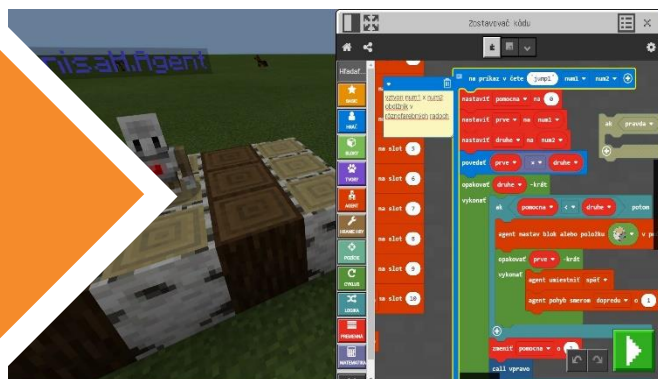


Metodický list č. 5

Cieľová skupina žiakov: 7. a 8. ročník ZŠ

Spracované podľa Inovovaného ŠVP pre 2.stupeň
ZŠ Matematika a práca s informáciami časť
Informatika- nižšie stredné vzdelávanie

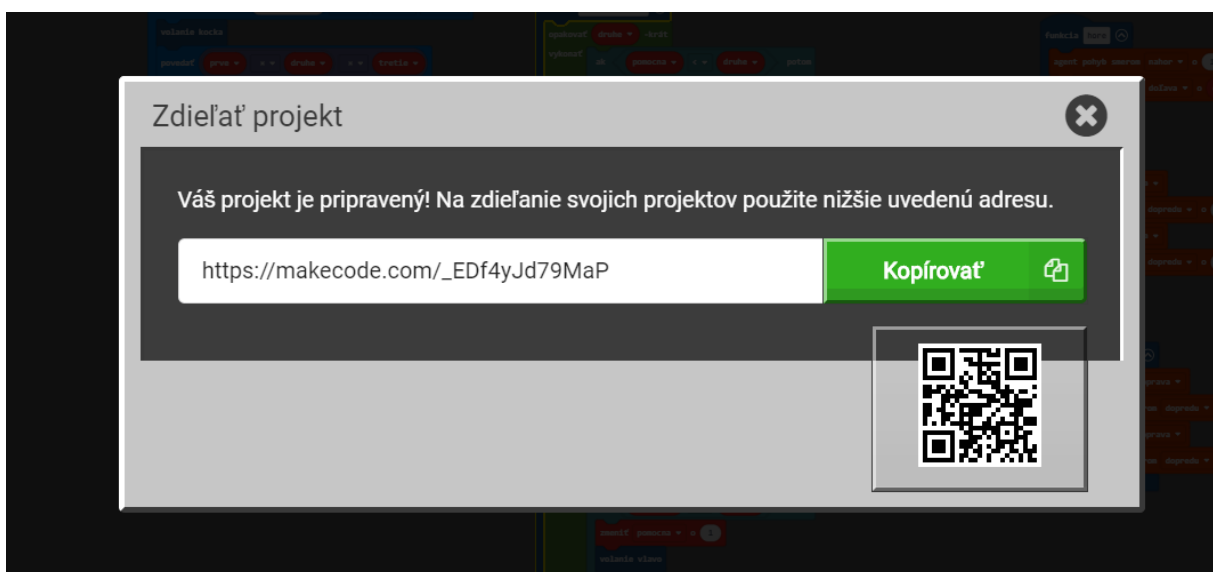


<p>Požiadavky na zručnosti žiakov:</p>	<ul style="list-style-type: none"> • Zostaviť jednoduchý program, vedieť interpretovať zapísané • Poznať pojem príkaz, cyklus, slot, vetvenie • Export programu, import programu • Poznať základné klávesové skratky E inventár, C programovanie, T chat
<p>Názov metodického listu:</p>	<h2>Tvorba pomôcky na MAT – hranol (objem a povrch)</h2>
<p>Učivo:</p>	<p>Algoritmické riešenie problémov</p> <ul style="list-style-type: none"> - pomocou vetvenia - pomocou premenných - pomocou nástrojov na interakciu - pomocou cyklov - analýza problému - jazyk na zápis riešenia
<p>Výkonový štandard (podľa iŠVP):</p>	<ul style="list-style-type: none"> ▪ rozpoznať situácie a podmienky, keď treba použiť vetvenie, ▪ rozpoznať, aká časť algoritmu sa má vykonať pred, v rámci a po skončení vetvenia, ▪ zostaviť a zapísať podmienku, ▪ vyriešiť problémy, ktoré vyžadujú vetvenie s jednoduchou podmienkou (bez logických spojok), ▪ zapísať riešenie problému s vetvením pomocou jazyka, ▪ interpretovať algoritmy s vetvením. ▪ identifikovať údaje zo zadania úlohy, ktoré musia byť zapamätané, resp. sa menia, a vyžadujú si použitie premenných, ▪ aplikovať pravidlá, konštrukcie jazyka pre nastavenie a použitie premennej, ▪ vyriešia problémy, v ktorých si treba zapamätať a neskôr použiť zapamätané hodnoty, ▪ zovšeobecniť riešenie tak, aby fungovalo nielen s konštantami, interpretovať algoritmy s výrazmi a premennými. ▪ zapísať algoritmus, ktorý reaguje na vstup, interpretovať zapísané riešenie, vytvoriť hypotézu, ako neznámy algoritmus spracuje zadaný vstup
<p>Obsahový štandard(podľa iŠVP):</p>	<p>Vetvenie, podmienka <i>Vlastnosti a vzťahy:</i> konštrukcia vetvenia s jednoduchou podmienkou, pravda, nepravda – splnená a nesplnená podmienka <i>Procesy:</i> zostavovanie, upravovanie vetvenia, vytvorenie podmienky, vyhodnotenie podmienky</p> <p>Premenná <i>Pojmy:</i> premenná, meno (pomenovanie) premennej, hodnota premennej, operácia (+,-,*,/) <i>Vlastnosti a vzťahy:</i> pravidlá jazyka pre použitie premennej, meno premennej – hodnota premennej <i>Procesy:</i></p>

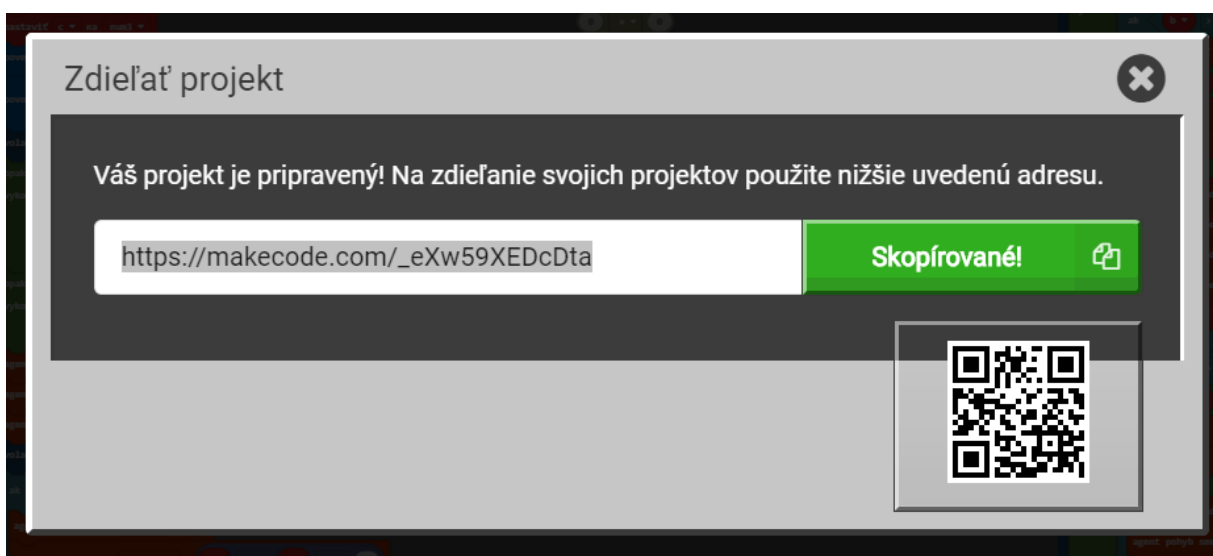
	<p>nastavenie hodnoty (priradenie), zistenie hodnoty (použitie premennej), zmena hodnoty premennej, vyhodnocovanie výrazu s premennými, číslami a operáciami</p> <p>Pomocou nástrojov na interakciu</p> <p><i>Vlastnosti a vzťahy:</i> prostriedky jazyka pre: získanie vstupu, spracovanie vstupu a zobrazenie výstupu</p> <p><i>Procesy:</i> čakanie na neznámy vstup – vykonanie akcie – výstup, následný efekt</p>
Ciele:	Utvrdiť a upevniť vedomosti o cykloch a vetvení. Ukázať využiteľnosť programu v bežnom živote. Vytvoriť učebnú pomôcku.

Programy potrebné pre tento metodický list:

https://makecode.com/_EDf4yJd79MaP objem hranola



https://makecode.com/_eXw59XEDcDta povrch



(Pripravený program je vždy iba ukázkový, necháme žiakom možnosť samostatnej práce, možno navrhnu efektívnejší program.

Pojmy:

Podprogram (v MEE sa nazýva funkcia)- relatívne samostatný čiastočný algoritmus (čiže časť programu, ktorý má vlastnosti malého programu a hlavný program ho môže volať). Spravidla ide o postup, ktorý bude v programe opakovaný viackrát, a to na rôznych miestach príkazovej časti programu. Ale sa používa na sprehľadnenie programu. Novú funkciu/ podprogram tvoríme v časti Pokročilá a do Vytvoriť funkciu... zadávame názov našej novej funkcie.

Vetvenie – sa vytvára pomocou podmienky v MEE sú na to určené dva bloky Ak pravda potom... a Ak pravda potom..., inak ...

Je tu možnosť rozhodnúť sa podľa pravdivosti/ nepravdivosti skúmaného znaku. V závislosti od splnenia/ nesplnenia podmienky sa postup vetví na rôzne prípady.

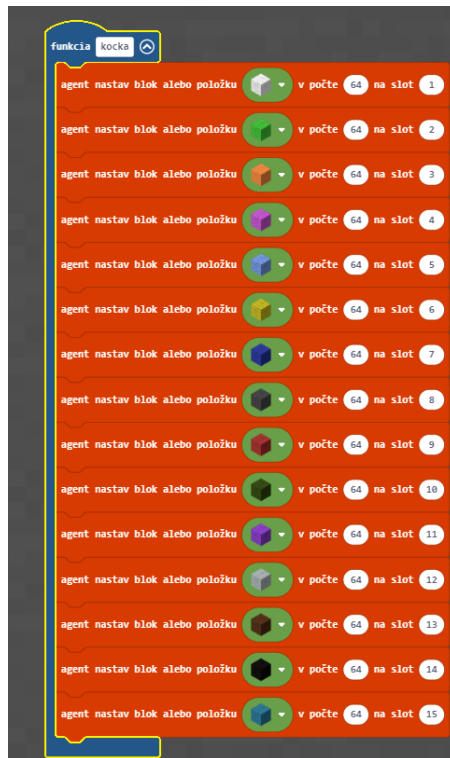
Premenná - objekt (môžeme ju považovať za nejakú pamäť, alebo miesto v pamäti) slúžiace počas behu programu na odkladanie údajov. Jej hodnota sa počas činnosti algoritmu môže meniť (a zvyčajne sa aj mení).

Motivácia:

Zameriame sa na výrobu pomôcky pre mladších žiakov na hodiny matematiky. Spomenieme si aké to bolo ťažké učiť sa násobilku. Neskôr prišlo násobenie troch čísel. Zároveň si ukážeme ako vypočítať objem hranola a navrhnuť výpočet jeho povrchu. V časti od Kroku 5 si ako motto vyberieme výrok Ľudovíta Štúra: “Nazad cesta nemožná, napred sa ísť musí!” My ho vezmeme doslovne, v našom programe nebudeme cúvať, pôjdeme iba dopredu a uvidíme ako sa nám bude dariť.

Krok 1

Chceme stavať mnohofarebné objekty, aby bolo dobre vidieť jednotlivé bloky, ktoré používame. Je to preto, aby žiaci videli farebne rozlíšené jednotlivé bloky a vedeli ich porátať. V prípade rovnakej farby blokov, by sa Hranice blokov zliali do jednej homogénnej masy a neboli by vidieť. Na takýto postup je potrebné vytvoriť si podprogram, ktorý nám zabezpečí naplnenie slotov blokmi rôznej farby. My sme si ho nazvali “kocka”. Každému bloku sme priradili iný slot a naplnili sme ho nekonečným počtom blokov, čo je vyjadrené číslom 64.



Obrázok 1 Podprogram "kocka"

Krok 2:

Začneme stavať pomôcku s názvom násobilka, kde by si aj žiaci prvého stupňa mohli vyskúšať zadať čísla a náš agent im vyráta koľkokoblokov bolo treba použiť a postaví im obdĺžnik s danou veľkosťou.

Podobné sme riešili v predchádzajúcich metodických listoch.

Začneme stavbou štvorca, ktorému chceme zadať rozmer. Práve rozmer sa stane premennou pod menom "num1". Do četu zadávame meno programu, medzeru a číslo. Číslo sa automaticky zadá do premennej.



Obrázok 2 Program "stvorec"

Krok 3:

Tu sme rozšírili program štvorec o možnosť zadania druhého parametra num2

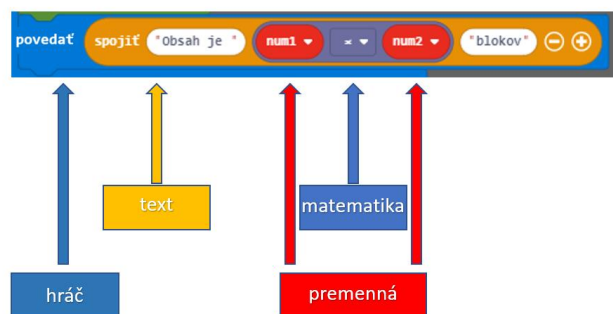
Jeho zadané hodnoty v čete sa nám načítajú do premennej num2. túto premennú sme umiestnili do vonkajšieho cyklu. Na obrázku vidíte porovnanie dvoch zadaní 3 x 5 a 5x 3



Obrázok 3 Program "obdĺžnik"

Krok 4:

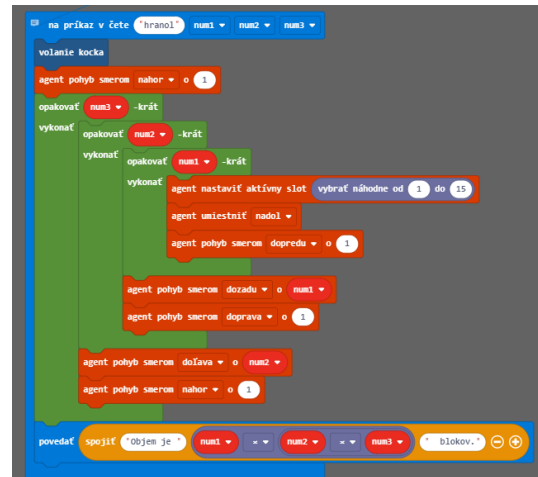
Je ešte potrebné, aby náš agent vedel vypočítať obsah takéhoto útvaru a zároveň ho zobrazí hráčovi na obrazovke. Na obrázku č.4 vidíme ako je príkaz poskladaný. My sme ho vložili až za telo cyklu, takže agent najskôr postaví obdĺžnik a potom vypíše koľko blokov potrebuje. Príkaz možno vložiť aj na začiatok, je to na uvážení programátora.



Obrázok 4 Kde hľadať jednotlivé časti príkazu

Krok 5:

Keď sa pozrieme čo reálne náš agent stavia, tak zistíme, že stavia hranol s rozmermi „num1“ a „num2“ s výškou 1 blok. Čo ak by sme chceli, aby hranol mal inú výšku? Čo v programe musíme zmeniť, tak aby sme vystavali hranol?



Obrázok 5 Stavba hranola

V týchto piatich krokoch sme si ukázali ako pomocou cyklov vytvoríť jednoduchú pomôcku pre matematiku.

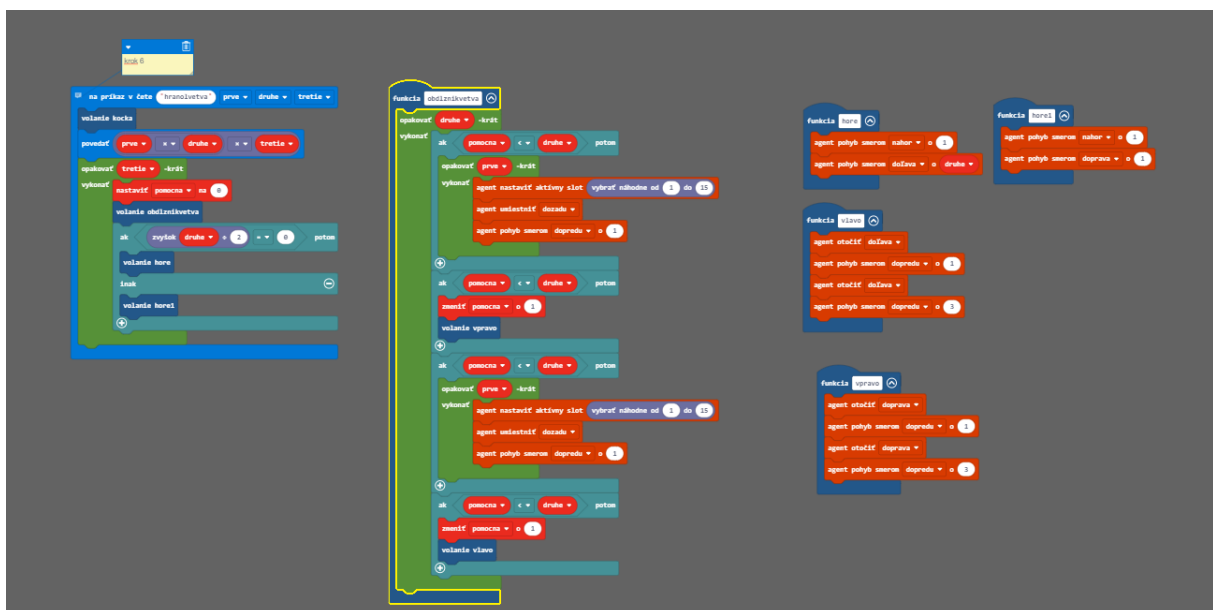
Krok 6:

VETVENIE a FUNKCIE:

Ak učiteľ chce zopakovať aj vetvenie, je možné použiť druhú sadu programov. V zadaní sa bude líšiť iba v tom, že agent bude mať zakázané ísť dozadu. Agent musí ísť iba smerom dopredu. Vtedy bude nutné používať vetvenie.

V tejto sade programov sme pre skrátenie zápisu používali aj podprogramy/ funkcie.

Vetvenie je potrebné pri rozlišovaní párných a nepárných čísel. Program, ktorý postupne vzniká je určovaný počtom cyklov

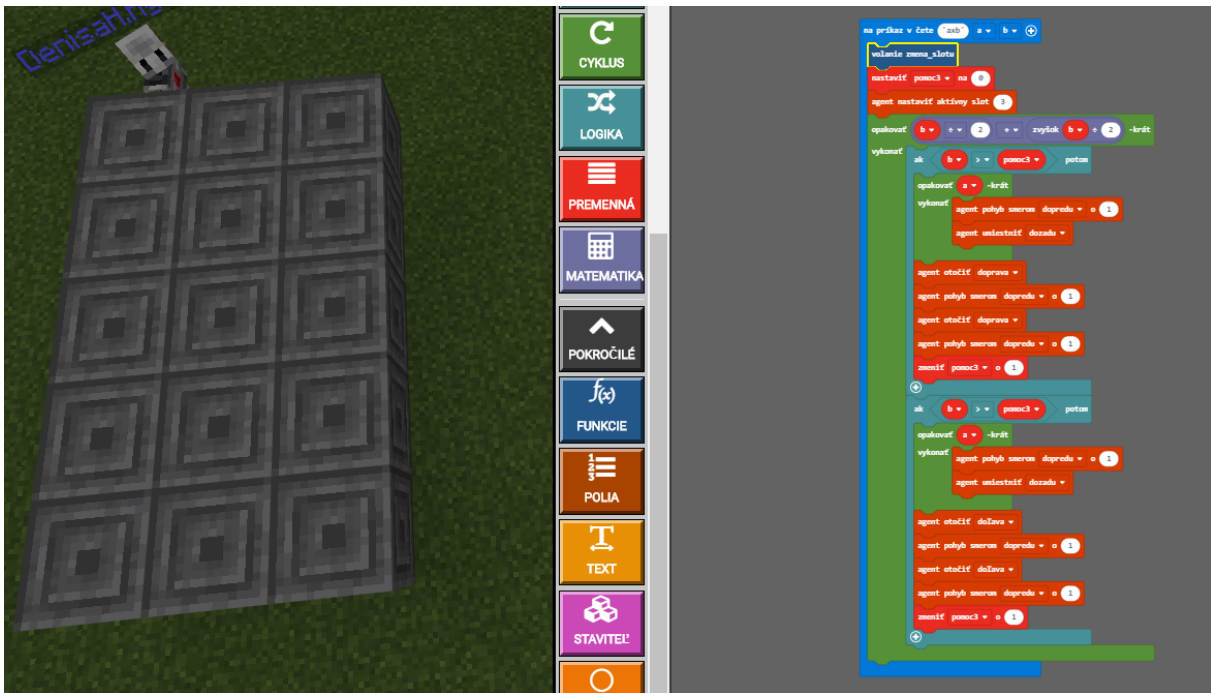


Obrázok 6 Hranol s vetvením

Krok 7:

V tomto kroku budeme potrebovať program na povrch. Agent nesmie ísť dozadu!

V tomto kroku si začneme budovať časti plášťa. V cykle máme vložené vetvenie s dvoma podmienkami, ktorými sme vyriešili, či agent stavia párny, alebo nepárny rad blokov. V počte opakovaní teda nemôže byť iba premenná, ale musí tam byť výpočet koľkokrát sa tieto dva rady zopakujú. Z tohto programu sa stane funkcia `a_x_b`



Rovnaký princíp sme uplatnili aj na Ďalšie dve funkcie `a_x_c` a `b_x_c`.

Pri skladaní do jedného veľkého programu, berieme do úvahy, ako vypadá plášť hranola.



Obrázok 7 Celý plášť

Aby sme rozoznali jednotlivé časti vytvárame ich z blokov, ktoré sú pekne vizuálne ohraničené, aby ich študenti vedeli ľahko zrátať a zároveň sú od seba odlišené jednotlivé steny.

Krok 8:

V inventári si vyberieme vajce z ktorého sa nám narodí nehrajúca postava (ďalej NPC) a jednu vyliahneme.



Obrázok 8 Vajce NPC

Aby sme mohli NPC vyliahnuť je potrebné prepnúť si svoj stav – zo staviteľa sveta na volajúceho. Urobíme tak v čete, kde vložíme `/wb`. Po umiestnení NPC sa opäť rovnakým spôsobom prepne do stavu staviteľa.



Obrázok 9 Prepnutie

Teraz môžeme postavu upraviť. Na obrázku 10 vidíme ako vypadá dialógové okno po kliknutí na NPC. Môžeme upraviť jej meno, vzhľad a napísať postave vlastný príhovor. V rozšírených nastaveniach vieme vložiť aj prepojenie na web stránku – „info o kvádri“



Obrázok 10 NPC- vytvorenie



Obrázok 11 NPC -Sprievodca