

Az algoritmikus gondolkodás fejlesztésének fontossága a közoktatásban

Németh Tamás¹, Széll Réka², Tornai Henrietta³

¹tnemeth@inf.u-szeged.hu

SZTE TTIK Informatikai Intézet

{²szell.reka, ³tornai.henrietta}@stud.u-szeged.hu

SZTE TTIK

Absztrakt. Tapasztalataink szerint az első éves hallgatók többsége nem rendelkezik megfelelő, az algoritmikus gondolkodáshoz nélkülözhetetlen alapkompenciákkal. Azokhoz a feladatokhoz, amelyek némi algoritmikus megközelítést igényelnek, és nem egy előre definiált séma alapján kell megoldani, már hozzá sem kezdenek a dolgozatokban. Az egyetemi informatikai képzések arra építenek, hogy a hallgatók képesek az alapvető probléma-megoldási módszerek alkalmazására, ezért fontos lenne, hogy már a közoktatásban megjelenjenek a programozáshoz és algoritmizáláshoz szükséges alapkompenciák fejlesztéséhez hozzájáruló elemek. Arra a kérdésre kerestük a választ, hogy mikor és hogyan lehet az algoritmikus gondolkodást a leghatékonyabban fejleszteni.

Kulcsszavak: gondolkodás, algoritmusok, informatika, oktatás, közoktatás

1. Jelenlegi helyzet

A 2012-es Nemzeti alaptanterv (NAT) javaslatot tesz a tanórák műveltségi területek közötti megosztásának százalékos arányára. Sajnos az informatika minden évfolyamon a lehető legkisebb százalékkal van jelen, és egyik évfolyamon sem haladja meg a 10%-ot. A logikus, algoritmikus gondolkodás, valamint a problémamegoldás tanítása az informatika oktatás alapvető céljai között szerepel a NAT-ban, igaz a legnagyobb hangsúlyt a felhasználói, alkalmazói tudás illetve a számítógépes készség- és képességfejlesztés kapja. Így azon a kevés informatika órán, ami a tanárok rendelkezésére áll, sokszor a középszintű érettségire való felkészítés is kihívást jelenthet.

Ilyen körülmények között nem csoda, hogy ilyen kevesen rendelkeznek megfelelő problémamegoldó képességgel, mire az egyetemre érnek. Ezen kellene változtatni, akár óraszám növeléssel, akár más módszerekkel, szemlélettel történő oktatással. Tanulmányunkban szeretnénk néhány ötletet mutatni arra, miként lehetne javítani a helyzeten.

Hiába van már a legtöbb otthonban számítógép, heti egy órában nehéz nagy eredményeket elérni. Otthon ugyanis csak kevesen foglalkoznak az órai programokkal, inkább játszanak. Érdemes tehát olyan játékokat mutatni, amelyek fejleszthetik például a diákok algoritmikus gondolkodását.

2. Emelt szintű érettségi eredmények

A kétszintű érettségi rendszer bevezetése óta informatikából az algoritmizálás és adatmodellezés emelt szinten követelmény.

Egyre több diák választja az emelt szintű informatika érettségét, és az eredmények is folyamatosan javulnak. Azonban vannak akik csak azért döntenek emelet, mivel a gyakorlati részben több idejük van megoldani azokat a feladattípusokat, amelyek középszinten is előfordulnak, és hozzá sem kezdenek a programozási feladatokhoz, vagy csak a könnyen megszerezhető, komolyabb gondolkodást nem igénylő részfeladatokat oldják meg. Akik viszont komolyabban foglalkoznak ezekkel, a könnyebbeket gond nélkül megoldják, a nehezebbekkel, amelyek komplexebb adatszerkezetek, és a programozási nyelv mélyebb ismeretét igénylik, pedig csak a legjobbak boldogulnak.

A felmérésekből az is kiderült, hogy sokszor szövegértési problémák okozzák a rossz teljesítményt, így erre is nagyobb figyelmet kellene fordítani. Találkozzon a diák akár informatika órán is hosszabb szöveggel, tanuljon meg értelmezni hosszabb feladatléírásokat, és fogalmazzon meg egész mondatos válaszokat akár szóban, akár írásban.

3. Algoritmikus gondolkodás tanítása és fejlesztése

A következőkben az algoritmikus gondolkodás tanításának lehetőségeit szeretnénk feltárni.

3.1. Mikortól fejleszthető?

A mindennapi életben rengeteg algoritmust használunk anélkül, hogy tudnánk róla. Ezekkel már kisgyermekkorban találkozunk: napirendi pontok követése, önálló öltözködés folyamata, cipőfűző kötés, zabpehelykészítés, stb.

A gyermekjátékok is alkalmasak az algoritmikus gondolkodás fejlesztésére, az egyre jobban elterjedő okostelefonok és tabletek pedig a játékkalkulációkkal kiváló lehetőséget biztosítanak erre. Ezekről még lesz szó bővebben.

Tehát már óvodában, és az alsó tagozatokon is fontos fejleszteni ezt a képességet a gyerekekben. Azt is meg kell mutatni, hogy egy feladatot többféleképpen meg lehet oldani (pl.: először a tejet vagy a zabpehelyt öntjük a tálba), de vannak olyan lépések, amik nem felcserélhetőek (pl.: először a zoknit vesszük fel, és utána a cipőt).

Igazából az alsó tagozatos matematika oktatása során kész algoritmusokat kapnak a diákok, ezeket gyakorolják be, tanulják meg alkalmazni. A feladatok megoldása során a már ismert algoritmusok közül megkeresik a problémának megfelelőt és használják.

3.2. Hol és hogyan fejleszthető?

Kézenfekvő válasz, hogy informatika órán, mikor sor kerül rá a tantervben. Fontos kiemelni, hogy nagyon sok rejtett lehetőség van az algoritmikus gondolkodás tanítására és fejlesztésére informatika órákon más témakörök, illetve a legtöbb tantárgy tanítása során is. Ezekre mutatunk példát a következő fejezetekben.

Ezen felül, a különböző iskolán kívüli tevékenységek (pl.: táncolás, zenélés), és a napi teendők, programok szervezése is megfelelő alkalom erre.

Mindig törekednünk kell arra, hogy az életkornak megfelelő feladatokkal közelítsünk. Az életből merített, valóságos problémák hasznosak, hiszen ezek többségével már találkoztak vagy találkozni fognak.

Meg kell mutatni, hogy lehet több jó megoldás is. Hagyjuk, hogy próbálkozzanak önállóan a megoldással. Több különböző megoldás születése esetén hasonlítsuk/ hasonlíttassuk össze őket, így fel lehet kelteni bennük az igényt, az algoritmus hatékonyságának vizsgálatára.

3.2.1. Családi környezet, óvodai évek

A korábban már említett önálló öltözködés egy elég komoly algoritmus, amit rengeteg ismétlés során sajátít el a kisgyermek. A ruhadarabok felöltésének sorrendje, az esetleges ruha ki- és megfordításokkal tűzdelve nem egyszerű feladat. Ezért kellő türelem, és pontos utasítás sorozat szükségeltetik a megtanításához.

Ugyanígy a közös sütés-főzés alkalmával is egy csomó algoritmust sajátíthat el a gyerek. A lépések sorrendje itt sem mindegy, de vannak olyan egységek, amiket egymástól függetlenül, egyéni sorrendben végezhetnek el. Hasonlóan eszközök, bútorok közös szétszedése, megszerelése, összerakása lépések sorozata, melyeket bizonyos sorrendben kell elvégezni.

Óvodában sokat játszanak, kézműveskednek. A kézműves foglalkozások esetében az elkészítendő produktum az óvónő által mutatott minta alapján történik. Előre meghatározott lépéseket követnek, persze szabad kezét kapnak a díszítésben, és minden óvódás a saját kezűgyességének megfelelő alkotást hoz létre. Megtanulnak, megtapasztalnak olyan tervezési eljárásokat, amelyeket a későbbiek során alkalmazni tudnak az általános iskolai rajz órákon is akár.

Tehát ezekben az években a kezűgyességükre, alkotásukra, cselekedeteikre vonatkozó algoritmusokat sajátítják el többségben.

3.2.2. Általános iskolai évek

Az óvodában elkezdett fejlesztési folyamatok folytatódnak. Rajz órákon új eszközök használatával új feladatok megoldására lesznek képesek. Megtanulnak írni, olvasni, számolni.

Írás tanulása közben a finommotoros képességeiket fejlesztő, és azokra építő algoritmusokat sajátítanak el. Például az írott kis f írásánál íves, visszakanyarodó saját magát metsző vonalat kell rajzolniuk, majd fel kell emelniük a ceruzát a lapról, és az előző vonalat a megfelelő helyen el kell metszeni egy új vonal rajzolásával. Ez nekünk ma már nem jelent kihívást, de nézzük meg egy első osztályos kisdíák gyakorló füzetét, hány próbálkozás után sikerül reprodukálnia. Amikor egy szót írnak le, egyenként “meghívják” az egyes betűk leírásához szükséges “programokat”, és a megfelelő összekötési módszerekkel kötik egymáshoz őket.

Az egyik fontos algoritmus csokor, amit elsajátítanak az alsó tagozatos matematika órákon, az írásos műveletvégzések. Ezek nagyon fontos lépéssorozatok, amiket később át kell tudniuk ültetni a papírról a fejükbe, és el kell jutniuk a fejben számolásig.

Az eddig említett algoritmusok teljesen belénk ívódnak, és ma már gondolkodás nélkül hajtjuk végre őket, hiszen ismerjük a lépéseket, tudjuk hogyan kell végrehajtani őket, tudjuk mely lépéseket lehet felcserélni, esetleg kihagyni. Felismerjük a hibát a mások által produkált eredményben, meg tudjuk mondani hol, melyik lépésben mit csináltak rosszul.

A felső tagozaton már megindul az önálló, otthoni tanulásra motiválás. Míg alsó tagozaton inkább gyakorló, felső tagozaton egyre inkább gondolkodtató jellegű házi feladatokat kapnak a diákok.

Eddig nem feltétlen mondtuk ki a diákok számára, hogy algoritmust tanulnak, használnak, de ilyenkor informatika órán már foglalkoznak magukkal az algoritmusokkal. Megfogalmazzák mi is az algoritmus, használják, feladatokat oldanak meg, leprogramozzák.

3.2.3. Középiskolai évek

Középiskolában már mélyebben elmerülhetünk az algoritmusokban informatika órán. Elemezhetjük őket futási idő, hatékonyság szerint. Itt már komolyabb problémák megoldása elé is állíthatjuk a diákokat. Modellézhetnek matematikai feladatokat, készíthetnek hozzá programot, saját játékot.

3.3. Milyen eszközök és programok segíthetnek?

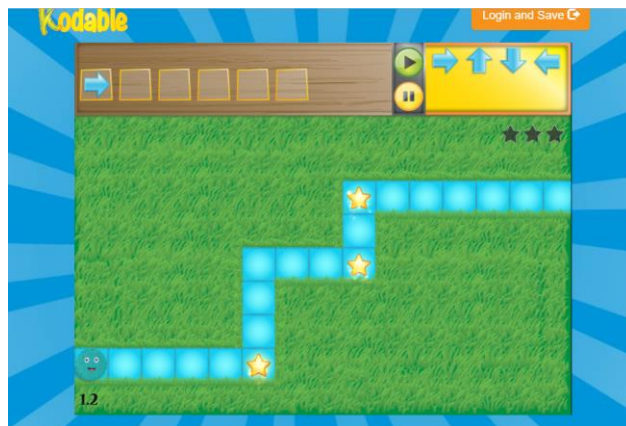
Az algoritmikus gondolkodás fejlesztéséhez olyan úton kell elindulni, ami felkelti, majd pedig meg is tartja a fiatalok figyelmét. Mi lenne erre alkalmasabb, mint a játékba ágyazott programozás, és algoritmusok készítése?

Azon, hogy a gyerekek naphosszat számítógépes játékokat játszanak, már senki nem lepődik meg. Tudományterületek foglalkoznak azzal, hogy ezeket a mozgatórugókat hogyan kellene beépíteni az oktatásba. Kézenfekvő tehát az ötlet, hogy játékokon keresztül fejlesszük a képességeket. Számos lehetőségünk van arra, hogy egészen fiatal kortól bevonjuk az alaptól informatikai eszközöket használó gyerekeket a programozás és az algoritmusok világába.

A legtöbb fejlesztő játék alap problémából, egyszerűen megoldható feladatból, egy gyorsan elérhető sikerből indul ki, és szépen lassan halad a nehezebb, komplexebb, és egészen sok gondolkodást igénylő feladatok felé. A következőkben bemutatunk néhány olyan játékos programot, mely nagyban segítheti az algoritmikus gondolkodás fejlődését.

3.3.1. Kodable (<https://dashboard.kodable.com/#/login>)

A Kodable nevű fejlesztő program 5-8 éves gyerekek számára készült, de idősebb korban is érdekes lehet. A fiatal gyerekeket első használatkor megragadhatja a kis színes szörnyecske, aki a játékunk főszereplője. A feladat röviden az, hogy a szörnyet ki kell vezetni az aktuális labirintusból. Több helyes útvonalból kell a legrövidebbet, a leoptimalisabbat kiválasztani, és megadni. Tehát létre kell hozni a legjobb algoritmust. Azt, hogy merre haladjon a szörnyünk, kártyákkal adhatjuk meg. A feladatok egyre nehezebbé válnak, egyre elgondolkodtatóbbak a kártyák utasításai, illetve a ciklusok szervezése. A játék használatához írni, olvasni sem szükséges, így tényleg optimális a legfiatalabbak érdeklődésének felkeltésére, és gondolkodásuk fejlesztésére.



1. ábra: Kodable egy feladata

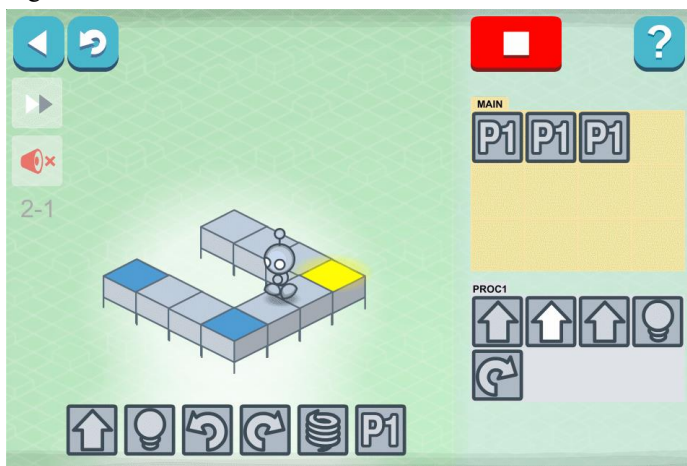
A játék használatához a szülőnek vagy tanárnak regisztrálnia kell a Kodable.com oldalon. Táblagépre történő letöltése a Google Play áruházon keresztül lehetséges.

3.3.2. Lightbot (<http://lightbot.com/flash.html>)

Azok számára, akik már végigjárták a Kodable játékot, remek továbblépést jelenthet a Lightbot nevű játék, melyet 9 éves kortól ajánlanak, de akár felnőttek számára is hasznos, és érdekes feladványokat rejt magában. A játékban egy robotot kell irányítanunk. Minden sötétkék

négyzetet sárgává kell változtatnunk úgy, hogy a robotunkkal felkapcsoljuk a lámpákat a kék négyzeteken. Jól végig kell gondolni, hogy melyik lépés után melyik következik. Lehetőségek lépésre: előre menni, elfordulni jobbra vagy balra, ugrás a magasabban elhelyezkedő négyzetre. Fejleszti a a probléma egészében vizsgálatát, részekre bontását és a lépésről lépésre tervezést.

A játékban 3 szint közül választhatunk. Az első szinten csak egyszerű, elemi feladatokat kell végrehajtani, az utasításokat egymás után kell behelyezni. A második szinten eljárásokat kell írni, amit utána beágyazunk a futtatott programba. Érdekes, hogy tapasztalataink alapján az ilyen függvények és eljárások működésének megértése nehezen megy azoknak, akik még kezdők a programozás világában, ebben a játékban pedig egyszerűen, alkalmazva tanulható meg a működésük. A harmadik szinten előkerül a rekurzió alkalmazása. Itt eljárásokat kell eljárásból meghívni. Ez a része a játéknak már több fejtörést és gondolkodást igényel, így ezek a feladatok már nehezebben megoldhatók.



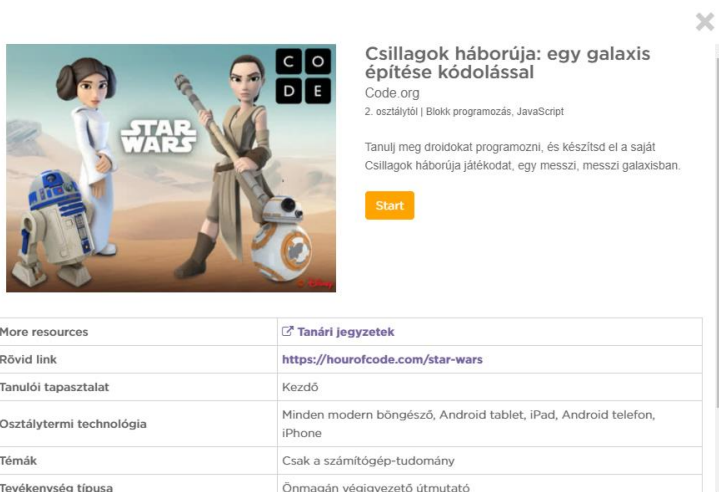
2.ábra: Lightbot második szintjének egy feladata

A játék próbaveziója elérhető a <http://lightbot.com/flash.html> címen, de megvásárolható táblagépre a teljes verzió a Google Play áruházban 2,34 USD- ért (kb. 630 HUF).

3.3.3. Code.org (<https://code.org/learn>)

A Code.org egy olyan weboldal, ami biztosítani szeretné, hogy mindenkihez eljuthasson a számítástechnika, és ne csak a tanórán lehessen foglalkozni vele. *“Minden diáknak, minden iskolában meg kellene adni a lehetőséget, hogy számítástechnikát tanuljon”*- ez a mondat áll a főoldalukon. Több projektjük is van, ami ezen a vonalon indult el, a legnépszerűbb pedig a **Kódolás órája (Hour of Code)** nevű globális mozgalom, melynek célja, hogy bepillantást engedjen a számítástechnikába, és a programozás alapjaiba. A Kódolás órája elérhetővé tesz számunkra megannyi tevékenységet, oktató játékot, melyeket feldolgozhatunk tanórai kereteken belül is.

Egy-egy játékra kattintva előugrik egy ablak, mely betekintést nyújt a játékba. Megadják, hogy milyen kortól ajánlott, hogy milyen programnyelveket, milyen programozási ismereteket fejleszt. Elérhető egy tanári jegyzet is, és tanulói tapasztalatok alapján besorolja kezdő vagy kényelmes nehézségi szintre. Hasznos még számunkra az is, hogy megadja, hogy milyen osztálytermi technológiára van szükségünk az adott feladat megoldásához.



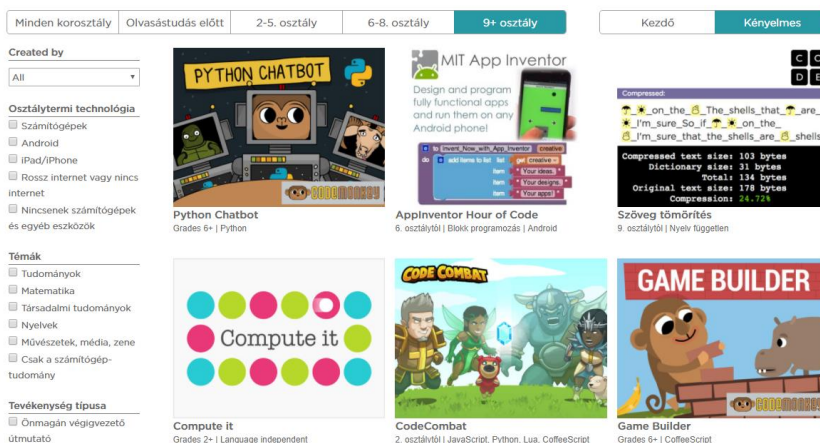
Csillagok háborúja: egy galaxis építése kódozással
Code.org
2. osztálytól | Blokk programozás, JavaScript

Tanulj meg droidokat programozni, és készítsd el a saját Csillagok háborúja játékokat, egy messzi, messzi galaxisban.

Start

More resources	Tanári jegyzetek
Rövid link	https://hourofcode.com/star-wars
Tanulói tapasztalat	Kezdő
Osztálytermi technológia	Minden modern böngésző, Android tablet, iPad, Android telefon, iPhone
Témák	Csak a számítógép-tudomány
Tevékenység típusa	Önmagán végigvezető útmutató

3. ábra: Csillagok háborúja: egy galaxis építése kódozással alkalmazásra kattintva felugró ablak



Minden korosztály | Olvasástudás előtt | 2-5. osztály | 6-8. osztály | **9+ osztály**

Kezdő | **Kényelmes**

Created by: All

Osztálytermi technológia:
 Számítógépek
 Android
 iPad/iPhone
 Rossz internet vagy nincs internet
 Nincsenek számítógépek és egyéb eszközök

Témák:
 Tudományok
 Matematika
 Társadalmi tudományok
 Nyelvek
 Művészetek, média, zene
 Csak a számítógép-tudomány

Tevékenység típusa:
 Önmagán végigvezető útmutató

Python Chatbot
Grades 6+ | Python

MIT App Inventor
Design and program fully functional apps and run them on any Android phone!
AppInventor Hour of Code
6. osztálytól | Blokk programozás | Android

Compressed
 I'm sure that the shells are...
 Compressed text size: 103 bytes
 Dictionary size: 31 bytes
 Total: 134 bytes
 Original text size: 178 bytes
 Compression: 24.7%

Szöveg tömörítés
9. osztálytól | Nyelv független

Compute it
Grades 2+ | Language independent

CodeCombat
2. osztálytól | JavaScript, Python, Lua, CoffeeScript

Game Builder
Grades 6+ | JavaScript

4. ábra: Code.org listázott játéka, szűrési feltételei

Sok játék tárul elénk, melyet bármikor kipróbálhatunk. Lehetőségünk van korosztály, nehézség (kezdő - kényelmes), osztálytermi technológia, téma, tevékenység típusa, időtartam és nyelv szerint szűrni a feladatokat, mely kényelmessé, és jól szervezhetővé teszi a tanórába való beépítéshez. Mindezek mellett lehetőség van osztályként és tanárként is regisztrálni az oldalra, így a tanár láthatja, hogy melyik diáknak melyik feladattal gyűlt meg a gondja, mennyit gyakorolt, ellenőrizheti a munkáját.

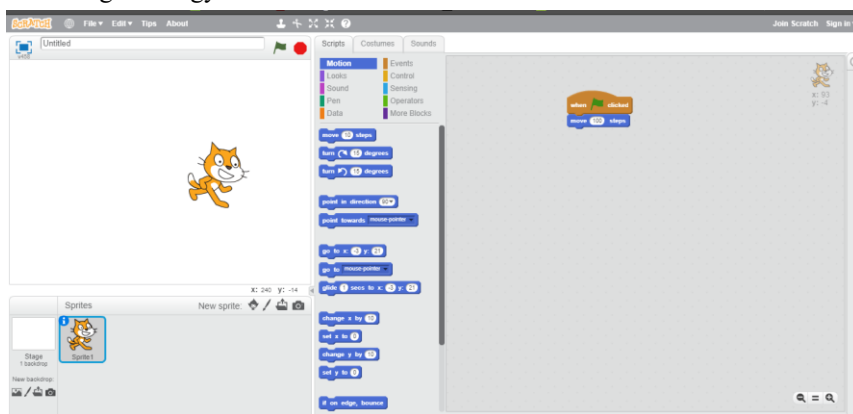
3.3.4. Scratch (<https://scratch.mit.edu>)

A Scratch programozási környezet alkalmas lehet általános iskolai és középiskolai diákok algoritmikus gondolkodásának fejlesztésére egyaránt. A program 7 éves kortól ajánlott, viszont már elkészült a Scratch Jr., amely az 5-7 éves gyerekeket is bevonja. Ebből is látszik, hogy a cél

az, hogy minél korábban elkezdjük a gyerekek gondolkodásmódját a problémamegoldásra, az algoritmikus gondolkodásmód felé terelni. A Scratch mottója: *“A kódolás az új írástudás!”*.

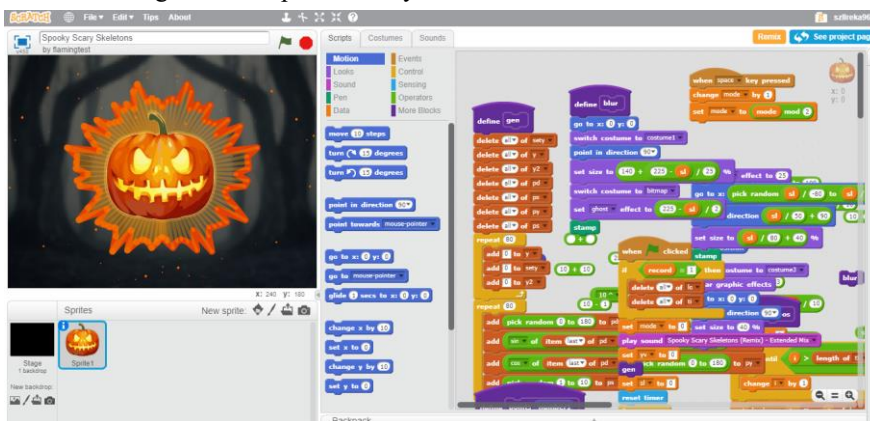
Ebben a programban egy macska áll a középpontban, őt irányítjuk, de lecserélhetjük a már meglévő szereplők egyikére, és létrehozhatunk sajátot is. Tervezhetünk vele saját számítógépes játékot, applikációt, melyet megoszthatunk a teljes Scratch közösséggel és visszajelzést kaphatunk róla. A gyerekek kiélhetik a kreativitásukat, a szereplő mellett a háttér is átalakítható, így tetszés szerint megtervezhetik a saját alkalmazásukat. Belekóstolhatnak abba, hogyan kell egy egész projektet felépíteni, a részletenkénti tervezésbe. Végig kell gondolni, hogy melyik lépés után melyik következik, milyen algoritmust kell kialakítani a helyes megoldáshoz, viszont nem kell alkalmazniuk semmilyen előzetes programozási tudást. Nem parancsokat írogatnak, nem kell megtanulniuk a szöveges utasításokat, egyszerűen összekapcsolják a megfelelő utasításokat egymás után, és futtatják. A parancsok köznyelven megfogalmazottak, így könnyen kezelhető bárki számára. A beépített parancsok egymásba ágyazása kizárja, hogy szintaktikai hibát vétsenek a gyerekek, így magára a mechanizmusra kevésbé kell koncentrálniuk, a teljes figyelmük a probléma megoldására irányulhat.

Megadhatunk egészen egyszerű utasítássorozatokat:



5. ábra: Scratch alapbeállításokkal, egyszerű utasítás

De létrehozhatunk egészen komplex és bonyolult feladatot is:



6. ábra: Egy összetett feladat Scratchben

3.3.5. Code Combat (<https://codecombat.com/play>)

A Code Combat egy olyan platform, ami segíti a programozás tanulását, fejleszti a gondolkodást, mégpedig úgy, hogy egy igazi számítógépes játékban érezzük magunkat. Alkalmazható osztálytermen belüli oktatásra is. Kihasználható, hogy szinteket kell elérni, versenyeztetni lehet a diákokat.

Az előzőektől eltérően itt kódot kell begépelni. Nem kell hozzá előzetes tudás, minden segítséget megkapunk ahhoz, hogy sikerüljön teljesíteni az egyes szinteket. Segítséget nyújt az órába integráláshoz, oktatási segédanyagokat ad, és hozzáférhető mindenki számára. A játék nagyon jól jutalmazza az úgynevezett *“alkotó erőfeszítést”*, ami nem egyhangú tanuláshoz vezet, hanem magával ragad. Minden esetben rögtön megkapjuk a visszajelzést, így azonnal helyesbíthetünk, javíthatunk a megírt kódunkon.

A játékban egy hős karaktert kell választanunk magunknak, és minden alkalommal egy akadálypályán kell végigvezetnünk őt az utasítások segítségével. Több jó megoldás is létezik, viszont több pontot lehet szerezni, ha a legjobb algoritmust írjuk meg. A szintek előrehaladtával arra is figyelniük kell, hogy hány sor utasítást írunk, illetve hány lépést teszünk meg összesen.

A program főként a középiskolai korosztály számára ajánlott. Ebben a játékban nem csak gondolkodási mód, algoritmusok elsajátítása a cél, hanem programozási nyelv megtanulása is. A játék kezdetekor választani lehet, hogy Python vagy JavaScript nyelven akarunk tanulni. Az utasítások fel vannak sorolva számunkra, nekünk csak meg kell érteni, hogy hogy működnek, majd helyesen, a helyes sorrendben használva be kell írunk a programunkba. A megértést nagyban elősegíti az, hogy a parancsokra kattintva előugrik annak magyarázata, példaprogramokkal ellátva.



7. ábra: Code Combat egy egyszerű feladata

4. Összegzés

Az algoritmikus gondolkodást számos területen használjuk, csakúgy mint az olvasást, írást, beszédet. Kézenfekvő tehát, hogy azt is fiatal korban el kell kezdeni fejleszteni. Ennek módja többféle lehet. Legyen az életszerű példák, kézműves foglalkozásokon, számítógépes játékokon keresztül, mind fontos, hasznos. A lényeg, hogy olyan módszerrel próbálkozzunk, amely az adott csoportot, gyereket érdekli, és fenntartja érdeklődését.

Irodalom

1. Nemzeti alaptanterv 2012
2. Lakó Viktória: Tanulói típushibák az informatika érettségi algoritmizálás feladataiban
In: Szlávi Péter, Zsakó László (szerk.) INFODIDACT 2014 – Informatika Szakmódszertani Konferencia. Konferencia helye, ideje: Zamárdi, Magyarország, 2014.11.20-2014.11.21. Budapest: Webdidaktika Alapítvány, 2014. Paper 1.. 8 p. (ISBN:9789631206272)
3. Németh Tamás, Sárkány Rita, Tornai Henrietta, Wiandt Zsófia, Németh-Szabados Klára Viktória, Holló Csaba: A programozás oktatásának motivációi a közoktatásban
In: Szlávi Péter, Zsakó László (szerk.) – INFODIDACT 2016: Informatika Szakmódszertani Konferencia. Konferencia helye, ideje: Zamárdi, Magyarország, 2016.11.24-2016.11.25. Zamárdi: Webdidaktika Alapítvány, 2016. Paper programozás. 7 p. (ISBN:978-615-80608-0-6)
4. <https://www.kodable.com>
5. <https://hourofcode.com>
6. <https://code.org>
7. <http://gyermekapp.hu/algoritmikusok-oktatasa-gyerekeknek>