

Fyzický programovací jazyk Code Jumper

Metodický materiál pre učiteľa

Autor: Ľudmila Jašková

Vytvorené v rámci projektu KEGA 018UK-4/2019: Rozvoj algoritmického myslenia ťažko
zrakovo postihnutých žiakov základnej školy

Vydala: KDMFI, FMFI UK Bratislava, 2021

Autor textu:

doc. RNDr. Ľudmila Jašková, PhD.,

Spolupráca pri overovaní edukačných aktivít so žiakmi:

PaedDr. Natália Kováčová, PhD.

Mgr. Mária Karasová

Spolupráca na tvorbe zvukových súborov:

Mgr. Mária Stankovičová

Dostupnosť:

<http://vin.edu.fmph.uniba.sk>

Obsah

Úvod	4
Hodina 1 – Zoznámenie s jazykom Code Jumper	5
Hodina 2 – Parametre príkazu	8
Hodina 3 - Vlákna	13
Hodina 4 – Hľadanie chyby v programe	18
Hodina 5 - Cyklus	23
Hodina 6 - Sekvencia príkazov pred a za cyklom	28
Hodina 7- Dekompozícia	33
Hodina 8 – Príkaz vetvenia	36
Hodina 9 – Čary s číslami	43
Hodina 10 – Premenná	46
Hodina 11 – Projekt.....	51
Príloha	52
Popis jazyka Code Jumper.....	52
Popis prostredia aplikácie Code Jumper.....	57
Zoznam kategórií zvukov k úlohám	59
Literatúra	61

Úvod

Tento metodický text je určený pre učiteľov nevidiacich žiakov nižšieho sekundárneho vzdelávania. Cieľom je poskytnúť im návod na to, aby dokázali atraktívnym spôsobom učiť programovanie pomocou fyzického programovacieho jazyka Code Jumper (<https://codejumper.com/>). Tento jazyk vyvinuli v Microsoft Research Lab – Cambridge v rámci projektu Torino. Obsahuje hmatovo odlišiteľné diely predstavujúce jednotlivé príkazy, z ktorých môžeme jednoduchým spájaním vytvárať sekvencie príkazov.

Tento metodický materiál bol vytvorený tak, aby umožňoval rozvíjať algoritmické myslenie a programátorské zručnosti nevidiacich žiakov konštruktivistickým spôsobom. To znamená, že ponúka dostatok úloh s gradovanou náročnosťou. Pri ich riešení nie je potrebný výklad učiteľa, ale jeho intervencia by mala mať formu vhodných otázok, ktoré žiaka navádzajú na riešenie.

Odporúčame, aby sa učiteľ zoznámil s programovacím jazykom Code Jumper vopred a doplnil do neho kategórie zvukov potrebné pre riešenie úloh. Všetky kategórie zvukov sa dajú stiahnuť zo stránky <http://vin.edu.fmph.uniba.sk/prostredia.html>. Inštrukcie týkajúce sa pridávania vlastných zvukov sú v častiach *Príloha – Popis prostredia aplikácie Code Jumper*, *Príloha – Zoznam kategórií zvukov k prílohám*.

Metodický materiál obsahuje jedenásť kapitol. Každá kapitola je koncipovaná ako metodický návod na jednu vyučovaciu hodinu. Táto časová dotácia je však iba orientačná. Potrebný čas závisí od počtu žiakov v skupine, od ich zručností a učiteľových skúseností.

Z hľadiska vstupných vedomostí a zručností žiakov, nie sú potrebné žiadne predchádzajúce vedomosti z oblasti práce s počítačom a s programovaním.

Z hľadiska potrebného technického vybavenia, okrem fyzického programovacieho jazyka Code Jumper, je potrebný počítač alebo tablet so systémom minimálne Windows 10 alebo tablet so systémom Android, ktorý podporuje pripojenie zariadenia Code Jumper cez Bluetooth. Informácie o možnostiach zakúpenia jazyka Code Jumper sú na stránke <https://codejumper.com/>.

Odporúčame, aby s jednou sadou jazyka Code Jumper pracovali maximálne traja žiaci, ktorí sa pri riešení úloh striedajú. V tejto malej skupinke môžu byť žiaci s rôznymi typmi zrakového postihnutia (slabozrakí, nevidiaci), ale aj intaktní.

Prajeme príjemnú zábavu pri riešení úloh. Nahrávky hotových programov, ktoré majú žiaci vytvoriť sa dajú stiahnuť zo stránky <http://vin.edu.fmph.uniba.sk/prostredia.html>. Odporúčame ich prehrať žiakom predtým ako budú riešiť jednotlivé úlohy.

Hodina 1 - Zoznámenie s jazykom Code Jumper

Cieľ

- Porozumieť pojmom príkaz (presne zadaný), sekvencia príkazov (dôležité je poradie), program (sekvencia dohodnutých príkazov zadaných v správnom poradí)
- Vytvoriť program pozostávajúci zo 4 až 8 základných príkazov

Priebeh hodiny

Motivácia

Povieme žiakom, že sa nasledujúcich pár hodín budeme hrať na programátorov. Spýtame sa žiakov, čo si myslia, že robia programátori. Asi každého napadlo, že programátori programujú. Povieme, že budeme programovať, ale najprv sami vykonáme jednoduchý program. Môžeme sa chvíľu rozprávať o tom, čo je podľa nich program a kto ho obyčajne vykonáva.

Úloha 1 - Kartičková aktivita

Vezmeme kartičky znázornené na obrázku (treba ich mať vytlačené v čiernotlačí aj Braillovom písme) a žiaci budú postupne vykonávať to, čo je na nich napísané.

Tleskni	Dupni
Lúskni	Zapískaj
Zaklop	Zapípaj
Štekaj	Mňaukaj
Mékaj	Kvíkaj

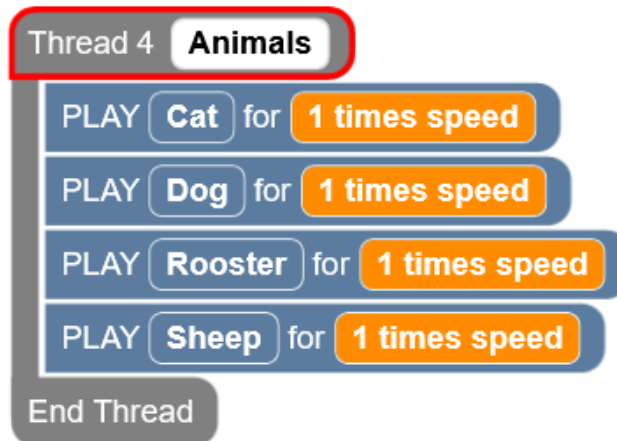
Postupne každému zo žiakov dávame niekoľko (3 až 5) kartičiek a žiaci majú vykonať, čo je na nich napísané v tom poradí, v akom sú kartičky zoradené.

Úloha 2 - Zoznámenie s jazykom Code Jumper

Ukážeme žiakom Code Jumper - základný modul a jeden príkazový modul **Zahraj** a necháme ich, aby si ich ohmatali.



Vyzveme žiakov, aby pripojili príkazový modul do konektora na pravej strane základného modulu (Thread 4 / Animals) a stlačili tlačidlo Štart (veľké s trojuholníkom). Budeme diskutovať o tom, ako to funguje. Nech žiaci skúsia pripojiť ďalší príkazový modul Zahraj a opäť stlačia tlačidlo Štart. Potom nech žiaci nastaví iné zvuky na zapojených dvoch príkazových moduloch (otáčaním väčšieho kolieska na moduloch). Nech skúsia vytvoriť program so štyrmi príkazovými modulmi. Napríklad nasledujúci program.



Zapojenie modulov bude vyzerat' nasledovne.



Úloha 3 - Stupnica

Do vlákna 1 (Thread 1) nastavíme zvuk klavíra (Sound Category – Midi instruments, Sound Set - Piano). Necháme žiakov zapojiť 8 príkazových modulov **Zahraj** a nastaviť na nich tóny tak, aby sa po stlačení tlačidla štart zahrála stupnica.

Thread 1 **Piano**

PLAY C5 for 1 beat

PLAY D5 for 1 beat

PLAY E5 for 1 beat

PLAY F5 for 1 beat

PLAY G5 for 1 beat

PLAY A5 for 1 beat

PLAY B5 for 1 beat

PLAY C6 for 1 beat

End Thread

Úloha 4 - Pieseň

Šikovnejší žiaci môžu naprogramovať jednoduchú melódiu, v ktorej budú mať všetky tóny rovnakú dĺžku. Napríklad to môže byť pieseň **My sme malí muzikanti**.

My sme ma - lí mu - zi - kan - ti,

THREAD 1 **Piano**

PLAY C5 for 1 beat

PLAY D5 for 1 beat

PLAY E5 for 1 beat

PLAY F5 for 1 beat

PLAY G5 for 1 beat

PLAY A5 for 1 beat

PLAY G5 for 1 beat

PLAY E5 for 1 beat

END THREAD

Záver

Zhodnotíme prácu žiakov. Povieme im, že vytvárali svoje prvé programy. Opýtame sa, čo je to program. Dôležité je, aby v ich odpovedi zaznelo, že je to **postupnosť dohodnutých príkazov** usporiadaných **v správnom poradí** - iné poradie dá iný výsledok.

Hodina 2 – Parametre príkazu

Cieľ

- Porozumieť pojmu vlastnosť (parameter) príkazu
- Vedieť nastaviť typ zvuku a dĺžku jeho trvania
- Zostaviť postupnosť príkazov s rôznymi vlastnosťami

Priebeh hodiny

Opakovanie

Zopakujeme si, čo sme robili na predchádzajúcej hodine.

- Vytvárali sme program pomocou príkazových modulov z jazyka Code Jumper.
- Naučili sme sa, čo je to príkaz a čo je program.
- Vytvorili sme niekoľko programov
 - postupnosť zvukov (pes rozbil sklo a jeho pán povedal Oh, no!)
 - postupnosť tónov – stupnicu, pieseň

Motivácia

Povieme žiakom, že na predchádzajúcej hodine sme v programoch zorad'ovali zvuky, ktoré sa budú prehrávať, ale neurčili sme ako dlho budú znieť.

Povieme žiakom, že keď ráno vstanú, tak vedia, ako dlho môžu byť v kúpeľni, ako rýchlo majú zjesť raňajky, koľko im má trvať cesta do školy, aby nezmeškali. V škole tiež majú na každú hodinu presne určený čas. Popis každej činnosti, ktorú majú vykonať, má dve časti – čo majú robiť a koľko to má trvať.

Úloha 1 - Kartičková aktivita

Vezmeme kartičky znázornené na obrázku (treba ich mať vytlačené v čiernotlačí aj Braillovom písme) a žiaci budú postupne vykonávať to, čo je na nich napísané. Každý príkaz má dve časti – čo treba spraviť a koľko krát to treba spraviť.

Postupne každému žiakovi dáme niekoľko (3 až 5) kartičiek a žiaci majú vykonať, čo je na nich napísané v tom poradí, v akom sú kartičky zoradené.

Tleskaj 1	Dup 2
Lúskaj 3	Pískaj 4
Klop 1	Pípaj 2
Štekaj 3	Mňaukaj 4
Mékaj 1	Kvíkaj 2

Úloha 2 - Experimentovanie

Povieme žiakom, že už na minulej hodine sa pýtali, na čo slúži druhé koliesko na príkazovom module Zahraj. Po kartičkovej aktivite ich možno napadne, že pomocou tohto kolieska nastavujeme dĺžku trvania zvuku.



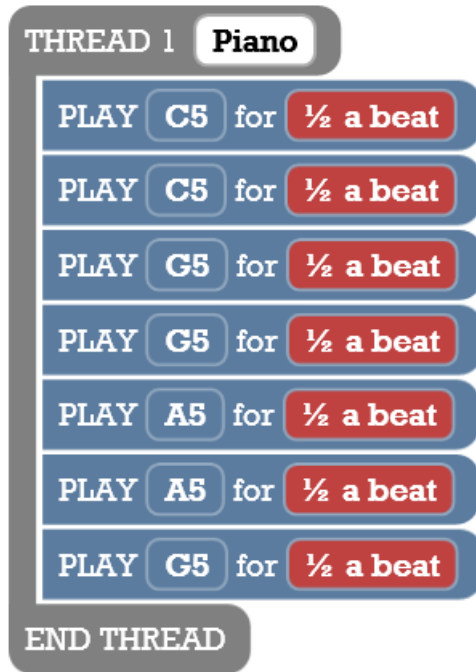
Do vlákna 1 (Thread 1) nastavíme zvuk klavíra (Sound Category – Midi instruments, Sound Set - Piano). Necháme žiakov, aby sami experimentovali s príkazovými modulmi. Najprv si pripoja jeden príkazový modul, napríklad do konektora Piano a vyskúšajú nastavovať rôzne tóny a ich dĺžky.

Úloha 3 - Jednoduchá melódia

Žiakom prehráme melódiu piesne **Twinkle, twinkle** (zo zvukového súboru alebo z telefónu) a žiaci budú vytvárať túto melódiu. Dĺžky všetkých tónov sú rovnaké. Žiaci by mali na to prísť sami.

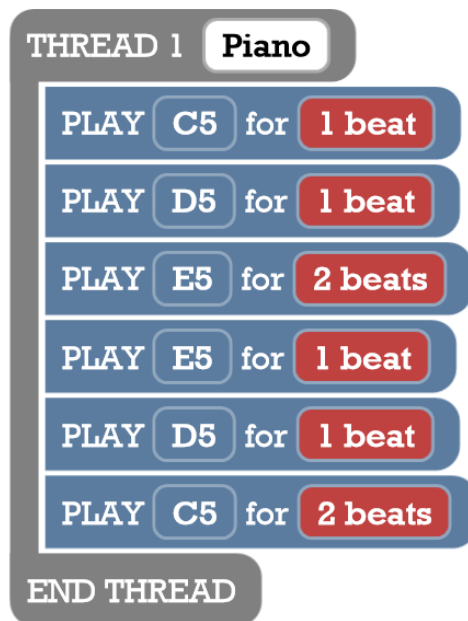
Opýtame sa žiakov:

- Koľko príkazov budeme potrebovať?
- Prečo toľko príkazov?
- Aké vlastnosti (parametre) nastavíme jednotlivým príkazom? Prečo?



Úloha 4 - Komplikovanejšia melódia

Necháme žiakov vytvoriť melódiu piesne **Medveďku, daj labku**. Žiakom prehráme melódiu piesne zo zvukového súboru alebo z telefónu. V tejto piesni sú rôzne dĺžky tónov.



Opäť sa žiakov budeme pýtať:

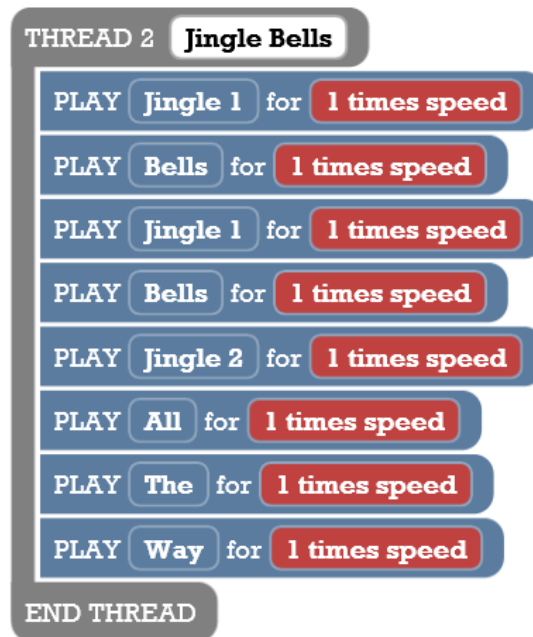
- Koľko príkazov budeme potrebovať?
- Prečo toľko príkazov?
- Aké vlastnosti (parametre) nastavíme jednotlivým príkazom? Prečo?

Úloha 5 - Básnička

Prehráme žiakom pieseň **JingleBells** (z telefónu alebo z audio súboru). Žiaci budú vytvárať túto pieseň. Musíme však v niektorom vlákne nastaviť kategóriu Sample Sounds / JingleBells.

Opýtame sa žiakov:

- Koľko príkazov budeme potrebovať?
- Prečo toľko príkazov?
- Aké vlastnosti (parametre) nastavíme jednotlivým príkazom? Prečo?



Úloha 6 - Príbeh

Prehráme žiakom takýto príbeh zo zvukového súboru, prípadne im ho dáme vytlačený.

Anička hľadá svojich kamarátov. Možno sú v skrini. Pomaly otvorí dvere. Jej kamaráti vyskočia a zvolajú Prekvapenie!

Žiaci budú vytvárať tento príbeh. Musíme však v niektorom vlákne nastaviť kategóriu zvukov *Custom sounds / Príbeh*, ktoré sme nastavili vopred podľa inštrukcií v Prílohe.

Opýtame sa žiakov:

- Koľko príkazov budeme potrebovať?
- Prečo toľko príkazov?
- Aké vlastnosti (parametre) nastavíme jednotlivým príkazom? Prečo?



Keď bude program hotový, vyzveme žiakov, aby program zmenili tak, aby príbeh skončil inak, napríklad nasledovne.

Anička hľadá svojich kamarátov.
 Možno sú v strašidelnom dome.
 Pomaly otvorí dvere.
 ...vŕzganie dverí...
 Zrazu zbadá strašidlo a vykrikuje.

Záver

Zhodnotíme prácu žiakov. Povieme, že naprogramovali ozajstné pesničky a príbehy. Opýtame sa, čo je to vlastnosť (parameter) príkazu. Čo by sa stalo, ak by sme na príkazových moduloch nemali kolieska umožňujúce meniť parametre príkazov?

Dôležité je, aby žiaci prišli na to, že:

- potrebovali by sme namiesto jedného príkazového modulu strašne veľa príkazových modulov, ktoré by vydávali vždy len jeden zvuk s nastavenou dĺžkou,
- každý príkazový modul by musel vyzerat' inak, aby sme ich vedeli rozlíšiť,
- oveľa dlhšie by nám trvalo, kým by sme vytvorili program,
- vďaka parametrom môžeme vytvárať kratšie programy a jednoduchším spôsobom.

Hodina 3 - Vlákna

Cieľ

- Porozumieť pojmu vlákno, paralelne – súčasne (naraz)
- Vedieť zapojiť vlákno do správneho konektora
- Zostaviť postupnosť príkazov pre dve rôzne vlákna

Priebeh hodiny

Opakovanie

Zopakujeme si, čo sme robili na predchádzajúcej hodine.

- Vytvárali sme program pomocou príkazových modulov.
- Naučili sme sa nastavovať zvuk a dĺžku jeho trvania.
- Vytvorili sme niekoľko programov
 - pieseň (JingleBells, Twinkle ..., Medved'ku ...)
 - príbeh (o Aničke a jej kamarátoch)

Motivácia

Povieme žiakom, že na predchádzajúcich hodinách sme zapájali postupnosť príkazov vždy len do jedného konektora a tie sa prehrali postupne za sebou. Čo ak by sme zapojili dve postupnosti do dvoch rôznych konektorov. Zahrali by sa súčasne?

Aj v reálnom svete preda prebieha niekoľko dejov odrazu – napríklad my máme teraz informatiku a druhá skupina žiakov má fyziku.

V orchestri tiež hrá niekoľko nástrojov súčasne. Skúsme aj my vytvoriť program tak, aby klavír a husle zahrali súčasne rovnakú melódiu.

Úloha 1 - Experimentovanie

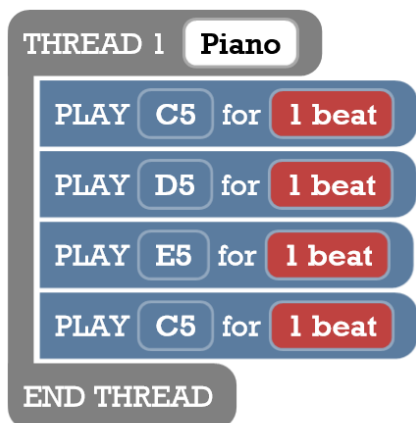
Najprv necháme žiakov, aby chvíľu sami experimentovali. Nech najprv zapoja po jednom programovom module do dvoch rôznych konektorov a nechajú ich prehrať. Znejú naraz?

Úloha 2 - Jednoduchá melódia

Potom žiakom prehráme melódiu piesne **Are you sleeping** (z audio súboru alebo z telefónu) a žiaci budú vytvárať túto melódiu. Najprv jeden žiak vytvorí vlákno pre klavír, potom druhý žiak vytvorí rovnakú melódiu pre husle.

Žiakom pomáhame nasledujúcimi otázkami:

- Koľko nástrojov hrá melódiu?
- Koľko postupností tónov (vláken) budeme vytvárať?
- Koľko príkazov je v každom vlákne?
- Prečo toľko príkazov?
- Aké vlastnosti (parametre) nastavíme jednotlivým príkazom? Prečo?



Jednoduchšou obmenou pre mladších žiakov môže byť melódia nasledujúcej piesne.

Medved'ku, daj labku



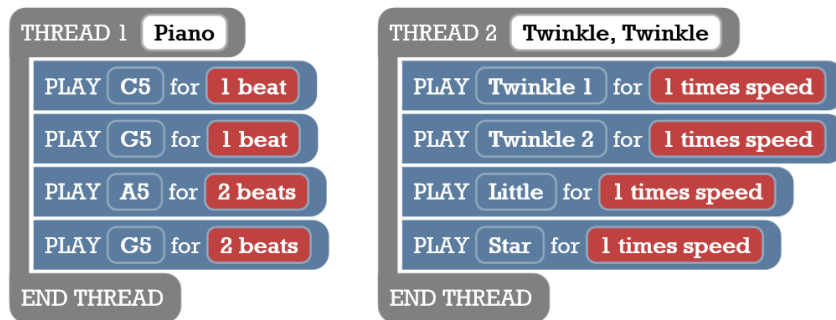
Úloha 4 - Komplikovaná melódia

Šikovnejší žiaci môžu skúsiť vytvoriť dvojhlasnú melódiu, keď každý nástroj hrá niečo iné, ale tak, aby to spolu vytváralo harmóniu.

Are you sleeping ?



Twinkle, twinkle



Poznámka: Zvuky pre pieseň Twinkle, twinkle sú v kategórii Sample sounds / Twinkle, twinkle.

Úloha 4 - Príkaz Pauza

Niekedy nehrajú dva nástroje súčasne, ale sa striedajú. Najprv hrá chvíľu klavír a husle nehrajú a potom klavír prestane hrať a hrajú husle. Hovorí sa tomu, že nástroj má sólo.

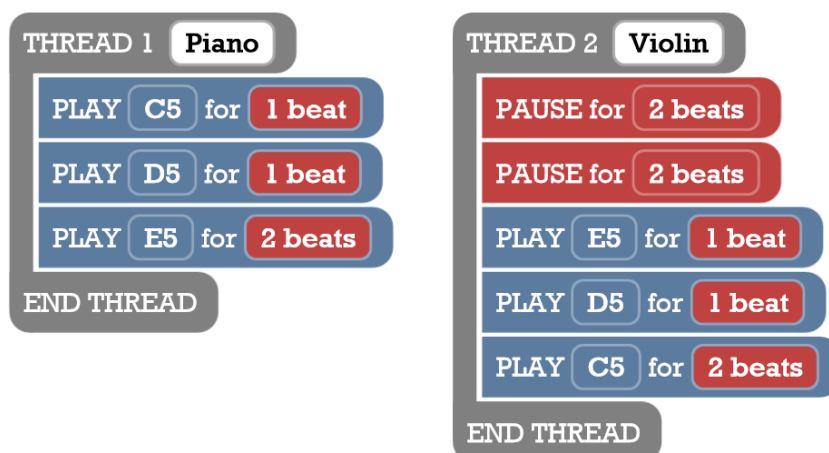
Teraz naprogramujeme pieseň **Medved'ku ...** tak, že prvé tri tóny zahrá klavír a druhé tri tóny zahrajú husle. Najprv jeden žiak vytvorí vlákno pre klavír a potom druhý žiak vytvorí vlákno pre husle, ale zatiaľ bez príkazu Pauza.



Opýtame sa žiakov:

- Aký príkaz potrebujeme, aby husle hrali až potom, keď klavír skončí?
- Ako dlho má byť pauza v postupnosti príkazov pre husle?
- Prečo tak dlho?

Medved'ku, daj labku

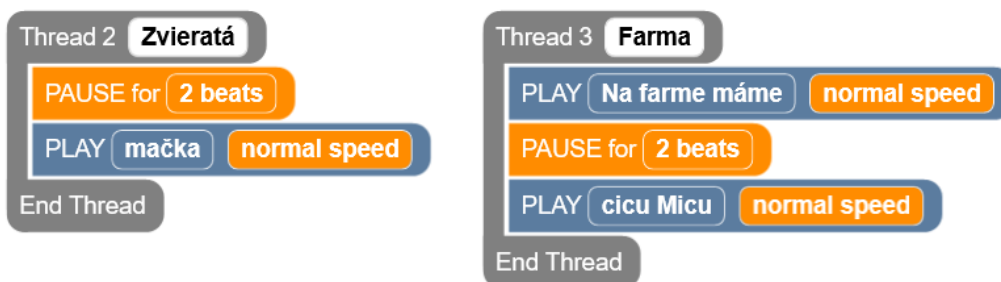


Keď žiaci vedia používať príkaz Pauza, povieme im, že vytvoríme krátky príbeh o farme. Použijeme na to dve vlákna, v jednom nastavíme sadu zvukov **Zvieratá** (Custom sounds / Zvieratá) a v druhom sadu zvukov **Farma** (Custom sounds / Farma). Príkazy v jednom vlákne vedia prehrávať zvuky zvierat a príkazy v druhom vlákne vedia hovoriť mená zvierat. Môžeme im prehrať hotový príbeh z nahrávky a požiadať ich, aby taký istý vytvorili.

Opýtame sa žiakov:

- Aký bude prvý príkaz vo vlákne so zvukmi zvierat?
- Aký bude prvý príkaz vo vlákne s pomenovaniami zvierat?
- Aká dlhá bude pauza na začiatku vlákna so zvukmi zvierat?
- Aký bude druhý príkaz vo vlákne so zvukmi zvierat?
- Aký bude druhý príkaz vo vlákne s pomenovaniami zvierat?
- Aká dlhá bude pauza vo vlákne s pomenovaniami zvierat?
- Aký bude tretí príkaz vo vlákne s pomenovaniami zvierat?
- Ako môžeme príkazy zmeniť tak, aby sa ozvalo iné zvieratko a jeho pomenovanie?

Na farme



Šikovnejší žiaci môžu vytvoriť strašidelný príbeh o Aničke zložený z dvoch vláken – v jednom budú zvuky (Sample sounds / People sounds) a v druhom rozprávanie (Custom sounds / Príbeh).

Príbeh



Záver

Zhodnotíme prácu žiakov. Povieme, že naprogramovali ozajstné pesničky a príbehy, v ktorých sa vykonáva viac dejov súčasne. Opýtame sa, čo je vlákno. Povieme žiakom, že aj počítač vie vykonávať viac činností súčasne. Čo by sa stalo, ak by počítač nedokázal vykonávať viac programov súčasne?

Dôležité je, aby žiaci prišli na to, že:

- by toho počítač za rovnaký čas stihol spraviť oveľa menej,
- nemohli by sme napríklad počúvať hudbu a zároveň písať, prípadne sťahovať z internetu nejakú pesničku a zároveň si čítať rozprávku.

Hodina 4 – Hľadanie chyby v programe

Cieľ

- Naučiť sa nájsť chybu v programe
- Naučiť sa opraviť chybu v programe
- Diskutovať o rôznych chybách, ktoré sa môžu v programoch nachádzať
- Zoznámiť sa s pojmom bug a debugovanie

Priebeh hodiny

Opakovanie

Zopakujeme si, čo sme robili na predchádzajúcich hodinách.

- Vytvárali sme program pomocou modulov zo sady Code Jumper.
- Na príkazovom module sme sa naučili nastavovať zvuk a dĺžku jeho trvania
- Naučili sme sa vytvárať programy pozostávajúce z viacerých vlákien
- Spoznali sme príkaz Pauza, ktorý sme používali na to, aby sa dva rôzne zvuky neprehrali súčasne.

Motivácia

Povieme žiakom, že včera sme so žiakmi inej školy vytvárali rôzne programy a oni v každom programe spravili nejakú chybu. Poprosíme žiakov, aby nám pomohli tie chyby opraviť.

Úloha 1 - Nesprávny príkaz – nesprávny zvuk

Budeme používať zvuky z kategórie **ABC_song** (Custom sounds / ABC song). Ešte pred začiatkom hodiny vytvoríme takýto program, v ktorom je potrebné nájsť, ktorý príkaz je nesprávny a opraviť ho.

Pýtame sa žiakov:

- Ktorý príkaz je nesprávny? (štvrtý v poradí)
- Prečo je nesprávny? (má sa zahrať zvuk d)
- Ako treba príkaz opraviť? (treba nastaviť správny zvuk otočením kolieska na nastavovanie zvukov)



Úloha 2 - Nesprávny príkaz – nesprávna dĺžka

Keď žiaci opravujú chybu, pochválime ich a povieme im, že včera žiaci urobili aj nasledujúcu chybu (keď budeme nastavovať chybu v programe, treba zoslabiť zvuk, aby žiaci nepočuli, čo nastavujeme).

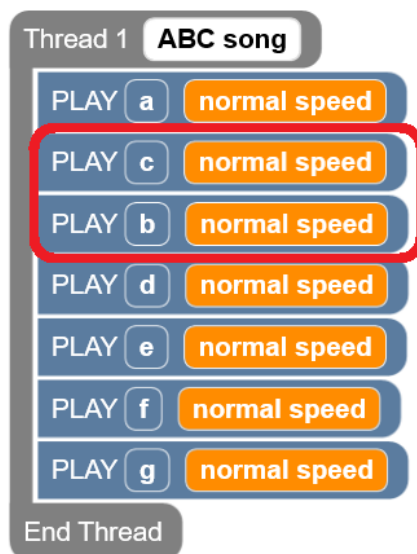
Opäť žiakom kladieme rovnaké otázky, ako v predošlej úlohe. Tentokrát je chyba v druhom príkaze, kde je nastavená nesprávna dĺžka.



Úloha 3 - Nesprávne poradie príkazov

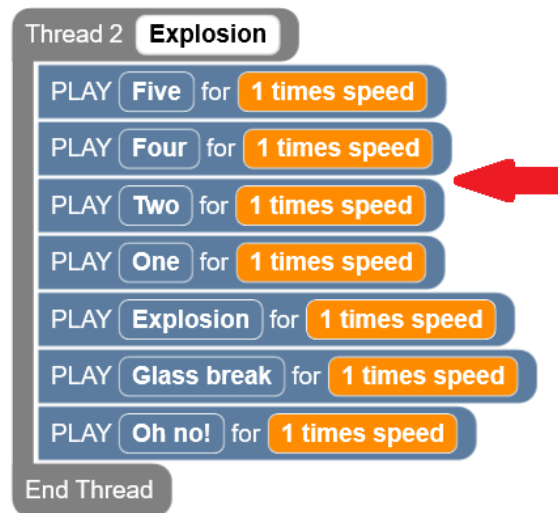
Opäť zoslabíme hlasitosť a zmeníme poradie druhého a tretieho príkazu. Opýtame sa žiakov:

- Kde je chyba?
- Ako sa dá program opraviť?
- Je aj iný spôsob ako opraviť program? (výmena príkazov alebo prestavenie zvuku na oboch príkazoch)



Úloha 4 - Chýbajúci príkaz

Opäť zoslabíme zvuk a vytvoríme program znázornený na obrázku (pre vlákno nastavíme kategóriu Explosion zo skupiny Sample Sounds). Žiaci majú prísť na to, že ako tretí v poradí má byť zvuk Three. Opäť diskutujeme so žiakmi o tom, kde je chyba a ako ju najjednoduchšie opraviť. Jedna možnosť je vložiť príkazový modul Zahraj so správnym zvukom na tretie miesto v poradí. Druhá možnosť je zmeniť zvuky na všetkých príkazových moduloch počnúc tretím, končiac posledným a na koniec pridať príkazový modul so zvukom Oh No.



Správne vytvorený program má vyzerať nasledovne.



Úloha 5 - Nadbytočný príkaz

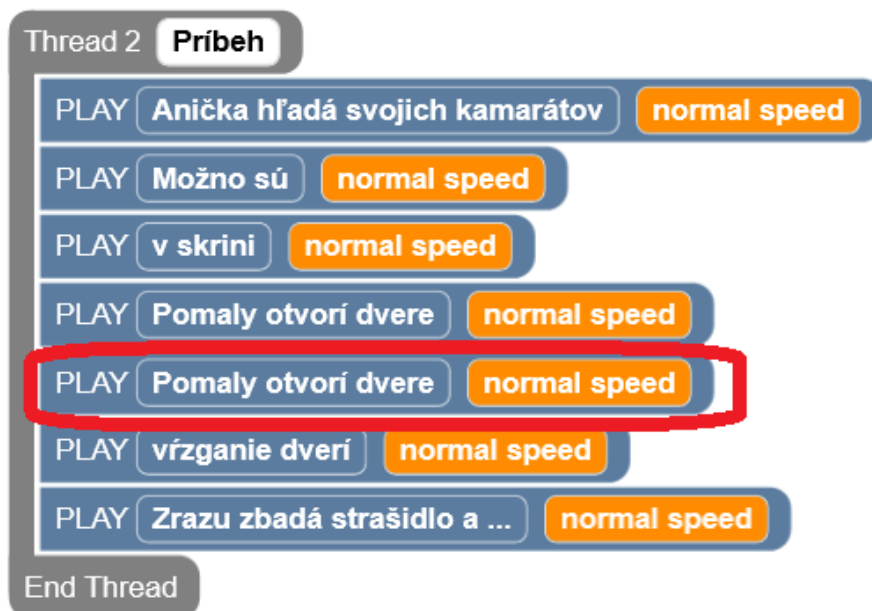
V nasledujúcich programoch je vždy nejaký príkaz navyše. Žiaci ho majú nájsť a program opraviť. Opäť im pomáhamo otázkami.

- Ktorý príkaz do program nepatrí?
- Prečo?
- Čo s ním urobíme?

V nasledujúcom programe je zbytočný príkaz na konci programu. Potrebne je nastaviť sadu zvukov **Kŕmenie1** (Custom sounds / Kŕmenie 1) do jedného z vláken.



V nasledujúcom programe je nadbytočný príkaz vo vnútri programu (kategória zvukov Custom sounds / Príbeh).



V nasledujúcom programe je nadbytočný príkaz Pauza (kategória zvukov Sample sounds / Twinkle, twinkle).



Záver

Zhodnotíme prácu žiakov. Pochválime ich, že úspešne našli v programoch veľa chýb a opravili ich. Opýtame sa, aké typy chýb sa dajú spraviť v programoch (nesprávny príkaz, nesprávne poradie príkazov, chýbajúci príkaz, zbytočný príkaz a podobne). Povieme žiakom, že aj ozajstní programátori trávajú veľa času hľadaním a odstraňovaním chýb. Programátori nazývajú chybu v programe **bug** a proces ich odstraňovania volajú **debugovanie** alebo ladenie programu. Môžeme sa ešte so žiakmi rozprávať o tom, kedy a kde oni robia chyby a ako ich opravujú.

Dôležité je, aby žiaci prišli na to, že:

- robiť chyby je prirodzené
- chyba sa dá nájsť iba trpezlivým analyzovaním toho, čo vykonajú jednotlivé príkazy v programe
- väčšinou je jednoduchšie chybu nájsť a opraviť, ako vytvoriť celý program od začiatku.

Hodina 5 - Cyklus

Cieľ

- Spoznať príkaz cyklu v jazyku Code Jumper (vedieť ho zapojiť a skombinovať so základnými príkazmi)
- Naučiť sa použiť príkaz cyklu v programoch
- Diskutovať o tom, kedy je vhodné použiť príkaz cyklu
- Zoznámiť sa s pojmom cyklus s pevným počtom opakovaní

Priebeh hodiny

Opakovanie

Zopakujeme si, čo sme robili na predchádzajúcich hodinách.

- Programovali sme pesničky a príbehy.
- Hľadali sme chyby v programoch a opravovali sme ich.

Motivácia

Povieme žiakom, že naprogramujeme zvuk hodín. Opýtame sa ich, aký zvuk vydávajú hodiny. Očakávame, že povedia, že hodiny tikajú, oznamujú čas. Opýtame sa, ako hodiny oznamujú čas. Očakávame, že žiaci povedia – odbíjajú toľkokrát, koľko je práve hodín.

Úloha 1 - Hodiny oznamujú presný čas

V jednom z vláken nastavíme kategóriu zvukov Hodiny (Custom sounds / Hodiny) a vyzveme žiakov, aby si prehrali zvuky v tejto kategórii tak, že ich budú nastavovať na jednom príkazovom module. Požiadame žiakov, aby okomentovali, čo tie zvuky predstavujú.

Povieme žiakom, že ideme vytvoriť program, ktorý napodobní hodiny oznamujúce čas 8 hodín. Žiaci by mali navrhnúť, že 8-krát odbije zvon na veži alebo na izbových hodinách. Najprv necháme žiakov, aby vytvorili program bez použitia cyklu.

Opýtame sa žiakov:

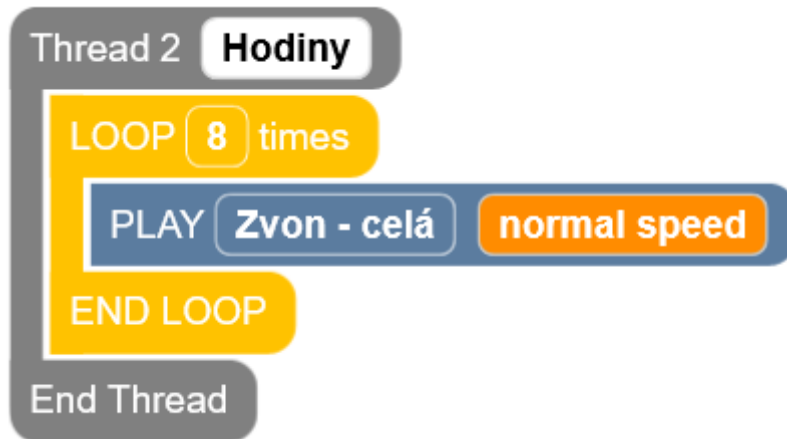
- Opakuje sa v programe nejaký zvuk?
- Aký zvuk sa opakuje?
- Koľkokrát sa tento zvuk opakuje?
- Stačil by nám len jeden príkaz, ktorý by sa prehrával opakovane 8-krát?

Zoznámime žiakov s novým modulom, ktorý umožňuje niekoľkokrát opakovať rovnaký príkaz.

- Vysvetlíme jednotlivé časti nového modulu – príkazu opakovania.
- Vysvetlíme, ako zapojiť príkaz, ktorý sa má opakovať.
- Vysvetlíme, ako nastaviť počet opakovaní.



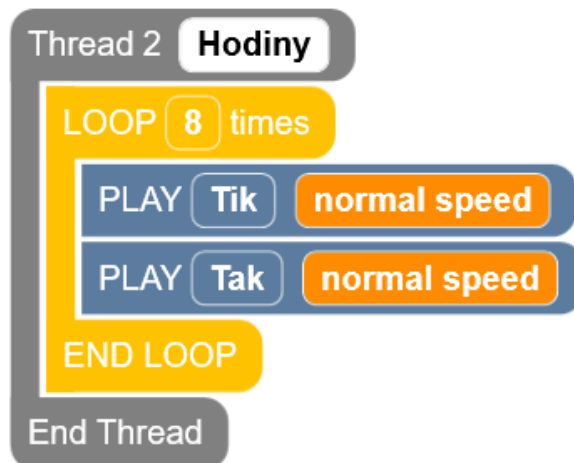
Žiaci vytvoria nasledujúci program s použitím príkazu cyklu.



Úloha 2 - Hodiny tikajú 8-krát

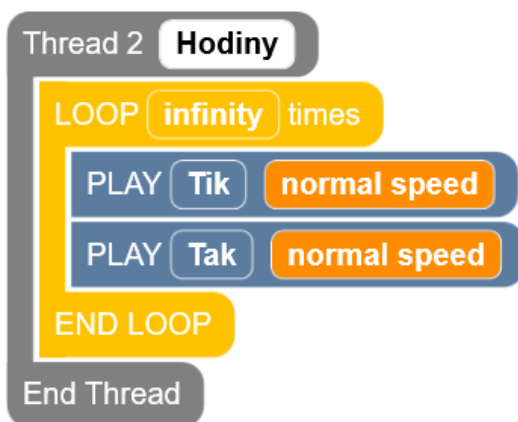
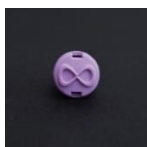
Požiadame žiakov, aby vytvorili program, ktorý napodobní zvuk tikajúcich hodín.

- Aká postupnosť zvukov sa bude opakovať? (Tik-Tak)
- Koľkokrát sa bude postupnosť opakovať? (najviac to môže byť 8-krát)



Úloha 3 - Hodiny tikajú stále (donekonečna)

Hodiny však netikajú len istý počet krát, ale tikajú stále. Vysvetlíme žiakom, že môžu vytvoriť nekonečný cyklus tak, že do kolieska na príkaze cyklu zasunú znak nekonečna.



Úloha 4 - Perkusie

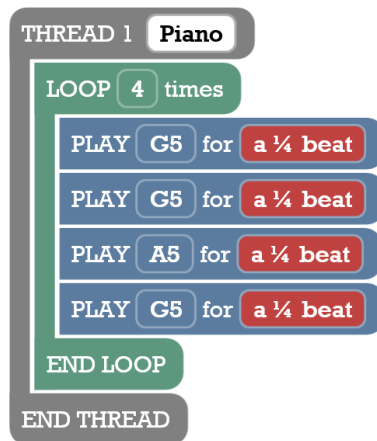
V jednom z vláken nastavíme zvuky z kategórie Percussion (Sample sounds / Percussion). Žiaci majú za úlohu vytvoriť program, ktorý nahradí hráča na bicie nástroje v rockovej kapele. V programe sa majú opakovať (niekoľkokrát alebo stále) tri rôzne zvuky.

- Žiaci si najprv zvolia tri zvuky, ktoré sa budú opakovať.
- Opakujúce sa zvuky vložia do cyklu.
- Nastavia počet opakovaní.



Úloha 5 - Naprogramujme pieseň

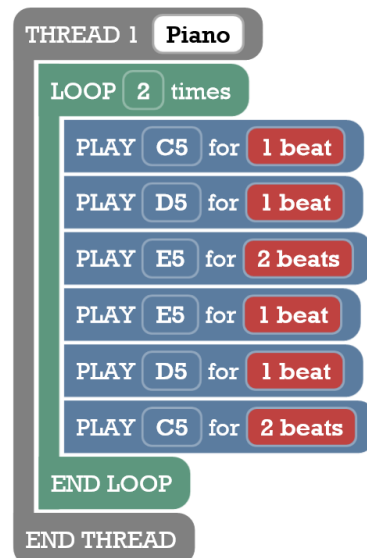
Požiadame žiakov, aby naprogramovali úvodnú časť piesne **Prší, prší ...**



Požiadame žiakov, aby naprogramovali úvodnú časť piesne **Are you sleeping ...**



Požiadame žiakov, aby naprogramovali úvodnú časť piesne **Medved'ku ...**



Záver

Zhodnotíme prácu žiakov. Povieme im, že sa naučili vytvárať programy, v ktorých sa nejaká postupnosť príkazov opakuje. Opýtame sa žiakov, či aj oni nejaké činnosti opakujú viackrát za sebou (napríklad dýchanie, prehĺtanie pri pití, kráčanie, ...). Opýtame sa žiakov, aké časti má príkaz cyklu. Tiež sa opýtame aké výhody má použitie príkazu cyklu.

Dôležité je, aby žiaci prišli na to, že ak by sme nemohli použiť príkaz cyklu:

- programy by boli veľmi dlhé,
- museli by sme veľa krát vytvárať rovnakú postupnosť príkazov.

Tiež by si mali uvedomiť, že pri použití príkazu cyklu, musíme vedieť:

- aká postupnosť príkazov sa má opakovať,
- koľkokrát sa má postupnosť príkazov opakovať.

Hodina 6 - Sekvencia príkazov pred a za cyklom

Cieľ

- Precvičiť si použitie cyklu v jazyku Code Jumper (správne zapojenie a skombinovanie so základnými príkazmi)
- Naučiť sa použiť príkaz cyklu v programoch a kombinovať ho sekvenciami príkazov **pred cyklom, za cyklom a v cykle**
- Diskutovať o tom, čo sa vykoná v cykle, čo pred ním a čo za ním

Priebeh hodiny

Priebeh hodiny môže byť nasledovný.

Opakovanie

Zopakujeme si, čo sme robili na predchádzajúcich hodinách.

- Programovali sme zvuky hodín, ktoré sa opakovali.
- Použili sme nový príkaz Opakuj.

Motivácia

Povieme žiakom, že naprogramujeme básničky a príbehy v ktorých sa niektoré časti opakujú.

Úloha 1 - Sekvencia príkazov za cyklom

Budeme používať zvuky z kategórie **KotKot_CinČará** (Custom sounds / Kot-Kot Čin-Čará) vo vlákne 1. Rozdáme žiakom papier s básničkami. Vyzveme ich, aby zapojili príkazový modul do vlákna 1 a vypočuli si zvuky, ktoré sa dajú nastaviť.

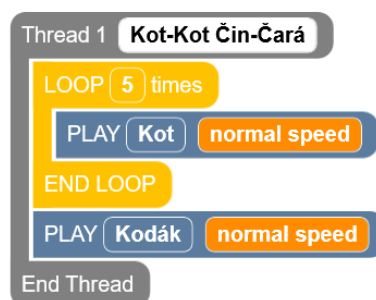
Úloha 1a

Vyzveme žiakov, aby naprogramovali nasledujúcu časť riekanky:

Kot- Kot- Kot- Kot- Kot-Kodák

Opýtame sa žiakov

- Aké zvuky musíme nastaviť v príkazových moduloch?
- Aký zvuk sa bude opakovať? Kde tento zvuk nastavíme?
- Koľko krát sa bude zvuk opakovať? Kde nastavíme počet opakovaní?
- Kam zapojíme modul so zvukom **Kodák**?

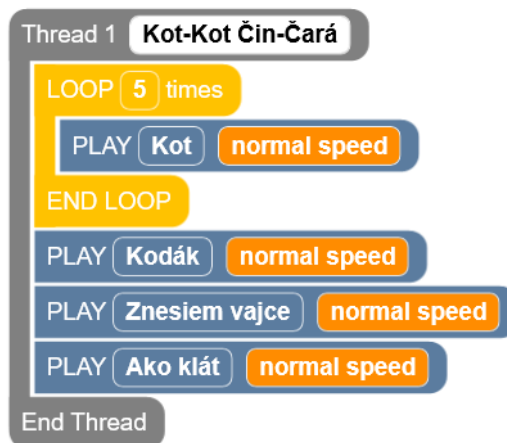


Potom vyzveme žiakov, aby doplnili program tak, že Code Jumper povie

***Kot- Kot- Kot- Kot- Kot-Kodák
Znesiem vajce ako klát***

Opýtame sa žiakov

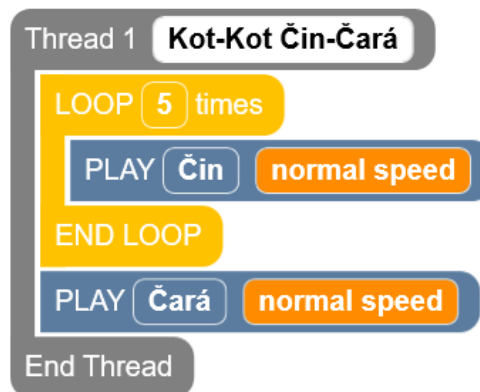
- Koľko ďalších modulov potrebujeme?
- Aké zvuky na nich nastavíme?
- Kam zapojíme tieto moduly?



Úloha 1b

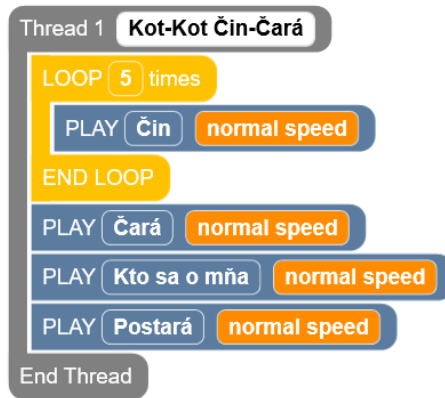
Obmenou tejto aktivity môže byť naprogramovanie časti riekanky

Čin-Čin-Čin-Čin-Čin-Čará



Potom treba doplniť riekanku, aby znela nasledovne:

***Čin- Čin- Čin- Čin- Čin-Čará
Kto sa o mňa postará?***

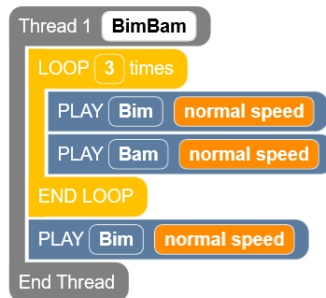


Úloha 1c

Ďalšia obmena úlohy si vyžaduje do cyklu zapojiť dva príkazy. Vo vlákne 1 treba nastaviť kategóriu zvukov **BimBam** (Custom sounds / BimBam).

Žiaci majú naprogramovať nasledujúcu časť riekanky:

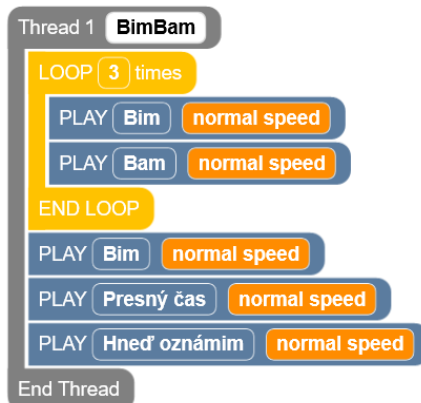
Bim-Bam Bim-Bam Bim-Bam Bim



Žiaci majú doplniť riekanku, aby znela nasledovne:

Bim-Bam Bim-Bam Bim-Bam Bim

Presný čas hneď oznámim



Úloha 2 - Sekvencia príkazov pred cyklom

V niektorom vlákne treba nastaviť kategóriu zvukov **Vystava** (Custom sounds / Výstava). Vyzveme žiakov, aby zapojili príkazový modul do tohto vlákna a vypočuli si zvuky, ktoré sa dajú nastaviť.

Úloha 2a

Vyzveme žiakov, aby naprogramovali Code Jumper tak, že povie

Sme na výstave psov.

Je tu bernardýn.

Haf haf haf

Opýtame sa žiakov

- Aké zvuky musíme nastaviť v príkazových moduloch?
- Aký zvuk sa bude opakovať? Kam zapojíme príkazový modul s týmto zvukom?
- Koľko krát sa bude zvuk opakovať? Kde nastavíme počet opakovaní?
- Kam zapojíme zvuky **Sme na výstave psov**, **Je tu, bernardín?**
- Aké trvanie musíme nastaviť zvuku **Haf**, aby znel ako bernardín?



Ďalší žiaci môžu potom meniť druh psa (kólia, čivava) a podľa toho prispôbiť jeho štekание nastavením vhodnej dĺžky pre slovo **Haf**. Môžu tiež meniť počet opakovaní v cykle.

Úloha 2b

V niektorom vlákne treba nastaviť kategóriu zvukov **Rodina** (Custom sounds / Rodina).

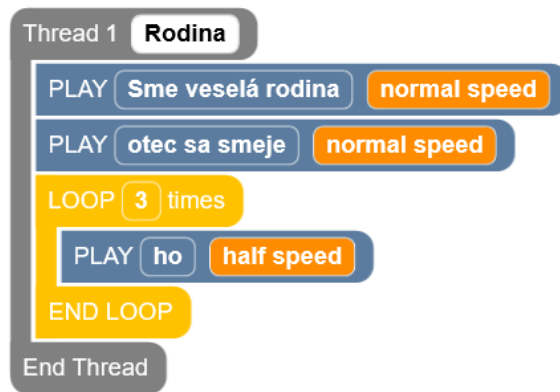
Obmenou predchádzajúcej úlohy je príbeh o veselej rodine, ktorý znie nasledovne:

Sme veselá rodina

Otec sa smeje

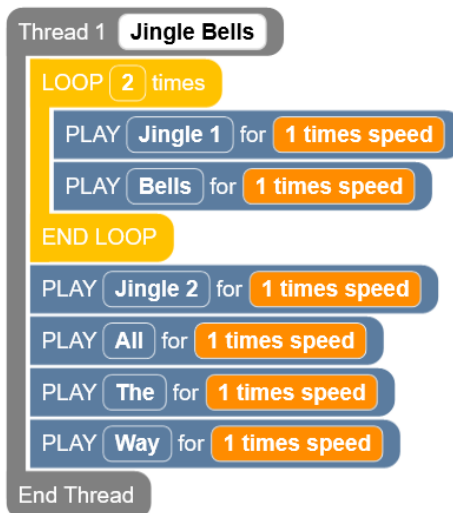
Ho ho ho

Pre slovo **ho** treba nastaviť takú rýchlosť, aby bol hlas hlboký. Potom sa môže program zmeniť tak, aby sa smial iný člen rodiny vyšším hlasom.



Úloha 3 - Programovanie piesne

Použitie cyklov si môžeme precvičiť na programovaní pesničky Jingle Bells (Sample sounds / Jingle Bells), ktorú žiakom pustíme z nahrávky.



Záver

Zhodnotíme prácu žiakov. Povieme im, že sa naučili vytvárať programy, v ktorých sa nejaká časť programu opakuje. Opýtame sa žiakov, ako vieme, kedy v programe použiť príkaz cyklu.

Dôležité je, aby žiaci prišli na to, že:

- Opakuje sa len sekvencia príkazov nasledujúcich bezprostredne za sebou. Medzi opakujúcimi sa príkazmi nemôže byť iný príkaz alebo iná sekvencia príkazov.

Tiež by si mali uvedomiť, že pri použití príkazu cyklu, musíme vedieť:

- aká postupnosť príkazov sa má opakovať,
- koľkokrát sa má postupnosť príkazov opakovať,
- aké príkazy sa nachádzajú pred opakujúcou sa sekvenciou a aké po nej.

Hodina 7- Dekompozícia

Cieľ

- Naučiť sa riešiť komplexnejšie úlohy pozostávajúce z viacerých podproblémov
- Naučiť sa rozdeliť úlohu na podúlohy, tie vyriešiť a skombinovať riešenia menších úloh do riešenia komplexnej úlohy

Priebeh hodiny

Opakovanie

Zopakujeme si, čo sme robili na predchádzajúcich hodinách.

- Programovali sme postupnosti zvukov.
- Na opakujúce sa zvuky alebo sekvencie zvukov sme sa naučili použiť príkaz cyklu.
- Na sekvencie zvukov, ktoré mali znieť súčasne sme použili viaceré vlákna.

Motivácia

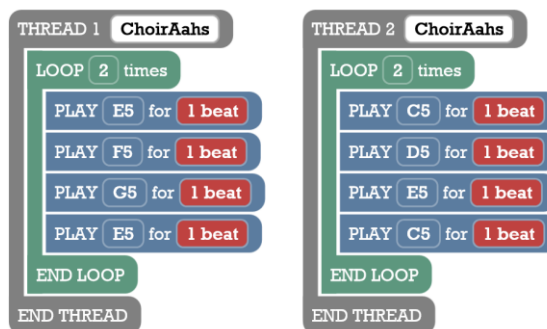
Povieme žiakom, že naprogramujeme skladby, ktoré hrajú viaceré nástroje súčasne, alebo je v nich viachlasný spev.

Úloha 1 - Programujeme dvojhlas

Žiakom prehráme výslednú melódiu – úryvok piesne **Are you sleeping** ... Žiakom sa to asi bude zdať komplikované, tak im prehráme ukážky jednotlivých hlasov (vláken) samostatne a povieme im, že naprogramujeme každú melódiu zvlášť.

Pýtame sa žiakov:

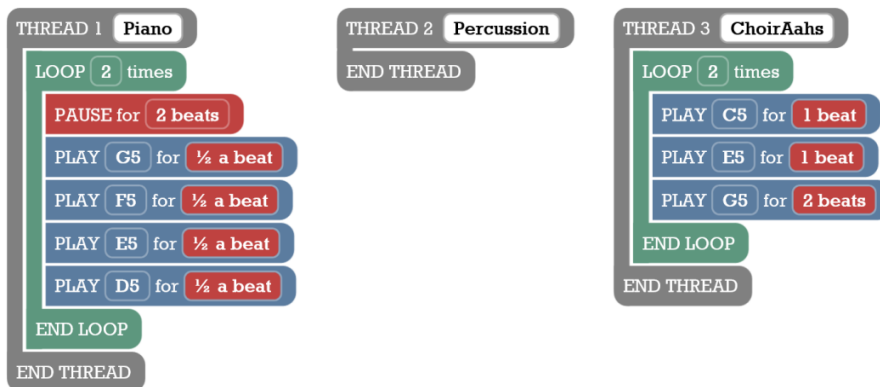
- Opakuje sa nejaká časť programu? Aká?
- Koľko tónov sa opakuje? Koľko príkazových modulov potrebujeme?
- Aké tóny treba nastaviť na príkazových moduloch?
- Ako zapojíme postupnosť modulov do príkazu opakovania?



Poznámka: V dvoch vláknach nastavíme kategóriu zvukov Choir Ahhs (MIDI instruments / Choir Ahhs).

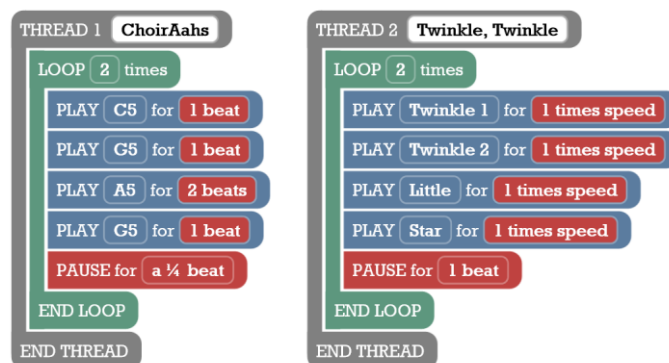
Úloha 2 - Programujeme spev s klavírnym doprovodom

Žiakov necháme samostatne naprogramovať pieseň **Ovčáci ...**, prípadne nasledujúcu pieseň **Twinkle, twinkle ...**. Treba im však pustiť nahrávku hotovej piesne a aj každého vlákna zvlášť.



Úloha 3 - Programujeme spev s vokálmi

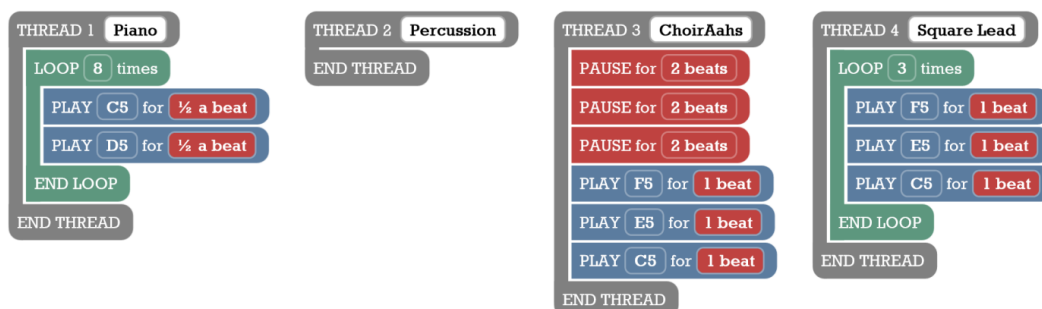
Žiaci samostatne naprogramujú pieseň **Twinkle twinkle ...**, ale pustíme im nahrávku hotovej piesne a aj každého vlákna zvlášť.



Poznámka: V jednom vlákne nastavíme kategóriu zvukov Choir Ahhs (MIDI instruments / Choir Ahhs) a v druhom Twinkle, Twinkle (Sample sounds / Twinkle, Twinkle).

Úloha 4 - Programujeme trio

Veľmi šikovní žiaci môžu naprogramovať nasledujúce trio. Opäť im pustíme ukážku hotovej skladby a aj ukážky jednotlivých vlákien.



Záver

Zhodnotíme prácu žiakov. Povieme im, že sa naučili vytvárať programy, ktoré spočiatku vyzerajú ako veľmi zložité, ale keď si ich rozdelíme na menšie časti a tie riešime samostatne, nie je to ťažké. Postupne dokážeme vytvoriť veľký program pozostávajúci z viacerých vláken, cyklov a postupností príkazov.

Hodina 8 – Príkaz vetvenia

Cieľ

- Naučiť sa vyhodnotiť jednoduché podmienky ako výsledok porovnania dvoch čísel.
- Naučiť sa určiť, aká sekvencia príkazov sa vykoná v jednotlivých vetvách príkazu vetvenia.
- Naučiť sa určiť, aká sekvencia príkazov sa vykoná pred, prípadne po príkaze vetvenia.
- Naučiť sa použiť príkaz vetvenia v jazyku Code Jumper.

Priebeh hodiny

Opakovanie

Zopakujeme si, čo sme robili na predchádzajúcich hodinách.

- Programovali sme postupnosti zvukov.
- Na opakujúce sa zvuky sme sa naučili použiť príkaz cyklu.
- Na sekvencie zvukov, ktoré mali znieť súčasne sme použili viaceré vlákna.

Motivácia

Povieme žiakom, že Code Jumper naučíme porovnať dve čísla. Použijeme na to nový príkaz – **Podmienka**.

Úloha1 – Porovnáваме čísla

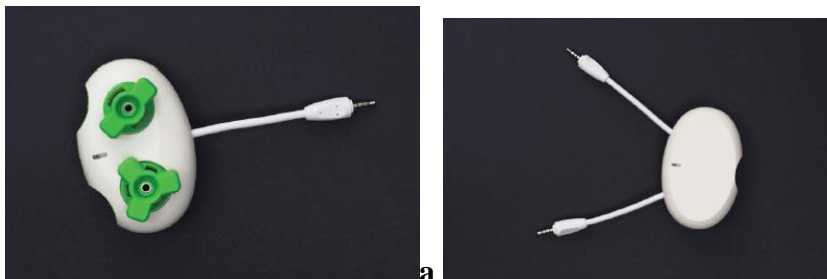
V jednom z vlákien nastavíme kategóriu **Čísla** a požiadame žiakov, aby do tohto vlákna zapojili jeden príkazový modul a vypočuli si zvuky, ktoré sa dajú nastaviť.

Povieme žiakom, že chceme vytvoriť program, ktorý bude vedieť porovnať dve čísla a povie, ktoré z nich je väčšie.

Opýtame sa žiakov:

- Dokážeme takýto program vytvoriť len pomocou príkazov, ktoré poznáme?
- Potrebujeme nový príkaz? Ako by mal ten príkaz fungovať a ako by mal vyzeráť?

Ukážeme žiakom príkazový modul **Podmienka** a necháme ich, aby si ho ohmatali.



Opýtame sa žiakov:

- Koľko káblikov vchádza do modulu **Podmienka**?

- Koľko modulov môžeme pripojiť do modulu **Podmienka**?

Požiadame žiakov, aby modul **Podmienka** pripojili do základného modulu a do každého konektora pripojili jeden príkazový modul.

Požiadame žiakov, aby v každom príkazovom module nastavili iný zvuk a program spustili.

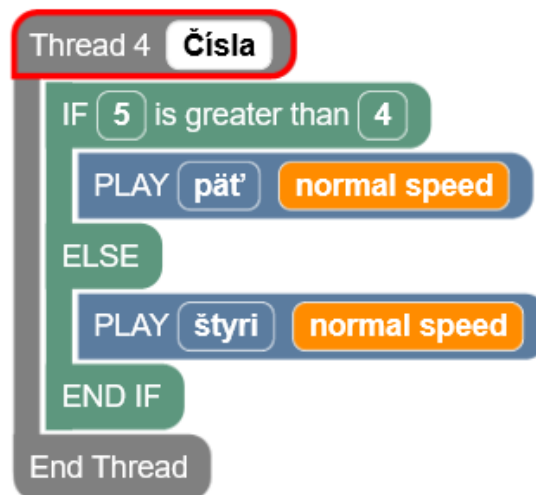
Opýtame sa žiakov:

- Čo sa stalo?
- Ako to funguje?
- Čo potrebujeme, aby sme mohli porovnať dve čísla? Pomohli by nám takéto konektory s číslami?

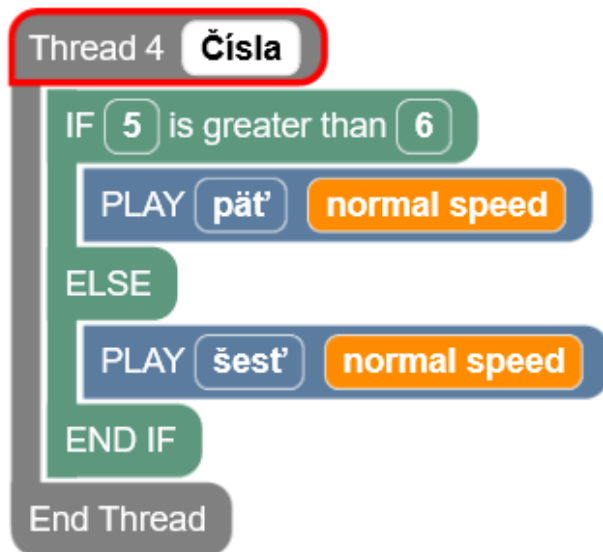


Požiadame žiakov, aby:

- konektory s číslami 5 a 4 zasunuli do dierok na module **Príkaz**,
- v príkazových moduloch, ktoré sú pripojené do modulu **Príkaz**, aby nastavili vyslovovanie čísel 5 a 4 tak, aby Code Jumper vyslovil vyššie číslo.



Požiadame žiakov, zmenili program tak, aby sa porovnali čísla 5 a 6, prípadne úplne iné čísla.



Úloha 2 – Aký zvuk zvieratka sa zahrá?

Vytvorte program, ktorý porovná dve čísla a buď zahrá zvuk psa, alebo kohúta (Sample sounds / Animals).



Opýtame sa žiakov:

- Aký zvuk sa zahrá?
- Prečo sa zahrá zvuk psa?
- Ako treba program zmeniť, aby sa zahrá zvuk kohúta?

```
Thread 4 Animals
IF 5 is greater than 6
  PLAY Dog for 1 times speed
ELSE
  PLAY Rooster for 1 times speed
END IF
End Thread
```

Úloha 3 – Náhodný výber zvieratka

Zmeňte predchádzajúci program tak, aby sme vopred nevedeli, aký zvuk sa zahrá. Dá sa povedať, že Code Jumper náhodne vyberie zvuk, ktorý sa zahrá.

Budeme potrebovať kolík, ktorý náhodne vyberie niektoré číslo od 1 do 8.



```
Thread 4 Animals
IF 5 is greater than random
  PLAY Lion for 1 times speed
ELSE
  PLAY Cat for 1 times speed
END IF
End Thread
```

Opýtame sa žiakov:

- Zvuk ktorého zvieratka bude zniet' častejšie?
- Ako treba program zmeniť, aby častejšie zaznel zvuk mačky?

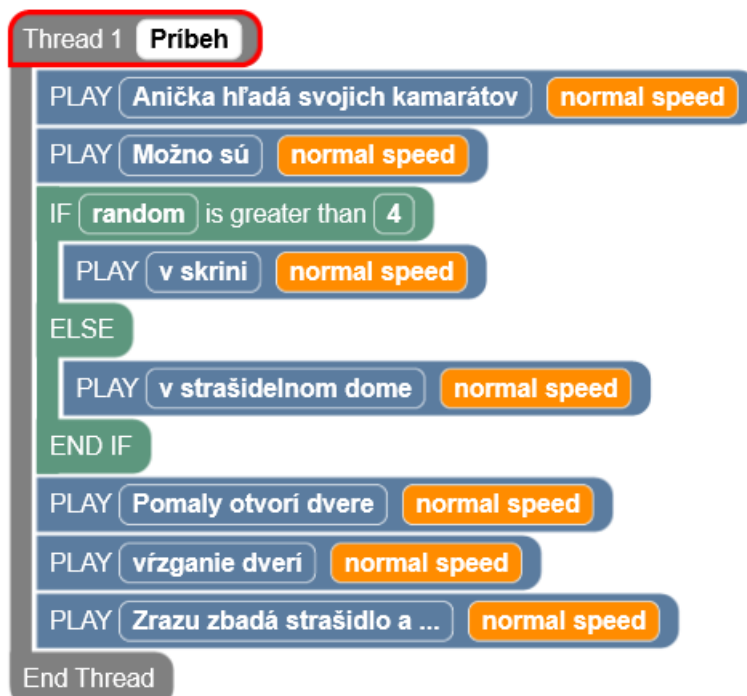
Úloha 4 – Náhodný výber melódie

Vytvorte program, ktorý náhodne vyberie, či klavír zahrá postupnosť tónov C5 E5 G5 alebo postupnosť tónov G5 E5 C5 (MIDI instruments / Piano).

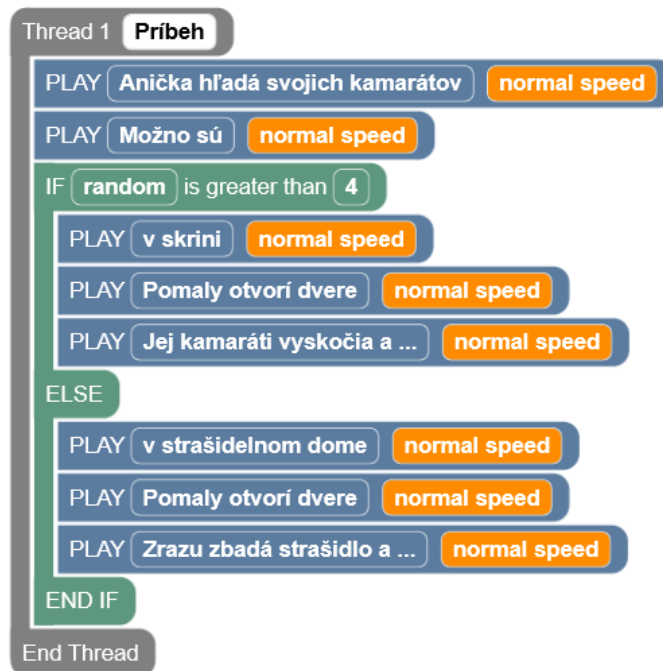


Úloha 5 – Náhodný výber pokračovania príbehu

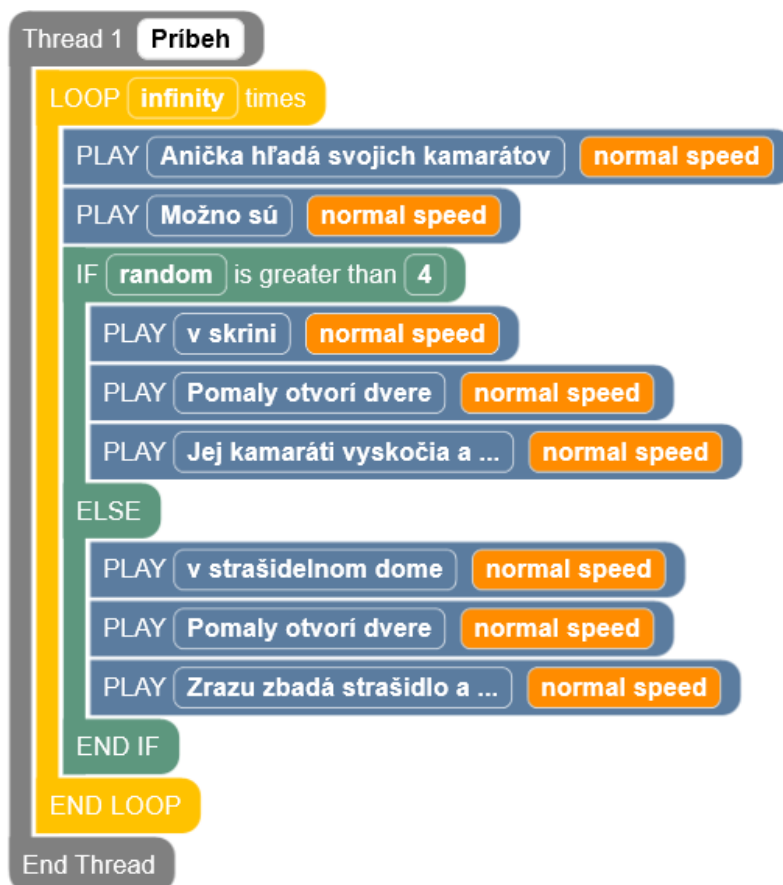
Vytvorte program o Aničke, ktorá hľadá svojich priateľov. V programe sa náhodne vyberie, či Anička otvorí skriňu alebo strašidelný dom (Custom sounds / Príbeh).



V jednotlivých vetvách môže byť aj viac ako jeden príkaz. Program teda môže vyzeráť aj nasledovne.



Šikovnejších žiakov možno napadne celý program vložiť do nekonečného cyklu.



Záver

Zhodnotíme prácu žiakov. Povieme, že sa naučili vytvárať programy, v ktorých sa až počas behu programu rozhodne o tom, čo sa bude diať. Dá sa to vďaka príkazu vetvenia.

Opýtame sa žiakov:

- Prečo je užitočné v programoch používať príkaz vetvenia?
- Stáva sa aj v živote, že nevieme, čo sa bude diať v nasledujúcom momente?
- Aký je rozdiel medzi vláknami zapojenými do základného modulu a vláknami zapojenými do modulu **Podmienka**?

Hodina 9 – Čary s číslami

Cieľ

- Naučiť sa v programoch používať konštanty a operácie s číslami (náhodný výber, inkrementácia, dekrementácia).
- Naučiť sa v jazyku Code Jumper používať konštanty a operácie s číslami a zapojiť ich do príkazového modulu pripojeného do sekvencie príkazov alebo do cyklu.

Priebeh hodiny

Opakovanie

Zopakujeme si, čo sme robili na predchádzajúcich hodinách.

- Programovali sme postupnosti zvukov.
- Na opakujúce sa zvuky sme sa naučili použiť príkaz cyklu.
- Na sekvencie zvukov, ktoré mali znieť súčasne sme použili viaceré vlákna.
- Vytvárali sme programy s náhodným výberom vetvy, ktorá sa vykoná.

Motivácia

Povieme žiakom, že na minulej hodine sme používali konektory s číslami v príkaze vetvenia. Preskúmame teraz, ako môžeme tieto konektory použiť v príkaze Zahraj.

Úloha 1 – Experimentujme s číslami

V jednom z vlákien nastavíme kategóriu **Čísla** (Custom sounds / Čísla) a požiadame žiakov, aby do tohto vlákna zapojili jeden príkazový modul a vypočuli si zvuky, ktoré sa dajú nastaviť.

Opýtajme sa žiakov:

- Čo sa stane, ak do kolieska na nastavovanie zvukov v príkazovom module vsunieme konektor s číslom 1?
- Čo sa stane, ak do kolieska na nastavovanie zvukov v príkazovom module vsunieme konektor s číslom 5?
- Čo sa stane, ak do kolieska na nastavovanie zvukov v príkazovom module vsunieme konektor s písmenom R?



Úloha 2 – Experimentujme s tónmi

Teraz vo vlákne nastavme kategóriu zvukov **Piano** (MIDI sounds / Piano) a zopakujme činnosti, ktoré sme robili v predchádzajúcej úlohe.

Opýtajte sa žiakov:

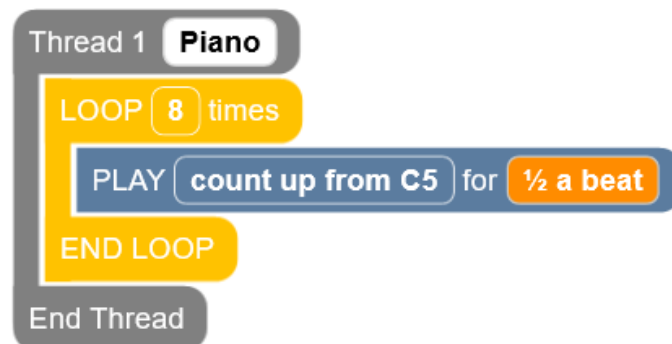
- Aký tón je priradený k číslu 1?
- Aký tón je priradený k číslu 2?
- Aký tón je priradený k číslu 8?
- Čo sa stane, ak príkazový modul Zahraj zapojíme do cyklu a vsunieme do krúžku na výber zvuku konektor s písmenom R?

Úloha 3 - Stupnica

Zistili sme, že každému zvuku je priradené nejaké číslo od 1 po 8. Ak použijeme nejaký konektor s číslom, zahrá sa zvuk s týmto poradovým číslom. Konektor s písmenom R náhodne vylosuje číslo zvuku, ktorý sa má zahrat'. Skúsme teraz zasunúť konektor so znakom + do príkazového modulu zapojeného v cykle.



Čo sa stane?



Úloha 4 – Prehrávanie piesne

Nastavme kategóriu zvukov **Twinkle, Twinkle** (Sample sounds / Twinkle, Twinkle), počet opakovaní v cykle nastavme na 4 a zvuk v príkaze Zahraj nastavme na Twinkle 1. Spustíme program. Čo budeme počuť? Prečo to tak funguje?



Úloha 5 – Stupnica naopak

Vytvorte program, ktorý zahrá stupnicu v opačnom poradí (MIDI instruments / Piano). Aký konektor budeme potrebovať? Aký počet opakovaní nastavíme v module Opakuj? Aký tón nastavíme v príkaze Zahraj?



Úloha 6 – Výbuch

Čo budeme počuť, ak nastavíme kategóriu Explosion (Sample sounds / Explosion)? Aký zvuk treba nastaviť v príkaze Zahraj?



Záver

Zhodnotíme prácu žiakov. Povieme, že sa naučili vytvárať programy, v ktorých sa používajú čísla a operácie s číslami. Opýtame sa žiakov, aké operácie s číslami sme sa naučili v programoch používať. Tiež sa opýtame, vďaka čomu môžeme pri práci so zvukmi používať čísla a operácie s číslami. Treba, aby si žiaci uvedomili, že zvukom sú priradené čísla. Takéto čísla voláme aj kódy. Teda zvuky sú zakódované do čísel.

Hodina 10 – Premenná

Cieľ

- Naučiť sa v programe použiť premennú na zapamätanie hodnoty.
- Naučiť sa nastaviť hodnotu premennej a meniť ju.
- Naučiť sa v jazyku Code Jumper vytvárať programy s využitím jednej premennej.

Priebeh hodiny

Opakovanie

Zopakujeme si, čo sme robili na predchádzajúcich hodinách.

- Programovali sme postupnosti zvukov.
- Na opakujúce sa zvuky sme sa naučili použiť príkaz cyklu.
- Na sekvencie zvukov, ktoré mali znieť súčasne sme použili viaceré vlákna.
- Používali sme príkaz vetvenia.
- Používali sme čísla a operácie s nimi v cykloch aj mimo nich.

Motivácia

Povieme žiakom, že na minulej hodine sme vytvárali programy, kde hral klavír náhodné tóny. Teraz vytvoríme program, v ktorom tie isté tóny bude hrať aj iný hudobný nástroj, prípadne ich bude niekto spievať.

Úloha 1 – Rovnaké tóny na dvoch nástrojoch

Budeme používať dve vlákna. V jednom nastavíme zvuky **Piano** (MIDI instruments / Piano) a v druhom **Choir Ahhs** (MIDI instruments / Choir Ahhs). Vyzveme žiakov, aby do každého vlákna zapojili jeden príkazový modul zahraj a aby pomocou konektora s číslom 4 nastavili v jednom z modulov zvuk s kódom 4.

Opýtame sa žiakov:

- Ako zabezpečíme, aby sa rovnaké tóny nastavili na oboch moduloch?

Žiaci majú prísť na to, že si potrebujeme zapamätať číslo zvuku, ktorý hrá klavír a zvuk s rovnakým číslom sa má zaspievať. Použijeme na to konektor **Premenná**. Premennú žiaci poznajú aj z matematiky, kde sa často označuje písmenom **x**.



Premenná v jazyku Code Jumper vyzerá ako konektor so zástrčkou, do ktorej môžeme vložiť konektor s konkrétnym číslom alebo číselnou operáciou.

Vložme jednu premennú do príkazového modulu vo vlákne pre klavír a druhú do príkazového modulu pre spev. Obe reprezentujú tú istú premennú **x**. Keď do jednej z nich priradíme číselnú hodnotu zasunutím konektora s číslom, táto hodnota sa použije na všetkých miestach

programu, kde sa premenná x vyskytuje. Funguje to dovedy, kým hodnotu premennej nezmeníme.

Vkladajme do jednej premennej konektory s číselnými hodnotami a sledujme, čo sa bude diať.

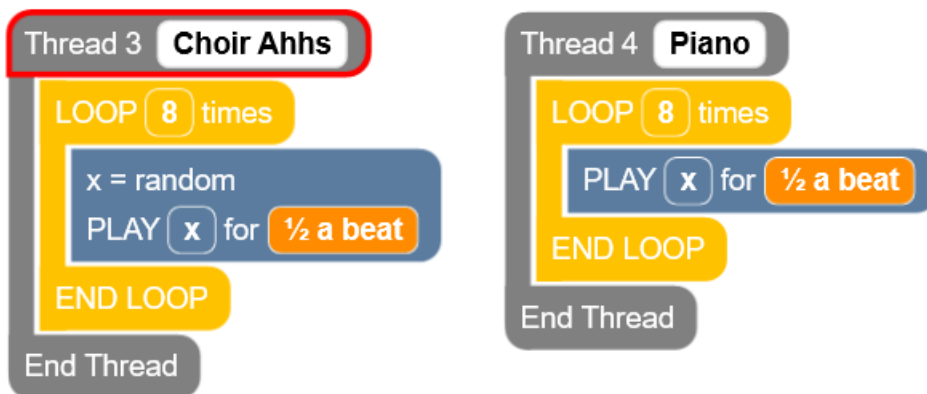
- Čo sa stane, ak do jednej premennej vložme konektor s písmenom R?

Úloha 2 - Náhodná melódia pre spev a klavír

Vytvorte program, aby sa v oboch vláknoch zahrlo 8 rovnakých náhodných tónov.

Opýtame sa žiakov:

- Ako by sme naprogramovali, aby klavír zahrlo 8 náhodných tónov?
- Ako zabezpečíme, aby sa rovnaké tóny zahrli aj vo vlákne pre spev?



Úloha 3 - Stupnica

Zmeňte program tak, aby obidva nástroje súčasne zahrli stupnicu.

```

Thread 3 Choir Ahhs
  LOOP 8 times
    x = x + 1
    PLAY x for ½ a beat
  END LOOP
End Thread

```

```

Thread 4 Piano
  LOOP 8 times
    PLAY x for ½ a beat
  END LOOP
End Thread

```



Úloha 4 - Stupnica odzadu

Zmeňte program tak, aby obidva nástroje súčasne zahrli stupnicu odzadu.

```

Thread 3 Choir Ahhs
  LOOP 8 times
    x = x - 1
    PLAY x for ½ a beat
  END LOOP
End Thread

```

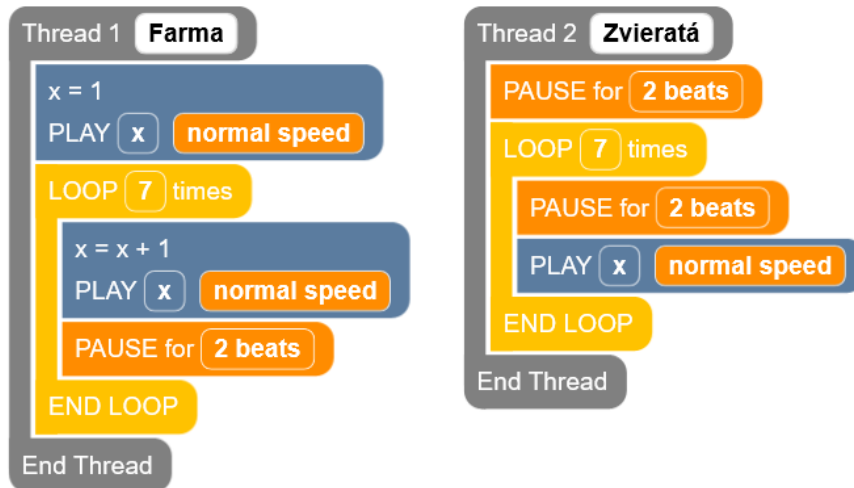
```

Thread 4 Piano
  LOOP 8 times
    PLAY x for ½ a beat
  END LOOP
End Thread

```


Úloha 5 – Na farme

Teraz v jednom vlákne nastavíme kategóriu zvukov **Farma** (Custom sounds / Farma) a v druhom kategóriu zvukov **Zvieratá** (Custom sounds / Zvieratá). Vytvoríte program, v ktorom sa striedavo v jednom vlákne povie názov zvieratka a v druhom vlákne sa zahrá zvuk tohto zvieratka (učiteľ si môže vopred pripraviť nahrávku hotového programu a žiakom ju prehrať).



Záver

Zhodnotíme prácu žiakov. Povieme, že sa naučili vytvárať programy, v ktorých sa používa premenná na zapamätanie nejakej hodnoty. Opýtame sa žiakov, prečo je dobré v programoch používať premenné.

Hodina 11 – Projekt

Nasledujúce úlohy slúžia na precvičenie vedomostí nadobudnutých počas predchádzajúcich hodín. Na vypracovanie každého projektu treba počítať s niekoľkými vyučovacími hodinami. Časť práce na projekte môžu žiaci spraviť na domácu úlohu. Na projekte môžu žiaci pracovať vo dvojiciach alebo menších skupinkách.

Úloha 1 – Obľúbená pesnička

Naprogramujte pesničku vytvorenú z častí uložených v samostatných zvukových súboroch. Zvukové súbory vytvorte najprv v programe na editovanie zvukov (napríklad Auda City).

Úloha 2 – Vlastný príbeh

Naprogramujte príbeh vytvorený z častí uložených v samostatných zvukových súboroch. Zvukové súbory najprv nahrajte a upravte v programe na editovanie zvukov (napríklad Auda City).

Úloha 3 – Naprogramuj básň

Naprogramujte básničku vytvorenú z častí uložených v samostatných zvukových súboroch. Zvukové súbory najprv nahrajte a upravte v programe na editovanie zvukov (napríklad Auda City).

Úloha 4 – Naprogramuj melódiu pre niekoľko hudobných nástrojov

Naprogramujte melódiu hranú rôznymi hudobnými nástrojmi tak, že použijete:

- príkazy Zahraj aspoň v dvoch vláknach,
- príkaz opakovania,
- príkaz vetvenia.

Príloha

Popis jazyka Code Jumper

Základný modul

Na základnom module sú dve tlačidlá – **Štart** a **Stop**. Po okraji sa na ňom nachádzajú štyri porty, do ktorých sa dajú pripájať príkazové moduly.

V strede je modré koliesko na regulovanie hlasitosti.

Tlačidlo Štart — Modré tlačidlo s trojuholníkom slúži na spustenie programu.

Tlačidlo Stop — Modré tlačidlo so štvorčekom slúži na zastavenie vykonávania programu.

Tlačidlá Štart a Stop stlačené súčasne — Nahlas sa prečíta program.



Príkazové moduly

Príkazové moduly predstavujú rôzne typy príkazov. V jednej sade sa nachádzajú štyri typy príkazových modulov, ktoré sú farebne odlišené – príkazový modul **Zahraj** (modrý), príkaz **Pauza** (pomarančový), príkaz **Opakuj** (žltý), príkaz **Podmienka** (zelený). Každý modul má tvar počítačovej myši a podľa jeho funkcionality je k nemu pripojený jeden alebo viac káblikov a obsahuje jeden alebo viac portov.

Príkazový modul Zahraj

Príkazový modul **Zahraj** má na vrchu dva modré otočné ovládače. Na jednom konci má káblík slúžiaci na pripojenie k inému modulu. Na opačnom konci má port, do ktorého sa dá pripojiť iný modul.



Otočný ovládač zvuku – Nižší a plochejší ovládač umožňuje nastaviť prehrávaný zvuk. Dostupnú ponuku zvukov môžeme nastaviť pomocou aplikácie Code Jumper (viď príloha).

Otočný ovládač trvania – Vyšší ovládač umožňuje nastaviť trvanie alebo rýchlosť zvuku. V prípade zvukov v kategórii zvukov **MIDI Instruments** tento ovládač mení ich trvanie na 1/4

doby, 1/2 doby, 1 dobu alebo 2 doby. Pre ostatné zvuky tento ovládač mení ich rýchlosť na 0,5-násobok rýchlosti, 1-násobok rýchlosti, 1,5-násobok rýchlosti a 2-násobok rýchlosti.

Ak je zvuk v kategórii zvuku **MIDI Instruments**, zmena trvania zvuku neovplyvní jeho výšku. Pri iných kategóriách zvuku však zmena rýchlosti predĺži alebo skráti trvanie zvuku.

Program – sekvencia príkazov

Sekvencia štyroch príkazov zapojených do základného modulu.



Príkazový modul Pauza

Príkazový modul **Pauza** umožňuje vytvoriť pauzu medzi dvomi zvukmi. Na module je otočný ovládač, ktorým môžeme nastaviť dĺžku trvania pauzy od 1/4 doby po 2 doby.



Príkazový modul Opakuj

Príkazový modul **Opakuj** umožňuje opakovať sekvenciu, ktorá je v ňom zapojená. Počet opakovaní môže byť 1 až 8 a nastavuje sa žltým otočným ovládačom.



Z modulu vychádzajú dva káble. Kratší kábel sa pripája ku základnému modulu alebo ku príkazom, ktorý nasleduje pred cyklom. Dlhší kábel sa pripája ku modulu, ktorý je na konci sekvencie príkazov nachádzajúcich sa v cykle.

Modul obsahuje dva porty. Do portu, ktorý je hneď vedľa kábla ukončujúceho sekvenciu príkazov v cykle, sa pripája sekvencia opakujúcich sa príkazov. Do druhého portu sa pripája príkazový modul, ktorý nasleduje po cykle.

Program – viac vláken

Program s dvomi vláknami, v jednom z nich je cyklus.

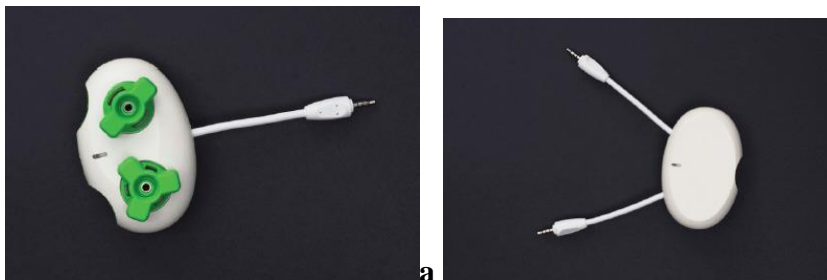


Príkazový modul Podmienka

Príkazový modul Podmienka je párový modul pozostávajúci z dvoch modulov – **rozvetvovací** a **zlučovací**.

Rozvetvovací modul má dva zelené ovládače – dvojramenný a trojramenný. Vychádza z neho jeden kábel, ktorým sa môže pripojiť k inému modulu nachádzajúcemu sa v programe pred týmto modulom. Na boku sú dva porty, ktoré umožňujú pripojiť ďalšie príkazové moduly.

Rozvetvovací modul umožňuje zvoliť jednu z dvoch rôznych postupností príkazov v závislosti od hodnôt nastavených na jeho ovládačoch. Každý ovládač má port umožňujúci zasunutie konektora s číselnou hodnotou od 1 do 8. Ak je číslo v dvojramennom ovládači väčšie ako číslo v trojramennom ovládači, vykoná sa postupnosť príkazov pripojených do portu pri dvojramennom ovládači. V opačnom prípade sa vykoná postupnosť pripojená do portu pri trojramennom ovládači.



Zlučovací modul má dva káble, ktorými sa pripojí ku postupnostiam príkazov vychádzajúcim z rozvetvovacieho modulu. Takto sa vlastne ukončí príkaz **Podmienka**. Na opačnom konci je na boku modulu port, do ktorého možno pripojiť príkazy nasledujúce po príkaze **Podmienka**.

Konštanty

Konštanty majú tvar zástrčky a obsahujú symboly čísel od 1 po 8. Používajú sa na nastavenie konkrétnej hodnoty zasunutím do:

- ovládača na nastavenie zvuku v príkazovom module Zahraj,

- ovládača na nastavenie počtu opakovaní v príkazovom module Opakuj,
- ovládača na nastavenie hodnoty v príkazovom module Podmienka
- konektora Podmienka.



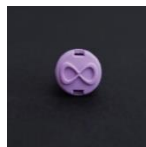
Náhodná hodnota

Konektor s písmenom R (random) sa používa na nastavenie náhodnej číselnej hodnoty od 1 do 8.



Nekonečno

Konektor so znakom nekonečno sa používa na nastavenie nekonečného počtu opakovaní v príkazovom module Opakuj. Príkazy v cykle sa budú opakovať dovtedy, kým sa na základnom module nestlačí tlačidlo Pauza.



Zvyšovanie a znižovanie hodnoty (inkrementácia a dekrementácia)

Konektory so znakom plus a mínus.



Poradie zvukov v každej kategórii zvukov je pevne dané, pričom každému zvuku je priradené číslo. Konektor **Plus** zvyšuje o 1 hodnotu, ktorú sme nastavili na ovládači zvuku príkazového modulu Zahraj, konektor **Mínus** túto hodnotu znižuje o 1. Takže na module Zahraj sa zahrá nasledujúci alebo predchádzajúci zvuk v sade zvukov.

Ak je konektor **Plus** alebo **Mínus** vložený do ovládača zvuku na module Zahraj, ktorý je zapojený v cykle, zmení sa zvuk na nasledujúci alebo predchádzajúci zvuk v sade zvukov pri každom opakovaní príkazov cyklu.

Premenná

Premenná má tvar konektora so zástrčkou, do ktorej je možné vložiť hodnotu a neskôr ju použiť v programe.



Všetky premenné, ktoré sa vyskytnú neskôr v programe, budú mať rovnakú hodnotu ako premenná s naposledy nastavenou hodnotou.

Do zástrčky **Premenná** je možné vložiť konektor **Plus** alebo **Mínus**, čo umožní zvýšiť alebo znížiť hodnotu uloženú v premennej pri každom opakovaní príkazov.

Do zástrčky **Premenná** je možné vložiť tiež konektor s písmenom **R**, čím sa do premennej vloží náhodne vybraná hodnota od 1 do 8.

Obsah jednej sady jazyka



Jedna sada obsahuje tieto časti:

- 1 základný modul (biely s modrými ovládacími prvkami)
- 8 príkazových modulov Zahraj (biele s modrými ovládačmi),
- 3 príkazové moduly Pauza (biele s pomarančový ovládačom),
- 2 príkazové moduly Opakuj (biele so žltým ovládačom),

- 1 párový príkazový modul Podmienka (biely rozvetvovací so zelenými ovládačmi, biely zlučovaci bez ovládačov)
- 12 číselných konštánt a funkcií, z toho 8 so symbolmi čísel, 1 so znakom nekonečno, 1 s písmenom R – náhodné (random), 1 so znakom plus, 1 so znakom mínus (fialové zástrčky)
- 3 premenné (sivé zástrčky so zásuvkou)
- 1 dlhý káblik (umožňuje zapojiť do cyklu alebo podmienky dlhšiu postupnosť príkazov).

Popis prostredia aplikácie Code Jumper

Aplikácia Code Jumper sa dá stiahnuť zdarma zo stránky Microsoft Store.

Časti prostredia

V hlavnej ponuke aplikácie Code Jumper sa nachádza šesť tlačidiel s ikonami.

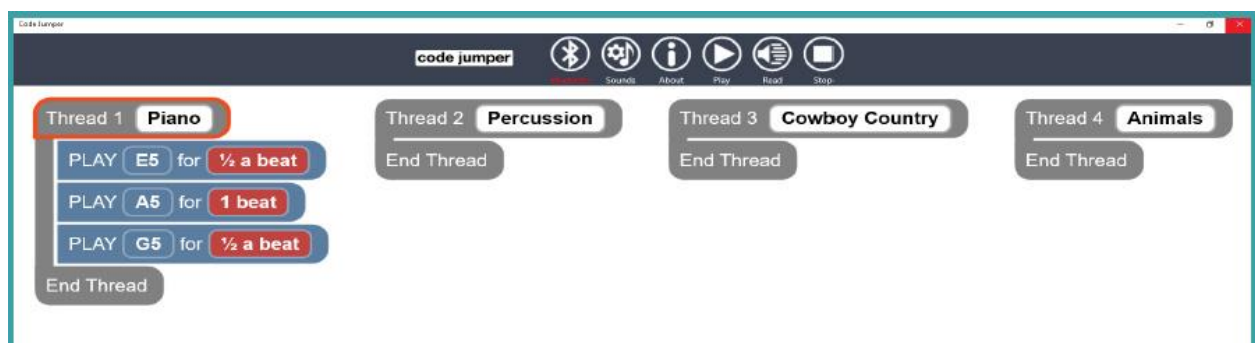


Zľava doprava sú to:

- **Bluetooth** – Symbol pre bluetooth (indikuje pripojenie základného modulu),
- **Sounds** – Symbol noty a ozubeného kolieska (umožňuje pridávanie vlastných zvukov),
- **About** – Malé písmeno i (zobrazí informácie o programe),
- **Play** – Symbol tlačidla Štart (spustí vytvorený program),
- **Read** – Symbol megafónu (prečíta príkazy programu),
- **Stop** – Symbol tlačidla Stop (zastaví program)

V okne pod ponukou sa nachádzajú štyri vlákna s príkazmi. Zodpovedajú sekvenciám modulov pripojených do jednotlivých konektorov základného modulu.

Na obrázku vidíme, že v prvom vlákne sú tri príkazy zodpovedajúce programu tvorenému tromi príkazovými modulmi Zahraj, na ktorých sú nastavené tóny E5, A5, a G5.



Používatelia, ktorí používajú čítač obrazovky, sa môžu pohybovať po programe pomocou klávesnice. Medzi vláknami sa dá presúvať klávesmi so šípkou doľava alebo doprava. Medzi

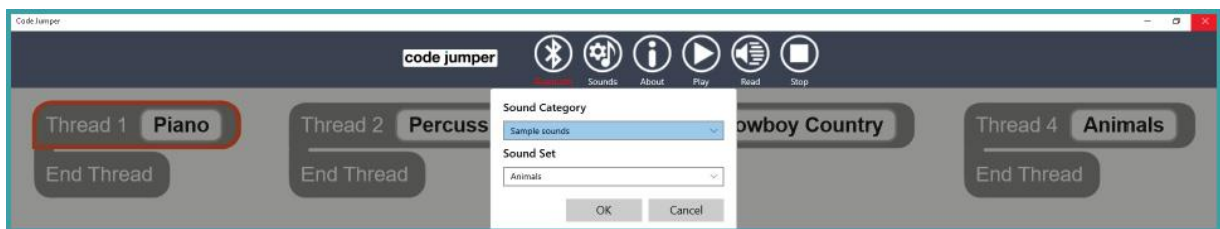
jednotlivými príkazmi vo vlákne sa dá presúvať pomocou šípok hore a dole alebo pomocou tabulátora.

Výber kategórie zvukov

Každé vlákno môže mať priradenú inú kategóriu zvukov. V jednej kategórii môže byť 8 tematicky súvisiacich zvukov a budú sa hrať v postupnosti príkazov, ktoré sú zapojené do príslušného portu na základnom module. Poradie zvuku v sade zvukov je pevné a je k nemu priradené číslo. Zvukové sady priradené ku vláknam je možné zmeniť v aplikácii. Názvy zvukových sád sú zobrazené vedľa čísla vlákna.

Keď klikneme na názov zvukovej sady, zobrazí sa dialógové okno, v ktorom môžeme zvoliť inú zvukovú sadu. Ak to chceme urobiť pomocou klávesnice, nastavíme sa na názov zvukovej sady šípkami, stlačíme tabulátor a potom kláves Enter.

Poznámka k nášmu klavíru a iným hudobným nástrojom: Niektoré z našich hudobných zvukových sád boli vytvorené pomocou nôt MIDI nástrojov. „Názvy nôt“ (C4, C5 atď.) zobrazené v aplikácii Code Jumper odrážajú noty MIDI nástroja. Takže zatiaľ čo na tradičnom klavíri poznáme stredné C ako C4, v jazyku MIDI je 60 nôt namiesto 88, takže stredné C bude C5.

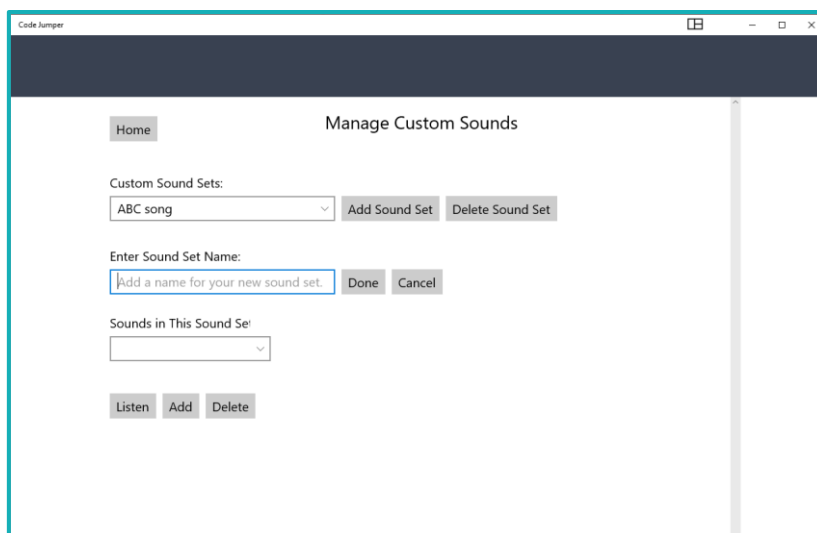


Vytváranie vlastnej kategórie zvukov

Ak chceme pridať svoje vlastné zvuky, najprv si vytvoríme vlastnú zvukovú sadu a potom do nej pridáme zvuky.

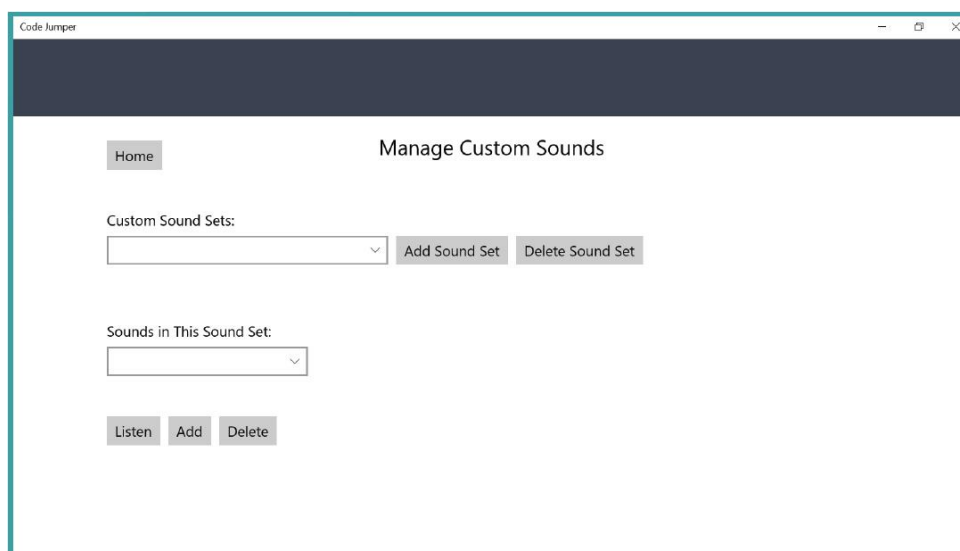
Vytvorenie vlastnej zvukovej sady

1. V hlavnej ponuke klikneme na symbol noty s ozubeným kolieskom (Sounds).
2. Klikneme na tlačidlo **Add Sound Set** nachádzajúce sa vedľa zoznamu **Custom Sound Sets**.
3. Do textového poľa **Enter Sound Set Name** zadáme názov našej zvukovej sady.
4. Klikneme na tlačidlo **Done**.



Pridanie vlastných zvukov

1. V zozname **Sounds In This Sound Set** vyberieme prvú voľnú položku a stlačíme tlačidlo **Add**.
2. Do textového poľa **Enter sound name** zadáme meno nášho zvuku.
3. Klikneme na tlačidlo **Add Sound File** a zvolíme .wav alebo .mp3 súbor.
4. Stlačíme tlačidlo **Done**. Automaticky sa v zozname s názvami zvukov presunieme na nasledujúcu položku.
5. Ak si chceme vypočuť, editovať alebo zmazať už pridaný zvuk, nastavíme sa naň v zozname zvukov **Sounds In This Sound Set** a stlačíme tlačidlo **Listen** alebo **Edit** alebo **Delete**.
6. Stlačíme tlačidlo **Home**, aby sme sa vrátili naspäť do hlavného okna aplikácie. Našu sadu zvukov nájdeme pod kategóriou **Custom Sounds** a môžeme ju použiť v našich programoch.



Sady zvukov k úlohám

Sady zvukov k úlohám si treba stiahnuť zo stránky

<http://vin.edu.fmph.uniba.sk/prostredia.html>

Po rozbalení súboru **zvuky.zip** budú jednotlivé sady zvukov v samostatných priečinkoch pomenovaných podľa názvov sád zvukov. V každom priečinku je aj textový súbor **info.txt** obsahujúci názov sady, názvy zvukov a mená príslušných zvukových súborov. Každú sadu zvukov treba do programu pridať podľa vyššie uvedeného návodu. Odporúčame dodržať poradie zvukov, v akom sú vymenované v súbore **info.txt** a použiť názvy zvukov uvedené v tomto súbore. Zabezpečíme tým prehľadnosť vytváraných programov, ktoré sa budú zobrazovať v aplikačnom programe na obrazovke počítača.

Bude potrebné pridať tieto sady zvukov:

- Príbeh (používa sa na hodinách 2, 3, 8)
- Zvieratá (používa sa na hodinách 3, 10)
- Farma (používa sa na hodinách 3, 10)
- ABC_song (používa sa na hodine 4)
- Kŕmenie1 (používa sa na hodine 4)
- Hodiny (používa sa na hodine 5)
- KotKot_ČinČará (používa sa na hodine 6)
- BimBam (používa sa na hodine 6)
- Výstava (používa sa na hodine 6)
- Rodina (používa sa na hodine 6)
- Čísla (používa sa na hodinách 8, 9)

Literatúra

1. Code Jumper User Guide, https://codejumper.com/downloads/htmls/userguide/en/CodeJumper_Guidebook.html
2. Code Jumper Lessons, <https://codejumper.com/lessons.php>