Metodický list č. 3

Cieľová skupina žiakov: 7. a 8. ročník ZŠ

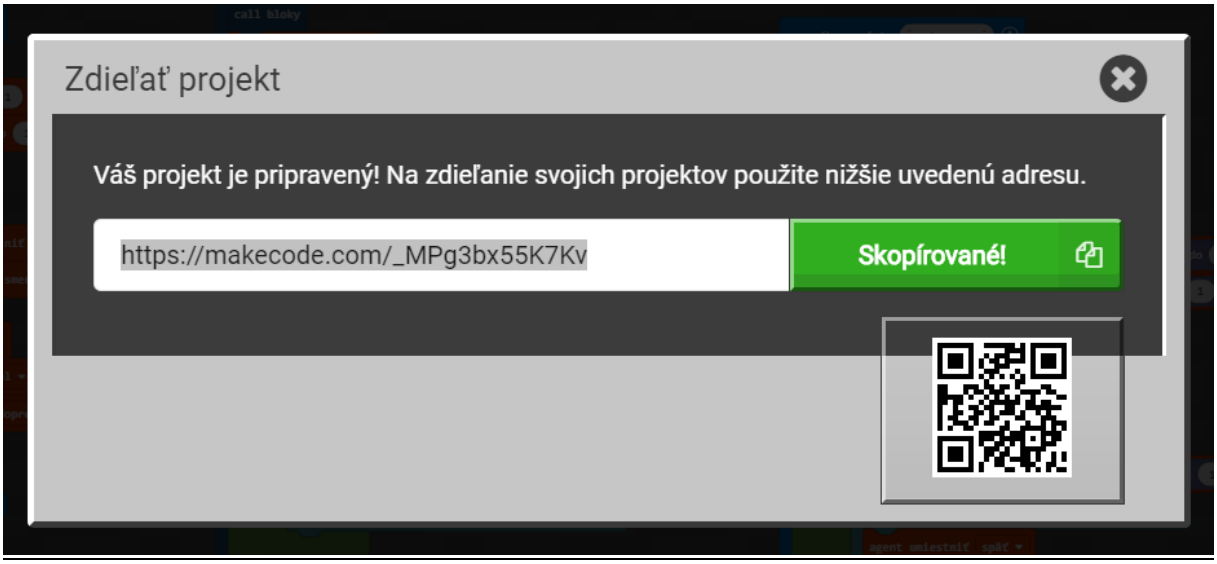
Spracované podľa Inovovaného ŠVP pre 2.stupeň ZŠ Matematika a práca s informáciami časť Informatika- nižšie stredné vzdelávanie

Požiadavky na zručnosti žiakov:	• Zostaviť jednoduchý program, vedieť interpretovať zapísané • Poznať pojem príkaz, cyklus, slot • Export programu, import programu • Poznať základné klávesové skratky E inventár, C programovanie, T chat
Názov metodického listu:	**Vetvenie**
Učivo:	**Algoritmické riešenie problémov** **- pomocou vetvenia** **- pomocou premenných** **- pomocou nástrojov na interakciu** - pomocou cyklov - analýza problému - jazyk na zápis riešenia
Výkonový štandard (podľa iŠVP):	▪ 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
Obsahový štandard(podľa iŠVP):	**Vetvenie, podmienka** *Vlastnosti a vzťahy*: konštrukcia vetvenia s jednoduchou podmienkou, pravda, nepravda – splnená a nesplnená podmienka *Procesy*: zostavovanie, upravovanie vetvenia, vytvorenie podmienky, vyhodnotenie podmienky **Premenná** *Pojmy:* premenná, meno (pomenovanie) premennej, hodnota premennej, operácia (+,–,*,/) *Vlastnosti a vzťahy*: pravidlá jazyka pre použitie premennej, meno premennej – hodnota premennej *Procesy:* nastavenie hodnoty (priradenie), zistenie hodnoty (použitie premennej),

	zmena hodnoty premennej, vyhodnocovanie výrazu s premennými, číslami a operáciami **Pomocou nástrojov na interakciu** V*lastnosti a vzťahy:* prostriedky jazyka pre: získanie vstupu, spracovanie vstupu a zobrazenie výstupu *Procesy:* čakanie na neznámy vstup – vykonanie akcie – výstup, následný efekt
Ciele:	Zaviesť a ukázať prácu s podprogramom, vetvením a premennou

Programy potrebné pre tento metodický list:

https://makecode.com/_MPg3bx55K7Kv



(Pripravený program je vždy iba ukážkový, necháme žiakom možnosť samostatnej práce, možno navrhnú efektívnejší program)

Motivácia:

Hra: Kto je lepší programátor?

Vytvoríme program, ktorý nám dovolí testovať akí sme programátori.

Prvá časť nám vytvorí náhodnú arénu s rôznymi blokmi.

Následne vytvoríme program, ktorý tie bloky vo vnútri arény bude ničiť a zároveň rátať – vyexportujeme si ho, uložíme a vymeníme si miesta pri počítačoch. Svoj program vložíme do spolužiakovho sveta a vyskúšame ako efektívne a s akou chybovosťou náš program vyčistí celú spolužiakovu arénu.

Môžeme zaznamenávať skóre úspešnosti každého žiaka.

Pojmy:

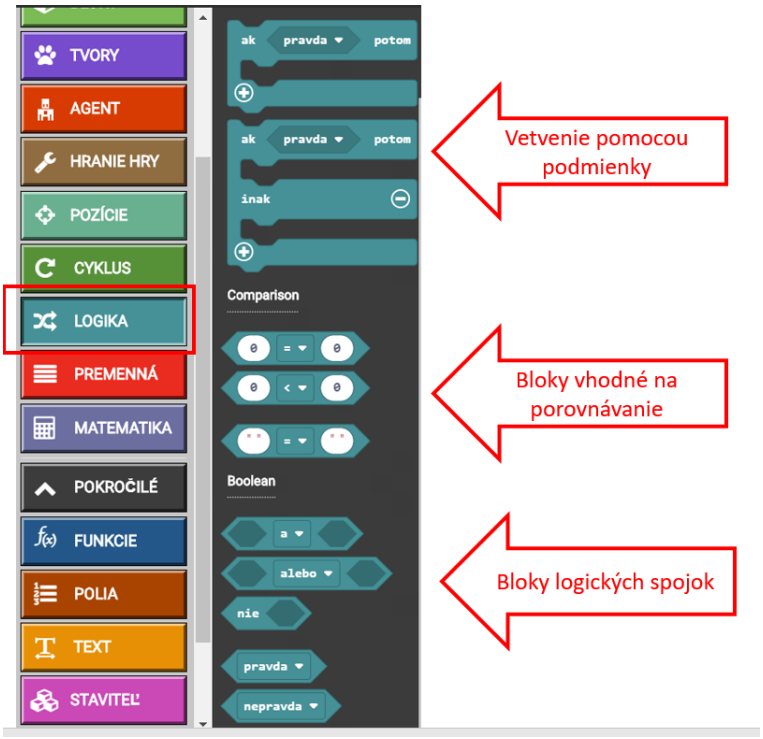
Podprogram (v MEE sa nazýva funkcia)- relatívne samostatný čiastočný algoritmus (čiže časť programu, ktorý má vlastnosti malého programu a hlavný program ho môže volať). Spravidla ide o postup, ktorý bude v programe opakovaný viackrát, a to na rôznych miestach príkazovej časti programu. Ale sa používa na sprehľadnenie programu.

Obrázok 1 Vytvorenie nového podprogramu

Novú funkciu/ podprogram tvoríme v časti Pokročilé a do Vytvoriť funkciu... zadávame názov našej novej funkcie.

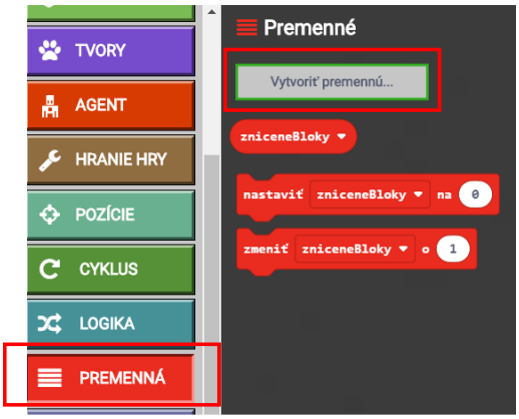
Vetvenie – sa vytvára pomocou podmienky v MEE sú na to určené dva bloky Ak pravda potom... a Ak pravda potom..., inak ...

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



Obrázok 2 Vetvenie

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).

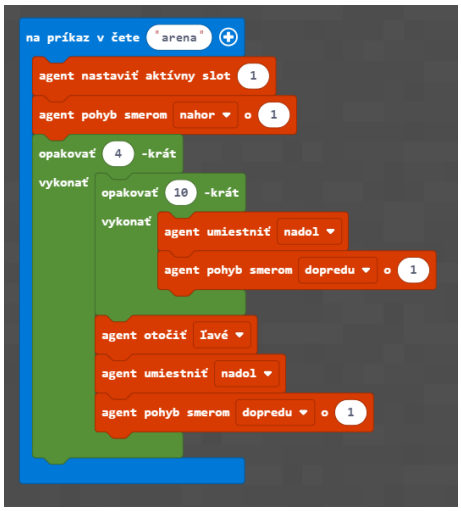


Obrázok 3 Vytvorenie premennej

Krok1:

Tvorba arény:

Zopakujeme si pojem cyklus a slot a po vzájomnom dohovore medzi žiakmi a učiteľom navrhneme veľkosť arény v ktorej sa budú umiestňovať jednotlivé bloky.

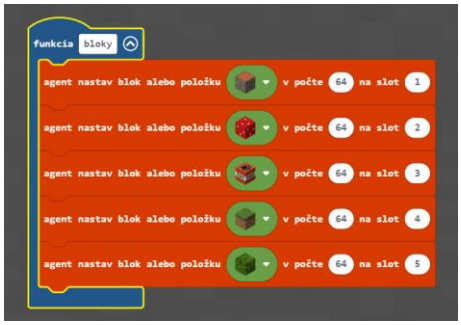


Obrázok 4 Aréna

Krok 2:

Do arény chceme umiestniť množstvo rôznych blokov, ktoré tam náhodne poukladá náš agent, aby sme sa tak vyhli nejakému vzoru ukladania blokov, čo by mohol využiť náš protivník a mohol by nad nami zvíťaziť. Počet uložených blokov si dohodneme

Potrebujeme meniť bloky náhodne a náhodne ich aj vyberať. Pre takúto príležitosť je dobré vytvoriť si pomocný program, ktorý sa volá funkcia, túto si nazveme “bloky”, a v tomto podprograme priradíme jednotlivým slotom ich bloky.



Obrázok 5 Funkcia "bloky"

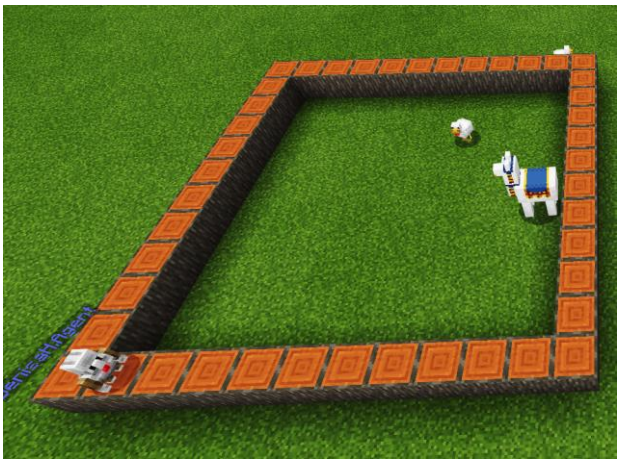
Ak by sme chceli, môžeme všetkým agentovým slotom priradiť nejaký blok a vtedy by mal tento podprogram až 18 riadkov. V celkovom programe by bol príliš obsažný a preto je dobré ho vyňať ako samostatny podprogram. Ak podprogram vložíme do programu hovoríme, že *voláme funkciu “bloky”*.

Krok 3:

Aj z programu “arena”, ktorý už máme odskúšaný, vieme vytvoriť funkciu s menom “arena”. Nachádza sa medzi programami.

Krok 4:

Ideme riešiť akým spôsobombude náš agent ukladať bloky vo vnútri arény.

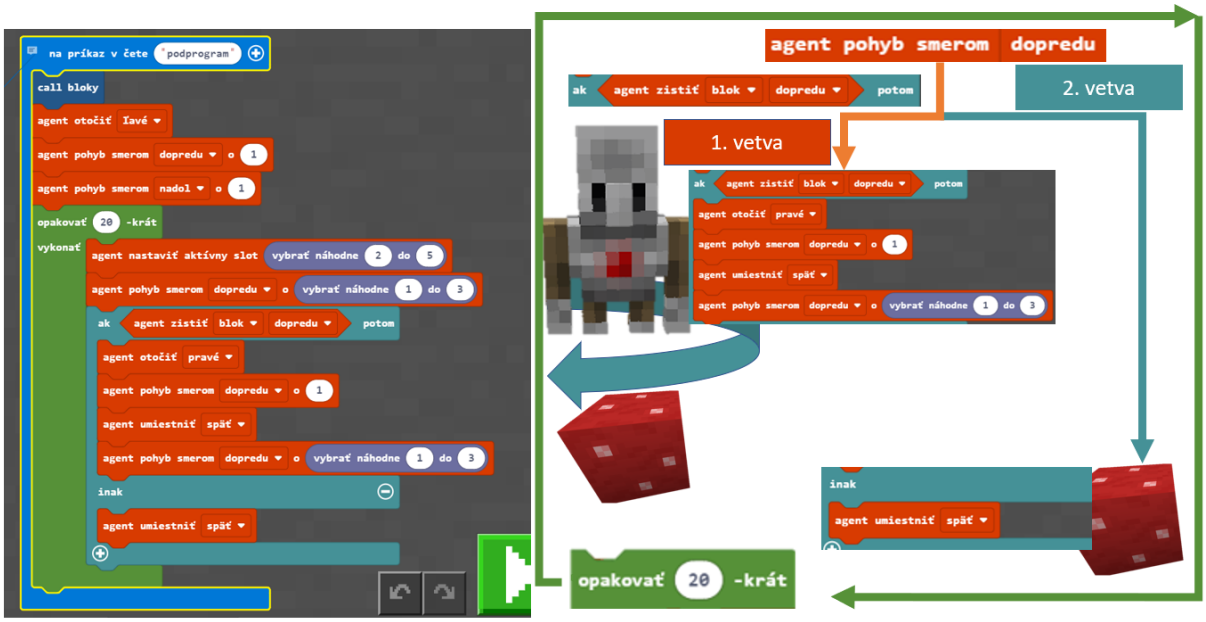


Obrázok 6 Postavená aréna

Po dostavaní arény zostane vpolohe ako je ukázané na Obrázku 6. Potrebujeme, aby vošiel dnu, ukladal bloky. Všetko má riadiť náhoda, takže ak narazí pri pohybe na nejaký blok, je žiadúce, aby sa nezasekol, ale aby sa otočil a išiel do inej strany.

Dôležité je tu slovo AK pred sebou zistí blok POTOM sa otočí, INAK (ak pred ním blok nie je) môže pokračovať v tom čo robí.

Tu zavádzame pojem vetvenie



Obrázok 7 Vetvenie

Takýmto spôsobom uloží agent do vnútorného priestoru ohrady maximálne 20 blokov z rôznych materiálov. V rôznych vzdialenostiach od seba.

Aréna je pripravená na zbieranie blokov.

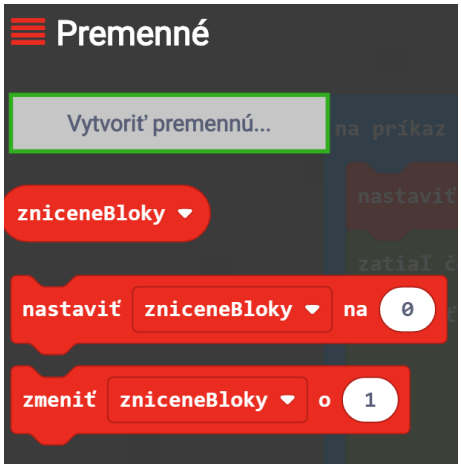
Krok 5:

Potrebujeme si vytvoriť program, ktorý bude ničiť bloky a rátať ich počet.

Je tu potrebné uviesť, že aréna, ohrada má zostať vcelku a neporušená. To musíme zabezpečiť v programe.

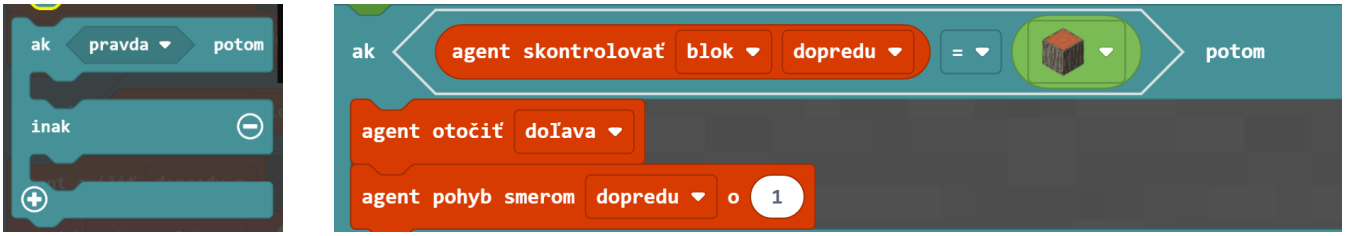
Program „zberaj“, ktorý uvádzame v tomto metodickom liste, je iba jednou z viacerých možností, ktoré môžu vaši žiaci vymyslieť. Je tu uvedený, ako príklad a ukážka.

Na rátanie zničených blokov si vytvoríme premennú, kde si budeme ukladať počet zničených blokov



Obrázok 8 Premenná "zniceneBloky"

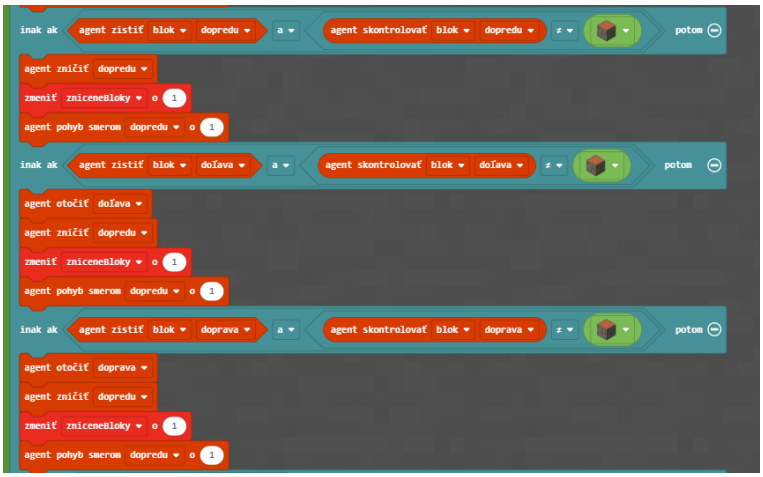
Najskôr sa musíme postarať, aby agent nezničil aj arénu:



Obrázok 9 Tvorba vetvenia podmienok

Budeme používať vetvenie. Do položky „pravda“ vložíme vyskladaný blok. Začneme blokom z „Logiky“ v časti „Porovnanie“ vyberieme „0 = 0“. Namiesto prvej nuly vložíme z položky „Agent“ oválny blok „agent skontrolovať blok dopredu“ a namiesto druhej nuly, vložíme blok materiálu, z ktorého sme postavili ohradu. To znamená, že ak agent narazí na blok z ohrady, otočí sa doľava a pohne sa smerom dopredu. Ohradu nezničí.

Ďalšou časťou je ničenie blokov a ich počítanie. To je zabezpečené vo vetvení programu:



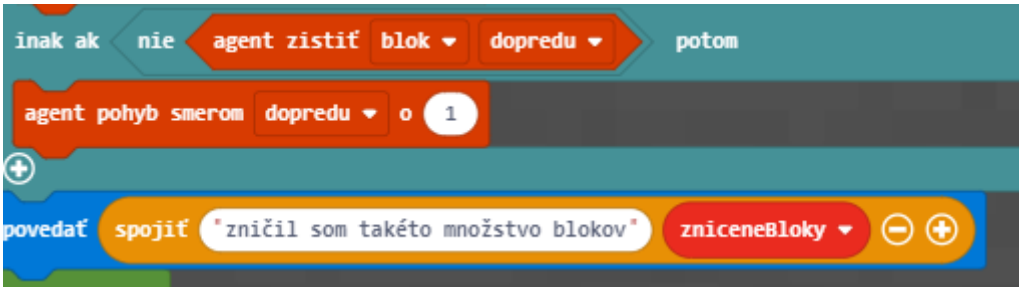
Obrázok 10 Ničenie blokov

Vo všetkých troch vetvách skúmame, či agent zaznamenal blok vo svojom okolí.

Každá vetva je zostavená z príkazov v „Logických premenných“ so spojkou „a“. a z už vyššie vytvoreného bloku. Ten sme zmenili z rovnosti na nerovnosť.

To znamená, že agent bude kontrolovať, či má blok pred sebou a ak ho zistí, bude kontrolovať, či tento blok nie je súčasťou ohrady. Ak nie je, zničí ho a zvýši premennú o jedna a pohne sa o jeden krok daným smerom, kde opäť vykoná kontrolu.

Posledná vetva programu rieši situáciu, ak v okolí agenta sa nenachádza žiadny blok.



Obrázok 11 V okolí nie je blok

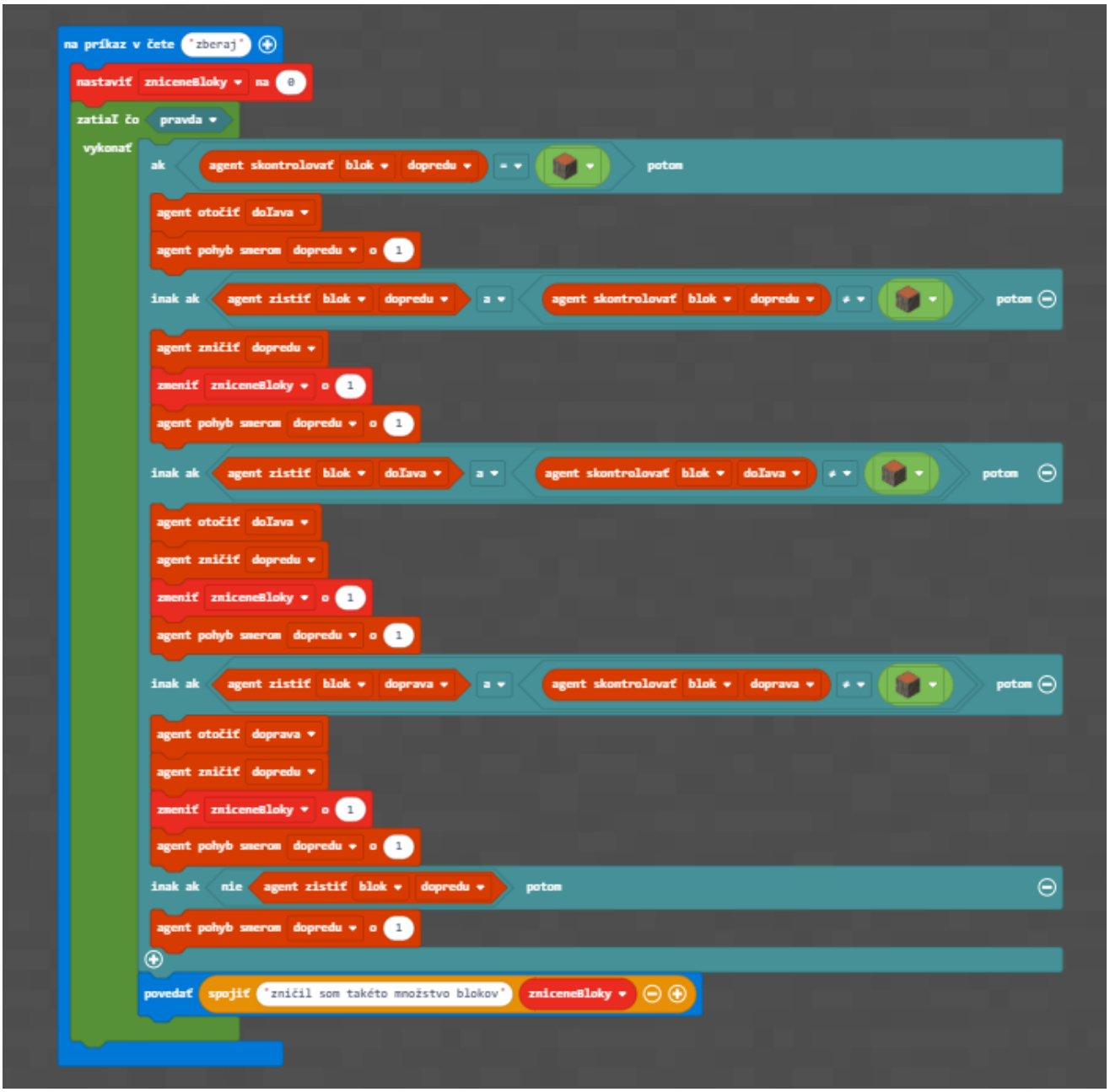
Ak agent nenachádza vo svojom okolí blok, iba sa pohne o jedno dopredu. Príkaz sme vytvorili spojením príkazu negácie z „Logiky“ a „agent zistiť blok dopredu“.

Po ukončení vetvenia je vložený príkaz „povedať“. Tento príkaz zabezpečí vypísanie textu na obrazovke. Vyberieme blok „spojiť“, do prvej časti napíšeme svoj text a do druhej časti vložíme výstup z premenne „zniceneBloky“

Obrázok 12 Rozšírenia o Text

Takto zabezpečíme informáciu pre študenta o tom, koľko blokov sa podarilo jeho agentovi zničiť.

Je potrebné zabezpečiť, aby sa program neustále opakoval, aby agent hľadal bloky neustále. Toto nám zabezpečí cyklus. Vytvorili sme nekonečný cyklus s vložením bloku „pravda“.



Obrázok 13 Celý program

Tento program je vhodný na precvičenie vetvenia, ale nezabezpečí, že agent nájde a pozbiera všetky bloky v aréne.

Pre šikovných:

Skúste žiakom zadať cvičenie na vytvorenie takého programu, ktorý určite nájde a pozbiera všetky bloky v aréne.