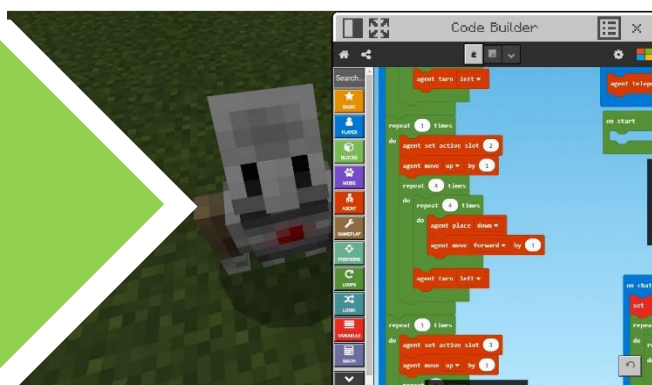


Metodický list č. 3

Cieľová skupina žiakov: 5. a 6. ročník ZŠ

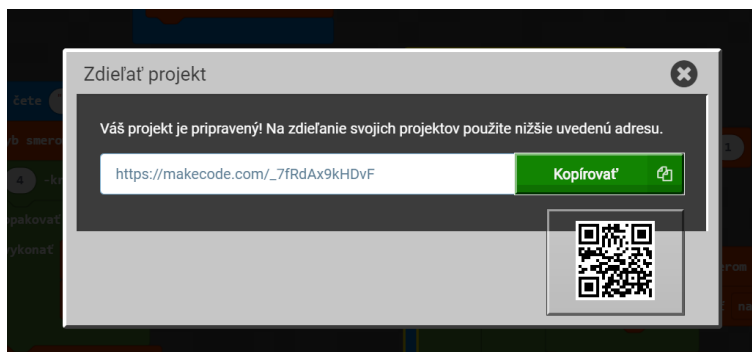
Spracované podľa Inovovaného ŠVP pre 2.stupeň
ZŠ Matematika a práca s informáciami časť
Informatika- nižšie stredné vzdelávanie



Požiadavky na zručnosti žiakov:	<ul style="list-style-type: none"> • Zostaviť jednoduchý program • Poznať základné klávesové skratky E inventár, C programovanie, T chat
Názov metodického listu:	<h1>Cyklus</h1>
Učivo:	Algoritmické riešenie problémov – <ol style="list-style-type: none"> 1. pomocou postupnosti príkazov 2. pomocou cyklov
Výkonový štandard (podľa iŠVP):	<ul style="list-style-type: none"> ▪ rozpoznať opakujúce sa vzory pri riešení zadaného problému, ▪ rozpoznať, aká časť algoritmu sa má vykonať pred, počas a po skončení cyklu, ▪ stanoviť počet opakovaní pomocou hodnoty, ▪ riešiť problémy, ktoré vyžadujú známy počet opakovaní, ▪ zapísať riešenie problému s cyklom pomocou jazyka, ▪ interpretovať algoritmy s cyklami
Obsahový štandard(podľa iŠVP):	<ul style="list-style-type: none"> ▪ <i>Pojmy:</i> opakovanie, počet opakovaní, telo cyklu ▪ <i>Vlastnosti a vzťahy:</i> ako súvisí počet opakovaní s výsledkom ▪ <i>Procesy:</i> zostavenie a upravenie tela cyklu, nastavenie počtu opakovaní
Ciele:	Zopakovať a utvrdiť zostavovanie programu z blokov, skladanie príkazov do postupností. Zamerať sa na 3D rozmer MEE. Vytvoriť program na postavenie veže/ domčeka.
Vyučovacie metódy:	Riadený rozhovor - analýza problému, brainstorming
Organizačné formy práce:	Spoločná práca, samostatná práca

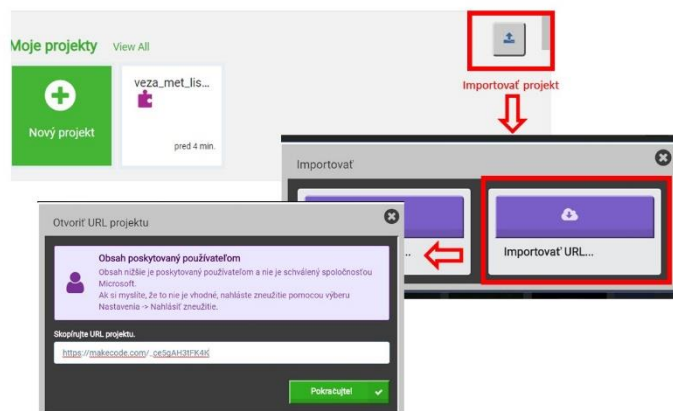
Všetky programy potrebné v tomto metodickom liste nájdete tu:

https://makecode.com/_7fRdAx9kHDvF



Obrázok 1 Programy

Keď chcete mať všetky programy potrebné k tomuto metodickému listu, nájdete ich na danom linku. Toto prepojenie je potrebné skopírovať. Po kliknutí na C v záložke Domov



Obrázok 2 Návod na vloženie prepojenia

Cyklus - programová konštrukcia, do ktorej vložíme nejaké príkazy a deklarujeme, koľkokrát ich je potrebné opakovať.

Motivácia:

Je potrebné si zopakovať ako sa jednotlivé príkazy spájajú do postupnosti.

Osvedčené je prirovnanie ku murárovi, ktorý postupne ukladá jednotlivé tehly vedľa seba až z nich postaví múr. Na tejto predstave budeme ďalej stavať. Ako vypadá miestnosť? Má štvorcový, alebo obdĺžnikový pôdorys. Túto jednoduchú hru si môžu zahrať aj žiaci: Keď idem dookola okolo miestnosti, čo robím? Idem, kým neprídem do rohu, otočím sa a opäť idem, až prídem do rohu otočím sa a tak ďalej. Kým obídem celú miestnosť, koľko razy zopakujem tú istú činnosť?

Ak by sme chceli aby náš agent, vystaval takúto miestnosť ako murár, čo by musel urobiť? Najskôr položí blok a posunie sa a opakuje to – koľkokrát? Opakuje jednu a tú istú činnosť viackrát za sebou – vytvára cyklus a koľkokrát to opakuje je počet opakovaní toho istého cyklu.

Krok 1:

Vytvor jeden rad blokov ľubovoľnej dĺžky.



Na tvorbu programu využívame záložku **CYKLUS**. V programoch, ktoré sa načítajú nájdeme aj program s názvom „rad“. Ten sme vytvorili minule. Tento program potrebuje 6 blokov na polozenie radu s dĺžkou 3 blokov. Porovnajme ho s programom „stena“, ktorý potrebuje 3 bloky na vytvorenie radu blokov s dĺžkou 10.

Postupnosť príkazov



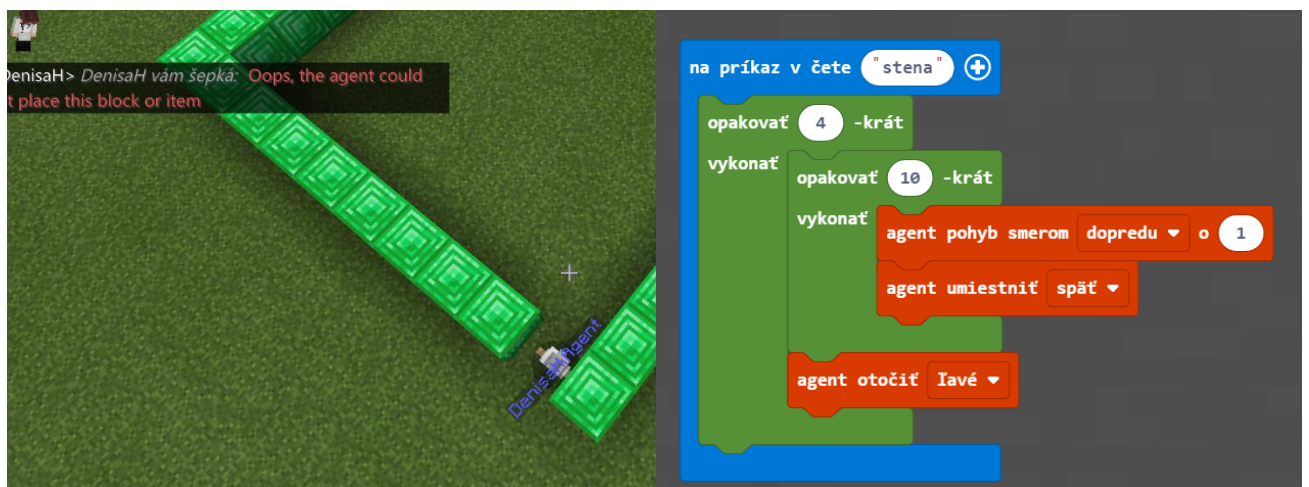
Obrázok 3 Efektivita použitia cyklu

Keď už vieme postaviť jeden bočný rad, skúsme vytvoriť celý štvorec.

Krok 2:

Brainstorming

Vytvor štvorec so stranou ľubovoľnej dĺžky.



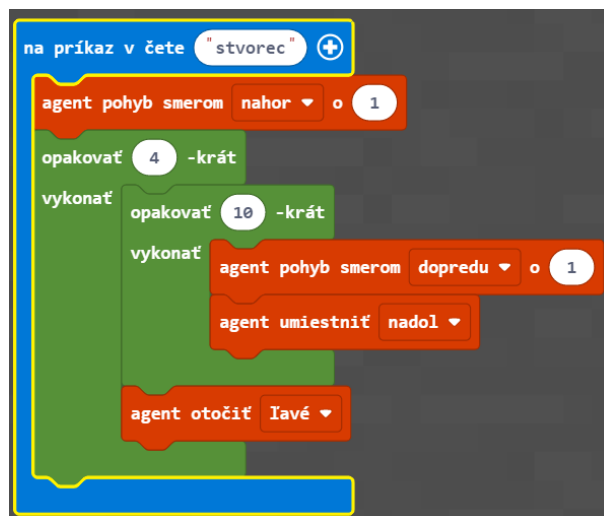
Obrázok 4 Chybové hlásenie

Pri vytvorení programu z pôvodného algoritmu, narazíme na problém – náš agent nevie položiť posledný blok, pretože mu zavadzia blok, ktorý položil ako prvý.

V tomto momente môžeme zaradiť brainstorming – nechať žiakov, aby prichádzali s vlastnými riešeniami. Nekomentujeme ich, iba ich zapisujeme na tabuľu. Po pár minútach prejdeme ku zvažovaniu jednotlivých možností.

Ak sme zvažovali všetky možnosti, ako správne, by sa nám mohli javiť viaceré možnosti, napr. agent by mohol ukladať bloky do boku – v tomto prípade, by sa nám komplikovane riešil prechod v rohu pri otočení.

Najjednoduchšie riešenie? Musíme si uvedomiť, že pracujeme v 3D. Náš agent, hneď na začiatku, vyletí o jeden blok hore a kladie bloky pod seba. Vtedy nenastáva žiadny problém. Tento program sa volá „štvorec“ v importovanom kóde.



Obrázok 5 Program pre štvorec

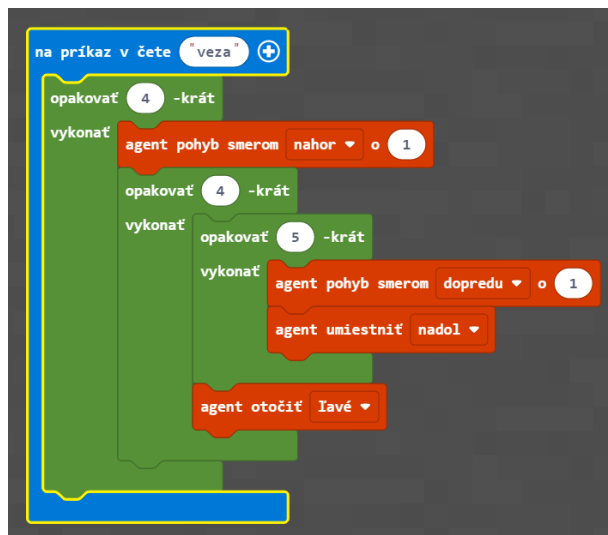
Krok 3:

Postav vežu!

Túto úlohu už môžeme zadať ako samostatnú prácu. Každý kto dokázal postaviť štvorec, teraz už vie, že po ukončení programu „štvorec“ musí ísť o jeden blok nahor a stavať ďalší identický štvorec.

Treba tu upozorniť žiakov na potrebu presnej postupnosti príkazov. Môžu experimentovať s tým, čo sa udeje ak zmeníme pozíciu bloku „agent smerom nahor o 1“. (Vytvárajú sa schody, alebo sa nevytvára nové poschodie a agent hlási chybu)

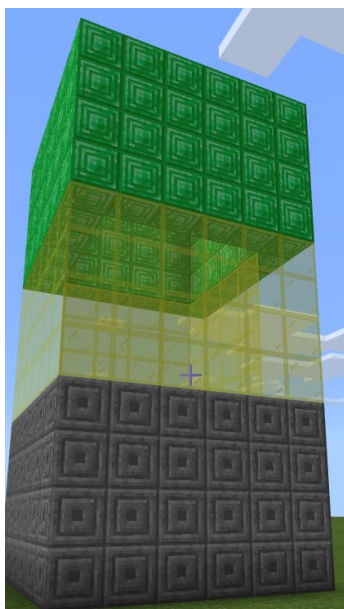
Vytvoríme teda program, kde sú vnorené cykly. V importovaných programoch sa tento nazýva „veža“.



Obrázok 6 Program veža

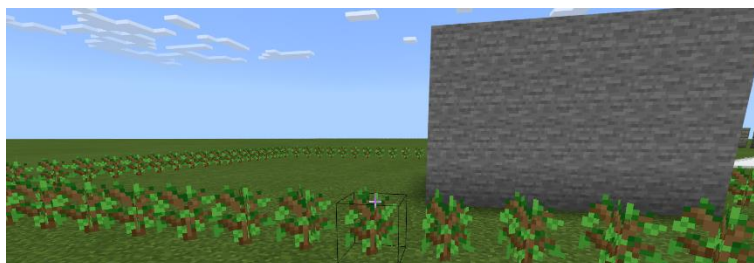
Extra úloha:

Postav viacfarebnú vežu!



Obrázok 7 Farebná veža

Pomocou programu štvorec okolo svojej veže vysaď záhradu, alebo postav plot!



Obrázok 8 Veža so živým plotom