

Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika

Riešenie problémov a základy programovania 2

Predmet: **Riešenie problémov a základy programovania**

Línia: **Vlastný odborový kontext informatiky a informatickej výchovy**



Riešenie problémov a základy programovania 2

Identifikácia modulu

Aktivita projektu: 1.1 Vzdelávanie učiteľov 1. stupňa ZŠ na informatiku a informatickú výchovu

Línia aktivity: Vlastný odborový kontext informatiky a informatickej výchovy

Predmet: Riešenie problémov a základy programovania

Zaradenie modulu

Modul Riešenie problémov a základy programovania 2 je druhým modulom tohto predmetu. V ďalšom vzdelávaní na oba moduly predmetu Riešenie problémov a základy programovania nadviaže modul Didaktika informatickej výchovy 5.



Predmet patrí do línie Vlastný odborový kontext informatiky a informatickej výchovy. Aby mohol účastník absolvovať tento modul, musí mať absolvované moduly Digitálna gramotnosť 1 až 4, Informácie okolo nás 1 až 3 a modul Riešenie problémov a základy programovania 1.

Abstrakt modulu

K tomu, aby učiteľ na 1. stupni ZŠ mohol kvalifikovane vyučovať predmet informatická výchova, je nevyhnutné, aby aj on sám zvládol základy programovania v nejakom programovacom jazyku. V rámci tohto modulu sme pripravili také prostredia, v ktorých bude pre učiteľa jednoduché pochopiť základné princípy programovania aj vďaka tomu, že pri vytváraní programov bude mať ihneď spätnú odozvu, napr. nakreslený obrázok. Pritom bude navrhovať iba krátke a jednoduché programy, ktoré nevyžadujú žiadne ďalšie vedomosti. K jednoduchej práci v prostrediach prispieje aj to, že sa v nich používajú slovenské príkazy alebo sa príkazy zobrazujú graficky svojím obrázkom.

Garant predmetu:

PaedDr. Monika Tomcsányiová, PhD.
KZVI FMFI UK, Bratislava
tomcsanyiova@fmph.uniba.sk

PaedDr. Monika Tomcsányiová, PhD., KZVI FMFI UK, Bratislava
RNDr. Lubomír Salanci, PhD.
KZVI FMFI UK, Bratislava
PaedDr. Daniela Bezáková, PhD., KZVI FMFI UK, Bratislava
RNDr. Andrej Blaho
KAI FMFI UK, Bratislava
Mgr. Daniela Onačilová
KZVI FMFI UK, Bratislava

Riešenie problémov a základy programovania 2	1
Identifikácia modulu	1
Zaradenie modulu	1
Abstrakt modulu	1
Obsah	2
Úvod	3
Cieľ modulu	3
Vstupné vedomosti	3
Požadované prerekvizity	3
Predpokladané vstupné vedomosti, skúsenosti a zručnosti	3
Preverenie vstupných vedomostí	3
Programovanie v prostredí Panák	4
Prostredie Panák	4
Riešte úlohy	5
Programovanie v prostredí Zjednodušené logo	8
Pohyb v priamom režime	8
Hľadanie cesty	8
Opravte postupnosť príkazov	9
Doplňte chýbajúci príkaz	10
Kreslenie obrázkov čiarami	10
Navrhovanie nových príkazov	11
Vytváranie živých obrazov	12
Prostredie	12
Objekty, ich tvary a pohyb	12
Neinteraktívne animácie	13
Interaktívne animácie	15
Skladačky	17
Zrážky	18
Labyrinty	19
Čo sme sa naučili v tomto module	21
Čo sme sa naučili	21
Absolvovanie predmetu Riešenie problémov a základy programovania	21
Literatúra a použité zdroje	21
Riešenia úloh z prostredia Panák	22
Riešenia úloh z prostredia Zjednodušené Logo	24
Riešenie zadaní 9 a 10	24
Riešenie zadania 13	24
Riešenie zadania 14	24
Riešenia úloh z prostredia Živý obraz	25
Aktivita 11 - Ďalšie objekty v rybníku	25
Aktivita 14 - Háďaj, kto je to?	25
Aktivita 16 - Tangram	26
Aktivita 18 - Včielka a biely kvet	27
Aktivita 19 - Preteky dinosaurov	27

Úvod

Základným cieľom informatickej výchovy je pre jednoduché problémy navrhnúť ich riešenie a toto riešenie dokázať prezentovať.

Modul Riešenie problémov a základy programovania 2 sa realizuje v počítačovej učebni s dataprojektorom. Každý účastník vzdelávania pracuje za vlastným počítačom. V rámci modulu budú účastníci vzdelávania pod vedením lektorov používať aj programy špeciálne navrhnuté pre tento modul.

Cieľ modulu

Cieľom modulu je, aby sa účastníci vzdelávania oboznámili s možnosťami zapisovania riešenia problémov pomocou vopred známej množiny slov. Aby si uvedomili, že postupnosť krokov, ktorá vedie k riešeniu problému, sa dá zapísať spôsobom, ktorý je vhodný na automatické spracovanie a ktorému dokáže porozumieť aj počítač.

V rámci modulu poukážeme na to, že rôzne typy úloh si často vyžadujú rôznorodé prostredia.

Vstupné vedomosti

Požadované prerekvizity

Aby mohol účastník absolvovať tento modul, musí mať absolvované moduly

- Digitálna gramotnosť 1 až 4,
- Informácie okolo nás 1 a 2,
- Riešenie problémov a základy programovania 1

Predpokladané vstupné vedomosti, skúsenosti a zručnosti

Účastník vzdelávania má základnú digitálnu gramotnosť, dokáže pracovať s textom a tabuľkami. Vie prezentovať svoje myšlienky pomocou digitálnych technológií. Dokáže tiež pracovať v niektorom grafickom editore, kresliť rastrové obrázky a je schopný pripraviť jednoduché animované obrázky. Rozumie práci so súbormi a priečinkami. Má základné znalosti o riešení problémov, o zápise týchto riešení rôznymi spôsobmi, napr. graficky, textom, či pomocou kartičiek s obmedzeným počtom príkazov.

Preverenie vstupných vedomostí

V rámci preverenia vstupných vedomostí pripraví účastník vzdelávania jednoduchú aktivitu v prostredí Kartičkové aktivity alebo Mravec. S týmito prostrediami pracoval v module Riešenie problémov a základy programovania 1.

Programovanie v prostredí Panák

Na zopakovanie vedomostí o tom, že na zapisovanie riešenia problémov môžeme použiť ohraničenú množinu príkazov využijeme prostredie Panák [5].

Program Panak.exe nájdete v prostredí Moodle.

Prostredie Panák

Prostredie Panák predstavuje pripravený mikrosvet, ktorý nám umožňuje riadiť panáčka pomocou základných príkazov - vpravo, vľavo, vyskoč, zoskoč a riadiacej štruktúry cyklus, ktorá umožňuje zopakovať postupnosť príkazov niekoľkokrát.

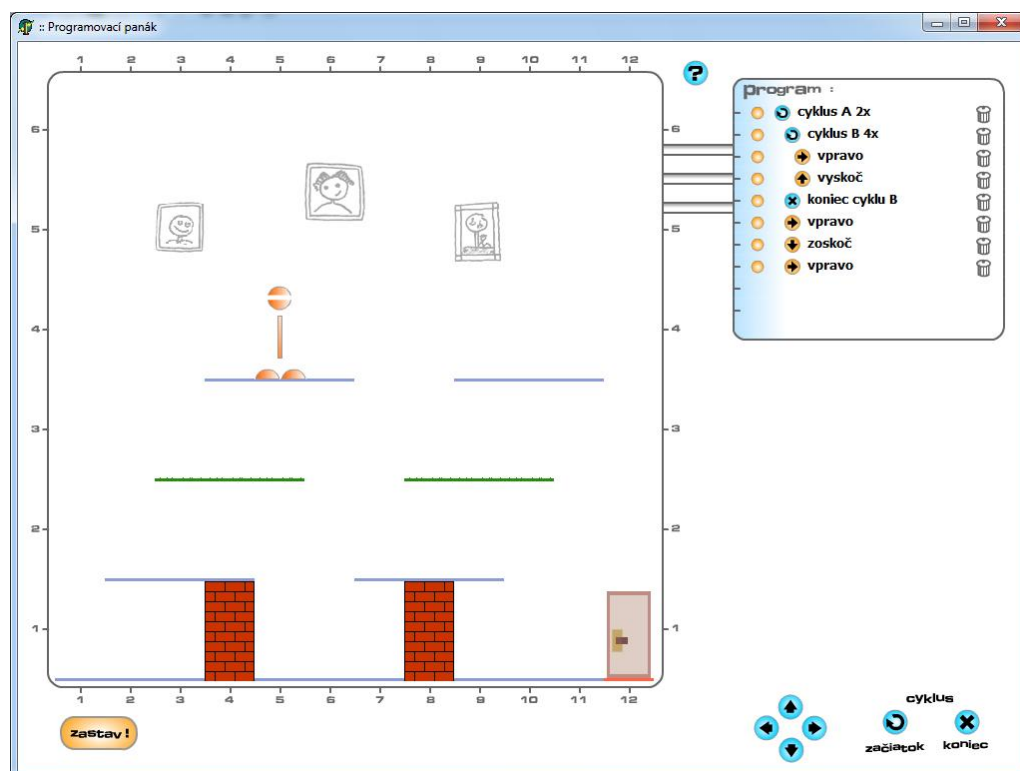
Príkazy sa zadávajú pomocou tlačidiel so šípkami v pravej časti prostredia. Týmto spôsobom vytvárame program, ktorý panáček vykoná po stlačení tlačidla spusti!. Ak sa panáček dostane do dverí, objaví sa tlačidlo, ktoré nám umožní prejsť do ďalšej, zložitejšej, úrovne.



Tlačidlá na riadenie panáčka



Tlačidlá na spúšťanie a zastavenie pripraveného programu



Prostredie Panák

Riešenia jednotlivých úloh nájdete na konci materiálu.

Samotné prostredie je pripravené tak, že umožňuje skúšať riešenie úlohy aj niekoľkokrát. Ak sa panáček pri vykonaní všetkých príkazov v programe nedostane ku dverám, zastane a my môžeme pokračovať v pridávaní nových príkazov. Pri spustení programu panáček začne vykonávať všetky príkazy vždy od začiatku. Do ďalšej úrovne prejdeme až po správnom vyriešení súčasnej úrovne.

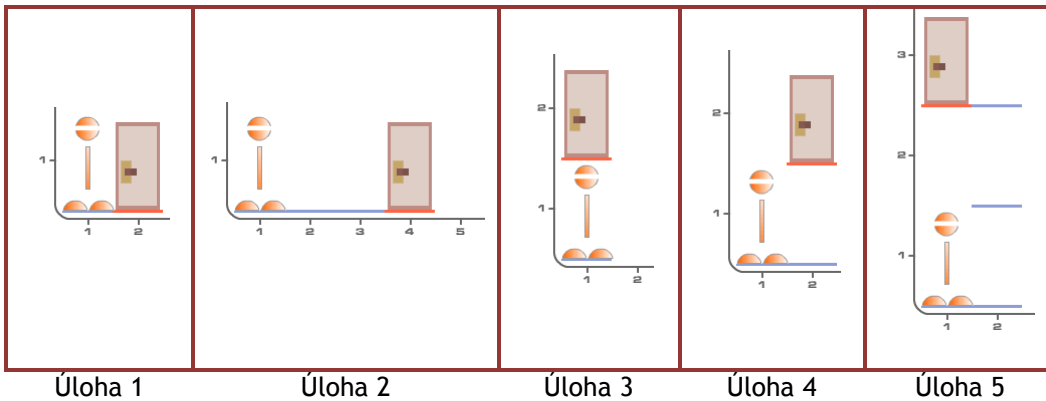
Na ďalších stranách materiálu, ale predovšetkým pri riešení cielene pripravených úrovní spoznáme možnosti prostredia, nové príkazy, ktoré môžeme využiť, ale aj obmedzenia, ktoré sú v prostredí realizované. Takýmto spôsobom sa naučíme nad svojím riešením uvažovať a kvôli obmedzeniam často aj hľadať nové, efektívnejšie postupy, ktoré vyriešia príslušnú úroveň.

Riešte úlohy




Do tejto časti materiálu sme pripravili niekoľko úloh, pri ktorých spoločne spoznáme prostredie, oboznámime sa s množinou príkazov, ktoré ponúka na riešenie jednotlivých úrovní a samozrejme vytvoríme niekoľko jednoduchých ale aj zložitejších programov. Zadanie všetkých úloh je rovnaké - priviesť panáčika na políčko, na ktorom sú dvere. Prostredie obmedzuje riešenia niekoľkými spôsobmi:

- prekážkami v podobe červenej tehlovej steny alebo dierami v podlahe,
- počtom príkazov, ktoré môžeme využiť pri riešení úlohy.

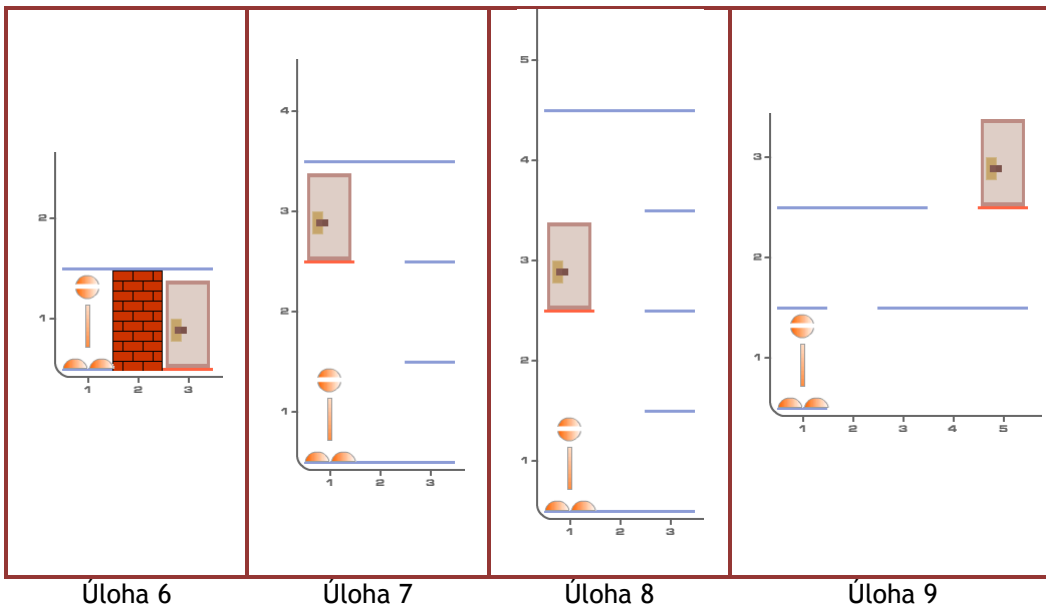
Úlohy 1 až 5




Pri riešení úloh 1 až 5 využite

- šípku vpravo , ktorá presúva panáčika o jeden krok vpravo,
- šípku vyskoč , ktorá umožní panáčikovi vyskočiť o jedno poschodie vyššie, ak je na tomto poschodí čiarka, t.j. podlaha,
- šípku vľavo , ktorá presúva panáčika o jeden krok vľavo,
- vo všetkých prípadoch panáčik zostane na políčku iba ak je na ňom podlaha, inak padá na najbližšiu podlahu.

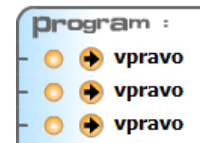
Úlohy 6 až 9



Všimnite si, že nohy panáčika sú na začiatku na čiaračke s číslom 1.

Stlačte modrú šípku .

V pravej časti prostredia sa objavuje program.



Panáčik zatiaľ nereaguje.

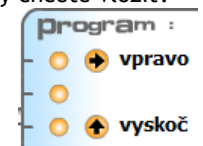
Stlačte tlačidlo .

Panáčik poslušne vykoná program.

Všimnite si, že panáčik začína vykonávať program vždy od začiatku.

Ak ste sa v programe pomýlili a chcete zrušiť niektorý príkaz, stlačte kôš pri ňom. Príkaz zmizne z postupnosti príkazov.

Ak chcete do programu vložiť nový riadok, kliknite na guľôčku v tom riadku, ktorý chcete vložiť.



Pozor, v úlohe 6 je hneď vpravo vedľa panáčika stena cez ktorú nevie prejsť!

Všimnite si, že panáčik spadne na najbližšiu podlahu, ktorá je pod ním.

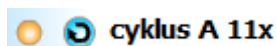
Panáčikovi nevadí, keď padá aj niekoľko poschodí.

Všimnite si, že počet čiaročiek udáva maximálny počet riadkov, ktoré môže mať náš program.



Tlačidlá začiatku a konca cyklu.

Klikaním ľavým tlačidlom na číslo sa zvyšuje počet opakovaní cyklu. Ak klikáte pravým tlačidlom, číslo sa znižuje.



Všimnite si, že využitím cyklu pre príkaz vpravo sme ušetrili niekoľko riadkov programu.

Uvedomte si, že opakovať sa môže jeden príkaz, ale aj viac príkazov.


Nájsť opakujúcu sa postupnosť príkazov niekedy nemusí byť celkom jednoduché.

V niektorých prípadoch sa miestnosť podobá na schodíky a vtedy to môže byť ľahšie.

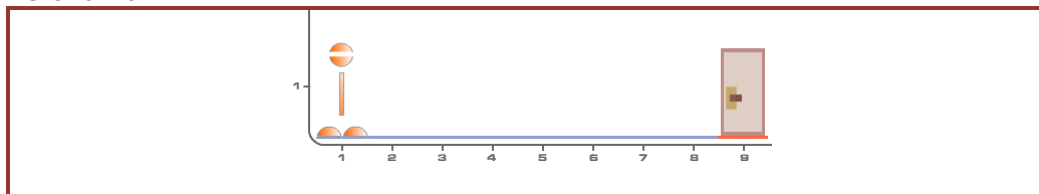
Inokedy ušetríme riadky v programe tým, že opakujeme iba jediný príkaz.

Pri riešení úlohy 13 uvažujte, kde stojí panáčik vzhľadom na prejdenný počet krokov na vyššom poschodí.

Pri riešení úloh 6 až 9 využite

- kombináciu šípok vpravo, vyskoč, vľavo,
- šípku zoskoč , ktorá umožní panáčikovi zoskočiť na nižšie poschodie bez ohľadu na to, či je pod ním podlaha, ak podlaha nie je na poschodí na ktoré zoskočí, padá na najbližšiu podlahu.

Úloha 10



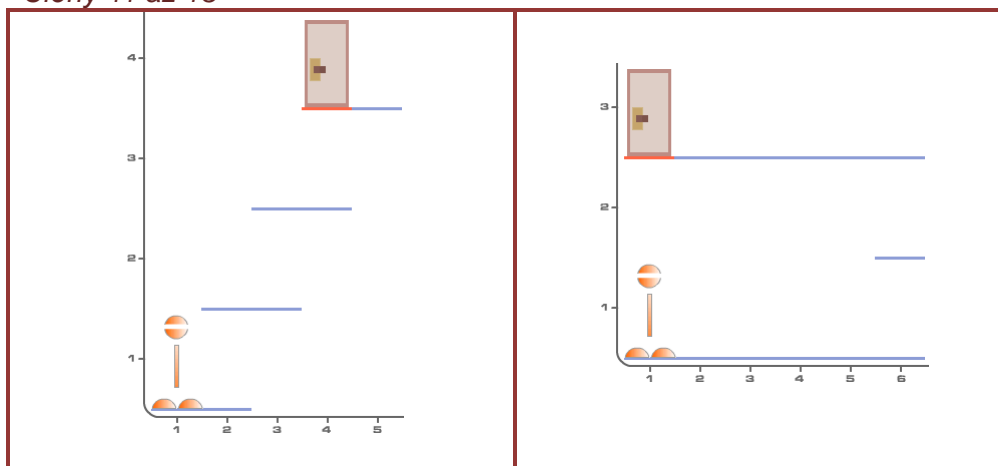
Podobnú úlohu ako úloha 10 sme riešili. Koľko riadkov programu budeme potrebovať? Máme ich toľko k dispozícii?

Na prejdenie príslušného počtu krokov môžeme využiť príkaz cyklu

- stlačte tlačidlo začiatku cyklu,
- upravte počet opakovaní cyklu,
- vložte príkazy, ktoré chcete, aby sa opakovali,
- stlačte tlačidlo konca cyklu.

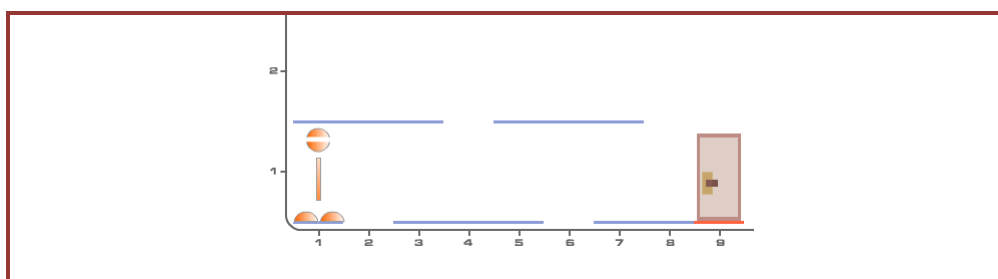
Príkazy v cykle vykoná panáčik príslušný počet krát.

Úlohy 11 až 13



Úloha 11

Úloha 12



Úloha 13

V miestnostiach v úlohách 11 až 13 nájdite opakujúce sa postupnosti a napíšte program, ktorý presunie panáčika ku dverám.

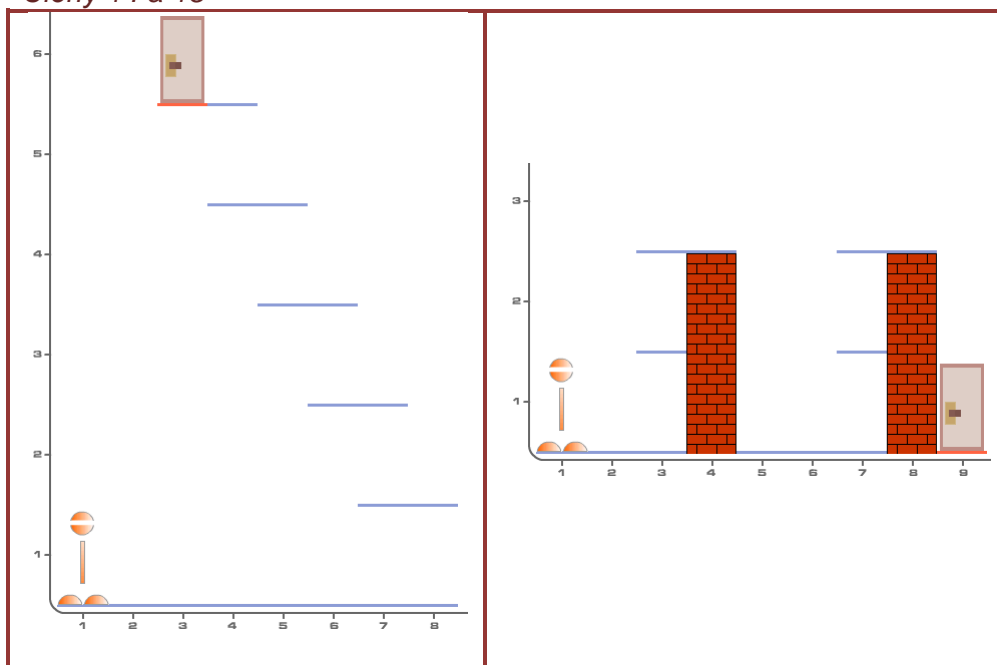
Zadanie 1

Pri každej úlohe uvažujte, koľko najmenej riadkov programu potrebujete.

Zadanie 3

Keby miestnosť v úlohe 13 vyzerala na prvých dvoch poschodiach rovnako, ale mala by podlahy aj na vyšších poschodiach, zmenilo by sa tým vaše riešenie?

Úlohy 14 a 15



Uvedomte si, že keď panáčik stojí na červenom múriku nemôže zoskočiť, ale musí vykročiť vpravo tam, kde už nie je pod ním podlaha.

Všimnite si, že ak panáčik stúpi mimo podlahy, začne padat' až k najbližšej nižšej podlahe. Panáčikovi opäť nevadí, že spadne aj o niekoľko poschodí.

V prostredí nájdete aj ďalšie úlohy, ktoré sme pripravili - úlohy 16 až 21. Keďže využívajú rovnaké princípy ako predchádzajúce úlohy, ich zadania, teda obrázky miestností, sme nedali do tohto materiálu.

Riešenia k nim si však môžete pozrieť na konci materiálu.

Zadanie 4

Riešte aj ďalšie úlohy - 16 až 21, ktoré nájdete v prostredí Panák.

Riešenie

Opäť v obrázku vyhľadávajte časti, ktoré sa opakujú, aby ste mohli využiť jeden, prípadne aj viac cyklov.

Zadanie 5

Uvažujte o tom, ktoré úlohy by ste mohli využiť na 1. stupni ZŠ.

Program IzyLogo.exe nájde v prostredí Moodle.



Pás s príkazmi

V prostredí budete riešiť aj ďalšie úlohy ako sú tie, ktoré uvádzame v materiáloch. Pomocou nich si precvičíte a pochopíte jednotlivé príkazy.

Sledujte, že autičko prechádza vždy do nasledujúceho mrežového bodu nakreslenej štvorčkovej siete.

Štvorčeky plochy, a teda aj celý obrázok, môžeme zväčšovať kolieskom myši.

↑	Dopredu	1
←	Vlavo	45
↑	Dopredu	1
←	Vlavo	90

Základné príkazy, z ktorých pripravujeme cestovanie včielky.

Pomocou šípok pri jednotlivých kartičkách môžeme zmeniť počet krokov alebo počet opakovaní.

Programovanie v prostredí Zjednodušené logo

Pre žiakov vo vyšších ročníkoch ZŠ sa často ako jeden z prvých programovacích jazykov využíva prostredie Imagine Logo, pozri [7] a [8]. V rámci nášho materiálu sme navrhli zjednodušený mikrosvet, ktorý využíva základné myšlienky jazyka Logo. Pomocou neho sa bližšie oboznámime so základnými princípmi programovania. Prostredie umožňuje riadiť objekt v priamom režime, ale dovoľí tiež zostavovať program pomocou pripravených kartičiek. Na samotnom tvare objektu pritom často nezáleží (niekedy ho budeme nazývať korytnačka, inokedy to bude autičko, či včielka). Od prostredí, s ktorými sme sa stretli doteraz sa tento mikrosvet líši predovšetkým tým, že dovoľí pripravovať a využívať **vlastné príkazy**, t.j. časti programov, ktoré riešia iba nejakú časť zložitejšieho zadania.

Mikrosvet umožňuje:

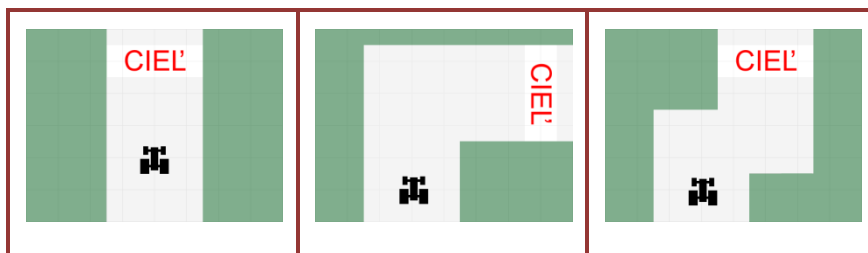
- riešiť sadu pripravených gradovaných úloh, ktoré sa postupne zobrazujú priamo v prostredí,
- riadiť jeden objekt - jeho pohyb do mrežových bodov a jeho otáčanie o násobky uhla 45° ,
- vytvárať program na riadenie objektu s využitím základných príkazov a aj vytvorením vlastných príkazov.

V prostredí nájde príkazy:

- na pohyb: dopredu, vľavo, vpravo,
- na nastavenie pera: nastav farbu, nastav hrúbku,
- pre vyplnenie nakreslenej oblasti,
- pre určenie náhodných vstupov pri kreslení a nastavení farby či hrúbky pera,
- na opakovanie skupiny príkazov (cyklus),
- na definovanie vlastných príkazov.

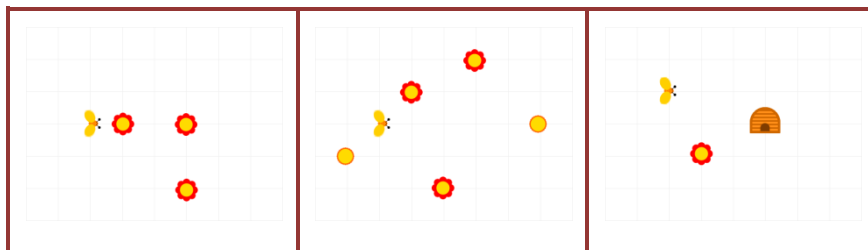
Pohyb v priamom režime

Klikaním na ikony s príkazmi v ľavom páse s príkazmi vyriešite úvodné úlohy, ktoré sú pripravené v rámci prostredia. Autičko sa pri klikaní na príkazy otáča alebo pohybuje podľa zvoleného príkazu vždy do nasledujúceho mrežového bodu.



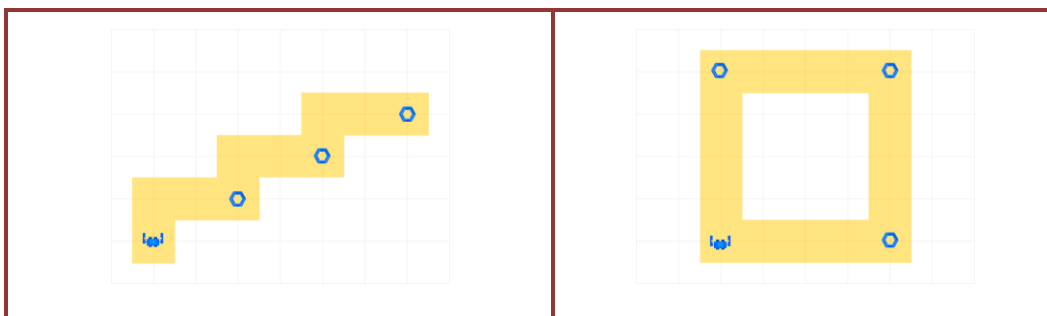
Hľadanie cesty

V nasledujúcich aktivitách je vašou úlohou prejsť včielkou cez všetky kvietky, ktoré sú na lúke. Nie však v priamom režime ako v minulých aktivitách, ale tým, že pripravíte **program**, ktorým včielka prejde a opelí všetky kvietky. Ak je na lúke aj úl, včielka má skončiť svoje putovanie v ňom.



Pri riešení zostavujte krátke programy. Sledujte pohyb a otáčanie včielky. Využívajte základné príkazy na pohyb a otáčanie vľavo alebo vpravo o 45° alebo 90° .

V nasledujúcich úlohách využite aj nový príkaz **opakuj**. V pripravených úlohách vyhľadajte časti, ktoré sa opakujú a navrhните príkazy na prechádzanie robota po vyznačenej cestičke.



V prostredí nájdete aj ďalšie úlohy, v ktorých si môžete vyskúšať hľadať cestu včielky po lúke s kvietkami, či robota, ktorý zbiera svoje súčiastky.

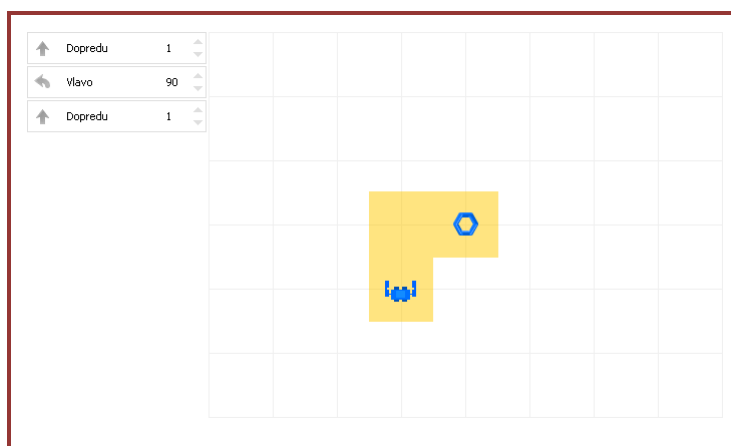
Pri zostavovaní programu ihneď vidíme, či sme úlohu vyriešili správne.

Zadanie 1

Uvažujte, či včielka vo vašom riešení putovala ku kvietkom najkratšou možnou cestou.

Opravte postupnosť príkazov

V nasledujúcom type gradovaných úloh je zostavený krátky program, v ktorom niekto omylom použil jeden alebo aj niekoľko nesprávnych príkazov a tak robot, včielka či autíčko neprechádzajú po vopred vyznačenej cestičke. Opravte chybné postupnosti príkazov v jednotlivých úlohách.



V prostredí nájdete aj ďalšie úlohy, v ktorých je potrebné opraviť už hotové postupnosti príkazov.

Ťahaním príkazov z ľavej časti nahradte niektoré kartičky v pripravenej postupnosti tak, aby program riešil pripravenú úlohu.

Uvedomte si, že chyba môže byť nielen v otáčaní robota, ale aj v nesprávnom počte krokov pre neho.

Zadanie 2

Je pre vás ľahšie upraviť cudzí program alebo navrhnúť svoje riešenie úlohy od začiatku?

Zamyslite sa

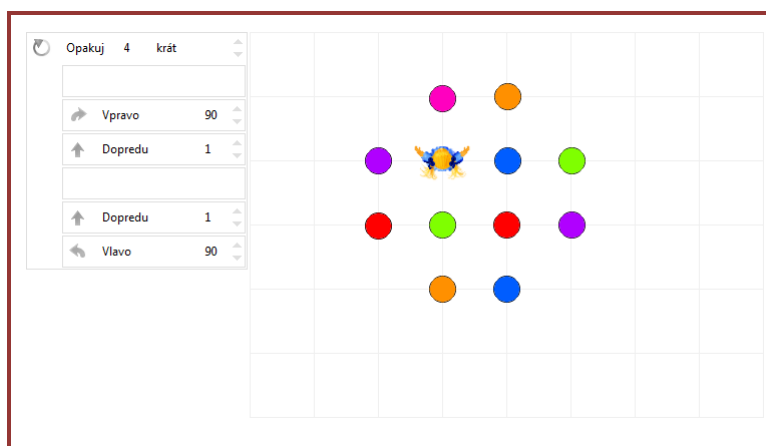
Prekvapilo vás elegantné riešenie niektorej úlohy?

Doplňte chýbajúci príkaz

V nasledujúcich úlohách je už vopred pripravená postupnosť príkazov na ich vyriešenie. Omylom však niektoré príkazy zmizli a ostali po nich len biele kartičky. Doplňte príkaz, ktorý patrí na bielu kartičku.

V tejto úlohe dievčatko stratilo svoje koráliky. Už sme aj pripravili cestu, na ktorej ich pozbiera, ale stratili sa nám z nej dva príkazy. Nájdite a doplňte ich.

Pripravte aj príkazy pre inú cestu, po ktorej môže prechádzať a pritom tiež pozbiera všetky koráliky.



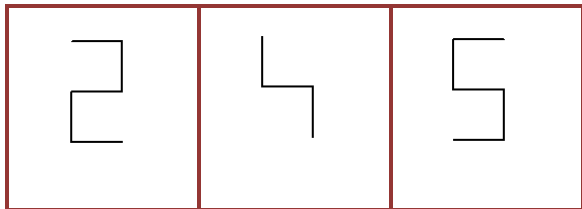
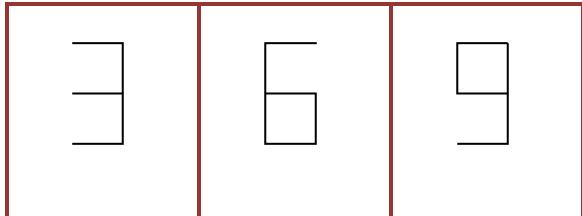
V prostredí nájdete aj ďalšie jednoduché alebo zložitejšie úlohy.

Kreslenie obrázkov čiarami

V nasledujúcich úlohách budeme pomocou čiar kresliť jednoduché obrázky.

Pri kreslení ihneď vidíte, či je obrázok správny.


Ak potrebujete, riešenie si môžete pozrieť na konci materiálov.

Zadanie 2	Nakreslite obrázky číslíc.  Všetky obrázky nakreslite tak, aby ste po žiadnej čiare neprešli viackrát.
Zadanie 3	Nakreslite aj niektoré z nasledujúcich obrázkov. Uvažujte, ktoré z nich sa dajú nakresliť tak, že po žiadnej čiare nepôjdete viackrát.  Pre ktoré z nich to nie je možné? Zdôvodnite.
Zadanie 4	Vymyslite niekoľko podobných obrázkov, ktoré pripomínajú písmená, rôzne predmety a pod. Pokúste sa pri ich kreslení nepoužiť veľa príkazov.

Navrhovanie vlastných príkazov

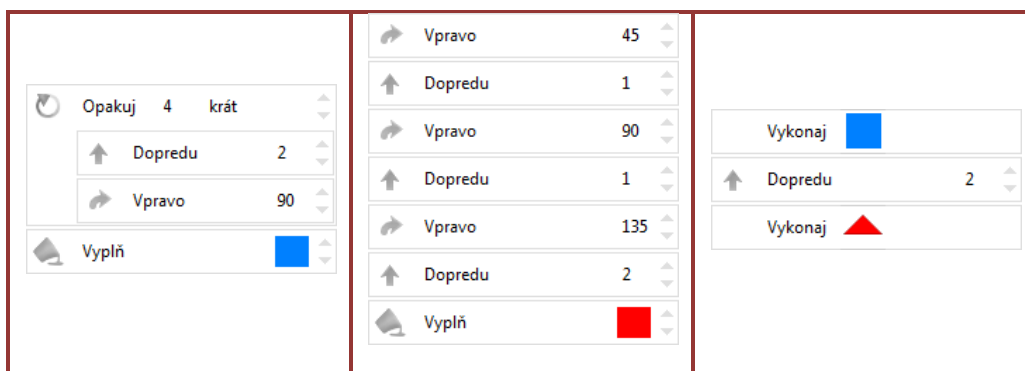
Ak nakreslíme niektorý obrázok, môžeme si ho uchovať ako hotový príkaz, ktorý budeme môcť využiť pri kreslení ďalších, zložitejších obrázkov.

Navrhujeme domček. Uvažujeme o dvoch útvaroch, z ktorých sa domček skladá - modrý štvorček a červený trojuholník.

Najprv nakreslíme modrý štvorček. Stlačením tlačidla  v ľavom páse uložíme tento obrázok ako nový príkaz. Ďalej navrhujeme príkazy, ktorými sa kreslí červený trojuholník. Príkaz opäť uložíme. Využijeme oba príkazy pri kreslení domčeka.



Domček



Nové príkazy sa objavujú v ľavom páse

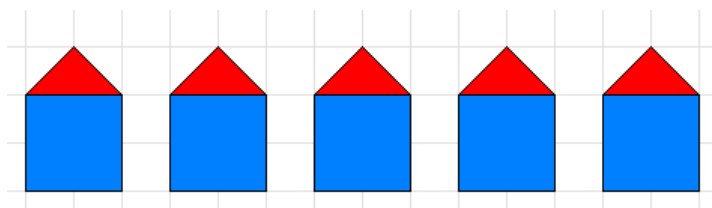
Aj príkaz na kreslenie domčeka si opäť uložíme, aby sme ho mohli použiť na nakreslenie ulice.

Pri navrhovaní každého príkazu si uvedomte, ktorým smerom je na začiatku korytnačka natočená.

Tiež je dôležité, v ktorom bode a ako natočená skončí kreslenie každého útvaru.

Zadanie 5

Nakreslite ulicu z piatich vedľa seba stojacich domov.

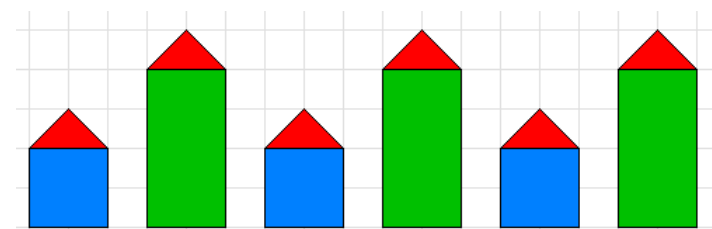


Zadanie 6

Podobne pripravte panelák - vyšší domček. Aj tento príkaz si uložte.

Zadanie 7

Kombináciou príkazov na kreslenie domčeka a paneláku nakreslite takúto ulicu.



Panelák

Na kreslenie strechy paneláku nezapomnite využiť už hotový príkaz, ktorý kreslí trojuholník.

Program zivyObraz.exe
nájdete v prostredí Moodle.

Vytváranie živých obrazov

V nasledujúcej časti materiálov spoznáme prostredie, v ktorom budeme vytvárať živé obrazy. Spolu budeme určovať, aké objekty dáme do živého obrazu a to, aké budú mať vlastnosti a ako sa budú správať ak nastane nejaká udalosť. Dozvieme sa, že objekty sa môžu hýbať aj naraz a pritom môžu reagovať ak na ne klikneme alebo ak sa stretnú s inými objektmi v prostredí.

Takýmto spôsobom vytvoríme

- neinteraktívne animácie, teda také, v ktorých sa budú objekty pohybovať istým spôsobom, ich činnosť nebudeme ovplyvňovať,
- interaktívne animácie, t.j. niektoré objekty v obraze budú reagovať na kliknutie, prípadne na zrážku s inými objektmi,
- skladačky, labyrinty, preteky.

V prostredí si všimnime nasledujúce časti

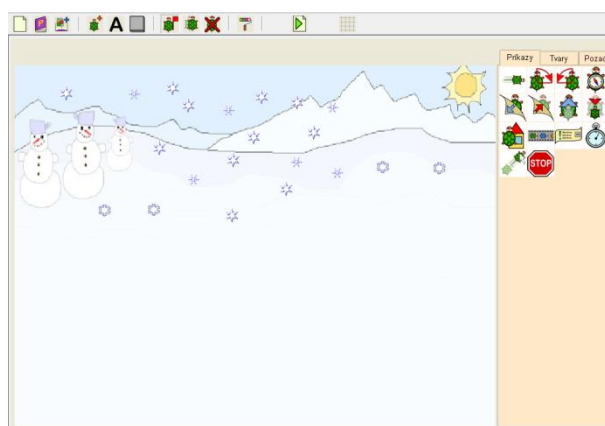
- pracovná plocha,
- vodorovný pás tlačidiel,
- tri záložky Príkazy, Tvary a Pozadia v pravej časti.

V pracovnej ploche môžeme meniť pozadie a vkladať do nej objekty.

Objekty môžeme riadiť priamo alebo programom pomocou príkazov zo záložky Príkazy.

Objektom môžeme vybrať tvary zo záložky Tvary.

Prostredie



Prostredie Živý obraz

Objekty, ich tvary a pohyb

Do prostredia môžeme vkladať:

- pozadie,
- objekty.

Objektom môžeme:

- meniť tvar,
- otáčať ich,
- pohybovať nimi.

Aktivita 1

Navrhňte nejaké pozadie.

Vložte do neho niekoľko objektov.

Zmeňte objektom tvar.

Smer každého objektu zmeňte tak, aby vyjadroval jeho skutočné natočenie.

Vyskúšajte pohyb každého objektu.

Vloženie pozadia

- zvolíme záložku **Pozadia** a vyberieme vhodné pozadie,
- kliknutím na obrázok v záložke sa tento obrázok vloží ako pozadie.

Vloženie objektu

- v hlavnej ponuke zvolíme **tlačidlo na vkladanie objektu**,
- kliknutím do pozadia vložíme nový objekt, ktorý má zatiaľ tvar korytnačky.

Tvar objektu

- zvolíme záložku **Tvary**, klikneme na niektorý z ponúkaných tvarov, objekt zmení tvar,
- pozor, ak máme na ploche niekoľko objektov, mení sa tvar toho, na ktorý sme naposledy klikli.

Natočenie objektu

- kliknutím na objekt vyberieme objekt, ktorého smer chceme meniť,
- v záložke **Príkazy** vyberme **príkaz na zmenu smeru**,
- uhol natočenia vyberáme v otvorenej **pomôcke** kliknutím do kruhu alebo tým, že priamo napíšeme nejaký uhol,
- natočenie sa udáva v stupňoch, 0 určuje smer hore, ostatné smery sú tak, ako sme zvyknutí z klasickej geometrie.

Pohyb objektu

- klikneme na objekt, ktorým chceme pohnúť,
- v záložke **Príkazy** vyberieme **príkaz pohybu**,
- otvorí sa pomôcka, ktorou môžeme určiť ako veľký krok má objekt urobiť,
- počet krokov vyberáme kliknutím do pravítka alebo napísaním čísla, ktoré udáva počet krokov.

Neinteraktívne animácie

Na internete sa čoraz častejšie objavujú obrázky, na ktorých sa niečo hýbe. V našich materiáloch ich budeme nazývať neinteraktívne animácie. Existuje niekoľko softvérov, v ktorých sa dajú takéto animácie pripravovať. V nasledujúcej časti materiálov pripravíme takéto animácie v prostredí Živý obraz.

Aktivita 2

Navrhnete neinteraktívnu animáciu, v ktorej bude padat sneh.

Popis aktivity a riešenie

Do pozadia plochy vložíme obrázok. Keďže budeme uvažovať o snežení, vhodná je zimná krajinka so snehuliakmi. Ostatné objekty na ploche budú vločky, ktoré sa pri spustení živého obrazu budú pohybovať. Tie nepatria pozadiu, pretože majú v rámci živého obrazu nejaké správanie - pohyb smerom dolu.

Pozadie a jeden objekt

- rovnakým spôsobom ako v Aktivite 1 pripravíme pozadie a tiež jeden objekt s tvarom vločky a smerom dolu.

Správanie objektu

- pre pripravený objekt-vločku navrhne jej správanie, ktoré bude mať po spustení živého obrazu,
- správanie objektu budeme navrhovať v novom okne, ktoré budeme nazývať **okno správania objektu**,
- toto okno otvoríme, keď klikneme na objekt **pravým tlačidlom**,

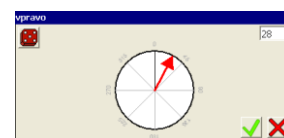
Kliknutím na obrázok pozadia sa tento obrázok vloží ako pozadie pracovnej plochy.



Tlačidlo na vkladanie objektu



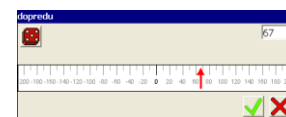
Ikona príkazu na zmenu smeru



Pomôcka na výber uhla



Ikona príkazu pohybu



Pomôcka na výber vzdialenosti

Všimnite si, že ak by sa mal objekt stratit' v pravej časti obrazovky, objaví sa vľavo.



Uvedomte si, že všetky vločky budú mať rovnaké správanie - budú natočené smerom nadol a po spustení živého obrazu budú padat smerom dolu.

Okno správania má niekoľko záložiek. Zatiaľ využívame záložku pri spustení, t.j. navrhujeme správanie objektu, ktoré sa bude vykonávať hneď po spustení živého obrazu.

Ak vyberieme nesprávny príkaz klikneme naň v okne správania pravým tlačidlom - príkaz sa zruší.

Ak chceme zrušiť niektorý objekt, vyberme tlačidlo na rušenie objektov a potom klikneme na objekt.



Tlačidlo na zrušenie objektu



Tlačidlá na spustenie a zastavenie živého obrazu

Všimnite si, že po zastavení živého obrazu sa vložka vrátila na svoje pôvodné miesto.



Tlačidlo na klonovanie objektu

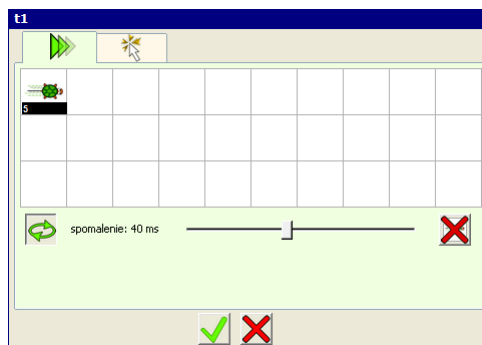
Ak pri stlačení tlačidla na klonovanie zároveň stlačíte tlačidlo Ctrl, môžete vkladať klony až kým znovu nestlačíte tlačidlo na klonovanie - takto môžete vložiť veľa vložiek bez toho, aby ste znovu museli určiť, ktorý objekt chcete klonovať.

Niektorým vložkám môžete zmeniť tvar tak, že na ne kliknete a v záložke tvary im vyberiete iný tvar.


Všimnite si, že pre objekt je dôležité správanie a nie jeho tvar.




Tlačidlo na uloženie projektu



Okno správania objektu

- v záložke **Príkazy** vyberme **príkaz pohybu**,
- v otvorenej **pomôcke** zvolíme dĺžku, o ktorú sa má vložka pohnúť,
- zatvoríme okno výberom tlačidla ,
- vyskúšame projekt tlačidlom na **spustenie živého obrazu**,
- po spustení sa vložka pohla o príslušný počet krokov,
- zastavíme projekt tlačidlom na **zastavenie živého obrazu**.

Nekonečný pohyb

- otvoríme opäť **okno správania objektu**,
- stlačíme tlačidlo donekonečna  - príkazy, ktoré sú určené v okne sa budú vykonávať stále, až kým nezastavíme živý obraz,
- určíme **spomalenie** - v pomôcke ťahajme posúvač,
- vyskúšajme projekt,
- zastavíme projekt a pokračujeme v jeho vývoji.

Klonovanie objektu

- ak projekt s jednou vložkou funguje správne, vložku naklonujeme na ďalšie miesta,
- vyberme tlačidlo na klonovanie objektu,
- klikneme na objekt, ktorý chceme klonovať,
- klikneme na nové miesto, kam chceme objekt s rovnakým správaním umiestniť,
- vyskúšajme projekt.

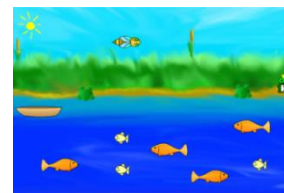
Aktivita 3	Do projektu vložte toľko vložiek, koľko potrebujete.
Aktivita 4	Upravte projekt tak, aby niektoré vložky pohybovali pomalšie.
Aktivita 5	Uvažujte o ďalších animáciách, ktoré by sa dali pripraviť v tomto prostredí.
Riešenie	Uvažujte napr. o prehliadači fotografií, ktoré sa pohyujú.
Aktivita 6	Uložte si projekt.
Aktivita 7 Diskusia	Aké sú výhody takto pripravených animácií oproti tým, ktoré ste vytvárali v niektorom grafickom editore?

Interaktívne animácie

V projekte sneženie sme na vločky po spustení živého obrazu už nemali žiadny vplyv - vločky len padali. Teraz pripravíme také živé obrazy, v ktorých správanie jednotlivých objektov budeme môcť ovplyvňovať aj počas behu živého obrazu a to kliknutím na ne. Pritom každý z objektov môže mať na kliknutie pripravenú inú reakciu - ryby sa otočia, žabka poskočí, slniečko sa skryje za obláčik...

Aktivita 8

Navrhnite živý obraz, v ktorom budú v rybníku plávať rybky, skákať žaby, lietať balón, atď.



Rybník s plávajúcimi rybkami a vyskakujúcimi žabkami

Popis aktivity a návrh riešenia

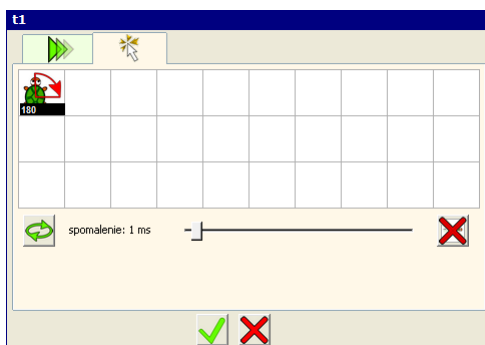
Tak ako v predchádzajúcich aktivitách najprv pripravíme **pozadie** a jeden **objekt**. Objektu:

- zmeníme **tvár** na obrázok ryby,
- zmeníme **smer** - aby plávala vodorovne,
- navrhne pohyb - v okne správania navrhne príkaz na nekonečný pohyb podobne ako v predchádzajúcej aktivite.

Správanie objektu pri kliknutí naň

Pri kliknutí na rybku potrebujeme, aby sa otočila do „protismeru“, t.j. o 180 stupňov.

Tie príkazy, ktoré má objekt vykonať pri kliknutí naň, vkladáme tiež v **okne správania**, avšak do záložky **pri kliknutí**:



Záložka pri kliknutí s príkazmi, ktoré objekt vykoná, keď naň klikneme



Ikona príkazu na otočenie

Uvedomte si, že v tomto projekte nepotrebujeme otočiť rybku do konkrétneho smeru, ale do „protismeru“, t.j. relatívne vzhľadom na jej súčasné natočenie.

V prípade potreby nájdete riešenia na konci materiálu.

Aktivita 9

Vložte do rybníka toľko rybiek, koľko potrebujete.

Riešenie

Využite klonovanie. Všimnite si, že pri klonovaní sa zachováva správanie ryby pri kliknutí.

Nezabudnite projekt vyskúšať. Poslúchajú rybky?

Aktivita 10

Vložte na breh rybníka nový objekt - žabku. Navrhnite jej správanie tak, aby pri kliknutí vyskočila a opäť sa vrátila na svoje miesto.



Ikona príkazu čakanie

Ak sa nám to hodí, objekt sa môže pri kliknutí skryť.



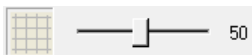
Ikona príkazu skry objekt.

Vždy najprv uvažujte, aké správanie pre objekt potrebujete. Až potom začnite hľadať, či prostredie Živý obraz takéto správanie objektov umožňuje.


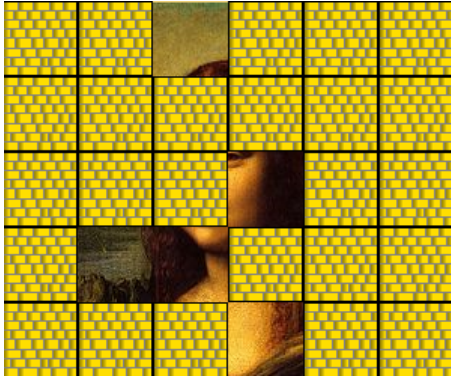
Hráč klikne na niektorý štvorček a odkryje sa mu časť obrázka. Vhodným výberom štvorčekov dokáže rýchlo uhádnuť, čo je na obrázku.

Najprv pripravte jeden štvorček s vhodným správaním (pri kliknutí sa štvorček skryje). Potom využite klonovanie, nezabudnite využiť kláves Ctrl.

Pri vytváraní hry využite sieť - štvorčeky, ktoré zakrývajú objekt budú pekne vyrovnané.



Tlačidlo siete a posúvač, ktorým sa dá vybrať hustota siete

Riešenie	<p>Žabka musí ísť trochu dopredu a potom sa vrátiť naspäť. Ak tieto príkazy dáme tesne za seba, stihne to urobiť tak rýchlo, že si to ani nevšimneme. Preto je potrebné medzi ne vložiť čakanie.</p>  <p>Reakcia žabky na kliknutie</p>
Aktivita 11	<p>Vložte do projektu ďalšie objekty. Navrhňte im nejaké správanie pri kliknutí.</p>
Aktivita 12	<p>Projekt si uložte.</p>
Aktivita 13	<p>Navrhňte iné interaktívne animácie.</p>
Riešenie	<p>Uvažujte napríklad pozadie s cestou, na ktorej sú domy a po ktorej chodia autá, bicykle a pes. Medzi pozadiami nájdete aj palmy a k nim pohyblivé objekty - domorodec, indián, opica, loďka.</p>
Aktivita 14	<p>Poznáte hru Háďaj, kto je to?</p>  <p>Háďaj, kto je na obrázku?</p> <p>Pozadie hry je fotka známej osobnosti, obraz, nejaký predmet... Táto fotka je zakrytá, štvorčekmi. Úlohou hráča je uhádnuť, kto je na obrázku odkrytím čo najmenšieho počtu štvorčekov. Keď hráč klikne na niektorý štvorček, tento sa mu odkryje, aby videl viac z fotky.</p> <p>Pripravte takúto hru.</p>

Skladačky

Ďalším typom aktivít, ktoré sa dajú pomocou prostredia Živý obraz pripraviť, sú skladačky. Rôzne typy tvári alebo iných obrázkov vytvoríme klikaním na ne a zmenou ich tvaru.

Aktivita 15

Navrhňte skladačku, v ktorej budeme môcť meniť snehuliakovi typ hrnca, očí, nosa, úst, metly a gombíkov.

Popis aktivity a návrh riešenia

Pripravme **pozadie** a **jeden objekt**, napr. s tvarom hrnca. Potrebovali by sme, aby hrniec (a neskôr aj nos, metla, či gombíky) reagoval na kliknutie takým spôsobom, aby svoj obrázok zmenil na hrniec, ktorý má iný tvar. Na takýto spôsob zmeny obrázka musí byť obrázok „pripravený“ a to tak, že má niekoľko tzv. záberov. Klikneme niekoľkokrát na ikonu **ďalší záber** a vyskúšajme, koľko záberov má hrniec.

Tento príkaz využijeme pri kliknutí - otvoríme **okno správaní** pre hrniec a do záložky pri kliknutí vložíme príkaz **ďalší záber**.

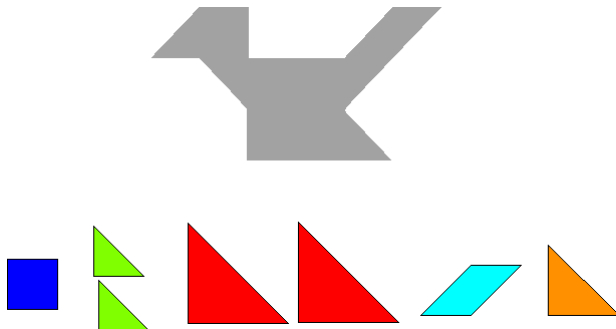


Príkaz pre hrniec, ktorý bude pri kliknutí meniť svoje zábery

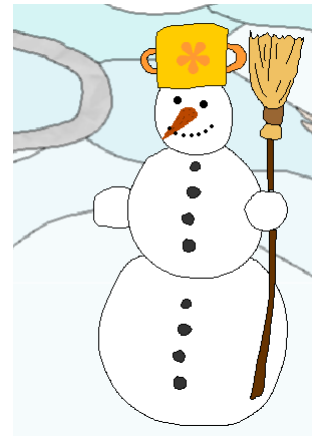
Projekt vyskúšajme a ak funguje, pripravme klonovaním aj ďalšie časti snehuliaka - oči, nos, metlu, šál, gombíky. Takto môže vzniknúť veselý alebo smutný snehuliáčik.

Aktivita 16

Navrhňte skladačku, tzv. tangram.



Tangram



Skladačka snehuliak



Ikona príkazu ďalší záber

Nezabudnite, že najvýhodnejšie je použiť klonovanie objektu, ktorý už má pripravené správanie.

Uvedomte si, že niekedy je správanie objektu dôležitejšie ako obrázok, ktorý objekt zobrazuje.

Viac o tangrame nájdete na webovej stránke [6].

Riešenie tejto aktivity si môžete pozrieť na konci materiálu.

Zrážky

Niekedy v projekte potrebujeme zistiť, či sa dva objekty stretli, teda narazili na seba. Navrhujeme takýto projekt: včielka lieta po lúke a keď nájde niektorý červený kvietok, zastane.

Kvietky, ktoré sú v zadnej časti obrázka môžete zmenšiť, aby sa zachovala perspektíva.



Ikona príkazu zmenši objekt.



Včielka na lúke

Pozadie a objekty s obrázkami kvietkov pripravíme rovnako, ako sme to robili v predchádzajúcich projektoch. Pripravíme aj objekt s obrázkom včielky.

Uvažujme, čo potrebujeme ďalej:

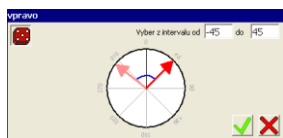
- včielka sa musí neustále pohybovať,
- keď sa stretne s niektorým kvietkom, zastane.

V okne správy včielky navrhujeme jej **nekonečný pohyb** rovnako ako sme to robili v prechádzajúcich aktivitách.

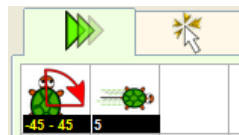
Aby sa však nehýbala stále rovnakým smerom, využijeme náhodné vstupy pre otočenie a pohyb:

- otvoríme **okno správy včielky**,
- vyberme príkaz **otočenie**,
- v pomôcke pre otočenie klikneme na kocku a zvolíme, ako sa má pri hľadaní kvietku otočiť,
- vyberme príkaz **dopredu**,
- v pomôcke pre príkaz dopredu stlačíme kocku a vyberme interval, z ktorého si bude vyberať, koľko krokov pôjde dopredu,
- v **okne správy** sa nám zobrazia vybrané voľby:

Pri kliknutí pravým tlačidlom si dajte pozor, aby ste klikli na včielku.



Pomôcka pre otočenie o náhodný uhol



Správanie včielky pri spustení živého obrazu

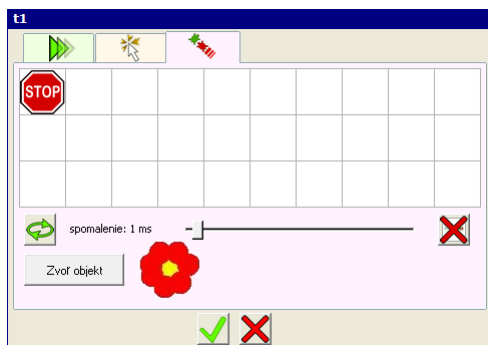
Ukončíme okno správy včielky a vyskúšajme projekt - včielka sa pohybuje.

Výberom červeného kvietka sme určili, že včielka bude rovnako reagovať na **všetky červené kvietky**.

Ďalej potrebujeme zistiť, či včielka našla niektorý kvietok. Na to využijeme v okne správy včielky záložku **pri zrážke**:

- klikneme pravým tlačidlom na včielku,
- vyberme záložku **pri zrážke**,
- stlačíme tlačidlo **Zvoľ korytnačku**,

- okno správania sa skrylo a našou úlohou je kliknúť na ten objekt, s ktorým sa má včielka zraziť,
- klikneme na ľubovoľný červený kvietok,
- týmto výberom sa vrátíme do okna správania,
- za tlačidlom Zvoľ objekt sa objaví kvietok,



Záložka pri zrážke včielky s kvetom

- vyberme ikonu stop zo záložky **Príkazy** - keď včielka stretne kvietok, zastane,
- vyskúšajme projekt - včielka lieta, keď sa dostane ku kvetu, zastane.

Aktivita 17	Na lúku pridajte ďalší typ kvetov. Upravte správanie včielky tak, aby reagovala aj na nové kvietky.
Aktivita 18	Navrhňte iný typ správania než zastavenie pohybu včielky, napr. rýchlo odletí preč od kvietku.
Aktivita 19	Pripravte projekt Preteky dinosaurov: <ul style="list-style-type: none"> • v ľavej časti plochy sa nachádza niekoľko dinosaurov, ktorí sa chystajú na preteky, • po spustení živého obrazu sa rozbehnú doprava, dbajte na to, aby nešli všetci rovnako rýchlo, • kto prvý prejde cieľovú čiaru vypíše, že vyhral a všetci zastanú.

Pri riešení tejto úlohy využijeme, že včielka môže reagovať na zrážku aj s viacerými objektmi.

Ak potrebujete pomoc, pozrite si riešenia na konci materiálu.



Ikona príkazu zobraz správu

Labyrinty

Ďalší typ aktivít, ktoré sa dajú v prostredí Živý obraz pripraviť sú labyrinty. Toto prostredie je jednoduchšie ako prostredie Mravec, s ktorým ste sa stretli v predchádzajúcom module. Učí nás však lepšie pochopiť, aké programátorské akcie sú ukryté pri príprave labyrintu.

Navrháme spolu jednoduchý labyrint a prechádzanie panáčka v ňom.

Steny labyrintu

Základom labyrintu je jeho tvar. Labyrint budeme zobrazovať pohľadom zhora podobne ako v prostredí Mravec. Navrháme nejaký labyrint:

- vytvoríme nový objekt, ktorý bude určovať stenu labyrintu,
- zvolíme mu tvar,
- klonovaním pripravíme a rozmiestnime steny tak, aby tvorili labyrint.

Aby boli steny labyrintu vyrovnané, zvolíme opäť tlačidlo pre štvorcovú sieť a prípadne zmeňme aj veľkosť siete, do ktorej budú jednotlivé objekty doskakovať.



Jednoduchý labyrint, ktorý pripravíme klonovaním jednej steny

Vďaka tomu, že máme nastavenú mriežku panáčik sa umiestni do stredu cestičky.

Pre panáčika vyberte taký tvar, na ktorom vidieť, ako je natočený.



Tlačidlo vkladania nových tlačidiel do projektu

Tlačidlom budeme ovládať objekt, na ktorý sme klikli tesne predtým, ako sme spustili živý obraz.

Obrázok na tlačidle môžeme zmeniť rovnako ako pre ľubovoľný objekt - v záložke Tvary klikneme na vhodný tvar, napr. šípku



Príkazy v okne správania tlačidla, pomocou ktorých budeme ovládať panáčika

Nezabudnite, že v záložke musíte mať vybraný príslušný objekt, pre ktorý pripravujete príkazy, t.j. v prvom prípade stenu, v druhom poklad.

Panáčik

Vytvoríme nový objekt, vložíme ho do vnútra labyrintu, vyberme mu tvar. Navrhujeme správanie pre panáčika. Je len na nás, akým spôsobom ho vyvedieme z labyrintu. Spomeňme niekoľko možností, ktoré máme pri prechádzaní labyrintu:

- panáčika budeme riadiť bez spustenia živého obrazu pomocou príkazov v záložke **Príkazy** a sami budeme strážiť, aby sme ho nevedli cez steny - toto je dosť zdĺhavý spôsob, pretože pri každom pohybe musíme vybrať počet krokov,
- panáčik bude prechádzať sám, musíme pre neho pripraviť správanie v situácii, keď by mal vkročiť do steny,
- panáčika budeme riadiť pomocou tlačidiel hore, dolu, vpravo, vľavo, sami budeme strážiť, aby sme neprechádzali cez steny,
- panáčika budeme riadiť pomocou tlačidiel hore, dolu, vpravo, vľavo, ale správanie panáčika upravíme tak, aby nechodil cez steny.

Ukážeme si riešenie popísané v poslednej možnosti. Postupne budeme riešiť nasledujúce problémy:

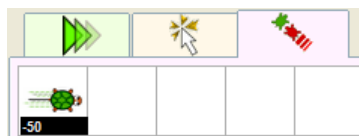
- pohyb panáčika všetkými štyrmi smermi: hore, dolu, vpravo, vľavo,
- zrážku panáčika so stenou,
- zrážku panáčika s pokladom.

Pohyb panáčika smerom hore pomocou tlačidla:

- vloženie tlačidla
 - výberom tlačidla pre vkladanie tlačidiel a následným kliknutím do plochy vložíme nové tlačidlo
- správanie tlačidla
 - pravým kliknutím na tlačidlo sa otvorí jeho okno správania,
 - pripravíme príkazy, ktoré otočia panáčika smerom hore a posunú o toľko krokov, ako široké sú steny labyrintu,
- podobne pripravíme tlačidlá aj pre pohyb panáčika ostatnými smermi.

Zrážka panáčika so stenou

- panáčik sa pri zrážke so stenou vráti naspäť na svoje pôvodné miesto
 - v okne správania panáčika v záložke pri zrážke navrhujeme príkaz



Správanie panáčika pri zrážke so stenou

- stlačíme tlačidlo **Zvoľ korytnačku** a vyberme niektorú stenu.

Zrážka panáčika s pokladom

- pri zrážke s pokladom vypíšeme správu a panáčika skryjeme



Správanie panáčika pri zrážke s pokladom

Aktivita 20

Navrhnete svoj vlastný labyrint, vložte do neho aj ďalšie objekty a nájdite poklad.

Čo sme sa naučili v tomto module

Čo sme sa naučili

Naučili sme sa riešiť úlohy pomocou vopred danej množiny príkazov. Pochopili sme, že pre úlohy existuje niekoľko riešení, pričom niektoré z nich sú efektívnejšie z pohľadu počtu použitých krokov.

Dokážeme tiež nakresliť obrázok pomocou príkazov čiarovej grafiky a vopred pripravenej množiny príkazov.

V prostredí Živý obraz sme videli, že správanie objektov môžeme riadiť pomocou udalostí - pri kliknutí, pri zrážke. Vieme tiež navrhnúť vlastné projekty v prostredí Živý obraz.

Uvedomujeme si možnosti, ale aj obmedzenia takto pripravených prostredí.

Preverenie výstupných vedomostí

Vedomosti, ktoré účastníci vzdelávania nadobudnú v tomto module budú preverované priebežným pozorovaním účastníka. Lektor sleduje, ako účastníci riešia jednotlivé úlohy a ako sa zapájajú do diskusií.

Absolvovanie predmetu Riešenie problémov a základy programovania

Za úspešného absolventa predmetu sa pokladá učiteľ, ktorý aktívne pracoval v pripravených prostrediach. Sám navrhol a spracoval niekoľko aktivít v prostrediach Kartičkové aktivity, Mravec, vyriešil úlohy v prostrediach Panák, Zjednodušené Logo a Živý obraz.

Na úspešné absolvovanie predmetu je dôležité, aby mal účastník v prostredí Moodle odovzdané súbory, ktoré dostal na vypracovanie od lektorov kurzu a ktoré dokladujú jeho aktívnu prácu v spomínaných prostrediach.

Literatúra a použité zdroje

- [1] Tomcsányiová, M. a kol.: Riešenie problémov a základy programovania 1, 32 s., Zvolen, Bratia Sabovci, 2009, ISBN 978-80-8118-023-1.
- [2] <http://www.int-edu.ru/logo/products.html>
- [3] <http://eurologo.web.elte.hu/lectures/kirill.htm>
- [4] <http://www.microworlds.com/solutions/mwjunior.html>
- [5] Gregor, O.: Semestrálny projekt Panák, FMFI UK, 2003
- [6] <http://en.wikipedia.org/wiki/Tangram>
- [7] Blaho, A., Kalaš, I.: Tvorivá informatika. 1. zošit z programovania. Bratislava : SPN - Mladé letá, 2007. 48 s. ISBN 80-10-01223-7.
- [8] Imagine Logo na webových stránkach projektu Infovek [online]. Dostupné na internete: <http://imagine.infovek.sk/>

Riešenia úloh z prostredia Panák

Úlohy 1 až 5

<p>Úloha 1</p> <pre> Program : - ● → vpravo </pre>	<p>Úloha 2</p> <pre> Program : - ● → vpravo - ● → vpravo </pre>	<p>Úloha 3</p> <pre> Program : - ● ↑ vyskoč </pre>	<p>Úloha 4</p> <pre> Program : - ● → vpravo - ● ↑ vyskoč </pre>	<p>Úloha 5</p> <pre> Program : - ● → vpravo - ● ↑ vyskoč - ● ↑ vyskoč - ● ↓ vľavo </pre>
--	---	--	---	--

Úlohy 6 až 9

<p>Úloha 6</p> <pre> Program : - ● ↑ vyskoč - ● → vpravo - ● → vpravo - ● ↓ zoskoč </pre>	<p>Úloha 7</p> <pre> Program : - ● → vpravo - ● → vpravo - ● ↑ vyskoč - ● ↑ vyskoč - ● ↓ vľavo - ● ↓ vľavo - ● ↓ zoskoč </pre>	<p>Úloha 8</p> <pre> Program : - ● → vpravo - ● → vpravo - ● ↑ vyskoč - ● ↑ vyskoč - ● ↑ vyskoč - ● ↑ vyskoč - ● ↓ vľavo - ● ↓ vľavo - ● ↓ zoskoč </pre>	<p>Úloha 9</p> <pre> Program : - ● ↑ vyskoč - ● ↑ vyskoč - ● → vpravo - ● → vpravo - ● ↓ zoskoč - ● → vpravo - ● → vpravo - ● ↑ vyskoč </pre>
---	--	--	---

V úlohe 9 existuje aj iné riešenie, ktoré potrebuje menej riadkov programu. Našli ste ho?

Aká vlastnosť správania panáčika nám umožnila skrátiť počet riadkov programu?

Úloha 10

```

Program :
- ● ↻ cyklus A 8x
- ● → vpravo
- ● ✘ koniec cyklu A
    
```

Úlohy 11 až 13

<p>Úloha 11</p> <pre> Program : - ● ↻ cyklus A 3x - ● → vpravo - ● ↑ vyskoč - ● ✘ koniec cyklu A </pre>	<p>Úloha 12</p> <pre> Program : - ● ↻ cyklus A 5x - ● → vpravo - ● ✘ koniec cyklu A - ● ↑ vyskoč - ● ↑ vyskoč - ● ↻ cyklus B 5x - ● ↓ vľavo - ● ✘ koniec cyklu B </pre>	<p>Úloha 13</p> <pre> Program : - ● ↻ cyklus A 2x - ● ↑ vyskoč - ● → vpravo - ● → vpravo - ● ↓ zoskoč - ● → vpravo - ● → vpravo - ● ✘ koniec cyklu A </pre>
---	---	---

Úlohy 14 a 15

Úloha 14	Úloha 15
<p>Program :</p> <ul style="list-style-type: none"> ○ ↻ cyklus A 7x ○ → vpravo ○ ✘ koniec cyklu A ○ ↻ cyklus B 5x ○ ↑ vyskoč ○ ← vľavo ○ ✘ koniec cyklu B 	<p>Program :</p> <ul style="list-style-type: none"> ○ ↻ cyklus A 2x ○ → vpravo ○ → vpravo ○ ↑ vyskoč ○ ↑ vyskoč ○ → vpravo ○ → vpravo ○ ✘ koniec cyklu A

Ak pri riešení úlohy 15 využijeme to, že vykonávanie programu sa skončí, keď sa panáčik dostane až ku dverám, môžeme ešte ušetriť jeden riadok programu. Podarilo sa to aj Vám?

Úlohy 16 až 18 (iba v prostredí)

Úloha 16	Úloha 17	Úloha 18
<p>Program :</p> <ul style="list-style-type: none"> ○ ↻ cyklus A 4x ○ ↑ vyskoč ○ ✘ koniec cyklu A ○ ↻ cyklus B 3x ○ ↑ vyskoč ○ → vpravo ○ → vpravo ○ → vpravo ○ ↓ zoskoč ○ → vpravo ○ → vpravo ○ ✘ koniec cyklu B 	<p>Program :</p> <ul style="list-style-type: none"> ○ ↻ cyklus A 3x ○ → vpravo ○ → vpravo ○ → vpravo ○ ↑ vyskoč ○ ↑ vyskoč ○ ✘ koniec cyklu A 	<p>Program :</p> <ul style="list-style-type: none"> ○ ↻ cyklus A 10x ○ → vpravo ○ ✘ koniec cyklu A ○ ↑ vyskoč ○ ↑ vyskoč ○ ← vľavo ○ ← vľavo ○ ← vľavo ○ ↻ cyklus B 3x ○ ↑ vyskoč ○ ← vľavo ○ ✘ koniec cyklu B ○ ↻ cyklus C 4x ○ ← vľavo ○ ✘ koniec cyklu C

Úlohy 16 až 21 nájdete iba v prostredí. Keďže využívajú rovnaké princípy ako ostatné úlohy, ich zadanie sme nedali do tohto materiálu.

V úlohe 16 sme dosiahli kratšie riešenie vďaka tomu, že si panáčik vyskočil a potom dopadol na to isté poschodie. V postate urobil jeden krok navyše, ale zápis riešenia sa skrátil.

V úlohe 17 sme využili, že ak panáčik nedokáže posledný príkaz vykonať, t.j. nemá kam vyskočiť, skončí vykonávanie programu.

Úloha 19 až 21 (iba v prostredí)

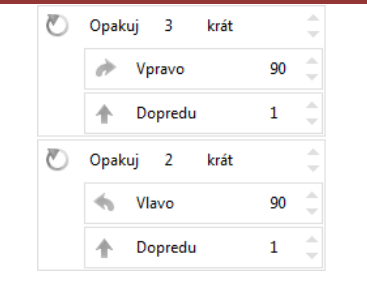

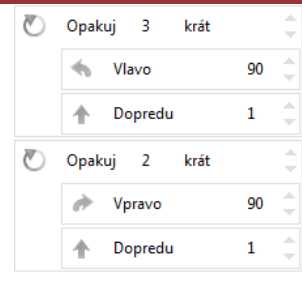
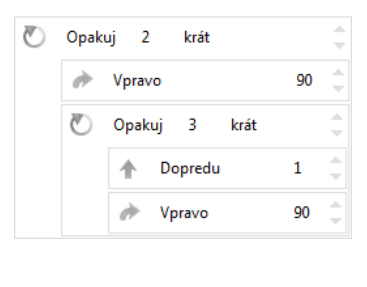
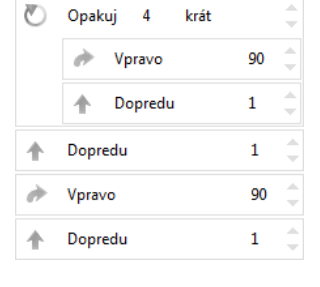
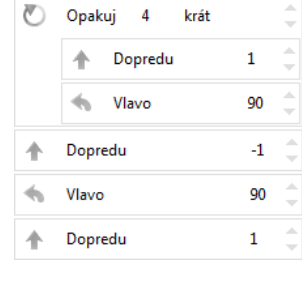
Úloha 19	Úloha 20	Úloha 21
<p>Program :</p> <ul style="list-style-type: none"> ○ ↻ cyklus A 3x ○ ↑ vyskoč ○ ↑ vyskoč ○ ↑ vyskoč ○ ↻ cyklus B 5x ○ → vpravo ○ ✘ koniec cyklu B ○ ↓ zoskoč ○ ✘ koniec cyklu A 	<p>Program :</p> <ul style="list-style-type: none"> ○ ↻ cyklus A 5x ○ ↑ vyskoč ○ ✘ koniec cyklu A ○ ↻ cyklus B 11x ○ → vpravo ○ ✘ koniec cyklu B 	<p>Program :</p> <ul style="list-style-type: none"> ○ → vpravo ○ ↻ cyklus A 3x ○ ↑ vyskoč ○ → vpravo ○ ✘ koniec cyklu A ○ ↻ cyklus B 7x ○ → vpravo ○ ✘ koniec cyklu B

Miestnosť v úlohe 21 vyzerá tak, ako by bolo potrebné zopakovať dvakrát vyjdenie na najvyššie poschodie. Keď však uvážime spôsob prechádzania panáčika v prípade, ak pod ním nie je podlaha, nájdeme jednoduchšie riešenie bez použitia príkazu zoskoč.

Riešenie úloh z prostredia Zjednodušené Logo

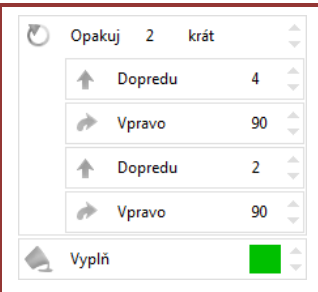

Riešenie zadaní 9 a 10

Všimnite si, že číslica 5 sa kreslí rovnako ako číslica 2, len kartičky, v ktorých sme sa otáčali vpravo o 90 sme vymenili za kartičky s otáčaním vpravo o 90.

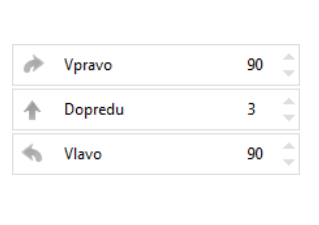
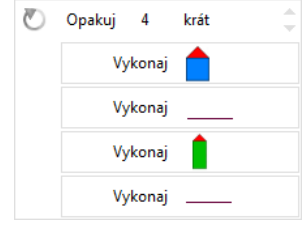
Riešenie zadania 13

Pri kreslení domčeka aj paneláku je výhodné kreslenie začať aj skončiť v rovnakej pozícii a natočení korytnačky.

	
--	---

Riešenie zadania 14

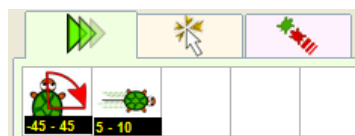
Aby sme si zjednodušili riešenie nakreslenia ulice z panelákov a domov vytvoríme pomocný príkaz, ktorým prejdeme vzdialenosť medzi domami.

	
---	--

Riešenia úloh z prostredia Živý obraz

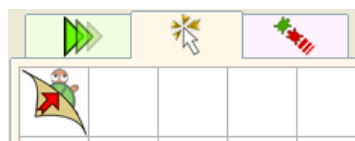
Aktivita 11 – Ďalšie objekty v rybníku

- nad rybníkom bude lietať včielka, ktorú budeme chytať klikaním myšou na ňu,
- nový objekt vložíme výberom tlačidla pre vkladanie objektov a následným kliknutím do stránky,
- novému objektu zmeníme tvar, vyberme napr. na včielku,
- v okne správania včielky vyberme záložku **pri spustení** a vyberme príkazy natočenie a posunutie o nejakú vzdialenosť,



Otočenie a presunutie včielky.

- včielky budeme chytať tak, že na ne klikneme,
- v okne správania včielky v záložke **pri kliknutí** vyberme príkaz na ukrytie objektu.



Ukrytie objektu pri kliknutí

Aktivita 14 – Hádaj, kto je to?

Vložme pozadie

- v záložke pozadia vyberme vhodné pozadie s nejakým obrázkom.

Prípravme jeden štvorček, ktorý bude zakrývať časť obrázka

- vytvoríme objekt - pomocou tlačidla nový objekt a následného kliknutia do plochy,
- zmeníme objektu tvar na štvorček, ktorý bude zakrývať časť obrázka - v záložke Tvary vyberme vhodný tvar.

Správanie štvorčeka pri kliknutí

- zmeníme správanie objektu - klikneme pravým tlačidlom na objekt, otvorí sa nám okno správania objektu, vyberme záložku pri kliknutí a vložíme príkaz na skrytie objektu pri kliknutí naň rovnako ako v Aktivite 11.

Ďalšie štvorčeky s rovnakým správaním

- naklonujeme štvorček spolu s jeho správaním
 - zapneme mriežku, aby sme štvorčeky rozmiestňovali pravidelne, ak je potrebné, môžeme zmeniť veľkosť mriežky,
 - klikneme na tlačidlo klonovania, potom klikneme na objekt-štvorček, ktorý chceme klonovať a stlačíme tlačidlo Ctrl,
 - klikajme do mriežky - na ploche vznikajú nové a nové štvorčeky,
- vyskúšajme projekt - pri kliknutí na štvorček sa ukáže časť obrázka pod ním,

Využitie projektu

- takýto projekt môžete pripraviť pre kolegu pri vedľajšom počítači a potom si s ním vymeniť miesto, aby skúsil uhádnuť, čo sa skrýva na vašom obrázku.

Ak do priečinku pozadia preniesete vlastné pozadia ako bmp, jpg a gif súbory, budú sa zobrazovať v záložke a tiež sa budú dať vyberať ako pozadia aktivít.



Tlačidlo na klonovanie objektov



Mriežka a jej veľkosť



Tangram

Dieliky tangramu sa budú otáčať vďaka tomu, že majú pripravených niekoľko záberov.



Ikona príkazu ďalší záber







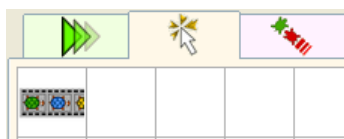
Ikona príkazu povol' ťahanie







Dieliky tangramu

Aktivita 16 – Tangram

- v záložke **Pozadia** vyberme niektoré pozadie, ktoré je pripravené pre tangram,
- pripravme jeden dielik tangramu
 - vyberme tlačidlo na **vkladanie objektu** - ,
 - klikneme do plochy - na to miesto plochy sa vloží objekt s obrázkom korytnačky,
- vyberme objektu **tvar**
 - klikneme na záložku **Tvary**,
 - klikneme na tvar oranžového trojuholníka,
 - tento tvar sa priradí objektu.
- **otáčanie** tvaru pri kliknutí
 - klikneme pravým tlačidlom na objekt,
 - otvorí sa okno správaní objektu,
 - vyberme záložku pri kliknutí  a v záložke Príkazy zvolíme príkaz ďalší záber ,
 - ukončíme okno správaní tlačidlom 



Pri kliknutí prejde dielik tangramu na ďalší záber

- **ťahanie** tvaru
 - v tomto projekte sa musí dať tvar ťahať po ploche, aby sme ho mohli umiestňovať do obrázka v pozadí,
 - v okne správaní objektu v záložke pri spustení vyberme príkaz **povol' ťahanie**.
- vyskúšajme projekt stlačením ; klikaním na objekt, sa trojuholník otáča, ťahaním ho môžeme umiestniť do obrázku,
- zastavme projekt stlačením  a pokračujme v jeho vytváraní pridaním ďalších dielikov,
- **ďalší dielik** vytvoríme klonovaním prvého dieliku
 - vyberme tlačidlo  a klikneme na oranžový trojuholník, ktorý už máme na ploche,
 - klikneme vedľa neho do plochy, vznikne tam rovnaký trojuholník,
- novému objektu zmeníme **tvar**
 - klikneme na záložku **Tvary**,
 - klikneme na tvar modrého štvorca,
 - tento tvar sa priradí objektu.
- opäť vyskúšajme projekt stlačením ; oranžový trojuholník aj modrý štvorec sa pri klikaní otáčajú, tiež sa dajú ťahať po ploche,
- rovnakým spôsobom - **klonovaním** a výberom ďalších tvarov pridáme všetky tvary tangramu - všimnite si, že v tangrame sú dva zelené a aj dva červené trojuholníky.

Aktivita 18 – Včielka a biely kvet

- pridajme nový objekt, vyberme mu tvar,
- pridajme včielke správanie pri zrážke s bielym kvietkom:
 - v okne správania vyberme záložku pri zrážke,
 - vyberme tlačidlo **Zvol' objekt**
 - kliknime na biely kvietok,
 - zo záložky **Príkazy** vyberme otočenie smerom od kvetu a prejdenie nejakej vzdialenosti,



Včielka sa pri stretnutí s bielym kvietkom otočí smerom od neho a prejde 30 krokov.

Aktivita 19 – Preteky dinosaurov

Podobne ako v predchádzajúcich aktivitách vyberme pozadie a vložme na plochu jeden objekt. Otočme objekt tak, aby bol natočený vpravo. Vyberme objektu tvar - z pripravených tvarov zvolíme dinosaura. Rovnako ako v projekte rybník pripravme správanie dinosaura - jeho pohyb. Vyskúšajme projekt.

Zamyslime sa nad tým, čo v projekte ešte potrebujeme - musíme zabezpečiť, aby sme vedeli skontrolovať, či sa dinosaur dostal do cieľa:

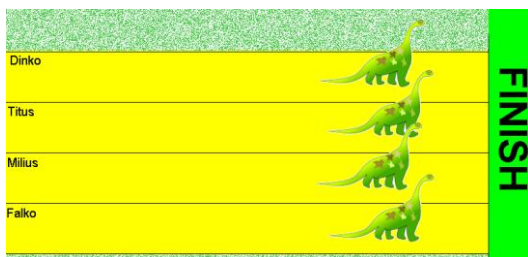
- vložme nový objekt, vyberme mu tvar cieľa - cieľ môže vyzerat' tak, že je to iba jednoduchá čiara alebo môžeme využiť širší obrázok,
- zmeňme dinosaurovi správanie **pri zrážke** s cieľom, v okne správania zvolíme objekt s obrázkom cieľa a navrháme správanie dinosaura, napr. vypíše správu, že vyhral a ukončí preteky.

Vyskúšajme projekt.

Naklonovaním dinosaura pripravme ďalších dinosaurov-pretekárov.

Čo musíme zmeniť, aby každý dinosaur vypísal svoje meno, keď vyhrá?

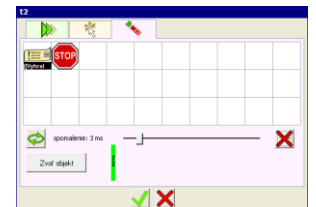
- novým dinosaurov, ktorých sme získali klonovaním, musíme upraviť príkaz pri zrážke a zmeniť text v dialógu, ktorý vypisujú pri zrážke objektom, ktorý zobrazuje cieľ.



Hotový projekt Preteky dinosaurov - kto bude prvý?

Uvažujte nad tým, či by bolo vhodné dať objektu konštantný krok, t.j. napr. 10 krokov.

Upravte pohyb tak, aby bol podľa vás spojitý, upravte mu dĺžku kroku a jeho spomalenie.



Stretnutie dinosaura s cieľom

Uvedomte si, čo by sa stalo, ak by sme všetkým dinosaurov dovolili pokračovať v pretekoch a prejsť až k cieľu.

Tento študijný materiál vznikol ako súčasť národného projektu Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika v rámci Aktivity „Vzdelávanie učiteľov 1. stupňa ZŠ na informatiku a informatickú výchovu“.

Autori © PaedDr. Monika Tomcsányiová, PhD.
RNDr. Ľubomír Salanci, PhD.
PaedDr. Daniela Bezáková, PhD.
RNDr. Andrej Blaho
Mgr. Daniela Onačilová

Názov Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika
Podnázov Riešenie problémov a základy programovania 2

Študijný materiál prešiel recenzným pokračovaním.

Recenzenti RNDr. Slávka Blichová
RNDr. Gabriela Lovászová, PhD.

Počet strán 28

Náklad 449 ks

Prvé vydanie, Bratislava 2010

Všetky práva vyhradené.

Toto dielo ani žiadnu jeho časť nemožno reprodukovat' bez súhlasu majiteľa práv.

Vydal Štátny pedagogický ústav, Pluhová 8, 830 00 Bratislava, v súčinnosti s Univerzitou Pavla Jozefa Šafárika v Košiciach, Univerzitou Komenského v Bratislave, Univerzitou Konštantína Filozofa v Nitre, Univerzitou Mateja Bela v Banskej Bystrici a Žilinskou univerzitou v Žiline

Vytlačil BRATIA SABOVCI, s r.o., Zvolen

ISBN 978-80-8118-029-3