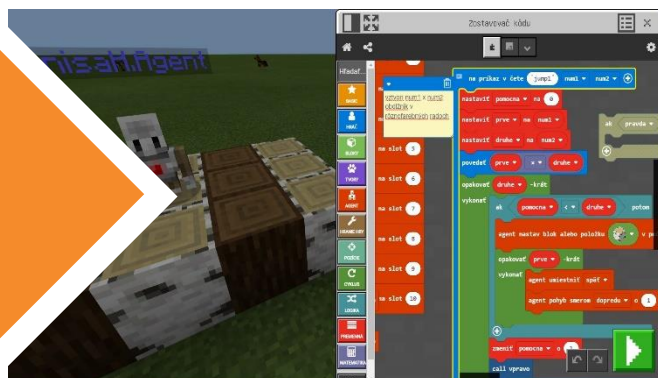


## Metodický list č. 2

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

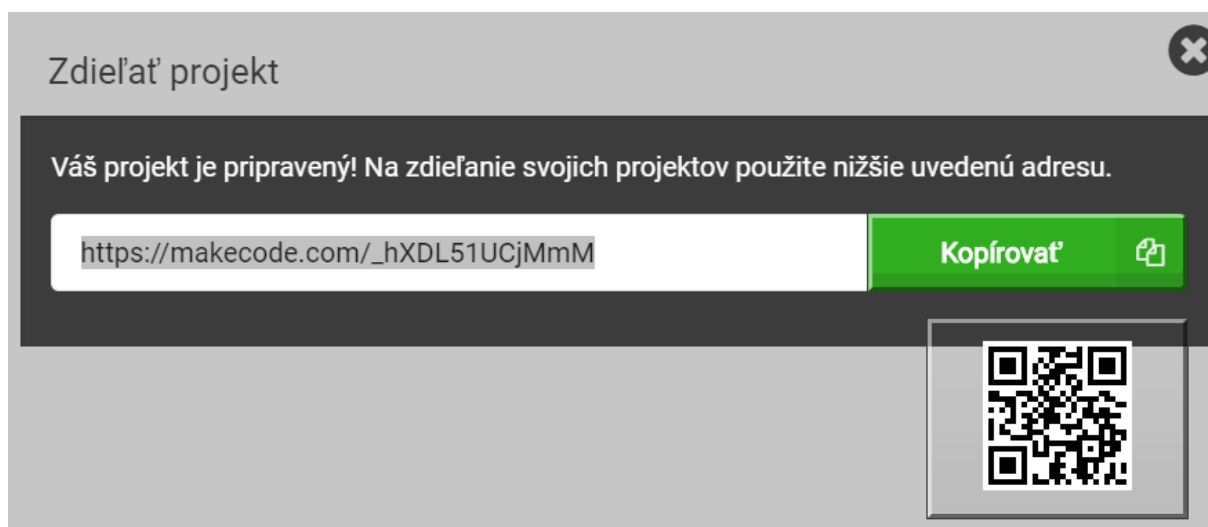


<p>Požiadavky na zručnosti žiakov:</p>	<ul style="list-style-type: none"> <li>• Zostaviť jednoduchý program, vedieť interpretovať zapísané</li> <li>• Poznať pojem príkaz, cyklus, slot</li> <li>• Export programu, import programu</li> <li>• Poznať základné klávesové skratky E inventár, C programovanie, T chat</li> </ul>
<p>Názov metodického listu:</p>	<h3>Vypracuj svoje riešenie – analýza programu</h3>
<p>Učivo:</p>	<p><b>Algoritmické riešenie problémov</b></p> <ul style="list-style-type: none"> <li>- pomocou vetvenia</li> <li>- pomocou premenných</li> <li>- pomocou cyklov</li> <li>- analýza problému</li> <li>- jazyk na zápis riešenia</li> </ul>
<p>Výkonový štandard (podľa iŠVP):</p>	<ul style="list-style-type: none"> <li>▪ rozpoznať situácie a podmienky, keď treba použiť vetvenie,</li> <li>▪ rozpoznať, aká časť algoritmu sa má vykonať pred, v rámci a po skončení vetvenia,</li> <li>▪ zostaviť a zapísať podmienku,</li> <li>▪ vyriešiť problémy, ktoré vyžadujú vetvenie s jednoduchou podmienkou (bez logických spojok),</li> <li>▪ zapísať riešenie problému s vetvením pomocou jazyka,</li> <li>▪ interpretovať algoritmy s vetvením.</li> <li>▪ identifikovať údaje zo zadania úlohy, ktoré musia byť zapamätané, resp. sa menia, a vyžadujú si použitie premenných,</li> <li>▪ aplikovať pravidlá, konštrukcie jazyka pre nastavenie a použitie premennej,</li> <li>▪ vyriešia problémy, v ktorých si treba zapamätať a neskôr použiť zapamätané hodnoty,</li> <li>▪ zovšeobecniť riešenie tak, aby fungovalo nielen s konštantami, interpretovať algoritmy s výrazmi a premennými.</li> </ul>
<p>Obsahový štandard(podľa iŠVP):</p>	<p><b>Vetvenie, podmienka</b> <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><b>Premenná</b> <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> nastavenie hodnoty (priradenie), zistenie hodnoty (použitie premennej),</p>

	zmena hodnoty premennej, vyhodnocovanie výrazu s premennými, číslami a operáciami
Ciele:	Zaviest' a ukázať prácu s vetvením a premennou

**Programy potrebné pre tento metodický list:**

[https://makecode.com/\\_hXDL51UCjMmM](https://makecode.com/_hXDL51UCjMmM)

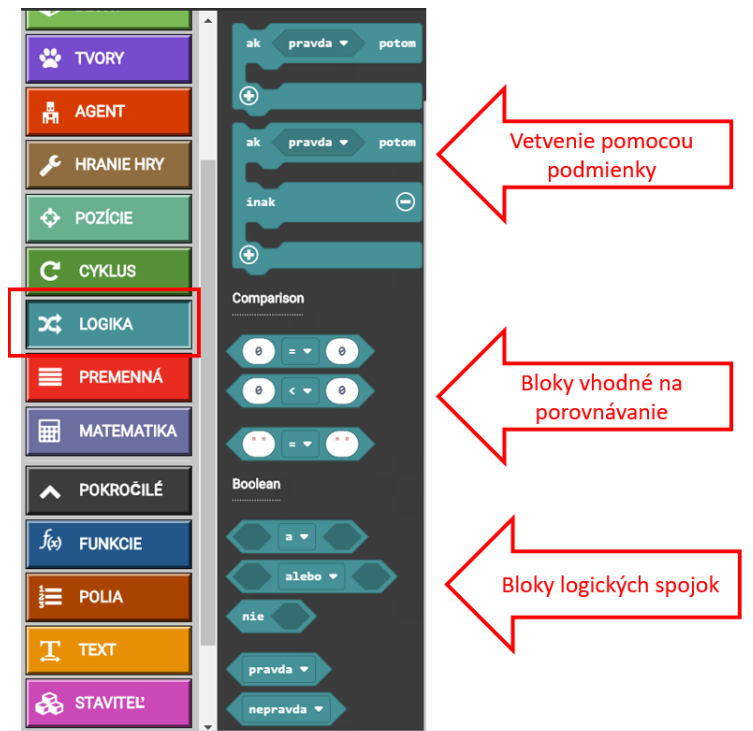


(Pripravený program je vždy iba ukážkový, necháme žiakom možnosť samostatnej práce, možno navrhnu efektívnejší program)

**POJMY:**

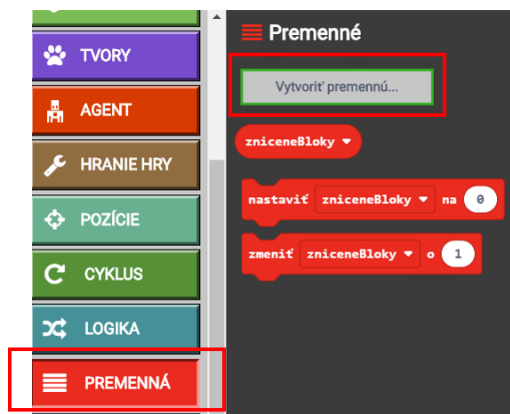
**Vetvenie** – sa vytvára pomocou podmienky v MEE sú na to určené dva bloky Ak pravda potom... a Ak pravda potom..., inak ...

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



Obrázok 1 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 2 Vytvorenie premennej

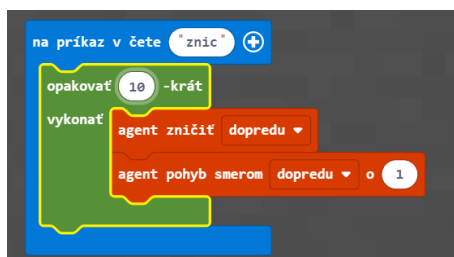
### **Motivácia:**

Žiaci sa vždy potešia, keď v Minecrafte môžu niečo rozbiť. Počas tejto hodiny budeme rozbiť bloky. Pretože študenti radi aj stavajú, počas tejto časti si budú pripravovať aj stavby na zbúranie. Žiaci medzi sebou môžu súťažiť, ktorému agentovi sa za hodinu podarí rozbiť čo najviac blokov.

## Krok 1

Necháme žiakov, aby si postavili rad blokov, agenta umiestnili pred rad a nechali ho ničiť bloky pred sebou. Môžeme to zadať ako samostatnú prácu, či prácu v skupine na rýchle opakovanie.

Vznikne nám jednoduchý program s cyklom:



Obrázok 3 Cyklus

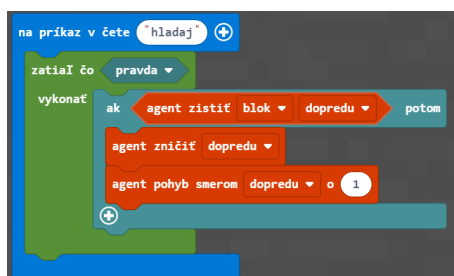
## Krok 2

Necháme žiakov, aby nám zdôvodnili tvorbu programu a skúsime sa zamyslieť nad otázkami typu:

*Aký dlhý bude rad, kde nevieme koľko blokov má agent zničiť a predsa chceme, aby zničil všetky bloky, ako to dokáže?*

Ukážeme záložku „Logika“, kde si vyberieme jednoduchú podmienku „ak pravda potom“. Namiesto položky „pravda“ vyberieme z „Agenta“ špicatú položku „agent zistiť blok dopredu“.

Do tela cyklu vložíme náš už existujúci program. A vyskúšame.



Obrázok 4 Krok 2

Kladieme otázky: *Kedy agent prestane rozbíjať bloky? Prečo?*

Keď máme utvrdené pochopenie tejto časti programu. Prejdeme ďalej:

## Krok 3

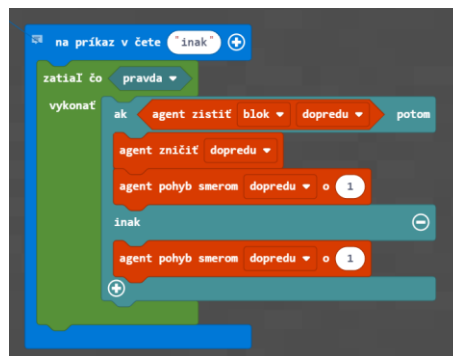
Položíme otázky:

*Čo sa stane v tomto programe s agentom, ak v rade bude chýbať nejaký blok? Čo agent urobí? Ako zabezpečiť, aby vedel prekročiť prázdny priestor a pokračoval v ničení ďalej?*



Obrázok 5 Rad blokov

Upriamime pozornosť na možnosť „+“ pri vetvení. Do vetvy „inak“ vložíme, čo má agent urobiť ak nemá pred sebou blok. Do vetvy „inak“ vložíme blok „agent smerom dopredu pohyb o 1“. Znamená to, že ak agent bude mať pred sebou prázdny priestor, bude kráčať o jeden blok dopredu, až kým nenarazí na nejaký blok a nezničí ho. Vyskúšame náš vylepšený program.



Obrázok 6 Krok 3

#### Krok 4

Vidíme, že agent naozaj rozbíja bloky a tam, kde je prázdne miesto, automaticky urobí krok dopredu. Nás ale zaujíma, koľko blokov vlastne zničil.

Kladieme otázky: *Ako by sme mohli získať počet blokov, ktoré agent zničil? Kde v programe by sme mohli takéto počítadlo krokov vložiť? V ktorej vetve programu by sa malo takéto počítadlo nachádzať?*

Počítadlo, ktoré bude počítať rozbité bloky sa nazýva premenná. Túto nájdeme v položke „Premenná“. Táto položka je zatiaľ prázdna. Napíšeme meno novej premennej do položky „Vytvor novú premennú“, tým sa nám vytvoria bloky na používanie premennej.

Na začiatku programu musíme našu novú premennú nastaviť na hodnotu 0. Premennú sme nazvali „pocitadlo“. Blok „nastaviť pocitadlo na 0“, bude hneď prvým blokom nášho programu. Môžeme diskutovať o tom, kam vložiť blok „zmeniť pocitadlo o 1“. My ho vložíme za blok „agent zničiť dopredu“.

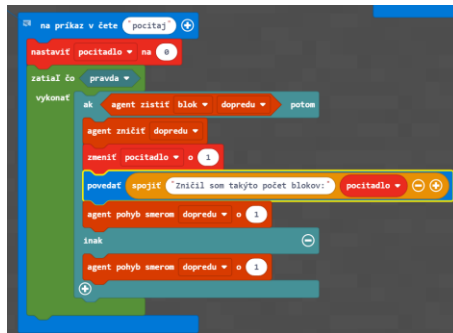
Teraz nám už program počíta, koľko blokov sme zničili, ale nevieme to, pretože sa nám nikde nezobrazuje výstup z premennej „pocitadlo“.

V „Pokročilé“ v položke „Text“ si vyberieme oválny blok „spojiť „Dobrý deň“ „Svet“ + -“. Do položky „Dobrý deň“ vpíšeme svoj text a do položky „Svet“ vložíme oválny blok z položky „pocitadlo“. Týmto spojením dosiahneme, že pri každom zavolaní tohto bloku sa nám na obrazovke vypíše aktuálny stav zničených blokov. Tento poskladaný oválny textový blok je ešte potrebné vložiť do bloku z časti „Hráč“, do bloku „povedať“. Tak ako to je na obrázku:



Obrázok 7 Blok "povedať"

Keď máme takto zložený blok textu, môžeme pouvažovať, kde ho chceme umiestniť. My chceme, aby sme boli informovaní o zmene po každom zničenom bloku. Takže textový blok umiestnime za blok „zmeniť pocitadlo o 1“.



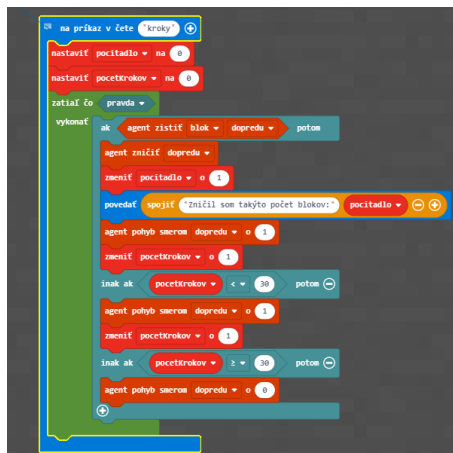
Obrázok 8 Krok 4

### Krok 5

Náš agent nám zatiaľ uteká a vieme ho zastaviť iba tak, že stlačíme klávesu C a tým prerušíme jeho program, alebo si ho privoláme naspäť, cez „teleportovať k hráčovi“.

My by sme však chceli, aby sa po 30 krokoch zastavil. Pýtame sa žiakov:

*Ako je možné zabezpečiť, že agent zastane po 30 krokoch? Je tu možné využiť vetvenie a premenné?*



Obrázok 9 Krok 5

### Pre šikovných:

Skúste program dotvoriť tak, aby agent reagoval aj na bloky vo svojom okolí (vpravo, vľavo a možno aj hore).

