



HungaroLogo'96

Budapest, október 17-18.

Válogatás a harmadik magyar Logo konferencia előadásából

Szerkesztette: Dr. Farkas Károly

Neumann János Számítógéptudományi Társaság

1996

ITA/395



ITA/395



HungaroLogo'96

Budapest, október 17-18.

Válogatás a harmadik magyar Logo konferencia előadásaiból

Szerkesztette: Dr. Farkas Károly

Neumann János Számítógéptudományi Társaság

1996

THE UNIVERSITY OF CHICAGO

PHYSICS DEPARTMENT

PHYSICS 311

LECTURE 1

LECTURE 1: INTRODUCTION TO QUANTUM MECHANICS

1.1

Tartalom

1. Bevezetés
2. Bedő Ferenc: Csiga-e a gyorsuló teknőc?
3. Radnai Ildikó: Tapasztalataim a "nyuszik" és a PC Logo együttműködéséről.
4. Dr. Farkas Károly: Rekurzio
5. Könczöl Tamás: Egy újabb magyar oktatóprogram, Betűvarázs
6. Kőrösné Mikis Márta: Informatika tantervek az országos adatbankban
7. Könczöl Tamás: Internet és az oktatás
8. Andrej Blaho, Ivan Kalaš, Monika Matyušová: Environment for Environments: New Metaphor for Logo
9. Szorongás teszt kérdései.
10. Nyit levél az iskolaigazgatóknak.

Bevezetés

Idén harmadszor rendezte meg az NJSZT a HungaroLogo konferenciát. Idén először a témát kibővítettük, a Logo Szakosztály és a Közoktatási Szakosztály együttműködve alakította ki a konferencia programját, amelyben nem csak Logo témák szerepeltek. Azonban a közoktatás kérdéseit is logo-pedagógia alapján szemléltük, tárgyaltuk. A konferencia szervezői úgy véljük Papert munkásságából nem is a Logo programnyelv az igazán jelentős érték, hanem azok a pedagógiai tézisek, amelyeket hazánkban is immáron pedagógusok százai megértettek, hirdetnek és fejlesztenek. Nagy János kollégánk Szolnokról nem is csak logo-pedagógiáról, de logo-filozófiáról beszélt.

Idén is volt a konferenciának határon túli előadója: Lőrinczy András. Javaslatára határoztuk el a konferencián, hogy ezen tanév végén Logo tanulmányi versenyt rendezünk. A versenyre az alsó tagozatos gyerekek nevezhetnek. Az érdeklődő iskoláknak, ha jelentkeznek április 1-ig írásban az NJSZT Titkárságán, májusban megküldjük a versenypéldákat. Az iskolákban folytatott első fordulók három győztesét Siófokra várjuk júniusban. A döntő első három helyezettje ingyen vehet részt a Hungest JIO táborában.

Jövőre a HungaroLogo az EUROLOGO'97 kísérő rendezvénye lesz. Aki igényli, annak az EUROLOGO konferenciára elfogadott előadásához tolmácsot biztosítunk. A HungaroLogo 97-re természetesen a főrendezvényen való részvételtől függetlenül is szeretettel várjuk a résztvevőket.



Csiga-e a gyorsuló teknős?



A Logo-pedagógia alkalmazása a haladó Logo oktatásában

Bevezető

Az informatika új műveltségterületként jelent meg, és szerepelt a NAT mindegyik változatában. A tanításának tartalmára, tantárgyban való elhelyezésére vonatkozó elképzelések viszont a múltban is sok vitát váltottak ki és valószínűleg a jövőben sem lesznek egységesek.

Az eddig is létező tantárgyakkal szemben a valószínűleg megjelenő új, informatikai ismereteket tanító tantárgyak közös jellemzője, hogy tantárgymódszertana jórészt még kialakulatlan, kidolgozatlan. A módszertan kidolgozására vállalkozók mégis előnyben vannak abból a szempontból, hogy nem kell ragaszkodniuk régi, esetleg sok éves, megcsontosodott módszerekhez, hanem bátran nyúlhatnak új megoldásokhoz.

Nem baj az, hogy a tanári szabadságot kihasználva a NAT által meghatározott informatikai ismereteket más-más keretek között tanítjuk majd, az sem fog problémát jelenteni, ha egyikünk adott évfolyamon többet tanít majd, mint a másikunk. Feltétlenül fontosnak azt tartom (és ez nem csak az informatikára vonatkozik), hogy mindenkor az adott életkori szakasznak megfelelő módszerek segítségével történjen az életkornak megfelelő tananyag elsajátítása.

Neves oktatási szakemberektől számtalanszor elhangzott az utóbbi években, hogy az eljövendő évek iskolai változásait a gyermekközpontség gondolatának kell befolyásolnia. Egy vélhetően új tantárgy tanításában nem szabad elszalasztanunk az alkalmat, és olyan oktatási módszereket kell kidolgoznunk, amit áthat a gyermekek sajátosságait figyelembe vevő gondolatvilág.

A módszertant megválasztó, kidolgozó munkánk során érdemes felhasználni a pszichológia és pedagógia kapcsolatos eredményeit. A leginkább figyelemre méltó ilyen elképzelésnek a Logo-pedagógiát tartom.

A Logo-pedagógián én azt értem, hogy a Logo nyelv, illetve a számítógép segítségével, vagy ezekhez hasonló érdekes eszközzel (LEGO készlet, origami, programozható robot, nyelvi játékok, stb.) olyan tanulási környezetet teremtünk tanítványaink számára, amiben megvalósulhat a Piaget-i, un. észrevétlen tanulás, vagyis a gyerekek játékos körülmények között komoly problémával foglalkozva, jelentős ismereteket szerezhetnek. Az ilyen környezetet nevezzük Logo szerű tanulási környezetnek.

Mindezt felhasználom az informatika tanítása során. Feltétlen bizalmat szavazva a Logo-nak, az 5., 6., 7. osztályos tananyag kb. egyharmad részét ennek segítségével igyekszem megismertetni a gyerekekkel.

Szándékosan nem írtam azt, hogy igyekszem "tanítani". Ez két okból sem jó kifejezés. Egyrészt azért, mert a Logot nem egy megtanítandó, megtanulandó tananyagnak tekintem, igazi haszna a vele való foglalkozásban, a már említett észrevétlen tanulásban rejlik. Másrészt mert a Logo pedagógiájában a Logo-val való foglalkozás során a tanító szerepe nem az ismeretek egyirányú közvetítése a diákjai felé. A Logo-záskor a tanár ötleteket ad, problémákat vet fel, segít ha kell, igyekszik észrevétlenül koordinálni a diákok felfedező kalandozását az adott problémán belül.

A Logo és a tanterv

A Logo-ról már hazánkban is számos méltató munka jelent meg. Úgy gondolom a tantervi szabadság küszöbén álló, a centralista tantervet, szigorú tanmeneteket elfelejtteni igyekvő iskolarendszerünkben hasznos lehet a Logo nyelv tantervi szemléletű vizsgálata.

Szeretném szemléltetni, hogy a Logo nyelv elemeivel való foglalkozást tervezhetjük úgy, egymás után, hogy a sorrendiséget a gyerekek tudása és kísérletező kedvének bővülése határozza meg. Tapasztalataim szerint ugyanis munkájuk során szinte észrevétlenül, igényük mutatkozik fejlettebb nyelvi lehetőségek megismerésére. Ha ezt kielégítjük és aktivitásukat kihasználjuk akkor újabb és újabb használható elemeket szeretnének megismerni, ezek segítségével pedig további ismeretekre tehetnek szert.

E megfontolás alapján a Logo nyelv rendszerét nem tematikus elvek szerint, hanem a tanulók igényei alapján építhetjük fel, és beilleszthetünk akár programozástechnikai, matematikai, nyelvi, kommunikációs elemeket is, de nem azért, mert ez a tananyag, hanem mert a gyerekek akarják, hiszen további munkájukat segíteni fogja.

Itt jelentkezik a Logo-pedagógiának egy másik meggyőződése, látszólagos nehézsége, mely szerint a gyerekek nem egyszerre, nem egy időben jutnak el egy-egy probléma (nyelvi elem) megismerésének igényéhez.

Mindezek alapján már sejthető, miért is írt Seymour Papert tanterv nélküli tanításról, és miért emlegeti alig leplezett ellenszenvvel a tanmeneteket.

Természetesen meg kell jegyezni, hogy mindezekon ő sem minden elképzelés, és cél nélküli tanítást értett.

Az igényszintek

Az előbbieket szerint a Logo-val kapcsolatos ismereteket feloszthatjuk ún. igényszintek alapján is úgy, hog tanulóinknak mikor mutatkozik kívánalma egy újabb, fejlettebb lehetőség megismerésére.

Az alap

Az ismerkedést a gyerekek, a számítógép-teknős irányításával kezdik, ami tulajdonképpen egy robot.

Tudjuk, hogy a fent nevezett hüllő a képernyő közepén van, a képernyő felső szélének irányába néz, a szájában eg ceruzát tart: ez az alapállapota. Ha elmozdul, a ceruza vonalat húz az alatta levő mezőbe (rajzlapra).

A forward (előre haladás), back (hátra haladás), left (balra fordulás), right (jobbra fordulás) és a home (haza) parancsokkal tudjuk irányítani. Ez a néhány parancs - kiegészítve a képernyő (rajzlap) letöltésének lehetőségével elég arra, hogy a gyerekek kimeríthetetlen fantáziájának határt nem szabó, sok-sok variációs lehetőséggel bír rajzeszközük legyen. A kezdeti próbálgatások, próbálkozások egyre inkább céltudatos rajzolási tevékenységg alakulnak.

Az első igényszint

A céltudatos munka, közvetlenül ez után új nyelvi elemek megismerésének igényét fogja eredményezni. A gyerekek munkájuk során maguk vetik fel annak szükségességét, hogy teknősük a ceruzáját felemelhesse a rajzlapról, (ekkk haladás közben nem húz vonalat), majd újra visszatehesse oda, ceruzáját radírra cserélhesse (ekkor amerre hala letöri a vonalat), ceruzáját színes ceruzára cserélhesse, vagy rajzlapját színes lappal helyettesíthesse. Mindezekkk lehetővé válik a képernyőn különálló, és színes ábrák létrehozása. Felmerül az igény arra, hogy a teknős a raj elkészülte után váljon láthatatlanná, de ha kell újra jelenjen meg.

A második igényszint

A következő fejlesztési lehetőség szükségességét szintén a gyerekek vethetik fel. Addigi munkájukban már előfordu néhány sokszög, de a négyzet és a háromszög biztosan. Azért, hogy ne kelljen újra és újra többször leírni ugyanazokk a parancssorokat meg kell ismerniük a repeat (ismételd) parancsot, aminek segítségével bizonyos tevékenységek többször is elvégezhetőek a teknőssel. Pl.:

```
repeat 3 [forward 50 left 120]
```

Így már könnyebben lehet négyzetet, hatszöget, nyolcszöget, stb.-t rajzolni, megismerkednek a szabályc sokszögekkel, azok viszonyaival, a sokszögek külső szögeire vonatkozó összefüggéssel. Az elfordulások száma és szög közötti kapcsolat megismerése során tapasztalatokat szereznek a fordított arányosságról. Ezen ismeretek elvezetik ökk a körirajzolás tudományához, ami egy igazán jelentős felfedezés lesz számukra.

```
repeat 360 [forward 1 left 1]
```

A harmadik igényszint

Az eljárások készítésének lehetősége már egy fejlettebb tanulói gondolkodás eredménye. Ennek lényege, hogy teknőst már nem csak parancsokkal irányítják, hanem megpróbálják tanítani is. Felnőtt szemmel néz tulajdonképpen már programot írnak.

Az eljárás megalkotása, létrehozása után a teknős tudása egy új szóval gyarapodik.

```
to négyzet
```

```
repeat 4 [forward 50 left 90]
```

```
end
```

Ekkor a négyzet rajzolása már egyszerűbb lesz. Nem kell az addigi hosszú sort beírni, hanem csak egy szót.

```
négyzet
```

Ha már készen vannak a megfelelő eljárások, akár egy házikó rajzolása is megvalósítható egyetlen szó beírásával.

```
ház
```

A negyedik igényszint

Haladó Logo-zásnak számít már - az eljárások megismerése után - a paraméteres eljárások használatána lehetősége.

Ha pl. egy eljárással nem csak 50 lépés oldalú négyzetet akarunk rajzolni, be kell vezetnünk a paraméterc négyzetrajzoló eljárást.

A paraméteres eljárás definiálásakor létrehozunk a memóriában pl. "egy oldalhossz feliratú fiókot".

```
to négyzet :oldalhossz
```

```
repeat 4 [fd :oldalhossz lt 90]
```

```
end
```

Ezzel az eljárással már bármekkora oldalú négyzetet tudunk rajzolni. Indításkor, az eljárás neve után meg kell adn (a fiókban el kell helyezni) a kívánt értéket.

```
négyzet 72
```

Az ötödik igény szint

A Logo nyelvvel való foglalkozás során már haladó szintnek számít a paraméteres eljárások használata. Igényként jelentkezik a gyerekek részéről, hogy egy eljárásban két, vagy több paraméter is legyen. Rögtön rájönnek arra is, hogy bizonyos értékek közt összefüggés van. Aztán kiderül annak szükségessége, hogy a paraméterek közötti összefüggéseket kihasználva egy eljárson belül akár számításokat is tudjunk végezni.

Az egyik ilyen lehetőség pl. egy általános sokszögrajzoló eljárás létrehozása.

```
to sokszög :oldalhossz :szögszám
make "fordulás 360 / :szögszám
repeat :szögszám [ fd :oldalhossz lt :fordulás ]
end
```

Ennek mintájára már könnyű megrajzolni az alábbi sugaras alakzat belső részét. A köré húzható kör megrajzolása azonban már nehezebb feladat, mert olyan körrajzoló eljárás megalkotására van szükség amiben az induló paraméter a kör sugara.



A kérdéses eljárás a kör területének ismeretében elkészíthető.

```
to kör :sugár
repeat 360 [ fd 2 * 3.14 *:sugár / 360 lt 1 ]
end
```

Az új ismeretet kihasználva, szinte minden gyerek el fogja készíteni a jobb oldali igazán látványos ábrát.

Érdekes problémák

Az előbbieket bizonyítják, hogy a Logo nyelv egy biztos pedagógiai alapokra épült, alkotó módon létrehozott eszköz, ami a gyerekek motiválását, aktiválását a vele való foglalkozás során képes fenntartani.

Egy szint után szükség lehet arra is, hogy a tanár néhány ötletet adjon, érdekes problémát vessen fel a gyerekeknek érdeklődésük további fenntartása érdekében.

Az első érdekes probléma

A Logo-val való foglalkozás talán legérdekesebb jelensége, - amit a BASIC-ben "nevelkedett" felnőttek meglehetősen hihetetlennek tartanak, a gyerekek viszont természetesen vesznek, - az önmagát hívó eljárás használata, a rekurzio.

Egy egyszerű kétparaméteres eljárással rajzoljunk újra sokszögeket!

```
to csillag :oldalhossz :fordulás
fd :oldalhossz
lt :fordulás
csillag :oldalhossz :fordulás
end
```

A ciklusból való kilépéshez billentyűkombináció használatára lesz szükség.

Az eljárás kipróbálása során a paraméterek szerepe világosabbá válik, valamint a módszeres kísérletezésre is rászoktathatjuk tanulóinkat, ha egyszer az egyik, majd a másik paramétert változtatjuk meg.

```
csillag 80 90
csillag 80 120
```

Nem szükséges csak a "kerek" elfordulás értékek használata. Néhány ötletet adva, a gyerekek kísérletező kedve segítségével sok érdekes ábrát kaphatunk.

```
csillag 80 144
csillag 80 135
csillag 80 100
```

Második érdekes probléma

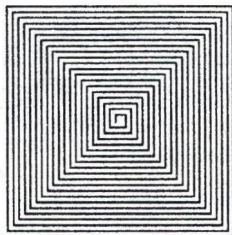
Az előbbi eljárás továbbfejlesztésével további látványos rajzokat készíthetünk. Alakítsuk át úgy az eljárást, hogy az oldalhossz minden lépésben megváltozzon! Ennek érdekében be kell vezetni egy `hnöv` nevű paramétert, amivel megadhatjuk, hogy mennyivel szeretnénk változtatni az oldalhossz értékét.

```
to csiga :oldalhossz :fordulás :hnöv
fd :oldalhossz
rt :fordulás
make "oldalhossz :oldalhossz + :hnöv
csiga :oldalhossz :fordulás :hnöv
end
```

A kipróbálásnál vigyázni kell az időbeni megállításra, hiszen végtelen ciklust alkotunk.

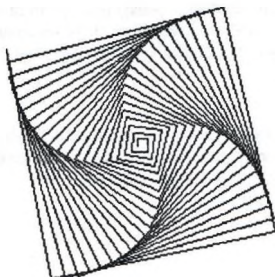
Érdekes ábrákat kapunk. Pl.:

```
csiga 5 90 2
```



Most sem kell ragaszkodnunk a megszokott szögértékekhez. A paraméterek kevés változtatásával az előbbi ábra jelentősen megváltozik.

```
csiga 5 91 2
```



Egy újabb igény

Annak érdekében, hogy ne kelljen billentyűkombinációt használni az eljárás megállításához, a gyerekek újabb programozási lehetőség megismerését igénylik. Ez a feltételes utasítás használata. A `csiga` eljárást ki kell egészíteni egy sorral.

```
to csiga :oldalhossz :fordulás :hnöv
fd :oldalhossz
rt :fordulás
make "oldalhossz :oldalhossz + :hnöv
if :oldalhossz > 150 [stop]
csiga :oldalhossz :fordulás :hnöv
end
```

A gyermeki kísérletező kedv

A gyerekek találékonyaságára jellemző, hogy szinte minden létező variációs lehetőséget kipróbálva találnak sokszor nagyon tanulságos ábrákat. Egyik példa erre a spirál. Pl.:

```
csiga 0 1 0.0005
```

A másik, a gyorsuló teknős, amit teljes egészében gyerekek találtak ki. A csigavonal-rajzoló eljárás különleges paraméterekkel, egy induló és gyorsulva mozgó teknőst fog szimulálni.

```
csiga 0 0 0.0005
```

Harmadik érdekes probléma

A második probléma továbbgondolásaként most egy olyan eljárást készítsünk, amiben az elfordulás változik meg minden lépésben!

```
to inda :oldalhossz :fordulás :fnöv
fd :oldalhossz
rt :fordulás
make "fordulás :fordulás + :fnöv
inda :oldalhossz :fordulás :fnöv
end
```

A kipróbálás során tapasztalható, hogy egy még érdekesebb világba csöppentünk. A variációk száma több, az alakzatok változatosabbak, mint az oldalhossz változtatása esetén. Csak ízelítőként három példa:

inda 5 0 5



inda 5 1 5



inda 5 1 7



A sok-sok változatos lehetőség a gyerekeket (és tanítóikat) további kísérletekre készíti. Itt igazán szükséges a módszeres kísérletezés.

Ha sikerül az érdeklődést megfelelő irányba terelni, akkor matematikai szempontú összefüggéseket is találhatunk a paraméterek közt.

Lehetséges más út is

A geometriai jellegű problémák mellett van lehetőség más irányú, érdekes, a gyerekek érdeklődését felkeltő, aktivitását fokozó lehetőségek megmutatására, ötletek adására is.

Az újabb Logo verziókban (pl. Logo Writer, Comenius Logo) lehetőség van sprite-ok kezelésére és több teknős mozgatására. Ez már önmagában is rengeteg ötletet ad a gyerekeknek. Mindezt kiegészíthetjük a hanghatások használatának lehetőségével, így a gyerekek számára vonzó animáció világában kalandozhatunk.

Néhány lehetőséget megmutatva gyerekeinknek rögtön aktivizálódik a fantáziája és alig győzünk válaszolni a sok, "Azt, hogy lehet megcsinálni, hogy" kezdetű kérdésre. Ha ügyesen koordináljuk a gyerekek munkáját, és persze bírjuk a kérdések, problémák ostromát, akkor szintén kialakul egy olyan környezet, amiben a gyerekek, azt észre sem véve ismerkednek meg fejlettebb programozási lehetőségekkel.

Mindez persze egy másik írásnak lehetne témája.

Az ismertett lehetőségeken kívül is biztosan adhat más hasznos "munkalehetőséget" is a Logo nyelv. Sok még a feltáratlan lehetősége, van még kevésbé ismert képességfejlesztő hatása.

Bedő Ferenc

Községi Általános Iskola
Pusztaszentlászló
Kossuth u. 112.
Tel.: (92) 369-033

1. The first part of the document discusses the importance of maintaining accurate records of all transactions. It emphasizes that proper record-keeping is essential for the integrity of the financial system and for the ability to detect and prevent fraud.

2. The second part of the document outlines the specific requirements for record-keeping, including the need to maintain original documents and to keep copies of all supporting documents. It also discusses the importance of ensuring that records are accessible and retrievable.

3. The third part of the document discusses the consequences of non-compliance with the record-keeping requirements. It notes that failure to maintain accurate records can result in the loss of the ability to defend against claims and may also result in the imposition of penalties.

4. The fourth part of the document discusses the importance of training and education in ensuring compliance with the record-keeping requirements. It notes that all personnel involved in the financial system should be trained in the proper record-keeping procedures and should be held accountable for their actions.

5. The fifth part of the document discusses the importance of regular audits and reviews in ensuring the accuracy and integrity of the records. It notes that audits and reviews should be conducted on a regular basis and should be conducted by independent parties.

6. The sixth part of the document discusses the importance of maintaining the confidentiality of the records. It notes that records should be stored in a secure location and should be accessible only to authorized personnel. It also discusses the importance of implementing appropriate security measures to protect the records from unauthorized access and disclosure.

7. The seventh part of the document discusses the importance of maintaining the accuracy and integrity of the records over the long term. It notes that records should be preserved for the full term of their legal retention and should be protected from loss, damage, and destruction.

8. The eighth part of the document discusses the importance of maintaining the accuracy and integrity of the records in the event of a disaster. It notes that records should be backed up regularly and should be stored in a secure location that is protected from natural disasters and other risks. It also discusses the importance of having a disaster recovery plan in place to ensure that records can be restored in the event of a disaster.

9. The ninth part of the document discusses the importance of maintaining the accuracy and integrity of the records in the event of a change in personnel. It notes that records should be transferred to the new personnel in a secure and accurate manner and that the new personnel should be trained in the proper record-keeping procedures.

10. The tenth part of the document discusses the importance of maintaining the accuracy and integrity of the records in the event of a change in the financial system. It notes that records should be updated to reflect the changes and that the new system should be tested thoroughly before being implemented.

TAPASZTALATAIM A "NYUSZIK" ÉS A PC LOGO EGYÜTTMŰKÖDÉSÉRŐL

A Logo-t azért tarom jó eszköznek, mert az algoritmikus gondolkodás játzsva sajátítható el a segítségével. Ezen kívül, könnyen kapcsolható a ma általánosan alkalmazott felhasználói programcsaládhoz, a Windows-hoz. PaintBrush-sal rajzolt képek behívhatóak a Logo-ba, Logo-val alkotott rajzaikat pedig felismeri a PaintBrush. Ettől kezdve a szövegszerkesztőben a program szövege és a rajzolt ábra egyszerre megjeleníthető. Ezek a távlatok.

A gyerekek 6-7 évesek (első osztályosok), nem átlagos képességekkel megáldva. Néhányan már a másodikos tananyagot tanulják közülük

ELŐSZÖR IS. MIÉRT ÉPPEEN PC LOGO?

- ◆ Amikor foglalkozni kezdtem ezzel a programmal, két versioja volt elérhető. Az általam használt és a LogoWriter. Ez utóbbinak a grafikája nem túl finom, így az elkészült művek sem igazán látványosak.
- ◆ Viszont a PC Logo-ban a legizgalmasabb alakzat maga a teknős, így a gyerekek figyelme teljesen a mozgására irányul.

MÁSODSZOR. AZ EDDIGI TAPASZTALATAIM:

- ◆ Kezdetben természetesen gátat jelentett volna a billentyűzet használata néhány csemeténél. Ezért egy egyszerű kis programot írtam, melynek segítségével az egérgombokkal irányíthatták a teknős mozgását. A felhasználó két óra alatt megtanulták az alapvető mozgásokat (haladás, forgás). Néhány egybetűs funkciót is beleépítve, szinesebb a játék, bővül a felfedezett lehetőségek köre. Túl sokáig nem érdemes itt megállni, mert utána esetleg nagyon ragaszkodnak az egérhasználat kínálta egyszerű mozgathoz.
- ◆ Következő lépés az alapparancsok használata a billentyűzetről. Itt tudatosítani kell, hogy vannak paraméteres és egyszerű utasítások. Ez természetesen nem ezt a szóhasználatot jelenti még, csak a parancskiadás két változatának felismerését. Ezzel párhuzamosan a házi feladatok segítségével el kell kezdeni a primitív szögfogalom kialakítását. Azt ugyanis rögtön érzik, hogy nagyobb számhoz nagyobb előrelépés tartozik, de ugyan ez forgás esetében már bonyolultabb. Erre az óra kiválóan alkalmas eszköz (mellesleg még az egész órák leolvasását is megtanulják).
- ◆ A REPEAT (ismétlés) használata már komolyabb gondolatokat igényel, mert valamilyen szinten tudni kell hozzá fejben követni az utasítások hatását. Ez sok feladattal érhető csak el, amelyekben ismétlődő utasítássorozatok vannak.

Egy idő után már az egyéni tempójuktól függ, hogy milyen feladatok adhatóak. Kilenc gyerek jár az osztályba, de hamar nyilvánvalóvá vált, hogy legalább 3 (esetleg 4) szinten vannak az ilyen irányú képességeik. Ezért néhányuknál nem előre, hanem vissza kellett lépni. A lényeg az, hogy egyiknek se szegje kedvét az unalom, vagy az elérhetetlen magas követelményszint.








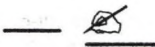





- ◆ Ehhez jól használható, hogy a "menőket" segédezőnek alkalmazom, a feladattal járó felelősséget is tudatosítva.
- ◆ Azoknak, akik nehezebben boldogulnak, előre leírt utasítássorozatokat (3-6 parancs) kell beírniuk, és a lapra általam odanyomtatott teknőshöz le kell rajzolniuk az eredményt. Itt egy-egy sorozat csak egy-egy momentumban tér el, hogy annak az utasításnak a hatása váljék tudatossá. Közben gyakorolják a gépelést is.

AZ EDDIG MEGTARTOTT NÉHÁNY ÓRA TANULSÁGAI SZÁMOMRA:





- ◆ Általános iskolában a számítástechnika hatékonyan csak nagyon kis létszám mellett tanítható.
- ◆ Jó képességű gyerekek esetében sem várható el mindenkitől, hogy egyformán teljesítsen minden tantárgyból.

A legfontosabb az érdeklődés fenntartása. Ezért jó, hogy már léteznek újabb Logo verziók is, melyekben bővül a lehetőségek köre (még több teknős, általuk tervezett alakokkal rajzolhatnak - egyszerű mozgások, "rajzfilmek" készíthetőek).

A LOGO program alaputasításai

Clear Screen CS		törli a képernyőt
HOME HOME		teknős középre
ShowTurtle ST		teknős látható
HideTurtle HT		teknős nem látható
PenUp PU	 	toll fel
PenDown PD	 	toll le
PenErase PE		radír
SETPenColor SETPC		szín példa: SETPC 12
Forward FD		előre példa: FD 50
Back BK		hátra példa: BK 75
Left LT		balra példa: LT 90
Right RT		jobbra példa: RT 45
REPEAT REPEAT		ismétel példa: REPEAT 4 [FD 50 RT 90]





Példák a LOGO utasításainak egyszerű használatára

<pre>(cs rt 90) setwidth 1 fd 30 setwidth 8 fd 30 setwidth 1 fd 30</pre>	
<pre>(cs rt 90) setwidth 8 fd 30 setwidth 1 fd 30 setwidth 8 fd 30 setwidth 1 fd 30</pre>	
<pre>(cs rt 90) setwidth 1 fd 30 setwidth 8 bk 30 setwidth 1 fd 30</pre>	
<pre>* repeat 2[PD fd 30 PU fd 30]</pre>	

HA EZEKET AZ UTASÍTÁSOKAT ADNÁNK NEKI MIT ALKOTNA A TEKNŐS?

RAJZOLD LE!

Példák a LOGO utasításainak egyszerű használatára

<p>fd 30 PU fd 30 PD fd 30</p>	
<p>PU fd 30 PD fd 30 PU fd 30 PD fd 30</p>	
<p>fd 30 PU bk 30 PD fd 30</p>	
<p>repeat 4 [PD fd 30 PU]</p>	

HA EZEKET AZ UTASÍTÁSOKAT ADNÁNK NEKI MIT ALKOTNA A TEKNŐS?

RAJZOLD LE!



S
E
T
W
I
D
T
H

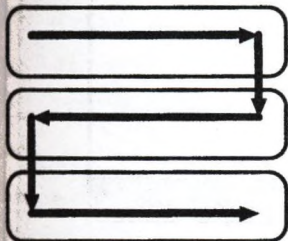


PÉLDA: SETWIDTH 40

(A KEZDETI BEÁLLÍTÁS VONALMÉRETE 1-ES)

MAGYAR ZÁSZLÓ

MILYEN SZINŰ? ÍRD A BETŰKET A KOCKÁKBA ÉS SZINEZZ!



RED

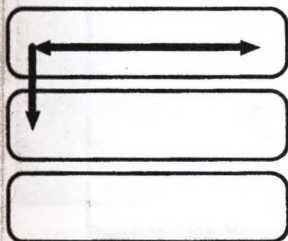


WHITE



GREEN

FOLYASD A TEKNŐS ÚTJÁT, HA ISMÉT A ZÁSZLÓT SZERETNÉNK LERAJZOLNI!



* HA TUDOD, AKKOR ÍRD LE A FÜZETEDBE AZ UTASÍTÁSOKAT IS, MELYEK HATÁSÁRA A TEKNŐS LERAJZOLJA A ZÁSZLÓT!

fd 30

bk

rt 20

fd 30

bk

rt 20

fd 30

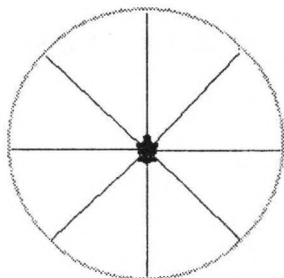
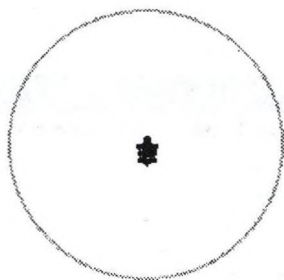
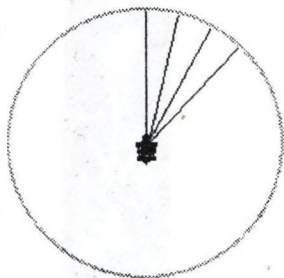
bk

rt

fd 30

bk

rt

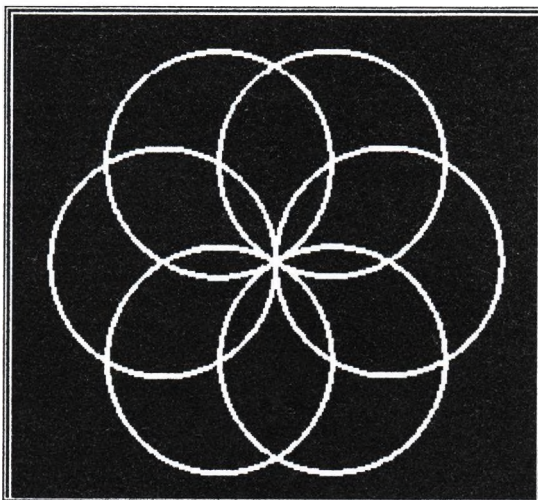


repeat 4 [fd 30 bk 30 rt 90]

repeat [fd bk 30 rt 45]

EGÉSZÍTSD KI!

VIRÁG



A program egy virágot rajzol körökből.

```
to virag
  draw
  ht
  setwidth 3

  repeat 6 [ kor rt 60 ]

end

to kor
  repeat 360 [ fd 1 rt 1 ]
end
```

A program neve: VIRÁG

Mindent alapállapotra hozok. (DRAW)

Eltüntetem a teknőst, gyorsítom a rajzolást. (HT)

A körövonalak vastagságát határozom meg.

Hat körből áll a kép, egy közös pontjuk van, és e pont körül 60fokkal vannak elforgatva a szomszédaikhoz képest.

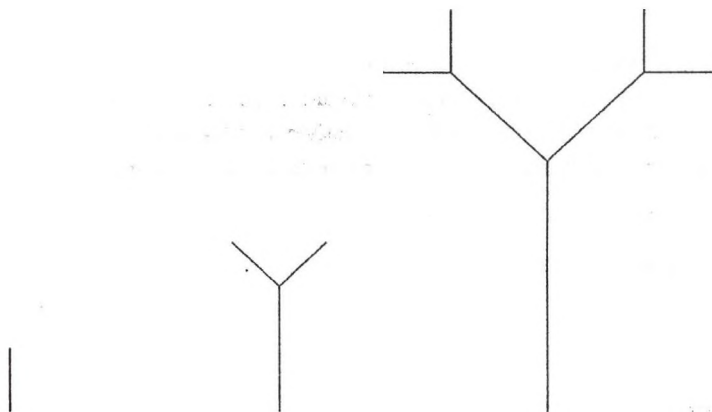
Program vége: (END)

Egy külön programot kell írunk a kör megrajzolásához. Így nem kell többször is leírunk ugyan azt a folyamatot. Ez az eljárás azt használja ki, hogy egy "nagyon sok" szögű sokszög, már körszerűnek látszik.



Dr. Farkas Károly: Rekurzió

A legmagasabb szinten valamit talán akkor tudunk, ha azt képesek vagyunk a gyerekeknek megmagyarázni. A Logo programnyelv és a logo-filozófia jelentős segítség, több, korábban felsőbb szintűnek minősített ismeret gyermekek számára történő tanításakor. A logo-pedagógia jellemzői: a szintonia (a beleélés, az eljátszás lehetősége), a láttatás, a vizuális szemléltetés. Ezt hasznosíthatjuk az önreprodukáló, önmagukat - felépítési szerkezetüket, magasabb szinteken ismétlő rendszerek technóc-geometriával történő bemutatásakor is. Számomra a rekurzív eljárások ilyen módon történő megértése emlékezetes heurisztikus élmény volt. A legegyszerűbb rekurzív alakzatsor egy növekedő fa évenkénti modellje.



A HungaroLogo '96 konferencia mágneslemezen a DEMO részben megtaláljuk ezt a gondolatsort - az itteninél szokásosabb módon és tovább vezetve. Jelen ismertetés originalitása (eredetisége) az, hogy az eljárások megírásakor csak egy paramétert használ, hiszen a faágak mérete a törzs és az életkor függvényeként kifejezhető.

Az egyéves fát (a hajtás egyetlen ág) rajzoljuk meg egyetlen vonalként. Az eljárások leírásához a LogoWriterrel elterjedt magyar logot használom: az **m** jelenti az előrehaladást, a **h** a hátrálást, a **j** a jobbra, a **b** a balra fordulást, a fordulás mértékét órában adom meg.

To léves.fa

m l h l

end

A növekedés legyen évente duplázódás a hosszban, tehát exponenciális. A kétéves fa modelljén annak törzse az egyéves hosszának kétszerese, és a fa két oldalhajtást is hozott, amelyek olyanok mint az egyéves fák: a kétéves fa szerkezete, így tartalmazza az egyévest. Az eljárás így írható:

```
To 2éves.fa
m 2 b 1.5
m 1 h 1
j 3
m 1 h 1
b 1.5 h 2
end
```

A harmadik és ötödik sorba a "m 1 h 1" helyére írhattuk volna 1éves.fa.

A hároméves fa törzse ismét kétszeres hosszúságra növekedik, az oldalhajtások pedig kétévesek, alakjuk azonos a kétéves fával. A hároméves fa alakja tartalmazza a kétéves fát, amely tartalmazza az egyévest. Térjünk vissza a kétéves fa eljárására és fogalmazzuk meg azt általánosabban:

```
To 2éves.fa
m 2 * 1 b 1.5
1éves.fa
j 3
1éves.fa
b 1.5 h 2 * 1
end
```

Ezen 2éves.fa eljárás mintájára a hároméves fa szerkezetét leíró eljárás:

```
To 3éves.fa
m 2 * 2 * 1 b 1.5
2éves.fa
j 3
2éves.fa
b 1.5 h 2 * 2 * 1
end
```


Az életkor tehát a törzsnövekedés kitevője. A struktúra, az alapszerkezet a két eljárásban (2 éves.fá és 3 éves.fá) azonos, a törzsmagasság (ami az életkor függvénye) és az oldalhajtások alakja (ami azonos az egy évvel fiatalabb fával) változik.

Általános formában az eljárás tehát egyetlen paraméterrel megírható, amely az életkort és egyben a rekúzió szintjét jelenti. Ezt jelöljük n -nel!

Az egyéves fá törzse 1 (magyar) teknőclépés, kettő a nulladikon, a kétévesé kettő, azaz kettő az első hatványon, a háromévesé négy, vagyis kettő a másodikon, a kitevő tehát: életkor mínusz egy.

Mivel a LogoWriter nem tartalmazza a hatványozáshoz szükséges primitívet (alapszót) ezért megírtam ezt is:

```
to hatv :a :k
  legyen "i 1 legyen "h 1
  művelet
  op :h
  end
to művelet
  ha :i > :k [stop]
  legyen "h :h * :a
  legyen "i :i + 1
  művelet
end
```

Ezután az n éves fá rajzolására szolgáló eljárás:

```
To n.éves.fá
  m hatv 2 :n - 1 b 1.5
  n.éves.fá :n - 1
  j 3
  n.éves.fá :n - 1
  b 1.5 h hatv 2 :n - 1
end
```

Vegyük észre a szimmetriát az eljárás leírásában is!

Ez az algoritmus végtelen, ha elindítjuk a teknőc nulla, majd mínusz éves hajtásokat is rajzol, egészen addig, amíg nem telik meg valamelyik memória verem, ameddig a gépnek "nem áll meg az esze". Ezért a használható eljárásba egy feltételt is be kell írunk, amelynek

teljesülése esetén a rekurzív szinteken való lépések leállnak: az oldalhajtások részletezésekor nulla évest már nem kell rajzolni. Eljárásunk használható formája így:

```
To fa :n
  ha :n = 0 [stop]
  m hatv 2 :n - 1
  b 1.5
  fa :n - 1
  j 3
  fa :n - 1
  b 1.5 h hatv 2 :n - 1
end
```

A fák modellje a valóságos növekedés egyre több tényezőjét veheti figyelembe, szépen példázva ezzel a számítógépes szimulációt is. Fejlesszük tovább a fa eljárást!

Könczöl Tamás:
Egy újabb magyar oktatóprogram
Betűvarázs

A számítógép hasznos segédeszköz a tanítási-tanulási folyamatban (ismeretszerzés, alkalmazás, rendszerezés, gyakorlás, ellenőrzés), de nem helyettesítheti a többi módszert. Sok vonatkozásban azonban többletet, hatékony kiegészítést nyújthat.

A Betűvarázs 2.0 program olvasás és az idegen nyelvek tanításában szolgál jól azoknak, akik nemcsak a hagyományos eszköztárat kívánják bevetni, hanem a számítógépet is. A programot (és elődjét, a DOS alatt futó 1.0-s változatot) a visszajelzések szerint sikerrel használják az ország több iskolájában is, főleg nyelvi tagozatos iskolákban, továbbá a Gyógypedagógiai Főiskolán, az enyhén sérült gyermekek rehabilitációjában.

A Betűvarázs nyitott rendszerű, interaktív multimédiás program, mely az olvasást és az idegen nyelvek tanítását és gyakorlását segíti 9+1 nyelven, 7 témában, 250 képhez kötött szóanyaggal, olvasástanító módszertan alapján felépített ötfokozatú feladatcsoporttal. A nyelvek, a szavak és a képek egyaránt bővíthetők. A számítógép adta lehetőségek kihasználásával a program színes képekkel, hangeffektusokkal, zenével, emberi beszédhangokkal, azonnali és később részletes értékeléssel, segítséggel, nagy variációs lehetőségekkel, teljes egér- és billentyűhasználattal, állandó interaktív üzemmóddal valósítja meg, hogy a fent leírt követelményeknek megfelelően, és egyre inkább közelítsen a multimédiában megszokott környezetekhez.

Szinte minden paraméter, beállítási mód, megjelenési forma testre szabható, ezáltal hatékonyabb, érdekesebb, izgalmasabb is lehet az olykor unalmas gyakorlás.

Ez főleg az iskolába lépő első osztályos tanulókra igaz, mivel az iskolába lépés ténye is elég nagy terhet jelent nekik. Így azonban örömmel fognak a számukra játéknak tűnő tanulási, gyakorlási módokhoz hozzáállni, s szinte észrevétlenül sajátítják el a tananyagot.

Az elsős kisgyermek életkori sajátosságaiból adódóan, megfelelő előkészítéssel anyanyelvórán, differenciált foglalkozásokon is használható a program a tanítás színesítésére. Emellett az alsó tagozaton informatikaórán is elsősorban a játékos képességfejlesztés a cél. A képanyag miatt esetleg még környezetórán is elővehetjük.

Hardver- és szoftverigény: A program futtatásához már egy 286-os gép elegendő, min. 2 Mbájt RAM-mal mono- vagy színes VGA monitorral. A futtatáshoz még szükség van az MS Windows 3.1-re és 15 Mbájt szabad helyre a merevlemezén.

Munka a programmal

Előkészítés: Mielőtt a program használatát megkezdénénk, meg kell ismertetni a tanulókat a képanyaggal. Ez a képeken kívül a program DIA-SHOW funkciójával is történhet.

Névmegadás: A program indítása után meg kell adni a tanuló nevét, ami az elsősőknl még tanári segítséget igényelhet, bár sok gyerek a nevét vagy legalább a monogramját ilyenkor már be tudja írni. Ezzel a névvel kapja majd az értékelést.

A programkezelés megtanítása: A program az egerkezelést támogatja, és azoknál a részekenél, ahol betűt vagy szót kell beírni, minden megszokott szerkesztőbillentyű használható. Az eger használatának megkönnyebbítésére a nyomógombok nagyméretűek.

Feladattípusok

1. Egy képnek megfelelő szó kiválasztása a megadott háromból. Egy kép, az ehhez tartozó szó és még két szó jelenik meg a nyomógombokon - véletlenszerűen. Ezekből kell kikeresnie a képhez tartozót.

2. Egy képnek megfelelő betűhiányos szó kikeresése. Egy kép, és a hozzá tartozó betűhiányos szó jelenik meg az adatbeviteli felületen - véletlenszerűen. A szóból a betű is véletlenszerűen választódik ki.

3. Egy képnek megfelelő szótaghiányos szó kiegészítése. Egy kép a munkaképernyőn, a hozzá tartozó szótaghiányos szó pedig az adatbeviteli felületen jelenik meg. A nyomógombokon a hiányzó szótag, ennek megfordítottja (olvasás-módszertanilag fontos!), és még négy véletlenszerűen választott szótag más, véletlenszerűen választott szavakból. (A kétfégyű betűk helyesen fordítódnak meg, pl.: cs = macs - csam.)

4. Egy képnek megfelelő szótagok rendezése. A munkaképernyőn megjelenik a kép és a hozzá tartozó adatbeviteli felület. A keresendő szó szótagjai nyomógombokon helyezkednek véletlenszerűen. A képhez tartozó szót kell kirakni a szótagokból.

5. Egy képnek megfelelő teljes szó megadása. A kép és a hozzá tartozó adatbeviteli felület jelenik meg. A szó hosszát a betűk számának megfelelő ponttal jelzi (magyar nyelvnl magánhangzó- mássalhangzó szerint is jelöli.) Be kell írni a teljes szót.

A teljesítmény értékelése

Szemléletes és kellemes a program megjelenése a használója előtt.

- Hangjelzés, pontozás (jó: zöld, rossz: piros), rajzos értékelés feladatonként (jó: mosolygó arc, rossz: szomorkás arc: mindegyikből több).

- Minden 5. feladat után pihentető zene.

- A tanuló számára: a beállított feladatszám letelte után - százalékhatar alapján-rajzos (oklevél) és zenés értékelés.

- A tanár számára statisztika a következőkkel:

- A tanuló neve.

- A munka dátuma (a program lekérdezi a gépben lévő naptárt).

- Feladattípusonként: a jó, a rossz megoldások száma, százalékos eredmény.

- Összteljesítmény értékelése százalékban és szavakkal is: nem felelt meg, megfelelt, jól megfelelt, kiválóan megfelelt.

A bevezetőben vázoltak szerint kell figyelembe venni egy jó oktatóprogram céljait. Ez a termék remélhetőleg eredményesen használható segédeszköz lesz sokak számára. Igaz, hogy a választott terület nem a legkönnyebb része az oktatásnak. Az is elképzelhető, hogy szemlélet módosításra is szükség lenne az itt tanító kollégák egy részénél. A feladatcsoportok az olvasásitanítás minden lépését lefedik. A részletes értékeléssel pedig egész pontosan meg lehet állapítani, hogy melyik gyermeknek milyen pontokon kell fejlesztenie képességeit.

1. Die ...
2. Die ...
3. Die ...
4. Die ...
5. Die ...
6. Die ...
7. Die ...
8. Die ...
9. Die ...
10. Die ...
11. Die ...
12. Die ...
13. Die ...
14. Die ...
15. Die ...
16. Die ...
17. Die ...
18. Die ...
19. Die ...
20. Die ...
21. Die ...
22. Die ...
23. Die ...
24. Die ...
25. Die ...
26. Die ...
27. Die ...
28. Die ...
29. Die ...
30. Die ...
31. Die ...
32. Die ...
33. Die ...
34. Die ...
35. Die ...
36. Die ...
37. Die ...
38. Die ...
39. Die ...
40. Die ...
41. Die ...
42. Die ...
43. Die ...
44. Die ...
45. Die ...
46. Die ...
47. Die ...
48. Die ...
49. Die ...
50. Die ...
51. Die ...
52. Die ...
53. Die ...
54. Die ...
55. Die ...
56. Die ...
57. Die ...
58. Die ...
59. Die ...
60. Die ...
61. Die ...
62. Die ...
63. Die ...
64. Die ...
65. Die ...
66. Die ...
67. Die ...
68. Die ...
69. Die ...
70. Die ...
71. Die ...
72. Die ...
73. Die ...
74. Die ...
75. Die ...
76. Die ...
77. Die ...
78. Die ...
79. Die ...
80. Die ...
81. Die ...
82. Die ...
83. Die ...
84. Die ...
85. Die ...
86. Die ...
87. Die ...
88. Die ...
89. Die ...
90. Die ...
91. Die ...
92. Die ...
93. Die ...
94. Die ...
95. Die ...
96. Die ...
97. Die ...
98. Die ...
99. Die ...
100. Die ...

Informatika tantervek az országos adatbankban

Kőrösné dr. Mikis Márta
Országos Közoktatási Intézet
1051 Budapest, Dorottya u. 8.

I. Az informatika: új műveltségterület

Az informatika, az információs kultúra önálló tantárgyban történő elsajátításának igénye először a 80-as évek végén vetődött fel. Az 1995 őszen elfogadott **Nemzeti Alaptanterv**ben új, különálló műveltségblokk az informatika, amely két részterületre oszlik: számítástechnikára és könyvtárhasználatra.

A **számítástechnika** az informatikai eszközök közül elsősorban a számítógéppel, annak felhasználásával kapcsolatos tudást közvetíti. Új lehetőséget jelent a *problémamegoldásban*, segédeszközt jelent a többi *tantárgy tanulásában* is és hálózathoz kapcsolva forradalmi változást hoz a *kommunikációban*, az információkhoz való gyors hozzáférésben, azok kezelésében és továbbításában.

Az információs technika a hagyományos információhordozók (könyvek, folyóiratok, enciklopédiák stb) körét is kibővítette. Fontos, hogy mind a hagyományos, mind az új technológiákon alapuló *informatikai eszközök és információhordozók kezelésében* jártasságot szerezzenek a tanulók. A könyvtár használata *mindenfajta műveltség* megszerzéséhez nélkülözhetetlen, az ezzel kapcsolatos ismereteket a **könyvtári informatika** részműveltségi terület tanulása során sajátíthatják el a tanulók.

Az informatika területén *az ismeretek igen gyorsan változnak*, az eszközpark óriási mértékű fejlődésével párhuzamosan a konkrét alkalmazói ismeretek nagyon hamar elavulnak, cserélődnek. A napjainkban használatos eszközök, szoftverek a jövő évtizedben valószínűleg túlhaladottá, ismeretlenné válnak. Éppen ezért fontos, hogy a diákoknak olyan **rugalmas és konvertálható tudást nyújtsunk**, amelyekkel képesek lesznek ismereteik folyamatos frissítésére, nyitottá válnak az újdonságok iránt, azokat könnyen megismerik és a hozzákapcsolódó tudásanyagot elsajátítják.

A NAT-ban megfogalmazott tantervi célok, tananyag és követelmények alapján **az iskolák önállóan készítik el saját, helyi tanterveiket**, figyelembe véve az iskola sajátosságait:

- az oktatási-nevelési elképzeléseket,
- az iskola hagyományait,
- a személyi és tárgyi feltételeket,
- a tanulók szociális-kulturális környezetét,
- a szülők elvárásait stb.

Az informatikai ismereteket elsősorban az *informatikai eszközök rendszeres használata* során szerezhethetjük meg. Természetesen az is cél, hogy a korszerű technika, a multimédia a **teljes tanulási folyamatot** segítse: a matematika gyakorlásától kezdve az idegennyelv-oktatáson át - a képernyőn megjelenő távoli múzeumban bolyongva - egészen a művészetekig. A tantárgyi alkalmazás azonban önmagában nem elég. A legfontosabb alapismeretek (eszközhasználat, dokumentumkészítés, adatkezelés stb) elsajátítását *külön szaktanár és tanóra* kell, hogy segítse.

II. Országos adatbank

A helyi tantervek kidolgozásában, összeállításában az **Országos Közoktatási Intézet információs adatbankja** nyújthat segítséget. Az ország különböző részein (a *Megyei Pedagógiai Intézetekben* és ún. referenciaiskolákban) is hozzáférhető számítógépes adatbank *teljes iskolai tanterveket* éppúgy tartalmaz, mint az egyes *műveltségi területek*, *részműveltségi területek* tanterveit. A tantervi adatbank jelenleg fejlesztés alatt áll, állománya fokozatosan bővül.

Az informatika műveltségterülethez első lépésben nyolc különálló tanterv készül, amelyek mindegyike *mintául szolgálhat* a helyi tantervek elkészítéséhez.

A tantervi variánsok változatos iskolai igények kielégítését szolgálják:

1. Informatika az 1-10. osztályok számára

A tanterv szinte bármely, még *szerenyebb eszközfeltétellel rendelkező iskola* számára is választható lesz. A *kezdő szakaszban nem igényel külön tantárgyat*, az ismeretek az első négy évfolyamon a többi tantárgyban jelennek meg (pl. matematika, életvitel és gyakorlati ismeretek, anyanyelv, ember és természet).

2. Informatika az 5-10. évfolyamok számára

Ez a tanterv *közepes vagy "erősebb" iskolák* informatikai igényét elégítheti ki, sőt folytatást is ad a 11-12. évfolyamok számára.

Mindkét tanterv a teljes informatikai anyagot lefedi, ezen belül Számítástechnika és Könyvtár részmuveltségi területekre tagozódik.

3. Számítástechnika az 5-10. évfolyamok számára

A tanterv a *NAT minimumszintjéhez* igazodik, szerény, mindenki számára elsajátítható követelményekkel. Az informatikai eszközöket tekintve kisebb, kevésbé korszerű géppark is elegendő tanításához.

4. Számítástechnika az 5-12. évfolyamok számára

Ez a tanterv a régi szóhasználat szerinti számítástechnika tagozatos iskolák *mélyebben érdeklődő diákjai* számára készül, emeltebb szinten. Bár a NAT csak a 10. évfolyamig rögzíti a tanulivalókat, *folytatást ad a 11-12. évfolyamok* szakirányú érdeklődésű tanulói számára is.

5. Számítástechnika a 7-12. évfolyamok számára

A tanterv a *hat osztályos képzést folytató*, az informatika iránt nyitott, azt a minimumnál viszonylag "komolyabban" oktató-alkalmazó gimnáziumoknak nyújt segítséget, jól összeilleszthető lesz (óraszámában is) a 7.-tel jelölt *könyvtári ismeretek* tantervvel.

6. Számítástechnika a 7-8. évfolyamok számára

A NAT szerinti tanítás 1998-ban az 1. és 7. évfolyamokon kezdődik. Mivel informatikát már a NAT szerint az 5-6. évfolyamokon is tanítani kellene, a megelőző évfolyamok anyagát össze kell sűríteni. Ez a tanterv a belépő 7. osztályok tanulói számára *összevomma tartalmazza az 1-6. évfolyamok előzetes alapismereteit* és rögzíti a tudnivalókat a folytatáshoz. A 8. évfolyam végére ugyanolyan szintre jutnak el a tanulók, mintha az összes előző évet a NAT szerint tanulták volna.

7. Könyvtári ismeretek a 7-12. évfolyamok számára

Ez a tanterv szintén a *hat osztályos gimnáziumoknak* készült, és a PROFIL Tantervi Adatbank 1995-ös változatában már szerepelt. Így a NAT igényeihez illeszkedve kisebb átdolgozás után kerülhet az új adatbankba. Alkalmas lesz a NAT szerint haladó, 1998-ban *belépő 7. évfolyamok számára* is.

8. Könyvtári ismeretek az 5-12. évfolyamok számára

A tanterv a NAT szerint ajánlott 5. évfolyamon vezeti be a könyvtári ismereteket, ezt egy *általós, közepes szinten* a kötelező 10. évfolyamig viszi, majd a *tantervi folytatást* is közli a 11-12. évfolyamok számára.

A tantervek kidolgozói középiskolai és általános iskolai számítástechnika tanárok, illetve könyvtár szakosak, akiknek - az elmúlt évek során - "tantervfejlesztői múltjuk" is van (Pl. OPI kísérleti tantervek, modul-tantervek, PROFIL-tantervek stb.), az Informatika és Számítástechnika Tanárok Egyesületének ajánlásával is rendelkeznek.

III. Informatikát tanítani - informatikával tanulni

A két fogalom összefügg. A minden diák számára fontos, az általános műveltséghez szükséges ismereteket **az informatika, mint tantárgy** közvetíti. A tanulási folyamat egészét, az ismeretszerzést, az egyéni ütemű képességfejlesztést, a gyakorlást, a szórakozást **az informatikai eszközök alkalmazása** motiválhatja, segítheti. Hatalmas és pedagógiailag még kiaknázatlan lehetőséget jelent *a világot behálózó számítógépes rendszerek* információinak iskolai megjelenése. A jövőben ennek a pedagógiai gyakorlatba való átvitele bizonyítja majd igazán, hogy *informatikára pedagógusnak és diáknak egyaránt szüksége van!*

Internet és az oktatás

(vázlat)

Bevezető

Az internet kezdetei – mint nagyon sok más műszaki fejlesztés – a katonai kutatásokra vezethetők vissza. A 60-as 70-es években indult meg a kutatás, és a 80-as évekbe már működött az ARPA nevű Unix alapú rendszer. Ennek lényege az volt, hogy egy esetleges nukleáris támadás esetén is működjön a nagy távolságokat áthidaló számítógépes rendszer. Ezt úgy próbálták elérni, hogy a gépeket hálószerűen kötötték össze. Az adategységek (data-gramok) automatikusan irányítása révén a rendszernek akkor is működnie kellett, ha a hálózat egyes elemei működésképtelenné váltak. Technikai alapján az ún. TCP/IP (Transfer Control Protocol / Internet Protocol) alkotja. Ennek lényege a különböző rendszerű, platformú gépek egységes hálózati kommunikációja. Ezt a rendszert később felhasználták az emberek egymás közötti kommunikációjára is, aminek kapcsán létrejött az elektronikus levelezés, az e-mail. Kialakultak azok a több embert érdeklő témák, amelyeket mindenki számára nyitottá tettek - létrejöttek az elektronikus hirdetőtáblák, vitakörök, ... Megvalósult a távoli gépek erőforrásainak használata, a Telnet, és az állomány átvitel az FTP. Ahogy bővültek az információk, egyre inkább az lett a kérdés, hogy az adatok milliói közül hogyan találjuk meg a számunkra szükségeseket. Elkészültek az első kereső rendszerek, melyek segítségével bizonyos kritériumok alapján leszűrhető az adatok köre.

Az Internet

Az Internet világméretű számítógép hálózat, amely különböző számítógép hálózatok között egy egységes „nyelv”, az Internet Protokoll segítségével működik. Nem kötődik közvetlenül semmilyen céghez, alapvetően önkéntes alapon működik. Hálózati szakértők felügyelik a munkát, és az általuk elfogadott ajánlások, „szabványok” (RFC) szabályozzák a rendszert, melyet a legfontosabb ilyen szervezet, az ISOC (Internet Society) koordinál. A hálózatokat független szolgáltatók üzemeltetik, melyek a közöttük lévő kapcsolatokkal együttesen alkotják az Internetet. Az információs objektumokat (adat, hang, kép, video, online video) A hiperlink technika révén úgy használhatjuk, hogy azokhoz nem helyük, hanem tartalmuk szerint tudunk hozzáférni. A kapcsolatok egymás adatainak elérését, utalásokat, ún. linkeket (másik adat-bázisra mutató elérhetőség) is jeltenek.

Az Internet és az oktatás

Az Interneten folyó információ csere (pl.: levelezés, távoli bejelentkezés-végrehajtás, fájl transzfer...) egyik legfontosabb grafikus alapú innovatív szolgáltatása a World Wide Web (WWW). A WWW az adatokat felhasználóbarát grafikus platformon képekkel, hangokkal, animációkkal egészítheti ki. Az oktatásban ez, az adatbankok elérése és az elektronikus levelezés lehet a legfontosabb célterület.

Az Internet (NET) kapcsolódása az oktatáshoz a közeljövő egyik legfontosabb oktatás-informatikai fejlesztési feladata. Ez a világméretű hálózat lassan az élet minden területéről tartalmaz információkat. Rajta keresztül a felhasználók, pl.: a diákok, tanárok egymással, intézményekkel is kapcsolatba tudnak lépni, akár többféle módon is. Hasznosságának megindokolásául néhány utalás:

- * egységes oktatási nyilvántartási rendszer létrehozása, mely a NET-en keresztüli adatközléssel napra, sőt percre pontos lesz;
- * információs adatbankok létrehozásának lehetősége az oktatási, a kulturális, társadalmi, gazdasági, és egyéb alternatív célú területeken, melyek közvetlenül vagy közvetve az oktatáshoz kapcsolódnak;
- * a magyar szerzők műveinek a NET-en keresztüli elérhetősége;
- * kapcsolat kiépítése és intézményekkel;
- * elektronikus levelezés;
- * hirdetés;
- * versenyek, vetélkedők a NET-en;
- * ...

Ennek infrastrukturális alapját építve a már meglévő feltételekre és a saját erőforrásokra a most indult MKM Országos iskolai modernizáció program fogja létrehozni. Ezt a társadalmi Informatikai Koordinációs Iroda szervezi. Olyan pályázati rendszerekből álló sorozatot dolgoztuk ki, amelynek alapelve a **regionalitás és a technológiai transzfer**. A megoldásokat régióként együttesen működő rendszerben képzeljük el;

- pl.;
- általános iskola,
 - középiskola,
 - önkormányzat,
 - könyvtár,

együtt pályázzanak. Az ütemezés a technológiai- és tudástranzfer elvén alapul, azaz a már jelenleg is fejlett iskolák, kapjanak bérelt vonali vagy ISDN hozzáférést, ennek feltétele a környezetében lévő kisebb iskolák beszerzése és továbbképzése legyen.

Cél az Internet szolgáltatások elérhetővé tétele a tanulók, tanárok és az intézmények számára, és bizonyos oktatási információs szolgáltatások beindítása és elérhetősége részükre. A távolabbi cél az igények alaptantervbe foglalása lehetne, hogy bizonyos szinten mindenki hozzájuthatna, használhatná.

Emellett lehetőséget adna a pedagógus fejlesztő munka segítségére, közös, NET-en keresztüli fórumot létrehozva erre. Ez gyorsabb, hatékonyabb módja lehetne az információ áramlásának.

Könczöl Tamás

Environment for Environments: New Metaphor for Logo

Andrej Blaho, Ivan Kalaš, and Monika Máfušová¹

Abstract

This paper characterises some fundamental features of a tool to serve as a flexible and powerful environment in which students, teachers, and developers can design creative software applications. Although traditional Logo philosophy claims to provide such powerful tool, real Logo implementations could not fully support it because of many technical drawbacks. New implementation of Logo, *Comenius Logo for Windows*, is described with respect to listed features, which constitute *Environment for Environments Metaphor*. The authors believe that recent advances in technology, interfacing and educational research incorporated into Logo make it possible to develop open-ended learning environments with active involvement of their designers and users - preferably children in both of these roles. An example of cooperative problem solving environment developed in Comenius Logo is presented.

Keywords

open-ended learning/teaching environments, Logo, cooperative problem solving

Three (Plus One) Approaches

There are many different ways how to introduce "information and communication technologies" (ICT) into schools, into families, into everyday life of students and their teachers, of children and their parents. In our paper we have a double intention: we describe different approaches that influenced our conception of new Logo implementation from the educational point of view, and we discuss new metaphor for Logo that resulted from enhancing it with some strong tools common to recent Windows applications. As we have already introduced *Comenius Logo for Windows*² elsewhere, see [1] and [2], here we add to the description of its *'Environment for Environments'* orientation.

Although there are considerable differences in cultural backgrounds of the countries all over the world (expressed, for example, in their different educational systems), we believe that

¹ Department of Informatics Education, Faculty of Mathematics and Physics,
Comenius University, 842 15 Bratislava, Slovak Republic,
tel/fax: + 42 7 724 826, Internet E-mail: kalas@fmph.uniba.sk

² Titled *SuperLogo voor Windows*, this Logo environment is published in the Netherlands by A.W.Bruna Uitgevers B. V., Utrecht

recent changes in education - especially with respect to ICT - have much in common and provide rich basis of experience for all current research projects for Central and Eastern European countries. In designing and developing *Comenius Logo for Windows* we have been extensively influenced by three (plus one) approaches which may be characterised (stressing only those aspects that are relevant to the topic) as:

- **Western European approach.** ICT is a compulsory and important agent in preparing pupils for their future occupations. Introduction of computers and other IT is (more or less extensively) supported by governments. ICT is considered to be a tool for other subjects, for cross-curricular projects, for national and international communication and cultural transfer. In many cases there is no obligatory *Informatics* (or *Computer Science*) course at the secondary level. Quite often, programming skills are not taught at schools at all. The importance of ICT's integration into teachers education is ranked very high, see [3], [4].
- **Central and Eastern European approach.** Although there is a growing tendency to integrate ICT into other subjects - as reported by [5] - the main emphasis still lies in more traditional *Informatics equals Programming* approach. Informatics is presented as a separate subject in general education, quite often teaching programming as ends in themselves. In spite of this, programming is very popular activity among young people who sometimes become surprisingly proficient programmers, as documented by the most recent placing of Comenius University team at The Final of ACM International Collegiate Programming Contest, Phoenix, Arizona, 1994. We realise that *teaching programming as ends in themselves* must be changed in our countries but we also believe that programming skills are important and fruitful in developing other skills like logical mind, problem-solving skills, and developing new ways of structured thinking. Therefore, educators in Central and Eastern Europe are trying to integrate ICT not as a substitute of more traditional informatics approach but as another paradigm.
- **The MIT Epistemology and Learning Group's research**, namely their ISD project, see [6]. "*Instructional Software Design Project ... offers a Constructionist vision of the use of computers in education. In a Logo-based learning environment, a fourth-grade class was engaged ... in the design and production of educational software to teach fractions,*" see page 19. "*The evaluation revealed greater mastery of both Logo and fractions as well as acquisition of greater metacognitive skills.*" Open-ended educational philosophy is applied here to provide children with an environment in which they design, build, construct - they simply **learn by doing** themselves. Within the context of what we have already said above, an approach of *learning programming as a means to different ends* is very inspiring for us. We feel the topic of teaching fractions is not of high importance here. Students as software designers are engaged in expressing their individuality and creativity, they are learning by creating, learning by teaching. "*By becoming designers of instructional software, the students gained distance and perspective in two senses. In the first place, they were dealing not with the representations themselves, but with a Logo representation of the representations. Moving between representations was subordinated to programming good examples of representations. Secondly, the students programmed*

not for themselves, but for others. They had to step outside and think about other children's reactions," see [6], page 43. The idea of turning school PC labs into 'design studios' coincides with our experience, that in the field of educational software, "the people who are having the most fun, and are learning the most, are the software designers and programmers."

- Sometimes underestimated by educational policy makers, planners and curriculum designers, the **fourth source of inspiration** (quite different from three previous ones) is the world of home computers and some entertainment software material. Boys and girls (and their parents) quite often do rather serious work with their computers outside the school. Educators introducing "gentle courses of text processing" - just to take an example - might be surprised quite soon by rather deep knowledge of a generation to enter their classrooms. Some excellent "out-of-school" courses of using ICT unbiased by official educational flavour are becoming very popular and, surprisingly, even enter schools from the backyard, see [7], [8], [9], see also SimCity simulation game or Carmen SanDiego software package used in regular geography education, etc.

From this broad and diverse spectrum of influences we have accepted several principles which - if implemented properly - should provide extremely flexible tool to meet rapidly changing needs of children, students, teachers, and designers (if, by chance, these are not all the same).

'Environment for Environments' Metaphor

In nearly all recent Logo implementations the central concept of work, design or development is a **project**. In *Comenius Logo for Windows* a project is a complex structure capturing all constituents of the user's work - to 'work' here means either to develop or to use a project. Let us concentrate on the development side of the coin: to develop a project means to create our own turtles, to create or import their shapes, to define their behaviour in procedures, variables, and words with properties, to create or import the contents of a graphics screen, and to customise the interface³.

If the underlying Logo is a complex tool with powerful developing and debugging features, we tend to call it *an environment*. If the resulting Logo project is an application with complex conception (either educational or not), nontrivial interactions, behaviour and structure, we also tend to call it *an environment*. Thus, if the underlying Logo makes it possible

³ More specifically, *Comenius Logo for Windows*' project (saved in a .LGP file) is a complex structure preserving complete information about:

- all turtles (that is, all facts concerning each existing turtle),
- all user-defined *Memory* objects (if we check this option while saving the project),
- complete state of the *Main Window*, that is, how the screen is divided between text and graphics screens, the actual contents of the graphics screen (if we check this option while saving the project), size of the graphics screen, its background color, text colors, whether the *Speed Bar* is displayed or hidden, whether the *Buttons Chooser* is shown or not and what are the texts defined for its buttons, whether the project has been saved with the *Native Language Version* on or off,
- if there is an image variable *ICON* (of exactly 32 x 32 size) defined at the time of saving the project, its value is converted into an icon which is used as the project's icon in the *Demo Projects Window*.

to develop such environments on their own, we call it *an environment for environments* - a tool powerful enough to give birth to other complex applications⁴.

We believe that cumulating one word in this way is not an end in itself. On the contrary, it coincides with our understanding of Constructionist approach in the use of computers in education: to develop means to use (a tool, an environment), to use might mean to develop (an environment). And more: to learn can mean to teach, and vice versa - not only in the sense of [6]. In [1] we argue that the roles of students, teachers and educational software developers are not so different as they might seem to be: learners want to develop, teachers are often developing their own applications and microworlds. Or they are tailoring existing material. Some teachers (hopefully all teachers) are learning to use ICT or new educational environments. Developers are often university or secondary teachers. Future teachers (our students) have to develop Logo-based open-ended environments and try them in our experimental fifth-grade class. We find an *Environment for Environments metaphor* to be a solution to cope with these partly mixed up roles.

Which features classify a tool to be an environment for environments?

- It must be an environment common to learners, teachers and developers. Only in this way the different roles can be shared. A teacher who wants to design an application will appreciate that it can be realised within the same environment he is communicating to his pupils in 'traditional' (Logo) course. Students will appreciate that they are using a 'real' software powerful enough to serve as a developing tool.
- It must be of an 'Onion Architecture': keeping traditional Logo philosophy, several new coats must be put on an elementary core part that provides for introductory Logo programming activities. By using additional coats one can design more complex applications. However, these higher levels must not be intrusive, so that Logo novices do not see them and do not need them. 'Onion Architecture' stresses **extendibility** and **high flexibility** within one environment: for example, our students - future teachers of mathematics and informatics meet *Comenius Logo* three times in their curricula, see [2]: as a learning microworld for children, as an environment for investigating advanced *data structures* and *symbolic computations*, and as a tool for developing and evaluating educational software.
- It must be powerful enough to support the development of broad palette of diverse projects: for different ages, for different purposes and different target groups, close to Logo turtle microworld and others that are far from it, more programming oriented or completely 'programming-free', educational, presentational, interactive, dynamic or static, closed or open-ended ... etc.
- It must be (rather) easy to **modify** such environment (not very easy, which is simply impossible). Teachers and learners often want to alter a project to better fit their needs, to customise it; sometimes they want to modify it because children are soon getting bored.

⁴ We illustrated such environments developed in *Comenius Logo for Windows* in [1], [2], [10].

- For any software to be modifiable, it also must be highly **readable**. Such quality results from the tool itself and from the way it is used⁵. If we want to support collaborative work of the users/designers (collaborative work in its broad sense: for example, to modify an environment means to collaborate with its original author), we must reflect on how easy it is to debug, transform, optimise, and understand someone else's code. Fortunately, Logo is a programming tool, in which a complete **source code** is always handed out with a project. Otherwise one finds himself in a situation that a high 'wall' separates the tool used to develop and the resulting product.

Comenius Logo for Windows

In following paragraphs we illustrate how our new implementation of Logo strives for *Environment for Environments* qualification.

It is complete Logo

The implementation is fully based on traditional Logo educational philosophy and traditional turtle microworld. Making use of this central part of its 'onion architecture', both introductory Logo projects and more sophisticated programming projects can be developed.

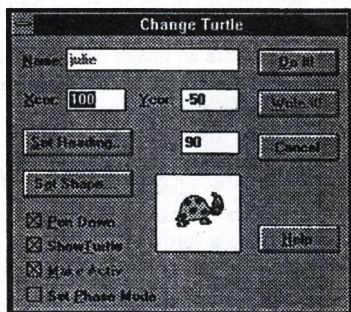
Interface is designed to be modified and customised

The *Main Window* consists of two, possibly overlapping parts - *graphics screen* and *text screen*. Whole communication with the application is done through menu commands, or through a *Speedbar* displayed across the top of the application window, just below the menu bar, to provide quick mouse access to some operations. It can be hidden or displayed back on the screen either by menu commands or by Logo language commands. As we already stated above, the central means of preserving and reloading our work is a **project**, which completely defines the state of the screen, all turtles and all *Memory* objects (i.e. procedures, variables and words with defined properties). Through the *Memory Window* any of these objects can be examined, created or redefined - this window provides an alternative to view and manipulate directly Logo memory, where all objects are stored.

Turtle: Change Me...
Turtle: Pen Down
Move Turtle
Load Screen...
Save Screen As...
Graphics Screen
Set Screens...
Clear Graphics

Whenever we click by the right mouse button into the graphics screen, a menu is displayed to list all possible operations that can be carried out on that particular place. Let us examine one of them: if the mouse click takes

place at a turtle and we choose a *Turtle: Change Me* command, *Change Turtle* dialogue box opens.



⁵ We believe in two principles here: "Let you read my project" game and "Let me design my project in such a way you can read it" discipline.

It characterises a lot of basic facts about the turtle - its name, home position and heading, state of its pen etc. In this way, an easy access to examine (and modify) someone else's turtle (turtles) is provided.

The *Memory Window* together with all other tools to create, view and modify a turtle (turtles) and other customisation features make it possible to understand, implement, modify, collaborate on, or share projects among their users/developers.

Multiple Turtles

We can create any number of turtles (with no practical limitations) or erase them dynamically either during the development of the project or later when we use it. For any of these turtles we can define its own behaviour. All turtles live on the screen together, either interacting or ignoring each other. We can use them both in a traditional 'turtle microworld' way, or in dozens of peculiar ways. In the figure to the right there are five turtles that can be dragged along the screen and probably form a correct word. (Here we are making use of a feature to be described soon - any of *Comenius Logo* turtles may have different shape, either ready-made or created by a user.



Shapes of the Turtles are 'First Order Data Types'

One of our most ambitious goals within *Comenius Logo for Windows* has been to improve limited abilities of previous Logo implementations to handle bitmap graphics. In traditional implementations, turtles draw pictures consisting of lines only. If we compare this with the abilities of more recent software packages to move full color pictures along the screen, we understand children who want to do the same.

In *Comenius Logo for Windows* we can handle full screen graphics and **images**. An image is a series of small bitmap pictures (not bigger than 64 x 64 pixels) called phases or frames. To draw an analogy, phases of an image are letters of a word. And this analogy is very literal: we work with images by traditional Logo primitives like *FIRST*, *BUTFIRST*, *LAST*, *WORD* etc. An image can be bound to a variable as its value by *MAKE* just like words and lists are.

We can create an image either in *Comenius Logo Image Editor* or by a Logo program and copy it from the screen into a variable by *GETIMAGE* procedure.

Following is a simple program to draw the vanes of a windmill: we repeat the same *draw.4.vanes* command for six times, always cleaning the screen, turning a turtle by 15 degrees, drawing 4 vanes and "taking a picture of them". The results are concatenated into one "word" or image.



The results are concatenated into one "word" or image.

```

to draw.a.vane
  fd 7
  repeat 2 [fd 20 rt 90 fd 6 rt 90]
  pu rt 45 fd 2 fill ; the vane is colored
  bk 2 lt 45 pd
  bk 7
end
to draw.4.vanes
  repeat 4 [draw.a.vane rt 90]
end

```



As far as any image can be set to a turtle as its **shape**, we now easily create a "real" windmill, i.e. a turtle with the shape of six different positions of the vanes. If we make the turtle turn around, one position is automatically replaced by another, which results in nice animated movement of the mill. The ease with which different kinds of movement are done in this way is surprising and hardly comparable with any other developing tool for Windows applications.

Complete Data Structures

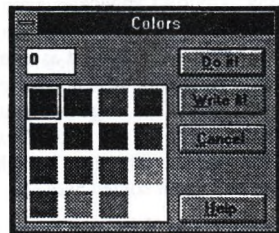
Comenius Logo supports all Lisp-like data structures extended by images (introduced above). That is, there are three types of data atoms: words, numbers and images. Like in every complete Logo implementation, these atomic data structures can be combined to create lists of atoms and lists of lists. Such advanced recursive data structures are important when developing more complex environments.

Mouse Interactions

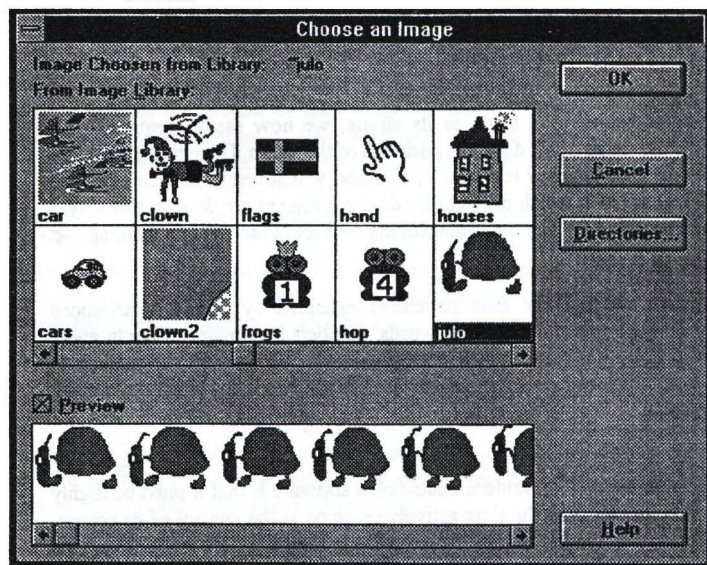
One of the most important features of modern educational software is that it must be highly interactive. This means that its user is all the time actively involved in the control of its course. In Windows, majority of all interactions is done by mouse. Therefore we are using it on two levels: communication with *Comenius Logo* environment is easily done by mouse, i.e. a user is using his mouse to design/develop his project easily. Besides, his Logo project can also react on clicking or dragging by mouse - his project can be fully controlled by mouse.

Choosers & Buttons

We are trying to prevent children users from seeing too many error messages on the screen. Instead, we prefer to contextually understand their intention and provide them with some help. For example, students often omit inputs for Logo procedures, which results in an error. Our solution is to open a special graphical menu instead of printing an error message and let a user choose one of possible inputs. If they type in *SETPENCOLOR* command without any further specification of the color, for example, a **Colors** chooser is displayed. In it a user just chooses one of the colors.



Actually, to prevent error messages is not the only reason for introducing choosers. We also believe that this is a right way of interactive **interfacing**: let a user choose from graphical menu of all options. Specially with young users this approach makes it possible to hide away many technical details - for example, children are able to work with files containing different turtle shapes (a library of pre-defined images) thanks to such abstraction: these are just shapes not displayed as a list of file names but simply by their contents.



Besides all choosers, there is another feature to enhance the interface and control: a window of empty buttons is available to be used either during the design of the project-environment or later by its user. It is possible to define any texts to appear on the buttons and any behaviour (defined by a Logo program)

to be taken when a button is clicked on by the mouse.

Cooperation with Other Windows Applications

Many Windows applications are supporting mutual collaboration. For developing a project in *Comenius Logo* one can prepare a full screen graphics in Paintbrush or KidPix or any other graphics editor and import it as a new contents of his/her Logo graphics screen, one can design new turtle shapes in any of these editors, one can import texts to become parts or complete definitions of his/her Logo procedures, one can import text to be displayed as an assignment for students to be solved in the *Problems Window* etc. Any kind of such cooperation can also go in the opposite direction: it might be interesting to design a full screen graphics by a Logo program and use it somewhere else.

Logo Projects as Windows Stand-Alone Applications

Any *Comenius Logo* project can be installed in *Comenius Logo Group Window* as a 'stand-alone application'. If the project contains an image variable



named *ICON*, its value can be used as the application's icon. The title of the project will become its title. However, the application will be functional only if *Comenius Logo* "engine" is present. If we then start this stand-alone application, a *Run Time Comenius Logo* runs the project as a one-purpose program.

Multimedia Features

We find multimedia features very important for powerful development tool in Windows. It is extremely expressive especially for young users/developers to combine graphics animation with funny sounds, with video sequences etc. In accordance with [6] we believe that such options are even more attractive for the developer of the project than for its user - in a traditional scheme. If the roles are unified, if using means developing, the developer of the project-environment is having the most fun.

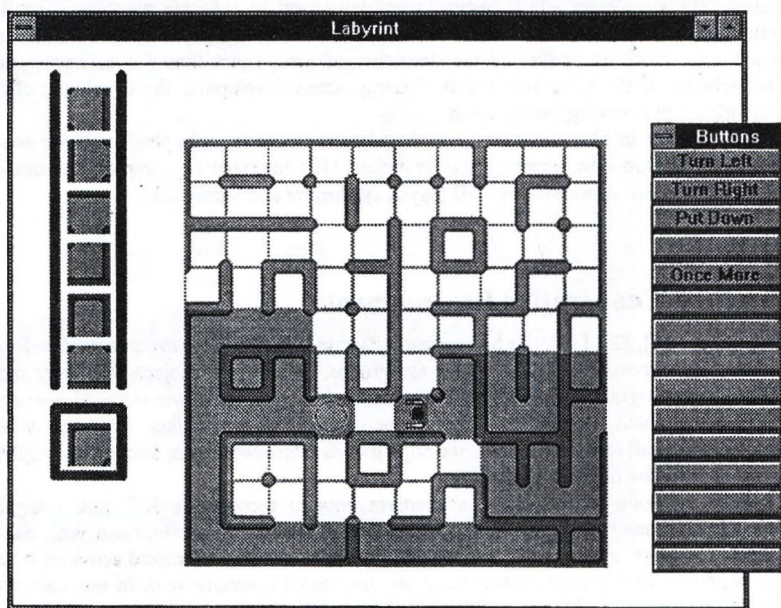
In current version of *Comenius Logo for Windows* we support only playing *.WAV* sound files. However, we are now implementing complete MCI interface (a complete multimedia language to support collaboration with CD player, synthesiser and video disk).

Labyrinth: a Cooperative Environment

Elsewhere, see [1], [2], [10], we have already illustrated different environments developed in *Comenius Logo environment* for different age groups, for different subjects, closer or more distant from Logo programming itself. Here we want to present some newer material, namely a cooperative game which we have used in our experimental fifth-grade class. Although we are not going to present all technical details, we hope that its educational goal and the atmosphere in the classroom will be obvious.

We believe that one of the nicest alternatives how to incorporate ICT into everyday classroom life is to use it together with other tools - in reasonable combination with paper, pencils, books, scissors, glue or ruler. We are often trying to design such mixed activities where computer is present only because some things are done more naturally by it. In our Labyrinth environment children are divided to form teams of four persons - two of them are sitting by the computer and other two are sitting at the desk in the next room where there are no computers at all. Neither of these pairs can solve the problem by itself - each pair receives only partial information - each new piece of information for a pair makes it possible to advance one step towards the final solution, so that they can now send another piece of information to the other pair to make them advance a little bit as well, etc. The messages are written on a sheet of paper and are carried from one pair of a team to the other by a "postman" - a human network of cooperating children and a computer is thus formed.

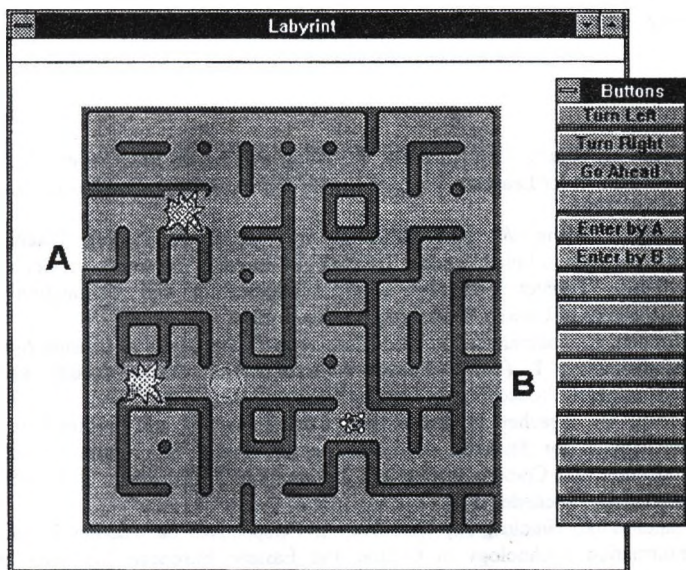
The activity has two stages: in the first one both pairs of children are presented with a labyrinth. However, it is not complete - the pair with a computer has to construct it from many small square parts by navigating a robot (a turtle in disguise) in a turtle-like way. To do so they press *Turn Left*, *Turn Right* and *Go Ahead* buttons. The pair of children without a computer constructs the same labyrinth printed on a sheet of paper in a puzzle-like way by fitting small cut-outs together.



After a while both pairs have the same labyrinth in front of them - now the second stage of the activity starts. In the middle of the labyrinth there is a treasure. The pair without a computer chooses one of two possible entrances, A or B, and prepares a path, i.e. a sequence of turtle-like commands on a paper, for example:

Enter by A
Turn Left
Go Ahead
Turn Right
Go Ahead
Go Ahead
Go Ahead
Turn Right
Go Ahead
Go Ahead
Go Ahead

where Go Ahead means to move one "square" ahead, i.e. to make one "step".



Children are constructing the path by moving and turning a small paper turtle through their paper labyrinth. Now the sequence is sent to the other pair and is interpreted by them with the help of their computer by pressing corresponding buttons. However, sometimes a problem occurs - when the turtle reaches certain squares of the labyrinth, they break down. If this happens (and it does quite often), the whole sequence is sent back with a note saying which part of it has been carried out successfully and after which step a break happened. The first pair must decode the position of an obstacle and try to prepare another path that avoids all broken squares. In this way the messages go to and fro until the correct path is found out.

Our goals in this activity are:

- to encourage cooperative work within a class, within a small group,
- to create a combined activity, in which a computer plays only one of many roles,
- to combine different representations of the same problem, to involve different ways of interpreting the same description of a process (a path of a turtle),
- to share and combine information coming from more sources,
- to develop introductory Logo programming skills - paths through the labyrinth are always described by a sequence of Logo-like commands. These sequences are both constructed and interpreted by children,

- to develop general problem-solving skills.

References

- [1] Blaho, A., Kalaš, I., Tomcsanyi, P.: Comenius Logo: Environment for Teachers and Environment for Learners, Proc. of the 4th European Logo Conference, Suppl. pp. 1 - 11, Athens, 1993
- [2] Kalaš, I., Blaho, A.: Not All Birds Are Turtles, Developing Teaching / Learning Environments in Initial Teacher Training at Comenius University, in: Collis, B., Nikolova, I. (eds): Teacher Education and Communication and Information Technologies, Proceedings of Unesco Workshop, Enschede, 1994
- [3] Moonen, J.: Communication and Information Technologies as Change Agents, in: Collis, B., Nikolova, I. (eds): Teacher Education and Communication and Information Technologies, Proceedings of Unesco Workshop, Enschede, 1994
- [4] Cornu, B.: Teacher Education and Communication and Information Technologies: Implications for Faculties of Education, in: Collis, B., Nikolova, I. (eds): Teacher Education and Communication and Information Technologies, Proceedings of Unesco Workshop, Enschede, 1994
- [5] Stanchev, I., Nikolov, R., Nikolova, I.: Perspectives on Teacher Training Related to Information Technology in Central and Eastern European Countries, in: Collis, B., Nikolova, I. (eds): Teacher Education and Communication and Information Technologies, Proceedings of Unesco Workshop, Enschede, 1994
- [6] Harel, I., Papert, S.: Software Design as a Learning Environment, in: Harel, I. (ed.): Constructionist Learning, MIT Media Lab, pp. 19 - 50
- [7] Stuur, A.: MS Word Voor Kinderen, A.W.Bruna Informatica, Utrecht, 1993 (in Dutch)
- [8] Stuur, A.: Windows Voor Kinderen, A.W.Bruna Informatica, Utrecht, 1993 (in Dutch)
- [9] Stuur, A.: Programmeren Voor Kinderen met SuperLogo, A.W.Bruna Informatica, Utrecht, 1994 (in Dutch)
- [10] Kalaš, I., Blaho, A.: Thomash the Clown's Circus: Order in Action in Picture Languages, Pre-conf. proc. of IFIP Open Conference: Informatics and Changes in Learning, Gmunden 1993, pp. DI 1 - 4

Válaszolj az alábbi kérdésekre az "igen" illetve a "nem" szavak (i és n betűk) beírásával, aszerint, hogy mi jellemző rád!

1. Nyugtalanít-e, ha a tanár azt mondja, hogy kérdéseket fog feltenni neked, hogy lássa, mennyit tudsz?
2. Nyugtalan vagy-e amiatt, hogy rossz tanulmányi eredményeid miatt esetleg megbuksz?
3. Amikor a tanár arra kér, hogy állj ki az osztály elé és olvass hangosan, tartasz-e attól, hogy néhány csúnya hibát csinálasz?
4. Amikor a tanár azt mondja, hogy ki fog hívni néhány fiút és lányt számtanpéldát megoldani, reméled-e, hogy valaki mást fog kihívni és nem téged?
5. Amikor a tanár azt mondja, hogy most feleltetni fog, gyorsabban kezd-e dobogni a szíved?
6. Szoktad-e néha azt álmodni, hogy iskolában vagy és nem tudsz válaszolni a tanár kérdéseire?
7. Matekórán szoktad-e úgy érezni, hogy más osztálytársaid jobban értik az anyagot, mint te?
8. Este, az ágyban szoktál-e azon gondolkodni, hogy milyen lesz a másnap napod az iskolában?
9. Amikor a tanár arra kér, hogy írd a táblára az osztály előtt, szokott-e kicsit remegni a kezed?
10. Magyarórán szoktad-e úgy érezni, hogy más gyerekek az osztályban jobban értik az anyagot, mint te?
11. Amikor otthon vagy és matekpéldán gondolkozol, elfog-e a félelem, hogy rosszul fogsz válaszolni, ha a tanár felszólít?
12. Gondolod, hogy többet nyugtalankodsz az iskola miatt, mint más gyerekek?
13. Ha beteg vagy és hiányzol az iskolából, félsz-e attól, hogy amikor visszamégy, rosszabbul fogsz felelni, mint a többiek?

14. Szoktad-e néha azt álmodni, hogy az osztálytársaid meg tudnak csinálni olyan dolgokat, amiket te nem tudsz?
15. Ha otthon vagy és a magyarórára gondolsz, félsz-e attól, hogy rosszul fogsz szerepelni?
16. Amikor a tanár azt mondja, hogy feleltetni fog, érzel-e valamilyen furcsa érzést a gyomrodban?
17. Ha nagyon rosszul felelsz, legszívesebben sírnál-e, ha nem lennél már ilyen nagy gyerek?
18. Szoktad-e néha azt álmodni, hogy a tanár mérges, mert nem tudod a leckét?
19. Félsz-e az iskolai feladatoktól?
20. Sokat izgulsz-e, mielőtt egy feladatot megoldasz?
21. Sokat izgulsz-e, mialatt egy feladatot megoldasz?
22. Miután megoldottad a feladatot, izgulsz-e, hogy hogyan sikerült?
23. Szoktad-e néha azt álmodni, hogy rosszul oldottál meg egy feladatot az iskolában?
24. Amikor dolgozatot írsz, szokott-e remegni a kezed?
25. Amikor a tanár azt mondja, hogy mindenki oldjon meg egy feladatot, tartasz-e attól, hogy rosszul fogod megoldani?
26. Amikor egy nehéz feladatot kell megoldani, előfordul-e, hogy elfelejtesz valamit, amit előtte még tudtál?
27. Szoktad-e gyakran kívánni, hogy bárcsak ne izgulnál úgy, amikor felelsz?
28. Amikor a tanár azt mondja, hogy egy feladatot ad az osztálynak, előfordul-e, hogy ideges, furcsa érzés fog el?
29. Mialatt egy feladatot végzel, szoktad-e úgy érezni, hogy rosszul csinálod?
30. Amikor mégis az iskolába, szoktál-e attól félni, hogy dolgozatot írtok?

Tisztelt Igazgató Asszony! / Úr!

Az informatika oktatása terén az egyik lehetséges, hazánkban jelenleg legerterjedtebb és immáron tudományosan is elismert tananyag és metodikai rendszer a Játékos Informatika (JIO). Használatát "Az általános iskolai nevelés és oktatás terve, technika és informatika" részeként jóváhagyta a Művelődési Minisztérium 41663/89. XII. sz. intézkedésével. Amennyiben Önök alkalmazzák vagy megismerni kívánják ezt, levelünk segítségükre lehet. Az informatikaoktatás terjesztésére, alkalmazásának minden életkorban való segítésére magam és munkatársaim készséggel állunk rendelkezésre. A módszer alaposabb megismerése publikációink tanulmányozása és/vagy az általunk szervezett Logo tanfolyamon való részvétellel történhet.

1. Publikációk:

1.1. A Pest megyei Pedagógiai Szolgáltató Iroda (Bp. Városház utca) kiadványai közül felhívjuk a figyelmet a "Játszd el a teknőcöt" tanári kézikönyv sorozatra. Az első három kötet az 1-2, 3-4, és az ötödik osztályhoz készült. A hatodik osztályos segédkönyv sajtó alatt, a JIO Bt-től kéziratként beszerezhető. A PEPSI -nél az első öt osztály számára tematikák is kaphatók.

1.2. A JI tanterve és vázlatos tematikája - elsőtől nyolcadikig - a második HungaroLogo konferencia kiadványban található meg. A Neumann János Számítógéptudományi Társaságnál a többi Logo konferencia-kiadvány is megvásárolható.

1.3. A JIO Bt által forgalmazott kiadványokból itt is ajánljuk a LogoWriter könyvet, a **Comenius Logo magyar nyelvű változatának ismertetőjét** és a **Comenius Logo példatárt**, a ROAMER padlóteknőcöt, a programlemezek közül a legfrissebb válogatást, és a gyorsolvasás programot (ez utóbbi kettő LogoWriter-ben készült).

A Játékos Informatika alkalmazásához **teachware-t** (tanári segédanyag gyűjteményt) is összeállítunk, amely a következő kiadványokat tartalmazza:

- A JI vázlatos **tanterve** (HungaroLogo 95 kiadvány)
- Tematika példák 1...5 osztályig (PEPSZI kiadvány)
- Tanári kézikönyvek 1...5 osztály (PEPSZI kiadvány)
- HungaroLogo'94 (NJSZT Kiadvány)
- LogoWriter Programismertető könyv (Múzsák Kiadó)
- Munkafüzet I. és II. osztály számára (JIO Bt)
- LogoWriter és demo válogatás (Programlemez IBM kompatibilis gépekhez)

Az összeállítás ára 10000.- Ft (+12% ÁFA)

2. Tanfolyam

A JIO Bt, 80 órás tanfolyamot indít **február** elején, Budapesten, a XI. ker. Hegytető u. 12. szám alatti villában. Részvételi díj 20 000.- Ft. A tanfolyam hetenkénti egy nap, péntekenként (13-17 óráig), vizsgával zárul. Elsősorban a LogoWriter és a Comenius Logo használatával ismerkednek meg a résztvevők. Egyaránt ajánljuk kezdőknek és a témában jártasabbaknak is, tanítóknak, tanároknak, óvodapedagógusoknak.

3. Keressük és ajánljuk a téma együttes továbbfejlesztését.

3.1. Az NJSZT Logo Szakosztálya és a Közoktatási Szakosztálya minden hónapban tart összejövetelt (általában szombat délelőtt). Az időpontok a Neumann Tájékoztatóban vagy a Társaság Titkárságától tudhatók meg. Minden érdeklődőt - nem Neumann tagokat is - szeretettel látunk. Januárban a hónap második szombatján 10-12 óra. Hegytető u. 12.

3.2. 1997. augusztusában hazánkban rendezzük meg az **EUROLOGO** konferenciát. Ezen a részvételt angolul tudó kollégáknak javasoljuk. Az EUROLOGO '97 kísérő rendezvénye lesz a HungaroLogo tanácskozás, amely ugyancsak jó lehetőség az informálódásra és a közös munkára. Részletesebb tájékoztatás az NJSZT -től kapható.

3.3 Készséggel tartunk bemutatót, előadást vagy kihelyezett tanfolyamot a kívánt helyszínen a Játékos Informatika pedagógiája, a Logo programnyelv, a Comenius Logo legújabb változata, gyorsolvasás stb. témákban.

4. Foglalkozások gyermekek számára

4.1. A Hungest Gyermeküdültetési Kft. 1997. nyarán Játékos Informatika táborokat szervez Siófokon júniusban (17...23 és 24...30 között 7 nap, 6 éjszaka). A táborban a gyerekek elsősorban a Logo nyelv segítségével alkotnak a számítógépekkel, képességfejlesztő számítógépes programokkal játszanak. A tábor díja: 16500 Ft. A részvétel biztosítható a cég valamelyik irodájában vagy a JIO Bt-nél 2000 Ft előleg befizetésével.

4.2. A JIO Bt. szombatonként 14 órától csoportos (vagy egyéneknek megbeszélt időpontban) Logo tanórákat ad gyerekeknek Budapesten, Hegytető u. 12.

További kívánt módon is (szaktanácsadás, tanulmányok stb.) állunk szíves rendelkezésükre.

Tisztelettel

Dr. Farkas Károly
az NJSZT Logo Szakosztályának elnöke

06 30403799

3194720

Hegytető u. 12

További információk:

Aranyos Gabriella

Neumann János
Számítógéptudományi Társaság
1054 Budapest, Báthori u. 16

Telefon: 132 9390, 132 9349
Fax: 131 8140
E-mail: eurologo@ella.hu

Lsd. állandó EUROLOGO lap:
<http://www.eurologo.org/>

*Learning
&
Exploring
with Logo*

Az EUROLOGO két évente megrendezésre kerülő konferencia,
amely a Logo és más hasonló, kísérletezést elősegítő szoftver oktatási alkalmazására
összpontosít különböző korosztályokat figyelembevéve.

Az EUROLOGO'97 témái

Tantárgyi módszertan
Alkotás közbeni tanulás
Szoftver fejlesztés
Kulturális és kultúrák közötti aspektusok
Otthoni tanulás
Sérült gyerekek
Tanárképzés



Hatodik európai Logo konferencia

Budapest

1997 augusztus 20-23



További információk:

Aranyos Gabriella

Neumann János

Számítógéptudományi Társaság
1054 Budapest, Báthori u. 16


Telefon: 132 9390, 132 9349

Fax: 131 8140

E-mail: eurologo@ella.hu

Lsd. állandó EUROLOGO lap:

<http://www.eurologo.org/>



*Learning
&
Exploring
with Logo*

Az EUROLOGO két évente megrendezésre kerülő konferencia,
amely a Logo és más hasonló, kísérletezést elősegítő szoftver oktatási alkalmazására
összpontosít különböző korosztályokat figyelembevéve.

Az EUROLOGO'97 témái

Tantárgyi módszertan

Alkotás közbeni tanulás

Szoftver fejlesztés

Kulturális és kultúrák közötti aspektusok

Otthoni tanulás

Sérült gyerekek

Tanárképzés



Hatodik európai Logo konferencia

Budapest

1997 augusztus 20-23