

# Vizuális programozás nyújtotta lehetőségek a középiskolai informatika órán

Kelemen András<sup>1</sup>, Árgilán Viktor Sándor<sup>2</sup>

{<sup>1</sup>kelemen, <sup>2</sup>gilan}@jgypk.szte.hu  
SZTE JGYPK Informatika Alkalmazásai Tanszék<sup>1,2</sup>  
Szegedi Radnóti Miklós Kísérleti Gimnázium<sup>1</sup>

**Absztrakt.** Az infokommunikációs technológiák látványos fejlődése és elterjedése egyre inkább megköveteli a felhasználói felületek és a mögöttes működési logika ismeretét. A Nemzeti Alap-tanterv alapján megfogalmazott érettségi követelményekben az irodai felhasználói ismertek (dokumentum szerkesztés, táblázat kezelés, egyszerű adatbázis kezelés) dominálnak. Emelt szinten a programozási feladat követelményrendszere teljes egészében csak konzol alapú programozási ismerteket követel meg. A konzol alapú programozás azonban nem látványos, többször elmaradnak a sikerélmények. Előadásunkban rámutatunk arra, hogy a vizuális programozás nyújtotta felhasználói élmény mennyire segíti a programozás megszertetését és a programozói gondolkodásmód fejlesztését.

**Kulcsszavak:** vizuális programozás, grafikus felület, eseménykezelés, projekt szemlélet

## 1. Bevezetés

Az infokommunikációs technológiák látványos fejlődése és elterjedése egyre inkább megköveteli a felhasználói felületek és a mögöttes működési logika ismeretét [4,5]. A Nemzeti Alap-tanterv alapján megfogalmazott érettségi követelményekben az irodai felhasználói ismertek (dokumentum szerkesztés, táblázat kezelés, egyszerű adatbázis kezelés) dominálnak [4,5]. Emelt szinten a programozási feladat követelmény rendszere teljes egészében csak konzol alapú programozási ismerteket követel meg. Azonban a konzol alapú programozás nem látványos, nehezebben jönnek a sikerélmények. Ráadásul a diákok heterogén érdeklődése, valamint a tanórák 45 perces beosztása miatt a programozás oktatása eleve nehéz feladat. A rendelkezésre álló idő – a menet közben óhatatlanul felmerülő kódolási, technikai problémák és a diákok eltérő sebessége miatt – gyakran nem elég a tanórára kitéűzött feladat teljes megoldására. A fentiek hatására a diákok jelentős része elfordul a programozástól, noha sokan szeretnének közülük informatikával foglalkozni. Megoldás lehet a programozás megszerettetésére, ha az informatika tanár programozás szakkört indít, azonban féltő, hogy ide csak azok fognak járni, akik már elkötelezettek [3]. Másik út lehet, hogy a programozás oktatását nem a klasszikus utat bejárva végezzük, hanem a vizuális programozás nyújtotta azonnali sikerélményeket próbáljuk a cél érdekében felhasználni. [1-3]

A cikkben egy tapasztalatokon alapuló módszert mutatunk arra, hogy a fent említett problémát hogyan lehet áthidalni és megszerettetni a programozást.

## 2. A módszer

A módszer megalkotásakor a kiindulási ötlet az volt, hogy a Szegedi Radnóti Miklós Kísérleti Gimnáziumban a 6 évfolyamos speciális matematika tagozaton a 7. osztályban LOGO-val tanulnak programozni a gyerekek. Az egyszerű utasításkészlet és a teknőc nyújtotta vizuális visszacsatolás még az informatikától idegenkedőket is arra sarkallta, hogy megpróbálják kiszervenndni az adott feladat megoldását. Amikor a 8. osztálytól kezdve a Logo-t felváltotta valamilyen komolyabb programozási nyelv

(C/C++, C#, Java) kevés kivételtől eltekintve azonnal elmaradtak a sikerélmények és jöttek a problémák, melynek egyik oka lehet, hogy programozási alapok oktatásánál elmarad a vizuális visszacsatolás.

A programozás alapjainak oktatása általában az alábbi tematika szerint szokott történni:

- Az adat és az algoritmus fogalma példákon keresztül.
- A hardver és a szoftver fogalma. Az operációs rendszer feladatai.
- A konzol (parancssor használata, fontosabb parancsok)
- Programok felépítése (adatok+algoritmusok= programok), programozási nyelvek, programozási paradigmák.
- A választott fejlesztő környezet ismertetése.
- A változó, mint adattároló fogalma. Elemi adattípusok a választott nyelvben.
- Változók deklarációja, értékadás.
- Adat bekérés és adatkiírás a konzolra.
- Matematikai és logikai műveletek.
- Iterációk,
- Elágazások
- Véletlen számok és használatuk a választott nyelvben.
- Tömbök használata.
- Keresés, kiválogatás, szétválogatás
- Unió, metszet
- Rendezés tömbökben.
- String műveletek.
- Láncolt listák, bináris fák
- File kezelés (írás, olvasás)

A tematikából már látható, hogy a programok futtatásához szükség van a parancssor használatára, és a baj itt kezdődik, nevezetesen ismerni kell néhány fájl és könyvtár kezelésre vonatkozó parancsot.

## 2.1. Programozás alapjainak oktatása vizuális programozással

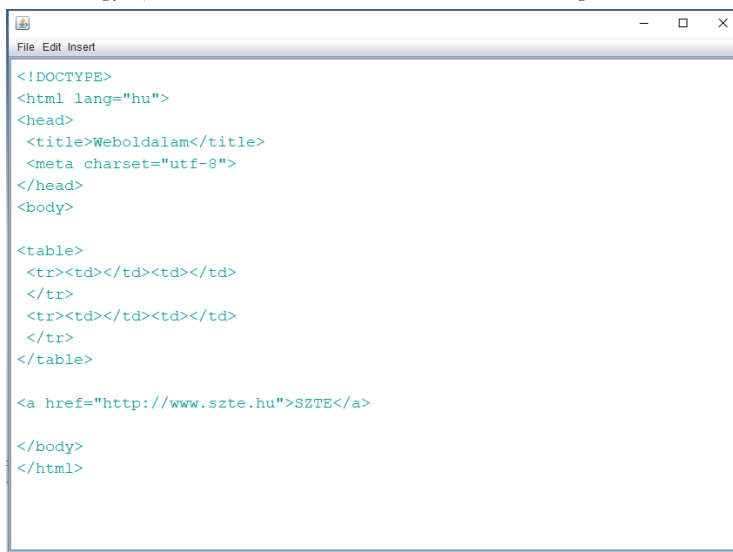
A vizuális programozást általában a felhasználói felület elkészítésével kezdjük, melyek a modern fejlesztő környezetekben (Visual Studio, Netbeans) nem igényelnek kódírást. A felület elemeit a „fogd és vidd” módszerrel lehet összerakni. A fejlesztő környezet, pedig automatikusan elkészíti felületet futtató programkódokat. Az első sikerélmény így megvan. Még nem írtunk semmilyen program kódot, de már van egy „működő” programunk. Működő, mert fut, de csak azokra az eseményekre reagál, melyeket a fejlesztő környezet automatikusan belerak. Itt el kell mondani a diákoknak, hogy minden olyan szoftver, amelynek grafikus felhasználói felülete van, ott a program a felhasználói felületen bekövetkező *eseményekre* reagál. Szerencsére a grafikus keretrendszerek az események kezelését úgy oldják meg, hogy az esemény bekövetkeztekor lefuttatnak egy üres metódust is. Ebbe a metódusba kell nekünk az adott eseményre vonatkozó logikát leprogramozni. Ez elsőre természetesen sokkoló hatású, de a diákoknak azt kell mondani, hogy az esemény kezelést fekete doboznak tekintsék, és csak azzal foglalkozzanak, hogy az esemény bekövetkeztekor hogyan reagáljon a rendszer.

Természetesen az oktatásban be kell tartani a fokozatosság elvét, ezért kezdetben csak kettő vezérlő elemet használunk. Szövegdobozt az adatbevitelre és megjelenítésre. Nyomógombot az esemény kezelésre. Mivel a szövegdoboz csak stringeket tud kezelni, ezért ez az egyszerű felület kiválóan alkalmas a string műveletek és velük együtt az iterációk és elágazások gyakorlására. Néhány példa:

- Szövegben adott betű másik betűre cserélése.
- Szövegben betűk előfordulásainak megszámlálása.

- Szöveg titkosítása.
- Szövegek darabolása, összefűzése.

Többsoros megjelenítésre alkalmas szövegdobozt valamint véletlen számokat használva már lehetőségünk van a tömbök használatának begyakorlására a klasszikus példákon (maximum és minimum keresés, átlagszámítás, indexek keresése, rendezések, stb) keresztül. Valamint lehetőségünk van a file műveletek (mentés, betöltés) illusztrálására (1. ábra). Ennél a feladatnál a nyomógombokat menükkal helyettesítettük. Ezzel egy újabb vezérlő elem használatát ismerték meg a diákok.



```
File Edit Insert
<!DOCTYPE>
<html lang="hu">
<head>
<title>Weboldal</title>
<meta charset="utf-8">
</head>
<body>

<table>
<tr><td></td><td></td></tr>
</tr>
<tr><td></td><td></td></tr>
</tr>
</table>

<a href="http://www.szte.hu">SZTE</a>

</body>
</html>
```

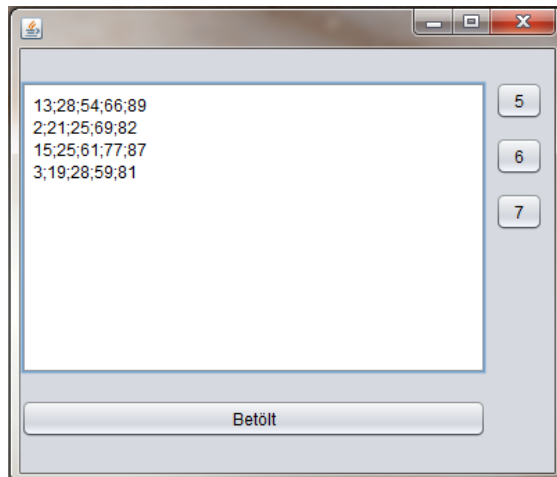
1. ábra: Egyszerű html editor képernyő képe

A következő órán a programot kiegészítettük a „New” az „Insert table”, és az „Insert link” parancsokkal, melyeknél string műveletekkel szűrtünk be HTML utasításokat a kurzor pozíciójába.

## 2.2. Projekt szemlélet

Természetesen, ahogy haladunk előre a vizuális programozásban egyre több vezérlő elem kerül elő és velük együtt az összetettebb adatszerkezetek (kétdimenziós tömbök, rekordok, láncolt listák). Az illusztrációul szolgáló példák pedig egyre összetettebbek, egyre inkább az alkalmazásfejlesztés irányába mutatnak [1,2]. Itt már egy 45 perces óra nem elegendő nulláról indulva egy feladat teljes megoldásához. Projektekben, projekt szemléletben kell gondolkodni. Az egyéni motiváció fenntartásához érdemes a diákokat 2-3 fős csapatokra osztani. Amennyire lehet, itt már hagyni kell őket önállóan dolgozni, természetesen szigorú határidőt szabva a projekt elkészültének. Néhány általunk használt példa projekt:

*Lottoszám generáló (2. ábra):*



2. ábra: A Lottószám generáló alkalmazás képernyő képe

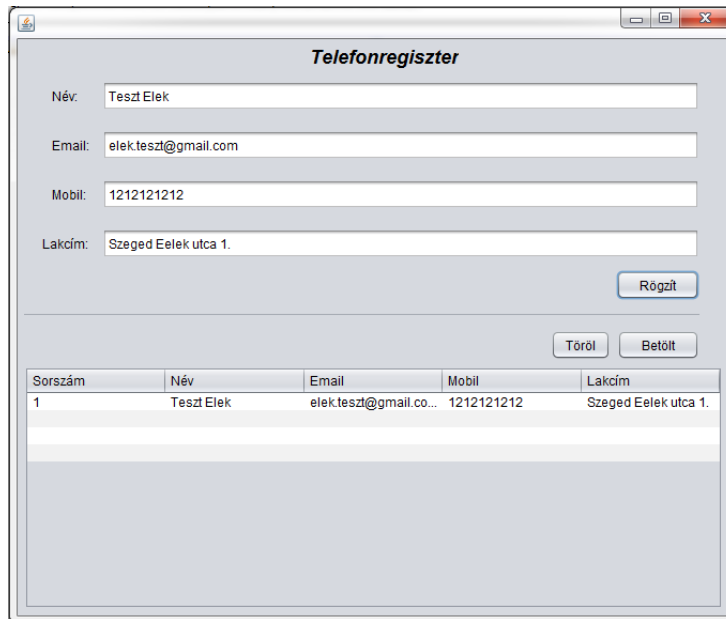
Ennek a projekt feladatnak a részletes specifikációja a következő:

Készítsen grafikus alkalmazást, amely lottószámok egyszerű kezelését teszi lehetővé.

- A program az 5-ös, 6-os és 7-es lottó szabályainak megfelelő számokat generál.
- A felhasználó az ábrának megfelelően nyomógombok segítségével dönti el, hogy melyik lottóhoz generál számokat.
- A nyomógombokon történő minden egyes kattintáskor az adott lottó szabályainak megfelelő számsor generálódik és a megjelenítő ablakban új sorban jelenik meg, a soron belül növekvő számsorrendben.
- A generált számok automatikusan elmentődnek az adott lottó szabályainak megfelelő oszlopszámú pontosvesszővel tagolt csv formátumú fájlba. A sor végén ne legyen pontos vessző!
- Az adott lottójátékhoz tartozó első mentés alkalmával a program a fájl kiválasztó ablak segítségével kérdezzen rá a fájl névre.
- Legyen lehetőség a már elmentett lottószámok betöltésére, megjelenítésére

Ennél a feladatnál már előkerül az egészszámok tömbökben való tárolása, tömbök véletlen számokkal adott feltételek mellett történő feltöltése. További programozási feladat még a tömbelemek szövegdobozban történő megjelenítése (egészszám - string konverzió).

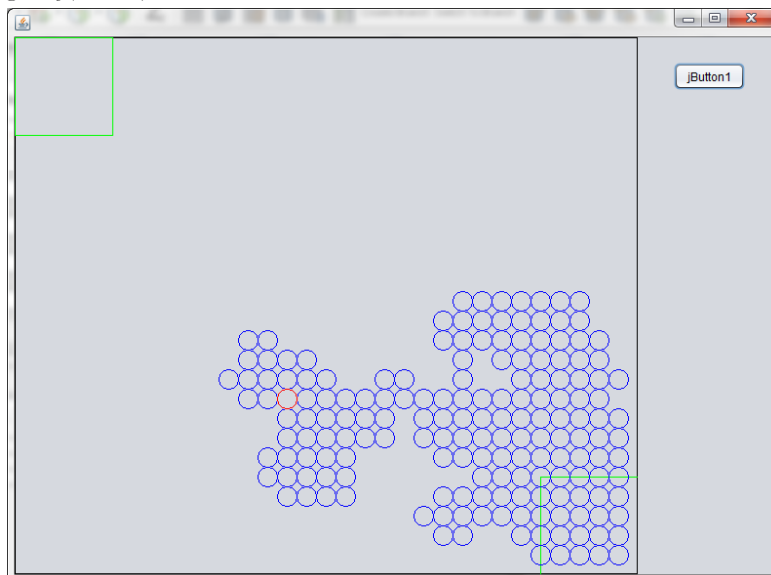
*Telefonregisztráció* (3. ábra):



3. ábra: A Telefonregiszter alkalmazás képernyő képe

Az oktatási cél itt a rekord és a láncolt lista használata a személyek memóriában történő tárolására, a lista tárolása csv fájlban, valamint a grafikus vezérlőelemek (szövegdoboz, nyomógomb táblázat, fájlválasztó) együttes használata.

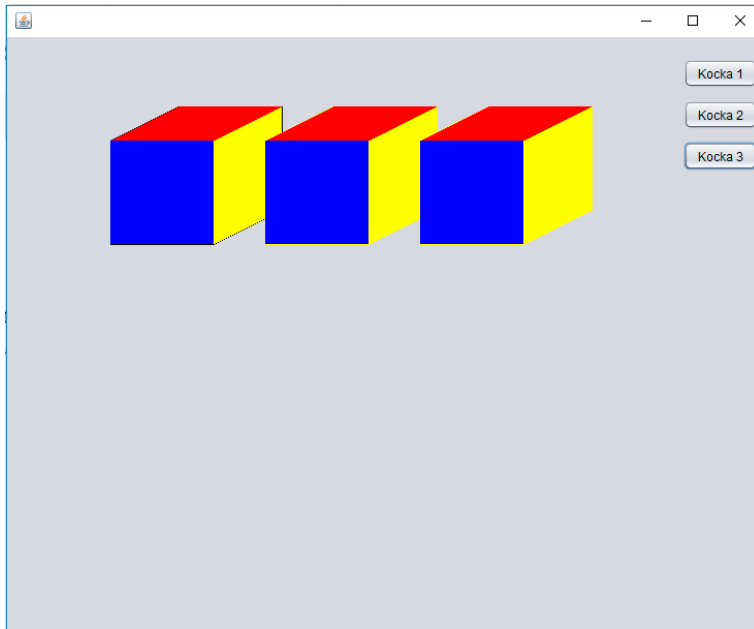
Részeg tengerész (4. ábra):



4. ábra: Részeg tengerész képernyő képe

A grafikus felületen az időzítő és a véletlen szám generátor együttes használatának bemutatására szolgál a *Részeg tengerész* nevű alkalmazás (4. ábra), mely alapvetően a Brown mozgás szimulálására szolgál. Ennél az alkalmazásnál a játéktér bal felső és jobb alsó sarkában van egy-egy zöld téglalappal jelzett „kocsmá”. A szimuláció kezdetén a piros körrel jelzett tengerészünk a jobb alsó kocsmában helyezkedik el, és itt alaposan felöntött a garatra. Ekkor szólnak neki, hogy a bal felső kocsmában sokkal finomabb a rum. Tengerészünk elindul, de olyan részeg, hogy minden egyes lépésénél véletlenszerűen jobbra vagy balra, vagy előre, vagy hátra lép. Az alkalmazás az időzítő minden egyes aktíválódásakor a tengerész aktuális pozícióját kék színnel kirajzolja, majd piros színnel az új pozícióba lépteti.

*Kocka rajzoló* (5. ábra):



5. ábra: Kocka rajzoló képernyő képe

Az oktatási cél itt a háromdimenziós ábrázolás gyakorlása. A részletes feladatspecifikáció szerint a nyomógombra kattintva mindig annyi kiszínezett kockát rajzol program, amekkora szám a nyomógombra írva van.

### 3. Tapasztalatok, eredmények

A középiskolában a tanulói és tanári munka sikerességét általában az emelt szintű érettségi és esetleg a tanulmányi versenyek eredményein szokták számszerűleg mérni. Tisztában vagyunk vele, hogy ez nem teljesen objektív mutató. Erősen függ a diákok kezdő felkészültségi szintjétől, az iskola és a tanár lehetőségeitől, a tananyag és tantervi előírásoktól.

Tapasztalataink szerint, ha az informatika órán a hangsúlyt és ezzel együtt az óraszámokat a programozásra fordítjuk, akkor az itt megszerzett készségek, gondolkodásmód más tanórán (pl. matematika, fizika, nyelvtan, egyes szakmai tárgyak) is előjönnek, hasznára válnak a diáknak. Jelentősen javul

a diákok probléma átlátó és problémamegoldó képessége. A programozásba és főleg a vizuális programozásba fektetett munka és idő busásan megtérül az alkalmazói szoftverek oktatásánál, hiszen itt már pontosan értik e szoftverek felépítésének és működésének a logikáját.

Az emelt szintű érettségien a programozási feladat „mumus” feladat szokott lenni. Tapasztalataink szerint vizuális programozás nyújtotta fejlesztői szemlélet sokkal könnyebbé teszi ezen összetett feladat megoldását.

Meglátásunk szerint a tanulmányi versenyek legfőbb hozadéka a rendszeres elmélyült tanulásra való késztetés. Amennyiben a vizuális programozással sikerült megszerettetni a programozást a diákkal, akkor sokkal könnyebb rávenni őt az algoritmusok és az adatszerkezetek mélyebb tanulmányozására és releváns tanulmányi versenyeken történő indulásra.

## Irodalom

1. Dr. Kelemen András: *Programozás oktatás projektalapon a közoktatásban*. In: Koltai, László (szerk.) Hazai és külföldi modellek a projektoktatásban Budapest, Magyarország: Óbudai Egyetem Rejtő Sándor Könnyűipari és Környezetmérnöki Kar, (2019) pp. 122-131, 9 p. ISBN 978-963-449-133-0
2. Dr. Kelemen, András; Árgilán, Viktor Sándor: *Szimulációs algoritmusok oktatása projekt szemléletben*. In: Bodáné, Kendrovics Rita (szerk.) Hazai és külföldi modellek a projektoktatásban: Nemzetközi Tudományos Konferencia tanulmánykötete Budapest, Magyarország: Óbudai Egyetem Rejtő Sándor Könnyűipari és Környezetmérnöki Kar, (2018) pp. 335-343, 9 p. ISBN 978-963-449-024-1
3. Kelemen, András; Árgilán, Viktor Sándor: *Szimulációs algoritmusok az informatikatanár szakos hallgatók képzésében*. In: Szlávi, Péter; Zsakó, László (szerk.) INFODIDACT 2017, Budapest, Magyarország : Webdidaktika Alapítvány, (2017) p. ISBN:9786158060813
4. Árgilán, Viktor; Kelemen, András: *Digitális kompetenciák a NAT-ban és a XXI. század elvárásai*. In: Zsakó, László; Szlávi, Péter (szerk.) INFODIDACT 2016: Informatika Szakmódszertani Konferencia Zamárdi, Magyarország: Webdidaktika Alapítvány, (2016) Paper: AVKA.pdf, 4 p.
5. Árgilán, Viktor Sándor; Kelemen, András: *Az elvárt digitális kompetenciák a XXI. században és az informatika oktatás gyakorlata a közoktatásban Magyarországon*. In: Bíró, Károly-Ágoston; Sebestyén-Pál, György (szerk.) XVII. ENELKO – XXVI. SzámOkt Nemzetközi Energetikai-elektrotechnikai és Számítástechnikai Konferencia Kolozsvár, Románia: Erdélyi Magyar Műszaki Tudományos Társaság (EMT), (2016)