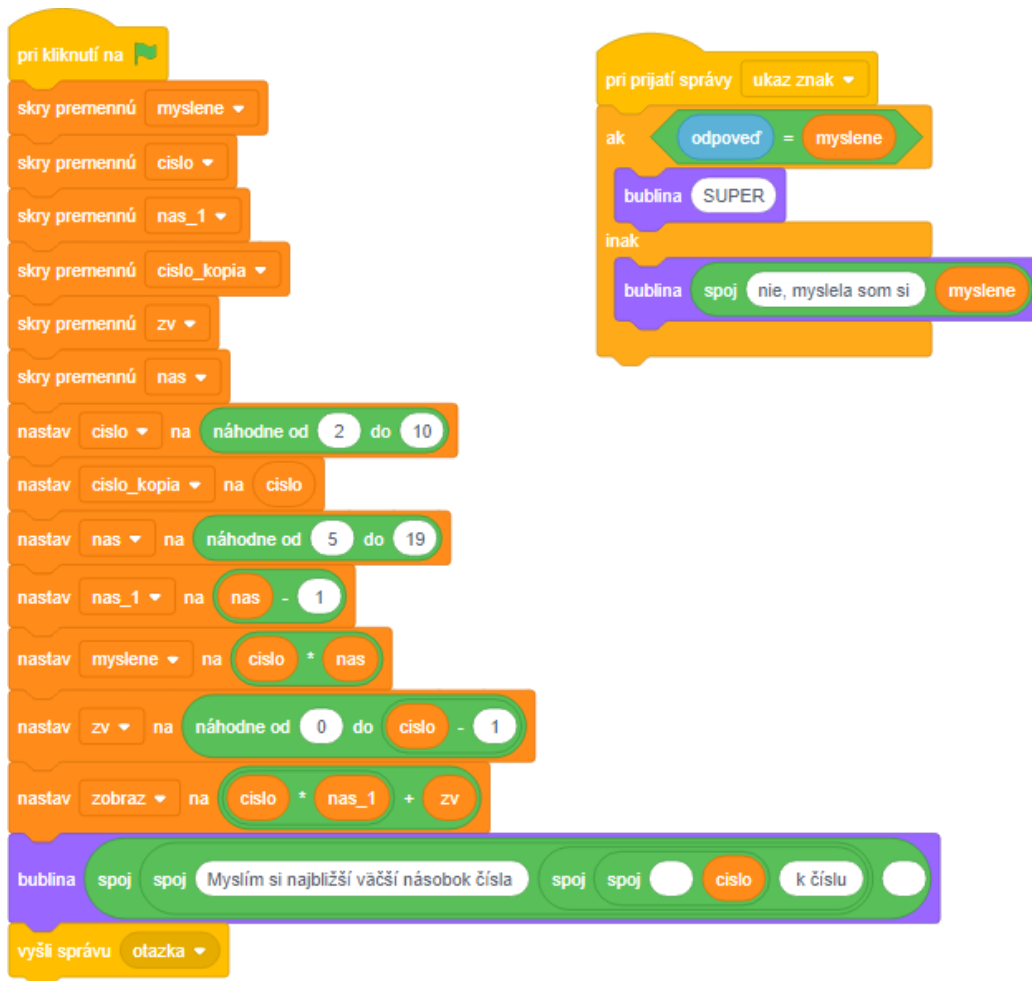
$6 \cdot 10 = 60$
 $55 < 60$
 $6 \cdot 9 + 1 = 55$

Postava: Sprite2, x: -224, y: -109
 Ukáž: Veľkosť: 100, Smer: 90
 Scéna: Pozadia 1

Zobrazenie vysvetlenia pri správnej odpovedi a postavy, ktoré sú v projekte

Podstatnú časť riešenia zabezpečuje základná postava, mačka. Počítač náhodne vyberie číslo a násobok, ktorý si mačka „myslí“. Potom mačka povie v bubline náhodné číslo, ktoré je **menšie** ako vygenerovaný násobok. Druhá postava zabezpečí, že počítač čaká na zadanie odpovede od hráča a postava mačka skontroluje správnosť zadanej odpovede.

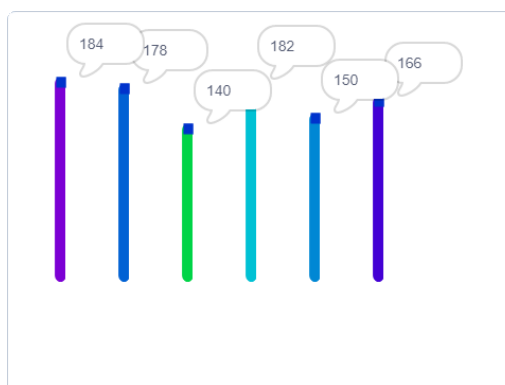
Celé riešenie je v Scratch projekte **najblizsi_nasobok**.



Riešenie časti programu, v ktorom sa generujú a zobrazujú čísla

HÁDZANIE KOCKOU

Téma sa nachádza v matematike pre 8. ročník. Žiaci môžu spracovať program, v ktorom bude počítač simulovať hádzanie jednou alebo dvoma kockami a zobrazovať, koľkokrát ktoré číslo „padlo“. Napr. na našom obrázku vidím , že po 1000 hodoch padlo číslo 1 184-krát, ale číslo 6 „len“ 166-krát. Počas vysvetľovania tejto simulácie môžeme so žiakmi diskutovať o tom, nakoľko je generátor náhodných čísel v **Scratch** „spravodlivý“.



Zobrazenie koľkokrát padli jednotlivé čísla pri 1000 hodoch jednou kockou.

Riešenie

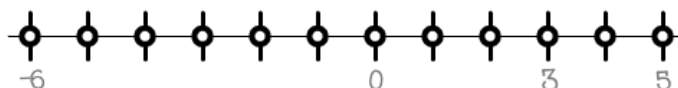
V jednoduchšej verzii môžeme využiť 6 rôznych premenných, ktorým budeme zvyšovať hodnoty pri padnutí príslušného čísla na kocke. Nakoniec môžeme na zobrazenie počtu využiť grafické znázornenie hodnôt pomocou čiar, podľa toho, ktoré číslo padlo.

Celé riešenie je v Scratch projekte **hadzanie_kockou**.

UMIESTNI ČÍSLA NA ČÍSELNÚ OS

Program môžu žiaci uplatniť v 8. ročníku pri opakovaní toho, ako vyzerá číselná os a kam patria jednotlivé čísla. Používateľ dostáva čísla a má kliknúť na miesto na číselnej osi, kam číslo patrí.

klikni na číselnú os, kam patrí číslo -4



Umiestni číslo na číselnú os

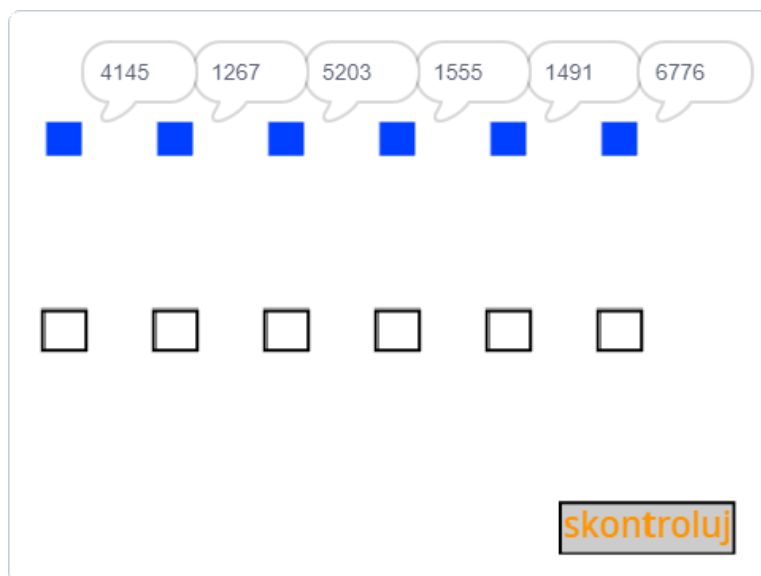
Riešenie

Čísla pod číselnou osou môžu byť pripravené ako jednotlivé postavy, ktoré sa pri kliknutí na správny bod na číselnej osi ukážu.

Zložitejšie riešenie môže zahŕňať aj samotné generovanie číselnej osi v nejakom (vždy inom rozsahu) a to, aby sa program nepýtal používateľa na žiadne číslo dvakrát – takéto riešenie je v Scratch projekte **umiestni_cisla_na_ciselnu_os**.

USPORIADAJ VZOSTUPNE

Téma usporiadania čísel je súčasťou matematiky v 6. ročníku základnej školy. Žiaci usporiadávajú **čísla nad tisíc** vzostupne alebo zostupne. Počítač – projekt, ktorý pripravíme – môže fungovať ako náhodný generátor čísel na usporiadanie.



Usporiadanie čísel vzostupne

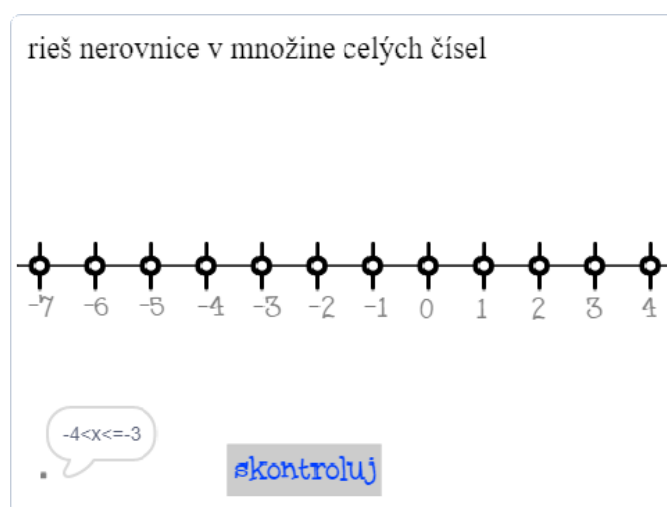
Riešenie

Funkčnosť programu zabezpečuje 6 rovnakých postáv, ktoré po štarte "povedia" náhodné číslo a pri kliknutí sa presunú do dolnej časti na miesto v postupnosti, ktoré „je v poradí“. Pomerne zložitý, hlavne čo sa týka logiky, je scenár pre postavu, ktorá kontroluje správnosť riešenia.

Riešenie je v Scratch projekte **usporiadaj vzostupne**. Veľmi podobný by bol program, ktorý by umožňoval usporiadať čísla zostupne.

RIEŠENIE NEROVNÍC

Téma nerovnice a ich riešenia sa preberá v 7. ročníku základnej školy. Program môže byť pripravený tak, že sa vždy zobrazuje rovnaká číselná os a následne sa generujú vhodné čísla pre nerovnicu.



Riešenie nerovnic

Riešenie

V tomto projekte sa žiak môže rozhodnúť, či budú hodnoty na osi vždy rovnaké, alebo či si zvolí náročnejšie riešenie, v ktorom sa bude meniť rozsah na číselnej osi. Pomerne náročnou časťou programu je generovanie nerovnice a následná kontrola správnosti riešenia.

Náročnejšie riešenie, v ktorom sa zobrazuje vždy iná časť číselnej osi je v Scratch projekte [ries_nerovnice_v_mnozine_celych_cisel](#).

URČ LINEÁRNU FUNKCIU

Lineárna funkcia patrí medzi témy matematiky na základnej škole. Počítač vygeneruje hodnoty pre x a vypočíta hodnoty y . Úlohou hráča je určiť hodnoty a , b , tým že vhodne nastaví posúvače pri menách týchto premenných. Počítač skontroluje, či hráč určil správne funkciu.

nájdí lineárnu funkciu
ťahanim posúvačov nastav
hodnoty pre a, b

$y = ax + b$

a 0

b 0

skontroluj

x hodnoty	y hodnoty
1 -27	1 85
2 -21	2 67
3 -17	3 55
4 -12	4 40
5 -6	5 22
6 -3	6 13
7 -1	7 7
8 3	8 -5
9 6	9 -14
10 8	10 -20
11 12	11 -32
12 19	12 -53
13 26	13 -74

+ dĺžka 15 = + dĺžka 15 =

Určovanie lineárnej funkcie podľa konkrétnych hodnôt

Riešenie

Program vygeneruje hodnoty pre koeficienty lineárnej funkcie a zobrazí niekoľko náhodných hodnôt pre x , ktoré sú usporiadané vzostupne.

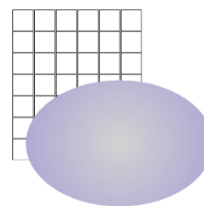
Pri riešení tejto úlohy nie je nevyhnutné použiť údajovú štruktúru zoznam (list), jednotlivé hodnoty by mohli byť samostatné premenné. Je tiež pravda, že z matematického hľadiska by stačilo vygenerovať hodnoty funkcie pre dve hodnoty x , ale pri viacerých vygenerovaných hodnotách si hráč môže overiť svoju hypotézu, resp. môže si vybrať také hodnoty, pre ktoré je podľa neho najjednoduchšie určiť, ako vyzerá funkcia. Práca

s posúvačmi je priamo súčasťou prostredia Scratch. Postava **skontroluj** zistí, či sa zvolené hodnoty pre **a** a **b** rovnajú tým, ktoré vygeneroval počítač.

Celé riešenie je v Scratch projekte **urc_linearnu_funkciu**.

Pre žiakov, ktorí **nemajú záujem o programovanie** sme v prezentácii **18_kolko_stvorcov** pripravili jednoduchšie zadanie. Riešenie nájde učiteľ v rovnako pomenovanom projekte.

Priprav projekt, ktorý umožní žiakom vypočítavať, koľko štvorcov sa nakreslilo. Na hracej ploche sa vykreslí niekoľko radov štvorcov. Časť z nich je zakrytá oválom. Úlohou žiaka je uviesť, koľko štvorcov sa nakreslilo. Žiak pritom vidí počet riadkov a počet štvorcov v prvom riadku.



Po zadaní výsledku počítač skontroluje, či žiak uviedol správny počet štvorcov. Žiak zrejme počet štvorcov vypočíta vynásobením počtu riadkov a počtu stĺpcov. Ak žiak uviedol správny počet štvorcov, program v bubline vypíše pochvalnú správu.

HODNOTENIE – CCA 5 MIN.

Keďže sa žiaci mohli sami rozhodnúť, resp. rozhodol sa učiteľ, ktorý projekt (z ukázkových alebo svoj vlastný) budú programovať, môžeme vyzvať viacerých žiakov, aby svoje programy prezentovali pomocou dataprojektora. Môžeme tiež vytvoriť **Scratch Štúdio**, v ktorom žiaci zverejnia svoje projekty a môžu komentovať projekty spolužiakov. Takýto spôsob **vzájomného hodnotenia** môže byť pre žiakov atraktívny a môže ich motivovať dokončiť projekt aj doma.

Žiakov hodnotíme podľa funkčnosti ich projektu a správnosti riešenia jednotlivých jeho častí. Je potrebné uvedomiť si, že v týchto projektoch je **správnych riešení viacero**. Pri hodnotení môžeme brať ohľad nielen na funkčnosť riešenia, ale aj na kreativitu žiaka.

19 STE(A)M ANIMUJEME CHEMICKÉ REAKCIE

Tematický celok / Téma	ISCED / Odporúčaný ročník
Algoritmické riešenie problémov / Animujeme chemické reakcie	ISCED 2 / 8. - 9. ročník
Požiadavky na vstupné vedomosti a zručnosti	
<ul style="list-style-type: none"> • Orientácia v prostredí Scratch, vedomosti a zručnosti zo všetkých predchádzajúcich aktivít. • Vedomosti a zručnosti z chémie <ul style="list-style-type: none"> ○ žiaci poznajú význam chemických skratiek (NaOH, HCl, NaCl a H₂O, KOH, HBr) a vedia, ako vyzerajú ich molekuly ○ vedia vytvárať a identifikovať vzorce solí, kyselín a hydroxidov ○ dokážu vysvetliť princíp neutralizácie a aplikovať ho na jednoduchom príklade 	
Ciele	
Žiakom osvojované vedomosti a zručnosti	Žiakom rozvíjané spôsobilosti
Prostredie Scratch <ul style="list-style-type: none"> • v rámci tejto aktivity žiaci nezískavajú nové vedomosti a zručnosti, ale učia sa vhodne využívať tie, ktoré nadobudli na predchádzajúcich hodinách. 	Informatické myslenie <ul style="list-style-type: none"> • (ALG3) vytvárať vlastné algoritmy na riešenie problému • (ALG6) využívať existujúce algoritmy v návrhu vlastných algoritmov • (ALG8) zapísať algoritmy v konkrétnom programovacom jazyku
Dominantné vyučovacie metódy a formy	
Bádateľská metóda (5E). Projektová metóda – medzipredmetový projekt. Frontálna forma.	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> • obrázky molekúl pre žiakov molekula1.sprite3 až molekula4.sprite3 • pre učiteľa hotové Scratch projekty 19_Anim_chem_reakcie, 19_Anim_KOH_HBr • úloha na opakovanie v prezentácii 19_chem_prvok_a_znacka a jej riešenie v projekte s rovnakým názvom 	
Diagnostika splnenia vzdelávacích cieľov	
Žiacke odpovede vo fáze Vysvetlenie . Pozorovanie činnosti žiakov vo fáze Skúmanie a Rozpracovanie . Vrstovnícke hodnotenie hotových projektov podľa ich vzhľadu a funkčnosti.	

ÚVOD

Nasledujúca metodika poskytuje učiteľovi možnosť prebrať, resp. precvičovať so žiakmi tému z chémie **neutralizačná reakcia**. Žiaci v tejto hodine využijú svoje vedomosti z chémie na to, aby simuláciu **naprogramovali** v prostredí Scratch na hodine informatiky.

Túto metodiku nám pomáhala vypracovať študentka chémie, ktorá vytvorila aj obrázky v online softvéri molview.org.

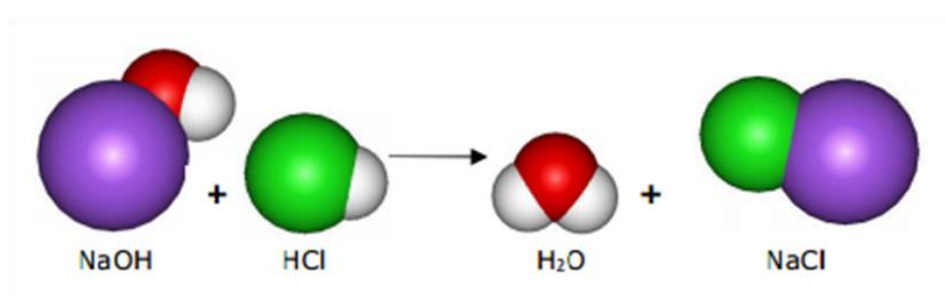
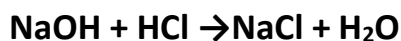
ZAPOJENIE – CCA 5 MIN.

Na začiatku hodiny (je vhodné, ak môže byť pritom aj učiteľ chémie) sa môžeme so žiakmi porozprávať o tom, kde v tele človeka sa môžu stretnúť s neutralizačnými procesmi. Žiaci by si mohli spomenúť na to, ako na hodinách chémie preberali túto informáciu a hovorili o možnosti prekyslení žalúdka, ľudovo „pálenie záhy“, ktoré môžu viacerí poznať od rodičov. V tomto prípade ide vlastne o zvýšenie koncentrácie HCl. Žiaci majú uviesť, či poznajú niečo proti páleniu záhy... napr. vypiť mlieko, alebo vodu s rozpustenou sódou bikarbónou.

Aktivita 1

V úvode vyučovacej hodiny spustíme cez dataprojektor pripravený projekt **19_Anim_chem_reakcie**. Projekt spúšťame v celostránkovom režime a spolu so žiakmi sledujeme, čo sa v ňom deje. Žiakov môže prekvapiť zameranie projektu – nejde o informatiku, ale o chémiu a chemické reakcie.

Bolo by dobre, kedy sa tejto časti vyučovacej hodiny mohol zúčastniť **učiteľ chémie**, aby dokázal motivovať žiakov a v prípade potreby usmerňoval ich diskusiu o simulácii, ktorú môžu sledovať v programe. Žiaci by v diskusii mali prísť k tomu, že vidia jednoduchú simuláciu neutralizácie – reakciu hydroxidu sodného NaOH a bezkyslíkatej kyseliny chlorovodíkovej HCl, pomer 1:1. Rozpoznanie molekúl, ktoré sú v simulácii nemusí byť pre žiakov jednoduché, preto by im **učiteľ chémie** mohol pripomenúť, kedy a čo sa o tom učili na hodinách chémie.



Projekt 19_Anim_chem_reakcie

Molekuly sa v programe pohybujú náhodne. Ak do seba narazí jedna molekula NaOH a jedna molekula HCl, vznikne jedna molekula vody H₂O a jedna molekula NaCl. Rýchlosť pohybu aj veľkosť molekúl sú v programe určené pevne a ich veľkosti sme sa snažili prispôbiť skutočným veľkostiam týchto molekúl tak, ako sa zvyknú zobrazovať v iných simuláciách, resp. animáciách, či videách.

SKÚMANIE A VYSVETLENIE – CCA 10 MINÚT

Žiaci riešia **úlohu 1** z pracovného listu a majú identifikovať koľko objektov sa nachádza na scéne, aké udalosti majú definované a akým spôsobom je simulovaný náhodný pohyb molekúl. Môžu pracovať samostatne, alebo vo dvojiciach, pričom vypracovávajú odpovede na otázky z pracovného listu.

Úloha 1

Podľa ukážky a diskusie so spolužiakmi k projektu **19_Anim_chem_reakcie** odpovedz na otázky a vypracuj nasledujúce úlohy.

Aký dej simulácia zobrazuje?	Reakcia neutralizácie.
Aké molekuly sú v projekte na začiatku?	Jedna molekula kyseliny chlorovodíkovej a jedna molekula hydroxidu sodného.
Aké molekuly sú v projekte na konci?	Molekuly vzniknuté pri reakcii sú jedna molekula vody a jedna molekula chloridu sodného.
Aké molekuly vznikajú?	Rovnaké, aké zostávajú zobrazené na konci reakcie, teda molekula vody a chloridu sodného.
Chemickú reakciu zapíš vzorcom.	$\text{NaOH} + \text{HCl} \rightarrow \text{NaCl} + \text{H}_2\text{O}$
Je počet molekúl rovnaký?	Áno, ale nastane chemická reakcia a molekuly sa menia na iné.
Keby sme zvýšili počet oboch molekúl na dvojnásobok, dostali by sme rovnaký výsledok?	Z chemického hľadiska áno, len by sa zvýšil počet molekúl.
Ako by mohla vyzeráť časť programu, ktorá zabezpečuje hýbanie molekúl?	opakuj stále vpravo náhodne od -100 do 100 dopredu 20 ak na okraji, odraz sa

Kedy simulácia skončí?	Keď prebehne reakcia a molekuly sa zmenia na iné, tak už sa v podstate nič ďalšie neudeje, len bude pokračovať náhodný pohyb nových molekúl po ploche.
------------------------	--

Vyvoláme niektorú dvojicu, aby prezentovali svoje závery pred triedou. Ostatní žiaci ich odpovede porovnajú so svojimi zisteniami.

Žiaci by mali prísť k záverom, že z programátorského pohľadu sú v simulácii:

- objekty s kostýmom molekúl, ktoré vstupujú do reakcie
- po stretnutí dvoch molekúl tieto molekuly zmiznú a zmenia sa na iné, nové molekuly

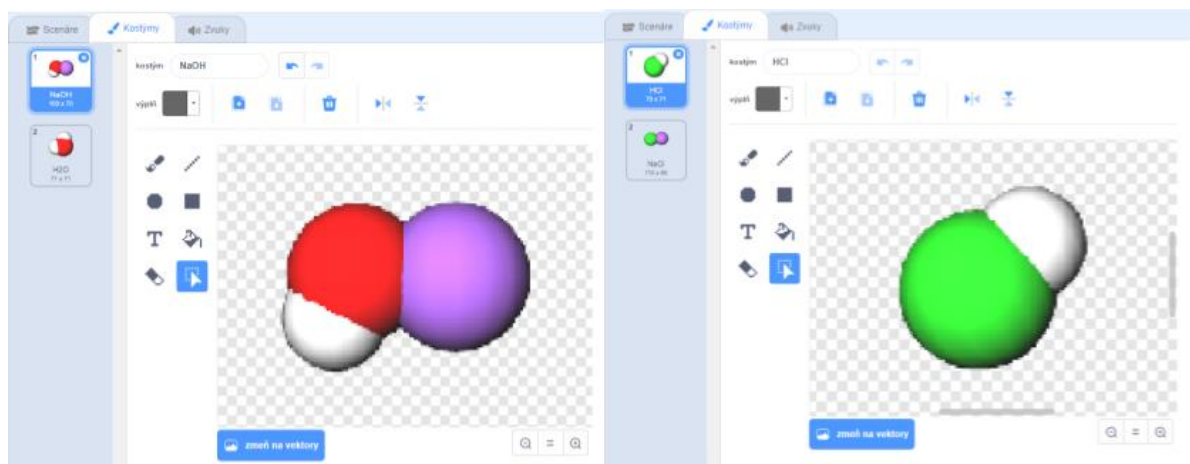
ROZPRACOVANIE – CCA 20 MINÚT

V tejto časti hodiny žiaci riešia úlohy 2 až 4 z pracovného listu.

Úloha 2

Navrhni rovnaký projekt, v ktorom budeš simulovať túto reakciu.

Vytvoríme 4 postavy, teda molekuly, napr. **Molekula1** bude mať kostým NaOH, **Molekula2** kostým H₂O, **Molekula3** bude mať HCl a **Molekula4** kostým NaCl.



Obrázky dvoch molekúl. Všetky obrázky dostane žiak ako samostatné súbory typu sprite3.

Každej z molekúl nastavíme náhodný pohyb po ploche pri udalosti pri kliknutí na vlajku.

V tejto časti je dobré, aby si žiaci nechali vlastnú dĺžku kroku. Časom uvidia, či si zvolili príliš dlhý krok, alebo krátky. Hodnotu kroku musia zistiť experimentovaní a meniť ju, keď budú testovať stretnutie dvoch molekúl.

Pre molekuly NaOH a HCl určíme, aby sa ukázali. Pre molekuly H₂O a NaCl, aby sa na začiatku simulácie skryli.

```

pri kliknutí na
opakuj stále
  čakaj 0.1 s
  vpravo náhodne od -100 do 100
  dopredu 20
  ak na okraji, odraz sa

```

Následne doplníme podmienku, ktorá bude testovať, či sa molekuly, ktoré vstupujú do chemickej reakcie dotýkajú. V prípade ak sa dotknú, tieto molekuly zaniknú (skryjú sa) a na ich mieste sa objavia molekuly, ktoré sú výsledkom chemickej reakcie. Do programuj je dobré doplniť aj nastavenia, ktoré zabezpečia, že molekuly budú začínať svoj pohyb v dostatočnej vzdialenosti od seba.

```

pri kliknutí na
skoč na x: 180 y: 100
ukáž sa
opakuj stále
  čakaj 0.1 s
  vpravo náhodne od -100 do 100
  dopredu 20
  ak na okraji, odraz sa
  ak dotýkaš sa Molekula3 ?
    vyšli správu reakcia
    skry sa

```

```

pri prijatí správy reakcia
skoč na x: pozícia x z Molekula1 y: pozícia y z Molekula1
ukáž sa
opakuj stále
  čakaj 0.1 s
  vpravo náhodne od -100 do 100
  dopredu 20
  ak na okraji, odraz sa

```

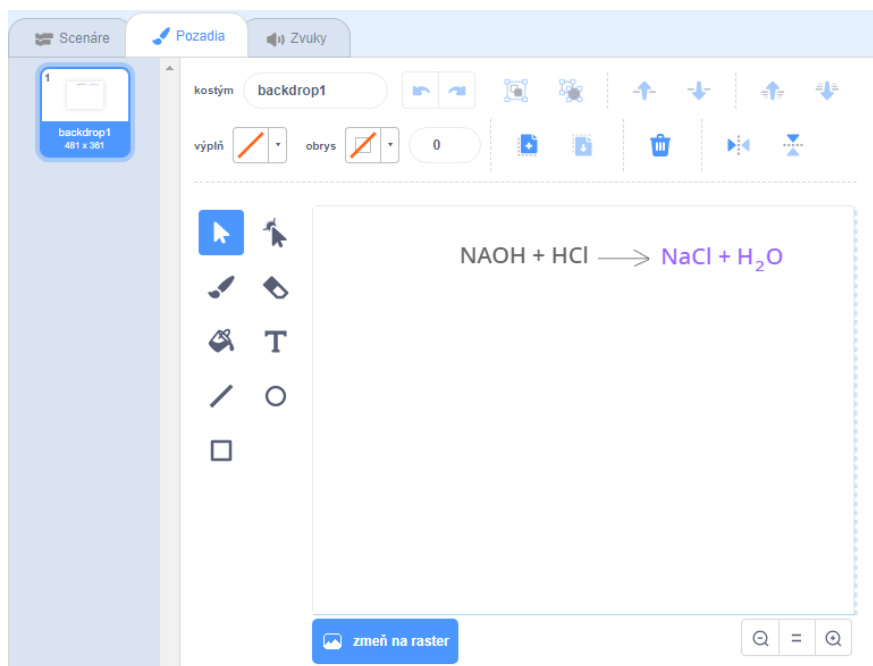
```

pri kliknutí na
skry sa

```

Riešenie úlohy 2 pre postavy Molekula1 a Molekula2.

Na záver môžeme pre **Scénu** vytvoriť **Pozadie**, ktoré bude zobrazovať aj zápis chemickej reakcie.



Zapísanie reakcie do pozadia Scény.

Úloha 3

Vytvor podobnú simuláciu pre reakciu **KOH + HBr**.

Učiteľ sa môže rozhodnúť, či nechá žiakov riešiť **úlohu 3** alebo ich hneď nechá pripravovať svoju vlastnú chemickú reakciu.

Riešenie je rovnaké, ako v **úlohe 2**, len s inými molekulami. Obrázky molekúl sú v jednotlivých .sprite3 súboroch.

Úloha 4

Navrhni a naprogramuj svoj vlastný projekt, v ktorom budeš simulovať nejakú chemickú reakciu.

Vyber si takú reakciu, ktorú poznáš z chémie. Navrhni dizajn a funkčnosť programu.

Obráť sa **na učiteľa chémie** a uisti sa, že tvoj program je navrhnutý správne a chemická reakcia aj v skutočnosti prebieha tak, ako ju chceš simulovať vo svojom programe.

Riešenie **úlohy 4** neuvádzame, zrejme bude mať každý žiak naprogramovanú inú chemickú reakciu.

HODNOTENIE – CCA 5 MINÚT

Môžeme vytvoriť **Scratch Štúdio**, v ktorom žiaci zverejnia svoje animácie rôznych chemických reakcií. Ostatní žiaci môžu komentovať riešenia spolužiakov. Takýto spôsob **vzájomného hodnotenia** môže byť pre žiakov atraktívny a môže ich motivovať dokončiť projekt aj doma.

Žiakov hodnotíme podľa funkčnosti ich projektov a správnosti riešenia jednotlivých jeho častí. Pri hodnotení môžeme brať ohľad nielen na funkčnosť riešenia, ale aj na kreativitu žiaka pri navrhovaní rôznych rozšírení programu.

Žiak môže ohodnotiť svoju prácu na hodine pomocou samohodnotiacej karty (zahŕňa len ciele, ktoré dosiahnu pri riešení základných úloh).

Označ X čo už vieš

- vytvoriť postavu z existujúceho súboru s príponou **.sprite3**
- umiestniť postavy na scéne tak, aby boli dosť ďaleko od seba
- pripraviť nekonečný pohyb postáv
- zistiť, či sa postavy stretli a ukázať iné postavy
- poslať správu inej postave, aby sa skryla, resp. ukázala

Učiteľ môže dať žiakom vyriešiť úlohu **Chemické prvky a ich značky** (na začiatku ďalšej hodiny alebo v rámci opakovania, či utvrdzovania učiva). Pomocou takto pripravenej úlohy zistí, či žiaci dokážu riešiť úlohy, v ktorých využijú doteraz preberané koncepty, či pochopili posielanie a prijímanie správ medzi postavami. Zadanie je pripravené ako prezerateľná prezentácia v súbore **19_chem_prvok_a_znacka**, v ktorej sa zobrazujú aj **animácie**, ako by mal projekt fungovať, riešenie úlohy je v príslušnom Scratch projekte.

Chemický prvok a jeho značka

*Priprav projekt, v ktorom sa bude nachádzať kartička s **názvom** prvku a niekoľko chemických **značiek** rôznych prvkov. Kartička s názvom prvku sa stále pohybuje smerom doprava. Hráč ju môže posúvať pomocou šípok vo všetkých štyroch smeroch.*

Úlohou hráča je, aby sa kartička s názvom dotkla svojej značky prvku. Vtedy hra zastane a kartička sa zmení na zelenú. Ak sa kartička s názvom dotkne inej značky prvku ako svojej, zmení sa na červenú a hra skončí. Takto sa žiak učí, akú chemickú značku má uvedený prvok.

kyslík

N

O

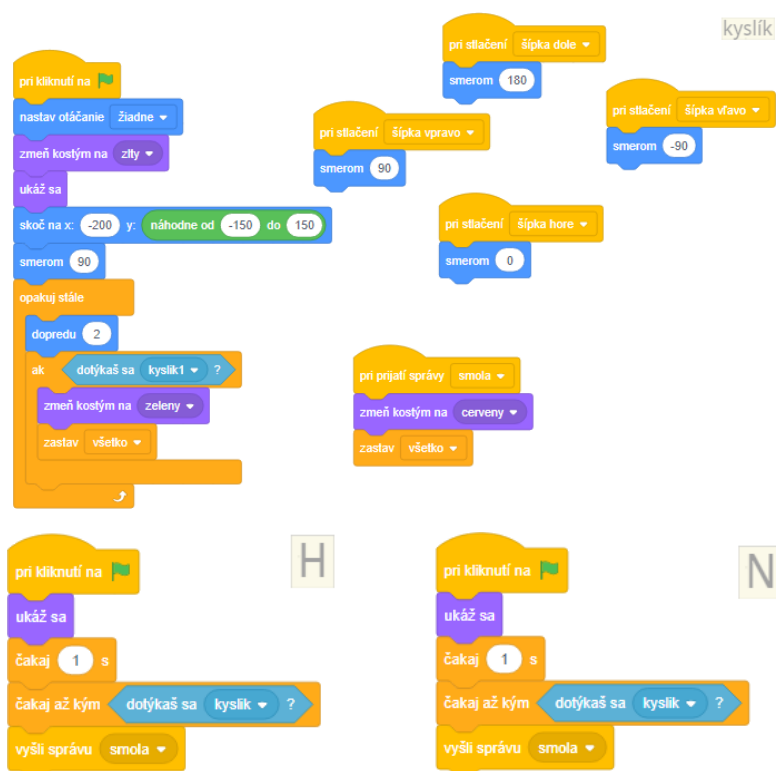
H

analýza riešenia

- potrebujeme 4 postavy – kartičku s názvom chemického prvku a tri značky prvkov (tých by mohlo byť samozrejme aj viac), všetky postavy nakreslíme v grafickom editore Scratch a vhodne ich pomenujeme, napr. kyslík (názov), kyslík1 (značka kyslíka), vodík1 (značka vodíka), dusík1 (značka dusíka)
- scenár pre názov prvku bude veľmi podobný tomu, ktorým sme sa pohybovali v labyrinte pomocou šípok, vytvoríme tiež nekonečný cyklus, v ktorom sa bude postava sama pohybovať doprava
- ak sa postava dotkne **svojej** chemickej značky, zmení si kostým na zelený

- ak postava dostane správu, že sa dotkla inej chemickej značky, zmení si kostým na červený
- každej kartičke s **nesprávnou** značkou vytvoríme scenár, v ktorom bude zisťovať, či sa dotýka kartičky s názvom a ak áno, pripravíme reakciu, v ktorej pošle správu, aby si kartička s názvom chemického prvku zmenila farbu kostýmu na červenú, pretože sa dotkla nesprávnej značky

riešenie



- všimnime si, že kartičky s **nesprávnymi** chemickými značkami majú **úplne rovnaké** scenáre
- takto sme vytvorili **prototyp** postavy, ktorá zobrazuje nesprávnu chemickú značku pre prvok, ktorým pohybuje po scéne
- ich **duplikovaním** a **zmenou tvaru** môžeme získať aj ďalšie chemické značky iných prvkov

ALTERNATÍVNY PRIEBEH HODINY

Učiteľ môže usmerniť žiakov tak, aby riešili buď **úlohu 3** alebo len **úlohu 4**, nie je dôležité, aby počas hodiny vyriešili obe úlohy.

Učiteľ môže namiesto úlohy 3 a 4 motivovať žiakov, aby riešili úlohu na precvičovanie **19_chem_prvok_a_znacka**, pričom v prípade, ak žiakom ostane čas, môžu vytvoriť viacero nesprávnych chemických značiek alebo dokonca viacero kartičiek s názvom chemického prvku (treba si však uvedomiť, že toto je oveľa náročnejšia úloha).

20 STE(A)M: PROJEKTY

<i>Oblasť informatiky / Téma</i>	<i>ISCED / Odporúčaný ročník</i>
Algoritmické riešenie problémov	ISCED 2 / 8. - 9. ročník
Požiadavky na vstupné vedomosti a zručnosti	
Orientácia v prostredí Scratch, vedomosti a zručnosti zo všetkých predchádzajúcich aktivít.	
Ciele	
Žiakom osvojované vedomosti a zručnosti	Žiakom rozvíjané spôsobilosti
Prostredie Scratch <ul style="list-style-type: none"> v rámci tejto aktivity žiaci nezískavajú nové vedomosti a zručnosti, ale učia sa vhodne využívať tie, ktoré nadobudli na predchádzajúcich hodinách 	Informatické myslenie <ul style="list-style-type: none"> (ALG3) vytvárať vlastné algoritmy na riešenie problému (ALG6) využívať existujúce algoritmy v návrhu vlastných algoritmov (ALG8) zapísať algoritmy v konkrétnom programovacom jazyku
Aktivita ukazuje, že v prostredí Scratch môžeme vytvárať malé edukačné programy pre rôzne predmety. Metodiku sme pripravili tak, že žiaci navrhujú a riešia projekt pre matematiku – konkrétne sme sa zamerali na náročnejší projekt na prácu so zlomkami. Dôležitou súčasťou metodiky je nami navrhovaný postup pre vytváranie projektov aj pre iné vyučovacie predmety. Vo všetkých prípadoch odporúčame, aby sa do návrhu projektu zapojil aj učiteľ príslušného predmetu, pre ktorý budú žiaci projekt vytvárať.	
Dominantné vyučovacie metódy a formy	
Bádateľská metóda (5E). Frontálna forma.	
Príprava učiteľa a pomôcky	
<ul style="list-style-type: none"> projekt 20_oznac cast celku (pre učiteľa) jednoduchý projekt pre alternatívny priebeh hodiny 20_scitaj_cisla úloha na opakovanie 20_odhadni_percenta.ppsx, riešenie v projekte 20_odhadni_pecenta 	
Diagnostika splnenia vzdelávacích cieľov	
Pozorovanie činnosti žiakov vo fáze skúmania a rozpracovania. Spoločné hodnotenie projektov.	

ÚVOD

Prostredie Scratch je vhodné nielen na tvorbu a vývoj hier, ale aj na vytváranie malých edukačných aktivít pre rôzne školské predmety. Podľa Paperta sa žiaci najviac naučia vtedy, **keď vytvárajú** niečo užitočné a to **nielen pre seba**, ale aj pre niekoho iného, v našom prípade, napr. pre svojich spolužiakov alebo mladších žiakov, ktorí si budú môcť programy vyskúšať. Vytváranie a programovanie takýchto projektov je vhodným prístupom k programovaniu náročnejších a väčších programov, mikrosvetov.

Žiaci pri vytváraní programov z rôznych STE(A)M oblastí, v tomto prípade **matematiky**, najprv **preskúmajú** témy, úlohy a zadania, s ktorými sa oni, alebo ich mladší spolužiaci stretli na hodinách matematiky. Následne **vyhľadajú** vhodné príklady, úlohy alebo iné aktivity z oblasti matematiky a predstavia ich svojim spolužiakom pomocou prezentácie. Pritom diskutujú o tom, či by sa dal vytvoriť program v prostredí Scratch pre tento typ úloh s vedomosťami, ktoré sa doteraz naučili. Potom **analyzujú** náročnosť programátorských konštrukcií, ktoré budú potrebné pri realizácii programu. V tejto fáze žiaci zisťujú a učia sa, že pre niektoré programy **nemajú dostatok vedomostí**, aby ich boli schopní zrealizovať v prostredí Scratch. Môže sa tiež ukázať, že pre niektoré aktivity prostredie Scratch neposkytuje dostatok nástrojov, aby sa v ňom (matematické) projekty dali naprogramovať.

Výsledkom tohto skúmania, diskusií a zdôvodnení bude, že **pre niektoré projekty** žiaci pripravia matematické programy – väčšinou malé edukačné hry.

Keďže bádateľské vyučovanie programovania, ktoré presahuje do rôznych školských vyučovacích predmetov, tzv. STEM, resp. STE(A)M projekty, môže mať rôznorodé a často aj divergentné výsledky – vytvorené projekty môžu byť veľmi rôzne a navzájom neporovnateľné a závisia nielen od vedomostí žiakov, ale aj od ich **tvorivých** schopností, je pomerne **náročné určiť presný postup**, ktorým by sa mali žiaci riadiť pri výbere a programovaní projektov.

Myslíme si, že je vhodné, aby sa v úvode bádateľského prístupu pre príslušný STE(A)M predmet do bádania a prípravy zapojil aj **učiteľ príslušného predmetu**, v našom prípade **učiteľ matematiky**, ktorý môže žiakom pomôcť pri vyhľadávaní vhodných aktivít na naprogramovanie.

VÝBER PROJEKTOV

Školská matematika obsahuje v jednotlivých ročníkoch na 2. stupni pomerne veľké množstvo tém. Je však **ťažké rozpoznať**, ktoré témy sú a ktoré nie sú vhodné ako programátorské projekty pre žiakov na 2. stupni. Samotní žiaci sa pri svojom bádani často **nebudú vedieť rozhodnúť**, či námet, ktorý našli, je vhodný na počítačové spracovanie v prostredí Scratch a to dokonca s vedomosťami a schopnosťami, ktoré z tohto programovacieho prostredia majú. Niekoľko námetov sme spracovali v metodike **18_STE(A)M: Matematika**.

Celý postup **od vyhľadávania** vhodného námetu na programovanie až po jeho **realizáciu** v prostredí Scratch na hodine informatiky, je proces, s ktorým budú žiaci potrebovať **pomoc od učiteľa** informatiky a aj od učiteľa príslušného predmetu.

Učiteľ informatiky môže na začiatku bádania nechať žiakov, aby medzi zverejnenými Scratch projektami vyhľadali matematické (alebo iné) aktivity, ktoré vytvorili iní detskí programátori. Je však dosť pravdepodobné, že väčšina projektov, ktoré nájdú, sa bude z matematiky orientovať na **prácu s číslami a základnými číselnými operáciami**. Takéto aktivity sa dajú nájsť medzi Scratch projektmi vo veľkom množstve a z programátorského hľadiska väčšinou vyžadujú iba základné programovacie techniky. Pritom učiteľ matematiky, a často aj sami žiaci, ihneď rozpoznať, že tieto programy **môžu byť** vo vyučovaní matematiky **užitočné**. Keď sú už naprogramované, žiaci ich využijú na vyučovaní matematiky pri **precvičovaní** a automatizácii základných početných úkonov, ako sčítanie, resp. násobenie, či delenie čísel.

Keďže takýto typ úloh je z programátorského hľadiska pomerne jednoduchý, rozhodli sme sa, že sa mu v našej metodike **nebudeme venovať**. Učiteľ si môže pozrieť viacero hotových projektov na internete a ak uzná za vhodné, môže nechať žiakov, aby podobný projekt vytvorili.



Projekt <https://scratch.mit.edu/projects/131961360/>

V tejto metodike navrhne postup, v ktorom sa budeme snažiť žiakov priviesť k tomu, aby vyhľadali a programovali témy matematiky, ktoré budú obsahovať pre používateľa, teda pre žiakov, ktorí budú neskôr s programom pracovať, aspoň **istú dávku experimentovania**, teda budú sledovať **konštruktivistický** prístup vo vyučovaní.

VYHLADÁVANIE NÁMETOV PRE MATEMATICKÉ EDUKAČNÉ HRY

Bádanie v oblasti STE(A)M v matematike, a aj v iných predmetoch, môžeme rozdeliť na niekoľko častí, z ktorých **niektoré** je potrebné zrealizovať **mimo jednu, či dve** vyučovacie hodiny **informatiky**, ktoré pre tento projekt má učiteľ k dispozícii v rámci vyučovania. Hodina informatiky by mala obsahovať len **samotné programovanie** projektu v prostredí Scratch.

Ďalej uvádzame návrh **skupiny bádateľských aktivít**, ktoré môžu viesť žiakov k vytvoreniu malého edukačného programu pre matematiku a aj iné vyučovacie predmety.

Žiaci:

1. spolu s **učiteľom príslušného vyučovacieho predmetu**, v našom prípade matematiky, skúmajú učebnice predmetu (matematiky) pre 6. až 9. ročník
2. v skupinách alebo v individuálnych konzultáciách s učiteľom príslušného predmetu (matematiky), **zistujú**, ktoré témy by mohli byť vhodné na spracovanie pomocou počítačovej aplikácie
3. následne si vyberú **aktivitu**, ktorú budú programovať
4. v **individuálnych** konzultáciách s učiteľom informatiky určujú **programátorskú náročnosť** svojho budúceho programu
5. v rámci domácej prípravy navrhujú **dizajn a ovládanie** svojho programu
6. pred samotným programovaním si pripravia prezentáciu svojej aktivity, mali by rozpoznať a vedieť prezentovať aj programátorské konštrukcie, ktoré budú potrebovať pri realizácii programu
7. **prezentujú** svoju aktivitu, dizajn a jej ovládanie pred spolužiakmi a v diskusii odpovedajú na otázky
8. v rámci hodiny informatiky **programujú** aktivitu, keďže pôjde o zložitejšie projekty, je pravdepodobné, že žiaci budú projekt **dokončovať** doma
9. pripravené projekty žiaci **zverejnia** v Štúdiu Scratch a dajú ich tak k dispozícii mladším žiakom, pre ktorých projekt vytvárali, v komentároch k projektu vidia, ako sa program žiakom páčil
10. na ďalšej hodine, resp. pri nejakej vhodnej príležitosti, môže učiteľ zorganizovať prezentáciu a diskusiu ku všetkým vytvoreným projektom

METODIKA K VYTVORENIU PROJEKTU

Aby učiteľ získal predstavu o princípe vyhľadávania a programovania aplikácií pre STE(A)M projekty, pripravili sme metodický postup pre jeden matematický program. Téma, ktorú sme si vybrali sa týka vyučovania práce so zlomkami v 7. ročníku základnej školy. Veríme, že tento, a podobné námety, by mohli žiaci spracovať a programovať v rámci hodín **STE(A)M v matematike**.

ZAPOJENIE – PREBIEHA V RÁMCI DISKUSIE NA HODINÁCH MATEMATIKY

Časti 1. až 3. skúmanie učebníc, diskusie na hodine matematiky, výber aktivity na programovanie, pozri postup na vytváranie projektov

V učebnici Černek, Žabka: Matematika pre 7. ročník, 1. časť sa nachádza téma zlomky. Na str. 42 je skupina úloh, ktoré sa venujú zobrazovaniu časti celku na rôznych reprezentáciách pre zlomky – používajú model čokoláda, pizza alebo torta.

V tejto časti bádania je dôležité, aby **žiaci dostávali podnety od učiteľa**. Bolo by dobre, ak by na takejto hodine matematiky bol aj učiteľ informatiky. Učiteľ by mal priviesť žiakov k tomu, aby sami **objavili**, akým spôsobom by sa dalo určovanie častí celku spracovať na počítači. Mohli by si spomenúť na to, ako mali na hodinách k dispozícii štvorčekový papier a ako na ňom vyznačovali časti celku, ktoré určovalo zadanie jednotlivých úloh. Žiaci by mali objaviť a uvedomiť si paralelu medzi štvorčkami na štvorčekovom papieri a štvorčekovou sieťou, ktorú vytvárali na niektorej z minulých aktivít v prostredí Scratch. Ak by si žiaci na štvorčekovú sieť nespomenuli, učiteľ informatiky, by im mohol pripomenúť programy, v ktorých využívali štvorčekovú sieť – **12 Štvorčeková sieť**.

V tejto fáze bádania, v ktorej si žiaci **vyberajú** vhodnú tému, by mohol mať učiteľ matematiky pripravených niekoľko vlastných tém, o ktorých si myslí, že sú vhodné na spracovanie na počítači. Tieto témy by mal mať **prekonzultované s učiteľom informatiky**, aby boli realistické a vhodné na spracovanie v prostredí Scratch a aby sa dali naprogramovať so súčasnými vedomosťami žiakov.

SKÚMANIE – PREBIEHA FORMOU INDIVIDUÁLNYCH KONZULTÁCIÍ A V RÁMCI DOMÁCEJ PRÍPRAVY

Časti 4. až 6. individuálne konzultácie, dizajn, ovládanie, programátorské konštrukcie projektu, pozri navrhovaný postup pre vývoj projektov

Aj napriek tomu, že si žiaci podľa učebnice matematiky, resp. po konzultáciách s učiteľom matematiky, vyberú program, ktorý chcú spracovať v prostredí Scratch, keďže ide o komplexný program, môže byť pre nich náročné určiť postup, ako ho naprogramovať. Navrhujeme preto, aby sa učiteľ informatiky venoval žiakom v rámci **individuálnych konzultácií**. Spolu so žiakmi by mal vyšpecifikovať, ako bude program pracovať, aké vstupné informácie bude požadovať od používateľa, aké informácie si musí uchovávať, aké bude mať ovládacie prvky, a napokon, ako budú s programom pracovať používatelia, v našom prípade žiaci 7. ročníka.

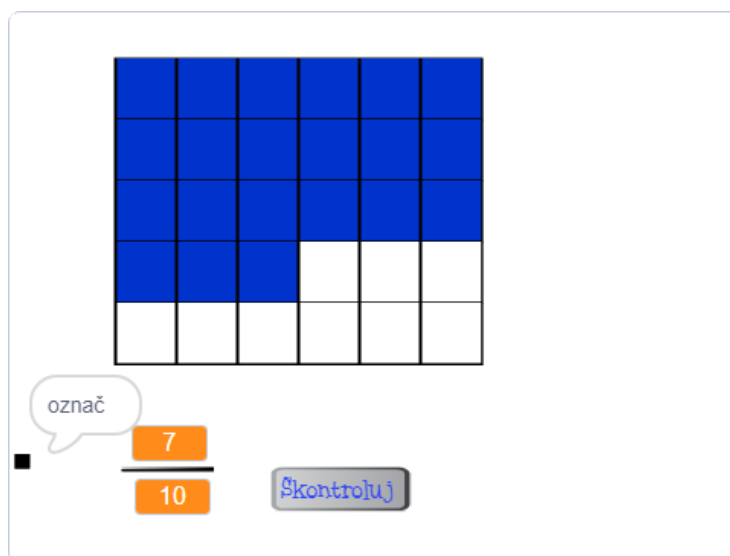
Pre aktivitu **označ časť celku** bude program vlastne simulovať štvorčekovú sieť, ktorú žiaci používali v učebnici alebo v zbierke úloh, resp. Pomocníkovi. Žiaci – používatelia programu – budú mať klikaním na jednotlivé štvorčeky zafarbiť časť celku, ktorú im program určí. Program bude kontrolovať, či žiak zafarbil správny počet štvorčekov tak, ako určovalo zadanie úlohy. Učiteľ by mal žiakov priviesť k tomu, aby si uvedomili, že počítač bude **generovať** zadanie úloh, teda bude určovať počet štvorčekov v celku a aj počet štvorčekov, ktoré má žiak zafarbiť. Žiak by si mohol uvedomiť, že počítač takýmto spôsobom umožňuje pripraviť „nekonečne veľa zadaní“. Učiteľ by mal otázkami priviesť žiaka k tomu, že je dôležité, koľko štvorčekov sa zmestí na plochu prostredia Scratch a aj na to, že počet štvorčekov, ktoré má používateľ v programe zafarbiť by mal byť menší ako počet štvorčekov, ktoré tvoria celok.

Keďže v našej metodike spracovávame **náročnejšie** projekty veríme, že žiaci, ktorí ich budú programovať majú **dostatok vedomostí** a aj programátorských skúseností a zručností na to, aby s nimi učiteľ nemusel podrobne konzultovať samotný program a jeho realizáciu v prostredí Scratch. Žiaci si s učiteľom môžu tieto koncepty pripomenúť a prípadne načrtnúť niektoré náročnejšie časti programu.

V prípade našej vybranej aktivity **označ časť celku** by učiteľ mohol žiakom pripomenúť, že v niektorej z predchádzajúcich aktivít zobrazovali štvorčeky v sieti, že štvorček mohol mať viac kostýmov. Žiaci by si mohli spomenúť, že každá postava môže reagovať na udalosť **pri kliknutí na mňa**.

Šikovnejších žiakov môže učiteľ priviesť k tomu, aby predtým, ako zobrazia, akú časť celku má používateľ označiť, dali zlomok do základného tvaru. Učiteľ môže uviesť odkaz na **Euklidov** algoritmus na zistenie najväčšieho spoločného deliteľa. Veríme, že žiaci **so záujmom** nielen o informatiku ale aj o matematiku môžu byť schopní tento algoritmus a jeho naprogramovanie v Scratch pochopiť.

Učiteľ by sa v rámci konzultácií so žiakmi mohol zaoberať aj dizajnom programu. V našom prípade tým, kde a ako sa zobrazí informácia o tom, ktorú časť celku je potrebné označiť a to, kde bude tlačidlo, ktorým program skontroluje správnosť riešenia, pozri obrázok.



Projekt označ časť celku

VYSVETLENIE

Časť 7. prezentácia návrhu projektu, pozri navrhovaný postup na str. 4.

Po individuálnych konzultáciách s učiteľom matematiky a informatiky si žiaci o svojom projekte pripravujú **krátku prezentáciu**. Táto časť bádateľskej aktivity slúži na to, aby žiaci dokázali stručne a jasne sformulovať výsledky svojho bádania o vyhľadávaní a príprave svojho matematického programu. Prezentáciu vytvárajú v rámci domácej úlohy alebo v rámci inej hodiny informatiky. Pri prezentácii a diskusii so spolužiakmi sa môžu objaviť rôzne otázky, resp. problémy, na ktoré autor počas návrhu projektu zabudol.

Žiak pomocou prezentácie (využitím dataprojektora alebo interaktívnej tabule) predstaví svoj návrh spolužiakom. Žiak prezentuje svoj návrh matematického projektu, vysvetlí tému, ktorej je projekt venovaný, ukáže svoj grafický návrh a vysvetlí ovládanie svojej budúcej aplikácie. Súčasťou prezentácie by mala byť aj analýza programátorských konštrukcií, ktoré bude pri riešení potrebovať a používať. Žiak vysvetlí rozdelenie programu na jednotlivé podprogramy, resp. časti programu, ktoré sa budú týkať jednotlivých postáv.

Súčasťou tejto časti hodiny je spoločná **diskusia žiakov** k projektu. Týmto spôsobom sa vyjasní téma projektu a často sa ukážu rôzne úskalia, o ktorých žiak neuvažoval, napr. so vstupnými hodnotami pre program, resp. reakciami na nesprávne vstupy, s dizajnom – prečo je tlačidlo umiestnené tam, kde žiak navrhuje, kde sa bude vypisovať informácia pre používateľa, atď.

ROZPRACOVANIE

Časť 8. a 9. programovanie a zverejnenie projektu, pozri navrhovaný postup na str. 4.

Počas vyučovacej **hodiny informatiky** pracujú žiaci samostatne na riešení svojich projektov. Učiteľ im pomáha, radí, usmerňuje ich. Keďže ide o programovanie náročnejších STE(A)M projektov mali by byť žiaci schopní pracovať pomerne samostatne a mali by dokázať tvorivo riešiť problémy, ktoré sa počas programovania vyskytnú.

HODNOTENIE

Časť 10. hodnotenie projektu, pozri navrhovaný postup na str. 4.

Žiaci zverejnia svoje projekty v Scratch Štúdiu alebo iným spôsobom ich poskytnú používateľom, v našom prípade žiakom 7. ročníka. V rámci zverejneného projektu môžu **vidieť hodnotenia** od používateľov, ktoré ich projekt dostal. Na ďalšej hodine informatiky, alebo pri inej príležitosti, môžu žiaci svoje projekty prezentovať. Celkové **hodnotenie** projektov prebieha v rámci spoločnej **diskusie**.

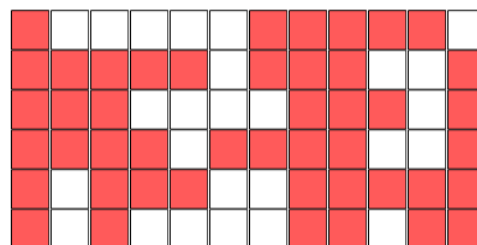
Učiteľ môže dať žiakom v rámci hodnotenia vyriešiť úlohu **Odhadni percentá** (na začiatku ďalšej hodiny alebo v rámci opakovania či utvrdzovania učiva). Pomocou nej zistí, či žiaci pochopili pečiatkovanie postavy a vytváranie viacriadkovej štvorčekovej siete. Zadanie je pripravené ako prezerateľná prezentácia v súbore **20_odhadni_percenta**, riešenie je v Scratch projekte s rovnakým názvom.

Odhadni percentá

Navrhni projekt, ktorý dovolí žiakom odhadovať počet percent zafarbených štvorčekov.

Na ploche sa vykreslí (opečiatkuje) 6x12 štvorčekov. Niektoré z nich sú zafarbené.

Postava sa spýta žiaka, aby odhadol koľko percent tvoria zafarbené štvorčeky. Program potom, ako žiak zadá svoj odhad zobrazí v jednej bubline odhad a v druhej skutočný počet percent zafarbených štvorčekov.



odhad:62

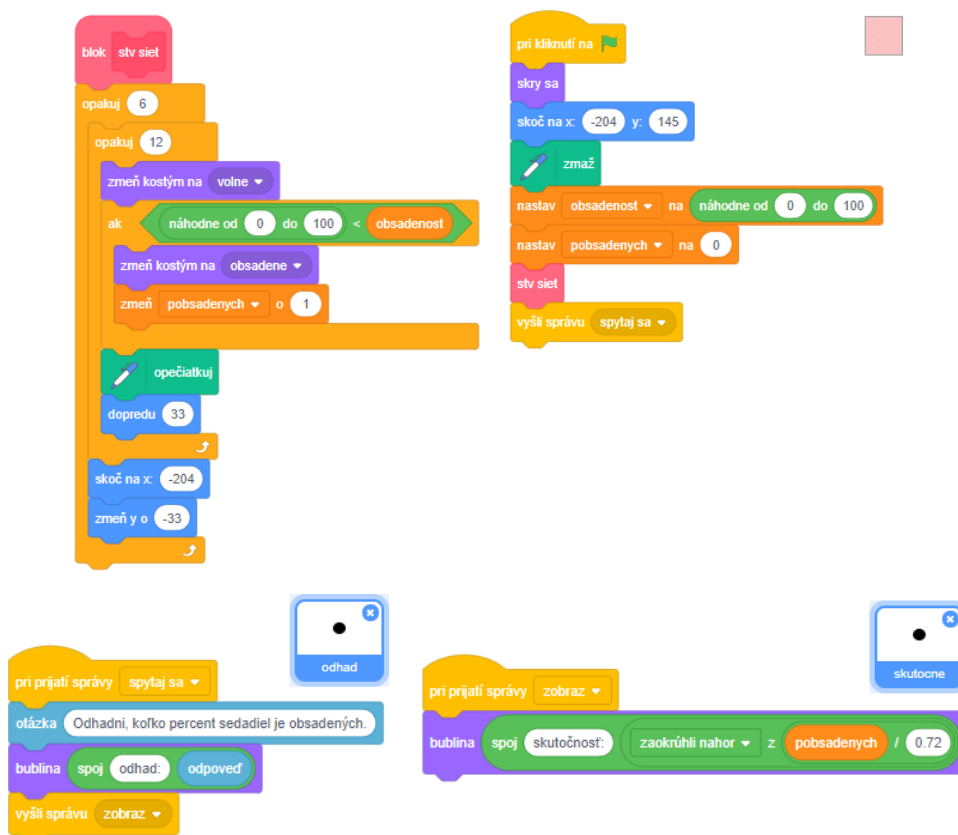
skutočnosť:64

analýza zadania

- potrebujeme postavu **štvorček** s dvoma kostýmami, bielym a červeným
- navrhne scenár, ktorý opečiatkuje štvorčeky do 6 riadkov v každom riadku 12 štvorčekov, v scenári sa pri každom štvorčeku sa rozhodneme, či bude biely alebo červený, podobnú úlohu sme riešili v zadaní **12 Štvorčeková sieť**

- pri pečiatkovaní spočítame (napr. do premennej **cervene**) počet červených štvorčiekov
- vytvoríme dve postavy (s menami napr. **odhad**, **skutocne**), ktoré budú zobrazovať bubliny s odhadom a skutočným percentom zafarbených štvorčiekov
- postava **odhad** sa spýta na žiakov odhad a po jeho zadaní zobrazí tento odhad vo svojej bubline
- potom postava **skutocne** zobrazí v bubline vypočítaný počet percent (uvedomme si, že postava odhad musí poslať správu, že žiak už svoj odhad zadal)

riešenie



ALTERNATÍVNY PRIEBEH HODINY

Ak má učiteľ žiakov, ktorí **nemajú záujem** o programovanie, môže im dať namiesto projektu **Označ časť celku** vypracovať **jednoduchší** projekt, napr. **Sčítaj dve čísla**, ktorý sa podobá na viacero projektov, ktoré zrejme žiaci našli na internete, keď hľadali matematické projekty. Projekt obsahuje tri postavy, celé riešenie vidíme na obrázku a je aj v projekte **20_scitaj_cisla**.

The image shows a Scratch project with a script on the left and a stage on the right. The script is triggered by a click and sets a score to 0. It then generates two random numbers, adds them, asks the user for the sum, and checks the answer. If correct, the score is incremented and a 'Výborne!' bubble is shown. If incorrect, a bubble prompts the user to try again. The stage displays the current score (3) and a math problem: $471 + 6 =$. Below the stage is a control panel with character positions and a toolbar with a dot, plus, and equals signs.

Script:

- when clicked on the flag
- set pspravne to 0
- when green flag clicked
- hide variable sucet
- set cislo1 to random number from 200 to 500
- set cislo2 to random number from 1 to 10
- set sucet to cislo1 + cislo2
- ask aký je súčet?
- if answer = sucet
- show variable sucet
- change pspravne by 1
- say Výborne! for 2 seconds
- else
- show variable sucet
- say Pozri si výsledok a porozmyšľaj... for 2 seconds

Stage:

pspravne 3

$471 + 6 =$

Control Panel:

- Postava: Postava1 (x: -203, y: -97)
- Ukáž: On
- Veľkosť: 100
- Smer: 90
- Scéna: 1

Toolbar:

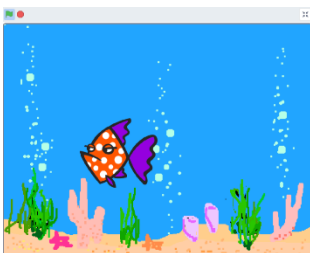
- Postava1 (dot)
- Postava2 (+)
- Postava3 (=)

PRÍLOHY

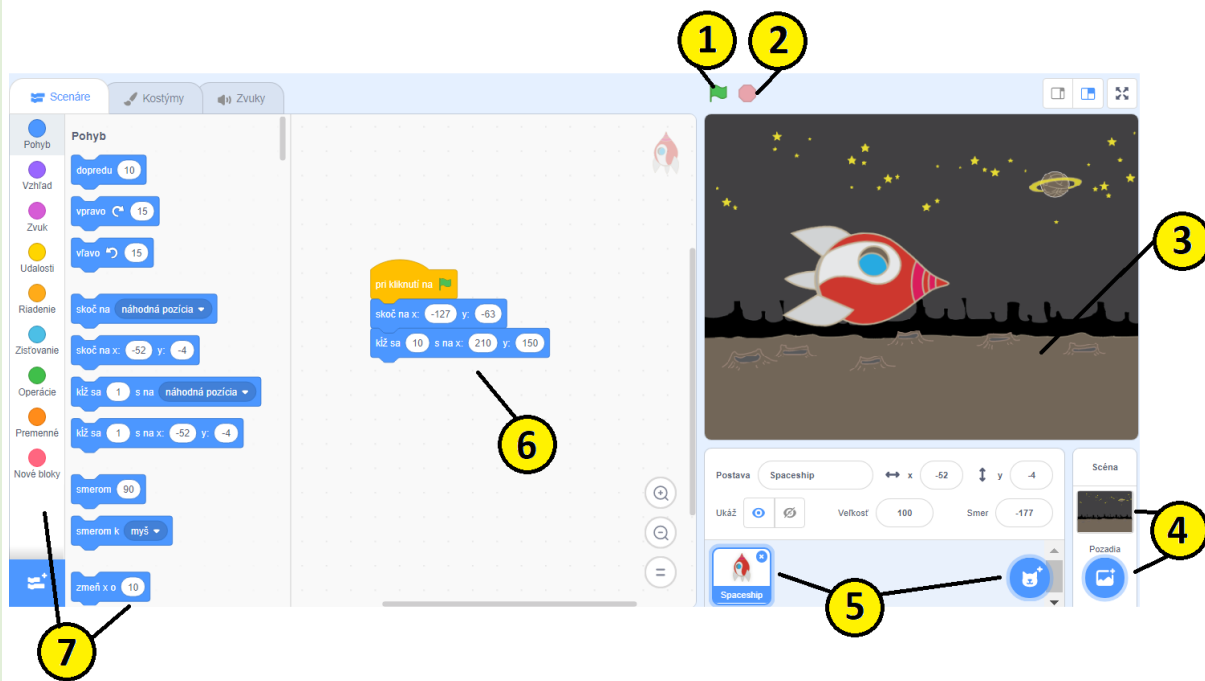
pracovné listy pre žiaka

PRESKÚMAJME SCRATCH

Zakresli, z ktorých častí sa skladá projekt **Akvárium**.



Úloha 1 Klikaj a skúmaj jednotlivé oblasti podľa čísel na obrázku a prirad čísla k správnej funkcii skúmanej oblasti podľa tabuľky:



3	okno bežiaceho programu
	skupiny príkazov pre programovanie (pre pohyb postáv, ich vzhľad, zvuky a pod...)
	práca s postavami (označenie postavy, zmazanie, duplikovanie a pod...)
	spustenie programu
	práca s pozadím scény (výber, kreslenie a pod...)
	program, t.j. čo má robiť zvolená postava (scénár)
	zastavenie programu

Úloha 23

- a) Preskúmaj oblasť pre prácu s pozadím scény a zisti a zapíš, akými spôsobmi sa dá vytvoriť nové pozadie. Spôsoby vytvorenia pozadia zistíš pomocou nápovede – stačí priblížiť myšku k ikonke a neklikať.
- b) Preskúmaj knižnicu s pozadiami a vyber si niektoré. Dokážeš pridať nové pozadie? **ÁNO NIE**

**Úloha 3**

- a) Preskúmaj oblasť pre prácu s postavami a zisti, akými spôsobmi vieš vytvoriť novú postavu:
- b) Preskúmaj knižnicu s postavami a vyber si niektorú postavu. Postava sa objaví na scéne aj v okne postáv. Dokážeš ju na scéne presunúť na iné miesto? **ÁNO NIE**
- c) Vyskúšaj kliknúť pravým tlačidlom myši na zvolenú postavu v okne postáv. Vieš ju zmazať? **ÁNO NIE**

**Úloha 4**

Preskúmaj skupiny blokov – nájdi a zapíš, v ktorých skupinách sa nachádzajú nasledujúce bloky.

Skupina blokov Udalosti	Skupina blokov	Skupina blokov

Úloha 5


Do hotového projektu **akvária** navrhni a doplň vlastné vylepšenia alebo rozšírenie.


Svoje návrhy zapíš sem:

ČO SOM SA NAUČIL/NAUČILA...

Viem vysvetliť, na čo slúžia základné časti prostredia Scratch	VIEM / VIEM S POMOCOU / NEVIEM
Viem vybrať pozadie z knižnice	VIEM / VIEM S POMOCOU / NEVIEM
Viem vybrať postavu z knižnice	VIEM / VIEM S POMOCOU / NEVIEM
Viem vytvoriť jednoduchý pohyb postavy	VIEM / VIEM S POMOCOU / NEVIEM

ČO SME SA NAUČILI

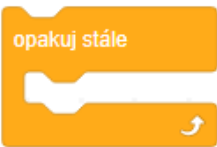
Postavy môžeme meniť alebo vytvárať kliknutím na ikonu  v pravom dolnom rohu prostredia Scratch.

Pozadia scény môžeme meniť alebo vytvárať kliknutím na ikonu 

Pri programovaní sme použili nasledovné bloky:



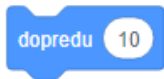
nastavenie začiatku programu (spustenie kliknutím na zelenú vlajku)



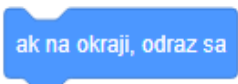
neustále opakovanie (blokov, ktoré sú vo vnútri tohto bloku)



pozastavenie programu na určitý čas (napr. pre spomalenie postavy)



pohyb postavy o 10 krokov dopredu



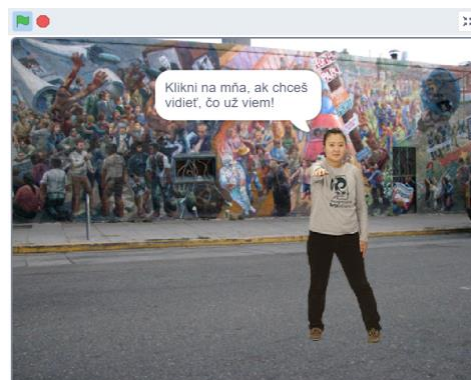
odrazenie postavy od okraja okna scény



nastavenie spôsobu otočenia postavy po odraze

ANIMÁCIE

Z ktorých častí sa skladá projekt **interaktívna postava**?



Úloha 1 Kliknutím na zelenú vlajku spustíte program, otestujte ho a doplňte tabuľku:



Udalosť	Akcia (čo sa deje?)
Po kliknutí na zelenú vlajku	
Po kliknutí na smajlíka	
Po stlačení medzerníka	

Úloha 2 Sledujte bližšie správanie sa smajlíka po stlačení medzerníka a usporiadajte správne jednotlivé fázy animácie – doplňte poradové čísla:

1.						

Úloha 3 Kliknite na záložku **Kostýmy**:



Preskúmajte v programe kostýmy smajlíka a dokreslite ich podľa názvov:

<i>smajlik1</i>	<i>smajlik2</i>	<i>smajlik3</i>	<i>smajlik4</i>	<i>smajlik5</i>

Úloha 4 Preskúmajte bližšie skupiny príkazov – nájdite a zapíšte, v ktorých skupinách sa nachádzajú nasledujúce bloky/príkazy:

Skupina príkazov: Udalosti	Skupina príkazov:	Skupina príkazov:
Skupina príkazov:	Skupina príkazov:	Skupina príkazov:

Úloha 5 Porovnajete v programe príkazy, ktoré sa vykonajú po kliknutí na zelenú vlajku, po kliknutí na smajlíka a po stlačení medzerníka a pokúste sa pridať nové správanie smajlíka tak, aby

- po stlačení klávesu **a** smajlík krátko žmurkol a chvíľu hovoril „Ahoj!“
- po stlačení klávesu **b** smajlík najprv povedal „Škoda, že už bude koniec...“ a potom postupne zosmutnel

Podarilo sa Vám to? **ÁNO - NIE**

Úloha 6 Do hotového projektu **interaktívna postava** navrhnete a doplňte vlastné vylepšenia alebo rozšírenie!

ČO SOM SA NAUČIL/NAUČILA...

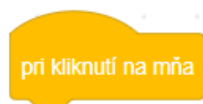
Viem zmeniť postavu kostým	VIEM / VIEM S POMOCOU / NEVIEM
Viem vytvoriť animáciu striedaním niekoľkých kostýmov	VIEM / VIEM S POMOCOU / NEVIEM
Viem pridať do programu zvuk	VIEM / VIEM S POMOCOU / NEVIEM
Viem zobraziť postavu bublinu na rozprávanie	VIEM / VIEM S POMOCOU / NEVIEM
Viem nastaviť, aby postava reagovala na stlačenie niektorého klávesu na klávesnici	VIEM / VIEM S POMOCOU / NEVIEM
Viem zastaviť vykonávanie scenára (napr. činnosť postáv, prehrávanie zvukov...)	VIEM / VIEM S POMOCOU / NEVIEM

ČO SME SA NAUČILI

Zvuky môžeme meniť alebo vytvárať kliknutím na ikonu:



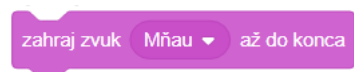
Pri programovaní sme použili nasledovné bloky:



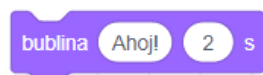
nastavenie udalosti – akcie sa vykonajú po kliknutí na danú postavu



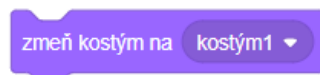
nastavenie udalosti – akcie sa vykonajú po stlačení daného klávesu na klávesnici



prehratie zvoleného zvuku



postava rozpráva – zobrazenie komiksovej bubliny



zmena vzhľadu postavy – výmena jej kostýmu



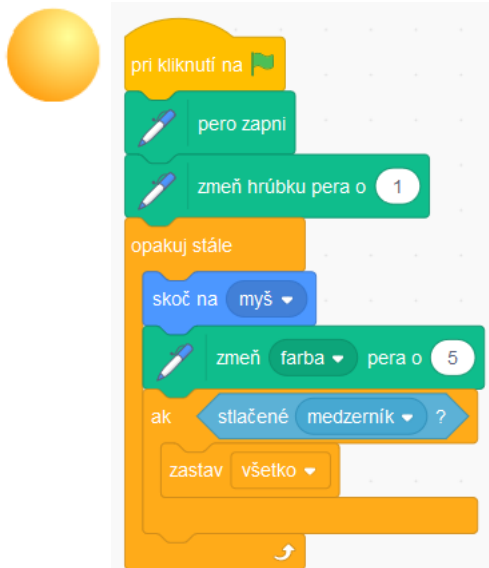
ukončenie vykonávania všetkých scenárov

UDALOSTI MYŠI A KLÁVESNICE

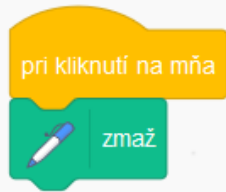
Úloha 1 Otvorte projekt **Chyť myšičku!** a zahrajte sa hru na základe inštrukcií v pozadí.



Úloha 2 Skúmajte program **Kreslenie**. Objavte na čo slúžia nasledujúce príkazy (bloky) a spojte ich s popismi:



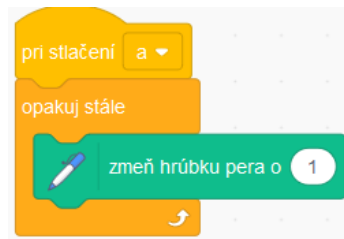
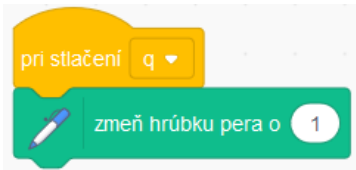
Spustí program
Zmení hrúbku pera o 1
Opakuje príkazy, ktoré sú vo vnútri bloku
Zmení farbu pera o 5
Skočí/prilepí sa na myš
Zapne funkciu pera - kreslenie
Ak je podmienka splnená, zastaví



Ak klikneme na postavu, tak sa niečo vykoná

Zmaže kresbu na scéne

Doplňte do projektu **zmenu hrúbky pera** ovládaním pomocou klávesov:



Zistite rozdiel medzi oboma spôsobmi.

Pridajte ďalšie ovládanie kreslenia pomocou klávesnice.

Úloha 3

Vytvorte vlastný projekt – Mapa

Vytvorte projekt na kreslenie mapy ulíc mesta, kde všetky ulice sú na seba kolmé. Bloky na ovládanie pera pridáte výberom rozšírenia v ľavom dolnom rohu – **Rozšírenie Pero**.

1. Zvoľte postavičku na kreslenie.
2. Vytvorte program, kde po kliknutí na zelenú zástavu sa bude postava neustále pohybovať, ťahať za sebou čiaru a meniť smer podľa stlačených šípok na klávesnici. Použite nasledovné bloky:

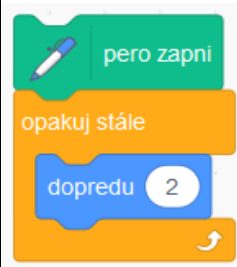
Smerovanie postavy môžete meniť napríklad takto:

3. Doprogramujte projekt tak, aby sa postava pohybovala v smere stlačených šípok HORE, DOLE, VPRAVO a VĽAVO.
4. Skúste zmeniť farbu, hrúbku pera, rýchlosť postavy.
5. Navrhňte vhodné zastavenie kreslenia.

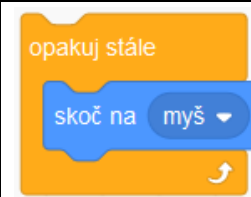
ČO SME SA NAUČILI

Spojte čiarami jednotlivé popisy ovládania s príslušným blokom príkazu.

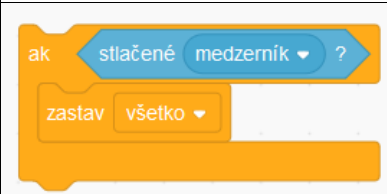
Postava sa stále otáča k postave s menom myšička.



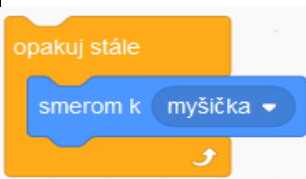
Postava pri kreslí čiaru.



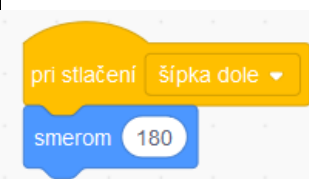
Hra sa zastaví stlačením klávesu medzerník.



Postava sa pohybuje spolu s kurzorom myši.



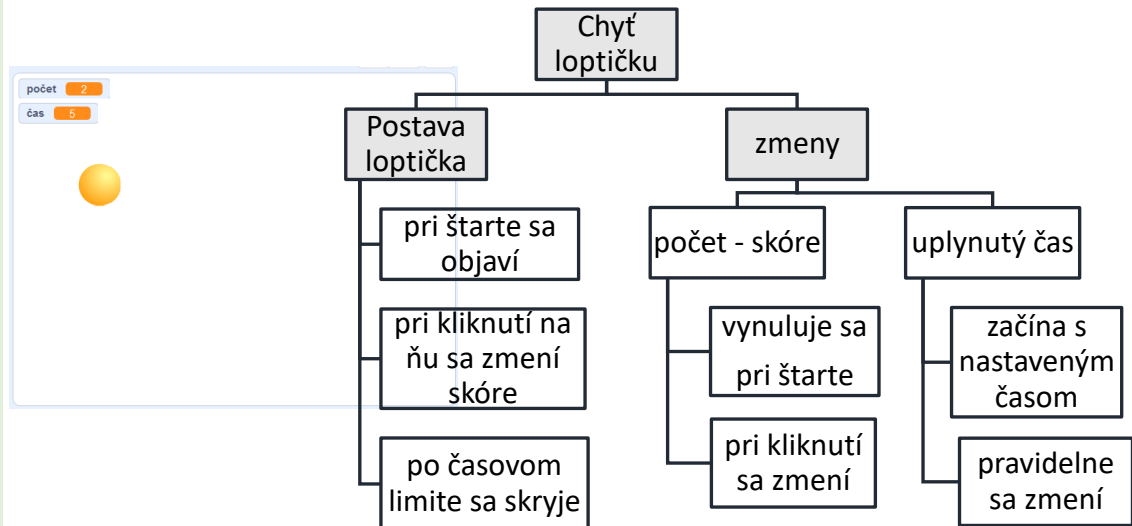
Zmení smer postavy po stlačení klávesu.



ZADÁVAME VSTUPY, VYTVÁRAME PREMENNÉ

Úloha 1 Otvorte si projekt **chyt loptičku** a zahrajte sa hru.

Zakrúžkujte udalosti, ktoré už by ste vedeli naprogramovať.

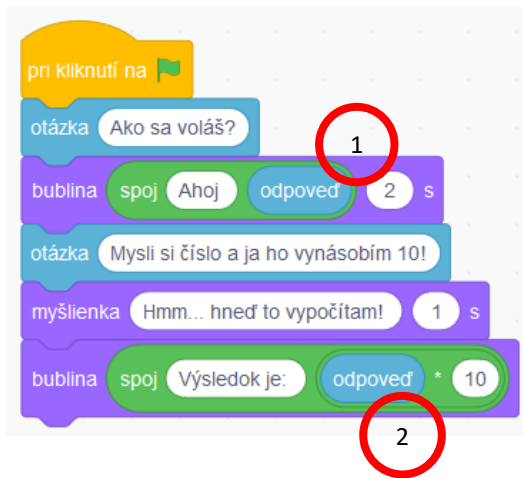


Úloha 2 Preskúmajte projekt **nasobime** a čiarami spojte jednotlivé príkazy a popisy.

		<p>Spojenie 2 textov (naráz sa vypíše text a hodnota odpovede na otázku).</p>
		<p>Blok otázka, teda výzva na zadanie odpovede.</p>
		<p>Uloženie odpovede.</p>
		<p>Postava si myslí text počas sekundy.</p>
		<p>Zadané číslo sa vynásobí s 10.</p>
		<p>Postava hovorí text počas 2 sekúnd.</p>

Úloha 3

Aká je hodnota **odpoveď** v jednotlivých fázach vykonávania programu?



	Hodnota odpoveď v na mieste označenom 1.	Hodnota odpoveď v na mieste označenom 2.
1. pokus		
2. pokus		
3. pokus		

Upravte program tak, aby zadané číslo vynásobil iným, napríklad dvojciferným. Program otestujte.

Úloha 4

Vlastný projekt **nakup**.

Navrhnite projekt s názvom **nakup**, v ktorom sa predávajúci opýta **Ako sa voláš?**

Potom sa opýta, za akú cenu si chcem kúpiť koláčik a koľko kusov koláčikov chcem kúpiť.

Program potom vypočíta koľko mám zaplatiť.

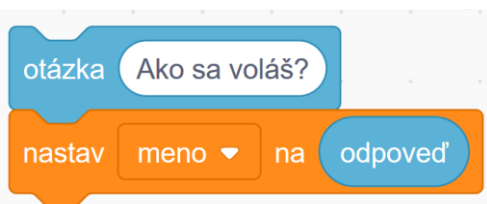
Vytvorte nové premenné:

meno

cena

počet

Po výzve na odpoveď uložte odpoveď do príslušnej premennej:



ČO SME SA NAUČILI?

Spojte čiarami jednotlivé popisy ovládania s príslušným blokom príkazu.		
Odpoveď na otázku sa uloží do premennej.		
Postava zadá otázku.		
Matematická operácia, ktorá vynásobí hodnotu dvoch premenných		
Hodnota premennej sa zmení o 1		
Postava skočí na náhodnú pozíciu		

POSIELAME SPRÁVY

Úloha 1 Preskúmajte projekt **rozhovor**. Prezrite si všetky postavy, scénu a odpovedzte:

Otázka	Odpoveď
Koľko pozadí je použitých v projekte a ako sa volajú?	2: stena,
Koľko kostýmov má chlapec?	
Koľko kostýmov má dievča?	
Kto posielal správu „otazka“ ?	
Kto na správu reaguje?	

Úloha 2 Doprogramujte projekt **rozhovor** tak, aby sa po skončení rozprávania dievčatka preplo pozadie scény na diskoteka a aby sa potom spustila hudba.

Spojte čiarami bloky, ktoré ste doprogramovali s postavami, ku ktorým ste bloky pridali.

The image shows a Scratch script editor with the following blocks:


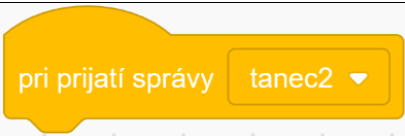
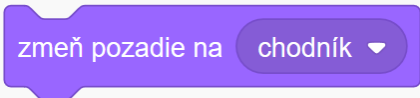
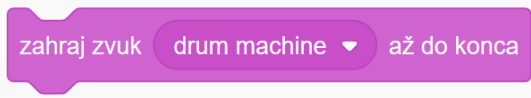
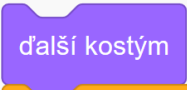
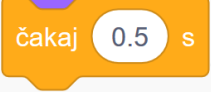
- Event block:** pri prijatí správy (when a message is received) with a dropdown menu set to 'sprava2'.
- Control block:** vyšli správu (send a message) with a dropdown menu set to 'sprava2' and 'a počkaj' (wait).
- Control block:** zmeň pozadie na (change background to) with a dropdown menu set to '2'.
- Sound block:** zahraj zvuk (play sound) with a dropdown menu set to 'bubny' and 'až do konca' (until the end).

There are also character images of a girl and a boy.

Úloha 3

Preskúmajte scenáre postavičiek a pozadia v projekte **tanec**.

Ktorá postava alebo pozadie vykonáva tieto príkazy?

Príkaz	Ktorá postava/pozadie vykonáva príkaz?
	
	
	
	
 	

V akom poradí vykonávajú príkazy jednotlivé postavy a pozadie?

1.	
2.	
3.	

Úloha 4

Do projektu **tanec** vložte novú postavu kráčajúceho dievčaťa Avery.

Jej pohyb naprogramujte tak, aby sa začala a skončila animovať súčasne s Ivanom. **Animácia** postavy je postupná **zmena kostýmu**.



Avery Walk...

ROZŠÍRENIE

Úloha 5 Preskúmajte projekt **klavír**.

Ako by sa na klavíri dalo hrať? Poskladajte jednotlivé bloky tak, aby sa pri kliknutí myšou na klávesu zahrál príslušný zvuk. Naprogramujte aj zmenu kostýmu počas hrania tónu, napr. kláves na chvíľu zafarbite.

Úloha 6 Nepovinná úloha.

Pridajte do projektu **klavír** iné funkcionality, napr. pri stlačení niektorého klávesu nech sa zahrá celá melódia.

Vymyslite a nižšie zapíšte aj ďalšie vylepšenia projektu **klavír**. Svoje nápady zrealizujte.

SEBAHODNOTIACA RUBRIKA

?	Viem? <input checked="" type="checkbox"/>
Posielať správy.	<input type="checkbox"/>
Spúšťať animáciu postáv po prijatí správy.	<input type="checkbox"/>
Opakovane animovať postavu.	<input type="checkbox"/>
Meniť pozadie projektu po prijatí správy.	<input type="checkbox"/>
Prehrávať hudbu a spúšťať ju udalosťou.	<input type="checkbox"/>

POUŽITIE KAMERY

Úloha 1

Vyskúšajte projekty **branka** a **klavir**.

Zahrajte si na klavíri, ktorý sa dá ovládať tromi spôsobmi:

- Kliknutím na klávesy myšou.
- Pomocou tlačidiel na klávesnici A, S, D, F, G, H, J.
- Ukazovaním na klávesy na obrazovke pomocou kamery. Nezabudnite si zapnúť kameru!

Úloha 2

Preskúmajte projekt **fotoaparát**.

Experimentujte s nastaveniami pre snímanie a odpovedajte na otázky v tabuľke:

Otázka	Odpoveď
Akým príkazom sa zapína zobrazovanie na scéne?	
Čo znamená príkaz nastav snímanie na 100 % ?	
Čo znamená príkaz nastav snímanie na 0 % ?	
Akým príkazom sa vypína zobrazovanie na scéne?	

Úloha 3

Preskúmajte projekt **odbijanie_dribling**.

Experimentujte s použitými príkazmi a meňte jednotlivé parametre, aby ste pochopili, ako príkazy fungujú. Na základe svojich pozorovaní popíšte príkazy.

Príkaz	Vysvetlenie príkazu

	Ako je zabezpečené neustále opakovanie pohybu lopty?
	Ako je zabezpečené neustále opakovanie pohybu lopty smerom nahor?

Úloha 4 Do projektu **odbijanie_dribling** doplňte inú loptu a naprogramujte ju tak, aby sa pohybom ruky odrážala od horného okraja plochy.

Pridajte prehratie zvuku a pridajte počítanie skóre – počet odbití rukou. Skóre sa vždy pri spustení programu musí vynulovať!

Úloha 5 Preskúmajte projekt **orchester**.

Doprogramujte ho tak, aby ste mohli snímaním pohybu rúk zahrať pripravené melódie pre saxofón a bicie.

Navrhňte aj ďalšie vylepšenia, napr. aby sa nástroje zväčšili a následne zmenšili a pod.



SEBAHODNOTIACA RUBRIKA

Zaškrtnutím zhodnoťte svoje vedomosti.	Viem	S pomocou viem	Neviem
Zapnúť snímanie, zastaviť snímanie.			
Nastaviť priehľadnosť snímania.			
Ovládať pohyb postavy pomocou pohybu pred			
Nastaviť citlivosť (senzitivitu) snímania.			
Meniť kostým postave pomocou pohybu pred kamerou.			

POUŽITIE MIKROFÓNU

Úloha 1

Preskúmajte projekt **sviecky**.

Experimentujte s použitými príkazmi a meňte jednotlivé parametre, aby ste pochopili, ako fungujú jednotlivé príkazy.



Úloha 2

Otvorte si projekt **stromy**.

Z ponúknutých blokov **zostavte program** tak, aby sa na určitú hlasitosť stromy kývali. Podľa toho, ako hlasno zafúkate, by sa stromy mali ohýbať. Využite zmenu kostýmov pre postavu stromu.

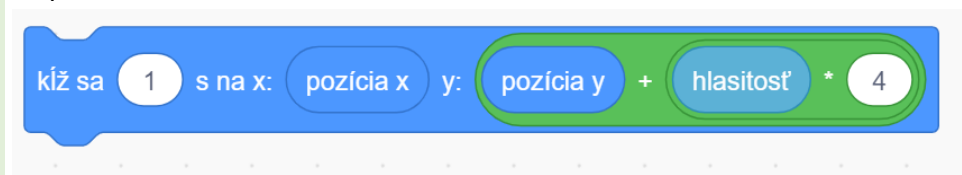


Úloha 3

Otvorte projekt **balony**.

Doprogramujte pohyb balónov tak, aby sa pri zmene hlasitosti pohybovali smerom nahor. Napr. ak zafúkate, alebo zatlieskate.

Môžete ovládať balóny aj tak, že čím väčší zvuk vydáte, tým budú stúpať rýchlejšie. Napr. takto.



Doprogramujte aj to, aby sa počas letu balóny súčasne aj zmenšovali, ako keby sa vzdďaľovali.

Úloha 4

Otvorte projekt **farebna_hudba**.

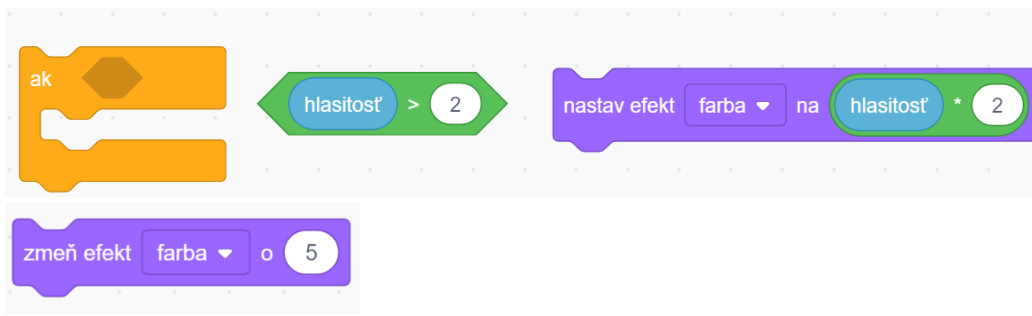
Doprogramujte ho tak, aby sa podľa hlasitosti prehrávanej hudby menili na postavách svetiel farebné efekty.

Vyberte, kde je intenzita hlasitosti hudby nastavená:

- pri Postave1
- pri každej postave
- pri Scéne



Možné riešenie môžete vyskladať aj pomocou takých blokov:

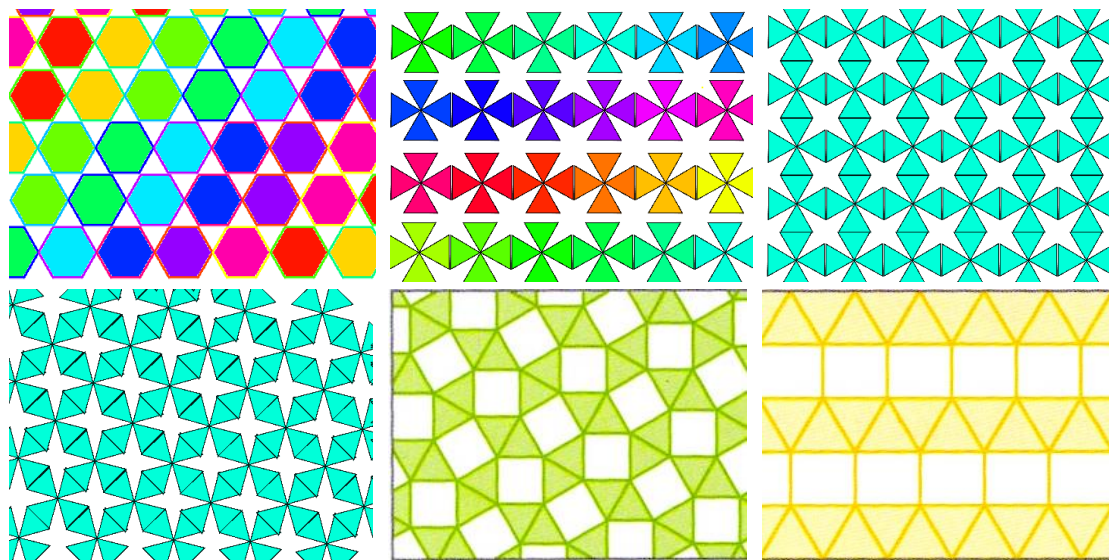


SEBAHODNOTIACA RUBRIKA

Zaškrtnutím zhodnotte svoje vedomosti.	Viem	S pomocou viem	Neviem
Spúšťať scenáre pomocou zvukov.			
Zistiť hlasitosť snímaného zvuku.			
Poznám rozdiel medzi hlasitosťou prehrávaného zvuku a hlasitosťou zvuku snímaného mikrofónom.			
Zmeniť pozíciu (súradnice) postavy v závislosti od hlasitosti.			
Meniť farebné efekty v závislosti od hlasitosti.			

MOZAIKY

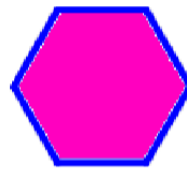
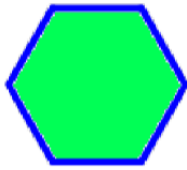
Úloha 1 Pozorujte jednotlivé mozaiky a vyznačte v nich opakujúce sa vzory.



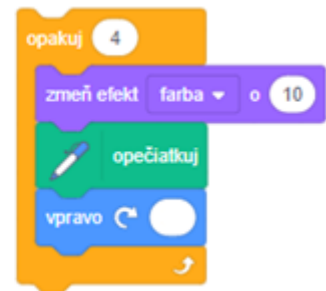
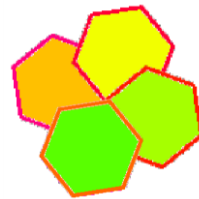
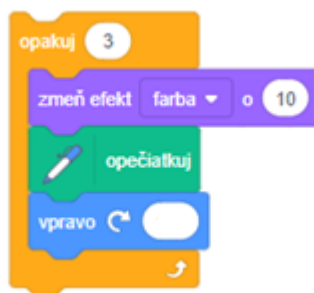
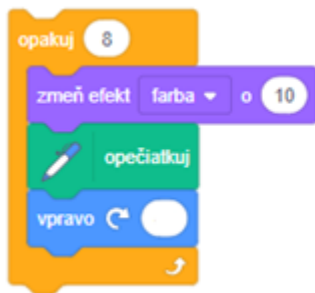
Úloha 2 Preskúmajte projekt **veterník**, vyhľadajte všetky programové bloky, s ktorými ste sa doteraz nestretli. Zapište názvy blokov a aj to, v ktorej kategórii blokov ste ich našli:

Kategória	Blok
	zmaž
Nové bloky	

Úloha 3 Otvorte projekt **mozaika**, vyskúšajte príkazy, ktoré sú naprogramované pre jednotlivé postavy. Prejdite do záložky **Kostýmy** a zistite a zakreslite bod, okolo ktorého sa otáča každý zo šesťuholníkov.



Pozrite si obrázky a ku nim prislúchajúce programy. Doplňte chýbajúce údaje do blokov.



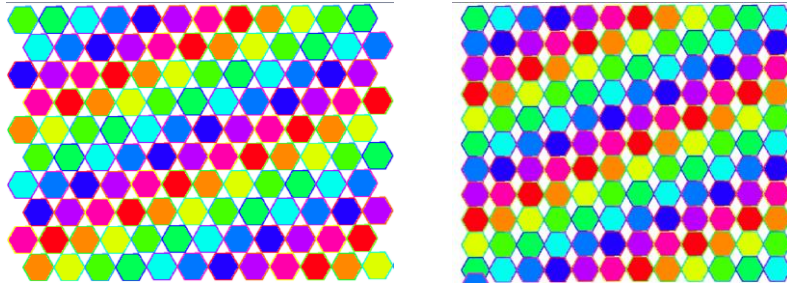
Úloha 4 Doprogramujte projekt **náramok** tak, aby ste nakreslili náramok, ktorý sa skladá z reťaze farebných korálikov a z ozdobného kvetu tak, ako vidíte na obrázku.



Vytvorte vlastné bloky **retaz** a **kvet**.

Môžete si nakresliť aj vlastné koráliky. Pri ich kreslení vždy uvažujte, okolo ktorého bodu obrázka sa má korálik otáčať.

Úloha 5 Naprogramujte návrh farebnej dlažby do svojej kúpeľne pomocou šesťuholníkových dlaždíc. Dlaždicu máte pripravenú v projekte **dlažba**. Vydláždenie môže vyzerať napr. takto:



Navrhňte vlastný blok **riadok**.

SEBAHODNOTIACI TEST

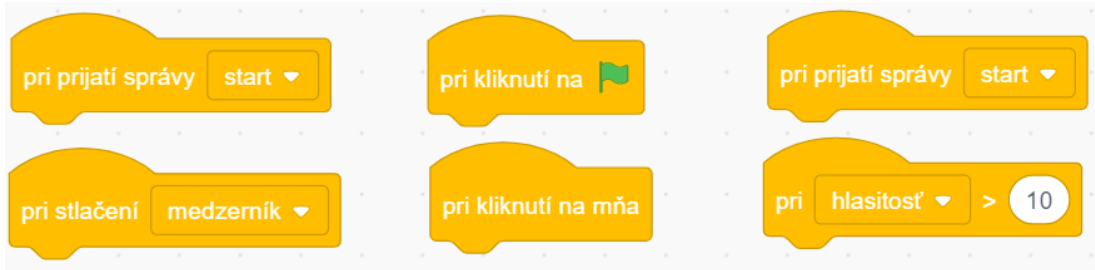
Spojte čiarami jednotlivé **popisy ovládania** s príslušným **blokom príkazu**.

<p>Skupina blokov, v ktorej je volanie nového bloku.</p>	<p>Vlastný blok, ktorý opečiatkuje postavu pootočenú okolo bodu, ktorý je stredom obrázka.</p>	<p>Vlastný blok, ktorý vedľa seba 10-krát opečiatkuje postavu.</p>

NÁHODA: PEČIATKUJEME A PRESÚVAME SA

Úloha 1 Preskúmajte projekt **hviezdy**.

Úloha 2 Zakrúžkujte, ktoré udalosti sa používajú v projekte.

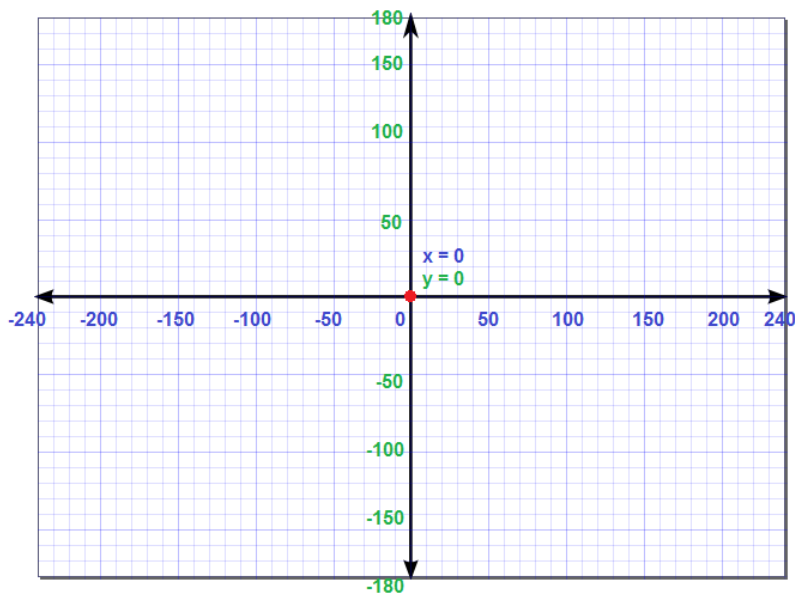


Úloha 3 Preskúmajte blok náhodne. Navrhňte, aké hodnoty má blok náhodne generovať, pre blok **súradnice x, y** ale tak, aby sa generovali hviezdy **len v pravej** časti scény.

Dopíšte do obrázka!



Označte časť scény, na ktorej sa objavili hviezdy.



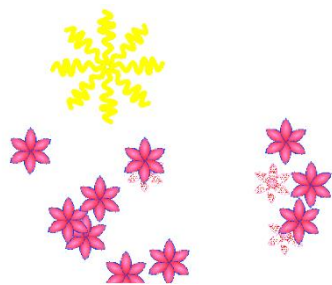
Úloha 4 Doprogramujte projekt tak, aby sa pri zmene hodnoty premennej **pohyb** postava hviezdy pohybovala dopredu o príslušný počet krokov. Usporiadajte príslušné bloky.



Úloha 5 Otvorte si projekt **kvety**.

Nachádzajú sa v ňom 3 postavy – 2 lupene a 1 lúč.

- Pre postavu **lupeň1** je vytvorený nový blok **kvet**. Doprogramujte projekt tak, aby sa vykresľoval kvet z lupeňov pomocou bloku kvet na 10 náhodných miestach v dolnej časti scény.
- Pre postavu **lupeň2** doprogramujte nový blok **kvet2**. Tento nový blok použite v programe, ktorý bude vykresľovať na náhodných miestach 3-krát kvet, ale iba v dolnej časti scény.
- Pre postavu **luc** vytvorte nový blok **slnko** a použite ho v programe tak, aby sa vykreslilo slnko na náhodnom mieste v hornej časti scény.

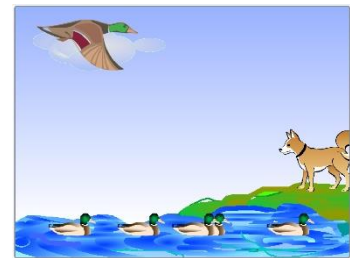


Môžete do svojho projektu pridať aj pozadie, alebo si nakresliť vlastné pozadie.

Úloha 6 Otvorte si pripravený projekt **kacicky**.

Naprogramujte pohyb postáv podľa nasledujúceho scenára:

- na náhodných miestach v dolnej časti scény (na vode) opečiatkujte 5-krát kačku
- nech sa postava pes objaví 2 sekundy po začiatku programu a zabreše
- po zabrechaní psa kačka odletí na náhodné miesto na oblohu



Úloha 7 Otvorte si pripravený projekt **kacicky_a_pes**.

Doprogramujte ho tak, aby po jazierku **plávala** len 1 kačička na náhodné miesta jazierka. Keď sa zjaví pes, vyletí hore nad jazierko.

SAMOHODNOTIACI TEST

Vyberte správny popis ovládania pre príslušné bloky príkazu.

skoč na x: náhodne od 0 do 240 y: náhodne od -180 do 180

Postava skočí náhodne do

- ľavej časti scény
- pravej časti scény
- hornej časti scény



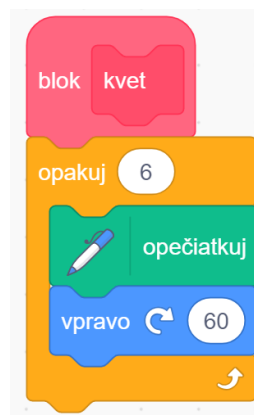
Postava, ktorá vykonáva tieto bloky a prijme správu:

- zmení svoju farbu a premiestni sa na náhodnú pozíciu
- zmení svoj kostým a bude kĺzať na náhodnú pozíciu
- zmení svoj kostým a bude vodorovne kĺzať na pozíciu v hornej časti scény

Vytvorili sme nový blok **kvet**.

Čo robí?

- pomocou postavy nakreslí kvet tak, že skočí na náhodné miesto a opečiatkuje sa
- niekde na scéne opečiatkuje 6 kvetov
- pomocou postavy nakreslí kvet tak, že sa 6-krát opakovane opečiatkuje a otočí sa nejaký uhol



PROGRAMUJEME VLASTNÚ HRU

Úloha 1 Zahrajte hru **Choppy Knight**.

Zapíšte, koľko postáv je v projekte.

Kde je definované pozadie jednotlivých úrovní hry?

Ktorá postava reaguje na zelenú zástavku?

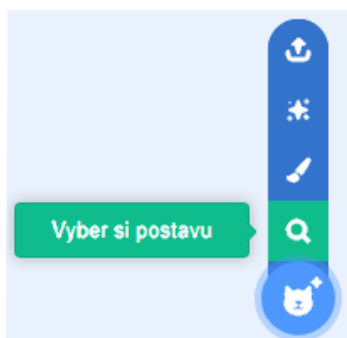


Úloha 2 Vytvorte hru **Lov na syr**, v ktorej:

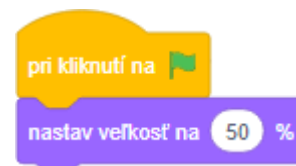
- bude postava myšky, ktorú budeme ovládať pomocou klávesov – šípok
- bude labyrint, myška nemôže prejsť cez stenu
- na náhodnom mieste scény sa objaví syr
- hra skončí, keď sa myška dostane k syru

1. Ovládanie postavy myšky pomocou šípok

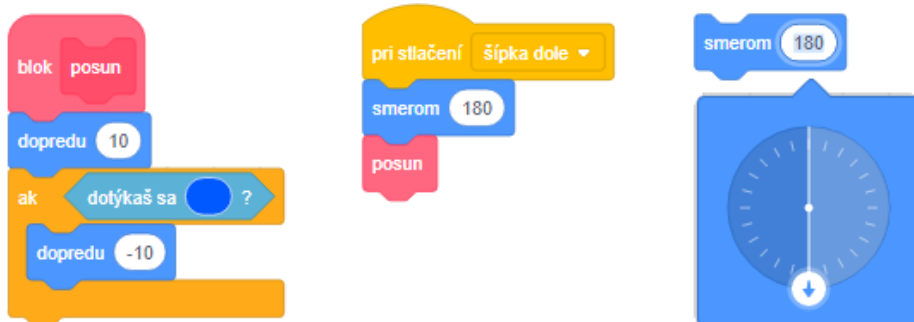
Vytvorte nový projekt, zrušte postavu mačky a vytvorte novú postavu **Mouse1** výberom z ponuky postáv Scratch



Pri štarte programu môžete zmenšiť veľkosť myšky



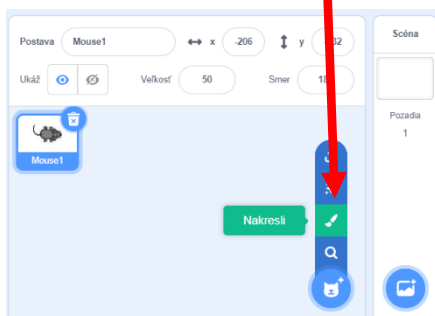
Úloha 3 Pre každý smer pohybu postupne naprogramujte ovládanie myšky pomocou klávesov – šípok. Vytvorte nový blok posun, v ktorom sa bude myška posúvať o nejaký krok. Na obrázku vidíte, bloky, ktoré zabezpečia pohyb myšky smerom hore (smerom 0).



Podobne doprogramujte aj pohyb pre ostatné klávesy – šípky dolu a vpravo. Nezapudnite pri stlačení ďalších šípok využiť už hotový blok **posun**.

2. Príprava hracej plochy: nakreslite postavu, ktorá bude mať obrázok labyrintu.

Otvorte kresliaci editor:



Nakreslite modré čiary labyrintu:

nastavte hrúbku a farbu pera

ak kreslíte so stlačeným klávesom Shift, budú sa kresliť rovné čiary

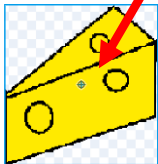
3. Doplňte podmienku pre pohyb myšky tak, aby myška neprešla cez stenu labyrintu:

```
blok posun
dopredu 10
ak dotýkaš sa [blue circle icon] ?
dopredu -10
```

Všimni si, že podmienku sme doplnili do bloku **posun**. Keďže sa tento blok využíva pri pohybe **pre všetky šípky**, myška reaguje na stenu vždy.

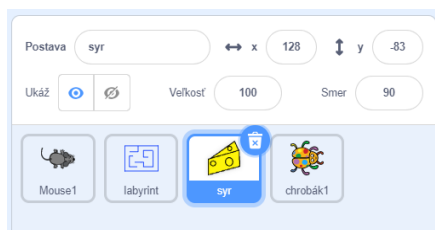
4. Syr, jeho náhodné objavenie sa, presun pri dotyku s myškou a zmena skóre.

Vytvorte novú postavu, nakreslite jej obrázok – syr. Pri kreslení dávajte pozor, aby ste jej stred umiestnili do stredu kresliacej plochy. V kresliacom editore si všimni si terčičk a umiestni syr tak, aby bol v strede syra.



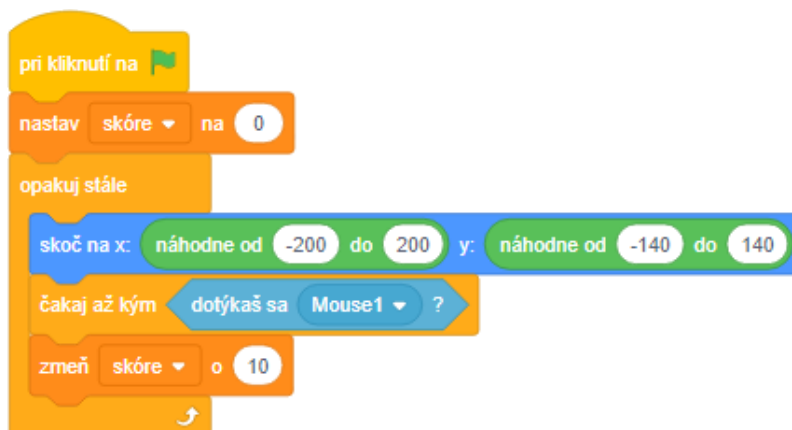
Naprogramuj, aby sa postava **syr** náhodne objavila na scéne.

Poloha postavy na scéne je určená pomocou súradníc. Aktuálnu polohu postavy vidíme v informáciách o postave. Preneste syr na ľavý okraj a pozrite si hodnoty pre x a y.



Náhodné umiestnenie syra naprogramujeme pri stlačení zelenej zástavky a čakanie na to, kým sa ho dotkne myška. Aby sme mohli počítať skóre, vytvorte premennú **skore** a doprogramujte jej zmenu pri stretnutí myšky a syra.

Celý **program pre syr** môže vyzeráť napríklad takto:

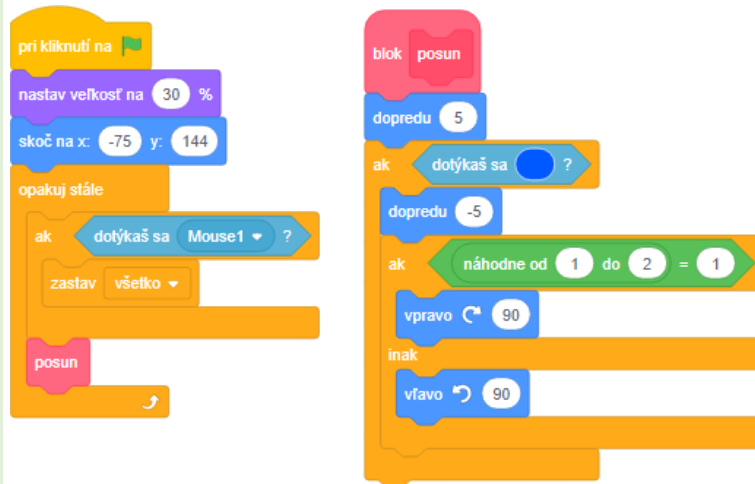


5. Invázia chrobákov

Vyberte postavu chrobáčka a umiestnite ho na konkrétne miesto scény.

Naprogramujte jeho správanie napr. tak, aby sa pohyboval a po narazení na stenu aby sa náhodne otočil, buď doprava alebo doľava. Ak sa dotkne s myšky, hra sa skončí.

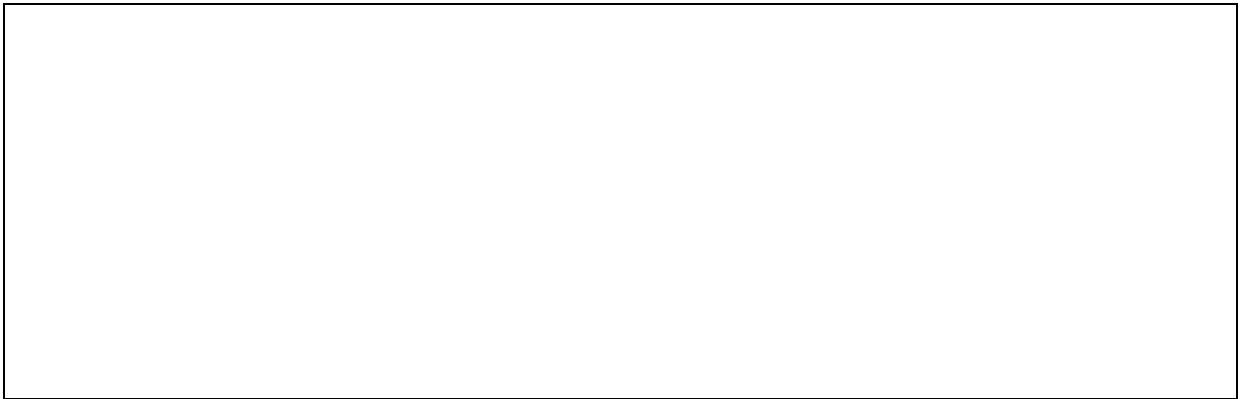
Program pre **jedného chrobáčka** môže vyzeráť aj takto:



Ďalších chrobáčikov vytvorte kopírovaním postavy chrobáka. Nezabudnite ich pri štarte programu umiestniť na iné miesta scény.

Napíšte, aké vylepšenia by ste chceli do svojej hry naprogramovať:

2. Aká je víťazná stratégia hry NIM 13 pre hráča, ktorý ťahá ako druhý v poradí?



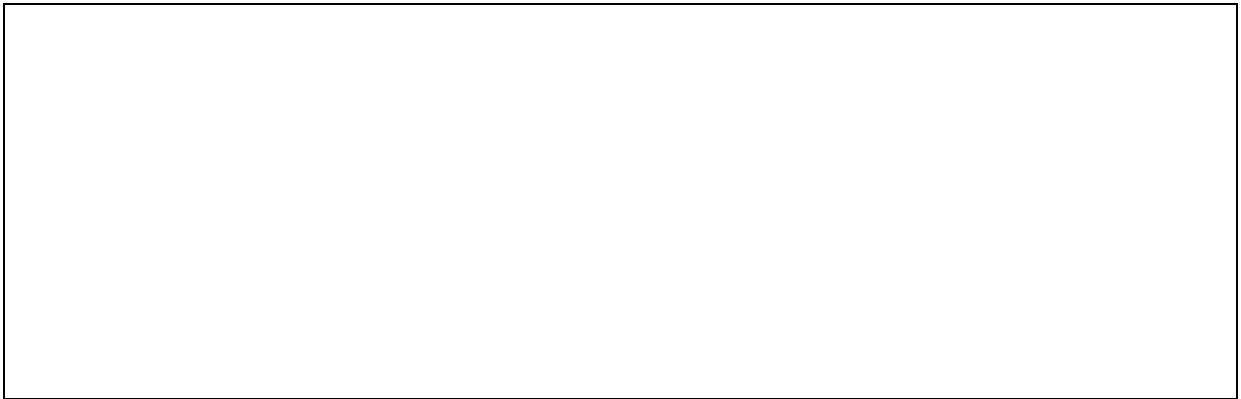
3. Ako sa bude ovládať Tvoja hra?



4. Naprogramuj hru NIM 13 pre dvoch hráčov.

5. * Naprogramuj hru NIM 13 s víťaznou stratégiou pre počítač.

2. Aká je víťazná stratégia hry NIM 13 pre hráča, ktorý ťahá ako druhý v poradí?

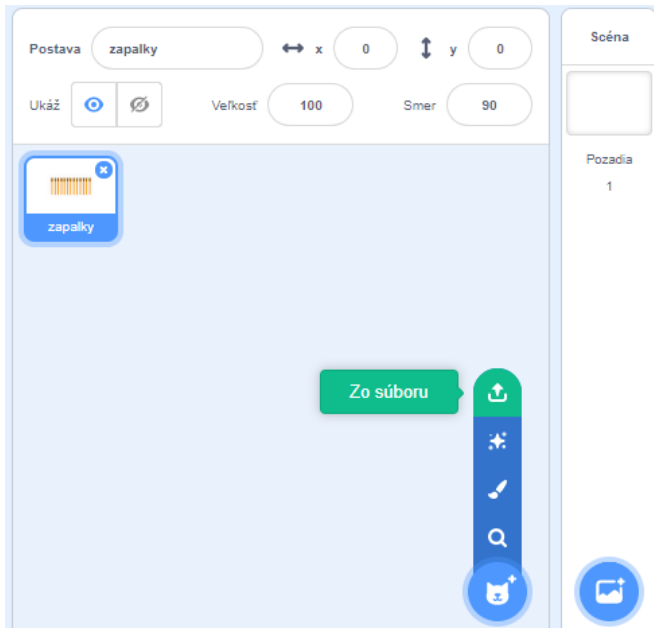


3. Zapiš, ako sa bude ovládať Tvoja hra?

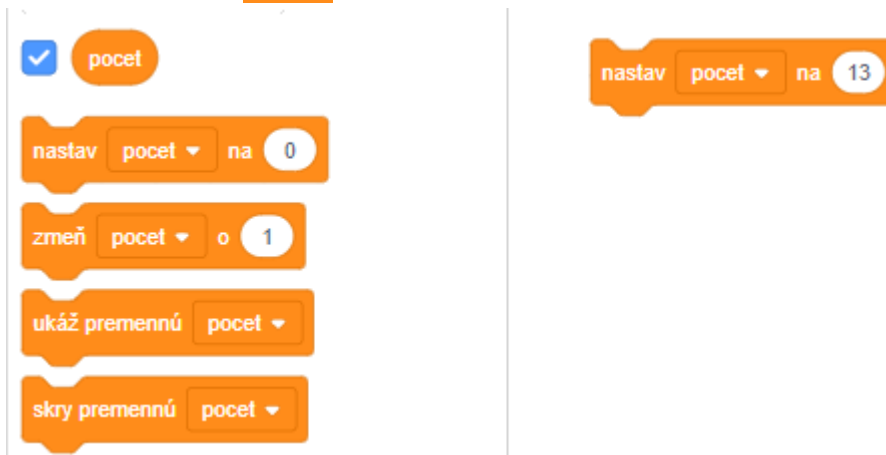


A. Naprogramuj podľa postupu hru **pre dvoch hráčov**, ktorá sa bude hrať tak, že hráči budú klikať na zápalky.

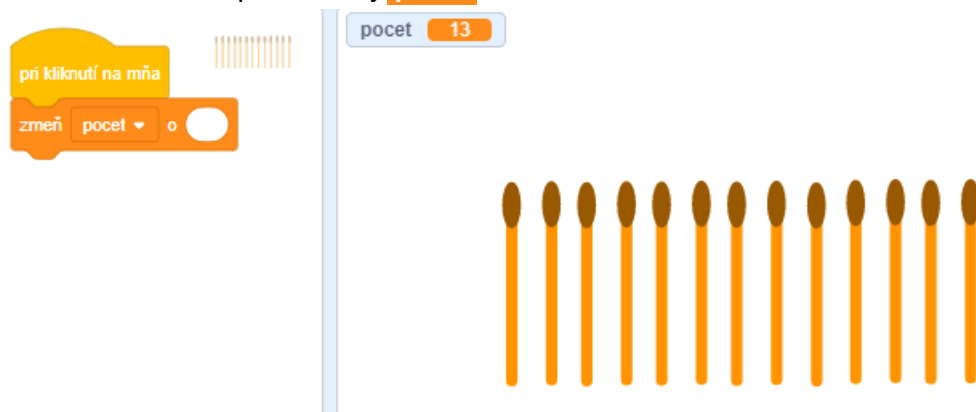
Vytvor postavu a vyber pre ňu obrázok so zápalkami **zapalky.sprite3**.



B. Vytvor premennú **pocet**. Daj do nej hodnotu 13.

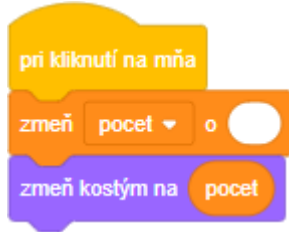


C. Akú hodnotu treba doplniť do pripravených blokov tak, aby sa po kliknutí na zápalky **zmenšila** hodnota premennej **pocet** o 1.



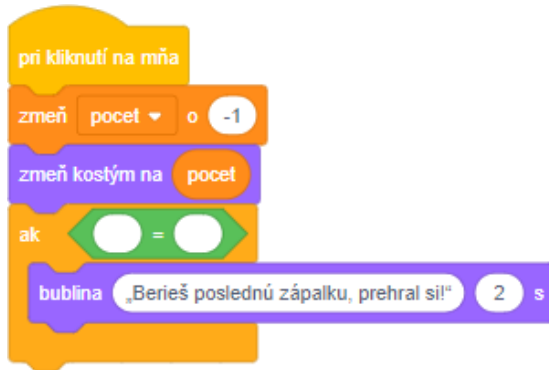
Vyskúšaj klikať a pozoruj hodnotu premennej **pocet**. Ak je to potrebné, znovu nastav jej hodnotu na 13.

D. Priprav bloky pre zápalky tak, aby ich kostým zobrazoval hodnotu podľa premennej **pocet**.



E. Pri kliknutí na zelenú zástavku nastav hodnotu premennej **pocet** na 13 a zobraz kostým s 13 zápalkami.

F. Dopln' do programu časť, v ktorej zistíš, či hráč berie poslednú zápalku. Ak áno, vypíš: „Berieš poslednú zápalku, prehral si!“.



Teraz sa zahraj svoju hru s kamarátom

- dohodnite sa, kto bude začínať
- každý klikne toľkokrát, koľko zápaliek chce zobrať
- potom dá myš kamarátovi

Nezabudni, že Ty musíš dávať pozor, aby kamarát neklikol a nezobral viac ako 3 zápalky.


A kamarát **musí kontrolovať**, aby si Ty nezobral viac ako 3 zápalky.

Označ X čo už vieš

- hrať hru NIM13 so zápalkami
- poznám víťaznú stratégiu pre hráča, ktorý ide ako druhý v poradí
- naprogramovať v Scratch hru pre dvoch hráčov, ktorí klikajú na zápalky a počítač ukazuje, koľko zápaliek ešte ostalo
- v mojej hre počítač zistí a vypíše, kto vyhral

BLUDISKO

1. Podľa **ukážky** a diskusie so spolužiakmi k projektu **11_Bludisko_ukazka** odpovedz na otázky:

Kde je nakreslené bludisko?	
Aký kostým má postava, ktorá sa v bludisku pohybuje?	
Po stlačení ktorých klávesov sa postava pohybuje?	
Cez akú farbu lopta nedokáže prejsť?	
<p>Na obrázku je časť programu, ktorá zabezpečí, že pri stlačení šípky vpravo postava nedokáže prejsť cez zelenú farbu.</p> <p>Napiš, čo sa v programe deje.</p>	 <p>Postava sa najprv posunie v smere x o 10 krokov.</p> <p>Potom pomocou príkazu ak zisťujeme, či sa dotýka zelenej farby.</p> <p> Ak sa dotýka zelenej, postava sa posunie v smere x o -10 krokov.</p>
Aký tvar majú postavy, ktoré po stretnutí s loptou zmiznú?	
Aká hodnota sa zobrazuje vpravo hore?	
Kedy hra skončí?	

2. Priprav svoj projekt, nakresli si vlastné bludisko, navrhni svoju vlastnú postavu a naprogramuj jej pohyb tak, aby neprechádzala stenami. Naprogramuj aj to, aby postava povedala, že našla východ z bludiska.

3. Doplň do bludiska ďalšie postavy, ktoré musí pohyblivá postava "pozbiarať". Vpravo hore zobrazuj číslo, ktoré určuje, koľko predmetov už postava pozbierala.
4. Uprav správu na konci hry tak, aby postava povedala, či pozbierala všetky predmety.

Rozširujúce úlohy

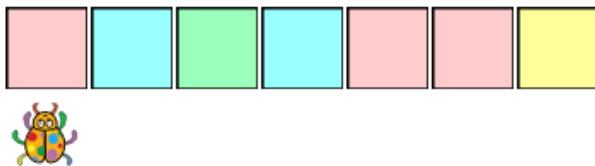
5. Doplň do bludiska postavy – tenké zelené steny, ktoré sa budú v nejakom čase zobrazovať a skrývať.
6. Doplň na vhodné miesta bludiska postavy, pri stretnutí s ktorými sa presunie lopta znovu na začiatok.
7. Keď sa lopta dostane do cieľa a pozbiera všetky predmety, zobraz ďalší labyrint. Obrázok vytvor ako ďalšie pozadie scény.

ŠTVORČEKOVÁ SIEŤ

1. Otvor projekt **12_Stvorcekova_siet**. Preskúmaj v ňom blok **riadok**, ktorý vykreslí vedľa seba do jedného riadku 7 štvorcov. Odpovedz na otázky.

otázka	odpoveď
Aký kostým má postava, ktorá vytvára štvorce? Koľko má kostýmov?	
Ktorý príkaz zabezpečí, že sa štvorce vykreslia?	
Na akom mieste Scény začína postava pri kreslení riadku?	
Aký smer má postava? Je viditeľná?	
Mení sa smer postavy počas vykresľovania riadku?	
Kde skončí postava po vykreslení štvorcov v riadku?	
Zapíš príkaz, ktorým sa postava pohybuje po opečiatkovaní jedného štvorca.	

2. Navrhni pohyb **lienky** pomocou klávesov – šípok. Na pohyb lienky dopredu o 50 krokov navrhni vlastný blok **posun**.



Po stlačení zelenej zástavky zobraz lienku **50 krokov pod** vykresleným riadkom štvorčekov.

3. Dopln' program tak, aby lienka do premennej **zlta** spočítavala, koľkokrát pri svojom pohybe stúpila na žltý štvorček.

otázka	odpoveď
Na ktorom mieste programu je potrebné nastaviť hodnotu premennej na 0?	
Kedy sa má zvýšiť hodnota premennej zlta ?	

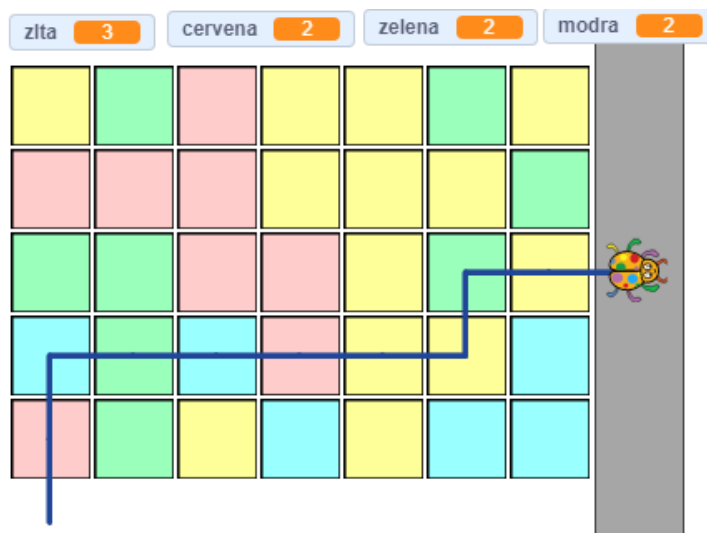
4. Dopln' program tak, aby lienka do premenných **cervena**, **zelena** a **modra** spočítavala, koľkokrát pri svojom pohybe stúpila na červený, zelený a modrý štvorček.
5. Do pozadia scény nakresli tmavosivú cieľovú čiaru. Dopln' program tak, aby lienka povedala: "som v cieľi", keď sa pri svojom pohybe dostane na túto čiaru.
Spúšťaj program niekoľkokrát a pohybuj lienkou tak, aby pri svojom pohybe chodila po štvorčekoch (a nie po bielej ploche), ale tak, aby počet zelených a modrých štvorčekov, na ktoré stúpi, bol rovnaký.
Pohybuj lienkou tak, aby na žiadny štvorček nestúpila viackrát.



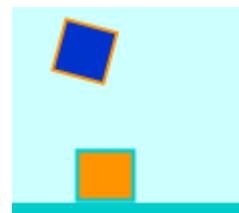
6. Doplň program tak, aby lienka vypísala správu "som v cieľi", **iba vtedy**, ak sa rovná počet zelených a modrých políčok, na ktoré počas svojho pohybu stúpila.

Rozširujúce úlohy

1. Doplň program tak, sa nakreslila hracia plocha s 5 riadkami.
2. Lienku zobraz na bielej ploche pod riadkami. Do bloku **posun** doplň podmienku, aby lienka počas svojho pohybu nemohla vyjsť do bielej plochy. Využi riešenie, pri ktorom sa lienka pôjde **najprv pozrieť** na miesto, na ktoré má prejsť a ak je tam biela plocha, vráti sa **naspäť**.
3. Zabezpeč, aby lienka **kreslila** za sebou tmavomodrú čiaru. Doplň program, aby sa lienka pri pohybe vždy vrátila, ak by sa mala dostať na políčko, na ktorom už bola. Pri riešení využi, že lienka pri pohybe najprv pôjde „pozrieť“ na políčko, na ktoré má prejsť. Ak na tom políčku modrá čiara **nie je** (teda na tomto políčku nikdy predtým nebola), vráti sa naspäť na svoje pôvodné políčko a na nové políčko prejde znovu, ale so zapnutým perom.



PRESKAKUJ



1. Podľa ukážky a diskusie so spolužiakmi k projektu **13_Preskakuj_ukazka** odpovedz na otázky a vypracuj nasledujúce úlohy.

Čo je nakreslené ako pozadie scény?	
Aké postavy sú v projekte?	
Po stlačení ktorého klávesu modrý štvorček vyskočí?	
Akými príkazmi sme zabezpečili vyskočenie a spadnutie modrého štvorčeka?	
Ktorý príkaz zabezpečí, že sa modrý štvorček pri vyskočení otáča?	
Ako sme zabezpečili, že modrý štvorček stúpa pomalšie a padá rýchlejšie?	
Ktorým smerom je natočený oranžový štvorček?	
Ktorý príkaz zabezpečuje, že sa oranžový štvorček pohybuje donekonečna?	
Zapíš podmienku, pri splnení ktorej sa oranžový štvorček presunie znovu k pravému okraju obrazovky?	
Kedy hra skončí?	

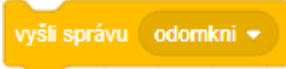

2. Do tabuľky nižšie napíš, ako chceš projekt dokončiť.
Uvažuj napr. o tom, kedy má program skončiť, čo navrhnúť do programu, aby bola hra pre hráča náročnejšia, ako by sa hra zmenila, keby sa modrý štvorček pohyboval pomocou klávesov doprava/doľava, ako vypisovať dĺžku trvania hry...

Zapíš svoje návrhy na dokončenie hry
Uveď, čo budeš v programovaní využívať

ODOMKNI



- Podľa ukážky a diskusie so spolužiakmi k projektu **14_Odomkni** odpovedz na otázky a vypracuj nasledujúce úlohy.

Aké postavy sú v projekte?	
Pomocou akých klávesov sa mačka hýbe?	
Čo je cieľom hry?	
Ako sa môže mačka dostať k darčeku?	
Kedy sa krabica skryje?	
Zapíš príkaz, ktorý umožňuje zistiť, že sa mačka dotkla kľúčika. Kedy by sa mal tento príkaz vykonávať?	
Keď sa mačka dotkne kľúčika, pošle správu s menom  . Na túto správu reaguje kľúčik a aj krabica. Napíš slovami, akým spôsobom tieto postavy reagujú?	

- Naprogramuj podobný projekt ako **14_Odomkni**. Môžeš vymyslieť aj iný scenár, napr. kľúčik môže odomkať časť steny labyrintu alebo dvere...

Označ x čo už vieš

- navrhnuť svoju vlastnú hru, v ktorej bude postava reagovať na správy
- vybrať si postavu z ponúkaných postáv prostredia Scratch a pohybovať ňou pomocou šípok
- ak nastane nejaká udalosť, poslať správu inej postave
- pripraviť scenár, v ktorom postava reaguje na správu, napríklad tým, že sa skryje, alebo že niečo povie

Rozšiřujúce úlohy

3. Naprogramuj hru **Tréning**, v ktorej dievčatko trénuje svojho psíka:
po stlačení zelenej zástavky dievčatko hodí psíkovi loptu, ktorá sa presunie na náhodné miesto v hornej časti scény
po kliknutí na dievčatko, dievčatko povie „Prines!“ , pozri obr. vľavo
keď psík začuje povel „Prines!“ , natočí sa k lopte a prejde ku nej, zoberie ju, vráti sa k dievčatku, ktoré ho pochváli, pozri obr. vpravo



4. Priprav projekt **Preteky**, v ktorom sa po stlačení zelenej zástavky prostredia Scratch všetky chrobáčky presunú k ľavému okraju, teda sa pripravujú na štart
po stlačení čiernobielej štartovacej zástavy (postavy s tvarom zástavy) všetky chrobáčky naraz vyrazia smerom k cieľu, t. j. v každom kroku sa chrobáčikovi vygeneruje náhodná dĺžka kroku, o ktorú sa posunie
ak sa hociktorý chrobáčik dostane do cieľa, povie správu „vyhral som!“ a on a aj všetky ostatné chrobáčky sa prestanú hýbať



DUPLIKUJ

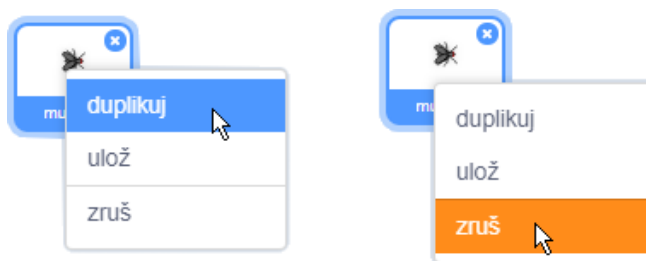
1. Podľa ukážky a diskusie so spolužiakmi k projektu **15_chytaj_muchy** odpovedz na otázky a vypracuj nasledujúce úlohy.

Aké postavy sú v projekte?	
Aký majú postavy kostým?	
Sú všetky postavy v projekte rovnaké?	
Akým spôsobom sa pohybuje každá mucha?	
Na akú udalosť reaguje každá mucha?	
Ako na túto udalosť mucha reaguje?	
Je správanie všetkých múch rovnaké?	
Kedy hra končí?	
Ako najrýchlejšie pridáme do hry ďalšiu muchu s rovnakým obrázkom aj s rovnakým správaním?	

2. Naprogramuj svoj vlastný projekt **Chytaj muchy**, v ktorom bude úlohou hráča kliknutím na muchu chytať muchy.

Uvedom si, že najrýchlejšie *takýto projekt vytvoriť* je navrhnuť **najprv jednu muchu**, ktorej pripravíš kostým aj správanie a potom ju pomocou **duplikuj** „rozmnožiť“ toľkokrát, koľko múch chceme mať v hre.

Ak zistíš, že si postave zabudol naprogramovať nejaké *správanie*, pomocou opakovaného mazania zruš okrem postavy **mucha1** všetky ostatné *muchy*, doprogramuj, čo potrebuješ a postavu znovu **duplikuj**.



Nástroje duplikuj a zruš.

3. Navrhni projekt **Balóny šťastia**, v ktorom budú k oblohe letieť farebné balóny tak, ako to bolo v ukážke.

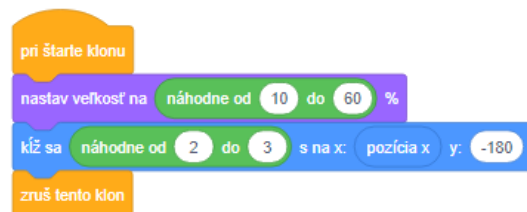
4. Naprogramuj projekt **Dážď**, v ktorom bude „pršať“, t.j. z hornej časti scény budú padať kvapky. Duplikovaním kvapky dosiahni, aby bol dážď čo najprirodzenejší.
5. Navrhni vlastný projekt, v ktorom využiješ duplikovanie postáv. Do tabuľky zapíš, aký projekt budeš programovať a akú postavu budeš duplikovať.

Rozširujúca úloha

6. (riešený príklad pre žiakov so záujmom o programovanie)

V projekte Dážď využijeme namiesto duplikovania postáv **klonovanie** jednej postavy:

- v projekte vytvor jednu postavu s tvarom kvapky, so správnym natočením smerom nadol
- pri kliknutí na zelenú zástavku sa postava stále presúva na náhodnú pozíciu v hornej časti scény a tam sa naklonuje
- potom vyvolaj udalosť pri štarte klonu
- naprogramuj udalosť pri štarte klonu, v ktorej si klon zmení veľkosť a bude sa (tak ako v projekte Dážď) kĺzať k dolnej časti scény
- po skončení kĺzania sa klon zmaže (toto je dôležitá časť programu, pretože v Scratch môže existovať len 400 klonov)
- uvedom si, že jediná postava kvapky (ktorá nie je klon) sa medzitým presunula niekoľkokrát na iné miesta, kde vytvorila ďalšie svoje klony

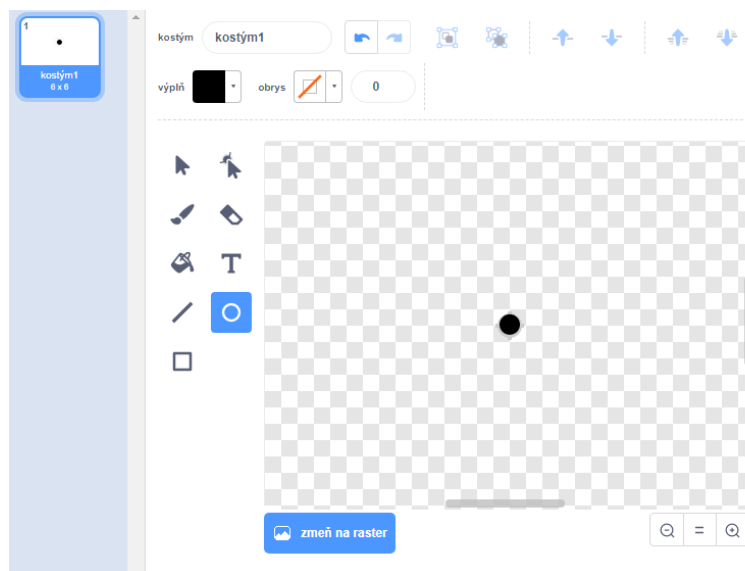


KRESLI ZVUK

1. Začni nový projekt.
2. Nájdi premennú, ktorá obsahuje hlasitosť zvuku z mikrofónu. Zobraz premennú, rob hluk a sleduj hodnoty, ktoré premenná nadobúda.
Zapíš do tabuľky, akú najmenšiu a najväčšiu hodnotu mala premenná.

premenná	najmenej	najviac
hlasitosť		

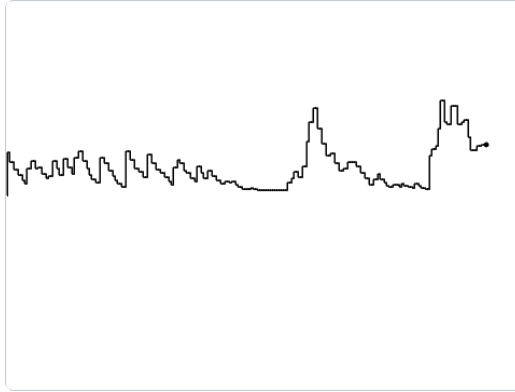
3. Zruš mačku a vytvor novú postavu, daj jej meno **kresli**. Postave nakresli kostým malej čiernej bodky



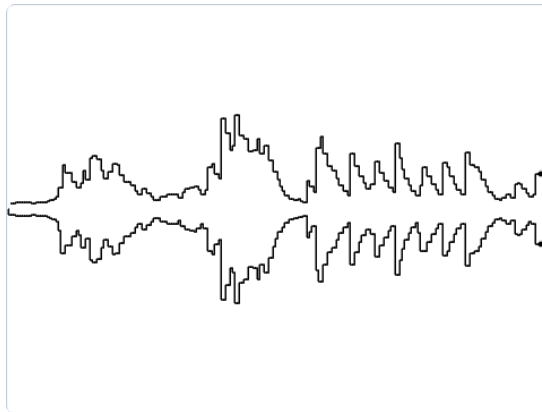
4. Zisti a zapíš do tabuľky, aká je x-ová súradnica ľavého a pravého okraja scény.

ľavý okraj	pravý okraj

5. Vytvor program, v ktorom budeš stále presúvať postavu na y-ovú súradnicu podľa aktuálnej hodnoty premennej **hlasitosť**. Súradnicu **x** meň o malé hodnoty, napr. o 1 alebo 2.
Vyskúšaj projekt, rob hluk tak, aby Ti vyšiel podobný obrázok:



6. Doprogramuj, aby postava po prejdení na pravý okraj scény zmazala doterajšiu kresbu a začala vykresľovanie znova od ľavého okraja scény.
7. Duplikovaním postavy **kresli** vytvor novú postavu, Scratch ju nazve **kresli1**. Uprav novej postave program tak, aby kreslila presne to isté, čo postava **kresli**, len zvislo symetricky prevrátené tak, ako vidíš na obrázku:



8. Obe postavy ktoré kreslia skry, aby ich nebolo vidieť, keď budú kresliť a vhodne ich premenuj. Napr. *kresli hore* a *kresli dole*.

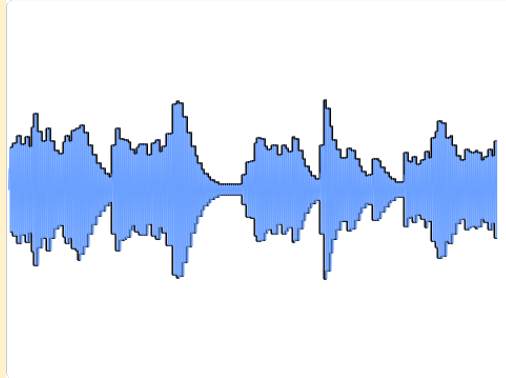
Označ x čo už vieš

- zobrazíť premennú, ktorá zachytáva zvuk a zistiť, aký najväčší hluK viem urobiť
- zmeniť hodnotu y-ovej súradnice postave podľa toho, ako veľký hluK mikrofón zachytil
- keď sa zvuk vykreslí na celú scénu, zmazať scénu a vykresľovať zvuk znova z ľavej strany
- podľa x-ovej osi vykresľovať symetricky hlasitosť

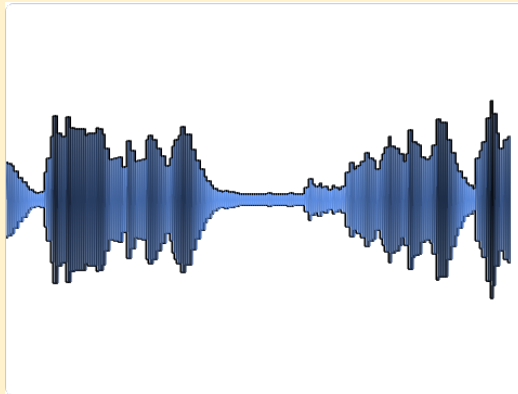
Rozširujúce úlohy

9. Naprogramuj vyfarbovanie vnútra tak, ako vidíš na obrázku, napr. modrou farbou. Vyfarbovanie nech zabezpečí nová postava, ktorá bude robiť farebné čiary medzi postavami kresli hore a kresli dole.

Pomôcka: Pre novú postavu využij pozície, ktoré majú postavy, ktoré kreslia.



10. Zabezpeč, aby sa farby vykresľovali v závislosti od výšky hlasitosti. Keď je zvuk slabý, bude sa kresliť svetlomodrou a keď je silný, bude sa kresliť tmavomodrou.



STE(A)M: ANIMUJEME CHEMICKÉ REAKCIE

1. Podľa ukážky a diskusie so spolužiakmi k projektu **19_anim_chem_reakcie** odpovedz na otázky a vypracuj nasledujúce úlohy.

Aký dej v simulácii zobrazuje?	
Aké molekuly sú v projekte na začiatku?	
Aké molekuly sú v projekte na konci?	
Aké molekuly vznikajú?	
Chemickú reakciu zapíš vzorcom.	
Je počet molekúl rovnaký?	
Keby sme zvýšili počet oboch molekúl na dvojnásobok, dostali by sme rovnaký výsledok?	
Ako by mohla vyzeráť časť programu, ktorá zabezpečuje hýbanie molekúl?	
Kedy simulácia skončí?	

2. Navrhni rovnaký projekt, v ktorom budeš simulovať túto reakciu.
3. Vytvor podobnú simuláciu pre reakciu **KOH + HBr**.
4. Navrhni a naprogramuj svoj vlastný projekt, v ktorom budeš simulovať nejakú chemickú reakciu. Vyber si takú reakciu, ktorú poznáš z chémie. Navrhni dizajn a funkčnosť programu. Obráť sa **na učiteľa chémie** a uisti sa, že tvoj program je navrhnutý správne a chemická reakcia aj v skutočnosti prebieha tak, ako ju chceš simulovať vo svojom programe.