

MICROSOFT VISUAL PROGRAMMING LANGUAGE A SZOFTVERFEJLESZTÉS OKTATÁSBAN

Illés Attila

Abstract

Nowadays we found some relevant problem in the education. Most of the student don't study and don't programming in any programming languages. There is not enough motivation to get success. Few visual programming language are good for education, but we need a global and useful visual programming language, which is solve our student's programming problem.

The Microsoft made a visual language, which name is MS Visual Programming Language (VPL). This language make algorithm with graphical objects such as LabView. The VPL is consist of Microsoft Robotics Studio, which made for simulation enviroment of robotics model.

The VPL easy to use programming language.

Összefoglalás

Napjainkban, a felsőoktatásban szignifikáns szintkülönbségeket figyelhetünk meg a hallgatók programozási képességeiben. Jelentős mértékben csökkent a hallgatók motivációja és érdeklődése az algoritmikus gondolkodás programozási nyelven történő megvalósítására. Néhol megfigyelhető a matematikai ismeretek hiányossága, hézagossága, de a legfőbb probléma mégis az önálló probléma megoldás. A kiadott feladat feldolgozása, osztályozása programozás-technikai szempontok alapján nehezen, sőt sokszor egyáltalán nem vezet eredményre. A helyes algoritmus megtalálása lenne az első lépés, ehhez a kreatív gondolkodáson kívül szükség lehet némi lexikális ismeretre is. A programozás oktatására számos megoldási mód létezik kisebb-nagyobb eredményességgel, de a lényeg bármely módszer használatakor az érdeklődés felkeltése. Ha ez sikerült, akkor már megnyertük a hallgató figyelmét, és eredményes lesz a programozási feladatok megoldása során.

Egy módszert kínál a vizuális programozás is. Számos ilyen nyelv és környezet készült a programozási feladatok egyszerű megvalósíthatóságára, gondoljunk csak a LabView lehetőségeire. A most bemutatni kívánt módszer egy a fiatalok által kedvelt készségfejlesztő játék, a Technic Lego vezérlésére kialakított környezet használata. Ezt a Microsoft fejlesztette ki és adta neki a Visual Programming Language nevet.

1. Bevezetés

A mérnök-informatikus hallgatók jelentős része gyengén szerepel a szoftverfejlesztéssel kapcsolatos tárgyakkól, nem ritka, hogy egy-egy tantárgyat háromszor is kénytelenek felvenni. Bár e lazaság kialakulására kétségtelenül hatással volt a kreditrendszer által nyújtott szabadság, de a probléma mélyebben gyökerezik. A jelenség okait számos szerző vizsgálta, illetve próbált megoldást találni a módszertan és a tananyagot érintő újítások, módosítások bevezetésével.

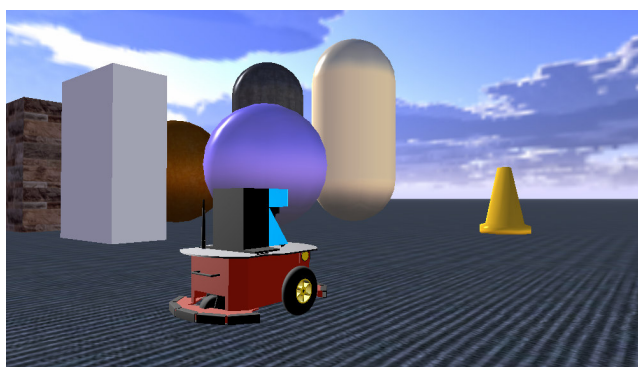
Tóth és Johanyák [3] a versenyszféra igényeiből és a közismert szoftverfejlesztési hibákból kiindulva a hangsúlyt a szoftvertechnológiai alapokra és a csoportmunkára helyezve kereste a kiutat. Johanyák és Szabolcsi [1] a gyors alkalmazásfejlesztési technikák és a Visual Studio – C# eszközpáros által

nyújtott vizuális programozás-támogatás oktatásba történő bevezetéséről számolt be. Egy későbbi felmérés kiértékelése [2] kimutatta, hogy bár az új megoldások javulást eredményeztek és bizonyos programozási hibák eltűnéséhez vezettek, azonban összességében hatásuk elmaradt a várakozásoktól. További lehetőségek után kutatva olyan eszközöket, módszereket kerestünk, amelyek segítségével két „fronton” is eredményekre számíthatunk. Egyrészt a hallgatók érdeklődésének felkeltésére, másrészt a rendszerezett gondolkodásra kívántunk összpontosítani.

A jelen dolgozatban röviden áttekintjük a Microsoft Robotics Studio által támogatott Visual Programming Language által nyújtott szolgáltatásokat.

2. Visual Programming Language

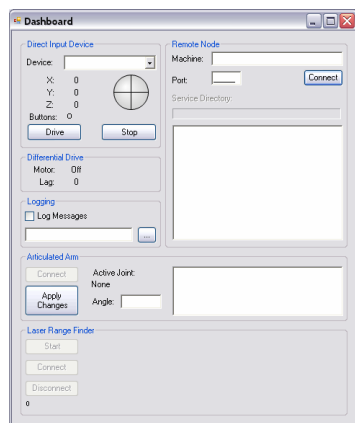
Sokan még tanultunk folyamatábrás tervezést az általános iskolás évek alatt, ám a mai főiskolás hallgatók, mintha elfelejtették volna a használatát. Talán igazolja a mai szoftverek bonyolultsága azt, hogy nem lehet folyamatábra segítségével tervezni, de egy alap algoritmus felépítésének megértése elvárt képesség egy informatikus hallgatótól.



1. ábra: *Pioneer 3DX szimuláció*

A Visual Programming Language, továbbiakban VPL képes egyszerű eszközkészlet segítségével megvalósítani bonyolult algoritmusokat. Eszközkészlete jóval túlnyúlik az egyszerűségeen. Számos Lego robot alkatrészhez kínál interfészt, amely segítségével a VPL-en keresztül megtervezhetjük a robot működését leíró programot. Lehetőségünk van hálózaton keresztül vezérelni mobil robotunkat, amelytől akár adatot is fogadhatunk. Így módunkban áll szenzorokat, különféle szkennereket elhelyezni a robot vázán. A teljes interaktivitás elérése érdekében a szenzorok mellett megtalálható a motorok vezérléséhez szükséges eszközkészlet is.

Mielőtt arra gondolnánk, hogy mennyibe is kerül nekünk ez a kis fejlesztés, figyeljük meg, hogy mit kínál még számunkra a VPL. A lehetőségek között szerepel egy virtuális szimulációs környezet, amely segítségével virtuálisan tesztelhetjük szoftverünket egy robotmodell segítségével. Ezt a modellt éppúgy felszerelhetjük különféle szenzorokkal, mint a való világban. A Microsoft Robotics Studio lehetővé teszi ennek a szimulációs környezetnek (1. ábra) a felépítését. A VPL eszközkészletében lévő elemeket összekapcsoljuk a szimulációs térben lévő objektumokkal, és már indíthatjuk is a szimulációt. Ha szeretnénk interaktív tenni, akkor készíthetünk egy panelt, amelyen elhelyezhetünk dinamikus paraméterek, és azokat futási időben

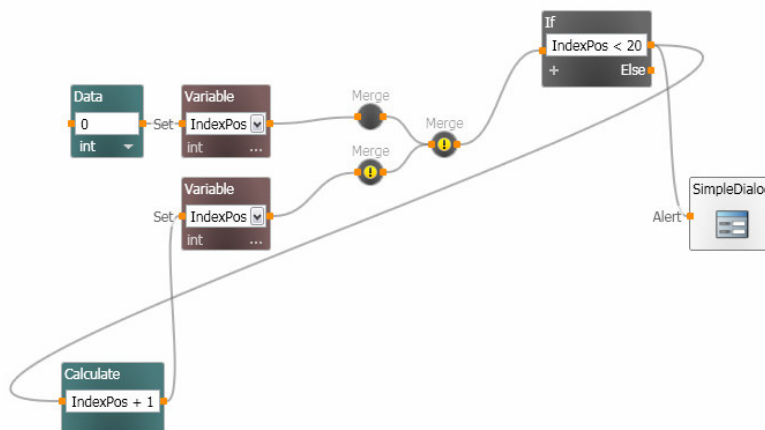


2. ábra: *Vezérlőpanel*

változtathatjuk meg. Ezzel a technikával távolról avatkozhatunk be robotunk életébe, vagy ha a szimulációt választjuk, akkor a virtuális robotunk életébe.

De térjünk vissza a VPL-hez, amelyet módszerként tekinthetünk az egyszerűbb algoritmusok kódolására. Az eszköztár elemei között van változó definiálására alkalmas elem is 'Variable', illetve vezérlési szerkezetek közül 'if' és 'switch' elágazás is. Változó értékadását lehetővé tevő 'Data' elem és a futásidejű megváltoztatását lehetővé tevő 'Calculate' elem. Lehetőségünk van lista készítésére is, amely megfelel a tömb fogalmának is.

A vezérlési szerkezeteknél még az is jelentősen elősegíti munkánkat, hogy több feltételt tudunk megadni, illetve a 'switch'-nél is több alternatívát lehet felsorolni. Ha egy bonyolult felépítést szeretnénk megjeleníteni, akkor célszerű a részegységeket modulokká



3. ábra: Változó értékadása

alakítani. Erre a feladatra használható az 'activity', amely segítségével kis dobozokat hozhatunk létre az egyes kiemelt részegységekből. Például, ha egy minimumkereső egységet szeretnénk létrehozni, akkor meg kell adni bemenő paraméternek a tömböt, és a kimenethez hozzácsatolni egy értéket. Ez egy tipikusan sorozathoz érték rendelése feladat. Ezt a megalkotott modult a későbbi fejlesztésben már, mint egy modult használhatjuk. Jelentősen megnöveli a program átláthatóságát, illetve esztétikailag is jobban mutat.

A 3. ábrán látható egy egyszerű ciklus, amelyben definiálunk egy változót, ez lesz az Indexpos változó, és hozzárendelünk egy 0 értéket. Behelyezünk egy if feltételt is, amely a változónk 19-es értékéig növeli annak értékét egy visszatérő ágban. Mindeközben a SimpleDialog elemmel kiírathatjuk a képernyőre a változó értékét. E néhány elemet felhasználva készíthetünk egyszerűen algoritmusokat, amelyeket hozzákapcsolhatunk a Microsoft Robotics Studio szimulációs környezetéhez, és irányíthatunk objektumokat, illetve lekérdezhethetjük az objektumok paramétereit. Ez a módszer kísérleti stádiumban van, számos területén kell még finomításokat eszközölni, de reményeink szerint lesz olyan terület, ahol nélkülözhetetlen lesz a programozás oktatásában.

A Microsoft Robotics Studio és ezen belül a Visual Programming Language használatában kiaknázatlan lehetőséget rejlenek.

3. Összefoglaló

A mérnök-informatikus hallgatók körében tapasztalható egyre lanyguló programozás iránti érdeklődés és a gyenge tanulmányi eredmények újabb módszertani és tananyagfejlesztési megközelítések keresésére ösztönöznek.

A Robotics Studio és a VPL alkalmazásától két téren remélünk javulást. Egyrészt a vizuális eszközök és a szimulációs környezet a hallgatók érdeklődésének erősödését eredményezhetik, másrészt a tervezett, rendszerezett megközelítés javulására számítunk.

Irodalom

- [1] Johanyák, Zs. Cs., Szabolcsi, J.: Experiences of teaching visual programming with C# and Visual Studio 2005, Pollack Periodica, Vol. 2, Suppl., 2007, pp.97-105.
- [2] Johanyák, Zs. Cs., Tóth, Gy. F.: Vizuális módszerek oktatásának hatása a hallgatók programozási hibáira, Matematika-, fizika- és számítástechnika oktatók XXXI. konferenciája, Dunaújváros, 2007. augusztus 23-25., pp. 126-131.
- [3] Tóth, Gy. F., Johanyák, Zs. Cs.: Teaching software engineering - Experiences and new Approaches, XIX Didmattech 2006, September 6-7, Komarno, Slovakia , pp. 261-265.
- [4] Segédlet a VPL megismeréséhez, fejezetekre bontva
[http://msdn2.microsoft.com/hu-hu/library/bb483087\(en-us\).aspx](http://msdn2.microsoft.com/hu-hu/library/bb483087(en-us).aspx)
- [5] Általánosságban a VPL-ről.
http://en.wikipedia.org/wiki/Visual_programming_language
- [6] Online video anyag a VPL elsajátításához
<http://msdn2.microsoft.com/en-us/robotics/bb383569.aspx>

Illés Attila

Munkahely: Kecskeméti Főiskola GAMF Kar,

Kalmár Sándor Informatikai Intézet, Elektrotechnika és Kibernetika Szakcsoport

Cím: 6000, Magyarország, Kecskemét, Izsáki út 10.

Telefon / Fax: +36-76-516447

E-mail: illes.attila@gamf.kefo.hu