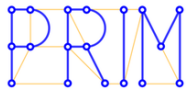




Proměnné

KAPITOLA 10

JIŘÍ VANÍČEK, INGRID NAGYOVÁ, MONIKA TOMCSÁNYIOVÁ



CO BUDEME DĚLAT

- počítat skóre
- hádat čísla
- hrát kostky
- krmit žraloka
- jezdit po autodráze

CO SE ŽÁCI NAUČÍ

- používat proměnné pro uchovávání hodnot
- použít v proměnných čísla a texty
- nastavovat a měnit hodnotu proměnné

NOVÉ BLOKY

nastav __ na __**změň __ o __**

POUŽITÉ PROJEKTY

- *Vesmír*
- *Házení kostkou*
- *Autodráha*

ČASOVÁ DOTACE **4** VYUČOVACÍ HODINY

Sled aktivit:

Chtáme jablkasnímky 3 - 4
20 minut**Počítáme do 100**snímky 5 - 8
25 minut**Měníme rychlost**snímky 9 - 10
20 minut**Myslím si číslo**snímky 11 - 13
25 minut**Hra v kostky**snímky 14 - 16
30 minut**Hra Žralok**snímky 17 - 23
60 minut**Autodráha**snímky 24 - 27
(rozšiřující aktivita)

snímek 3



Chytáme jablka

Otevři si prázdný projekt.

1. Vyber postavě kostým jablka. Vytvoř scénář, aby postava vždy po 1 sekundě skočila na náhodnou pozici.
2. Vytvoř další scénář, v něm když klikneš na jablko, zmizí. Do původního scénáře doplň, aby se vždy na novém místě jablko znovu ukázalo.



3. Vyzkoušej hru. Snaž se jablko „chytit“, kliknout na něj myší. Nastav rychlost skákání jablka, aby hra nebyla nudná. Můžeš také jablko zmenšit.

Aktivita začíná vytvořením nového projektu.

Na následujících dvou snímcích žáci vytvoří jednoduchou hru. Naučí se vytvořit novou proměnnou a využít ji při počítání skóre.

METODICKÉ POZNÁMKY

1. Přípravná úloha.

Žáci mohou vytvořit novou postavu jablka (a kočičku skrýt nebo smazat). Mohou také přidat kočičce další kostým jablka.

2. Druhý scénář, který reaguje na kliknutí na jablko, je důležitý: v dalším snímku právě v něm budeme počítat chycená jablka.

Skrytím jablka předcházíme tomu, že by hráč mohl na jablko kliknout vícekrát za sebou, než skočí jinam, a tím nezaslouženě získat lepší skóre.

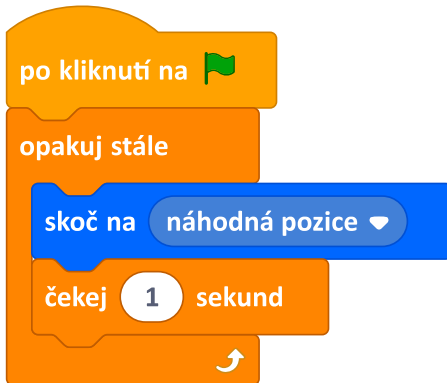
Zvědaví žáci se u správného řešení mohou dotazovat, proč se „plýtvá“ opakováním bloku **ukáž se** po každém skoku jablka (učili se, že chtít ukázat již viditelnou postavu je zbytečné). Odpovědí je, že toto je jednodušší způsob, než stále kontrolovat, jestli je jablko skryto. A naprogramovat, aby jablko skočilo jinam ihned poté, co jej hráč „chytne“, je náročnější, než jablko na chvíli skrýt.

3. Žáci testují vytvořený scénář a zkouší hrát, kliknout na jablko.

Doporučujeme využít režim *Prezentace*, hru lze spustit paporkem.

ŘEŠENÍ ÚLOH

1.



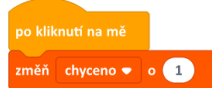
2.



snímek 4

Pamatujeme si jablka

1. Aby si hra pamatovala, kolik jablek jsi chytil, vytvoř si v nabídce **Proměnné** novou proměnnou, nazvi ji **chyceno**.
2. Když klikneš na jablko, spustí se scénář, v němž se číslo v proměnné zvětší o 1. Sestav pro postavu další scénář podle obrázku. Vyzkoušej.



3. Když hraješ znovu, potřebuješ vynulovat počet chycených jablek. Použij blok . Kam jej do scénáře umístíš?

METODICKÉ POZNÁMKY

1. Žáci vytvoří proměnnou **chyceno**. Vytvořená proměnná se svojí aktuální hodnotou se objeví na scéně v poli . Současně se v nabídce bloků objeví dva důležité bloky: pro nastavení hodnoty proměnné a pro její změnu. Proměnná uchovává svoji hodnotu, dokud ji nějaký blok nezmění.
2. Žáci postavě přidají scénář podle instrukcí na snímku – použijí blok **změň chyceno o 1**. Scénář vyzkouší a otestují – při každém kliknutí na jablko by se měla hodnota proměnné **chyceno** zvyšovat.
3. Žáci mají objevit, že vynulovat počet chycených jablek je třeba na začátku hry (po praporku).

ŘEŠENÍ ÚLOH

3. Příklad správného řešení. Výsledná hra obsahuje dva scénáře pro jablko; jeden pro pohyb postavy a druhý pro počítání chycených jablek.




snímek 5



Objevujeme proměnné

Otevři si prázdný projekt.

1. V nabídce **Proměnné** vytvoř novou proměnnou **číslo**. Poté si prohlédni bloky, které se objevily.
2. Vyzkoušej, co dělá blok **změň číslo o 1**.
Opakovaně na něj klikej.
Vyzkoušej také blok **nastav číslo na ___**.
Vysvětlí, co tyto bloky dělají, k čemu slouží, kde se mohou použít.
3. Vyzkoušej blok  .

Žáci pracují v novém projektu proto, aby se jim nepletly vytvořené scénáře pro hru o jablko.

METODICKÉ POZNÁMKY


1. V těchto úlohách se žáci s proměnnými seznámí podrobněji. Měli by umět **rozlišit** mezi **názvem proměnné a její hodnotou**.

Koncept dvojice bloků **změň** a **nastav** žáci znají například z nastavování tloušťky kreslicího pera nebo změny velikosti postavy. Zde je tento koncept využít ke změnám hodnot proměnných.

2. OTÁZKY UČITELE

1. V proměnné **číslo** je hodnota 7. Jak se změní její hodnota po spuštění bloku **změň číslo o 1**? (o **5**, o **100**, o **-1**)? Zvýší se hodnota proměnné vždy?
2. Hodnota proměnné je nyní 8 a já potřebuji, aby byla 18 (100, 5...). Jaké číslo uvedu v bloku **změň číslo o ___**?
3. Jak se změní hodnota proměnné **číslo** po použití bloku **nastav číslo na 8**?

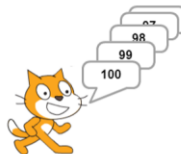
3.

Žáci si vyzkouší, že pomocí bloku  řekne postava aktuální hodnotu proměnné **číslo**.

snímek 6

☞ Počítáme do sta

1. Vytvoř scénář, v němž postava řekne prvních 100 čísel. Postava bude říkat hodnotu proměnné **číslo**, kterou bude zvětšovat.



2. Vytvoř další scénáře, v nichž se bude počítat do sta po dvou, po násobcích deseti.

3. Vytvoř scénář, v němž bude postava počítat pozpátku od 100 do 1.

METODICKÉ POZNÁMKY

1. Žáci použijí blok **bublina číslo** ve scénáři. Žáci mají přijít na to, že vhodný je pevně daný počet opakování (blok **opakuj 100 krát**).

Žáci by měli vyzkoušet scénář vícekrát, aby si uvědomili, že potřebují na začátku nastavit hodnotu proměnné **nastav číslo na 0**.

2. Úloha je modifikací předchozí úlohy. Žáci mají objevit, že nutno upravit velikost změny proměnné **číslo** a také počet opakování. Žáci mění parametry scénáře buď pokusem a omylem, nebo úvahou.

3. Těžší úloha, protože pro počítání „pozpátku“ musíme v bloku **změň číslo** mít parametr záporný. Především je třeba **správně nastavit počáteční hodnotu proměnné**.

Učitel by měl zkontrolovat, zda postava opravdu začala počítání **s hodnotou 100 v bublině** a skončila **s bublinou** hodnoty 1. Změnou pořadí bloků v opakování lze docílit jiné počáteční i koncové hodnoty čísla v bublině. Existuje více správných řešení.

ŘEŠENÍ ÚLOH

1.	2.	3.

snímek 7



Diskutujeme o proměnné

Prohlédni si scénáře. Jaké bude poslední číslo, které postava řekne? Jaká bude po vykonání scénáře hodnota proměnné **číslo**?

a

b

c

d

METODICKÉ POZNÁMKY

Žáci pracují bez počítače, úloha se promítá. Žáci se poradí ve dvojicích a poté diskutují o jednotlivých scénářích. Žáci si mohou nakreslit tabulku, do níž vypíší pro jednotlivé scénáře a) až e) tipy na správnou odpověď.

K jednotlivým scénářům:

a) Žáci musí dávat pozor na pořadí bloků. Důležité je poznat, že hodnota se nejprve řekne a teprve pak se změní v proměnné. Proto jsou na konci obě hodnoty odlišné (viz správná řešení).

Ve scénáři b) se jako poslední řekne hodnota proměnné, proto budou oba výsledky stejné.

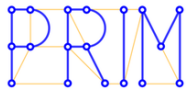
Scénář c) řeší otázku, zda po splnění podmínky se ještě jednou vykonají scénáře uvnitř bloku **opakuj** či nikoliv.

Scénář d) nikdy neskončí a je zde zajímavé diskutovat o tom, jestli má nějakou poslední hodnotu (v proměnné i v bublině).

Pokud se žáci nemohou shodnout, je chybou učitele říci „Je to takto.“ nebo „Toto je správně“. Pokud je potřeba, **je lépe scénáře sestavit a spustit** – v tom případě doporučujeme vložit do scénářů zpoždění pomocí **čekej**.

OTÁZKY UČITELE

- Proč scénář d) nikdy neskončí?
(scénář používá blok **opakuj stále**)
- Jak je možné, že kočička říká pořád 9 i přesto, že scénář e) nekončí?
(hodnota proměnné **číslo** se vždy vrátí zpět na 9 dřív, než bublina znovu zobrazí její hodnotu)
- Jak by se změnila situace ve scénáři c), kdybychom podmínku **číslo** > 10 zaměnili za podmínku **číslo** < 10 nebo **číslo** = 10?



(Při podmínce **číslo** = 10 scénář nikdy neskončí, protože číslo 10 se „přeskočí“, neboť proměnná bude nabývat lichých hodnot 1, 3, 5, 7, 9, 11, 13, ... a postava bude počítat pořád dále.

Pro podmínku **číslo** < 10 opakování ani nezačne, takže postava neřekne nic a scénář skončí s hodnotou **číslo** = 1.)

ŘEŠENÍ ÚLOH

1.

	V bublině bude	V proměnné číslo bude
a	198	200
b	30	30
c	11	11
d	9	--

V úloze e) scénář nikdy neskončí, takže v proměnné **číslo** nikdy nebude nějaká poslední hodnota. Od určité doby se v ní budou neustále střídat hodnoty 9 a 11, v bublině ale číslo 11 nikdy nebude, protože podmínka před bublinou hodnotu proměnné vždy sníží pod 10.


Ovšem poslední hodnota v bublině se určit dá, protože bublina se vypisuje vždy až po kontrole, jestli hodnota nepřekročila 10. V bublině nikdy nebude číslo větší než 10, to znamená, že poslední hodnota, která se bude neustále znovu a znovu zobrazovat, je 9.

snímek 8

Měníme rychlost

Otevři si projekt *Vesmír*.

1. Nech raketu stále létat a odrážet se od okraje scény.
2. Raketě můžeme měnit rychlost létání. Vytvoř proměnnou **rychlost**, která bude určovat, o kolik kroků má raketa letět v bloku **dopředu**.
3. Proměnnou **rychlost** budeme zvětšovat nebo zmenšovat o 1 klávesami [šipka nahoru] a [šipka dolů]. Sestav oba scénáře.



po stisku klávesy šipka nahoru ▼

Ze stránek projektu si žáci otevřou projekt *Vesmír*.

METODICKÉ POZNÁMKY

1. Opakovací a přípravná úloha.
S projektem se žáci setkali v kapitole 7 a tuto úlohu řešili.
2. Žáci si zopakují postup při vytváření proměnné.
Vložení proměnné **rychlost** do bloku **dopředu o __ kroků** se po spuštění scénáře raketa nerozletí, protože počáteční hodnota proměnné **rychlost** je 0. Lze ji nastavit např. na 10 v bloku **nastav rychlost na 10**.
3. Úloha umožní ovládnutí změny rychlosti hráčem.
Proměnná **rychlost** si rychlost pohybu pamatuje a **po stisku klávesy __** se mění hodnotou 1, resp. -1 v bloku **změň rychlost o __**.

ŘEŠENÍ ÚLOH

1.	2.	3.
		

snímek 9



Raketa necouvá



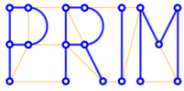
1. Když změníš rychlost na nulovou nebo zápornou, vypadá to divně. Raketa se přeci nemůže zastavit nebo couvat.
Jak bys tuto chybu opravil?
Který scénář se musí změnit?
Jaké bloky k tomu použiješ?
2. Uprav scénář tak, aby **rychlost** nemohla klesnout na nulu nebo záporné číslo.

METODICKÉ POZNÁMKY

1. Při kontrole správnosti předchozí úlohy žáci zjistí, že ubíráním hodnoty může rychlost dosáhnout záporné hodnoty, což způsobí podivné chování rakety (zvláště při odrazu od okraje).
Žáci jsou postaveni před **problém**, jak zařídit, aby se hodnota proměnné nestala zápornou.
Úloha je určena k diskusi. Žáci se mohou nejprve radit ve dvojicích a domlouvat se, proč raketa couvá a jak problém vyřešit.
Důležité je, aby žáci rozuměli tomu, pro jakou aktuální hodnotu proměnné **rychlost** je ještě povoleno ji snižovat (žáci budou navrhnout čísla 1, 0, 2).

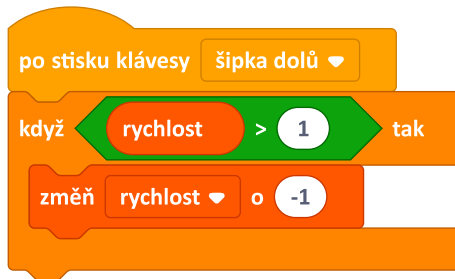
OTÁZKY UČITELE

1. Proč raketa couvá? Jakou rychlostí se při couvání pohybuje?
(Hodnota v proměnné je záporná.)
 2. Který scénář musíš opravit, aby se raketa nezastavila ani necouvala? Jakou úpravu je ve scénáři potřeba udělat?
(je nutné opravit scénář pro klávesu [šipka dolů]; proměnnou **rychlost** zmenšíme pouze v případě, že je větší než 1.)
 3. Jaké bloky k tomu použiješ?
(ke kontrole hodnoty proměnné **rychlost** použijeme blok **když __ tak**)
2. Na základě diskuse žáci scénář upraví. Diskuse je zde ovšem důležitá. Chceme, aby žáci argumentovali, uměli vysvětlit, nejen aby sestavili kód.
Řešení žáci otestují nejlépe v režimu prezentace.



ŘEŠENÍ ÚLOH

2.



snímek 10

Myslím si číslo

Otevři si prázdný projekt.

1. Postava si po spuštění mezerníkem bude myslet nějaké číslo od 1 do 15. V proměnné **myšlené** si jej bude pamatovat. Vyzve hráče, aby hádal.
2. Po spuštění praporkem postava nechá hráče hádat a poté odpověď porovná s myšleným číslem. Když budou stejné, řekne: „Trefa“, když nebudou, řekne: „Vedle“.

Aktivita začíná vytvořením nového projektu.

Žáci sestaví hru, která umožní hráči hádat náhodné číslo. Úloha se také věnuje optimalizaci: zamyslíme se nad počtem pokusů, během nichž lze dané číslo s jistotou uhodnout.

METODICKÉ POZNÁMKY

1. Zde se sestavuje scénář pro start hry. Žáci vytvoří proměnnou **myšlené**, nastaví její hodnotu na náhodné číslo a bublinou „Myslím si číslo od 1 do 15“ postava odstartuje hru hádání.
2. Žáci vytvoří scénář pro hádání čísla. **Na každé kliknutí na praporek** se spustí scénář pro jeden dotaz. Scénář se tak bude spouštět opakovaně. Po uhodnutí čísla se hra znovu odstartuje scénářem z 1. úlohy. Je nutné ověřit správnost scénáře. Žáci musí vyzkoušet **reakci postavy na obě možnosti**, tj. když číslo neuhodnou stejně jako když jej uhodnou. Při ověřování pomůže aktuální hodnota proměnné **myšlené**, která se vypisuje na scéně vlevo nahoře.

OTÁZKY UČITELE

Žáků se zeptáme, jaká čísla jsou „skryta“ v podmínce , aby si udělali konkrétní představu o splnění podmínky.

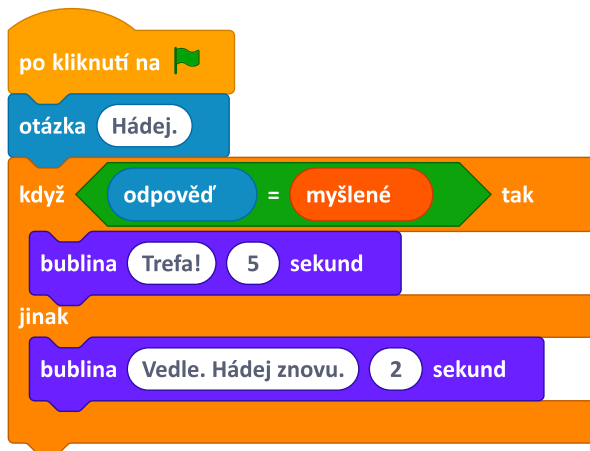
1. Jaká hodnota je nyní v odpovědi? Jaká hodnota je v proměnné **myšlené**?
2. Jsou hodnoty stejné nebo nejsou? Je podmínka splněna?

ŘEŠENÍ ÚLOH

1.



2.



snímek 11

Myslím si číslo pokračování

1. Aby byla hra férová, je třeba skrýt proměnnou **myšlené**.
2. Budeme počítat, na kolik pokusů hráč číslo uhodne. Vytvoř proměnnou **počet pokusů** a do scénáře přidej bloky, které budou při každém hádání přičítat jeden pokus.

počet pokusů 6 TREFA!



3. Když budeme chtít hádat nové číslo, musíme ve scénářích ještě něco upravit. Přijdeš na to?

METODICKÉ POZNÁMKY

1. Hodnota proměnných je na scéně vlevo nahoře implicitně viditelná. To je výhodou při ladění scénáře; při hře je potřeba výpis hodnoty proměnné **myšlené** skrýt – zrušit zaškrtnutí v nabídce **Proměnné**.
2. Je vhodné, aby se blok zvyšující **počet pokusů** o 1, vykonal hned poté, co hráč odpoví – když jej zařadíme na konec scénáře, po dobu zobrazení bubliny bude počet pokusů zobrazen nesprávně.
3. Žáci mají přijít na to, že na začátku hádání je třeba vynulovat proměnnou **počet pokusů**. Je třeba hrát hru včetně úvodního nastavení hry vícekrát, aby žáci zjistili, že počet pokusů zůstává v proměnné z předchozí hry.


ŘEŠENÍ ÚLOH

2.	3.


snímek 12

Myslím si číslo pokračování

1. Postava může hráči při hádání pomáhat. Namísto odpovědi „Vedle“ oznámí hráči v bublině, jestli je hádané číslo větší nebo menší než hráčův tip.



Moc velké.



Moc malé.

2. Hraj a odpověz: Na kolik pokusů jistě uhodneš číslo od 1 do 15? Dokážeš to vždy na 4 pokusy? A na kolik pokusů uhodneš číslo od 1 do 100? Uprav si scénář a vyzkoušej.

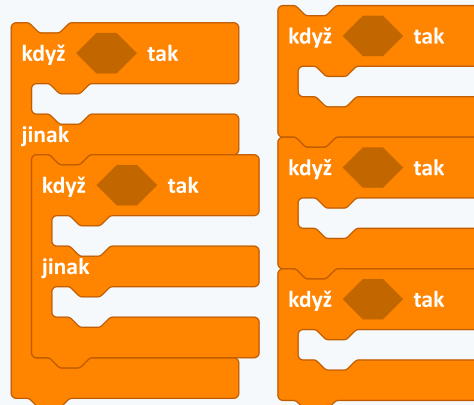
Úlohy jsou určené pro pokročilé žáky.

METODICKÉ POZNÁMKY

1. V úloze se musí scénář rozhodnout mezi třemi možnostmi: $>$, $=$, $<$.

K tomu potřebuje použít buď dvě rozhodování **když __ tak __ jinak** vnořená do sebe, nebo tři rozhodování **když __ tak** za sebou. Je to již obtížnější koncept, ale protože jde o poměrně častou situaci, zařazujeme ji zde pro pokročilé žáky.

Čistě logicky je vhodnější řešení vlevo; preferujeme jej, protože u něj obecně máme vždy jistotu, že každý případ, který může nastat, bude nějak vyhodnocen (právě kvůli přítomnosti slova JINAK). U řešení vpravo pořád můžeme cítit, že nepokryjeme všechny možnosti, že nám nějaký případ „proteče“ všemi podmínkami. U této konkrétní úlohy jsou ale obě řešení rovnocenná.



2. V této úloze nebudeme programovat. Použijeme naprogramovanou simulaci **k hledání optimálního řešení** problému.

Odpověď na otázky souvisí s aplikací známého **algoritmu binárního vyhledávání**, kdy v každém kroku volíme prostřední prvek v hledaném úseku a podle toho, zda je hledané číslo menší nebo větší než tento prostřední prvek, pokračujeme v dalším kroku hledáním v jedné jeho polovině.

Tzv. metoda půlení intervalu každým krokem snižuje počet možností na polovinu (tedy z 15 čísel na 8, pak na 4, na 2 a konečně na 1 možnost).

Není smyslem úlohy tuto teorii žákovi sdělit, aby „se ji naučil“, ale aby sám na takový postup sám přišel. Učitel by měl být více spokojen tehdy, když žák nad problémem přemýšlel, i kdyby

správný postup nevymyslel, než kdyby žák pouze pochopil jeho výklad nebo si dokonce pouze zapamatoval, co mu učitel řekl.

U úloh s jiným počtem čísel žáci snadno upraví parametry v bloku náhodných čísel. Lze se ptát, na kolik kroků uhodneme čísla od 1 do 1 000, do 1 000 000. Výsledky těchto úloh mohou žáky ohromit.

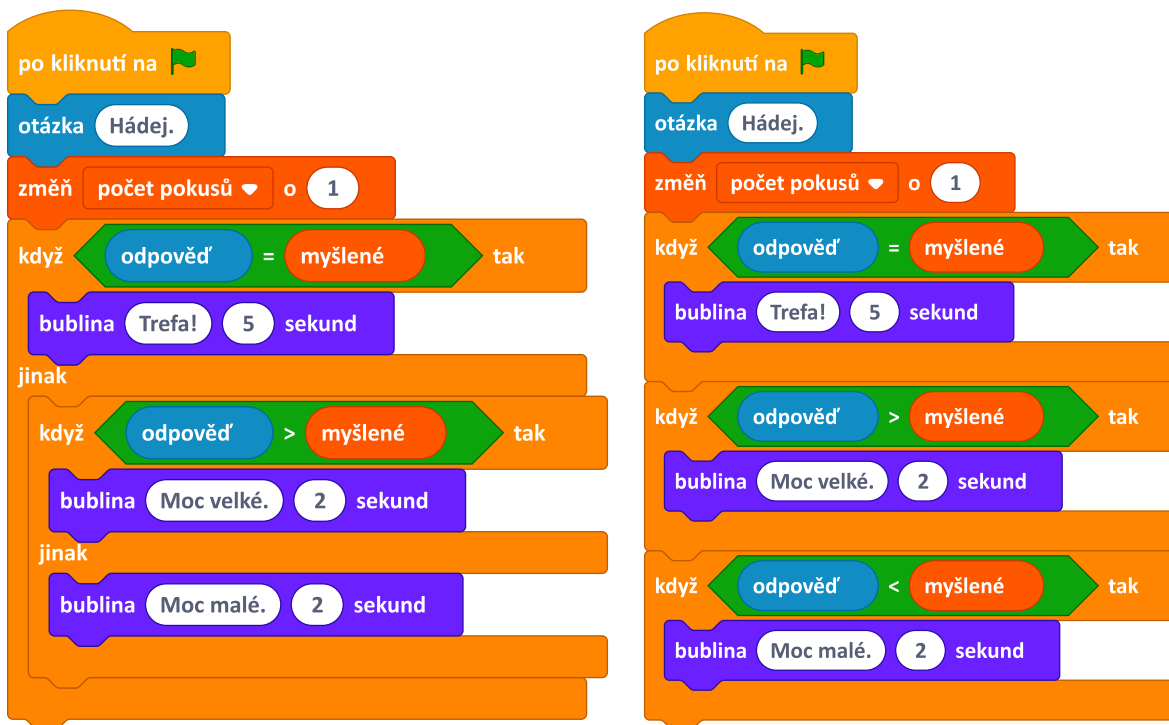
ŘEŠENÍ ÚLOH

1.

Dvě možná řešení:

vnořením podmínek když – tak – jinak

řetěžením podmínek když – tak.



2.

Číslo od 1 do 15 lze vždy uhodnout na maximálně 4 pokusy.

Pokud budeme hádat čísla v rozmezí 1 až 100, k uhodnutí budeme potřebovat maximálně 7 pokusů.

U čísel do 1000 to bude 10 pokusů a do 1 000 000 pak 20 pokusů.

snímek 13

Hra v kostky

Otevři si projekt *Házení kostkou*. Postavy v něm budou „hrát v kostky“.

- Vytvoř dvě proměnné **hod1** a **hod2**. Postavě rozhodčí sestav scénář, který po kliknutí na praporek vloží do proměnných náhodné číslo od 1 do 6 a vyšle zprávu „hod“.
- Zbylé dvě postavy mají po přijetí zprávy „hod“ ukázat v bublině hodnotu proměnných **hod1**, **hod2**.

Aktivita začíná otevřením projektu *Házení kostkou*. Úkolem žáků bude ve čtyřech navazujících úlohách vytvořit simulaci, při níž budou dvě postavy hrát v kostky.

METODICKÉ POZNÁMKY

- Žáci pracují podle instrukcí. Postava *rozhodčí* vysílá zprávu „hod“, aby v následující úloze na ni mohly postavy reagovat. Pokud žáci začnou z prázdného projektu, postavy si postupně přidávají.
- Opakovaným spouštěním praporeku žáci vlastně odhalí, co naprogramovali. Volme raději blok **bublina** než **bublina 2 sekund**. Nevadí, že se čísla budou zobrazovat až do dalšího hodu. Při zautomatizování výpočtů v 6. úloze se během scénáře nebude „zdržovat“ dlouhým čekáním.

ŘEŠENÍ ÚLOH

1.	2.
prostřední postava (<i>rozhodčí</i>):	levá postava: pravá postava:

snímek 14



Kdo vyhrál hru?



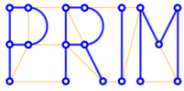
3. Rozhodčí bude určovat, kdo vyhrál. Prohlédni si její kostýmy, které ukazují směr.



4. Přidej do scénáře bloky, v nichž rozhodčí zjistí, jestli je v proměnné **hod1** stejné, větší nebo menší číslo než v **hod2**. Podle toho ukáže na hráče, který říká větší číslo, nebo neukáže nikam.

METODICKÉ POZNÁMKY




3. Pokud vytváříme hru z prázdného projektu, pak je potřeba, aby rozhodčí mohla svými kostýmy ukazovat vlevo, vpravo (vhodná je např. šipka *Arrow1*), případně to bude postava s jedním kostýmem, která se bude natáčet směrem doleva, doprava a dolů).
4. Úloha je podobná porovnávání $>$, $=$, $<$ při aktivitě *Myslím si číslo*. Porovnávání začne po vyslání zprávy „hod“. Podle výsledku porovnání rozhodčí nastaví kostým, aby ukazoval na tu postavu, která říká větší číslo.



ŘEŠENÍ ÚLOH

4.

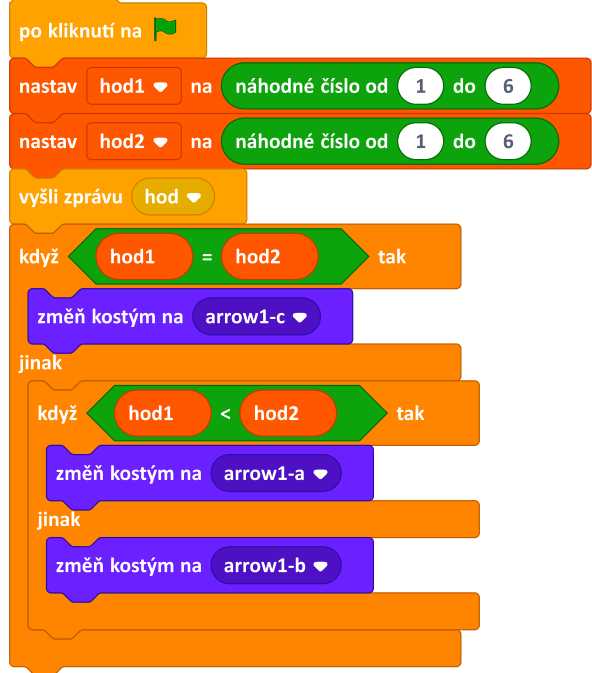
V tomto řešení jsou použity kostýmy z postavy

Arrow1:   . Snadno si dosadíte názvy kostýmů postavy *rozhodčí*.

Zobrazené řešení je jen jedno z možných,


bloky **když __ tak __ jinak**, případně

když __ tak lze seskládat mnoha jinými způsoby (viz řešení u aktivity *Myslím si číslo*).



snímek 15


Kdo vyhrál častěji?

5. Budeme počítat, kolik kdo získal bodů. Vytvoř jednomu hráči proměnnou **body1** a druhému **body2**. Přidej rozhodčí do scénáře bloky, které přičtou k těmto proměnným 1 bod, pokud daný hráč vyhraje. Pak hraj, opakovaně spouštěj .

6. Hru můžeme zautomatizovat, nastavit třeba 1000 opakování hodu kostkou a na konci se podívat, kdo z hráčů získal více bodů.

body1	405	3
body2	431	

Bude rozdíl mezi hráči velký?



Aktivity 5 a 6 jsou určené pro pokročilé žáky. Aktivita 6 má přesah do zkoumání jako úvod do počtu pravděpodobnosti.

Přístup žáků k řešení úloh se může od dole naznačených postupů značně lišit. Učitel by do žakovských postupů neměl zasahovat, neměl by prosazovat své řešení. Důležité je, aby výsledná řešení odpovídala zadání. Žák si může upravit i zadání úlohy – podporujeme jeho tvořivost.

METODICKÉ POZNÁMKY

5. Žáci vytvoří další dvě proměnné **body1** a **body2** a přidají do scénáře přičítání bodů. Pod podmínkou $\text{hod1} > \text{hod2}$ přidají blok, který zvětší hodnotu proměnné **body1**. Pod podmínkou $\text{hod1} < \text{hod2}$ přidají zvětšení proměnné **body2**.

Projekt žáci opakovaně spouští praporkem a pozorují změny hodnot **body1** a **body2**.

6. Doporučujeme celý dosavadní scénář hodu kostkou pojmenovat jako nový blok **jeden hod**. Tento nový blok pak necháme opakovat v novém scénáři. Na jeho začátku bude ještě potřeba vynulovat proměnné **body1** a **body2**.

Následné experimentování s touto simulací (motivované otázkou v zadání, zda bude mezi hráči velký rozdíl) přivádí žáky k teorii pravděpodobnosti. Vidí tedy **užitečnost informatiky pro zkoumání světa**. Žáci sami mohou na mnoha pokusech ověřit, že počet bodů u obou postav je pokaždé přibližně stejný. Tento poznatek lze ověřit opakováním na opravdu velkém množství pokusů.

Aby se zrychlil výpočet při velkém počtu opakování, odstraníme ze scénáře všechny grafické akce na scéně (tedy změny kostýmů a vytváření bublin). Potom je Scratch schopen vykonat i řádově milión hodů v rozumném čase.

ŘEŠENÍ ÚLOH

5.	6.
<p>Oba dlouhé scénáře jsou až na hlavičku identické.</p> <p>Ponechali jsme je oba, aby bylo patrné, jaké změny se provedou se scénářem v úloze 6.</p>	

snímek 16

Žralok – příprava

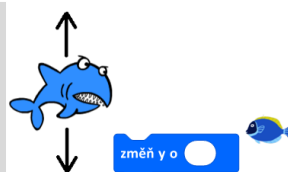
Naprogramujeme hru, v níž žralok chytá rybky.

Otevři si nový prázdný projekt.

1. Přidej si novou postavu žraloka (kočičku smaž).
Postavu můžeš zmenšit. Umísti ji k levému okraji scény.

nastav velikost na 50 %

2. Žraloka budeme ovládat šipkami nahoru a dolů.
Po stisku šipky žralok poskočí o 20 kroků nahoru, případně dolů.



Žáci si otevřou nový prázdný projekt. Následující aktivity směřují k vytvoření jednoduché hry, v níž hráč - žralok bude lovit rybky.

METODICKÉ POZNÁMKY

1. Sada úloh, v níž žáci mají popsáný postup tvorby úlohy, ale musí ji naprogramovat. Je možné dojít do určité úrovně a skončit, projekt ale obsahuje i těžší úlohy pro pokročilé. Doporučujeme práci ve dvojicích.

Všechny informace žáci najdou na snímku.

2. Žralok bude reagovat na každé stisknutí klávesy [šipka nahoru] a [šipka dolů] poskočením nahoru nebo dolů, tj. změnou souřadnice y o 20, resp. -20. Je možné nastavit jiné číslo, velké či malé hodnoty však sníží hratelnost hry.

ŘEŠENÍ ÚLOH

2.

Two Scratch code blocks are shown. The first block is: 'po stisku klávesy šipka nahoru' (when key pressed: up arrow) followed by 'změň y o 20' (change y by 20). The second block is: 'po stisku klávesy šipka dolů' (when key pressed: down arrow) followed by 'změň y o -20' (change y by -20).

MOŽNÉ POTÍŽE A JEJICH ŘEŠENÍ

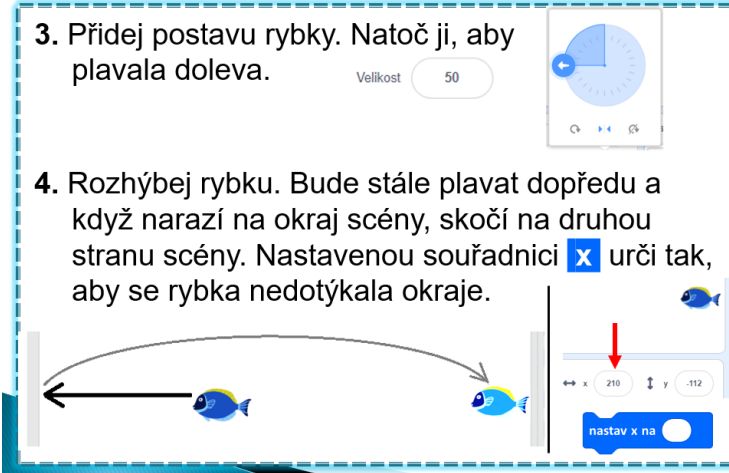
- Pokud žralok skáče podivně nebo není vidět; zkontrolujme, zda se mění y o 20 kroků.

snímek 17


Žralok a rybka

3. Přidej postavu rybky. Natoč ji, aby plavala doleva. Velikost 50

4. Rozhýbej rybku. Bude stále plavat dopředu a když narazí na okraj scény, skočí na druhou stranu scény. Nastavenou souřadnici **x** urči tak, aby se rybka nedotýkala okraje.



METODICKÉ POZNÁMKY

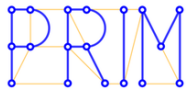
3. Ve hře bude rybka plavat doleva a žralok jí bude u levého okraje scény chytat. Rybka se bude pohybovat pouze vlevo.

Rybku je vhodné zmenšit tak, aby se vešla žralokovi do tlamy, např. na 30 %.



4. Žáci programují pohyb rybky tak, aby se pohybovala sama (bez klávesnice). Rybka nemá plavat sem tam, ale po dotyku levého okraje scény **přeskočí** k pravému okraji. Nelze tedy použít blok **když narazíš na okraj, odraz se**.

Souřadnici **x**, na kterou má rybka přeskočit, zjistíme po přesunu odečtením hodnoty **x** pod scénou (šipka na obrázku).

Doporučujeme použít blok **nastav x na** raději než **skoč na x: _ y: _**. V dalších úlohách bude rybka vybírat hladinu, ve které bude plavat, a tudíž náhodně měnit souřadnici **y**. To je snazší pochopit, když ke scénáři přidáme blok **nastav y** na náhodné číslo, než v bloku **skoč na x: _ y: _** měnit **y** za náhodné číslo. Jinak jsou ale oba postupy rovnocenné.



ŘEŠENÍ ÚLOH

3.	4.
<p>Nastavení rybky se provede jednou, uchová se pro další úlohy.</p> <p>Kromě ručního nastavení lze spustit scénář:</p> 	 <p>Číslo v bloku nastav závisí na velikosti rybky. Při nezmenšené velikosti rybky vyhovuje číslo do 160, ale je třeba jej nastavit individuálně.</p>

MOŽNÉ POTÍŽE A JEJICH ŘEŠENÍ

- Rybka přestane plavat a stojí u pravého okraje scény. Po posunutí dovnitř scény myší rybka plave, ale pak opět skočí na pravý okraj a stojí. Patrně má nastavenou příliš velkou souřadnici x , takže se ocasem stále dotýká okraje. Rybka pak stále „skáče“ na toto místo a neplave dál. Stačí změnit souřadnici x v bloku **nastav x _** na menší číslo.
- Rybka plave vzhůru břichem. Stačí jí nastavit způsob otáčení.
- Rybka podivně skáče. Patrně má nastavený velký krok při pohybu dopředu.
- Pokud rybka neplave vodorovně, nemusí doplavit až k levému okraji nebo k žralokovi. Hra pak nebude správně fungovat. Nastavíme směr -90 .

snímek 18

Žralok a rybka pokračování

5. Aby rybka neplavala pořád po stejné dráze, při skoku na začátek scény jí nastav náhodnou souřadnici **y**.

Čísla **od – do** vyber tak, aby rybka neskákala na horní nebo dolní okraj scény.

nastav y na

náhodné číslo od do

6. Když rybka narazí do žraloka, také skočí na začátek scény na náhodné místo. Doplň scénář.

dotýkáš se Shark ?

METODICKÉ POZNÁMKY

5. Na této úloze je nejtěžší vhodně vybrat meze pro souřadnice **y**, na které rybka skočí. Výběr vhodných souřadnic záleží na velikosti rybky. Budou-li příliš velké, rybka se bude dotýkat horního či dolního okraje scény a hra nebude fungovat. Budou-li meze příliš malé, rybka nebude využívat horní a dolní část scény, bude plavat jen uprostřed.
6. V úloze přidáváme další rozhodování. Při něm se rybka chová stejně jako po dotyku okraje, pouze má jinou podmínku pro dotyk jiné postavy: **dotýkáš se Shark?**

ŘEŠENÍ ÚLOH

5.	6.
<p>Bezpečné meze pro rybu zmenšenou na 30 % jsou y od -150 do 150, pro nezmenšenou rybu pro y od -125 do 125.</p>	

snímek 19


Kolik rybek je sežráno?

7. Můžeme si počítat, kolik rybek žralok sežral. Vytvoř proměnnou **sežráno** a do scénáře pohybu rybky přidej blok **změň sežráno o 1**. Vymysli, kam blok umístíš.

8. Můžeme také ukazovat, kolik rybek uteklo žraloku a doplulo k okraji. Vytvoř proměnnou **uteklo** a blok pro práci s ní umísti do scénáře rybky. Hru vyzkoušej.



METODICKÉ POZNÁMKY

7. Úloha vrací žáky k práci s proměnnými. Žáci vytvoří proměnnou **sežráno** a mají přijít na to, kam umístit blok pro její změnu. Mají tři místa na výběr:

- do bloku **když __ tak** s podmínkou **dotýkáš se Shark?**
- do bloku **když __ tak** s podmínkou **dotýkáš se okraj?**
- mimo tyto bloky dovnitř bloku **opakuji stále**.

Pokud počítání rybek nefunguje, je třeba zkontrolovat umístění bloku **změň sežráno**. Učitel by neměl ukázat na chybu, ale pomoci otázkami žákovi, aby chybu našel sám:

OTÁZKY UČITELE

1. Kde se nachází Tvůj blok **změň sežráno**? Je na správném místě?
2. Podle které podmínky počítač pozná, že má zvětšit počet sežraných rybek?

8. Úloha je obdobou předchozí úlohy. Proměnná **uteklo** se bude zvyšovat po splnění podmínky **dotýkáš se okraje?**

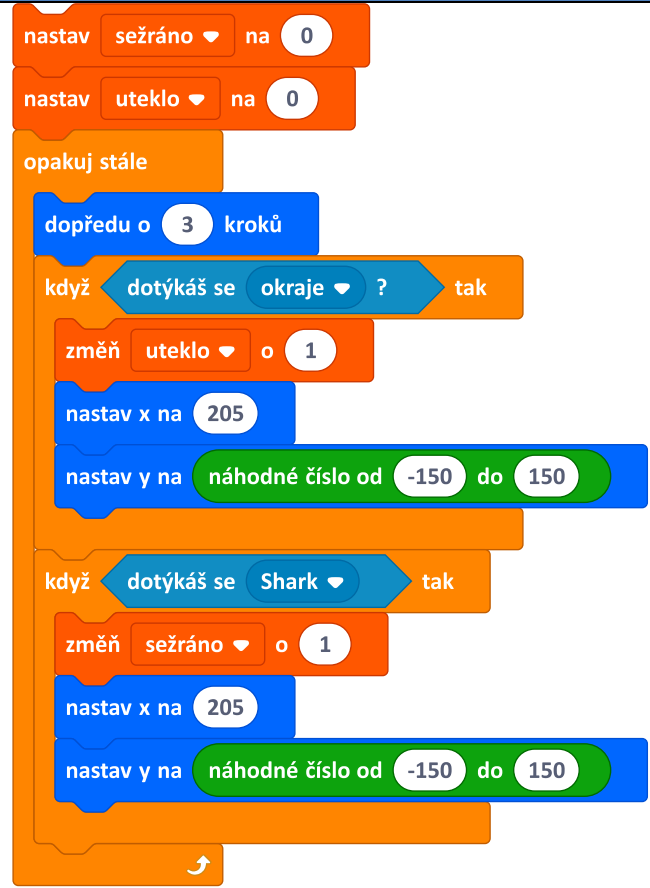
Žáci hru vyzkouší. Sledují, zda se mění hodnoty proměnných správně v obou situacích – když žralok rybku sežere a když mu rybka uteče.

Na začátku scénáře (před blokem **opakuji stále**) musí být nastavena nulová hodnota obou proměnných **sežráno** a **uteklo**.

Žáci mohou doladit rychlost rybky, aby hra byla lépe hratelná.

ŘEŠENÍ ÚLOH

7. a 8.



```

nastav sežráno na 0
nastav uteklo na 0
opakuj stále
  dopředu o 3 kroků
  když dotýkáš se okraje ? tak
    změň uteklo o 1
    nastav x na 205
    nastav y na náhodné číslo od -150 do 150
  když dotýkáš se Shark? tak
    změň sežráno o 1
    nastav x na 205
    nastav y na náhodné číslo od -150 do 150
  
```

snímek 20

Polykání rybky - animace

9. Můžeme sežrání rybky animovat. Žralok otevře tlamu, rybka do ní vpluje a žralok tlamu zavře. Vymysli, kam zařadit do scénáře zprávu, aby otevřel tlamu, kterou mu rybka pošle. Rybka může ještě chvíli plavat, než žralok tlamu zavře.

sežráno 6



sežráno 6



sežráno 7



Následují úlohy pro pokročilé žáky, které hru rozšiřují o další prvky. Vyžadují situaci dobře promyslet a orientovat se v dlouhém scénáři.

METODICKÉ POZNÁMKY

9. V úloze musí žáci synchronizovat dvě postavy a u žraloka střídat kostýmy.

Když se rybka dotkne žraloka, vyšle zprávu „otevři tlamu“. Žralok na zprávu reaguje otevřením tlamy, tj. změnou kostýmu. Po chvíli tlamu zavře (**čkej __ sekund**).

Mezitím rybka ještě plave, ale přitom nesmí testovat, zda se dotýká žraloka (k proměnné **sežráno** by se přičítaly další body). Vhodné řešení je nechat rybku několikrát popojet stejnou rychlostí jako předtím (obrázek vpravo).

Dobu, po kterou má žralok otevřenou tlamu, lze řešit odhadem a stanovit, jak dlouho bude mít žralok otevřenou tlamu (např. 0,3 s).

Jiné řešení pomocí posílání zpráv mezi rybkou a žralokem sem a tam je komplikovanější v tom, že je třeba si dobře rozmyslet, kdy se bude zvětšovat počet sežraných, aby nedocházelo k chybnému počítání.



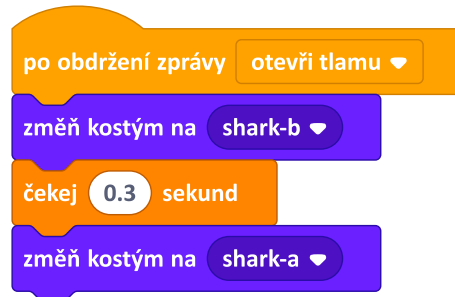
ŘEŠENÍ ÚLOH

9.

Rybka (ve scénáři je červenou závorkou vyznačeno, které nové bloky přibyly)



Žralok



snímek 21

Jak zakončit hru?

Naše hra nikdy nekončí, běží stále. Potřebujeme podmínku, podle které bude moci ryбка zastavit. Tou může být např. počet 10 sežraných rybek.

10. Nahraď blok **opakuj stále** blokem s vhodnou podmínkou. Hraj hru, spouštěj ji . Nejlepšímu hráči neuteče ani jedna ryбка.

11. Nakopíruj několikrát postavu ryбки a vytvoř více rybek. Můžeš jim změnit kostýmy.



METODICKÉ POZNÁMKY

10. Způsob zakončení hry si může rozmyslet každý žák po svém. Podmínkou může být počet sežraných rybek, počet rybek, které utečou, nebo čas hry. Žáci budou mít jistě i další nápady.

Aby se hra nějak ukončila, je třeba nahradit blok **opakuj stále** blokem **opakuj dokud nenastane** s podmínkou např. **uteklo** = 10, **sežráno** = 20 nebo **stopky** > 20 (neboť spouštění praporkem nuluje stopky).

Vpravo: upravený začátek scénáře.



11. Žáci postavu ryбки nakopírují.



Žáci mohou jednotlivým rybkám změnit kostýmy. Lze nastavit každé rybce jinou rychlost plavání a také jinou souřadnici x, kam skočí po sežrání.

Hra se pak stává více reálnou.

Další náměty pro rychlé žáky: postavy mohou jednotlivé činnosti komentovat vtipnými hláškami v bublině. Jednak rozvíjí tvořivost, jinak musí vymyslet, kam bublinu do scénáře umístit.

snímek 22



Zkoumáme žraloka

Prozkoumej a výsledek svého bádání zapiš:

1. Proč bylo důležité nejprve úplně dokončit scénář ryбки a teprve pak postavu ryбки kopírovat?
2. Bylo důležité, že jsme nechali spouštět scénář ryбки praporkem? Vysvětli.
3. Proč není vhodné, aby se dotyk s rybkou kontroloval ve scénáři žraloka?
K jaké chybě by mohlo dojít?



METODICKÉ POZNÁMKY

Na závěr čeká na bystré žáky pokročilé bádání, v němž prokážou, že mají v programování přehled. Žáci (nebo dvojice) budou pracovat samostatně svým tempem. Je možné zakončit diskusí v celé skupině pokročilých žáků: ti by se měli nakonec dohodnout na jednotném názoru na odpovědi. V případě, že se neshodnou, bude učitel moderátorem sporné situace. V takovém případě by žáci měli přesvědčovat druhé; varianta, že tak učiní učitel, není přínosná.

ŘEŠENÍ ÚLOH

1.

Pokud by nebyl správně odladěn scénář pro jednu ryбку, chyba by se nakopírovala k dalším rybkám a každou opravu bychom pak museli provádět několikrát, pro všechny ryбки. Proto je vhodné vytvořit nejprve správný scénář pro jednu ryбку a až následně postavy kopírovat.


2.

Praporek spouští scénáře všem postavám najednou (podobně jako stisk klávesy). Pokud bychom praporek ani klávesu nepoužili, museli bychom každou ryбку spouštět zvlášť.


3.

Kdybychom testovali dotyk s rybkou ve scénáři žraloka, po nakopírování rybek by žralok pořád testoval jen dotyk s původní rybkou, na ostatní, které mají jiné jméno, by nereagoval.

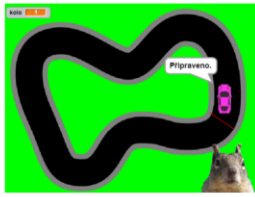
snímek 23



Autodráha



Otevři si projekt *Autodráha*.
Prohlédni si postavy, proměnné
a vyzkoušej připravené scénáře.



1. Naprogramuj pohyb závodního auta. Po spuštění praporkem auto stále pojedí za myši.
2. Blok **vyjelo z trati** bude kontrolovat, zda auto nevyjelo na zelený trávník. V takovém případě auto zacouvá o 10 kroků.
Tento nový blok použij ve scénáři auta.

Aktivita začíná otevřením projektu *Autodráha*.

Sada úloh pro rychlé a bystré žáky nebo pro gymnaziální třídy. Postupným řešením úloh žáci vytvoří jednoduchou hru, kterou opatřují dalšími funkcemi. Proměnné využijí pro řízení hry.

V autodráze jezdí závodní auto ovládané myši, které mění rychlost a snaží se udržet v trati, což počítač kontroluje. Hra má počítání kol a sankce za vyjetí z trati. Postava komentátora sděluje informace o závodě a výsledky.

Práce je popsána v jedenácti navazujících krocích, jejichž pořadí nelze měnit. Každý ze žáků (nebo každá dvojice) bude na úkolech pracovat svým tempem, a proto doporučujeme nechat žáky pracovat a řešit jednotlivé problémy samostatně. Učitel by měl radit či pomoci pouze individuálně, a to v případě, kdy žáci nebudou umět samostatně pokračovat.

V projektu je předem vytvořen blok **na start**, který slouží nastavení startovní pozice a parametrů auta před závodem. Blok lze spustit stisknutím mezerníku.

METODICKÉ POZNÁMKY

1. Opakovací a přípravná úloha. Žáci sestaví pohyb auta podobný pohybu rakety v kap. 7.
Scénář se bude spouštět zeleným praporkem.
Žáci scénář otestují a stanoví vhodnou rychlost auta (blok **dopředu o __ kroků**).
2. Kvůli přehlednosti i z důvodu procvičení žáci vytvoří nový blok **vyjelo z trati**, který bude testovat, zda auto vyjelo z trati, tedy zda se auto nenachází na zeleném trávníku.
V dalších úlohách budeme blok **vyjelo z trati** měnit.
Z trávníku má auto zacouvat o 10 kroků (blok **dopředu o -10 kroků**). Není vhodné dát menší počet kroků zpět – je potřeba, aby auto opravdu z trávy vycouvalo najednou, jinak by se mohlo při nevhodném pohybu myši stát, že ztratí kontakt s jízdní dráhou.

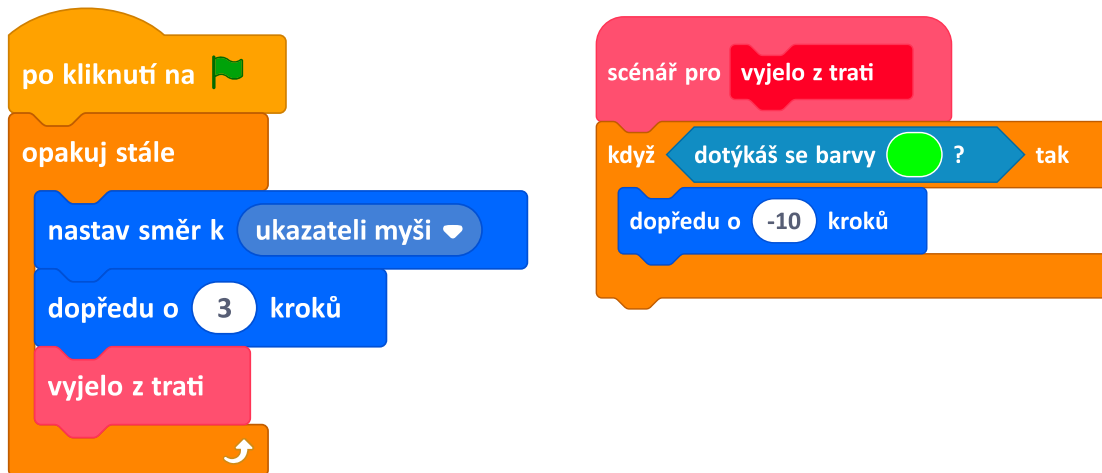
ŘEŠENÍ ÚLOH

1.





2.


Ve scénáři auta:



snímek 24

 **Autodráha** pokračování


3. Nastav ovládání rychlosti auta klávesami šipka nahoru a dolů. Pamatuj, že auto nesmí couvat.
4. Při vyjetí z trati bude závodník penalizován: v takovém případě nastav auto rychlost na 0.
5. Hra by měla poznat, že auto projelo závodní okruh, a ukončit závod. Můžeš k tomu využít postavu *cílová čára*.



METODICKÉ POZNÁMKY

3. Řízení rychlosti pohybu auta žáci znají z pohybu rakety včetně zamezení couvání. Proměnná **rychlost** je již v projektu definována. Žáci pouze zpracují události stisku kláves [šipka nahoru] a [šipka dolů].
Aby se hodnota proměnné **rychlost** projevila v pohybu auta, je nutné upravit scénář pro pohyb auta – počet kroků **dopředu** bude záviset na proměnné **rychlost**.
Doporučujeme prohlédnout si nastavení parametrů auta v bloku **na start**. Počáteční rychlost auta je zde již nastavena.
Blok **na start** je sestaven tak, aby se dal spouštět samostatně, nezávisle na startu celé hry. Je na žákovi, jestli jej zakomponuje do začátku scénáře, nebo jej nechá stát zvlášť.
4. Žák změní blok **vyjelo z trati** – vynuluje **rychlost**.
Blok pro vycouvání auta z trávníku musí ve scénáři zůstat.
5. Hra nemá běžet donekonečna. Skončí, když auto projede závodní okruh a dotkne se postavy *cílová čára*.
Žáci nahradí blok **opakuje stále** blokem **opakuje dokud nenastane ___** s podmínkou doteku postavy *cílová čára*.
Žáci mohou mít jiné návrhy, jak auto pozná, že objelo jeden okruh, nebo jiné nápady, jak dlouho hra potrvá. Měla by ale nějak skončit.

ŘEŠENÍ ÚLOH

3.

Ve scénáři pro pohyb auta:





4.



5.






snímek 25

 Autodráha – komentátor


6. Na scéně je postava komentátora. Prohlédni si, jaká hlášení říká a čím se mají spustit.

7. Umísti do scénářů auta bloky se zprávami tak, aby komentátor mohl ve správnou chvíli komentovat dění na závodišti.

METODICKÉ POZNÁMKY

6. Postava veverky – komentátora je již připravena včetně všech scénářů. Scénáře se spouštějí po příjmu zpráv. Žáci mohou hlášení modifikovat.

Úkolem pro žáka je **umístit do scénářů auta bloky pro vysílání zpráv**.

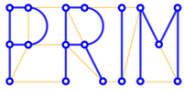
7. Zprávou **start** odstartuje komentátor závod. Její vyslání je třeba umístit na začátek scénáře pro pohyb auta.

Zpráva **vyjelo z trati** oznamuje pohyb auta po zeleném trávníku mimo trať. Měla by být vyslána v bloku **vyjelo z trati**.

Zpráva **v cíli** souvisí s ukončením pohybu auta po objetí závodního okruhu. Bude vyslána na konci scénáře pro pohyb auta.

ŘEŠENÍ ÚLOH

7.



The image shows a Scratch script for a train simulation, divided into two parts. The left part is the main script, and the right part is a sub-scenario for 'vyjelo z trati' (left the track).

Main Script:

- po kliknutí na
- na start
- vyšli zprávu start
- opakuji dokud nenastane **dotýkáš se cílová čára**
- nastav směr k ukazateli myši
- dopředu o rychlost kroků
- vyjelo z trati
- vyšli zprávu v cíli


Scenario for 'vyjelo z trati':

- když **dotýkáš se barvy** (green circle) **tak**
- vyšli zprávu vyjelo z trati
- dopředu o -10 kroků
- nastav rychlost na 0

snímek 26

Autodráha – více okruhů

8. Přepni pozadí scény na žlutý okruh. Uprav podmínku vyjetí z trati.
9. Uprav scénář pro auto, aby při dojetí do cíle skočilo na start a přičetl se jeden okruh.
10. Vytvoř závod na tři okruhy. Závod se ukončí tehdy, když auto vjíždí do okruhu č. 4.
11. Vymysli, jak by se dala počítat ujetá kola v zeleném okruhu, kde auto cílovou čáru projíždí.



Tento snímek ukazuje již jen náměty pro další činnost. Žák již musí být dostatečně samostatný, aby si s problémy poradil sám. Pracuje vesměs individuálně nebo ve skupince, bez učitele.

METODICKÉ POZNÁMKY

8. Žáci pouze změní pozadí scény na žlutý okruh a změní nastavení barvy v podmínce v bloku **vyjelo z trati**.

Žáci vidí, že stejný scénář funguje i pro úplně jiný tvar trati. Auto ale v cíli nezastaví, což je motivací pro další úlohu.

9. Potřebujeme přidat do scénáře další rozhodování, jestli je auto v cíli. Cíl zde není označen postavou *cílová čára*, ale červenou barvou pozadí. Žáci musí přidat testování, zda se auto nachází na červené barvě.

Doporučujeme vytvořit pro testování, jestli je auto na konci kola, vlastní blok **na konci kola**. Kód hry bude přehlednější.


Auto má po dosažení cíle skočit na startovní pozici. Žáci mohou použít přímo blok **na start**.

10. Hra se jede na tři okruhy (kola). Podmínkou k ukončení hry není dotyk cílové čáry, ale zda počet ujetých kol dosáhl 3. Žáci vymění podmínku u bloku **opakuj dokud nenastane**.

11. Připravili jsme autodráhu na závod o více kolech a nyní se vracíme zpět k původní kruhové autodráze.



Nyní auto v podstatě funguje, akorát po dotyku červené cílové čáry „poskočí“ na startovní pozici. Pokud tento blok odstraníme, nastane problém, že auto se dotýká této čáry po dlouhou dobu a přitom se stále zvětšuje proměnná **kolo**.

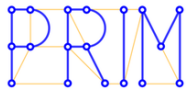
Problémem k řešení je, jak zařídit, aby se proměnná **kolo** zvětšila pouze jednou. Nabízí se řada řešení, které by měli ale žáci vymýšlet sami:

- Postava *cílová čára* čeká, až se dotkne auta, a pak se na určitou dobu (třeba 5 sekund) skryje.
- Postava *cílová čára* může počkat, až auto z červené barvy odjede. Do jejího scénáře zařadíme blok . Poté, co se auto dotkne červené barvy, tento blok způsobí, že bude postava čekat, a teprve poté přičte jedno ujeté kolo.
- Další možností je přidat další čáru zhruba v polovině hřiště, teprve když se jí auto dotkne, objeví je cílová čára. Tím zároveň znemožníme hráči podvádět tím, že by auto točil pořád kolem cílové čáry a „lacino“ získával ujetá kola.

Každé z těchto řešení má svá úskalí a poskytuje další motivaci k tomu, jak se dál zlepšovat.

ŘEŠENÍ ÚLOH

9.	10.
	 <p data-bbox="874 1547 1388 1653">Poznámka: V řešení úlohy 9 ještě nenulujeme proměnnou kolo, protože se každé kolo jízdy spouští zvlášť.</p>



Co už umíš

- používat proměnné k pamatování si hodnot
- nastavovat a měnit hodnotu proměnné
- porovnávat hodnoty v proměnné
- používat více proměnných
- řešit problémy použitím proměnné

Poslední snímek kapitoly slouží k uvědomění si, co již žáci dokázali.

Pro ověření znalostí žáků lze po probrání této kapitoly použít testy po 10. kapitole, které jsou dostupné spolu s ostatními materiály této učebnice.