

# Individuálny algoritmický jazyk ako prirodzená forma vyjadrovania sa študentov

Ján Guniš

# Algoritmy a programovanie

- najťažšie témy školskej informatiky
- história
  - dôraz na programovanie
  - aplikačný softvér
  - rozumná miera algoritmizácie a programovania v školskej informatike

# Algoritmy okolo nás

- vykonávame a realizujeme množstvo algoritmov
- vytvárame a overujeme správnosť algoritmov
- => algoritmy takmer nevnímame, prirodzená súčasť života
- prečo spôsobujú toľko problémov študentom?

# Výučba alg. a programovania

- podľa časovej dotácie
  - vykonávatele algoritmov
  - detské programovacie jazyky/prostredia
  - vyššie programovacie jazyky
- podstata je tvorba algoritmov a rozvoj algoritmického myslenia
- nie nutne najmodernejší jazyk
- ktorý je vhodný jazyk?

# Programovacie prostredia

- čo robí programovanie ťažkým?
- prostredia reflektujúce na odpovede, ale ideálne prostredie neexistuje
- Karel – obmedzené prostriedky a konštrukcie
- Logo – odlišné od iných programovacích jazykov
- Pascal – nemoderné, DOS režim
- Delphi – komplikované, neprehľadné

# Programovací jazyk

- odlišný od prirodzeného jazyka  
=> časť problémov študentov
- konštrukcie a prostriedky prešli vývojom  
(optimalizácia)
- iné chápanie syntaxe  
     $axb \leftrightarrow a*b$   
     $a=a+1$  neriešiteľné  $\leftrightarrow$  priradenie, test rovnosti  
    if  $\leftrightarrow$  chápaný ako cyklus  
    ...

# Ciele experimentu

- aký je prirodzený alg. jazyk?
- aké konštrukcie študenti objavia?
- ako budú zapisovať algoritmy?

# Experiment

- 1+8 študentov, 2. ročník, zameranie informatika
- navrhnuť „programovací“ jazyk a zapísať pomocou neho algoritmy riešiace konkrétne problémy
- potreba riešiť problém => jazyk
- IPOS problém - algoritmus jazyk program – riešenie problému



# Experiment – možné problémy

- výber vykonávateľa ovplyvní výsledný jazyk
- riešenie nebudú originálne (iné zdroje)
- študenti budú navzájom komunikovať (skupinové riešenie)
- môžu vzniknúť špecifické, všeobecne nepoužiteľné príkazy

# Očakávania od experimentu

Odpovede na otázky:

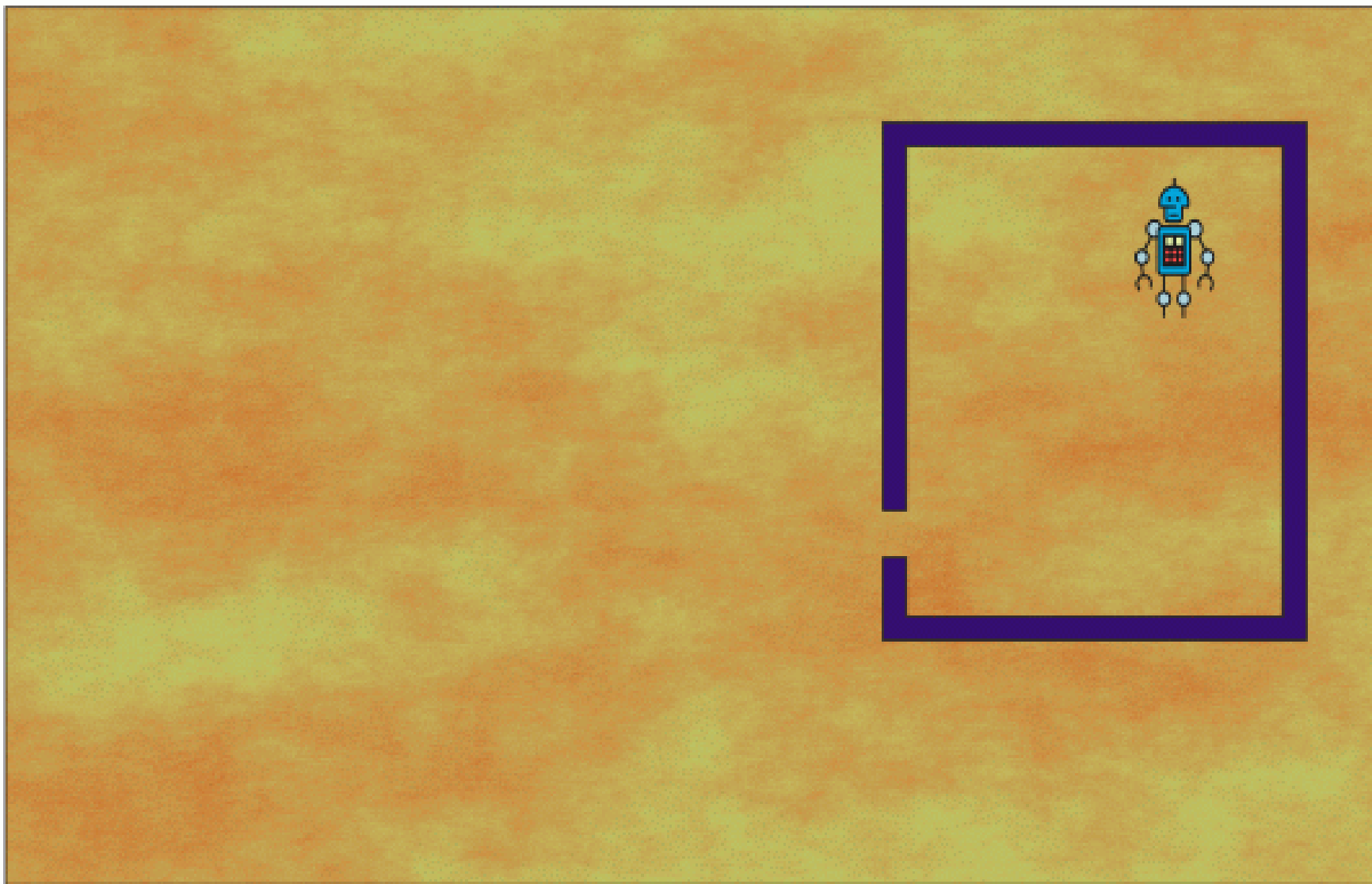
- Do akej miery dokážu študenti riešenie problému rozložiť na postupnosť elementárnych krokov?
- Aké jazykové konštrukcie a aký spôsob vyjadrovania algoritmov sú pre študentov prirodzené?
- Ak študent sám navrhne „programovací“ jazyk, môže mu to pomôcť lepšie pochopiť a využívať konštrukcie cudzieho programovacieho jazyka?
- Nájdem odpoveď (aspoň čiastočne) na otázku „Prečo je programovanie pre študentov také ťažké?“?

# Priebeh a realizácia

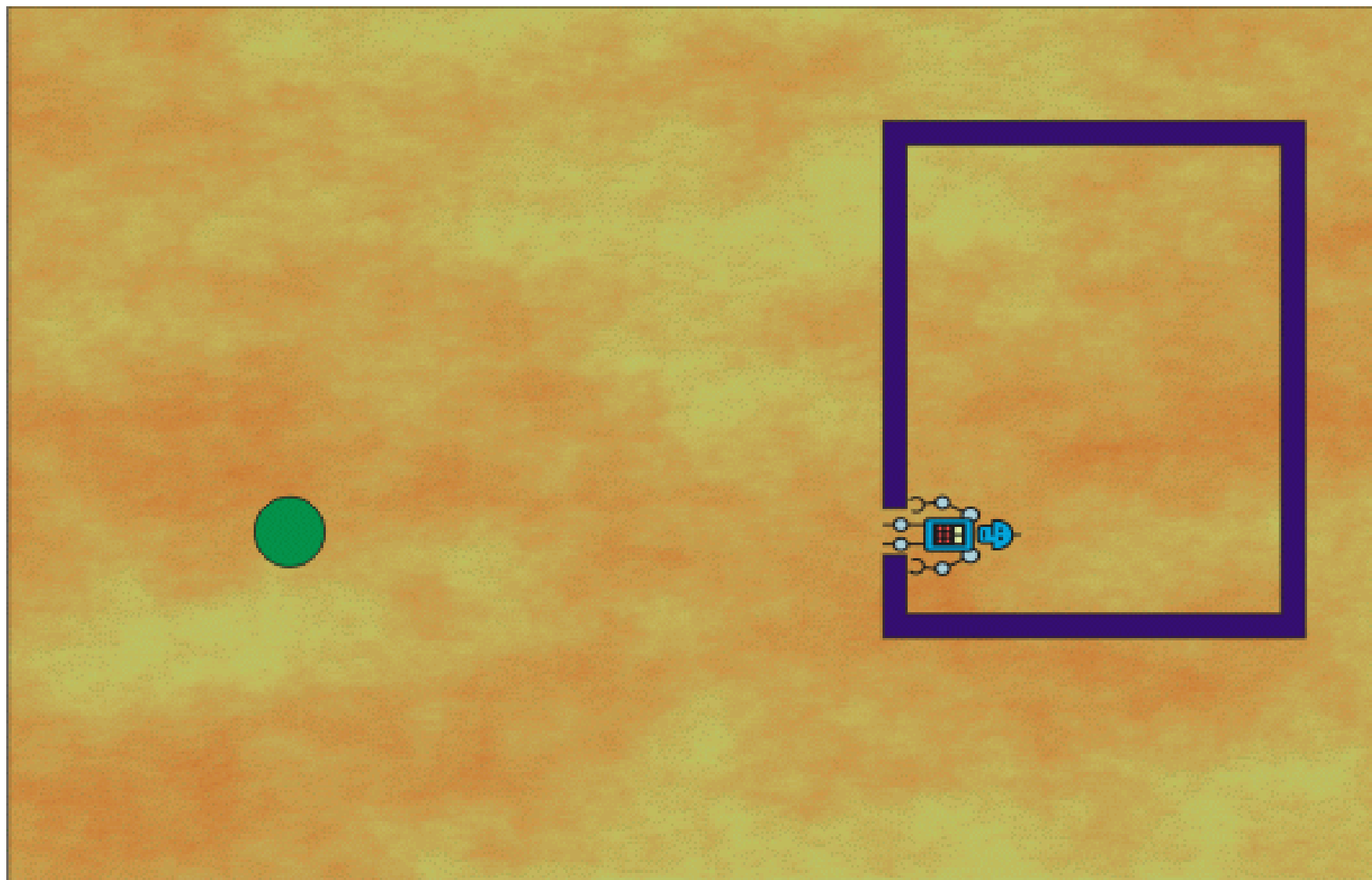
- mimo vyučovacích hodín
- prostredie Moodle => textová komunikácia  
=> stručnosť
- riešenie úlohy – komentár k riešeniu
- „nutkanie“ korigovať neštandardné riešenia

# Úlohy pre IPOS-a

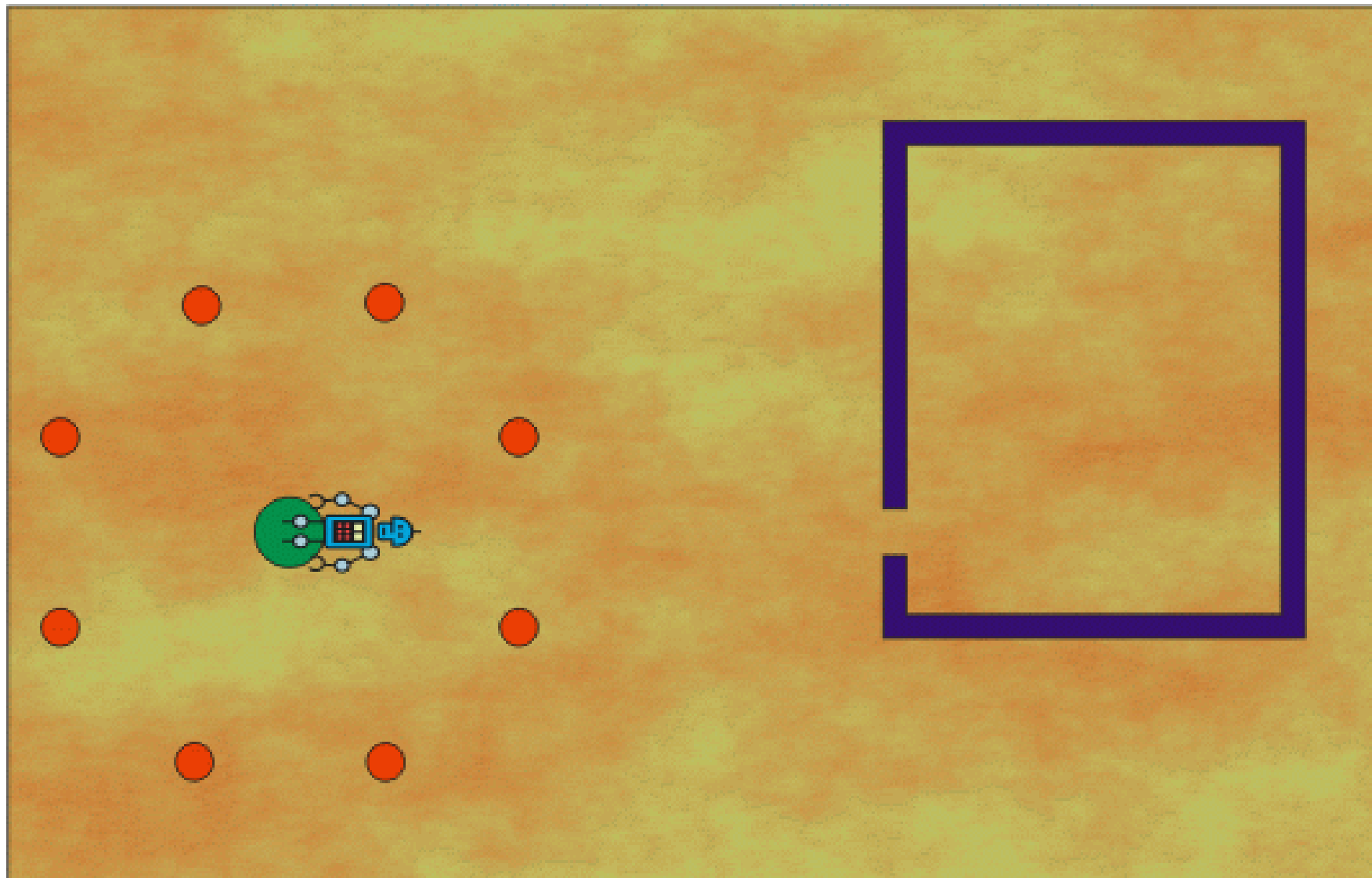
1. Nájsť východ z pristávacej plochy
2. Dopraviť sa na základňu
3. Odčítanie teplôt zaznamenaných meracími stanicami
4. Odčítanie teplôt zaznamenaných meracími stanicami, ak sú merané v rôznych jednotkách, odoslanie priemerov na Zem



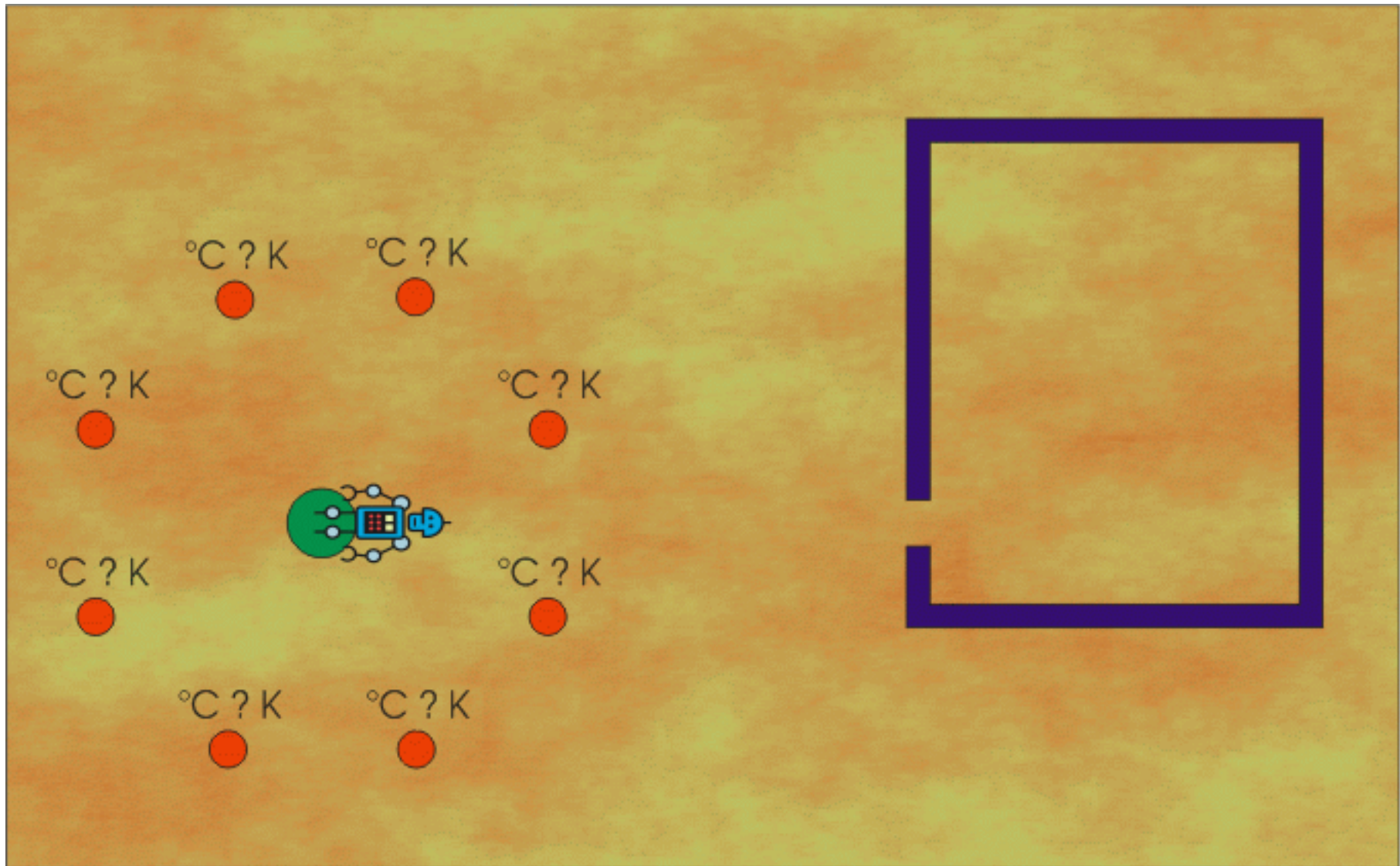
Oddelenie didaktiky informatiky a podporných technológií, ÚINF PF UPJŠ v Košiciach



Oddelenie didaktiky informatiky a podporných technológií, ÚINF PF UPJŠ v Košiciach



Oddelenie didaktiky informatiky a podporných technológií, ÚINF PF UPJŠ v Košiciach



Oddelenie didaktiky informatiky a podporných technológií, ÚINF PF UPJŠ v Košiciach



# Výsledky a postrehy [1]

- nemožno zovšeobecniť
- predstavujú vzorku toho, s čím sa môže študent/učiteľ stretnúť pri výučbe alg. a prog.
- naznačujú budúce komplikácie

# Výsledky a postrehy [2]

- jasne definovaná množina príkazov kontra časové vsuvky

**potom** otočenie doprava

**a zase** dopredu pokým nenarazí

- lepšie orientácia v časovej nadväznosti činností robota

# Výsledky a postrehy [3]

- zlá prehľadnosť programov

6. ak žiadny naraz, tak najdena diera - koniec, inak naraz na stenu, potom opakovať krok 4. až 5., pokiaľ nebude uskutočnený krok vpred a žiadny naraz, vtedy najdena diera - koniec

- prehľadnosť sa zrejme zlepší po požiadavke na modifikáciu, alebo pri väčších programoch

# Výsledky a postrehy [4]

- stroj robí len to, čo sme ho naučili

chod rovno k ľubovolnej cervene  
meracej stanici a zastan

alebo

vrátiť sa o kúsok späť

- „neriešiteľná“ úloha: ľubovoľná, kúsok

# Výsledky a postrehy [5]

- predpoklad, že stroj vychádza z kontextu
  2. rýchly postup vpred
  3. opatrný postup vpred
- počítač nerobí to čo chceme, ale to čo sme mu prikázali

# Výsledky a postrehy [6]

- neelementárne (inak nepoužiteľné) príkazy

pohyb do kruhu v smere  
hodinových rucíciek

- sústredenie sa na podstatu veci
- neskoršia konfrontácia s „primitívnosťou“  
príkazov jazyka

# Výsledky a postrehy [7]

- nie postupnosť inštrukcií, ale definovanie výsledného stavu

Dojdi k stene a postav sa tak, aby si ju mal po pravom boku

- schéma:  
vstup – postup – zodpovedajúci výstup  
sa mení na  
vstup – definovanie výstupu – výstup
- sústredenie sa na problém **čo** naprogramovať (Prolog),  
namiesto toho, **ako** to urobiť (Pascal, C)

# Výsledky a postrehy [8]

- vsuvky popisujúce stav
  - 1.stoji priamo vo východe
  - 2.urob krok aby si sa dostal von z obdĺžníka
  - 3.teraz si vonku
- pomôcka pre autora (súvislosti, invarianty)
- potreba (užitočnosť) komentárov v prog. jazykoch



# Výsledky a postrehy [9]

- duplicita, neefektívnosť kódu

krok vpred, tento krok opakovať 150-krát, pokiaľ nebude naraz na stenu, alebo naraz do niečoho,

- problémové zápisy podmienok (duplicita, kombinácia vylučujúcich sa podmienok)

# Výsledky a postrehy [10]

- činnosti prebiehajúce súvisle, ale v inom kontexte
  - 2.urob krok aby si sa dostal von z obdĺžníka
  - 3.teraz si vonku
  - 4.teraz sprav 49 krokov a chod pri tom stale rovno
  - 5.po 50 kroku zastan
- študenti si často neuvedomujú, že postupnosť n-krokov sa dá zapísať jedným príkazom

# Výsledky a postrehy [11]

- premenné nie sú potrebné  
... Zaznamenaj teplotu  
... odmeria teplotu
- prvé náznaky až vo 4. úlohe, len intuitívne  
merania si stále uklada a ku každemu  
meraniu priraduje poradove číslo od  
1 po 8 (číslo stanice), čas merania a  
či meral v kelvinoch alebo v  
celsiových stupnoch
- premenná, jej inicializácia problém pre  
začiatočníka (ako sa dostane hodnota do  
premennej?)

# Výsledky a postrehy [12]

- originalita, ktorú by sme pri programovaní zrejme potlačili  
... sa pohybovať po dráhe v tvare špirály
- z rýchlosti a dráhy => čas, cyklus času
- cyklus s podmienkou v strede
- možnosť získať kvalitatívne iné, prekvapivé, originálne riešenia

# Výsledky a postrehy [13]

- prirodzené použitie príkazu vetvenia
- používanie cyklov
- tvorba „podprogramov“  
príkaz č.3 obsahuje aj kroky 4.  
a 5. (ktoré sú inak samostatné)
- intuitívna práca s premennou
- bez ohľadu na zápis, algoritmy boli správne

# Záver

- rozdiel medzi jazykom prirodzeným a programovacím => problémy
- potreba porozumieť mysleniu začiatočníka
- prispôsobenie štýlu a formy výučby alg. => odstrániť zbytočnú frustráciu
- krása alg. a programovania => neodradit' študentov
- navrhnúť (modifikovať) doterajší spôsob výučbu alg. a prog.