

FELFEDEZÉSEN ALAPULÓ AKTÍV TANULÁS MIKROVILÁG KÖRNYEZETBEN

CZAKÓOVÁ, Krisztina, SR

Resumé: A mikrovilágok olyan környezetek, melyek megengedik a felhasználónak a közvetlen tapasztalást egy virtuális szimulációs környezetben. A tanulók számára sok hasznos és értelmes aktivitást biztosítanak. Teret nyújtanak a felfedezéshez és aktív tanuláshoz, effektíven alkalmazva modern oktatási stratégiákat, mint a problémamegoldó és projektanítási módszer. Tipikus jellemzők: nyitottság, objektumorientáltság, egyszerű nyelvezet, egyszerű és intuitív kezelhetőség, interaktivitás, szemléletesség (vizualizáltság).

Key words: aktív tanulás, mikrovilág környezetek, mini nyelvek

DISCOVERY-BASED ACTIVE LEARNING IN MICROWORLD ENVIRONMENT

Abstract: The microworlds are environments, which allow the user to direct experience in a virtual simulation environment, provide space for active and discovery-based learning, effectively applying the modern teaching strategies such as problem solving and project teaching methods. The microworlds are providing a lot of useful and meaningful activities for the students. The typical characteristics of microworlds: openness, object-orientation, simple language, intuitive usability, interactivity, visualization.

Key words: active learning, microworld environments, application development

1 Bevezetés

A mikrovilág az alapvető ismeretek aktív tanulással történő megismerésének és megszerzésének lehetőségét biztosítja a tanulók számára. Wilson (1995) szerint a hatékony oktatás magától a tanulótól is megkövetel bizonyos fokú kezdeményezést (iniciatívát). A környezet, ahol a tanulónak meghatározott tere van a felfedezésre és vizsgálódásra, ahol bizonyos fokig saját maga határozhatja meg a célokat és tanulási aktivitásokat, - vezérelve a kíváncsiság és tudásszomj által, attraktívnak és hatékonyan tűnik az új ismeretek effektív megszerzése függvényében. Azok a tanulók, akik számára gazdag az információ hozzáférés, valamint azok feldolgozását támogató eszközök lehetősége is biztosított, és megkapják a pedagógustól a szükséges segítséget és helyes útbaigazítást, nagy valószínűséggel valamit meg is tanulnak és ez a tudás tartós marad, mivel ezen koncepció alapján a tanulás támogatott, segített és szabályozott, de nem szigorúan irányított vagy bármilyen módon diktált.

2 Aktív tanulás lehetősége mikrovilág környezetben

A mikrovilág környezetet biztosít a felfedezéshez/vizsgálódáshoz, mellőzve a szigorú didaktikus utasításokat, összhangban a Piaget elvű tanulással. A mikrovilágnak megvannak a maga korlátai is. Kell, hogy specifikus legyen valamilyen adott tudásterületre (doménra) nézve. Ahhoz, hogy az oktatási rendszerben is differenciáljunk az ismereti szintek között szükséges, hogy több szinttel rendelkezzen. A legalacsonyabb szintnek biztosítania kell az egyszerű hozzáférést már az új tematikus egység megismerése kezdetén. A legmagasabb szint is kell, hogy biztosítsa magát a környezet bővítését és továbbfejlesztését.

A mikrovilágnak mindenekelőtt biztosítania kell a tanuló számára olyan aktivitást, ami neki személyesen hasznos és értelmes (Jenkins, 2012). „Ez nem azt jelenti, hogy nincs szükség semmilyen oktatásra, vagy nem léteznek semmilyen „behaviorális célok“. A tanító és tanuló kapcsolata (viszonya) alapján eltér a klasszikus oktatási folyamatban zajlótól: a tanító a tanulót mondhatni inkább vezeti abba mikrovilág környezetbe, amelyben a felfedezéseket végezni fogják, mintsem magához a felfedezéshez.” (A szerző szabad fordítása: Papert, 1980, 209. o.)

A mikrovilágok egyik nagy előnye az, hogy egyszerű és alacsony számú szintaktikai egységekkel rendelkeznek. A mikrovilágok metaforákra épülnek, biztosítják a problémák színes halmazának kialakítását, melyek közvetlenül összefüggnek a felhasználó (tanuló) élettapasztalatával. Az objektum által végrehajtott utasítások mindig láthatóak (a vizualizáció, vagyis a szemléletesség támogatja a tanulás hatékonyságát) és azok sorrendje lehetővé teszi megérteni a bevitt vezérlő konstrukciók szemantikáját.

3 Mini nyelvek és mikrovilág környezetek helye az oktatásban

Létezik több indok, miért kell tanítani a mini nyelveket és hogyan kell tanulni az általuk létrehozott mikrovilág környezetekben. Mindenekelőtt új utat és új lehetőségeket nyitnak az oktatási aktivitások felé. A mini nyelvek erős alapot biztosítanak a problémák szisztematikus és algoritmikus feldolgozásához azok számára is, akik csak alkalmazói programozó szinten fognak "programozni" (táblázatkezelés, adatbázis-kezelés, stb.) – vagyis valamilyen applikációt (alkalmazást) szeretnének majd kialakítani.

A programozás didaktikai aspektusból teret ad a logikus és absztrakt gondolkodás alapjainak kialakításához. A probléma megszületése után az ember annak megoldására a gondolkodást hívja alapul – tervez, döntéseket hoz, ellenőriz és megold. Ezáltal tökéletesíti saját analitikus és logikus gondolkodását, amelyre többek között a programozási nyelvek is jó alapul szolgálnak. Egyértelmű menetek és algoritmusok megoldásának tervezésével nemcsak saját magát fejleszti az egyén, de a mindennapi életproblémák megoldására hasznos metódusokat és technológiákat is kialakít.

4 Gondolkodásfejlesztés és felfedezésen alapuló tanulás mikrovilág környezetben

Felmerül a kérdés, hogyan lehet biztosítani a gondolkodásfejlesztést olyan gyermekek esetében, akik többsége nem mutat érdeklődést a magasabb szintű programozási nyelvek bonyolultabb környezete iránt, mint a Pascal, C nyelv, Visual Basic, Java és így tovább. Ezen probléma megoldását a gyermekközpontú mini nyelvek jelenthetik, melyek kínálata valóban színes. Néhány közülük már az óvodás korosztályú gyermekek számára is megfelelő, mások inkább az iskolaköteles tanulók esetében alkalmazhatók eredménnyel. Azonban egyikről sem mondhatjuk, hogy korlátolt lenne a felhasználó életkorára való tekintettel. Az alábbi általános jellemzés minden mikrovilágra vonatkozik úgy, ahogyan a további gyermekeknek szánt programozási nyelvekre és környezetekre is:

- a mikrovilág környezetek játékos formában tanítják a programozás alapjegyzeit, melyek a későbbiekben felhasználhatók a magasabb szintű programozási nyelvek és környezetek esetében;
- egyszerűek, könnyen kezelhetők, interaktívak, szemléletesek, érdekesek;
- a gyermekek valamilyen objektumot (teknőc, robot, varázsló és egyéb szereplőt) vezérelnek, megoldásaikat parancsok segítségével szerkesztik, melyek megfogalmazása gyakran a mindennapi nyelvhez hasonló;
- lehetővé teszik az egyszerűbb utasítások összevonását bonyolultabb egységekbe.

Mikrovilág alapú programkörnyezetnek van néhány előnye is a magasabb szintű programozási nyelvekkel szemben, mégpedig a:

- *szintaktikai egységek kis halma és egyszerű szemantika* – a tanuló programozói alapismeretek és jártasságok hiányában is rövid időn belül elsajátíthatja a munkát egy ilyen programozási nyelvben (annak mikrovilágában), mindemellett a problémamegoldás algoritmizációs elveit és a programozás alapelveit is megismeri;
- *metaforákra épülnek* – ezáltal teszik lehetővé olyan problémák széles skálájának kialakítását és megoldását, melyek a tanuló élettapasztalataival szoros összefüggésben vannak;
- *vizualizáció és szemléletesség* – az objektummal végrehajtott operációk mindig vizualizáltak, ez által felfedésre kerül az alkalmazott bázis programozási nyelv szemantikája és annak szerkezete (építőelemei), ami nagymértékben hozzájárul a hibalehetőségek gyakoriságának csökkentéséhez; a vizualizáció támogatja a felfedezéssel (kísérletezéssel) alapuló tanulást, valamint fejleszti a tanulók fantáziáját;
- *a felhasználó használhatja a számára természetes kommunikációs nyelvet* – nincs kötve a "nagy" programozási nyelvek szintaxisához se szemantikájához;
- *attraktív környezet* – színes grafika, egyszerűség, intuitív kezelés, szereplők (objektumok) "élőlény" adta tulajdonságokkal felruházva.

5 Oktatásban alkalmazható mini nyelvek és mikrovilágok rövid jellemzése

A mini nyelvek kialakulását erősen befolyásolta a logo teknőcgrafika megjelenése, mely a szó bizonyos értelmében az első mini nyelv alapú mikrovilágnak is tekinthető. Ezen alapfilozófiára épül több nevezetes mini nyelv is, melyek rövid jellemzését, az objektumaik (központi szereplők) alaptulajdonságainak és azok kezelési elvének a leírását a következő részben taglaljuk.

Karel robot

A Karel robot környezete eredetileg egyetemi hallgatók számára került kifejlesztésre, egyszerű bevezetésként a strukturált programozásba Pascal nyelven. Lehetővé teszi a procedurális absztrakció fogalmának kialakítását (minden jelentős Pascal vezérlési szerkezettel), a szekvenciális és feltételes végrehajtás elsajátítása mellett.

A módosított cseh verzióban a robot (objektum) falakkal, jelekkel és téglákkal teli világban él (városnak is nevezhető négyzethálón), ahol különböző feladatokat hajt végre – a hátizsákjából téglákat és különböző alakzatú jeleket rak le, mozog, forog, tud téglát vagy jelet elhelyezni, ill. felemelni, stb. A mikrovilág állapotát, melyet épít, alapinstrukciók és feltételek segítségével kontrollálja. Minden megadott parancs vizualizálódik a képernyőn, ezért azok megvalósulása azonnal nyomon követhető és ellenőrizhető.

Jelen pillanatban a Karel nyelvnek több implementációja is létezik. A Karel 3D verzió térorientált. A tanulókat leginkább a robot-építkezés művészete ragadja meg. Szemléletességével lehetővé teszi a strukturált programozás bonyolultabb fogalmainak és vezérlő szerkezeteinek elmagyarázását (Pauchly, 2002).

Imagine Logo

Az Imagine Logo (továbbiakban Imagine) az egyik legelterjedtebb és leghozzáférhetőbb gyermekközpontú programozási (mini) nyelv az iskolákon. Implementációja a 2001-es évben fejeződött be, mint a Comenius Logo közvetett fejlesztése. A Logo nyelvet használja. Központi „hőse” egy teknőc, aki hallgat a parancssorba beírt utasításokra. Képes változtatni az alakját, (átöltözni különféle statikus vagy dinamikus képekbe – animációként). Nyomot hagyva maga után tud rajzolni a grafikus háttérre, vagy csak nyom nélkül mozogni

rajta. A teknőc tulajdonságai változtathatók, olyan események is beállíthatók, melyekre a teknőc reagálni tud – vagyis a felhasználó nem egy összefüggő programot ír, hanem részfeladatokat old néhány parancsból álló alprogramok definiálása által, melyek az adott eseményre automatikusan meghívásra kerülnek (példák eseményekre: haKlikk, haMutatLap, haÜtközik, stb.)

Grafikus kiegészítők előkészítését, mint a teknőc alakja, animációk, háttérképek, stb. a Logo Motion grafikus editor támogatja. Az Imagine multimédiákkal is dolgozik, teret nyújt az alkalmazás vizuális megjelenésének tervezéséhez, lehetőséget kínál objektumok egyszerű beillesztéséhez és változtatásához. (Blaho, 2002)

IzyLogo

Ez a programkörnyezet a teknőcgrafika egy leegyszerűsített környezetét kínálja. A környezet fő jellemző vonása, hogy kártyalapokon előredefiniált parancsok által egyetlen objektumot vezérelhetünk benne – egy teknőcot. Korlátolt a mozgása valamint a forgása is, mely a 45° többszörösére épül. Nem standard a távolság meghatározása sem a négyzetháló egyes pontjai között: **e 1** parancs (vagyis 1 lépés előre) a négyzetháló legközelebbi pontjára való átlépést jelenti a teknőc aktuális elfordulási irányában haladva.

Annak ellenére, hogy a parancsok szövegalapúak (vagyis azok használata és ismerete olvasási készséget igényel), maguk a programok írást nem igényelnek, mivel elődefiniált ikonokat tartalmazó kis kártyalapok mozgatásával (vonzolásával) tudjuk megalkotni (összerakni / felépíteni) a programot az erre adott felületen. A parancsok paramétereit kiválaszthatjuk a megadott alternatívák listájából. Az IzyLogo lehetőséget nyújt saját parancsok (eljárások) definiálására is, azonban csak paraméterek nélkül.

Könnyű kezelhetősége és egyszerű munkakörnyezete miatt alkalmas az alapiskola alsó tagozatos tanulói számára is.

Élő kép (Živý obraz)

Tulajdonképpen az Imagine leegyszerűsített változatáról beszélünk, fókuszálva az egyszerű aktivitások körére – élő képek kialakításával, melyek statikus vagy dinamikus grafikus objektumokat tartalmaznak. Olyanok is akadnak közöttük, melyek képesek reagálni a felhasználó által vezérelt eseményekre/akciókra (klikkelésre vagy egér általi vonzólásra). A program nyelve teljesen ikonikus, csak néhány utasítás paramétere van számmal kifejezve. Ez a környezet egy szlovák nemzeti projekt (ESF) igényére lett kifejlesztve, mely célja a pedagógusok továbbképzése volt az informatika szakterületén belül.

Ez a környezet alkalmas az Imagine programkörnyezetben való programozás bevezetésére (felkészítésre). A tanulók már ismert objektumokkal fognak találkozni, ismerni fogják az egyes utasítások jelentését, melyeket már írásos formában kell megadniuk. A program a felhasználók számára szabadon hozzáférhető, így Szlovákiában spontán terjed a tanítók körében.

Scratch

A Scratch programozási nyelv 2006-ban került napvilágra, környezetének kezelhetősége egyszerű és intuitív. A program felépítése interaktív, megvalósítása során a program egyes elemei a fő ablakba kerülnek elhelyezésre. Az utasításpaletta tartalmazza az összes parancsot (alapvezérlő struktúrát), mellyel a programok kialakíthatók.

Egyszerű módon teszi lehetővé az interaktív prezentációk, képregények, játékok, animációk, stb. létrehozását. A Scratch a világ számos beszélt nyelvén elérhető. Installáció szükséglete nélkül is használható (online). Különböző lehetőségeket kínál a felhasználók számára, mint pl.: a létrehozott programok megosztása az interneten, a világ másik részén alkotott programok megtekintése, továbbgondolása, inspiráló hatás gyakorlása, stb.

Game Maker

A Game Maker interaktív programozói környezet játékok mikrovilágának kifejlesztésére alkalmas. Szerzője M. Owermars holland egyetemi tanár. Elsősorban szórakoztató animációs, multimediális, interaktív számítógépes játékok készítésére adott, oktató jelleggel alkalmazható. Mindenekelőtt az iskolaköteles tanulók számára javasolt, de számos téren kielégítheti az idősebb korosztály programozói igényeit, valamint a professzionális programozók érdeklődését is felkeltheti.

Az alkalmazásfejlesztés interaktív jellegű, az előre kialakított objektumok aktiválása által történik, szimbólumokkal ellátott nyomógombok segítségével. Az objektumok két csoportra oszthatók. A statikus objektumok csoportját képezik azok a grafikus objektumok, melyek a jelenetek (játék környezetének) létrehozását szolgálják. A másik csoportba sorolhatók a dinamikus objektumok, melyek a játék kialakítása során lesznek aktívak (pl. mozognak). Az aktivitás az adott dinamikus objektum kiválasztását követően a paraméterek megadása által specifikálódik.

A Game Maker környezete modern vizuális nyelvek tulajdonságaival rendelkezik. A kialakított programszerkezet objektumorientált, és az aktivitások eseményvezéreltek. A programozó nincs terhelve az objektum implementációjának részleteivel, valamint az események kiértékelésével. (Stoffová, 2001)

Baltík

A Baltazar program utódjaként keletkezett. A főszereplő egy varázsló (Baltík). A programkészítés általa nagyon egyszerű és gyors, mivel itt is ikonelemek mozgathatásával dolgozunk, egér használat segítségével. A nyelv szintaxisa nem okoz problémát, mivel a parancsok készlete teljesen ikonikus. A varázsló a 2D háttéren tud mozogni és varázsolni (négyzet formájú képeket). Következő üzemmódjai vannak: színhely létrehozása, a varázslás és szekvenciális programozás.

A Baltík nemcsak gyermekek számára alkalmas nyelv, de a tapasztaltabb programozók számára is kihívást jelenthetnek. A Baltík programnak több verziója létezik – Baltík 2, 3, Baltie 4C#, melyek interaktív programozást tesznek lehetővé ikonok és nyomógombok segítségével, akár 3D verzióban is (Baltie 4C#).

Kodu Game Lab

Könnyen kezelhető, átlátható, haladó szintű programkörnyezet, mely speciálisan 3D játékok fejlesztésére szolgál, gyermekek és felnőttek igényét egyaránt kielégítve. Vizuális programnyelvről van szó, melyet a Scratch program ihletett. A programozás logikája az egyszerű WHEN – DO típusú eseményeken alapszik (hasonlóan, mint ahogyan az IF – THEN feltétel működik). A feltételeket és parancsokat egy adott grafikus kínálatból ikonok segítségével választhatjuk ki, melyek leírásukban tartalmazzák a tevékenység végrehajtását és annak lehetőségeit. Ezen feltételek kínálata elég színes ahhoz, hogy érdekes és szórakoztató játékokat alkothassunk benne. Mivel a programozás vizuálisan történik, ami létrejön és megvalósul, megfigyelhető és tesztelhető a képernyőn. (Vlček, 2011)

E rövid áttekintés messzemenően nem tartalmazza az összes mini programozási nyelvet, mely a programozás oktatásának bevezetésénél és azon túl is jól alkalmazható (haladó szintű programozásnál). A tanulókkal való közös kreatív munkához minden tanítónak saját tapasztalata alapján kell kiválasztania a számára és a tanulók tanulási stratégiájának „legjobbát és legmegfelelőbbet“.

Befejezés

A mini nyelvek mikrovilág környezeti teret nyújtanak modellek és szimulációk létrehozásához egyéb hasznos aktivitás mellett. Miközben könnyen és intuitívan kezelhetők, alkalmasak bonyolultabb és igényesebb programozói feladatok megvalósítására is. Egyértelmű, hogy a mini nyelvek megalkotásánál a hangsúly mindenekelőtt a mikrovilág központi objektumának attraktivitására helyeződik. Előnyös, ha a programkörnyezet egy strukturált editorral is rendelkezik, mely támogatja a szintaktikai hibák kiküszöbölését, valamint segít tájékozódni a szerkezeti konstrukciók megnevezésében is. Nagyon fontos, hogy minél közelebb kerüljünk a reális világhoz és ez által – alkalmazva a problémamegoldás módszerét – a programozási nyelv és gondolkodásmód elsajátításához.

Bibliography

- [1] BLAHO, A. (2002). *Imagine. Popis*. AB KVI, 2002. (Online). Available: <http://imagine.input.sk/popis.html>. [Accessed on: Mar. 20, 2013].
- [2] JENKINS, C. (2012). *Microworlds: Building powerful ideas in the secondary school*. (pdf). In: *ICICTE 2012 : proceedings*. 2012, s. 61- 69. (Online). Available: <http://www.icicte.org/Proceedings2012/Papers/02-2-Jenkins.pdf>. [Accessed on: Jan. 21, 2013].
- [3] PAPERT, S. (1980). *Mindstorms: Children, computers and powerful ideas*. New York : BasicBooks, 1980.
- [4] PAUCHLY, K. (2002). *Výučbové programovacie mini-jazyky*. In: *Učiteľská stránka učiteľa na ZŠ s MŠ Chlebnice*. 2002. (Online). Available: <http://www.flatulent.szm.com/karel/minijazyky.html>. [Accessed on: Mar. 20, 2013].
- [5] STOFFOVÁ, V. et. al. (2001). *Informatika, informačné technológie a výpočtová technika : Terminologický a výkladový slovník*. Nitra : Fakulta prírodných vied UKF v Nitre, 2001. 230 s. ISBN 80-8050-450-4.
- [6] VLČEK, M. (2011). *Kodu Game Lab: Nový programovací jazyk pro školy*. In: *Česká škola – portal pro ZŠ a SŠ*. 2011. ISSN 1213-6018. (Online). Available: <http://www.ceskaskola.cz/2011/06/miroslav-vlcek-kodu-game-lab-novy.html>. [Accessed on: Mar. 20, 2013].
- [7] WILSON, B. G. (1995). *Metaphors for instruction : Why we talk about learning environments*. In: *Educational Technology*. 35(5), 1995, s. 25 – 30. (Online). Available: <http://carbon.ucdenver.edu/~bwilson/wils95>. [Accessed on: Jan. 20, 2013].

A cikk a *KEGA 010UJS-4/2014 Modelovanie, simulácia a animácia vo vzdelávaní* nemzeti projekt támogatásával készült.

Lectured by: dr. habil. Attila Kiss, CSc.

Contact address:

Krisztina Czakoová, PaedDr. PhD.,
 Department of Mathematics and Informatics, Faculty of Economics, J. Selye University, SK-94501
 Komárno, Bratislavská cesta 3322, Slovakia,
 phone: +421905763694 , e-mail: czakoovak@ujjs.sk