

Metodický list č. 1

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

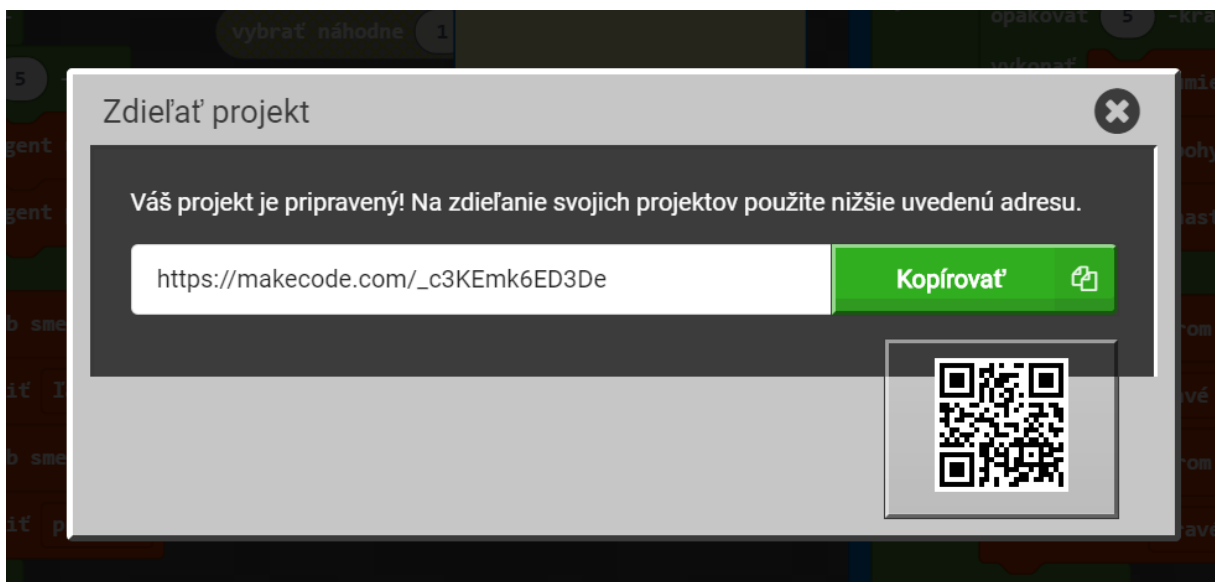


<p>Požiadavky na zručnosti žiakov:</p>	<ul style="list-style-type: none"> • Zostaviť jednoduchý program, vedieť interpretovať zapísané • Poznať pojem príkaz, cyklus • Poznať základné klávesové skratky E inventár, C programovanie, T chat
<p>Názov metodického listu:</p>	<h2>Práca so slotom a náhodné čísla</h2>
<p>Učivo:</p>	<p>Algoritmické riešenie problémov</p> <ul style="list-style-type: none"> - analýza problému - pomocou postupnosti príkazov - pomocou cyklov - jazyk na zápis riešenia - pomocou nástrojov na interakciu
<p>Výkonový štandard (podľa iŠVP):</p>	<ul style="list-style-type: none"> ▪ identifikovať opakujúce sa vzory, ▪ rozpoznávať miesta, kde sa treba rozhodovať, ▪ vlastnými slovami sformulovať plán riešenia, ▪ rozhodnúť sa o pravdivosti/nepravdivosti tvrdenia/výroku, ▪ uviesť kontra príklad, keď niečo neplatí, nefunguje, ▪ uvažovať o rôznych riešeniach. ▪ zapísať algoritmus, ktorý reaguje na vstup, interpretovať zapísané riešenie, vytvoriť hypotézu, ako neznámy algoritmus spracuje zadaný vstup
<p>Obsahový štandard(podľa iŠVP):</p>	<p>Analýza problému <i>Vlastnosti a vzťahy:</i> aký informatický problém je v zadaní úlohy, platí – neplatí, a/alebo/nie (neformálne) <i>Procesy:</i> idea sekvencie, opakovania, vetvenia, manipulovania s údajmi, rozhodovanie o pravdivosti tvrdenia</p> <p>Jazyk na zápis riešenia <i>Vlastnosti a vzťahy:</i> algoritmus – programovací jazyk, vstup – algoritmus – výsledok, chybný zápis, konštrukcie jazyka ako: postupnosť príkazov, cyklus s pevným počtom opakovaní, podmienený príkaz, pomenovaná postupnosť príkazov</p> <p><i>Procesy:</i> zostavovanie programu v jazyku na zápis algoritmov, spustenie programu</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:	<p>Zopakovať a utvrdiť zostavovanie programu z blokov, skladanie príkazov do postupností a cyklov. Výber náhodného čísla</p> <p>Zaviesť pojem „slot“ a použiť ho. Identifikovať chybu a navrhnúť opravu. Zopakovať pojem premenná a pracovať s ňou</p>
Vyučovacie metódy:	<p>Frontálne vyučovanie založené na konštruktivistických princípoch stavania na predchádzajúcich vedomostiach</p>
Organizačné formy práce:	<p>Spoločná práca, práca vo dvojici/ skupine. Práca s pracovným listom</p>

Programy potrebné pre tento metodický list:

https://makecode.com/_c3KEmk6ED3De



Motivácia:

Doteraz sme stavali s agentom jednofarebné stavby, alebo sme mu ručne menili bloky, ktoré používal. Dnes sa naučíme vytvoriť Patchwork z blokov. Ako príklad ukážeme podobnú vzorku Patchworku.



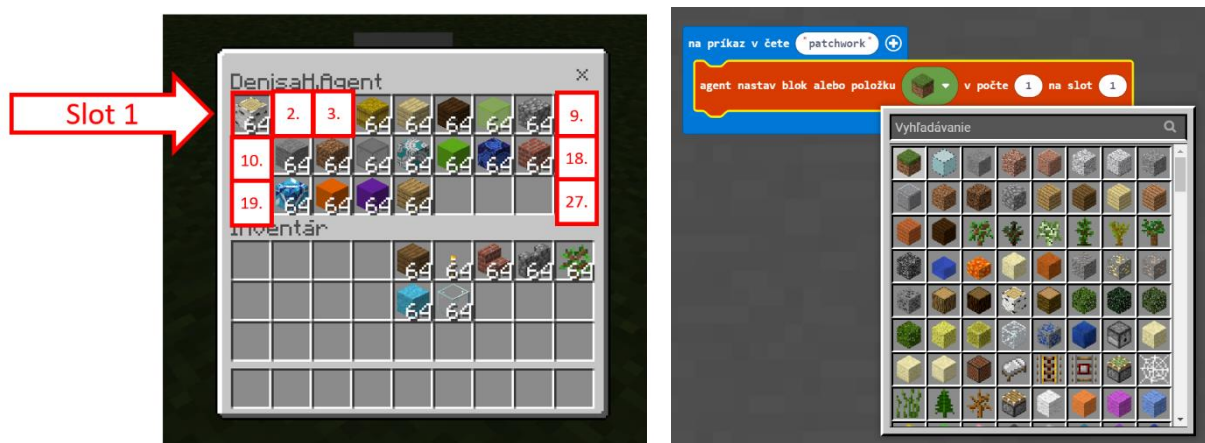
Obrázok 1 Patchwork

[Link na obrázok](#)

Aby sme mohli prepínať medzi jednotlivými blokmi, potrebujeme vedieť, kde a ako sa sloty dajú naplniť.

Prvá možnosť je jednoduchým pridaním do agentovho inventára. Keď máme v inventári 64 blokov z nejakého materiálu, máme ich nekonečne veľa.

Druhou možnosťou pridať konkrétny blok na konkrétne miesto/ slot je pomocou bloku „agent nastav blok alebo položku __ v počte__ na slot__“ priamo v programe.



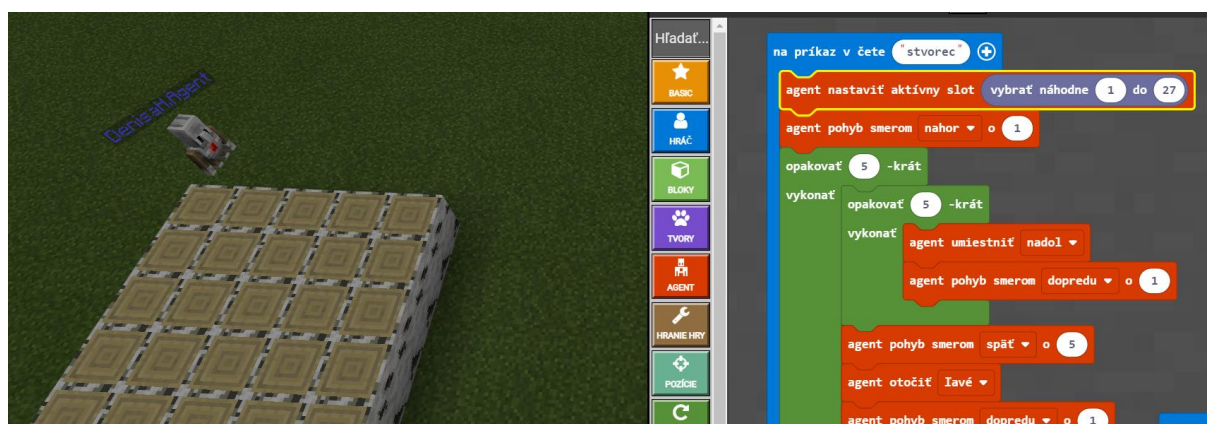
Obrázok 2 Slot

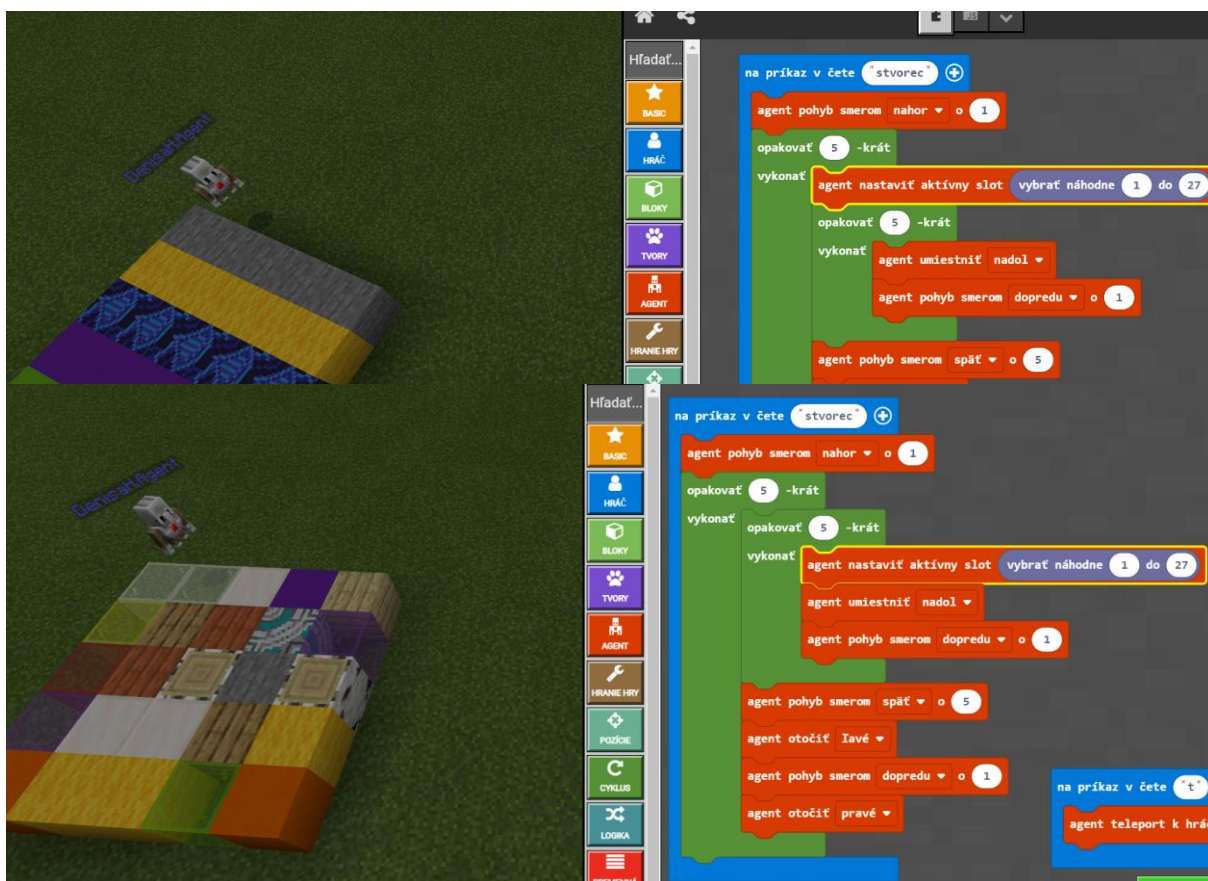
Krok 1:

V programoch je vložený program „štvorec“.

Prvý v programe je umiestnený blok, „agent nastaviť aktívny slot“, vo vnútri je vložený náhodný výber čísla slotu (nájdete ho v časti matematika). Nechajte žiakov experimentovať s tým, ako sa bude meniť vzor postaveného štvorca, keď budú presúvať tento blok na rôzne miesta v programe.

Pýtame sa na dôvod, prečo sa program mení a kde nastávajú tieto zmeny





Obrázok 3 Rozdiely v pozícií a ich ovplyvnenie výsledného štvorca

Krok 2:

Upravme program tak, aby sme agentovi mohli zadať aj stranu štvorca, ktorý má postaviť a aby vypísal koľko blokov na jeho stavbu spotreboval.

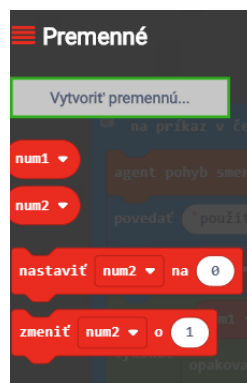
Tento program sa nazýva aj program s parametrom

Blok „na príkaz v čete „square2““ určuje aký program sa bude vykonávať, ale o tom aký veľký štvorec postavíme sa postará práve parameter príkazu . Jeho zadaním v príkazovom riadku, sa hodnota parametru zavedie ako premenná „num1“.

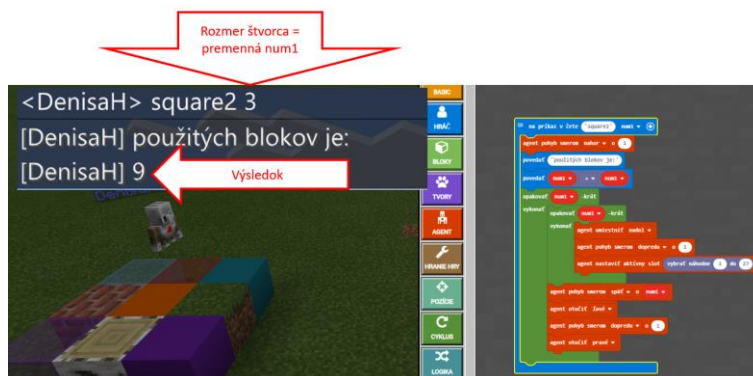


Obrázok 4 Program s parametrom

Premennú vytvoríme v záložke premenná. Ak využívame parametre, zavedú sa nám nasledujúce premenné automaticky.



Obrázok 5 premenné "num1" a "num2"



Obrázok 4 Štvorec s parametrom

Krok 3:

Upravme program tak, aby sme mohli stavať aj obdĺžniky (Potrebné zaviesť aj druhú premennú)

Popremýšľajme, na aký iný účel vieme v škole použiť takéto útvary? Vedeli by niekomu pomôcť pri predstavení si nejakých matematických operácií?

Odpoveď: Tento program je výbornou pomôckou pri učení sa násobilky, pretože žiaci naozaj vidia ako sa pred nimi stavia/prebieha matematická operácia.

premenne num1 num2

The image shows a Scratch-like environment. On the left, a 3D scene displays a tall tower of various colored blocks (wood, stone, brick, etc.) on a green grassy field. A blue text label 'DenisaAgent' is visible near the base of the tower. The top-left corner shows a chat window with the following text: 'DenisaH> nasobilka 9 2', 'DenisaH] použitých blokov je:', and 'DenisaH] 18'. The right side of the image is a code editor with a dark background. The code is written in a block-based language and is titled 'na prikaz v čete "nasobilka" num1 num2'. The code includes several blocks: 'agent pohyb smerom nahor' with a value of 1; 'povedat' blocks for 'použitých blokov je:' and a multiplication operation 'num2 * num1'; a loop 'opakovat' with 'num2' iterations containing 'vykonat' blocks for 'opakovat' with 'num1' iterations, 'agent umiestniť nadol', 'agent pohyb smerom dopredu' with a value of 1, and 'agent nastaví aktívny slot' with 'vybrať náhodne' from 1 to 27; and final movement blocks 'agent pohyb smerom späť' with 'num1', 'agent otočiť ľavé', and 'agent pohyb smerom dopredu' with a value of 1, followed by 'agent otočiť pravé'. A green play button is visible at the bottom right of the code editor.

Obrázok 5 Násobilka