

Szegedi Innovatív Informatika Verseny

2015

A Szegedi Tudományegyetem informatika versenye középiskolásoknak

The logo for SZIIV (Szegedi Innovatív Informatika Verseny) is a blue circle containing the text "SZIIV" in white. The background of the entire page features a grid of large white binary digits (0s and 1s) on a grey background. The central circular graphic is divided into four quadrants by a blue cross. The top-left quadrant shows a computer monitor displaying C++ code with static variables for a 3D scene. The top-right quadrant shows a 3D landscape with green hills and red trees. The bottom-left quadrant shows a close-up of a computer motherboard with various components. The bottom-right quadrant shows a close-up of a network switch or router with several ports and cables.

SZIIV

Alkoss SZIIV-vel, lélekkel!

Program

Időpont	Műszaki informatika szekció (Díszterem)	Informatika szekció (Informatórium)
9:00–9:45	Regisztráció	
9:45–11:00	Előadások I.	
9:45–10:00	Megnyitó	Megnyitó
10:00–10:20	Linux distributions: <i>Néma riasztás, avagy silent alarm</i>	A jövő reménye: <i>Hamal</i>
10:20–10:40	ERROR: <i>Automatizált étterem szimuláció LEGO Mindstorms NXT robotokkal</i>	AkosSoft: <i>MSG</i>
10:40–11:00	ESZI szakkör: <i>Mosógépvezérlés</i>	BetonSoft: <i>CuStudy</i>
11:00–11:20	Kávészünet	
11:20–13:00	Előadások II.	
11:20–11:40	MPD: <i>Rajintelligencia viselkedés megvalósítása robotokkal</i>	Budavári Márk: <i>Wise</i>
11:40–12:00	SMARTduino: <i>Időjárás előrejelzés alapú intelligens fűtésszabályozás</i>	Lia: <i>Lia – az ételérzékenyek asszisztense</i>
12:00–12:20	Eszközök bemutatása	Marcell Games: <i>LightLabirinth</i>
12:20–13:00		Szoftverek bemutatása
13:00–14:00	Ebéd	
14:00–15:00	Plenáris előadások (Díszterem)	
15:00–15:30	Eredményhirdetés (Díszterem)	

Támogatóink



National Instruments
hungary.ni.com



NNG
www.nng.com



Magyar Mérnöki Kamara
www.mmk.hu



Szent-Györgyi Albert Agóra
www.agoraszeged.hu



**SZTE TTIK Informatikai
Tanszékcsoport**
www.inf.u-szeged.hu

A kiadvány az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával készült.

Projektazonosító: **TÁMOP-4.1.1.F-14/1/KONV-2015-0006**

„Munkaerő-piaci igényekre támaszkodó gyakorlatorientált képzések, szolgáltatások a Szegedi Tudományegyetem fókuszában”

Tartalomjegyzék

<i>Program</i>	1
<i>Támogatóink</i>	2
<i>Előszó</i>	5
<i>Programbizottság</i>	6
<i>Műszaki informatika szekció</i>	11
Zsibók Márk: Néma riasztás, avagy silent alarm.....	13
Szakali Benedek, Finta Imre: Automatizált étterem szimuláció LEGO Mindstorms NXT robotokkal	17
Biró Tamás, Kovács Tamás: Mosógépvezérlés.....	21
Coulibaly Patrik, Csontos Dávid, Kiss Máté: Rajintelligencia viselkedés megvalósítása robotokkal.....	25
Debnárik Roland, Klein Armand: Időjárás előrejelzés alapú intelligens fűtésszabályozás.....	29
<i>Informatika szekció</i>	35
Kiss Benedek, Huszka Adrián, Nagy Roland: Hamal.....	37
Juhász Ákos: MSG	41
Guzsik Dávid József, Kiss Antal, Mészáros Bálint: CuStudy	45
Budavári Márk: Wise	49
Kovács Márk, Lator Anett, Benkovics Ernő: Lia - az ételérzékenyek asszisztense	53
Kovács Marcell, Tóth Máté: Light Labyrinth.....	57
<i>Informatikus képzés a Szegedi Tudományegyetemen</i>	63



Előszó

A Szegedi Tudományegyetem Informatikai Tanszékcsoportja immáron második alkalommal hirdetett versenyt a középiskolás diákok számára, ahová ezúttal már két szekcióban is várta a jelentkező diákok pályamunkáit. A verseny kapcsán célunk volt lehetőséget adni a tehetséges és kreatív középiskolás diákoknak, hogy a műszaki informatika vagy a szoftverfejlesztés területén elért eredményeiket bemutathassák és megmérettessék.

A verseny keretein belül ezúttal sem kötöttük meg a diákok kezét és kreativitását, bármilyen informatikai témájú probléma vagy feladat megoldásával kapcsolatos pályamunkával nevezhettek az alábbi két szekció valamelyikére:

A **Műszaki Informatika** szekcióba olyan pályamunkák beérkezését vártuk, melyeknél csupán egyetlen kikötésünk volt: a feladat ne csupán programozási feladatból álljon. A diákoknak meg kellett építeni, vagy már meglévő elemekből összeállítani egy rendszert, amely működtetéséhez szükséges szoftvert is a diákoknak kellett elkészítenie.

Az **Informatika** szekcióba olyan pályamunkákat vártuk, amelyekben a diákok egy elkészített szoftvert mutatnak be. A fejlesztéshez tetszőleges programozási nyelv és programozói függvénykönyvtár használható volt. A szoftver kategóriáját illetően nem tettünk megkötést, lehetett játék, asztali számítógépen futtatható alkalmazás, webes alkalmazás, mobil applikáció, vagy Kinect-es alkalmazás.

A versenyre végül a két szekció zsűrije összesen 8 település iskoláiból érkező 11 csapatot válogatott be a szegedi döntőbe, mely 2015. október 16-án került megrendezésre a Szent-Györgyi Albert Agórában.

Ez a kiadvány a döntőn bemutatásra kerülő 11 pályamunkát foglalja egy kötetbe.

Szeged, 2015. október

Vadai Gergely

a kiadvány szerkesztője

Programbizottság

Műszaki informatika szekció

A zsűri elnöke:



Dr. Gingl Zoltán
tanszékvezető egyetemi tanár
SZTE Műszaki Informatika Tanszék

A zsűri tagjai:



Dr. Pletl Szilveszter
főiskolai tanár
SZTE Műszaki Informatika Tanszék



Dr. Mingesz Róbert
adjunktus
SZTE Műszaki Informatika Tanszék



Engedi Antal
elnökségi tag
Csongrád Megyei Mérnöki Kamara

Informatika szekció

A zsűri elnöke:



Dr. Nyúl László
tanszékvezető egyetemi docens
SZTE Képfeldolgozás és Számítógépes Grafika Tanszék

A zsűri tagjai:



Dr. Balázs Péter
egyetemi docens
SZTE Képfeldolgozás és Számítógépes Grafika Tanszék



Dr. Beszédes Árpád
adjunktus
SZTE Szoftverfejlesztés Tanszék



Fritz Zsombor
fejlesztési központ vezető, Szeged
NNG Kft.

Szervezők



Dr. Kincses Zoltán
adjunktus
SZTE Műszaki Informatika Tanszék



Dr. Németh Gábor
adjunktus
SZTE Képfeldolgozás és Számítógépes Grafika Tanszék



Dr. Siket István
adjunktus
SZTE Szoftverfejlesztés Tanszék



Dr. Varga László Gábor
adjunktus
SZTE Képfeldolgozás és Számítógépes Grafika Tanszék



Vadai Gergely
tanársegéd
SZTE Műszaki Informatika Tanszék

Műszaki informatika szekció

Néma riasztás, avagy silent alarm

Linux distributions

Zsibók Márk

Karolina Óvoda, Általános Iskola, Gimnázium, Alapfokú Művészeti Iskola és Kollégium, 6725 Szeged, Szentháromság u. 70-76.

1. Bevezetés

9. évfolyamos tanulóként szeretnék bekapcsolódni a műszaki informatika innovatív világába. Régóta várok erre a pillanatra! A pályázati felhívásról két hete értesültem. Egy olcsó megoldást kerestem az értékeink védelme érdekében.

Az emberek, ha lakásukat szeretnék bebiztosítani a betörés ellen, komoly összegeket fizetnek ki a szakemberek által telepített szerkezetekre.

Célom, hogy a társasházakban, lakótelepeken élő, szerényebb anyagi háttérrel rendelkező emberek részére egyszerű szerkezet létrehozása, mely biztonságot nyújt számukra. A berendezés a lakásban történő mozgás esetén, e-mailben értesíti a tulajdonost az illetéktelen behatolástól. A készülék egy aranyos gyerekjátéknak tűnik az ismeretlenek számára. Egy magasabb bútoron, elektromos eszközön (pl.:hűtő) elhelyezve, a bejárati ajtóra irányítva „látja be”, 170⁰-os szögben a védeni kívánt területet.

2. Probléma megoldásának menete

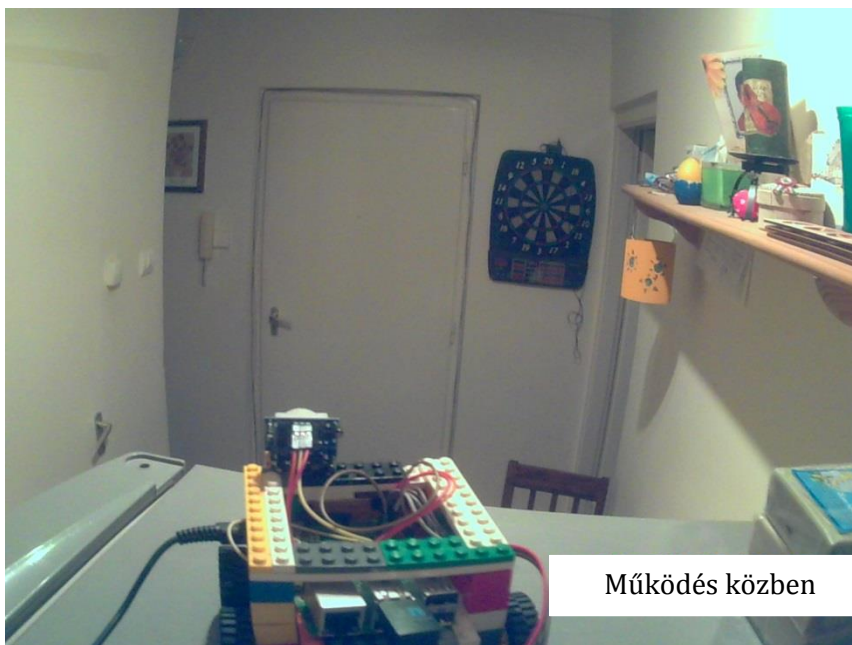
Az első feladat a hardware kiválasztása volt. Én az Element 14 által gyártott Raspberry Pi 2 eszközt használtam fel, mert kis méretű és kis fogyasztású. A Raspberry Pi az 1.ábrán látható . A programozása egyszerű, gördülékeny. Python nyelven írtam a védelmi programot. A kiegészítő hardwareket internetes rendeléssel szereztem be: egy infravörös mozgásérzékelőt, egy 4 csatornás relét és egy 300mbit/s wifi vevőt .

Az összeszerelés kezdetén megismerkedtem a Raspberry Pi GPIO portjaival, telepítettem a megfelelő drivereket, és kikísérleteztem az érzékelő tulajdonságait. A relé hozzákapcsolásával akár a hálózati áramról működő eszközöket is vezérelhet a mozgásérzékelő. Az e-mail küldést az smtp nyílt forráskódú programmal végeztem el. Fontos volt, hogy bárki, informatikai tudás nélkül is képes legyen használni a szerkezetet, ezért egy autorun parancsot programoztam bele a scriptbe. Így, amikor a tulajdonos elmegy otthonról a szerkezetet csak áram alá kell helyeznie. A wifi vevőnek köszönhetően, az egész lakásban elhelyezhető az eszköz. A wifi adapter a 4.

ábrán látható. Az utolsó feladat a LEGO autó megépítése volt, az álcázás és a jövőbeni fejlesztés céljából. A 2. ábrán működés közben látható a riasztórendszer.



1. ábra. Raspberry Pi



2. ábra. Működés közben

3. Elért eredmények

Általában az emberek rendelkeznek okos telefonnal. A nap bármely szakában biztonságban tudhatják értékeiket az otthonukban a néma riasztás eszköz segítségével. Előnye, hogy széles körben elterjedhet, mivel kis költség- és energia igényű szerkezet. Előzetes ismereteket illetve képzettséget nem igényel a használata. Hasznosságát mutatja, hogy a tulajdonos azonnal értesül a lakásában történt illetéktelen behatolásról a nála levő telefon segítségével, ugyanis e-mailt küld számára a berendezés. Nyugodtan töltheti az éjszakát is távol értékeitől, mivel bármilyen behatolásnál értesül arról. Saját lakásomban teszteltem a készüléket, több esetben is. Több e-mail címet adtam meg, így a családtagok mindannyian kaptak értesítést. Mivel kevés idő állt rendelkezésemre, így az alapgépet építettem meg. Tovább szeretném fejleszteni a riasztót: egy kamera modullal és egy GSM adó-vevővel, továbbá egy helyváltoztatás megvalósulásához szükséges alkatrész beépítésével. Ezek beüzemelése után akár képet is készíthet a betörőkről a készülék.

Automatizált étterem szimuláció LEGO Mindstorms NXT robotokkal

ERROR

Szakali Benedek, Finta Imre

Felkészítő tanár: Kiss Róbert

Kecskeméti Bányai Júlia Gimnázium, 6000 Kecskemét, Nyíri u. 11.

1. Bevezető

A XXI. század egyik vezető kutatási területe a robotika. A XX. századot pedig a kommunikáció évszázadának is szokás nevezni. A technológiai fejlődés a két terület ötvözésével hozott létre olyan autonóm rendszereket, amelyek a folyamatirányításban, termelésben, szolgáltatásban precizitásuk következtében helyettesítik a korábban emberi munkaerőre épülő ágazatokat. Az étteremautomatizálás is ilyen terület, ahol nem csak a folyamatszabályozás lehet automatikus, hanem a teljes szolgáltatási kör.

2. A koncepció

A csapat által elkészített szimuláció egy „robotizált” éttermet valósít meg, nyolc darab NXT minikomputer felhasználásával. A rendelésvételt, az ételek kiadását és az asztalokhoz szállítása is robotok végzik automatikusan. Az egyszerűség kedvéért a vendégeket is robotok jelképezik.

A projektben található programvezérelt elemek három különböző csatornán kommunikálnak egymással: bluetoothon, I²C porton keresztül és optikai jeltovábbítással.

A robotokon futó programok a karakteres C nyelvhez közeli NXC nyelven készültek.

2.1. A szimuláció részei

A „konyha”: A koncepció legnagyobb méretű konstrukciója, az étterem „irányító központja”. Itt történik a megrendelt ételek kiadása, illetve az asztalokkal és pincérekkel való kommunikáció koordinációja. A konyhát két NXT téglá vezérli, amelyek egymással I²C porton, illetve színszenzorok segítségével küldött optikai jeleken keresztül kommunikálnak.

Az „asztalok”: A szimulációban kettő van belőlük, de számuk tovább bővíthető. Az asztaloknál történik az ételek rendelése. Három féle étel közül lehet választani, amelyeket érintésszenzorok segítségével lehet

megrendelni, és vizualitás szempontjából színkódokban különböznek. A konyhával bluetooth kapcsolaton keresztül történik az adatcsere.

A „pincérek”: Két darab MINDSTORMS NXT robot végzi a kiszolgálást. Az egyes számú a megrendelt ételeket szállítja a konyhából, és helyezi el az asztalon. A kettes számú pedig az asztalok leszedését végzi. A konyhával bluetooth-on keresztül kommunikálnak.

2.2. Technikai részletek

A szimuláció során először a vendég odamegy az egyik asztalhoz, ahol három érintésszenzor segítségével kiválasztja, a rendelt ételt (az ételeket szimbolizáló elemek különböző színűek). Az asztal ezt továbbítja a konyhának.

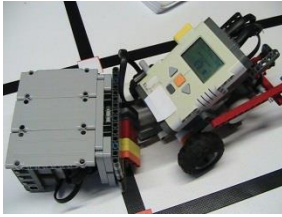
A konyha egyforma tálcaikat használ az ételek kiadásához. A tálcaikat mozgatását három motor végzi, a különböző ételek kiadását szintén három. A motorok futószalag segítségével továbbítják a tálcaikat. A tálcaikat mozgatása során a tálca pillanatnyi pozícióját egy színszenzorral figyeli a robot. A „konyha” a rendelt ételnek megfelelően a tálcaikat a megadott pozícióba mozgatja, majd adagolás után ráhelyezi a pincér 1. robotra.

A pincér 1. egy koordináta rendszerben színszenzorai segítségével tájékozódik és viszi a tálcaikat az asztalhoz, majd visszatér a kiinduló pozíciójába. A tálca asztalra helyezéséhez pontos pozicionálás és jól tervezett konstrukció szükséges. Mindehhez a robotok a modellben színérzékelő szenzorokat használnak és a felületre felfestett vonalhálózatot, amely a koordináta rendszer szerepét tölti be.

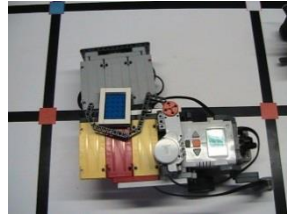
Az asztalnál elhelyezett érintésszenzor felengedésének hatására az asztal minikomputere elküld egy üzenetet a konyhának, amely utasítást küld bluetoothon keresztül a pincér 2-nek, amely szintén színszenzorai segítségével tájékozódik. Az asztalról tálcaikat egy tárolóba szállítja és visszatér a kezdő pozíciójába.

A teljes folyamatot a konyha irányítja, amely a megkapott jeleket értelmezve vezérli a pincérek mozgását és az étel adagolását.

2.3. Ábrák



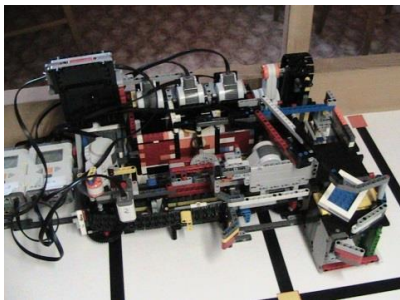
4. ábra. Ételválasztás



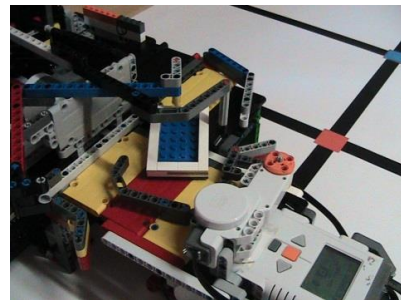
2. ábra. A pincer 1 (kirakodó)



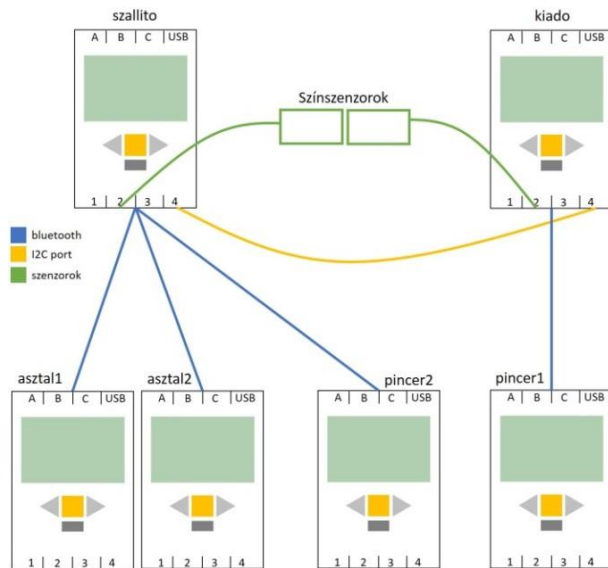
3. ábra. A pincer 2 (berakodó)



1. ábra. A „konyha”



5. ábra. Tálca a pincerre kerül



6. ábra. A hálózat kommunikációs modellje

3. Elért eredmények

A megvalósított szimulációban egy olyan éttermi modellt építettünk és programoztunk, amely autonóm módon képes kiszolgálni például egy menza közönségét. A koncepcióban a kommunikációs modell és megvalósítása volt programozás technikai szempontból kihívás. A robotkonstrukciók terén a konyha ételadagoló vezérlése tartalmaz sok egyedi ötletet.

Mosógépvezérlés

ESZI szakkör

Biró Tamás, Kovács Tamás

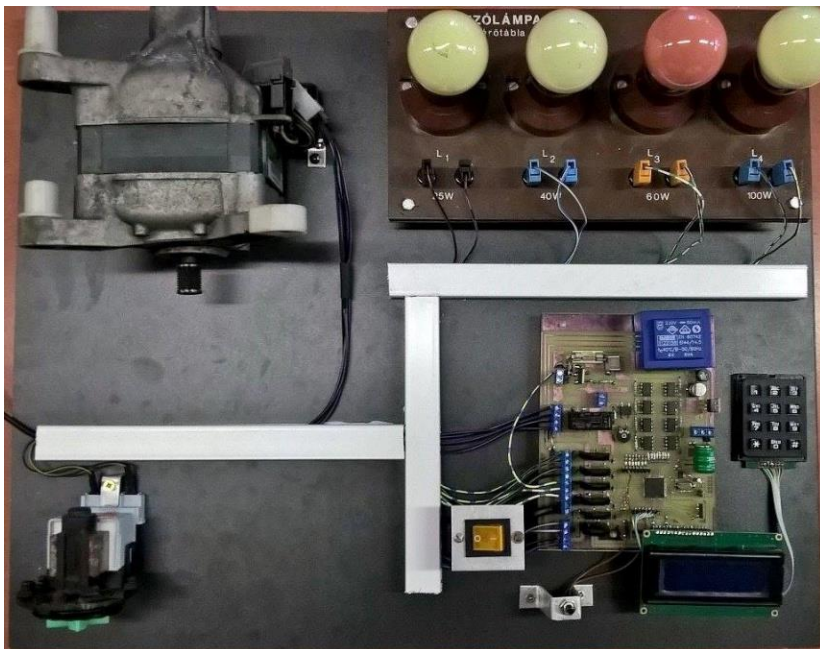
Felkészítő tanár: Manzinger József

Energetikai Szakközépiskola és Kollégium, 7030 Paks, Dózsa György út 95.

1. Bevezetés

A megépítés ötlete egy INDEZIT gyártmányú mosógép tönkremenetele során merült fel. A gép szivattyúja leégett, mely során a zárlat visszahatott a vezérlőpanelig, és azon belül is egészen a processzorig. A megjavítása csak vezérlő cserével lett volna megoldható, ami nagyon költséges, viszont a gép mechanikai szerkezete még jó állapotban volt.

Ez által jött az ötlet, hogy mi lenne, ha építenénk egy saját fejlesztésű vezérlőt, ami kiküszöböli azokat a hibákat, amelyek gyárilag előfordulhatnak. Ez a vezérlő és a hozzá tartozó perifériák láthatóak az 1. ábrán.



1. ábra. A vezérlő és a perifériák

2. Probléma megoldásának menete

2.1. Kapcsolási rajz

A kapcsolási rajzot saját elképzeléseink alapján állítottuk össze. Figyelembe vettük a gyári áramkör hibalehetőségeit és igyekeztünk a problémák kiküszöbölését megoldani. Olyan alkatrészeket válogattuk össze, amelyek a piacon mindennap megvásárolhatók. Ügyeltünk arra, hogy az alkatrészek ára ne emelje túlságosan a berendezés értékét.

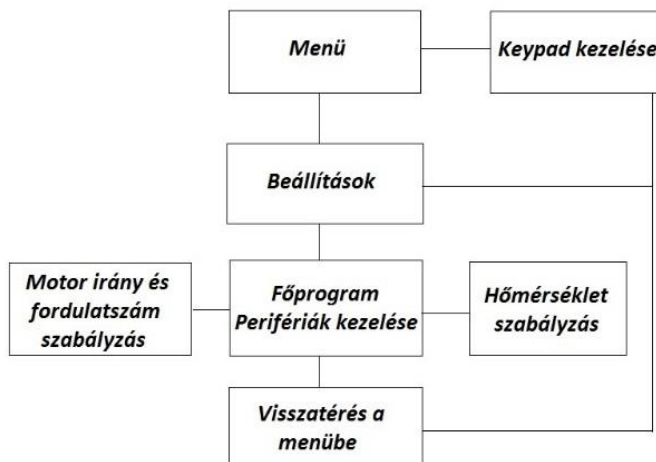
2.2. Áramkör leírása

A vezérlő alapját egy AVR ATmega 128-as mikrokontroller alkotja, e köré vannak építve azok a perifériák, amelyek a különböző egységek működését biztosítják. A tervezés során igyekeztünk olyan szolgáltatásokat lehetővé tenni, melyet egy átlagos gyári mosógép nem tud, a felhasználó szempontjából szélesebb lehetőségeket biztosítottunk a beállítások megvalósításához. A gép működését egy 4x20-as LCD kijelzőn keresztül kontrollálhatjuk, a paramétereket pedig egy numerikus billentyűzet segítségével tudjuk bevinni a vezérlőbe. Az összes periféria galvanikusan le van választva a mikrokontrollerről, ezáltal nem fordulhat elő az a helyzet, ami az eredeti hibát előidézte. A galvanikus leválasztást szilárdtestrelék alkalmazásával valósítottuk meg. A motor fordulatszámát egy triakos vezérlő szabályozza, melyet egy 8 bites digital-analog átalakító paraméterez. A motorról visszacsatolást egy analog-digital átalakítón keresztül csatlakoztatjuk. A vízszint magasságának visszajelzését egy nyomáskapcsolón keresztül digitális bemenetre csatlakoztattuk. A vízhőfok érzékelését egy DS1821 digitális hőmérő IC-vel oldottuk meg. A vezérlő belső idejének paraméterezését egy DS1307 digitális óra IC biztosítja. A gép paraméterezésének beállításakor a felhasználó teljes szabadkezet kap, tetszőleges hőfokokat, időket, öblítésszámot, fordulatszámot állíthat be. Ezzel a felhasználói szabadsággal energiát, időt spórolhat, és a mosás intenzitását növelheti.

2.3. Szoftver leírás

A mosógép szoftvere menüvezérelt, ezáltal könnyen kezelhető bárki számára. A numerikus billentyűzetet mátrixkapcsolásban működtetjük. A hőmérő IC-t one wire üzemmódban, az óra IC-t I²C bus-os kommunikációval oldottuk meg. Az LCD kijelzőt 4 biten működtetjük. A gép bekapcsolásakor a

kijelzőn megjelenő négy soros menüből választhatunk, ezek a beállítás, az időzítés, a centrifuga üzemmód és az S.O.S. üzemmód. A centrifuga üzemmódban egy átlagos centrifugálási programot futtatunk le. Az időzítés üzemmódban be lehet állítani bármely mosás kezdeti idejét. S.O.S. üzemmódban közvetlenül a szivattyút tudjuk működtetni. Ezt azért hoztuk létre, hogy a gép bármilyen meghibásodása esetén a vizet könnyedén el tudjuk távolítani a tartályból. A beállítás menüben a mosási idő, víz hőfok, öblítésszám és centrifuga sebesség beállítására van lehetőség. A centrifuga sebességet 0-tól 5-ig értékben határoztuk meg. A numerikus billentyűzet mátrixos működése egy szubrutinban van összefoglalva, amit az aktuális ciklus mindig meghív. Ezekből a kombinációkból hajtja végre a program a következő lépéseket és az értékeket regiszterekben tárolja. A beállítás során az összes paraméterező regiszter feltelik egy számmal és a start elindítása után a főprogram ezen regiszterek számértékeit használja fel a program futtatása során. A program a regiszterek értékeket korlátozza, hogy ne lehessen irreális mennyiséget beírni. A főprogram aktuális állapota állandóan kijelvezve van az LCD-n, mely pontos információkat nyújt a felhasználó felé. A digital-analog átalakító értékei fix értékkel indul és ehhez meg van határozva az analog-digital konverter értéke. Ha ez az érték +- 5 tűrésen kívül esik, akkor ennek megfelelően változtatja a digital-analog értékét. A szoftver blokkvázlata a 3-as ábrán látható.



4. ábra. A szoftver blokkvázlata

3. Elért eredmények

A mosógép a valóságban is megépült, jelen pillanatban napi használatban van. A bemutatására megépítettük a vezérlő másolatát, ami csak a gyakorlati paraméterekben különbözik a működő berendezéstől. Igaz, hogy az alapgondolat csak egy elromlott mosógép megjavításából indult ki, de az elkészítés során jelentkeztek azok a lehetőségek, hogy ezen a téren mekkora fejlesztéseket lehet végrehajtani. A szoftver folyamatos fejlesztés alatt áll melyben a szolgáltatások és a működési biztonság újabb és újabb ötletekkel bővül. Jelen pillanatban is dolgozunk olyan megoldásokon, melyek segítik a felhasználó számára a szolgáltatások kiterjesztését.

Rajintelligencia viselkedés megvalósítása robotokkal

MPD

Coulibaly Patrik, Csontos Dávid, Kiss Máté

Felkészítő tanár: Kiss Róbert

Kecskeméti Bányai Júlia Gimnázium, 6000 Kecskemét, Nyíri u. 11.

1. Bevezető

Az utóbbi évtizedekben az informatika fejlődésével lehetővé vált olyan gyakorlatilag is kivitelezett kutatások lebonyolítása, amely a természetben szabadon élő állatok viselkedésmintáit tanulmányozva robotok segítségével próbálják azt modellezni. A mesterséges intelligencia kutatások közé sorolt „raj intelligencia” is ide tartozik. Olyan kommunikációra épülő decentralizált rendszerekről van szó, amelyek az egyedek viselkedésének összességéként egy rendszerszintű globális viselkedéshez vezetnek (swarm intelligence). Ilyen lehet például a hangyakolóniák viselkedése.

A modellezéshez olyan „robot raj-ok létrehozása szükséges, amelyek egymástól független egyedei önálló viselkedésmintát követnek, szenzoraikkal érzékelik környezetük fizika, kémiai jellemzőit, és kommunikációs kölcsönhatásaikból egy globális viselkedés alakul ki.

A vizsgálatok céljaként felmerül a nehezen, veszélyt hordozóan megközelíthető terep feltérképezése (pl.: elaknásított terület, Mars, katasztrófa sújtotta övezet, tengeri olajfoltok).

2. A projektfeladat és a megvalósítás

2.1. Biológiai háttér¹

A hangyák (Formicidae) családja a társaséletű hártáyszárnyúak rendjébe tartozik. A kutatási projekt szempontjából a táplálék felkutatására használt módszereiket használtuk és fordítottuk le a robotok nyelvére.

Az általunk szimulált sivatagi hangya (*Cataglyphis fortis*) viselkedése eltér a feromon nyomokat hagyó rendtársaitól. A sivatagban a tájékozódásra nem áll rendelkezésére semmilyen tereptárgy, illetve szagnyomokat sem tud hagyni, mert a hatalmas hőségben a feromon nyomok hamar

¹ Kathrin Steck, Matthias Wittlinger, Harald Wolf: Estimation of homing distance in desert ants, *Cataglyphis fortis*, remains unaffected by disturbance of walking behaviour, *J Exp Biol* 2009 212:2893-2901. doi:10.1242/jeb.030403

Lixiang Li, Haipeng Peng, Jürgen Kurths, Yixian Yang, Hans Joachim Schellnhuber: Chaos-order transition in foraging behavior of ants, *PNAS* 2014 111 (23) 8392-8397; published ahead of print May 27, 2014, doi: 10.1073/pnas.1407083111

elpárolognának. Ez a hangyafaj az irányt a nap állásából határozza meg, míg a bolytól megtett távolságot lépéseiket számlálva.

A projektben a sivatagi hangyák biológiai viselkedésén keresztül modelleztünk egy terep-felderítési feladatot.

2.2. Szakirodalomban talált megoldások hasonló feladatok esetén²

Az eddigi kutatások a szakirodalom feldolgozása alapján több szempont szerint csoportosíthatók. A robotrajok célviselkedése kiterjedhet térképezésre, célobjektum keresésére, gyülekezésre. A tájékozódáshoz használt, környezet jeleit figyelő szenzorok köre igen széles. A rajok egyedeinek kommunikációja, a gyűjtött információk megosztása többféle csatornán is megvalósulhat (pl.: bluetooth, wifi).

2.3. A megvalósított rajintelligencia kutatás koncepciója

A pályázat anyagi háttérét és a megvalósíthatóság középiskolai tényezőit figyelembe véve a tervezett projekt céljai és a megvalósítás terve a következő volt:

- A projektben a sivatagi hangyák viselkedésmintáját modelleztük a táplálékkeresés során.
- Kétféle robot készült el a projektben, ezzel is modellezve a hangyák biológiai specifikációját.
- A kutató robot feladata, hogy megtalálja a táplálékot, amit egy piros színű felület szimbolizál, majd visszatérve a bolyhoz az ott várakozó társait a táplálékhoz vezesse.
- A munkás robotok feladata, hogy a kutató robotot követve a táplálékforrásig jussanak.
- Informatikai szempontból a kutató robotok felépítése a hardver és a szoftver szempontjából egységes csakúgy, mint a munkás robotoké, de a kétféle típusú robot esetén különböző.

A kutató robotok egységes, de önálló keresési algoritmus alapján mozognak egy 8-10 m² területű kijelölt terepen és színérzékelő szenzorok segítségével egy felület színétől eltérő mintázatot keresnek. A robotok

² Harald Wolf: Odometry and insect navigation | Exp Biol 2011 214:1629-1641. doi:10.1242/jeb.038570

Attila Pásztor: GATHERING SIMULATION OF REAL ROBOT SWARM, ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)

Pásztor Attila - Kovács Tamás: Statikus bluetooth kommunikációs láncon alapuló, multi-robot területfelfedező algoritmus, A GAMF Közleményei, Kecskemét

G.Beni, J. Wang, Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, 1989.

MINDSTORMS EV3 típusúak, rendelkeznek színszenzorral, giroszkóppal és infra jelvevővel. Három darab kutató robot készült el.

Amennyiben valamelyik kutató robot megtalálta a célterületet, visszatér a munkás robotok várakozási területéhez (boly), majd a munkás robotokat a célterülethez vezeti.

A munkás robotok MINDSTORMS NXT típusúak és rendelkeznek ultrahang szenzorral, valamint iránytű szenzorral. A raj követése a mágneses mező változására épül, amely a kutató robotra szerelt állandó mágnes mozgásának változására épül. Az állandó mágnes rövid távon belül elfedi a Föld mágneses mezejét, így a munkás robotok iránytű szenzoruk segítségével képesek a változások érzékelésére és a követésre, akár láncolt alakzatban is.

2.4. Tájékozódás a terepen

A kétféle típusú robot (EV3 és NXT) egyaránt képes bluetooth alapú kommunikációra, de a két kommunikációs protokoll különbözik egymástól, ezért az egymás közötti adatcsere csak egy bridge beiktatásával lehetséges (megvalósítható pl. mobiltelefonon keresztül). A kommunikáció összetettsége miatt nehezen hozható párhuzamba a biológiai mintával, így a kommunikációmentes megoldás mellett döntöttünk, amelyre nem találtunk megvalósítási példát a feldolgozott szakirodalomban.

A kutató robotok egymástól függetlenül bejárják a területet. A célt elérő robot visszatér a bolyhoz, míg a sikertelenül keresők a kiinduló pozíciójukba térnek vissza.

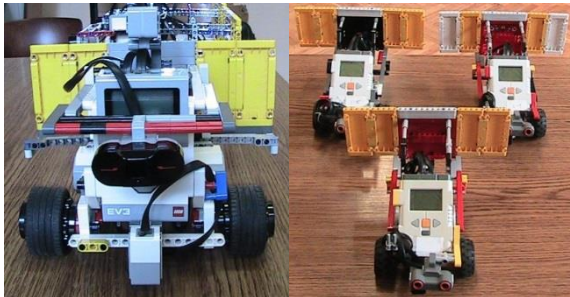
A tájékozódáshoz az ortogonális bejárás technikáját választottuk. A szakirodalom szerint az egyik legegyszerűbb módszer. A mozgás jellegéből adódóan az x illetve y koordináták meghatározása természetes egyszerűséggel adódik. A koordináta-rendszer origója a bolyon van. A koordináták számlálására a szervó motor elfordulás számlálóját használtuk, amely 360 fokos elfordulásonként 1 egységgel növekszik (ez a kutatások szerint párhuzamba állítható az sivatagi hangya tájékozódásával - lépésszámlálás).

A pontosabb pozícióba álláshoz infra kapukat használtunk, amely a biológiai rendszer napfény utáni tájékozódásának felel meg. Az infra jelvevők képesek a jeladó távolságát és irányát meghatározni. A pontos fordulási szögek meghatározásához giroszkóppal szereltük fel a kutató robotokat.

A munkás robotok algoritmusuk csupán a kutató követésére szolgál és mágneses jelérzékelésen alapul. A biológiai párhuzam itt is megvan a

sivatagi hangyák rövid távon belüli érzékelés és követés mintázatával (bár ott nem mágneses a jel).

2.5. Ábrák



1. ábra. Kutató robot

2. ábra. Munkás robotok



3. ábra. Teljes robotraj

3. Hasznosíthatóság

A mesterséges intelligenciakutatás csakúgy, mint a robotika a XXI. század kiemelt műszaki, informatikai területe. A komplex műszaki, természettudományi kutatás összekapcsolja biológiai élőlények viselkedésében rejlő mintákat a gépi, programozott viselkedéssel, ezáltal a két nagyon távoli terület között teremt kapcsolatot. Az általunk megvalósított kommunikációmentes megoldás egyaránt alkalmas terület felderítésre, vagy katasztrófa elhárításra olyan területen, ahol az egyedek információcseréje gátolt vagy veszélyes.

A megvalósított projektben az egyedi a mágneses jelváltozásra épülő követési mechanizmus, valamint a gyakorlatilag kommunikációmentes megvalósítás (csak indirekt kommunikációt használtunk).

Az elkészült robotkonstrukciók, a kidolgozott szenzorhasználati módszerek, valamint a robotraj egyedeinek viselkedését vezérlő algoritmusok saját munkánk.

Időjárás előrejelzés alapú intelligens fűtésszabályozás

SMARTduino

Debnárik Roland, Klein Armand

Tatabányai SZC Szent Imre Középiskolája, 2500 Esztergom, Főapát u. 1.

1. Bevezetés

Korunk egyik legnagyobb kihívása a globális felmelegedés. Úgy tűnik, hogy a világot felfűtöttük a kiáramló széndioxiddal, amely üvegházhatást okozott. A szakemberek szerint az energiahatékonyság fejlesztése lehet az egyik út, ami megálljt parancsolhat a felmelegedésnek. Az energia leghatékonyabb felhasználása az, ha el sem használjuk. Az épületek passzív hőszigetelésén túl is van még megoldás az energiateljesítmény csökkentésére - ráadásul kényelmesen -, ha kihasználjuk az IT rendszerek gyors döntéshozási képességét és felvértezzük azokat a megfelelő döntési algoritmussal.

A projekttel a célunk, hogy egy fűtésrendszer működését befolyásoljuk a leghatékonyabb módon. Olyan vezérlő-szabályozó fejlesztése a cél, mely a mért és a rövidtávon várható időjárás figyelembe vételével még hatékonyabbá teheti egy építmény hőszabályozását. Az aktuális és várható hőmérséklet, a napsütéses órák mért és előre jelzett száma és ideje olyan adatok, amelyekből jól lehet következtetni egy néhány órán belül várható átlagos hőmérsékletre. Ezeknek az adatoknak a felhasználásával elkerülhető, hogy az aktív eszközök feleslegesen fogyasszanak energiát nem sokkal azelőtt, hogy a természet megoldaná a céljainknak megfelelő állapot kialakulását. A rövidtávon várható időjárás adatok ma már ingyenesen hozzáférhetők különböző webes szolgáltatók segítségével.

2. A probléma megoldásának menete

A bevezetésben felvázolt probléma legegyszerűbben így demonstrálható:

- Ne kapcsoljon be egy vezérelhető fűtőtestet, ha látja előre a közeljövőben történő felmelegedést. Ennek legjobb felhasználási területe egy közintézmény például iskola vagy munkahelyi iroda lehet.
- Ha reggel 8 órától kell melegen lennie egy intézményben, akkor hajnalban ne használjunk el energiát fűtésre, amikor tudjuk, hogy reggel 6 órától napsütés várható a déli oldalon, ami rengeteg energiát sugároz majd be az épületbe.

2.1. Megvalósításhoz szükséges hardver eszközök (az 1. ábrán láthatóak):

- Arduino Mega – 54 db digitális ki/bemenet, 16 analóg bemenet, 16 Mhz-es processzor, 128 kbyte flashmemória, 8 kbyte SRAM) – Ez a „micro controller” vezérli a létrehozott algoritmus alapján a fűtőtestet
- Ethernet shield – a WEB-es kezelőfelület szervere és egyben időjárás adatokat letöltő kliens
- Hő érzékelők (Arduino kompatibilis DHT11 v. 22)
- Fénymérő (Arduino kompatibilis)
- SD memóriakártya
- LCD kijelző – 16*2 karakter, IIC csatlakozási felület Arduinohoz
- Vezérlő gombok – Arduinohoz
- Relé (Arduino által kapcsolható) – 5 V bemenet alapján 230V-os eszközt kapcsol (fűtőtest)
- Fűtőtest
- A felsorolt eszközök fogyasztása elhanyagolható.

2.2. Megvalósításához szükséges adatok:

- Időjárás előrejelzés adatai – ország és város paraméterekkel helyi adatokat kapunk online

<http://api.openweathermap.org/data/2.5/forecast?q=Hungary,esztergom&mode=xml>

- Mért hőmérsékleti adatok, kültéri és beltéri
- Mért fényerősség – ebből következtetünk az ablak mérete és tájolása alapján a besugárzott energiamennyiségre - -> **Sugárzási energiaáram sűrűség**
- Fűtőtest teljesítménye
- Előrejelzés felhősödésről –Felhős időben nem számít a nagy fényerő, energiát nem kapunk. Honlapról kapott információ:”clear sky”, „Partly sunny”, stb.

2.3. A megvalósítás során alkalmazott szoftveres és hardveres feladatok, nehézségek:

2.3.1. Felhasznált programnyelvek, függvénykönyvtárak

A projekthez szükséges C nyelvhez hasonló Arduino fejlesztői környezethez további függvénykönyvtárakat kellett hozzárendelni: SPI.h; Ethernet.h; Wire.h; LiquidCrystal_I2C.h; DHT.h; SD.h. A WEB alapú irányítófelület kialakítása az AJAX scriptnyelvet igényelte.

2.3.2. Nehézségek

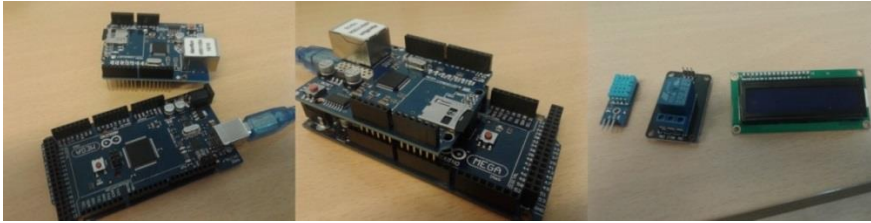
Legnagyobb nehézséget a döntést hozó algoritmus megalkotása okozta, mivel számításba kellett venni a nap fényerejét, a déli oldalon lévő ablakok felületét, az aktuális hőmérsékletet (kint és bent) illetve a várható hőmérsékletet. Ezekből számított várható energia besugárzás mértékét kellett összevetni a benti energia igényvel, ami alapján a döntést meghozhatta a mikroszámítógép.

További nehézséget okozott a szabatos fizikai fogalmak megtalálása, mint például a „Sugárzási energiaáram sűrűség”. Nem volt egyszerű az előrejelzési információkat megtalálni, de sikerrel jártunk az openweathermap.org segítségével, ahol XML formátumban valós időben rendelkezésre bocsájtják a helyi (pl.:esztergomi) időjárásállomások adatait ingyen. A több Arduinohoz kapcsolt szenzor bonyolult vezetékezési és programozási feladatot rótt ránk, mivel mindegyik függvénykönyvtárát meg kellett ismerni. A WEB-es irányítófelület kialakítása is egy újabb programozási nyelv elsajátítását igényelte.

2.3.3. Megvalósítás

A hardver és szoftverkövetelmények számba vétele után, a fűtésrendszer kapcsolásával összefüggő emberi igények algoritmizálása következett. Az Arduino C nyelvhez hasonló fejlesztőkörnyezetében le kellett kódolni az igényeket. Ezek után az LCD paneles vezérlés megvalósítása következett, aminek analógiájára egy WEB-es – ergonomikusabb - felület is elkészült, ami a távolról – online - történő irányítást is lehetővé teszi. Miután a szoftver és hardverszerkezet kialakult, már csak egy relé kapcsolása volt szükséges ahhoz, hogy a fűtőtest hőt bocsásson ki, vagy várjon az időjárás támogatására.

2.4. Ábrák



1. ábra.

Arduino Mega és ethernet shield; összeszerelt eszközök; hőmérő, relé, lcd kijelző

3. Elért eredmények

A megvalósítás során sikerült egy megalapozott döntést hozó szerkezetet megalkotni, amely képes az online időjárési adatbázisból beavatkozás nélkül előrejelzési információt szerezni, amit felhasznál a fűtésrendszer szabályozására. A kötött idejű hő felhasználás során jelentős eredményeket lehet elérni az energiahatékonyság terén, hiszen tényleg nem kapcsol be a fűtés a hajnali órákban, ha a következő időszakban napsütés várható. A reggel 8 órai iskolakezdéskor elegendő mennyiségű ingyen hőenergiával van ellátva az épület. Ezzel naponta több kilowattnyi energiát lehet megspórolni akár egy családi ház méretű ingatlan esetében is. A borús és napos időszakok váltakozása is ugyanilyen helyzetet teremt, ahol igazán hatékony az algoritmus. A megtakarítás az intézmény méretével arányosan nő. Amennyiben egy ilyen rendszer elterjed, az óriási energia megtakarítást jelenthet az egész világ számára.

Informatika szekció

Hamal

A jövő reménye

Kiss Benedek, Huszka Adrián, Nagy Roland

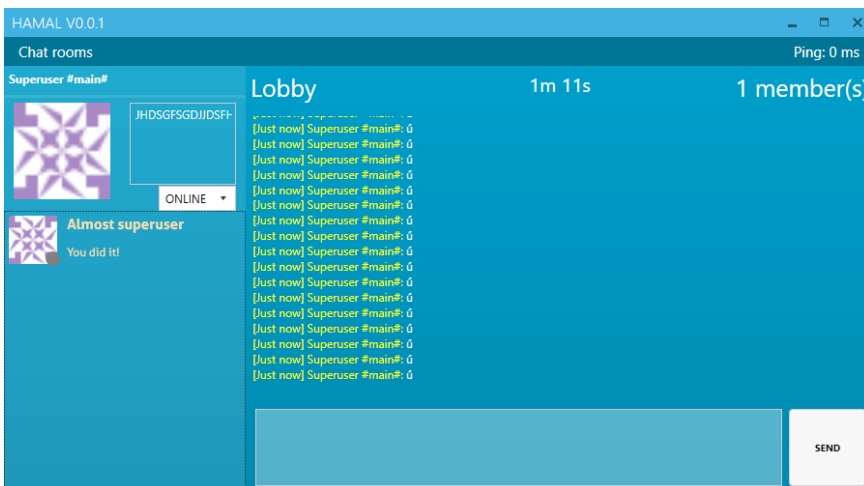
Felkészítő tanár: Vári Ágnes

*Hódmezővásárhelyi SZC Pollák Antal Szakképző Iskolája, 6600 Szentés
Apponyi tér 1.*

1. Bevezetés

A Hamal egy valós időben történő kommunikációs alkalmazás, amelyben képesek vagyunk barátainkkal, ismerőseinkkel beszélgetni, sőt csoportot is létre tudunk hozni, ahol képesek vagyunk többen kommunikálni. A Hamal beszerzése után nincs más dolgunk, mint regisztrálni az erre fejlesztett oldalon, ahol egy felhasználónevet, egy email címet, egy valódi nevet és egy jelszót kell megadnunk. Miután ezt megtettük, nincs más dolgunk, mint belépni és élvezni a Hamal által nyújtott lehetőségeket. A program Microsoft Visual Studio 2013-ban íródott, C# nyelven. A forráskód teljes mértékben objektumorientált. A megadott adatok egy tömörített bináris ini fájlban mentődnek el (melynek kiterjesztése hdf = **H**amal**D**ata**F**ile), szerveroldalon. A bejelentkezés során, képesek vagyunk menteni az adatainkat, hogy később ne keljen azzal foglalkozni, hogy beírni a felhasználónevet és jelszót. Be tudjuk állítani, hogy automatikusan léptessen be minket a program. Miután beléptünk, azonnal a chat ablak kerül a szemünk elé.

1. ábra. A bejelentkezési felület



2. ábra. A főoldal

2. Problémák

A programozás során sokszor előfordult az, hogy rossz csomag lett elküldve a szervernek, ami azt eredményezte, hogy összeomlott a program. A másik gyakori hiba a UUID megjelenítésével volt.

```

if (!File.Exists("Users\\0.hdf"))
{
    DataFile t_File = new DataFile("Users\\0.hdf");
    t_File.AddEntry("USER_DATA", "DisplayName", Encoding.UTF8.GetBytes("Superuser #main#"));
    t_File.AddEntry("USER_DATA", "UUID", new UUID().ToBytes());
    t_File.AddEntry("USER_DATA", "Status", new Byte[] { (Byte)UserStatus.OFFLINE });
    t_File.AddEntry("USER_DATA", "StatusMessage", Encoding.UTF8.GetBytes("Just do it!"));
    t_File.AddEntry("USER_DATA", "DatabaseID", BitConverter.GetBytes((Int64)0));
    t_File.AddEntry("USER_DATA", "ProfilePicture", File.ReadAllBytes("D:\\programming\\5fd80d98f96f9adbde7876cbda181715.png"));
    t_File.AddEntry("USER_DATA", "Contacts", Helper.Int64Array(new Int64[] { 1 }));
    t_File.AddEntry("USER_LOGIN_DATA", "Username", Encoding.UTF8.GetBytes("admin"));
    t_File.AddEntry("USER_LOGIN_DATA", "Password", Encoding.UTF8.GetBytes("1234"));
    t_File.Save();
}

if (!File.Exists("Users\\1.hdf"))
{
    DataFile t_File = new DataFile("Users\\1.hdf");
    t_File.AddEntry("USER_DATA", "DisplayName", Encoding.UTF8.GetBytes("Almost superuser"));
    t_File.AddEntry("USER_DATA", "UUID", new UUID().ToBytes());
    t_File.AddEntry("USER_DATA", "Status", new Byte[] { (Byte)UserStatus.OFFLINE });
    t_File.AddEntry("USER_DATA", "StatusMessage", Encoding.UTF8.GetBytes("You did it!"));
    t_File.AddEntry("USER_DATA", "DatabaseID", BitConverter.GetBytes((Int64)1));
    t_File.AddEntry("USER_DATA", "ProfilePicture", File.ReadAllBytes("D:\\programming\\5fd80d98f96f9adbde7876cbda181715.png"));
    t_File.AddEntry("USER_DATA", "Contacts", Helper.Int64Array(new Int64[] { 0 }));
    t_File.AddEntry("USER_LOGIN_DATA", "Username", Encoding.UTF8.GetBytes("admin2"));
    t_File.AddEntry("USER_LOGIN_DATA", "Password", Encoding.UTF8.GetBytes("1234"));
    t_File.Save();
}

```

3. ábra. A teszt admin felhasználók létrehozása

2.1. Szerver rész

A szerver minden egyes művelet megjelenít egy konzolban, amelynek a segítségével hibára tehetünk szert és nyomon követhetjük a felhasználók tevékenységeit. A szerver hozza létre a bináris ini fájlokat, amelyben a felhasználók és a szobák adatait tároljuk.

2.2. Kliens rész

A kliens rész a műveletekért felelős. Ha például küldünk egy üzenet, a kliens küld egy parancsot a JSON Packet-nek. A parancs mellett a szükséges adatok is elküldésre kerülnek (ilyen a felhasználó azonosítója, vagy a szoba azonosítója). Miután elküldtük a parancsot és a szükséges adato(ka)t, a JSON végrehajtja azt a függvényt, amelyre a parancs hivatkozik.

2.3. Helper fájl

A szerver is és a kliens is rendelkezik ezzel az úgynevezett helper fájjal. A helper fájl az alapvető metódusokat tartalmazza, mint például a karakterkonvertálások, a dátum megformázása, alap fénykép betöltése (későbbiekben lesz lehetőség fényképet feltölteni)

```
public static String BeautyDate(DateTime a_DateTime)
{
    DateTime t_Now = DateTime.Now;

    if (t_Now.Day - 1 == a_DateTime.Day)
        return "Yesterday at " + a_DateTime.ToShortTimeString();

    if (t_Now.Day == a_DateTime.Day)
    {
        if (t_Now.Hour == a_DateTime.Hour)
        {
            if (t_Now.Minute == a_DateTime.Minute)
                return "Just now";

            TimeSpan t_Diff = t_Now - a_DateTime;

            return t_Diff.Minutes.ToString() + (t_Diff.Minutes == 1 ? " minute ago" : " minutes ago");
        }

        return "Today at " + a_DateTime.ToShortTimeString();
    }

    if (t_Now.Day + 1 == a_DateTime.Day)
        return "Tomorrow at " + a_DateTime.ToShortTimeString();

    return String.Format("{0}-{1}-{2}. {3}", a_DateTime.Year, a_DateTime.Month, a_DateTime.Day, a_DateTime.ToShortTimeString());
}
```

4. ábra. A dátum formázása

3. Elért eredmények

A program aránylag stabilan működik, stabilabban, mint a Skype. Osztályszinten működik a program és nagy lelkesedéssel használják a tanulók, mint bármi más ami IRC alapon működik, a program JSON sztringekkel kommunikál a klienssel. A fejlesztés közben a Bitbucket nevezetű oldalon tudtuk egymással megosztani a változtatásainkat és ötleteinket. A fejlesztés során többször is kikértük tanáraink véleményét, amely azt eredményezte, hogy hatékonyabb és letisztultabb forráskódot tudtunk írni.

MSG

AkosSoft

Juhász Ákos

Erkel Ferenc Gimnázium, Gyula, Bodoky u. 10.

1. Bevezetés

A programom címe: MSG. A cím is arra utal, hogy a programom fő témája a kommunikáció, ugyanis az angol "message" kifejezésből ered(1.ábra). A szoftver egy általam felépített szerverből, illetve egy kliens felületről áll. Az alapja a következő: a felhasználók belépés után felcsatlakoznak a szerverre és kommunikációt folytatnak.

2. Program felépítése, problémák feldolgozása

A programom első állomása egy saját készítésű weboldal (2. ábra), ahol a felhasználókat regisztráció, illetve belépés után számos funkció várja. Többek között felhasználónév, illetve jelszó megváltoztatása, valamint üzenhetnek az adminnak az oldalon keresztül és fel is iratkozhatnak a legújabb hírekre és frissítésekre. A regisztráció is nagyon fontos szerepe az oldalnak, mert a felhasználók később ezek alapján lesznek azonosítva. Azonban az oldal fő célja a készített szoftver elérhetővé tétele, ugyanis a felhasználók itt tudják letölteni azt.

A program annyiban különbözik más chat programoktól, hogy kettő verziója van. Az egyik verzió a tömegkommunikációról szól. Ez a típus is a letölthető a weboldalon. Itt is érvényes az, hogy a honlapon megadott adatok használatával kell bejelentkezni a programban (3. ábra). A program ezen változata saját szervert működtet. Kapcsolódás után a szerver státusza folyamatos jelleggel listázza ki a bejelentkezett felhasználókat. Azt értem tömegkommunikáció alatt, hogy a felhasználók felcsatlakozás után a kliens felületen, egymással tudnak beszélgetni, de mindenki mindenkivel. Például, ha én belépek és egy másik barátom, illetve egy harmadik barátom is, akkor nem tudjuk megválogatni, hogy ki kivel akar beszélgetni, hanem csak egymással tudunk közösen, nyíltan. Tulajdonképpen ez egy chat szoba. Ez a típus könnyen felhasználható egy családban vagy egy baráti körben.

A program másik verziója szintén elérhető a weboldalon, illetve szintén saját szervert használ. Működési elve annyiban egyezik meg az előző verzióval, hogy letöltés után a honlapon megadott adatokkal kell bejelentkezni. Belépés után több funkciót használhatnak a felhasználók. A

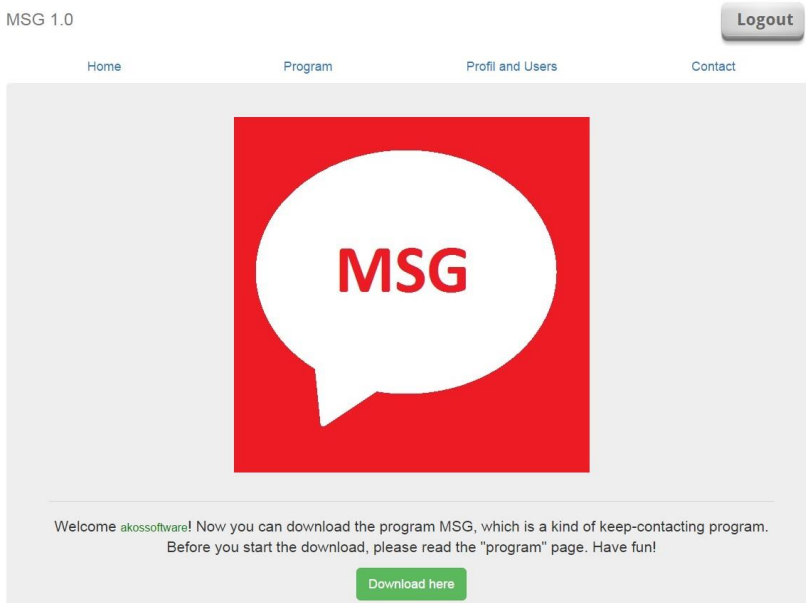
programban partnerkérelmet küldhetnek az adott felhasználó neve segítségével. A weboldalon egy lista található, ahol az összes felhasználó megtalálható és betű alapján is lehet keresni köztük. A partnerkérelem elküldése után bármikor számíthatnak szerver által küldött üzenetre, ami értesít, hogy az adott felhasználó elfogadta-e a felkérést(4. ábra). Ha elfogadta, akkor az ő neve beépül a partnerlistába. Viszont a partnerlistában csak azok a felhasználók jelennek meg, amelyek online állapotban vannak, azaz be vannak jelentkezve. Innentől kezdve beszélgetést lehet kezdeményezni az online tagokkal, illetve törölni is lehet őket. Valamint jelteni is (reportolni) a szervernek adott ok szerint, amit majd az admin lekezel(5. ábra).

A szerver úgy van felépítve, hogy az adatok rendszeresen frissítésre, illetve mentésre kerülnek. A weboldal MySQL adatbázist használ, a program pedig Java nyelven készült PHP API-k segítségével.

2.1. Ábrák



1. ábra. A program logója.



2. ábra. A weboldal kezdőlapja.



3. ábra. A programba való bejelentkezés.



4. ábra. Partnerkérelemről való értesítés.



5. ábra. Főmenü.

CuStudy

BetonSoft

Guzsik Dávid József, Kiss Antal, Mészáros Bálint

Felkészítő tanár: Fekete Balázs

*Debreceni SZC Mechwart András Gépipari és Informatikai Szakközépiskolája,
4025 Debrecen, Széchenyi u. 58.*

1. Bevezetés

A XXI. század diákjainak figyelme nagyon megosztott az iskolai elfoglaltságok (házi feladatok elkészítése, tanulás, különórák) és az iskolán kívüli, szabadidős elfoglaltságok (barátok, pihenés, szórakozás) között. A tanulók érdeke, hogy minél több ismeretet és minél nagyobb szakmai tudást sajátítsanak el. A tanórákra való felkészülés, a házi feladatok elkészítése nagyon fontos. A figyelem megosztottsága miatt a diákok gyakran szétszórtak, feledékenyek, nem tudnak képességeikhez illő eredményeket elérni. Minden iskolás ismeri az órarendekkel járó bosszúságokat, véletlenül A hét helyett B hétre pakol a diák. Előfordul, hogy eltéveszti, hogy milyen nap következik (például a csütörtöki nap helyett a pénteki órákra készül). Bármely tévesztés miatt megeshet, hogy a házi feladat nem készül el, más tanítási órák felszerelését hozza magával.

2. Probléma megoldásának menete

Web applikációkkal szeretnénk egy mindenki által jól hozzáférhető felületet biztosítani, ahol a bevezetésben említett problémákra megoldást kínálunk, és ezt a segítséget a diákok bármikor, bármilyen eszközön elérhetik, bárhol is legyenek éppen.

2.1. Az alkalmazás ötletadója, elődje

A program elődjét a csapatunk egyik tagja által készített BetonHomeWork szoftver adja, aminek létrejöttét az inspirálta, hogy tudomásunk szerint hasonló szoftver nem volt megtalálható a piacon. Ez a program asztali számítógépeken futott, mint natív szoftver, képes volt a házi feladatokat, és korlátozott mértékben az órarendet is megjeleníteni. A programot az osztályunk két harmada (mintegy 24 tanuló) használta, és köztük nagy sikernek örvendett. A program, bár népszerű volt, közel sem volt teljes: csak a Microsoft Windows operációs rendszert futtató számítógépeken futott,

fejben tartani. Ezenfelül ha az adott szolgáltatásban már be van jelentkezve, akkor a belépés csak egyetlen gombnyomás.

A házi feladatok nyomon követésében segít az oldal: az elkészített házi feladatokat „kész” jelzéssel lehet ellátni, mely elrejtí az adott feladatot a főoldalról, de ezzel a jelzéssel a házi feladatok listájában továbbra is elérhető lesz, ha utólag szeretne a tanuló valamilyen, a feladattal kapcsolatos információt leellenőrizni.

Az iskolai élet nem csak a tanulásról szól: a legtöbb iskolában időnként rendezvények és egyéb fontos, akár az egész osztályt érintő események, tanulmányi versenyek is lehetnek, melyeknek pontos idejét, dátumát és helyét nem könnyű észben tartani, és még emellett nem elfelejteni időben megjelenni. A szoftverünk erre a problémára is megoldást nyújt.

A felhasználóink adatainak biztonságára is ügyeltünk, így például kötelezővé tettük a HTTPS kapcsolatot az egész weboldalon, emellett a POST kéréseket is védjük a CSRF támadások ellen.

The screenshot shows the CuStudy application interface. On the left is a dark sidebar with navigation options: Főoldal, Órarend, Házi feladatok, Események, Profilom, Tanárok, Tantárgyak, Csoportok, Felhasználók, and Kijelentkezés. The main content area is titled 'Jó estét Guzsik Dávid József!' and shows 'Elkészítésre váró házi feladatok' (Pending homework) and 'Következő napi órarend' (Next day's schedule). The homework list includes 'Irodalom' (1.óra) with a note about an Arany János award, 'Angol nyelv' (2.óra) with a note about a topic review, and 'Matematika' (4.óra) with a note about a 13-page assignment. The schedule lists 7 hours: 1.óra: Irodalom, 2.óra: Angol nyelv, 3.óra: Testnevelés, 4.óra: Matematika, 5.óra: Matematika, 6.óra: Osztályfőn., 7.óra: Fizika.

1. ábra. A program főoldala egy adminisztrátor szemszögéből

3. Elért eredmények

A szoftver alapjának fejlesztését 2014 decemberében kezdtük azzal a céllal, hogy egy olyan motort alkossunk, melyet nem csak a CuStudy, hanem a jövőbeli projektjeink bázisaként is használhatunk majd. Nem sokkal a motor fejlesztésének beindulása után a szoftver fejlesztésére tudtunk koncentrálni, majd ez év szeptemberében a szoftvert tesztelés céljából elérhetővé tettük a program elődjét használóknak, akiktől a pozitív visszajelzések mellett

hibajelentéseket és javaslatokat is kaptunk, amelyekből tanulva jobbá tudtuk tenni szoftverünket. Fejlesztési munkák eredményeképpen több mint 6000 soros kód bázis jött létre, melyet a mai napig gyarapítunk.

A program elkészítése során számos problémába ütköztünk, melyek megoldásával személyileg is fejlődött csapatunk. Mint az emberek általában, mi is tanultunk hibáinkból, és a projekt elkezdése óta szélesebb rálátással rendelkezünk a webes alkalmazások világára. Sok technológiával és sebezhetőséggel megismerkedtünk, melyeket korábban csak hírből ismertünk.

Bízunk benne, hogy a program elkészítésébe fektetett idő nem vész kárba, és sikerül a modern kor eszközeivel egy olyan szolgáltatást biztosítanunk a tanulók számára, amelynek valóban hasznát veszik majd tanulmányaik során.

Wise

Budavári Márk

Tatabányai SZC Szent Imre Középiskolája, 2500 Esztergom, Főapát u. 1.

1. Bevezetés

Már 12 éves korom óta aktívan foglalkozom a programozással és ez idő alatt meglehetősen sok programozási nyelvet sajátítottam el, melynek köszönhetően ismerem az adott programozási nyelvek hiányosságait, illetve a programozás általános problémáit.

Adott nyelvi hiányosság lehet az, hogy például nem tartalmaz olyan függvényeket, amelyeket a programozók gyakran használnak, így ők nekik kell az algoritmust létrehozniuk, amely fontos időkiesést jelenthet. Általános programozási problémák közé tartozik az, hogy egy programozó létrehoz egy adott algoritmust, amit később egy másik projektnél újra használna, meg kell keresni melyik projektben hozta létre, hol is van az a projekt, melyik fájlban van a függvény, és ha ez egy projektben sokszor történik, akkor szinte folyamatos a keresés és a ctrl+c, ctrl+v kombináció.

Ezeket a problémákat oldja meg a Wise, amely gyorsítja, egyszerűsíti, valamint nyílt forráskódú szoftverként egyedivé szabható vele akármilyen programozási nyelv. A Wise ráépül egy adott nyelvre és kiegészíti azt.

2. Probléma megoldásának menete

2.1. Nyelvi hiányosságok

A Wise segítségével minden programozási nyelv személyre szabható és tetszőleges metódusokat lehet hozzárendelni, így gyorsítva és egyszerűsítve a programozást. Ezen kívül a már meglévő függvényeket is átszabhatjuk egyszerűbb formára. A Wise-ban programozás menete: Először létrehozuk a személyre szabott nyelvben a programunkat (1. ábra), ezután megnyitjuk a Wise konzolos szoftvert, amiben kiadjuk a fordítás parancsot (2. ábra) és ha a fordító nem talál hibát, akkor a kimenet mappában már a lefordított program vár minket (3. ábra).

```

<?php
    for 0..9{
        $tomb[$IT] = $IT+1;
    }

    for 0..$tomb.|Length-1{
        echo $tomb[$IT];
    }
?>

```

1. ábra. Személyre szabott, egyszerűsített PHP nyelv

```

Wise> translate
translate | t - átfordítás más nyelvre: translate főFájl kimenetMappa
Wise> translate d:\WisePHP\index.wise d:\WisePHP_out\

```

2. ábra. Wise fordító

```

<?php
    for($IT = 0; $IT <= 9; $IT++){
        $tomb[$IT] = $IT+1;
    }

    for($IT = 0; $IT <= count($tomb)-1; $IT++){
        echo $tomb[$IT];
    }
?>

```

3. ábra: Fordítása kimenete, az eredeti PHP nyelv

2.2. Felhő alapú függvénykönyvtárak

A Wise kiegészíti az adott nyelvet egy függvénykönyvtár felhőszolgáltatással, amelyben a saját és mások által feltöltött függvénykönyvtárakat lehet elérni és behívni, mint osztály, illetve külön is behívni az osztályból egy adott függvényt tetszőleges helyre, de akár külön az adott függvénynek az utasítássorozatait is be lehet hívni szimplán.

Minden Wise programban a kimeneteli nyelvtől független lehetőség van a fejrészben behívni a felhőből függvénykönyvtárakat és osztályként használni (4. ábra). A „Wise” a rendszer előjele, „BudavariMark” a felhasználó, akitől függvénykönyvtárat szeretnénk használni, a „PHP” pedig maga a könyvtár neve. Az „as BM” utótaggal jelölöm, hogy ezt a könyvtárat BM nevű osztályként szeretném használni.

```
[
@ Wise.BudavariMark.PHP as BM;
]
```

4. ábra: Függvénykönyvtár behívása a felhőből és osztályként használása

Ha esetleg osztály nélkül szeretnénk behívni egy adott függvényt a függvénykönyvtárból, akkor ezt is megtehetjük (5. ábra).

<pre><?php \$szam1 = 10; \$szam2 = 15; putWiseFunc Wise.BudavariMark.PHP.osszeadas; \$eredmeny = osszeadas(\$szam1, \$szam2); ?></pre>	<pre><?php \$szam1 = 10; \$szam2 = 15; function osszeadas(\$szam1, \$szam2) { return \$szam1+\$szam2; } \$eredmeny = osszeadas(\$szam1, \$szam2); ?></pre>
--	--

5. ábra: Függvény behívása: bal oldalt Wise, jobb oldalt a kimenet

A szerintem a legjobb használati módja a felhőnek, amikor úgy hívjuk be a függvényt, hogy csak az utasítássorozatot szedjük ki belőle és alakítjuk át (6. ábra).

<pre><?php \$tomb[0] = 5; \$tomb[1] = 10; \$tomb[2] = 15; putWise Wise.BudavariMark.PHP.atlag(\$tomb)->\$eredmeny; ?></pre>	<pre><?php \$tomb[0] = 5; \$tomb[1] = 10; \$tomb[2] = 15; \$sum = 0; for(\$i = 0; \$i < count(\$tomb); \$i++){ \$sum += \$tomb[\$i]; } \$atlag = \$sum/count(\$tomb); \$eredmeny = \$atlag; ?></pre>
---	---

6. ábra: Függvény utasítássorozatának behívása: bal oldalt Wise, jobb oldalt a kimenet

3. Elért eredmények

Aki már dolgozott szoftverfejlesztőként az tudja, hogy a határidőt szinte mindig képtelenség tartani, de a Wise segítségével sokkal gyorsabban lehet befejezni a projekteket. Ezen kívül olyanra tudjuk átszabni a programozási nyelveket amilyenre csak akarjuk, de a munka nem áll meg, mivel még rengeteg lehetőség rejlik a szoftverben, illetve még sok nyelvre nem lett ráírva a Wise, mivel egyedül dolgozom rajta és a fejlesztés csak akkor indult el, amikor hallottam a versenyről. A verseny után githubon is elérhető lesz a projekt, így akárki szabadon továbbfejlesztheti.

Lia - az ételérzékenyek asszisztense

Lia

Kovács Márk, Lator Anett, Benkovics Ernő

Felkészítő tanár: Stéberné Urbán Anna Ilona

Budapest Műszaki Szakképzési Centrum Neumann János Számítástechnikai Szakközépiskolája, 1144 Budapest, Kerepesi út 124.

1. Bevezetés

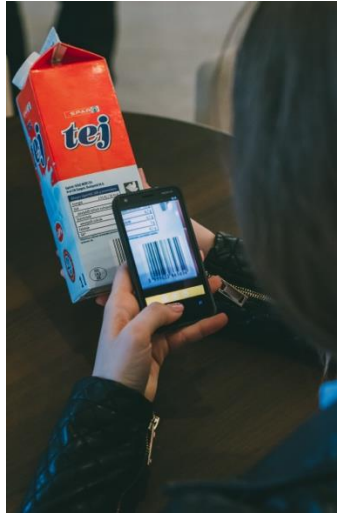
A missziónk, hogy minél több ételérzékeny ember életét megkönnyítsük és egyszerűbbé tegyük a megfelelő étel kiválasztását a mindennapokban. Így létrehoztunk egy multifunkciós alkalmazást, melyben integráltunk egy vonalkód olvasót, amely a termék vonalkódja alapján megmondja a felhasználónak, hogy eheti vagy sem az adott terméket. Ez időt spórol a fogyasztónak, így nem kell elolvasni a termék hátoldalán az összetevőket. Továbbá az alkalmazás felajánl kifejezetten ételérzékenyeknek szóló receptgyűjteményt is, így nem kell órákat böngészni az interneten a megfelelő recept után. Gondolva a gyerekekre is, a közeljövőben az alkalmazás egy játékot is fog tartalmazni, ahol játékos formában megtanulhatják a kicsik, hogy mit ehetnek és mit nem.

2. Probléma megoldásának menete

A vonalkód olvasóval a felhasználó gyermeki könnyedséggel le tudja olvasni a vonalkódot és Lia pillanatok alatt megmondja, hogy eheti-e vagy pedig nem az adott terméket. Ezzel sok vásárlási órát megspórolva, amit mással tud tölteni. A receptgyűjtemény egy helyen egyszerűen elérhető, mivel kifejezetten nekik szánt receptek. A naptárba tudják vezetni az étkezéseiket. A jövőbeli játékkal a gyerekeknek szeretnénk megtanítani, hogy mit ehetnek és mit nem, mind ezt egy játékos keretrendszerrel. Az applikációt minden platformra ki fogjuk adni. Az applikáció 90% már készen van. Gőzerővel dolgozunk, hogy minél hamarabb eljuthasson a felhasználóinkhoz.

2.1. Ábrák

Kevés időt szakítunk arra, hogy minden egyes alkalommal elolvassuk a termékek hátoldalán szereplő leírást. Ebben segítünk Mi! Egyszerű mozdulattal, a telefonoddal leolvasod a termék vonalkódját (1. ábra) és máris tudod, hogy eheted vagy sem az adott terméket.



1. ábra. Vonalkód olvasó

Megspóroljuk neked azokat a hosszú órákat, amit a receptek után való kereséssel töltenél. Összegyűjtöttük egy helyre a kifejezetten ételérzékenyeknek szóló recepteket (2. ábra). Így több szabad időt kapsz, hisz ez egy „asszisztens” feladata, mint amilyen a LIA is.



2. ábra. Receptek

3. Elért eredmények

- **Microsoft Imagine Cup Academy (2015 december-április):** A legjobb 15 csapattal egy intenzív 3 hónapos üzleti felkészítést csináltunk végig angolul.
- **V. Petrik TISZK Diák Konferencia (2015 március):** Megnyertük a Petrik TISZK Diák Konferencia informatika szekciót.
- **Matehetsz Üzleti Logika képzés (2015 február-április):** 3 hónapos üzleti képzés Magyar Start-up világában már elismert emberekkel.
- **INNOversitas (2015 május):** Egyetemistáknak rendezett verseny, amit középiskolásként sikerült megnyernünk.
- **T-Systems Innovációs Campus (2015 június):** 2 napos rendezvény, ahol meghívott inspiráló előadóként vettünk részt. Továbbá ott maradtunk együtt dolgozni több kreatív emberrel.
- **Kutatók Éjszakája (2015 szeptember):** Az EIT Digital Budapest meghívott előadói leszünk az éjszaka folyamán az Egymásba gabalyodva rendezvényen (Szervezők: Ericsson, ELTE, BME, EIT Digital Budapest)

Light Labyrinth

Marcell Games

Kovács Marcell, Tóth Máté

Felkészítő tanár: Tekse Marianna

Verseghy Ferenc Gimnázium, Szolnok, Tisza park 1.

1. Bevezetés

A videojátékok piaca mára meglehetősen telített lett. Szinte minden műfajban rengeteg, akár a nagy kiadók által készített játék készült. Ezekkel nyilvánvalóan diákként nem lehet versenyezni. Azonban a játékok piacán is vannak olyan műfajok, amelyekben még nem sok játék készült, illetve még lehetne új, innovatív elemeket belevinni a játékokba. Ilyenek például a labirintusos játékok is. Ha megnézzük a műfaj képviselőit, túl nagy változatosságot nem találunk. A legtöbb ilyen játékban egy adott labirintusból kell kijutnunk a megadott időn belül. E műfajba kíván színt, innovációt vinni e pályázatnak a tárgya: a Light Labyrinth.

2. A Light Labyrinth bemutatása

A játék alapvetően két fő elemből áll: magából a játékból, illetve egy pályaszerkesztőből.

2.1. A játék

A fő innováció a következő: A játékban a játékos nemcsak az idővel versenyzik, hanem a hátralévő fény mennyiségével is. A játékosnak végig kell mennie egy meghatározott útvonalon a starttól a célig, azonban nem látja az egész pályát, hanem csak egy, a karakter körüli körben láthatja az őt körülvevő pályaelemeket. Ahogy az idő fogy, úgy csökken ennek a körnek a sugara, egyben a belátható terület is. a hátralévő időt, a képernyő felső részén látható sárga csík mutatja Ez a koncepció, drámaiabbá teszi az összhátást, hiszen gondoljuk csak el: a fogyó idő mellett mi más tehetné még izgalmasabbá a játékmentet, mint a növekvő sötétség. Természetesen a pályákon találhatóak interaktív elemek is. Ilyenek például a felvehető fény, ami egy meghatározott mennyiségű plusz fényt, ezzel időt ad, vagy akár az egy- illetve kétirányú teleportok is.

A menüben, illetve a pályák alatt is hangulatfokozó zene hallatszik.

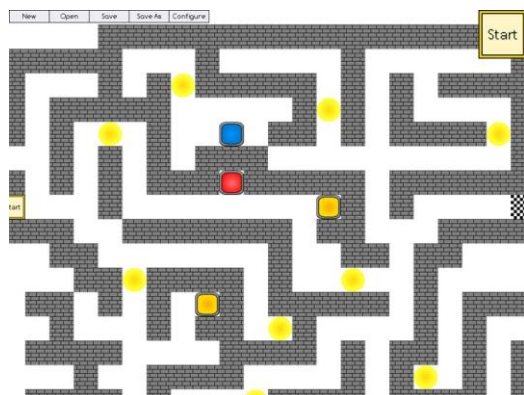
Az irányítást, illetve a pályákon található elemek magyarázatát a következő oldalon álló táblázat fogja tartalmazni. A játék menet az 1. ábrán látható.



1. ábra. Játék menet











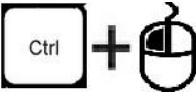





2.2. A pályaszerkesztő

A játék tartalmaz egy pályaszerkesztőt is, melyet a főmenü level editor menüpontjával érhetünk el. A pályaszerkesztőt megnyitva a „new” gombbal tudunk új pályát létrehozni. Ekkor a megjelenő ablakban a készítendő pálya méretét kell megadnunk (szélesség*magasság) ezután megkezdhetjük a pálya létrehozását. (az elemek és az irányítás a táblázatban van kifejtve). Az éppen kijelölt elem a kijelző jobb felső sarkában látható. A kezdő fénymennyiséget, a fény csökkenésének sebességét és a felvehető fények értékét a „configure” gombbal tudjuk állítani.



2. ábra Pályaszerkesztő

Az „open” gombbal a már elkészült pályákat tudjuk visszatölteni szerkesztésre, a „save” gombbal az elkészült pályát tudjuk menteni, a beírt néven. a program alapértelmezetten a C\Felhasználók\{felhasználó neve}\AppData\Roaming\Marcell Games\Light Labyrinth mappába menti. A pályaszerkesztő a 2.ábrán látható.

Billentyű	Hozzárendelt esemény	grafikai elem	Megnevezés
	„előre” (pályaszerkesztő módban)		játékos
	„hátra” (pályaszerkesztő módban)		start
	„balra” (pályaszerkesztő módban)		cél
	„jobbra” (pályaszerkesztő módban)		blokk (fal)
	„teleport használata”		egyirányú teleport be
	„két teleport összekötése” (pályaszerkesztő módban)		egyirányú teleport ki
	„lerakható elemek közti váltás” (pályaszerkesztő módban)		kétirányú teleport
	„kijelölt elem lerakása” (pályaszerkesztő módban)		
	„elem visszavonása” (pályaszerkesztő módban)		

3. Probléma megoldásának menete

A játék C++ programozási nyelv 2011-es ISO szabványának használatával készült, az SDL 2.0 nevű multimédia könyvtár segítségével valamint ennek kiegészítő könyvtárainak az SDL_image, SDL_mixer és SDL_ttf igénybevételével. A program fejlesztésében a saját fejlesztésű game engine is szerepet kapott, amely szintén a felébb említett technológiák segítségével készült. A kivitelezés a Microsoft Visual Studio 2015-ös IDE-ben történt. Mivel ezek a technológiák az összes PC-s és mobil platformon működnek a játék rendkívül könnyen portolható.

3.1. Technikai problémák és megoldásaik

Mivel soha nem használtunk GUI könyvtárakat és az idő nem volt elegendő ezek használatának kitanulására és beépítésére így saját készítésű grafikai kezelő felülettel rendelkezik a program.

Informatikus képzés a Szegedi Tudományegyetemen

SZTE Informatikai Tanszékcsoporthat

Intézetünk 1990-ben alakult Informatikai Tanszékcsoporthat néven a Szegedi Tudományegyetem, Természettudományi és Informatikai Karának részeként. Alapfeladatunk a modern informatikai és számítástudományi ismeretek oktatása, valamint a tudományos életben és innovatív fejlesztésekben való aktív részvétel.

Tekintettel a korszerű informatikai ismeretek iránti általános igényre képzési programjainkban számos hallgató találta meg helyét alap-, és mesterszakos, illetve osztatlan képzésinkben. Az informatikai tudományok világa iránt érdeklődők a doktori képzésünkben mélyíthetik el tudásukat. A mesterképzés feltételeire készülő leendő hallgatók, részismereti képzéseinkben szerezhetik meg a tanulmányik folytatásához szükséges tudást. Végül, de nem utolsó sorban pedig a Szegedi Tudományegyetem, Természettudományi és Informatika Karán számos további szakon biztosítunk informatikai alapképzést.

Hallgatóink számára számos lehetőség nyílik az utazásra, munkára, és kutatásra is. Lehetőség szerint támogatjuk a hallgatók külföldi részképzését. Ennek érdekében külföldi partnerekkel tartunk fent kapcsolatot közös projektek útján, aminek eredményeképpen számos hallgatónk élhetett külföldi részképzésben. Természetesen a hallgatóink bekapcsolódhatnak a tanszékeken folyó kutatásokba is. Ennek eredményeként hallgatóink a mai napig is számos színvonalas kutatási eredményt mutattak be a rendszeresen rendezett Helyi, és Országos Tudományos Diákköri Konferenciákon.

A diploma megszerzése után a kutatás iránt érdeklődő hallgatóknak az Informatika Doktori Iskolánk és oktatóink adnak számos érdekes kutatási témát 2002 óta.

Munkánkban az oktatás mellett számos intézményi, és ipari partnerrel állunk kapcsolatban kutatás-fejlesztési és innovációs projektek megvalósításában is.

Képzéseink közül a gazdaságinformatikus, mérnökinformatikus és programtervező informatikus szakok programja lefedi a teljes képzési skálát, a következőkben ezeket mutatjuk be röviden.

Mérnökinformatikus

BSc/MSc/PhD

Milyen szakterület a mérnökinformatikus?

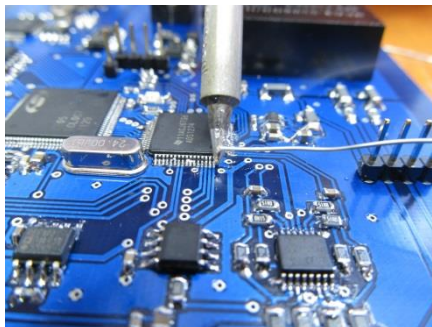
- Informatikai és villamosmérnöki területek kombinálása, az informatika műszaki alkalmazása.

Mit oktatunk?

- Informatika, matematika, fizika alapok,
- számítástudomány, szoftverfejlesztés, képfeldolgozás
- optimalizálás, mesterséges intelligencia,
- egycsipes számítógépek, mobiltelefonok,
- grafikus processzorok programozása,
- robotok, ipari vezérlések, szabályozások, mechatronika,
- elektronika, mérés technika, számítógép-vezérelt műszerek,
- áramkörtervezés, szenzorhálózatok, PLC-k, FPGA-k,
- orvosi informatika, gazdaságinformatika,
- sok más tárgyat szabadon választhatsz.

Hogyan oktatunk?

- Kiemelkedő a gyakorlati óráink, laborjaink aránya.
- A legmodernebb szoftvereket, műszereket találod nálunk.
- Bekapcsolódhatsz kutatásainkba, ipari fejlesztéseinkbe – partnerünk a Siemens, Samsung, National Instruments, Omron és sokan mások.
- Megkereshetsz minket saját ötleteiddel, segítünk a megvalósításban.
- Segítünk, hogy igazi mérnöki szemléleted alakuljon ki.



BSc – szakmát tanulsz meg

- Megismered a szoftverek, modern informatikai és műszaki rendszerek használatát, működtetését.
- Sokféle problémára műszaki informatikai megoldásokat tudsz adni.
- Készen állsz az ipari részvételre, az ipar működtetésére.

MSc – alkotó szaktudást szerzel

- Jóval komolyabb szaktudást szerzel, kreativitást igénylő, izgalmas munkahelyed lehet.
- Komolyabb, önálló fejlesztéseket végezhetsz.
- Hamar vezető beosztású lehetsz.
- Jól tudsz együttműködni informatikusokkal, mérnökökkel, fizikusokkal, vegyészekkel, orvosokkal, közgazdászokkal.

PhD – egy lépéssel az ipar előtt!

- A legmagasabb szintű tudásod, legnagyobb alkotói szabadságod lesz.
- Olyan szakmai munkát végezhetsz, amit a legjobban kedvelsz.
- Sok cég szakmai igazgatója ma már PhD fokozattal rendelkezik.
- Az iparban a legmagasabb szinten állod meg a helyed, de akár kutató is lehetsz.

Mit javasolunk, mit várunk tőled?

- Rád is szükség van a diploma megszerzéséhez, ne hanyagold a tanulást!
- Az állásinterjúknál egy komoly cég azt nézi, hogy mennyit tudsz!
- Kövesd a tananyagot, ne maradj le. Meglátod, így egyre könnyebben fog menni!
- Ha már kellő tudásod van, keress minket, hogy vonjunk be fejlesztéseinkbe!



Tisztelt Szülők!

Miért a szegedi mérnökinformatikus képzést válassza gyermeke?

- Teljes képzési skálát kínálunk: alap – BSc; mester – MSc; doktori – PhD.
- Oktatóink tudása kimagasló, legtöbbjük tudományos fokozattal rendelkezik.
- A műszaki informatikai területet kiemelten fejlesztjük.
- A magyar egyetemek közül csak az SZTE és az ELTE került be a világ legjobb 300-400 egyeteme közé.
- Törődünk a hallgatókkal, bevonjuk őket fejlesztéseinkbe, kutatásainkba
- Szeged sokat nyújtó, emberi léptékű, igazi egyetemváros, szellemi centrum.

Milyen karrier előtt állhat nálunk végzett gyermeke?

- Lehetőséget kap alap ipari állástól egészen a magas szintű vezetői pozíció betöltésére.
- Nemzetközi szinten is értékes diplomát, széleskörű tudást adunk.
- Egyetemen/kutatásban, iparban, Szegeden vagy máshol az egyik legbiztosabb, sokféle állásajánlatnak megfelelő végzettséget biztosítunk.
- A gyermekeiknek nyújtott szülői és anyagi támogatásuk bizonyosan megtérülő befektetés.

Tisztelt Tanárok, Kedves Kollégáink!

Miért ajánlja diákjainak a szegedi mérnökinformatikus képzést?

- Komplex, modern, színvonalas képzés egy komoly egyetemi városban.
- Kiemelten sok labort tartunk, ahol kedvező az oktató/hallgató arány.
- Önállóságra, kreativitásra nevelünk, figyelünk a tehetségekre, bevonjuk őket fejlesztéseinkbe, kutatásainkba.
- Oktatásunk modern eszközökre, gazdag interdiszciplinaritásra épül, oktatóink informatikusok, mérnökök, matematikusok, fizikusok és mások is.
- Középiskolai kapcsolatainkat fejlesztjük, nyílt napokat, bemutatókat, versenyeket szervezünk.

Programtervező informatikus

BSc/MSc/PhD

Mi a programtervező informatikus képzés célja?

Számtalan olyan eszközzel, berendezéssel találkozhatunk, amelyek alapvető működését valamilyen program (szoftver) biztosítja. Ma már nemcsak a számítógépet, hanem a video lejátszót, a tévét, a mosógépet, és a fényképezőgépet is beépített szoftverek működtetik. A programtervező informatikus képes szoftverek létrehozására, továbbfejlesztésére, karbantartására egyénileg vagy csoportmunkában.

Mit oktatunk?

- Matematika alapok,
- számítástudomány, logika,
- optimalizálás, mesterséges intelligencia,
- rendszertervezés, szoftverfejlesztés, alkalmazásfejlesztés,
- programozási nyelvek, web programozás,
- digitális képfeldolgozás, számítógépes grafika,
- fejlett programozási technikák, Windows programozás,
- szabadon választható tárgyak egyéni érdeklődés szerint.

Hogyan oktatunk?

- Az elméleti alapok mellett hangsúlyt helyezünk a gyakorlati képzésre is.
- A tehetséggondozó programunkkal többlet ismeretanyagot biztosítunk.
- A legkorszerűbb szoftvereket, gépeket, speciális és általános célú laborokat biztosítjuk.
- Lehetőséget adunk kutatásainkban és ipari fejlesztéseinkben való részvételre.
- Segítünk saját ötleteid megvalósításában.
- Azon dolgozunk, hogy problémamegoldó szemléleted alakuljon ki.

BSc – egy szakmát tanulhatsz meg, mely képessé tesz

- vállalati információs rendszerek tervezésére és készítésére;
- döntéstámogató rendszerek tervezésére, készítésére, működtetésére;
- szakértői rendszerek fejlesztésére és működtetésére;
- multimédia alkalmazások tervezésére, fejlesztésére és működtetésére.

MSc – alkotó tudást szerezhetsz, mely alkalmassá tesz

- komplex rendszerek fejlesztési, alkalmazási, működtetési tevékenységét önállóan és csoportmunkában magas szinten ellátására;
- a megszerzett tudás alkalmazására és gyakorlati hasznosítására, a problémamegoldó technikák felhasználására vállalati információs rendszerek tervezésében és készítésében valamely korszerű modellező eszköz felhasználásával;
- a tudományágban megszerzett szakmai tapasztalatból származó információk, felmerülő új jelenségek feldolgozására problémák, új jelenségek feldolgozására;
- döntéstámogató rendszerek tervezésére, készítésére, működtetésére, ilyen területen irányító feladatok ellátására;
- a megoldandó problémák megértésére és megoldására, eredeti ötletek felvetésére, az informatika fennálló modelljeinek alkalmazására;
- önművelésre, önfejlesztésre, az egyéni tudás, ismeret elmélyítésére, bővítésére a multimédia eszközeinek felhasználásával is;

A nappali tagozaton a mesterképzés keretében három szakirány közül választhatsz, ezek a következők:

- képfeldolgozás,
- informatikai modellalkotás
- szoftverfejlesztés.

A levelező tagozaton a képzés szakirány nélkül folyik. A mesterképzés nappali és levelező tagozaton a tavaszi és az őszi félévben egyaránt indul. A programtervező informatikus mesterszakon végzettek az SZTE Informatika Doktori Iskolában folytathatják a tanulmányaikat és tudományos (PhD) fokozatot szerezhetnek.

Tisztelt Szülők!

Miért érdemes a programtervező informatikus képzést választani?

- Teljes képzési skálát kínálunk: alap – BSc; mester – MSc; doktori – PhD.
- Oktatóink tudása kimagasló, legtöbbjük tudományos fokozattal rendelkezik.
- Dinamikusan fejlődő, izgalmas terület, rengeteg kapcsolattal élvonalbeli vállalatokhoz.
- A magyar egyetemek közül csak az SZTE és az ELTE tartozik a világ leg jobb 300-400 egyeteme közé.
- Az érdeklődő hallgatókat bevonjuk fejlesztéseinkbe, kutatásainkba.
- Szeged sokat nyújtó, emberi léptékű, igazi egyetemváros, kulturális központ.
- Állások széles skálája az informatikustól egészen a magas szintű vezetői pozícióig.
- Nemzetközi szinten is elismert diplomát, széleskörű tudást adunk.
- A képzés bizonyosan megtérülő befektetés, mivel a gazdaság valamennyi ágazatában keresettek a programtervező informatikusok.

Tisztelt Tanárok, Kedves Kollégáink!

Miért ajánlja diákjainak a szegedi programtervező informatikus képzést?

- Komplex, modern, színvonalas képzés egy rangos egyetemi városban.
- Önállóságra, kreativitásra nevelünk, felfigyelünk a tehetségekre, bevonjuk őket fejlesztéseinkbe, kutatásainkba.
- Oktatásunk a legkorszerűbb eszközökre, sokrétű interdiszciplinaritásra épül, oktatóink informatikusok, matematikusok és más tudományágak képviselői.

The image shows a screenshot of a C code editor. The main window displays the following code:

```
volatile unsigned char INT0counter, PCAcounter;
void INT0handler(void) __interrupt 0
{
    INT0counter++;
}

void PCAhandler(void) __interrupt 9
{
    CF=0;
    PCAcounter++;
}

void main(void)
{
    Init_Device();
    SFRPAGE = PCA0_PAGE;
    PCAUCN = 0x40;
    while (PCAcounter<200);
}
```

An inset window titled "Disassembly Window" shows the following assembly code:

Address	Code	Comment
00FA C2 DE	LDI R16, 0	
00FB 05 09	LDI R17, 9	
00FC 32	LDI R18, 0	
00FD 12 00 E7	LCALL 00E7	
00FE 75 08 40	MOV R18, R18	
0100 74 38	MOV R18, R18	
0101 74 38	ADD R18, R18	
0102 74 38	MOV R18, R18	
0103 74 38	MOV R18, R18	

Gazdaságinformatikus *BSc/MSc/PhD*

Milyen szakterület a gazdaságinformatika?

- Informatikai és közgazdász területek kombinálása, különös tekintettel az informatika alkalmazásaira és az elemző munkára.

Mit oktatunk?

- Matematika alapok, statisztika, sztochasztikus modellek, számítástudomány,
- mikro- és makroökonómia, menedzsment és marketing,
- vállalati pénzügyek, pénzülmélet, számvitel, ipargazdaságtan,
- gazdasági informatika, bankinformatika, üzleti web- technológiák,
- értékpapír piacok, változásmenedzsment, projektmenedzsment,
- logisztika és üzleti modellezés, technikai elemzés a tőzsdei kereskedelemben,
- operációs rendszerek, programozási nyelvek, web tervezés,
- rendszertervezés, szoftverfejlesztés, alkalmazásfejlesztés,
- optimalizálás, mesterséges intelligencia, döntéstámogatás,
- adatbázisok, vállalati információs rendszerek,
- szabadon választható tárgyak egyéni érdeklődés szerint.

Hogyan oktatunk?

- Az elméleti alapok mellett hangsúlyt helyezünk a gyakorlati képzésre is.
- A tehetséggondozó programunkkal többlet ismeretanyagot biztosítunk.
- A legkorszerűbb szoftvereket, gépeket, speciális és általános célú laborokat biztosítjuk.
- Lehetőséget adunk kutatásainkban és ipari fejlesztéseinkben való részvételre.
- Segítünk saját ötleteid megvalósításában.
- Azon dolgozunk, hogy problémamegoldó szemléleted alakuljon ki.

BSc – egy szakmát tanulhatsz meg, mely képessé tesz

- a valós üzleti folyamatok, a folyamatokban rejlő problémák megértésére és megoldására;
- az üzleti problémák IT-vel támogatott megoldására;
- kommunikációs készség birtokában, gazdasági, közgazdasági szakemberekkel, informatikai fejlesztéseket végző munkatársakkal való hatékony együttműködésre

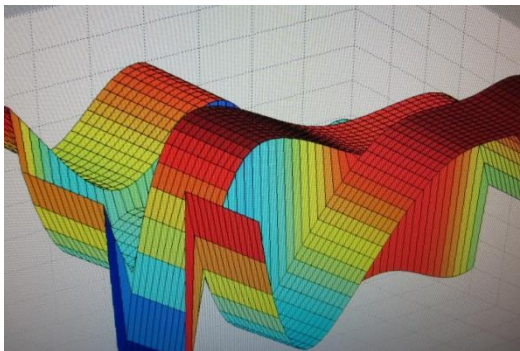
Az alapképzést nappali tagozaton kínáljuk. Az alapszak elvégzése után lehetőség van felvételizni a programtervező informatikus, a gazdaságinformatikus, vagy a mérnökinformatikus mesterszakra.

MSc – alkotó tudást szerezhetsz, mely alkalmassá tesz

- az információtechnológia korszerű lehetőségeinek kihasználására, szervezetek üzleti intelligenciájának a növelésére;
- az IT-támogatott üzleti alkalmazások vállalati szintű, modellszemléletű tervezésére;
- szakmai, emberi és etikai szempontokat mérlegelve önálló irányítói feladatok ellátására;
- az infokommunikációs technológiák együttműködésének megtervezésére, különböző modellnézetek generálására.

A mesterszak nappali tagozaton a tavaszi és az őszi félévben egyaránt indul.

A gazdaságinformatikus mesterszakon végzetek az SZTE Informatika Doktori Iskolában folytathatják a tanulmányaikat és tudományos (PhD) fokozatot szerezhetnek.



Tisztelt Szülők!

Miért érdemes a gazdaságinformatikus képzést választani?

- Teljes képzési skálát kínálunk: alap – BSc; mester – MSc; doktori – PhD.
- Oktatóink tudása kimagasló, legtöbbjük tudományos fokozattal rendelkezik.
- Dinamikusan fejlődő, izgalmas terület, rengeteg kapcsolattal élvonalbeli vállalatokhoz.
- A magyar egyetemek közül csak az SZTE és az ELTE tartozik a világ leg jobb 300-400 egyeteme közé.
- Az érdeklődő hallgatókat bevonjuk fejlesztéseinkbe, kutatásainkba.
- Szeged sokat nyújtó, emberi léptékű, igazi egyetemváros, kulturális központ.
- Állások széles skálája az informatikustól egészen a magas szintű vezetői pozícióig.
- Nemzetközi szinten is elismert diplomát, széleskörű tudást adunk.
- A képzés bizonyosan megtérülő befektetés, mivel a gazdaság valamennyi ágazatában keresettek a gazdaságinformatikusok.

Tisztelt Tanárok, Kedves Kollégáink!

Miért ajánlja diákjainak a szegedi gazdaságinformatikus képzést?

- Komplex, modern, színvonalas képzés egy rangos egyetemi városban.
- Önállóságra, kreativitásra nevelünk, felfigyelünk a tehetségekre, bevonjuk őket fejlesztéseinkbe, kutatásainkba.
- Oktatásunk a legkorszerűbb eszközökre, sokrétű interdiszciplinaritásra épül, oktatóink informatikusok, matematikusok és más tudományágak képviselői.

Az SZTE Informatikai Tanszékcsoporthoz rendezvényei 2015-ben

Laborlátogatások

Tanszékcsoporthoz laborjainkban 2014 óta kínál bemutatókat az érdeklődők (elsősorban diákok) számára, illetve előzetes egyeztetés alapján akár iskolalátogatásokat is szervez az oktató és kutató területek bemutatása, népszerűsítése céljából.

Műszaki informatikai témájú oktató és kutató laborjainkban a látogatók betekintést nyerhetnek az oktatásban is használatos eszközökkel végzett munkára, megtekinthetik az említett célokra használt és fejlesztett robotokat, ipari irányítási feladatokat végrehajtó illetve mérés-technikai, adatgyűjtő műszereket is.

A képfeldolgozás és számítógépes látás laborunkban bemutatjuk az iparban használatos speciális optikákat, képalkotási módszereket valamint ízelítőt kaphatnak a napjainkban egyre népszerűbb és rohamos tempóban fejlődő mobil képalkotási és képfeldolgozási technikákról is.

Kollégáink örömmel tartanak bemutatókat csoportok részére előzetes egyeztetés alapján.

www.inf.u-szeged.hu/dti/laborlatogatas

www.inf.u-szeged.hu/ipcg/laborok

Szakkörök, foglalkozások iskolások számára

A Szoftverfejlesztés Tanszék dolgozói lelkesedéssel és kitartó munkával dolgoznak olyan adatvizualizációs technológiákon, amelyek hasznosak lehetnek mind az iparban, mind pedig a programozás oktatásában is. Fejlesztésük, a *CodeMetropolis* a program forráskódját egy háromdimenziós városként jeleníti meg. Ezzel a módszerrel segít megértetni és megismertetni a diákokkal a programozásban és a szoftverfejlesztésben használt absztrakt fogalmakat.

Az idei évtől kollégáink kísérleti jelleggel szakköri foglalkozásokat is tartottak egyes iskolákban.

www.sed.inf.u-szeged.hu/codemetropolis

Lányok napja

Április 23-án Tanszékcsoportunk is bekapcsolódott az országos *Lányok napja* rendezvénysorozatba. Az érdeklődőket a képzéseinkről tartott előadással, laborlátogatással vártuk. A kreatív vendégek elektronikai alkatrészekből nyakláncot készíthettek maguknak.

www.inf.u-szeged.hu/erdeklodoknek/lanyoknapja

www.lanyoknapja.hu

Kutatók éjszakája

Tanszékcsoportunk ismét látványos bemutatókkal és rejtvényekkel készült az év legnagyobb tudományépszerűsítő programjára, a szeptember 25-én megrendezett Kutatók Éjszakájára.

A Szent-Györgyi Albert Agórában az érdeklők a Szoftverfejlesztés Tanszék munkatársai segítségével kipróbálhatták, hogyan lesz egy számítógépes program forráskódjából város a *CodeMetropolis*-ban. A Műszaki Informatika Tanszék munkatársai látványos izgó-mozgó robotokkal, ipari irányításokat bemutató modellekkel és élettani jelek telemedicinás mérésével szórakoztatta a közönséget. A Képfeldolgozás és Számítógépes Grafika Tanszék pedig az orvosi képalkotási technológiákról és informatikai módszerekről tartott bemutatót.

www.kutatokejszakaja.hu/2015/programok/

Kari nyílt nap

Hagyományos nyílt napunk 2016. január 16-án lesz, ahol a szülők és a leendő hallgatóink tájékoztatást kaphatnak intézetünk vezetőitől mind a képzéseket, mind pedig tanulmányok befejeztét követő álláskereső sikerességét illetően. Munkatársaink ismertetik az oktatásban használatos szoftver és hardver eszközök széles palettáját, valamint bemutatót tartanak kutatásaikhoz kötődő fejlesztéseikből továbbá a diákok órai illetve szakdolgozati és diplomamunkáiból.

www.u-szeged.hu/felveteli/kari-nyilt-napok

Kiadta: SZTE Informatikai Tanszékcsoport

Készítette: Vadai Gergely

Design: Németh Gábor

Nyomda: Innovariant Nyomdaipari Kft.

Készült: 100 példányban



**NATIONAL
INSTRUMENTS**



SZTE TTK
INFORMATIKAI INTÉZET



NNG



MAGYARORSZÁG
KORMÁNYA

SZÉCHENYI 2020



Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE