

Sprego solutions in science and arts

Sprego megoldások a tudományokért

1. Bevezetés

A Sprego módszer – Spreadsheet Lego – egy programozási megközelítés táblázatkezelői környezetben, annak funkcionális nyelvére építve (Csernoch 2014). A módszer alapjaiban a klasszikus algoritmizálási, programozási és számítógépes problémamegoldási megközelítésekre és eszközökre épít (Soloway 1993; Lister 2016). Újszerűsége egyrészt abban rejlik, hogy a kódoláshoz nincs szükség egyetlen „igazi” programozási nyelvre sem, hanem a táblázatkezelő felhasználók által is ismert eszközeit adaptálja és innovatív módon alkalmazza a valós világ adatfeldolgozási problémáinak megoldására. Másrészt, annak érdekében, hogy kisgyermekektől egész életen át tanítható és alkalmazható legyen, olyan unplugged, semi-unplugged és virtuális eszközkészlettel rendelkezik, amely nagyban segíti a problémák megértését, algoritmizálását, kódolását, tesztelését és diszkuszióját, a tanulók számítógépes gondolkodásának fejlődését (Wing 2006) és a számítógépes problémamegoldást.

2. Sprego programozás – elméleti megalapozás

A Sprego programozás (**Spreadsheet Lego**) (Csernoch 2014) egy nem-hagyományos programozási, ugyanakkor a klasszikus programozási módszereket adaptáló, megközelítés és eszköz felhasználói, elsősorban táblázatkezelői környezetben. A megközelítés főbb jellemzői:

- olyan végfelhasználóknak kínál programozási eszközt, akik nem nyitottak a magasszintű programozási nyelvek befogadására vagy még nem rendelkeznek megfelelő háttérismeretekkel és algoritmizálási készségekkel,
- lassú-gondolkodást (Pólya 1954; Kahneman 2011) igénylő koncepcióalapú problémamegoldás,
- sémák kialakítása (Skemp 1971; Merriënboer és Sweller 2005), amelyekkel megbízható gyors-gondolkodás alapú döntések hozhatók (Kahneman 2011),
- gyors- és lassú gondolkodás közötti hatékony és megbízható váltások (Kahneman 2011),
- autentikus tartalmak (real world problems) feldolgozása,
- a táblázatkezelő programok programozási nyelvéből adódóan funkcionális programozási nyelv, amely nagyban épít a matematikai függvény fogalomra, bővíti azt (többváltozós és többszintű függvények) és gyakorlatokkal segíti annak fejlődését,
- kiemelten támogatja az unplugged és a semi-unplugged megértést elősegítő eszközök tanórai használatát,
- önálló tanulást támogató 2D és 3D applikációk bevonása az oktatási folyamatokba,
- tudás transzfer: a tartalmakon keresztül kapcsolódás informatikán kívüli tantárgyakhoz, témakörökhöz, a programozási megközelítésen keresztül kapcsolódás adatbáziskezeléshez és magasszintű programozáshoz, az alkalmazói környezetben keresztül kapcsolódás fájlkezeléshez, szöveg- és weblapszerkesztéshez, -tervezéshez,
- szakít a hagyományos felületi megközelítéssel, amelynek központjában a speciális célú függvények alkalmazása található,
- méréseink bizonyítják, hogy hatékonyan fejleszti a tanulók/végfelhasználók számítógépes gondolkodását, algoritmikus készségét.

Mivel a Sprego programozás végfelhasználóknak kínál algoritmizálási és programozási lehetőséget, ezért minden olyan eszköz adaptálására szükség volt, amellyel a programozás iránt nem vagy csak minimálisan érdeklődő tanulók esetén is hatékonyan fejleszthető a számítógépes gondolkodás. A Sprego módszer adaptálja a „klasszikus” programozásoktatás bizonyítottan hatékony módszereit, a nem-hagyományos algoritmus-készségfejlesztő megközelítéseket és eszközöket, Kahneman gyors- és lassú-gondolkodás elméletét, a Cognitive Load Theory (2005) megközelítést, valamint a HY-DE (Hyper Attention – Depp Attention 2016) modellt.

3. Sprego függvények és tömbképletek

A Sprego (S) egyik sajátossága, hogy szakít a hagyományos és széles körben elfogadott felület-orientált táblázatkezelési megközelítésekkel és oktatási módszerekkel. A szoftvergyártó cégek „felhasználóbarát” (F) szlogenje helyett a programozási aspektusra helyezi a hangsúlyt:

- probléma elemzése, algoritmus építése (Pólya 1954) (S) vs. indítsuk el a számítógépet (F),
- függvények hívása (S) vs. eszköztárak, varázslók böngészése (F),
- többváltozós függvények argumentumlistájának tudatos használata (S) vs. varázsló-mezők véletlenszerű feltöltése (F),
- a hangsúly az alapkészségeken van (S) vs. a hangsúly a szoftver újdonságain van (F),
- outputok tervezése és diszkussziója (S) vs. ha kijön valami, az már jó (F).

Ezen keretek között, a többszáz táblázatkezelői függvény helyett, a Sprego egy tucat általános célú függvény használatát preferálja, és ezekre a függvényekre építve valósít meg valódi programozásalapú problémamegoldást egy funkcionális-nyelvi programozási környezetben (Csernoch 2014). A függvénykészlet nyitott, tehát a problémák jellegéből adódóan tovább bővíthető, szem előtt tartva a készlet általános jellegét.

Sprego_szöveg: BAL(), JOBB(), HOSSZ(), SZÖVEG.KERES(),

Sprego_szám: SZUM(), MIN(), MAX(), ÁTLAG()

Sprego_pro: HA(), INDEX(), HOL.VAN(), HIBÁS()

A függvények alacsony száma megköveteli továbbá azok egymásba építését, bevezetve ezzel az összetett függvények gyakorlatát, amely elvezeti a tanulókat az összetett függvények fogalmához, amely hiánypótló, mivel a téma jelenleg hiányzik a közoktatási célú matematikaoktatásból, ugyanakkor elengedhetetlen a programozáshoz.

A Sprego további sajátossága a tömbképletek használata, ezzel bevezetve az n-dimenziós vektor és az $n \times m$ dimenziós mátrix fogalmát, elkerülve a képletek másolását, valamint a relatív, abszolút és vegyes hivatkozások korai bevezetését, amelyek bizonyítottan a táblázatkezelői dokumentumok leggyakoribb hibaforrásai (Panko 2008).

4. Autentikus Sprego tartalmak – tantárgyközi kapcsolatok

A Sprego programozásnál használt autentikus táblázatok elsődleges forrása az internet. Az internetes tartalmak sokféleségéből adódóan a tanulók életkorának, érdeklődésének, tantárgyainak, tantárgyi témaköreinek figyelembe vételével kereshetünk és tölthetünk le olyan multimediális tartalmakat, amelyek tartalmazznak táblázatokat vagy táblázat-szerű egyéb

adatelrendezéseket. A webtáblák, tanórai keretek között, konverziós lépéssorozatokon keresztül normalizált adattáblává alakíthatóak, amelyek a későbbiekben felhasználhatóak táblázatkezelési, adatbáziskezelési és programozási problémák inputjaként. Jelen keretek között a konverziók végrehajtására egyrészt algoritmizáló, másrészt az algoritmusok megvalósításához alkalmazói – fájl-, szöveg-, általános táblázatkezelői és Sprego – eszközöket használunk.

A konverziós lépéssorozatok váltásokat igényelnek a különböző figyelmi állapotok és a különböző sebességű gondolkodásmódok között. Ezen állapotok és gondolkodásmódok közötti tudatos váltásokat Dani „The HY-DE Model: An Interdisciplinary Attempt to Deal with the Phenomenon of Hyperattention” (2016), valamint Kahneman „Thinking Fast and Slow” (2011), művei alapján azonosítjuk. A célunk az, hogy a kevésbé megbízható, felületes gyors-gondolkodás (hyper-figyelmi állapot) helyettesíthető legyen – a probléma jellegéből adódóan – vagy megbízható (reliable) gyors-gondolkodással vagy tudatosan megtervezett, felépített lassú gondolkodással (mély-figyelmi állapot).

Méréseink bizonyítják továbbá, hogy a weblap→adattábla konverziók elvezetnek egy weblap-szerkezeti validálási módszerhez és eszközhöz (Csernoch és Dani 2017), amellyel meg tudjuk mutatni, hogy a weblaptáblák tervezése és szerkezete mennyiben felel meg a jól-tervezett adattáblákkal szemben állított követelményeknek. Ezzel, a már meglévő szintaktikai helyességet ellenőrző és validáló programok mellé, tudunk adni egy szemantikai elemző módszert, amely felhasználható weblapszerkesztésnél, tervezésnél és oktatásnál.

5. A Sprego programozás unplugged eszközei

Napjainkra bizonyításra került, hogy a számítógépes problémamegoldás nem kizárólagosan számítógépes eszközökkel fejleszthető. Az egyik ilyen megközelítés a számítógép kiiktatása, elhagyása a probléma megértési, tervezési szakaszában. Bell és Newton (2013) kutatásai visszanyúlnak, újra felfedezik és felfedeztetik az unplugged eszközöket, melyeket adaptálva fejlesztettük ki a Sprego (Spreadsheet Lego) unplugged eszközkészletet.

A Sprego oktatása során kombináljuk a programozásból átvett hatékony módszereket és az alkalmazói szintű problémamegoldást. Olyan eszközöket használunk, amelyek felgyorsítják a megértés folyamatát, segítik a számítógépes gondolkodás fejlődését (Wing 2006), a gyerekek élvezettel használják azokat, valamint gyakorlóteret biztosítanak a többváltozós és összetett függvények elsajátításához (ami jelenleg nem része a közismereti matematika tananyagnak), a függvény fogalom kialakulásához és megszilárdulásához.

Az elsődleges unplugged Sprego eszköz a matrjoska baba, amellyel kiválóan szemléltethető az összetett függvények egymásba építése és a függvények közötti értékátadás menete. Az eredeti babák hozzáférhetőségi korlátai miatt azonban, eszközök széles választékát adaptáltuk vagy fejlesztettük ki: kereskedelmi forgalomban elérhető hordókészletek, 3D nyomtatással előállított matrjoska babák, különböző méretű és típusú origami hajók (vagy bármilyen más, könnyen, gyorsan meghajtogatható, egymásba ágyazható origami figurák), továbbá ragasztószalagok a címkézéshez. Ennek köszönhetően az unplugged eszközök használata minimális anyagi ráfordítást igényel. A sajátkészítésű eszközöknek megvan továbbá az az előnye, hogy a táblai, mágnesezett verzió is előállítható: az origami hajók esetében nagy kartonokat hajtogatunk, a 3D babáknál pedig nagyobb átmérőjű babakörvonalakat nyomtatunk. Mivel a módszer

középpontjában az algoritmusépítés, a problémamegoldás áll, kisebb gyerekekkel, különböző színű jelölőket használva (például csapat mellények, kitűzők, sapkák) eljátszhatjuk az algoritmusok lépéseit.

Az unplugged eszközök további előnye, hogy a problémamegoldás közösen történik, több szempont és megoldási javaslat kerül megvitatásra, így a tanulók szociális képességei is fejlődnek (a tanár leginkább koordinátor, coach szerepet tölt be). Nem elhanyagolható az az előnye sem a módszernek, hogy napjainkra bizonyítottá vált, hogy a programozás, számítógépes problémamegoldás, a látszat ellenére, nem önálló feladat, hanem egyre inkább csapatmunka.

6. A Sprego módszertan támogatása interaktív és virtuális valóságra alapuló eszközökkel

A Sprego módszertan támogatására készítettünk egy virtuális valóságra épített oktatási teret, illetve egy 2D-s (Scirra 2018) és egy 3D-s (Unity 2018) oktatószoftvert. A MaxWhere (Galambos et al. 2010, 2011) virtuális valóság környezetben kialakított oktatóteret annak szem előtt tartása mellett alakítottuk ki, hogy a tanulók számára otthoni egyéni, valamint tanórai keretek között folytatott munkavégzés keretein belül is biztosított legyen valamennyi szükséges információ és forrás a Sprego alapú problémamegoldáshoz. A térben, annak elhagyása nélkül a tanulók rendelkezésre állnak információk a módszertanról, munkafájlok, adatelemzési segédletek, feladatleírások és opcionálisan igénybe vehető algoritmusfelépítési magyarázatok. Az oktatóter tartalmazza az oktatási keretek között felmerülő táblázatkezelői témaköröket, figyelve a Kerettantervben meghatározott követelményekre. A tanulók szövegfüggvényekre, számítási-, valamint keresőalgoritmusokra épülő tananyagok gyakorlását is elvégezhetik a segítségével.

A Sprego eszközkészlet további eleme egy 2D-s HTML5 technológiára épülő, platformfüggetlen alkalmazás, amely a leggyakrabban felmerülő táblázatkezelői problémák (feltételes számlálás, valamint lineáris keresés) algoritmusainak felépítését és működését prezentálja való életben is előforduló kontextusokon keresztül. A lépésenként történő bemutatást elősegíti a táblázatkezelői környezetekben megtalálható képletkiértékelőhöz hasonló felület, amely bemutatja, hogy a táblázatkezelői képletek végrehajtása során a környezet milyen lépéseket végez el, és ezek részeredményét is kijelzi. A 2D-s animációs oktatószoftver mintájára megkezdjük annak 3D-s adaptációját. Ezzel a megoldással egy olyan újabb vizuális elemekben gazdag eszközt kínálunk, amely még inkább segíti a megértést, a tanulók számítógépes gondolkodásának fejlődését.