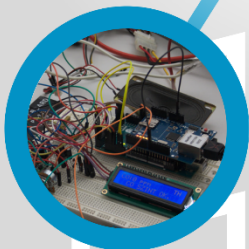
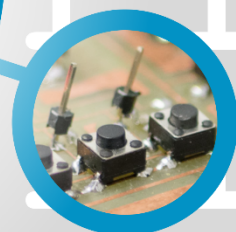
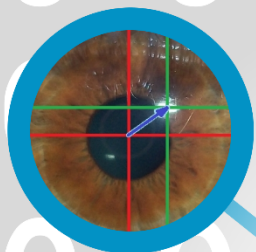


Szegedi Innovatív Informatika Verseny

2018



SZÉCHENYI  2020



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

SZIIV Program (2018. április 6. – péntek)

Időpont	Műszaki informatika szekció	Informatika szekció
9:00–10:00	Regisztráció	
10:00–10:30	Megnyitó	
10:40–12:20	Előadások I.	
10:40–11:00	Faficák: <i>Gyógyszerek szilárdságának mérése</i>	magányosHarco\$: <i>ConTags</i>
11:00–11:20	Bodó István: <i>Project smart home</i>	HungaroCode: <i>MySchedule</i>
11:20–11:40	Šitte\$\$hekh: <i>Az okos napelemes hőkollektor</i>	SootSoft: <i>e-Gradu</i>
11:40–12:00	//noComment: <i>Déri-door</i>	Cyber-sentiment: <i>SentiMentality – karnyújtásnyira a közvélemény</i>
12:00–12:20	Szvoreny Viktor: <i>Az ember, mint gépezet – Orvostudomány számítástechnikai szemmel nézve</i>	Voidloop: <i>Memeium & Memeium Services</i>
12:30–13:30	Szendvicsebéd	
13:40–15:20	Előadások II.	
13:40–14:00	Fajka Viktor: <i>Vetőgép 2.0</i>	csapatnév: <i>Arcfelismerés alapú beléptetőrendszer</i>
14:00–14:20	Berendi Bence: <i>Okos ház</i>	Mészáros Bálint: <i>Sapphire</i>
14:20–14:40	Robot TÉP: <i>Robot TÉP Szakkör a Szikszai 100-ban</i>	fsociety: <i>Rubik kocka kirakó program</i>
14:40–15:00	BlueIsBetter: <i>„Csepp” kilövő HUSAR rover</i>	My8it: <i>Ne szokj rá a dohányzásra!</i>
15:00–15:30	Eszközök, szoftverek előkészítése	
15:30–16:00	Kávészünet	
16:00–17:30	Eszközök, szoftverek bemutatása	

KTPK Program

Időpont	2018. április 6. - péntek
9:00–18:30	A három verseny döntője, zárórendezvénye
9:00–17:30	Szegedi Innovatív Informatika Verseny SZTE Bolyai épület
13:00–18:30	Polygon Pályázat Matematikából Középiskolásoknak SZTE Bolyai Épület
15:00–17:30	Játsszunk fizikát! Kármán Tódor Emlékverseny SZTE Dóm tér 9.
18:00–20:00	Vacsora
	2018. április 7. -szombat
9:00–13:00	Előadások, interaktív programok és eredményhirdetés SZTE Dóm tér 9.
13:00–14:00	Szendvicsebéd



Tartalomjegyzék

<i>SZIIV Program</i>	1
<i>KTPK Program</i>	2
<i>Előszó</i>	5
<i>Programbizottság</i>	6
<i>Műszaki informatika szekció</i>	11
Végh Natasa, Gémes Gergő, László Csaba, Szilágyi Péter: Gyógyszerek szilárdságának mérése.....	13
Bodó István: Project smart home.....	17
Hugyik Kornél, Barsi Árpád, Horváth István: Az okos napelemes hőkollektor.....	21
Zsibók Márk, Matkó Kende, Juhász Norbert: Déri-door.....	25
Szvoreny Viktor: Az ember, mint gépezet – Orvostudomány számítástechnikai szemmel nézve.....	29
Fajka Viktor: Vetőgép 2.0.....	33
Berendi Bence: Okos ház.....	35
Magyar Dávid, Felföldi János, Vitkos Bence: Robot TÉP Szakkör a Szikszai 100-ban.....	39
Bejó Mátyás, Joób Zalán, Molnár Barnabás: „Csepp” kilövő HUSAR rover..	43
<i>Informatika szekció</i>	49
Horváth István: ConTags.....	51
Ignác Dominik Bence, Patrovits Levente, Salamon Donát, Bitai Bence: MySchedule.....	55
Korom Richárd: e-Gradu.....	59
Kónya Leon, Sztarek Norbert: SentiMentality – karnyújtásnyira a közvélemény.....	63
Rátki Barnabás: Memeium & Memeium Services.....	67
Suszter Ágnes Eleonóra, Robotka Adrián: Arcfelismerés alapú beléptetőrendszer.....	71
Mészáros Bálint: Sapphire.....	75
Bakos Kristóf Nándor, Hunkó Dániel, Nagy Ádám: Rubik kocka kirakó program.....	79
Cseri Gábor, Kocsis Kamilla, Kiss Gergő Csaba: Ne szokj rá a dohányzásra!.....	83
<i>Együttműködő partnereink</i>	89
<i>SZTE TTIK Informatikai Intézet</i>	93



Előszó

A Szegedi Tudományegyetem Informatikai Intézete immáron negyedik alkalommal hirdetett versenyt a középiskolás diákok számára. Ezévtől a verseny az egyetem Természettudományi és Informatikai Kara által indított Kreatív Tudományos Program Középiskolásoknak rendezvény keretei között kerül megrendezésre további két, nagy múlttal rendelkező verseny mellett.

A természettudományos és műszaki területek iránt érdeklődő tehetséges középiskolás diákok e két napos eseményen találkozhatnak a területek egyetemi és ipari képviselőivel, a versenyeket követő második napon az Együtt a Jövő Mérnökeiért Szövetség közreműködésével szervezett izgalmas bemutatókon és interaktív programokon vehetnek részt.

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 csak programozási feladatból álljon. A diákoknak meg kellett építeni, vagy már meglévő elemekből össze kellett állítani egy rendszert, amely működtetéséhez szükséges szoftvert is a nekik kellett elkészíteni.

Az **Informatika** szekcióba olyan pályamunkákat vártunk, 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 a két szekció zsűrije 13 belföldi és határon túli település iskoláiból beérkező pályaművekből összesen 18 csapatot válogatott be a szegedi döntőbe, mely 2018. április 6-án került megrendezésre az SZTE Bolyai épületében.

Ez a kiadvány a döntőn bemutatásra kerülő 18 pályamunkát foglalja egy kötetbe.

Szeged, 2018. április

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. Mingesz Róbert
adjunktus
SZTE Műszaki Informatika Tanszék



Dr. Pletl Szilveszter
főiskolai tanár
SZTE Műszaki Informatika Tanszék



Kovács Tamás
ügyvezető
Optin Kft.



Trauer János
középiskolai tanár (fizika-informatika szak)
Makói József Attila Gimnázium, Makó

Informatika szekció

A zsűri elnöke:



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

A zsűri tagjai:



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



Dr. Dombi József Dániel
projektvezető, tudományos munkatárs
SZTE Szoftverfejlesztés Tanszék



Fritz Zsombor
fejlesztési központ vezető, Szeged
NNG Kft.



Fodor Zsolt
középiskolai tanár (matematika-fizika-informatika
szak)
SZTE Gyakorló Gimnázium és Általános Iskola, Szeged

Szervezők



Vadai Gergely
tanársegéd
SZTE Műszaki Informatika Tanszék



Dr. Németh Gábor
adjunktus
SZTE Képfeldolgozás és Számítógépes Grafika Tanszék



Dr. Jász Judit
adjunktus
SZTE Szoftverfejlesztés Tanszék



Dr. Kincses Zoltán
adjunktus
SZTE Műszaki Informatika Tanszék

Műszaki informatika szekció

Gyógyszerek szilárdságának mérése

Fafcicák

Végh Natasa, Gémes Gergő, László Csaba, Szilágyi Péter

Felkészítő tanár: Gutai Árpád Tamás

Szegedi Radnóti Miklós Kísérleti Gimnázium, 6720 Szeged, Tisza Lajos krt. 6-8.

1. Bevezetés

A gyógyszerek a mindennapjaink szerves részévé váltak, és nap mint nap rengeteg fajtájukkal találkozhatunk. Mi elsősorban a tablettákkal, és azoknak a szilárdságával foglalkozunk.

A tabletták keménysége sok szempontból fontos lehet, ezért már az 1930-as évektől kezdve elkezdtek mérni ezt a tulajdonságot először levegőszivattyú, majd elektromechanikus gépek segítségével.

A két fő fajtája a tablettáknak a szájbán és a bélben felszívódó tabletták. A szájbán felszívódó tablettáknak legelterjedtebb formái a rágótabletták. Ezeknél a tablettáknál fontos, hogy a szállítás és tárolás közben ne essenek szét, viszont viszonylag könnyen megrághatóak legyenek. A bélben felszívódó, vagyis a „lenyelős” tablettáknál már több dolgot figyelembe kell venni. Ezeknél is fontos, hogy a szállítás és tárolás közben egyben maradjon a tableta, de még a szájbán se, és lenyelés közben se essen szét, de eközben kárt se okozzon a nyelőcsőben. Továbbá a felszívódásnál egy tömörebb, keményebb tableta nehezebben bomlik szét, vagyis hosszabb ideig tart a hatóanyag kioldódása, lassabban szívódik fel és később kezd el hatni, valamint több ideig hat, tehát azoknál a gyógyszereknél, amelyeknek meghatározott idő alatt kell felszívódniuk és megadott ideig hatniuk, fokozottabban figyelembe kell venni a tableta szilárdságát.

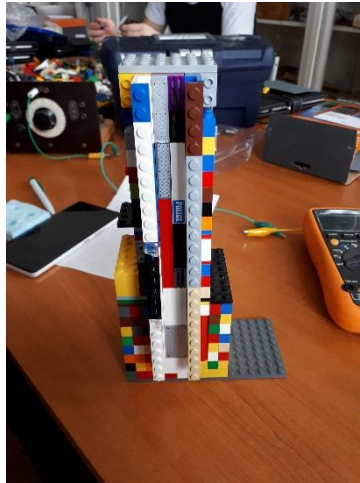
A mi célunk egy olyan berendezés létrehozása volt, amellyel az eddigiéknél egyszerűbben és olcsóbban, de viszonylag pontosan meg tudjuk mérni a tabletták szilárdságát nagyrészt könnyen beszerezhető eszközök segítségével.

2. Probléma megoldásának menete

A teljes berendezés több részből áll. Alapvetően szükségünk volt egy állványra, amelyre a tablettát ráhelyezhetjük, egy olyan eszközre, amely segítségével a tablettára az idővel arányosan egyre nagyobb erőt tudunk kifejteni, valamint egy olyan eszközre, amely segítségével tudjuk mérni ezt a bizonyos erőt.

2.1. Kivitelezés

A tabletta egy könnyen mozgó, rugókkal alátámasztott platformon van. A tablettára egy elektromotor erőt fejt ki egy fogaslécen keresztül. A fogasléc sínje legóból készült.



1. ábra: A fogasléc sínje

A tablettát tartó rugós platform (házi esztergálással) fémből készült, a fogasléc és a fogaskerekek 3D nyomtatással készülnek el.



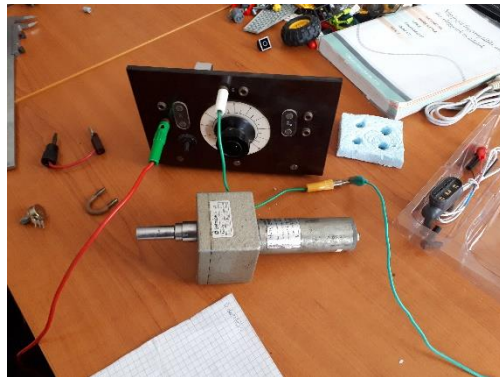
2. ábra: A rugós platform

A motor tápellátását egy házi-készítésű transzformátor és egyenirányító szolgáltatja.



3. ábra: A motor tápellátását biztosító készülék

A motor fordulatszámát egy potenciométerrel szabályozzuk, amit az iskola szertárából vettünk kölcsön.



4. ábra: A motor és a szabályozást segítő potenciométer

2.2. A berendezés működése

A motor egy fogaskereket hajt meg, ami egy fogaslécet kezd el mozgatni. A fogasléc rányomódik a tablettára, amely egy bizonyos terhelés hatására megreped, széttörik. A terhelés növekedése közben a platformot tartó rugók egyre inkább összenyomódnak. Mivel párhuzamosan kötött rugókról van szó, így rugóállandójuk összeadódik. A tabletták átlagosan 60-80 N nagyságú erők hatására törnek szét, az állványunkkal több, mint 100 N nagyságú erőt is tudunk mérni.

A mérés úgy zajlik, hogy a platformra szintén ráerősítünk egy fogaslécet, amely egy fogaskereket forgat mozgása során. A fogaskerék egy olyan potenciométerre lesz ráerősítve, amely egy Arduino UNO eszközzel lesz kapcsolatban. Az Arduino segítségével 5 V feszültséget tudunk kapcsolni a potenciométerre, és tudjuk mérni két szomszédos lába között eső feszültséget. Mivel a platform összenyomódása egyenesen arányos a rá kifejtett erővel, ezért a potenciométer elfordulása is az, a mért értékeket ábrázolva a grafikonon látszódik, hogy mikor tört meg a gyógyszer.

3. Elért eredmények

Sajnos a berendezés a pályázat beadási határidejére nem készült el teljesen, csupán a két fogaskerékre, valamint a két fogaslécre várunk, hogy elkészüljön, így eredményeket egyelőre nem tudunk felmutatni.

Project smart home

Bodó István

Felkészítő tanár: Fekete Balázs

*DSZC Mechwart András Gépipari és Informatikai Szakgimnáziuma,
4025 Debrecen, Széchenyi u. 58.*

1. Bevezetés

Napjainkban elég széleskörben elterjedtek az okos eszközök, berendezések. Többségüket (a magas árcédula mellett) az jellemzi, hogy alkalmazás segítségével vezérelhető okostelefonról. Ez több problémát is felvet. Először is ha nem egymárkájú termékeket vásároltunk különböző alkalmazásokat kell használnunk az egyes eszközök kapcsolásához, vezérléséhez. Ez rendkívül időigényes és alkalmanként az ember türelmét is próbára teheti. Nem beszélve arról, hogy legtöbbször Bluetooth technológiát használ, így a távoli vezérlés nem megoldható, illetve egy lassú kevésbé biztonságos átviteli közegről beszélünk. Ez a projekt a „nem okos” eszközök felokosítását veszi alapul, illetve azok irányítását egy webes felület segítségével.

2. Probléma megoldásának menete

A cél egy olyan rendszer megalkotása mellyel központilag vezérelhetjük eszközeinket, jelen esetben a mennyezeti lámpát, monitort, számítógépet, erősítőt. A megvalósításhoz a hardveres, illetve szoftveres ismeretek is szükségesek.

2.1. Hardveres megvalósítás

A fizikai rész egy sajátkézzel alkotott „doboz”, mely az (1) ábrán látható. A rendszer központjában egy Raspberry Pi 3-as miniszámítógép ül, melyhez relék vannak kapcsolva. Egy reléhez egy konnectoraljzat tartozik, így lehetőség van egyesével kapcsolni őket. A doboz oldalán megfigyelhető 2 Rj-45-ös falialjzat mely egyrészt az internet kapcsolatot, másrészt különböző eszközök, szenzorok kapcsolását teszi lehetővé. Továbbá egy tápcsatlakozó, és egy billenőkapcsoló, mellyel az egész doboz áramellátását lehet szabályozni. Az (1) ábrán látható 8 konnector csoportokba rendezhető, majd ezt követően elnevezhető. A csoportosítás például egy több monitoros rendszerrel alkalmazható. Az eszközöket IR-LED-el is lehet irányítani, példaként az erősítő bekapcsolás után leveszi a hangerőt 30-as értékre és átkapcsol a PC hangbemenetére. (A számítógép bekapcsolása WOL csomag küldésével történik, kikapcsolását egy C#-ban írt szoftver végzi.) A lámpa kapcsolásánál

külön a falba épített Raspberry Zero W végzi a feladatot, szintén egy relét kapcsolgatva zár, illetve nyit egy áramkört. A falba épített doboz (2) ábra, szigetelve van, illetve tűzálló.

2.2. Szoftveres megvalósítás

A Raspberry-n egy webkiszolgáló várja a felhasználó interakcióját. Az egyes kérésekre lefut a megfelelő, php-ban megírt script.

Fontos szempont volt, hogy az eszközöket ne csak a webes felületről lehessen vezérelni (3) ábra. Például a lámpa felkapcsolását egy nyomógombbal is véghez lehessen vinni, hiszen nem mindig van nálunk okostelefonunk, illetve így kényelmes az egész rendszer használata. Éppen ezért logikailag el van választva a vezérlésért felelős rész a webes felülettől.

Minden eszköznek megvan a saját be/ki kapcsolást vezérlő programja. A lámpa felkapcsolása esetén a következő játszódik le:

- A felhasználó megnyomja a nyomókapcsolót
- A falba épített Raspberry érzékelve ezt kérést küld a vezérlő Raspberry-nek
- A vezérlő értelmez és végrehajtja a megfelelő programot, ami jelen esetben egy felkapcsolást kezdeményező kérés a falba épített Raspberry felé
- A kérés végrehajtásra kerül, a relé zárja a lámpa áramkörtét

2.3. Monitorozás

Ha naplóbejegyzést készítünk az eszközök be/ki kapcsolásáról, kiszámíthatóvá válik az adott eszköz működésének időintervalluma, ebből statisztikákat állíthatunk fel. Illetve grafikusán ábrázolhatjuk egy felületen. (4) ábra. Továbbá, ha tudjuk az eszközök átlagos fogyasztását, kiszámolható mennyi kilowattórát fogyasztottak.

2.4. Biztonság

Fontos az ilyen rendszerekben a biztonság. Nem szeretnénk, hogy az eszközöket jogosulatlan felhasználók vezéreljék, ezért elengedhetetlen legalább felhasználó, illetve jelszó páros a grafikus felület hozzáférésehez. Tovább fokozhatjuk a biztonságot, hogy publikus hálózatokból csakis VPN hozzáféréssel legyen elérhető a felület. A Raspberry-k egymás között tanúsítvány alapú kliens oldalú hitelesítéssel kommunikálnak HTTPS protokollon keresztül, így titkosítás is történik.

2.5. Automatizálás

Lehetőség van az eszközöket időzíteni vagy valamilyen szenzor által érzékelt adat alapján bekapcsolni. Jelenleg a rendszer mozgás és fényérzékelővel rendelkezik. Így lehetőség van például egy reggeli lámpakapcsolásra (6:00-tól,

ameddig a felhasználó nem hagyja el az épületet), vagy egy éjszaka a mozgásra 5 percre felkapcsoló „éjjeli” lámpára. Másik jó példa a betörés elleni védelem, mellyel a rendszer képes „tettetni” hogy a felhasználó otthon van.

3. Ábrák



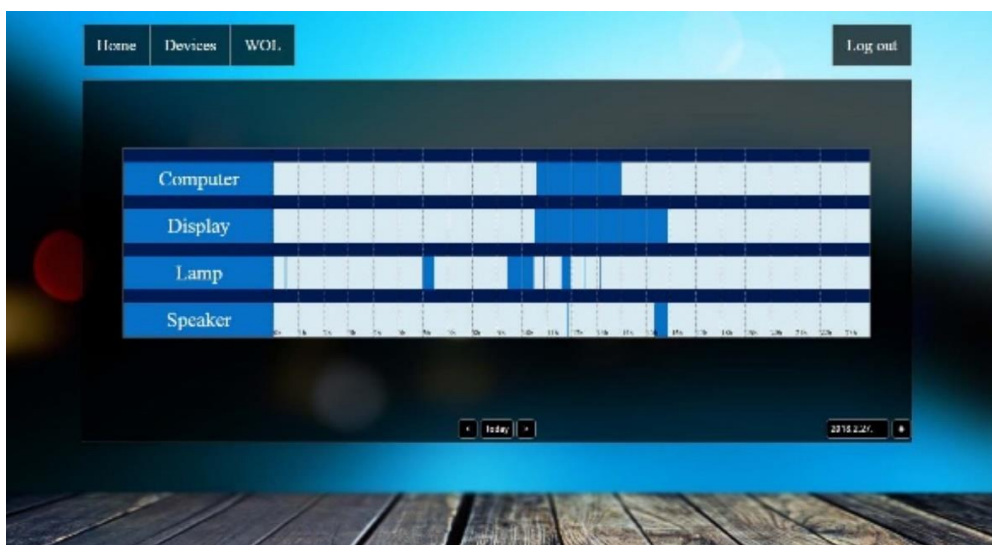
1. ábra: Vezérelt konnektorok



2. ábra: A falba épített doboz, illetve kapcsoló



3. ábra: Grafikus felület



4. ábra: Monitorozás grafikus felületen

4. Elért eredmények

A céloom egy olyan rendszer építése volt mely megkönnyíti a hétköznapi életem. A projekt jelenleg a hálósobámban fut már több hónapja. Már szinte természetesnek érzem az eszközök ily módon kapcsolását. Később szeretném kibővíteni a rendszert az egész házra, beleértve a redőnyök, riasztó, garázskapu stb. vezérlését.

Az okos napelemes hőkollektor

\$itte\$hekh

Barsi Árpád, Huguik Kornél, Horváth István

Felkészítő tanár: Esztelecki Péter

Bolyai Tehetséggondozó Gimnázium és Kollégium, 24400 Zenta, Posta utca 18.

1. Bevezetés

1.1. A kollektorról magáról

A kollektort tanárunk készítette azzal a céllal, hogy ne kelljen olyan sokat költeni a fűtésre. 2017 tavaszára lett kész, és attól kezdve használatban van. Egy szép napos tavaszi napon akár elérheti a 70C-t is.

1.2. A vezérlés előtti helyzet

Tanárunk egy ismerősétől szerzett egy „Rezgő adagolót”, ami egy PWM vezérlő elektronika, az eszközbe kettő potenciométer volt dugva, amivel tudta manuálisan irányítani a motorokat.

1.3. A vezérlés

A 2017/2018-as iskolaév kezdetén kapta csapatunk a feladatot, hogy csináljunk egy vezérlést ehhez a kollektorhoz, a következőkben a vezérlés három része lesz részletesen bemutatva. Elsőször a hardware, majd a két software, először a netes adatbázis, majd a telefonos alkalmazás.

2. A Hardware

2.1. A szenzorok

A kollektoron, és körülötte három hő, és egy fényszenzor van elhelyezve. A három hő szenzor, melyeknek pontos fajtája DS18B20 a következő helyeken vannak elhelyezve: egy a kollektorba fűrt lyukban, ami a kollektoron belüli hőmérsékletét méri, egy a kollektor tetején, ami a külső hőmérsékletet méri, és egy bent a házban, ami a benti hőmérsékletet méri. Az egy fényszenzor pedig a kollektor tetején van elhelyezve, ami a napfény mennyiséget nézi, és ezzel meg lehet határozni, hogy a kollektorban meddig lesz még meleg. A szenzorok méréseit egy Node MCU nevű microcontroller fogadja, dolgozza fel, vezérli a motrokat, és küldi a szervernek. Pontosan azért lett ez a microcontroller választva, mert semmi külső modul nélkül rá tud csatlakozni az internetre (beépített wifi chip), elég erős a processzora a számolásokhoz, és van elég lába (PIN-e), mind ennek a lekezelésére.

2.2. A motorok irányítása

Vagy a szenzorok által mért értékeket felhasználva, vagy a szerverről kapott manuális vezérléssel irányítja a motrokat. Én most az előbbiről szeretnék beszélni. Az irányítás profilokra épül, ezek a profilok egy egyszerű egydimenzós Fuzzy vezérlési görbék. Alapvetően két fajta profil van, a belső fókuszú, ami a benti hőmérsékletet nézi, és az alapján irányít, és külső fókuszú, ami a kollektorban lévő hőmérsékletet nézi, és az alapján irányít. Ezeken belül még van 3-3 kis profil, melyek a következők: optimális profil, ami kellemes hőfokot biztosít a szobában, egy minimális, amely a csendes működésért felel, és egy maximális, ami ha meleg van a kollektorban, akkor 100%-osan fűj. Amikor kiszámolja az értéket a microprocesszor, akkor készít egy PWM jelet, és köztes elektronikának küldi, ami az előbb említett "Rezgő adagoló"-nak alakítja úgy a jelet, hogy az az értékeknek megfelelően hajtsa meg a motrokat.

2.3. A szerverrel való kommunikáció

Mivel a Node MCU-ban van egy beépített wifi chip, ezért tud kommunikálni a szerverrel. Ha nappal van, akkor minden 5 másodpercben küldi az adatokat, és olvassa a szervertől kapott utasításokat, ha pedig este van, akkor percenként kommunikál vele. A szerveren lehet beállítani, hogy melyik profilt használja, és akár manuálisan is megadhatjuk, hogy mennyivel menjenek a motrok. A telefonos alkalmazás pedig ezzel a szerverrel kommunikál, hogy megjelenítse a kollektor mostani állását, és hogy állíthassunk a vezérlésen.

3. A Szerver és az Adatbázis

3.1. A Hardware-rel való kommunikáció

Erre a célra két PHP alapú weboldal készült, az egyik fogad adatokat, a másik pedig "küld". A fogadó rész úgy működik, hogy GET-es átadással működik, a microprocessor meghív egy webcímet, az oldal pedig kiolvassa az URL-ből az értékeket (hőmérsékletek, fény, motorok forgása), és eltárolja azokat egy adatbázisban. A "küldő" rész pedig szimplán annyi, hogy kigenerál egy weboldalat, melyeken a manuális irányítás paraméterei, profilok kiválasztva jelenik meg ","-vel elválasztva, és ez az oldalat olvassa, és szedi szét a microprocessor.

3.2. A manuális vezérlés

Erre a célra van egy weboldal, amin van 3 csúszka, az első az 0-1 értékekkel, és azt mondja meg, hogy a manuális vezérlés be legyen e kapcsolva, a másik kettő 0-100 as skála, amin a két motor forgását lehet állítani.

3.3. A statisztikák

Erre a célra több „Google Charts”-ra épülő grafikon is készült, különböző „felbontásokkal”. A felbontás itt az takarja, hogy mennyi adatból állítsa össze a grafikont, hiszen ha túl sok adatot használ hozzá, akkor sok memóriát fog foglalni, és sokáig fogja generálni a grafikont. Két fajta grafikon van, az egyik, ami a mai nap eddigi eredményeit foglalja magába, a másik pedig, ami az előző nap eredményeit.

4. A ritkítások, avagy helytakarékoskodás

Mivel naponta több száz-ezer adat érkezik, ezért gyorsan betelhet a hely. Például, a rendszer a téli szünet kezdetekor lett beüzemelve, és azóta 2GB adat gyűlt össze. Ezért minden hét végén végig megy az adatokon, átlagolja őket, az előzőeket kitörli, és beírja a jókat.

5. Az alkalmazás

5.1. Az alkalmazásról

Az alkalmazás „WebView”-kal jeleníti meg a weboldalról a grafikonokat, a profilválasztót, és a manuális vezérlést. Igen kevés helyet foglal, kevés memóriát használ, gyorsan fut, és a célt tökéletesen teljesíti.

5.2. Login

A mobil applikációban és a webes részben is található egy Login felület, ami megakadályozza, hogy csak a megfelelő személyek juthassanak hozzá a kollektor adataihoz és irányításához, emellett az adatokat titkosítja az adatokat.

Déri-door

//noComment

Zsibók Márk, Matkó Kende, Juhász Norbert

Felkészítő tanár: Húséné Zsédely Ibolya

*SzSzC Déri Miksa Szakgimnázium és Szakközépiskola,
6724 Szeged, Kálvária tér 7.*

1. Bevezetés

A világ az IoT eszközök fejlődésével teljesen megváltozott. A környezetünkben lévő okos eszközök ott könnyíthetik meg akár egy iskolai közösség hétköznapijait is, ahol nem is számítanának rá.

Iskolánk udvarára, a kerékpártárolókhoz igen nehéz bejutni, mert ez csak egy, a főépület mögött lévő ún. kiskapun lehetséges, ami iskolaidőben zárva van, és csak kulccsal nyitható. A diákok nem rendelkeznek kulccsal, s ha a tanítás előtt el is helyezték kerékpárjukat a kijelölt helyre, nehéz kivinni délután azt.

Az iskola fő profilja az automatizálási technikák, módszerek oktatása, így eszünkbe jutott, mi lenne, ha a rendelkezésünkre álló eszközökből, néhány jó ötlet és természetesen sok munka árán, megoldást találnánk a fent vázolt probléma megoldására.

A célunk egy olyan skálázható rendszer fejlesztése, mely intelligens beléptetést tesz lehetővé, és nem utolsó sorban könnyen menedzselhető és olcsó.

Iskolánk igazgatóságának engedélyével összeállt egy csapat a beléptetőrendszer megtervezésére és megvalósítására, 10-11.évfolyamos, informatika tagozatos diákokból.

2. Probléma megoldásának menete

Az iskola környezeti viszonyai megkövetelik a könnyű felszerelhetőséget és az egyszerű elérhetőséget. Egy másik fontos szempont, amit figyelembe kellett vennünk a rendszer tervezésénél, hogy gyors és követhető visszajelzést nyújtson a felhasználók felé. Választásunk egy könnyen kezelhető chat botos megoldásra esett. Mivel nem mindenki rendelkezik okos eszközzel, amely képes lenne az alkalmazás futtatására, ezért egy kártyás beléptető modul implementáltunk a Telegramos chat bot mellé. Ez a modul érintés nélküli azonosításra képes, ki- és belépéskor.

2.1. Fejlesztéshez használt eszközök

- NodeMcu ESP8266-os chip alapú mikrokontroller
- 2db RC522-es magas frekvenciás kártyaolvasó modul
- Optocsatolt relé
- Nyitó kapcsoló
- 12 V-os DC zárnyelv

2.2. Fejlesztéshez használt szoftverek

- Atom szöveg szerkesztő szoftver
- PlatformIO beépülő modul
- GitHub szoftver történet követő szolgáltatás
- Travis-CI folyamatos integrációs szolgáltatás
- Eagle nyáktervező szoftver

3. A fejlesztés menete

A fejlesztés a megfelelő központi egység kiválasztásával kezdődött. A választásunk az ESP8266 alapú NodeMcu-ra esett, mivel beépített wifi modullal, 4MB flash memóriával és C++ támogatással rendelkezik. Mivel többen fejlesztettük az eszköz firmware-ét, ezért nem a szokványos Arduino IDE környezetet választottuk. A PlatformIO bővítménnyel rendelkező Atom szövegszerkesztőt használtuk, mely több hasznos funkciót tartalmaz.

3.1. Szoftver leírása

A szoftver képes online és offline azonosítani a rendszert használó tanárokat és diákokat. A rendszer folyamatosan szinkronizálja a hozzáférési adatokat és a ki és belépéseket egy távoli szerverre. Ezzel létrehozva egy felügyeleti rendszert. Az eszköz rendelkezik egy NTP klienssel is, amely folyamatosan frissíti a pontos időt az eszközön, így megspórolva egy RTC hardveres modult. Mivel a tanítás véget ér 15:00 órakor, a rendszer lehetővé teszi a diákoknak az iskolából való kijutást, belülről egy kapcsolóval könnyedén kinyithatja az ajtót. Ez csak egy irányú.

3.2. Felmerülő problémák

A forráskód moduláris felépítése ellenére is nehézséget okozott a megosztott kódolás. A projektvezető javaslatára a GitHub nevezetű szoftververzió-kezelő rendszert használtuk. Használatát a csapatunk tagjainak meg kellett tanulnia, de viszonylag könnyen elsajátították a szükséges ismereteket. Az elektronika elkészítésekor figyelembe kellett venni, hogy a tároló magas helyre lesz felszerelve, ezzel megnehezítve a firmware folyamatos frissítését. Két frissítési

módszert terveztünk hozzá, Over-the-Airt és webserviceUpdate-et. Ezek a technikák közösen hozzájárulnak az eszköz skálázhatóságához és folyamatos frissítéséhez.

A kódban nem akartuk statikusan lefixálni a felhasználók belépési adatait, mivel ez nem biztonságos és nem bővíthető. Ezért létre kellett hoznunk egy távoli sql szerveret, mely tárolja a hitelesítési adatokat és naplózza a be és kilépéseket. Így a forráskód módosítása nélkül, közvetlenül a chat boton keresztül tudunk regisztrálni új felhasználókat.

Mifare Classic magas frekvenciás kártyákat használtunk az RC522-es modulhoz, amelyek nagyon olcsók, de rendelkeznek egy biztonsági hibával: Crypto1-es nevezetű titkosítást használnak, és minden adatot tartalmazó szektor 48 bites titkosítási kulccsal rendelkezik. Egy egyszerű telefonnal 10 percen belül le lehet másolni a kártyát. A támadás megelőzéséeként egy integritás védelmi modult kellett fejlesztettünk hozzá, mely detektálja a kártya esetleges másolását, így nem lehet ezzel az épületbe belépni.

4. Elért eredmények

A Déri Miksa Szakgimnáziumunk számára intelligens, skálázható beléptető rendszert terveztünk, mely egy chat boton keresztül menedzselhető. A eszköz online és offline is képes beléptetni a felhasználókat az RFID kártyaolvasójának köszönhetően. A tervezett eszköz olcsó, egyszerűen kezelhető, moduláris. A projektben sikerült megvalósítanunk egy olyan modellt, amely működőképes, mind hardver-, mind szoftver-, mind kommunikációs területen. Így leegyszerűsödik, és korszerűvé válik az intézményünkbe való bejutás.

5. További fejlesztések

A mikrokontroller rendkívül könnyen fejleszthető és bővíthető egység. (De a rövid határidő miatt nem sikerült kihasználni az eszköz által nyújtott lehetőségeket!) A projektet szeretnénk tovább fejleszteni, hogy később az iskolai termeket is felszerelhessük vele, ezáltal a teljes iskolában feltérképezhető lenne a termék kihasználtsága, beosztása, akár helyettesítés esetén a szabad helyiségek azonnali átadhatósága.

A jövőben, szeretnénk lecserélni a jelenlegi mikrokontrollert egy több porttal rendelkező ESP32 chippel rendelkező controllerre. Emellett létrehoznánk egy lokális, központi szerveret, amely akár egy Raspberry Pi is lehetne, az alacsony fogyasztása miatt.

A chatbot rendszerét átköltöztetnénk a messengerre, mivel sokkal több ember rendelkezik Facebook hozzáféréssel, mint Telegrammal.

Az ember, mint gépezet – Orvostudomány számítástechnikai szemmel nézve

Szvoreny Viktor

Felkészítő tanár: Esztelecki Péter

Bolyai Tehetséggondozó Gimnázium és Kollégium, 24400 Zenta, Posta utca 18.

1. Bevezetés

A mai világban egyre inkább kezd teret nyerni a mérnöki tudományok, az informatika és a számítástechnika. Az első elektromossággal működő programozható számítógép 1943 évben történő megjelenése óta rohamosan hódít magának teret a világháló, és a világ kezd egyre inkább számítógép-centrikusabb lenni. Az emberiség jelentős része használ különböző elektronikus adathordozókat, felhőtárhelyeket, adatbázisokat adataiknak tárolására. A számítógép a népszerűségét több tényezőnek is köszönheti, de mind közül talán a gyorsasága és reprodukcióra alkalmassága a legfontosabb.

Azonban adódik a kérdés: mikor és hogyan fonódott össze ilyen szinten az egészségügy, illetve orvostudomány a mérnöki tudományokkal és informatikával?

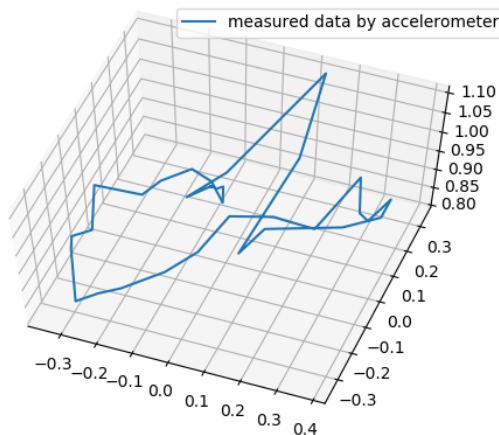
Az egészségügyi informatika definíciója szerint mérnöki tudományok törvényeinek és módszereinek alkalmazását értjük, egészségügyben, biológiában felmerülő problémák megoldásának céljából. Ez magába foglalja az elektromossági, mechanikai, kémiai és élettani törvényeket, új egészségügyi technológiák és rendszerek fejlesztése céljából, és annak érdekében, hogy többet megtudjunk az emberi élet mechanizmusairól. Ez azt jelenti, hogy az évtizedek során nem csupán a munkakörnyezetnek volt szükséges fejlődnie, de természetesen ez is lényeges folyamat, hanem az alkalmazott módszereknek, látásmódoknak is. Mivel komplex és élő rendszerekről van szó, nem minden esetben létezik egy olyan eljárás, amelyik minden témában megoldásra tud vezetni. Azonban módszertanilag általánosan két fajta szemlélet létezik: az analitikai és a kísérleti, ezek nem állnak messze egymástól, inkább kiegészítik egymást. Ezek segítik elő az algoritmusok és az elméletek megfelelő tervezését, melyek alapján pedig létrehozható, megformálható maga az innovatív eszköz. Eközben több hasonló területet is érint a folyamat és több tudományág és hatással van a készülő termékre.

2. Probléma megoldásának menete

A pályamunka során szerettem volna foglalkozni hardverek programozásával és elektronikával, továbbá nagy sportkedvelőnek vallom magam. Így egy olyan eszközt terveztem készíteni, mely leköveti az ember mozgását, így az pontosan követhetővé válik.

E célból gyorsulásmérő szenzor került alkalmazásra. Több lehetőség is szóba került, végül az LSM9DS1 chipes érzékelőre esett a választás, mely sokrétű, hiszen háromféle 3 dimenziós mozgást képes rögzíteni: magnetométerként, gyorsulásmérőként és giroszkópként is funkcionál. Ezekből a gyorsulásmérő létfontosságú, hogy a pozíció lokalizálható legyen XYZ-koordinátarendszerben. Továbbá, szükséges még a giroszkóp, hogy az egységnyi idő alatt megtörtént fokváltozás megjeleníthető legyen a mozgás irányának pontos meghatározásához. Igény szerint változtatható a szenzorok érzékenysége, hiszen vannak sportok, melyek különösebb pontosságot igényelnek. Ezenfelül, elengedhetetlen egy mikroprocesszor, amely fogadja és lekezelet az adatokat. Esetemben ez egy ATmega328P. Az adattárolás SD-kártyára történik, mivel nem szükséges jelen idejű adatfeldolgozás.

Az adatok feldolgozása Python programnyelv segítségével megy végbe, mert egyúttal a grafikus ábrázolás is megvalósítható. A vizuális megjelenítéshez a Matplotlib könyvtár használom, mely a testreszabhatósága miatt lehet ideális választás.

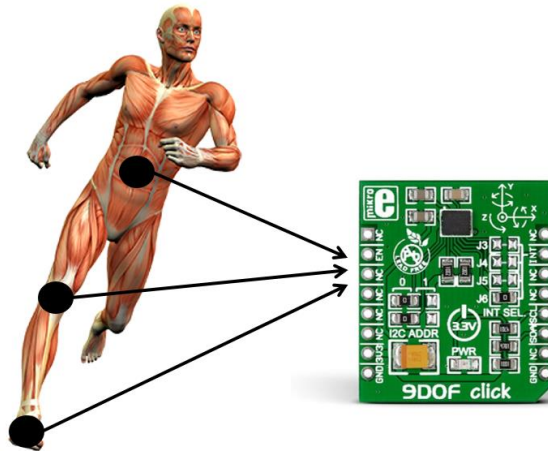


1. ábra: A nyers adatok ábrázolása

A folyamat teljesen offline, de a későbbiekben bevezetésre fog kerülni az online adatbázis is, hogy a mért adatok bárhol elérhetőek legyenek.

2.1. A mérés

Az eszközt rá kell erősíteni a mérni kívánt testrésze. A sporttól és végtagtól függően lehet több szenzort is használni. A sporttevékenység után a mért adatok a megírt programmal megfelelően kielemezhetőek.



2. ábra: Lehetséges rögzítési pontok

3. Elért eredmények

A pályamunka elkészítése ütemezett munkaterv szerint halad a kezdetleges verziótól a prototípusig. Jelenleg az eszköz szenzorjai működnek és mérnek, azonban esetenként előfordulhat feszültségingadozás, melyre a megfelelő kondenzátor hiánya lehet a magyarázat, ezért az rövidesen felkerül a készülékre. A program jelenleg képes kimutatni a mért adatokat, azonban ez nem áttekinthető teljes mértékben, tehát szükséges finomítani a grafikon beállításain ezt illetően. Mivel sporttevékenység közbeni aktivitásnak a rögzítése az elsődleges cél, így fontos az eszköz mérete, ezért a végleges típusig az ATmega328P mikroprocesszor lecserélésre kerül mérete miatt.

A sportolás mellett, az eszköz használható még az egészségügyben is különböző mozgáskoordinációt érintő betegségek kimutatására, de akár a szórakoztatóiparban pl. animációk elkészítésénél.

A jövőbeli tervek között szerepel még az is, hogy az eszköz mérje az adott személy fiziológiai állapotát is, ezzel tovább segítve az edzésterv megfelelő kialakítását.

Vetőgép 2.0

Fajka Viktor

Felkészítő tanár: Kőrösi Gábor

Bolyai Tehetséggondozó Gimnázium és Kollégium, 24400 Zenta, Posta utca 18.

1. Bevezetés

Első sorban végeztem egy kutatómunkát kis falumban, ahol a lakosság 90%-a földműveléssel foglalkozik.

Először a saját gazdaságunk hiányosságairól érdeklődtem. Mivel fejleszthetnék? És hasonló ehhez a témához kapcsolódó kérdéseket tettem fel. Legtöbb válasz az volt, hogy a vetőgép nem a legpontosabb, de az új, okos gép nagyon drága. Ekkor jött az ötlet, hogy építhetnék egy saját, okos vetőgépet.

2. Működése

Legfőbb szerepe ennek a szerkezetnek az, hogy figyelje a magokat. Oda vessük el ahová mi szeretnénk, ne ahová a gép akarja.

Mindössze egy arduino mega+kijelzőből, arduino nanoból, 4x-es billentyűzetből és szenzorokból áll.

Minden csőben van egy-egy szenzor. Ha mag esik le előttük, akkor jelet küldenek az arduino nanonak, ami feldolgozza, majd egy adott időközönként továbbítja a megának (azért van a nanora szükség, hogy a mega ne legyen annyira terhelve). A földműves megadja, hogy hány százalékkal térhet el egy-egy sorban esett mag az átlagtól. Vagyis számol egy átlagot majd megnézi, hogy valamelyik sorban több vagy kevesebb mag esett-e le mint amennyit a gazda meghatározott. Ha az eltérés nagyobb a megengedettnél, akkor átugrik egy másik oldalra és jelzi, hogy melyik sorban van a hiba. De lehet, hogy arányosan esnek a magok, viszont a vetőgép nem halad, ezért szereltem még egy szenzort a kerékre amivel mérem a távolságot, és ha egy adott távon több, vagy kevesebb mag esik mint amennyi kell, akkor is hibát jelez. A földműves még az ültetés elején azt is meg kell, hogy adja, hogy mekkora távolságra szeretné a magokat ültetni, és mekkora a kerék sugara.

Amikor a gazda az adott parcellán befejezte a vetést megnézheti, hogy hány magot ültetett el.

3. Elért eredmények

Számomra elég nagy eredménynek számít az, hogy megépítettem saját, okos vetőgépet - amit forgalmazni is szeretnék, - s evvel hozzájárulhattam falum gazdaságának fejlődéséhez.

Okos ház

Berendi Bence

Felkészítő tanár: Hegedüs Tamás

Tatabányai Szakképzési Centrum Széchenyi István Közgazdasági és Informatikai Szakgimnáziuma, 2900 Komárom, Táncsics Mihály utca 75.

1. Bevezetés

11. évfolyamos tanulóként szeretnék bekapcsolódni a műszaki informatika folyamatosan fejlődő innovatív világába.

A mai világban az embereknek fontos, hogy otthonukban teljes kényelemben éljenek, és minél kevesebb energia befektetéssel élvezhessék otthon eltöltött idejüket.

Célomnak egy olyan vezérlés megépítését tűztem ki, amely a külső körülmények alapján biztosít egy automata vezérlést, valamint egy olyan webes felület biztosítását, ahol a felhasználó kényelmesen, akár mobiltelefonjáról is tudja vezérelni a házat. A megépített modellház az ismeretlenek számára egy egyszerű babaháznak tűnhet, azonban aki jobban megnézi, az rögtön rájön, hogy itt többről van szó.

2. Probléma megoldásának menete

Az első, egyben legfontosabb feladat a vezérlést végző mikrovezérlő kiválasztása volt. Az én választásom több szempont észben tartásával az Arduino Mega 2560-as mikrovezérlőre esett. Főként azért, mivel összesen 54 darab digitális be- és kimenete van, amiből 15 PWM kimenet, valamint 16 analóg bemenettel is rendelkezik, ezáltal az esetleges későbbi fejlesztések során is elegendő lesz. Hasznos funkciója - amit kihasználtam -, hogy hardveres I2C buszrendszerrel rendelkezik. Az Arduino panelt továbbá felszereltem egy Ethernet shield-del, ami a webes felület szervereként szolgál. Az Arduino Mega 2560 az Ethernet shield-del felszerelve az 1. ábrán látható.



1. ábra: Arduino Mega 2560 mikrovezérlő Ethernet shield-del felszerelve

Az Arduino panelhoz továbbá csatlakoztattam kettő DHT11-es hőmérséklet és páratartalom mérő szenzort, melyek a külső és belső hőmérséklet mérését hivatottak ellátni, egy esőérzékelőt, egy fénymérő szenzort, ami a külső fényt méri, egy talajnedvesség mérő szenzort, ami a növényeink talajának nedvességét vizsgálja, valamint 8 relét, amik arra szolgálnak, hogy kapcsolhassuk vele a hálózati berendezéseinket. Ezek az eszközök a 2. ábrán láthatók.



2. ábra: DHT11 szenzor; esőérzékelő; fénymérő szenzor; talajnedvesség mérő szenzor; két csatornás relé modul

Elkészítettem egy modellházat, amin elhelyeztem a fényforrásokat, a szenzorokat, valamint elkészítettem a nyílászárókat: három ablakot, egy ajtót és egy padlásajtót. Ezek nyitását illetve zárását SG90-es szervomotorok végzik. A modellház a 3. ábrán látható. Mivel a szellőztetéshez szükség volt időzítésre, ezért egy DS3231-es RTC modult is összekapcsoltam az Arduino-val. Ez azért volt fontos, mivel ha csak egy várakozást írok a programba, akkor a szellőztetés idejére „megbénul” a ház, azonban ha csak vizsgáljuk, hogy eltelik-e „x” idő, akkor nem szükséges várakozás a programban. Az elektronikus eszközöket és a kábelezést a modellház hátulján helyeztem el. További biztonsági funkcióként elhelyeztem egy MQ-4-es gázérzékelő szenzort, ami gázszivárgás esetén kiszellőzteti a házat, és amíg túl sok a gáz a levegőben, addig villogtatja a lámpákat is. Az SG90-es szervomotor, a DS3231-es RTC modul, a kábelezés és az MQ-4-es gázérzékelő a 4. ábrán látható.



3. ábra: A modellház



4. ábra: SG90-es szervomotor; DS3231-es RTC modul; a ház hátulján elhelyezett kábelezés; MQ-4-es gázérzékelő

Ha zárva van a ház, akkor biztonsági okokból nem tanácsos mobiltelefonról nyitni az ajtót, ezért az ajtó mellett elhelyeztem a numerikus billentyűzetet – ez az 5. ábrán látható -, amin meg lehet adni a jelszót, és így csak helyes jelszó után nyit ajtót.

3. Elért eredmények

A felhasznált szenzoroknak és eszközöknek köszönhetően hasznos automata funkciókat sikerült elérnem. Az egyik leghasznosabb, hogy ha zárt állásban van a ház, akkor bizonyos időnként nappal szellőztet a szellőztető rendszer segítségével, amit a modellen egy ventilátorral modelleztem. A szellőztetésen kívül automatikusan meglocsolja virágainkat egy szivattyú segítségével, ha száraz a földjük. A szivattyú és a ház oldalán elhelyezett numerikus billentyűzet az 5. ábrán látható.



5. ábra: A locsolást végző szivattyú; a ház oldalán elhelyezett numerikus billentyűzet

További biztonsági funkció, hogy sötétben automatikusan felkapcsolja a külső világítást, ezáltal ha valaki arra jár, kisebb eséllyel esik el, ugyanakkor az energiatakarékos fényforrás biztosítja a gazdaságos felhasználást. Ha a ház nyitott állásba kerül, akkor az automatikus funkciók listája bővül. A szellőztetést ilyenkor a külső hatásokat figyelembe véve végzi, tehát ha bizonyos hőmérséklet felett van a külső hőmérséklet, és nem esik az eső, valamint nappal van, akkor az ablakon szellőztet adott ideig, ha viszont valamelyik feltétel nem felel meg, akkor a szellőztető rendszert használja. További kényelmi szempont továbbá, hogy nyitott állásban az összes belső világítást is automatikusan kapcsolja. A ház felkapcsolt lámpákkal a 7. ábrán látható.

Robot TÉP Szakkör a Szikszi 100-ban

Robot TÉP

Magyar Dávid, Felföldi János, Vitkos Bence

Felkészítő tanár: Bakti András Tamás

*Nyíregyházi SzC Széchenyi István Közgazdasági, Informatikai Szakgimnáziuma
és Kollégiuma, 4400 Nyíregyháza, Városmajor u. 4.*

1. Bevezetés

Az iskolánkban 2017 novemberében indult a Robot TÉP (Tervező, Építő, Programozó) Szakkör. Miután a tanár úr elmondta a szegedi programozó matematikus és a béka esetéről szóló viccet, (poén: Nem csókollak meg, mert én programozó vagyok, nekem nincs időm, a nőkre... de azért egy beszélő béka az nem semmi!) eldöntöttük, hogy csinálunk egy beszélő békát. (Hőmérséklet, páratartalom, fényerősség szenzor + Raspberry Pi + 3D nyomtatóval nyomtatott béka forma. Bekapcsoláskor közli a mért értékeket.)

Az innovatív megoldás abban rejlik, hogy tervezzük meg 3D-ben, nyomtassuk ki 3D-ben, szereljük össze és programozzuk a működését. Ezáltal sokkal kreatívabb tevékenységet végzünk, mintha boltban megvevünk kapható robotot, és felprogramozzuk. Ezzel többféle területen is fejlesztjük magunkat: tervezés, nyomtatás, (újratervezés), szerelés, programozás. A végeredmény pedig bár hosszabb idő alatt készül el és fáradtságosabb munkával, de sokkal nagyobb meglepedést jelent.

Mivel már háromkerekű ultrahangos távolságszenzorral működő rovert építettünk részben az ebay-en vásárolt alkatrészekből, adott volt a felvetés, hogy készítsünk 4WD-s autót, amit szintén Raspberry vezérel. A kitűzött cél, az volt, hogy szereljünk rá kamerát, és a kamera által érzékelt kép alapján döntse el, merre menjen. (Jobbra nyílra jobbra fordul, balra nyílra balra fordul, stb.) Természetesen most is felhasználunk vásárolt alkatrészeket, (motorok + kerekek, motorvezérlő, Raspberry Zero W, akkumulátorok, stb.) de kiegészítve az általunk tervezett és nyomtatott karosszéria elemekkel. Ezenkívül mivel idén 100 éves az iskolánk, úgy tervezzük, hogy az ebből az apropóból tervezett rendezvényeken is szeretnénk majd bemutatni a robotokat.

2. Probléma megoldásának menete

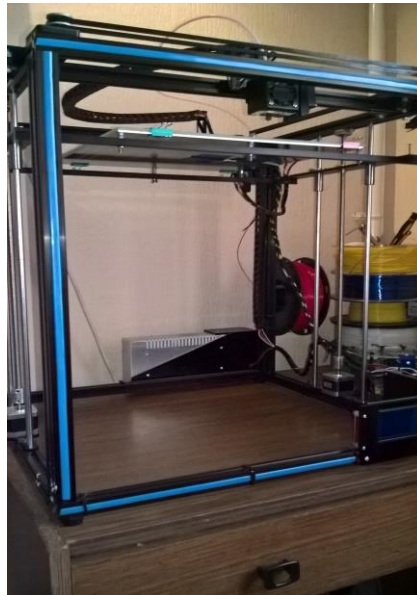
Az alkatrészek összegyűjtése már korábban elkezdődött (saját forrásból). A motorok, kerekek, vezérlő, a kamera, és egyéb mechanikai alkatrészek rendelkezésre álltak. Az összeszerelés nem okozott problémát, bár sok új ismeret megszerzéséhez segített hozzá minket. mi problémás volt, az a

karosszéria elemek tervezése, nyomtatása. Részben a tapasztalat hiánya miatt, részben a nyomtató hiányosságai miatt. Eredetileg ABS, vagy PETG anyagból szeretnénk volna elkészíteni, de a nyomtató, mint utóbb kiderült nem tudja felfűteni a tálcáját 50°C-nál magasabb hőmérsékletre, ezért át kellett térni a PLA-ra. Alacsonyabb tálca hőmérsékleten nem tapad le az ABS és a PETG filament, akkor sem, ha hajlakkot, Tesa szalagot, vagy kenőfejes ragasztót használunk segítségként. A PLA-nak rosszabbak a mechanikai tulajdonságai, viszont nem igényli a tálca fűtését, vagy elég a 45°C-os hőmérséklet.

Miután áttértünk a PLA-ra, sikerült különböző tárgyakat nyomtatni, például kacsacsőrű emlőst, Minion figurát, franciakulcsot, 4x4-es kocka kirakót, valamint lithophane-t. Ezen tárgyak nyomtatása során szerzett tapasztalatok segítettek az alkatrészek kinyomtatását. Innentől kezdve csak a fantáziánk, és az idő szab határt annak, hogy mit készítünk a nyomtatóval.

2.1. Ábrák

A nyomtatóról készült fénykép az 1. ábrán látható. Típusa: Tronxy X5S. Nyomtatási tartománya 330x330x400 mm. A fűvóka átmérője 0,4 mm, a használható filament átmérője: 1,75 mm.



1. ábra: A nyomtató (Tronxy X5S)

A 4WD autóról készült fénykép a 2. ábrán látható. Itt még várja a különböző karosszéria kiegészítő elemeket.



2. ábra: A félkész négy-kerék hajtású autó

2.2. A programozás

A programozás területén az jelentett kihívást, hogy az iskolánkban a Java programozást preferálják, a Raspberry-t viszont egyszerűbb Pythonban programozni. A korábban megvalósított átalakított távirányításos autó programozása (mobiltelefonnal rácsatlakozva a wifijére irányítható webes felületen keresztül), valamint a szintén korábban készített ultrahangos távolságszenzorral működő háromkerékű programozása bizakodásra adott okot. Persze a kamera plusz, a képfeldolgozás plusz, szóval volt kihívás... Ezenkívül a Moto Zero vezérlőt sem használtuk eddig, mivel az L298N vezérlő elegendő volt a két motorhoz. Ez azonban 4 db motor vezérlésére alkalmas, ezért esett erre a választásunk.

A Raspberry ZeroW-ről és a MotoZero-ról készült fénykép a 3. ábrán látható.



3. ábra: A Raspberry Zero és a Moto Zero

3. Elért eredmények

A cél elérése közben sikerült olyan problémákat megoldanunk, amelyek eszünkbe sem jutottak a tervezés fázisában.

Például a mechanika összeszerelésekor meg kellett tanulnunk, hogy hogyan lehet menetet fúrni plexibe, távtartót szerelni, az alkatrészeket rögzíteni. Ezenkívül végül sikerült elfogadható minőségben nyomtatni a 3D nyomtatóval, de egyes alkatrészeket újra kellett tervezni, és nyomtatni. Sok ismeretet szereztünk az elektronikával kapcsolatban is, a vezérlő működésének megértése, bekötése területén. A kamera segítségével közvetített kép felhasználása újabb távlatokat nyitott a felhasználhatóság területén. Persze az, hogy a Tesla konkurenciája legyünk és önvezető autót építsünk, az csak távlati cél lehet számunkra. :-)

A projekt során megszerzett tapasztalatok segítették a későbbi terveink megvalósítását. Szeretnénk szervó motorokkal működő lépegető robotokat készíteni PCA9685 16 csatornás vezérlővel. Kétlábú, négy lábú, hatlábú, valamint humanoid stílusú robotot szeretnénk tervezni, építeni, programozni. Reméljük, hogy a pályamunkánk kedvező elbírálásban részesül, és lesz lehetőségünk bemutatni. Valamint a verseny során szerzett tapasztalatokkal további innovatív megoldások megvalósításán dolgozni.

Végül egy kép amelyen Magyar Dávid egy iskola rendezvényen bemutatja a Robot TÉP Szakkört a 4. ábrán látható.



4. ábra: A Robot TÉP bemutatása a szakmai napon

„Csepp” kilövő HUSAR rover

BluelsBetter

Bejő Mátyás, Joób Zalán, Molnár Barnabás

Felkészítő tanár: Lang Ágota

Soproni Széchenyi István Gimnázium, 9400 Sopron Templom utca 26.

1. Az ötlet alapja: egy új űrkutatási technológia

Csapatunk a Bérczi Szaniszló (ELTE) által vezetett HunVeyor-HUSAR projekt keretében HUSAR-okat, vagyis „(Hungarian University) Surface Analiser Rover”-eket épít és programoz. A roverek alapja egy távirányítós terepjáró modell, de a „végtermék” már önállóan lenne képes ténykedni egy bolygó felszínén és ott méréseket, vizsgálatokat végezni.

Míg elődeink roverjei maguk végezték a méréseket a rajtuk elhelyezett szenzorok segítségével, addig mi egy új technika mentén indultunk el. Ennek alapját az úgynevezett MSSM (Micro Sized Space- Mothership) illetve NPSDR (Nano, Pico Space Devices and Robots) módszerek képezik, amelyek a nevükből is sejtethetően a méretek lecsökkentését tűzik ki célul (azonban a prefixumok nem a szó szoros értelmében vett méretet jelölnék).



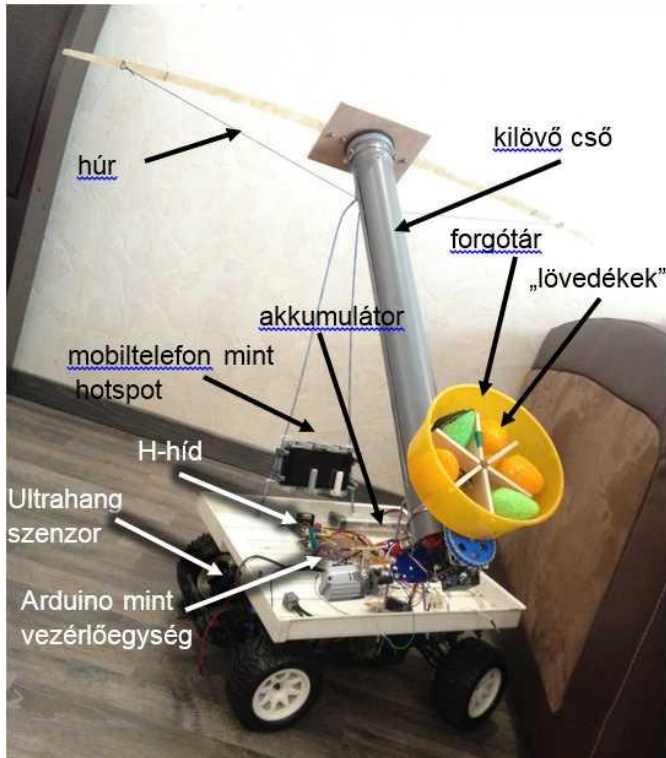
1. ábra: Illusztráció az MSSM és NPSDR technikához

Ennél fontosabb, hogy ezek az eszközök már olcsón előállíthatóak, így megéri nagy mennyiségben szétszórni őket pl. egy bolygó felszínén egy anyaűrhajóról, amellyel kapcsolatban maradnak és küldik neki az adatokat (1. ábra). Ily módon nagy területről szerezhető információ a szenzorok által küldött értékek alapján.

Mivel egyelőre olyan járművet nem tudunk építeni, amellyel fentről szórhatnánk szét a szenzorokat tartalmazó kis „kockákat”, ezért a roverünket tettük erre alkalmassá.

2. Megvalósítás

2.1. A rover felépítése



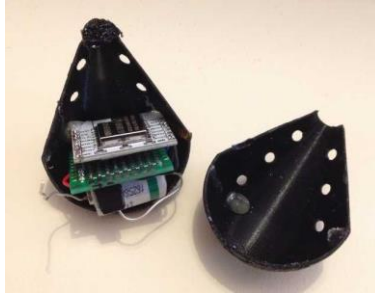
2. ábra: A HUSAR-rover

Ahogy a 2. ábrán láthatjuk, a kilövéshez egy nyílpuska elvén működő eszközt alkalmaztunk. A „lövedékek” 5 cm átmérőjű csepp alakok (a továbbiakban cseppek), amelyek a forgótárban helyezkednek el. Ezt egy szervomotor forgatja a cső megfelelő helyére, ahol egy nyílason bele tud esni. Majd egy másik motor felhúzza az íjat, és egy kioldással a csepp kirepül a csőből.

A rovert egy Arduino vezérli. Ez irányítja a mozgását, figyelembe véve az ultrahang szenzortól kapott adatokat; a mozgatáshoz egy H-híd vezérlőt építettünk be. A program meghatározott időközönként megállítja a rovert illetve az íjat felhúzza. Ez a művelet viszonylag lassú, ezért menet közben végzi. Miután a cseppet kilőtte, folytatja mozgását.

Az akkumulátor feszültsége $-11,2\text{ V}$ – nagyobb, mint amit az Arduino illetve a legtöbb elektronika kibír, ezért DC-DC átalakítót iktattunk be, amelynek segítségével a feszültség a kívánt értékre csökkenthető.

2.2. A cseppek felépítése

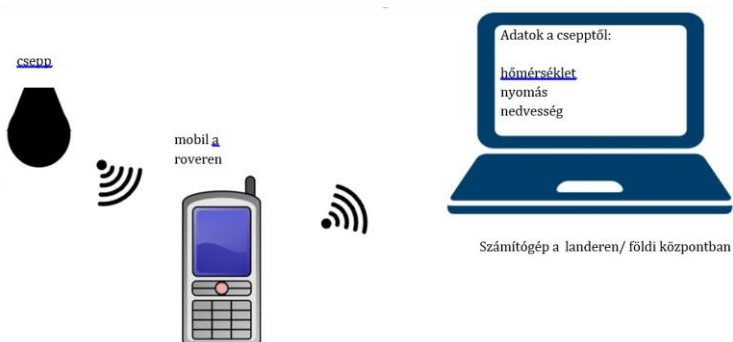


3. ábra: A csepp belseje

A cseppben kapott helyet a szenzor, egy ESP modul és egy kisméretű akkumulátor. A csepp „agya” az ESP-8266, amelyet Arduino nyelven tudunk programozni. Azért esett erre a választás, mert képes wifit létrehozni magának és azon keresztül kommunikálni. Ez teszi lehetővé, hogy a szenzor adatai eljussanak egy központi számítógéphez (pl. a landeren, és az továbbítja a földi irányítóközpontba). Szenzorként BME/BMP-280 típust használunk, amely egy kombinált érzékelő, egyidejűleg mérhetünk vele légnyomást, hőmérsékletet és nedvességet. A csepp falát lukakkal láttuk el, hogy ténylegesen a külső levegő értékeit mérje. Maga a csepp 3D nyomtatással készült, a szükséges szoftvert egy ebben jártas diáktársunk programozta.

2.3. A kommunikáció megvalósítása

Első gondolatunk egy mesh hálózat kialakítása volt, hiszen a szenzorok mérési értékeire nem egyszeri alkalommal van szükségünk, hanem folyamatosan. Ezt úgy szerettük volna elérni, hogy a cseppek riadólánc-szerűen adják volna át egymásnak az adatokat, így jutott volna el a roverhez.



4. ábra: Kommunikáció a csepp és egy számítógép között

Ezt azonban nem sikerült megvalósítani, mert olyan programozási ismereteket igényelt, amivel jelenleg még nem rendelkezünk. Ezért végül – több próbálkozás után – egy szerver-kliens rendszert alkalmaztunk, ahol a cseppekben lévő ESP-k voltak a szerverek, míg kliens egy laptop volt. Utóbbi egy hotspoton keresztül tudott kapcsolódni az ESP-k hez. (4. ábra) A laptop egy linken keresztül érte el az adatokat, amelyeket ki is irattunk ezen a weboldalon. Az adatok begyűjtését tehát a mozgó rover végzi: miután kilőtte a tárat, ugyanazon az útvonalon furikázik fel-le – így biztos hatótávolságon belül van – és felveszi a kapcsolatot az egyes cseppekkel.

3. Hol tartunk?

A rover mozog és kilövi a cseppeket. Azonban borulékony, és a kioldószerkezet is beragad olykor. Ezek a hardveres problémák, amiket igyekszünk kiküszöbölni. Teljesen működőkész állapotban 2 csepp van, a többi kézi 3D nyomtatással készült, hogy a forgótárat kipróbálhassuk.

Nem adtuk fel a mesh topológiát sem, folyamatosan keressük ennek lehetőségét. Tervezzük az ESP-modulok lecserélését Wemos modulokra, mert ezek megbízhatóbbnak tűnnek mind a program áttöltésekor, mind pedig azok futását illetően.

Végezetül megjegyezzük, hogy bár a rover az első elgondolások szerint egy bolygó felszínén tevékenykedne, de bevethető földi körülmények között is olyan helyeken, ahova ember nem mehet be pl. magas radioaktivitás vagy mérgező gázok miatt.

Források:

https://www.researchgate.net/publication/272941616_APPLICATION_OF_THE_FLEET_OF_MICRO_SIZED_SPACE-MOTHERSHIPS_MSSM_WITH_NANO_PICO_SPACE_DEVICES_AND_ROBOTS_NPSDR_FOR_LIFE_SIGNAL_SEARCH_ON-DDS_SITES_USING_GLOBAL_DIGITAL_DUNE_DATABASE_OF_MARS

Informatika szekció

ConTags

magányosHarco\$

Horváth István

Felkészítő tanár: Esztelecki Péter

Bolyai Tehetséggondozó Gimnázium és Kollégium, 24400 Zenta, Posta utca 18.

1. Bevezetés

Akik gyakran bővítik új névjegyekkel telefonjaik névjegyzékét, sokszor előfordul velük, hogy olyan személy hívja őket, aki a névjegyzékükben szerepel, viszont a képernyőn megjelenő adatok alapján nem ismeri fel, hogy éppen ki hívja. Ha sok új névjegyet adunk hozzá telefonunk névjegyzékéhez egy idő után megtörténik velünk az, hogy nem tudunk köztük különbséget tenni, vagy nem találjuk a kívánt névjegyet. Ez gyakran annak köszönhető, hogy a beépített névjegykezelőkben kevés információt tárolhatunk el egy névjegyről. Ha pedig egy ingyenes letölthető alkalmazással szeretnénk orvosolni a helyzetet csak olyan alkalmazásokat találunk, melyek lassúak vagy tele vannak hirdetésekkel.

2. Hogyan működik a ConTags és milyen megoldásokat találtam a problémákra?

Az előbb említett problémákra a ConTags olyan megoldást talált, hogy egy névjegyhez „tageket” azaz plusz tulajdonságokat is hozzáadhatunk. Egy névjegyhez bármennyi taget hozzáadhatunk, ami bármilyen szöveget tartalmazhat. Így egyszerűen megtalálhatjuk névjegyünket akkor is, ha több ezer közül kell keresnünk. Amikor először a ConTagsben kereshetünk a névjegyek között bármilyen megadott információ alapján, sőt akkor is megtalálhatjuk a kívánt névjegyet, ha a keresőbe beírt szöveget csak tartalmazza a találat. Híváskor, ha a névjegy szerepel a ConTags névjegylistájába az összes megadott adat, az utolsó hívás dátuma és az utolsó a hívó személytől érkezett üzenet megjelenik egy listán, így sokkal könnyebb dolgunk van, ha kevésbé ismerjük azt a bizonyos személyt és rég beszéltünk vele

3. Az alkalmazás funkciói

3.1. Szinkronizálás a meglévő névjegyekkel

A ConTags lehetőséget nyújt az első indításkor két gombnyomással az összes telefonban szereplő névjegyet hozzáadni a ConTags névjegylistájához. Ezt a

funkciót később is elérhetjük a beállításokból. A már hozzáadott telefonszámokat már nem fogja megjeleníteni a hibák elkerülése érdekében.

Válasszon névjegyeket

ÖSSZES KIJELÖLÉSE

Név: János
Telefonszám: 046757878

Név: Béla
Telefonszám: 97876478

Név: Éva
Telefonszám: 97673787583

MÉGSEM IGEN

1. ábra: A meglévő névjegyek hozzáadása

3.2. Új névjegy és névjegyek szerkesztés

Az alkalmazásban adhatunk hozzá új névjegyeket és szerkeszthetjük őket. Ilyenkor adhatjuk hozzá a tageket is, amik bármilyen szöveget tartalmaznak. Egy névjegyhez hozzárendelhetünk bármennyi taget.

Keresztnév

Becenév

Utónév

E-mail cím

Telefonszám

MENTÉS #Tag

TAGEK HOZZÁADÁSA

2. ábra: Új névjegy hozzáadása vagy szerkesztése

3.3. Tagek hozzáadása egy névjegyhez

Névjegyek hozzáadásakor vagy szerkesztésekor lehetőségünk van új taget hozzáadni emellett a régebben hozzáadott szövegeket is hozzárendelhetjük egy névjegyhez. Mindezt egy felugró lista segítségével tehetjük meg.

Tagek

android app

bolyai

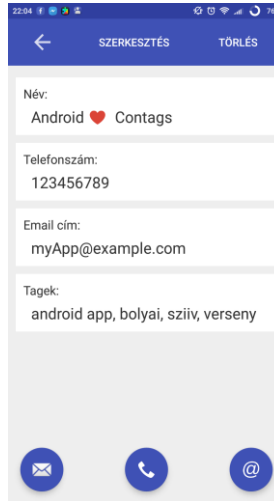
sziiiv

verseny

3. ábra: Tagek kiválasztása

3.4. Műveletek egy névjeggyel

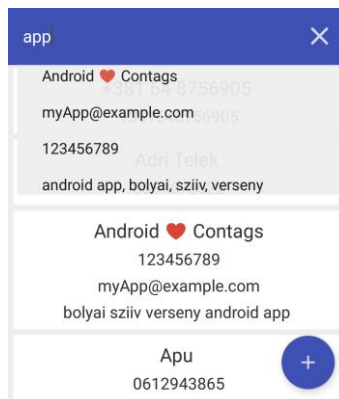
Mint bármelyik névjegykezelőben a ConTagsban is a megadott adatoktól függően minden általános funkció elérhető (Hívás, SMS-küldése, E-mail írása), a beépített alkalmazásokkal együttműködve.



4. ábra: Névjegyek megtekintése és műveletek elvégzése

3.5. Keresés a névjegyek között

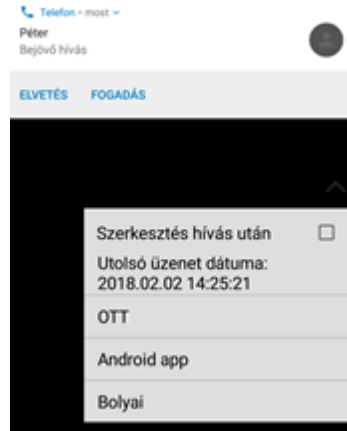
Kereséskor bármilyen megadott szövegben kereshetünk, viszont szűkíthetjük a keresés területét a beállításokban (Név, E-mail cím, telefonszám vagy tagek alapján). A találatok egy listán jelennek meg, először csak a felhasználó neve, viszont ha rákattintunk a listára, akkor minden adat megjelenik a névjegyről.



5. ábra: Keresés a névjegyek között

3.6. Bejövő hívás

Bejövő híváskor, ha ismert a névjegy egy gomb jelenik meg a képernyőn. Ha a gombra rákattintunk minden megadott információ megjelenik az adott névjegyről többek közt megjelenik az utolsó bejövő üzenet a személytől és az utolsó hívás dátuma. Hívás után az alkalmazás megkérdezi, hogy szeretnénk-e szerkeszteni, vagy ha ismeretlen a névjegy hozzáadni a névjegylistához.



6. ábra: A bejövő híváskor megjelenő lista

4. Az alkalmazásról

Az alkalmazás Android Studio-ban főként Java nyelvben íródott viszont szerepel egy kevés XML és SQLite is a kódban. A forráskód több mint 6000 sor kódot tartalmaz. Az alkalmazás kevés helyet foglal több ezer névjeggyel is. Kevés erőforrást igényel futás közben és nem terheli az akkumulátort. Android 4.0-tól támogatott minden androidos okostelefonon.

5. Miért válassza a ConTagsot?

Az alkalmazás nem tartalmaz reklámokat és a felhasználtól függően nem osztja meg harmadik féllel az adatokat (A lista megjelenését le lehet tiltani). Az alkalmazás az ismerőseim használják és megvannak vele elégedve. A ConTags egy megbízható asszisztens a hívásokkor a mindennapokban.

MySchedule

HungaroCode

Ignác Dominik Bence, Partovits Levente, Salamon Donát, Bitai Bence

Felkészítő tanár: Drávucz Katalin

*BMSzC Neumann János Számítástechnikai Szakgimnáziuma,
1144 Budapest, Kerepesi út 124*

1. Bevezetés

A felgyorsult világunkban az embereknek nagyon fontos az időbeosztás. Ez nemcsak a cégek vezetőire igaz, de az egyszerű diákoknak is nagy szerepet játszik az életében. Olyan programot csináltunk, ami ennek a problémának a megoldását automatizálja, így felhasználója időt takarít meg a manuális naptár rendezéshez viszonyítva.

2. Probléma megoldásának menete

A program célja, hogy minimalizálja azt az időt, amit napi feladataink ütemezésével kell tölteni. Különböző eseményeket, határidőket a felhasználóhoz igazodva rendezi el, valamint képes ezt a Google Naptár rendszerével szinkronizálni is.

2.1. Funkciók

Az alkalmazás főbb funkciói:

- Események importálása a Google Naptárból
- Események exportálása a Google Naptárba
- Események zárolása (Pl.: órarend, üzleti megbeszélés)
- Prioritások beállítása
- Napirend kezdeti és befejezési időpontjának beállítása (Pl.: 6:00 és 21:00 között legyenek csak események)
- Események közötti pihenő megadása
- Feladatok több részre való bontása

2.2. Események bevitele

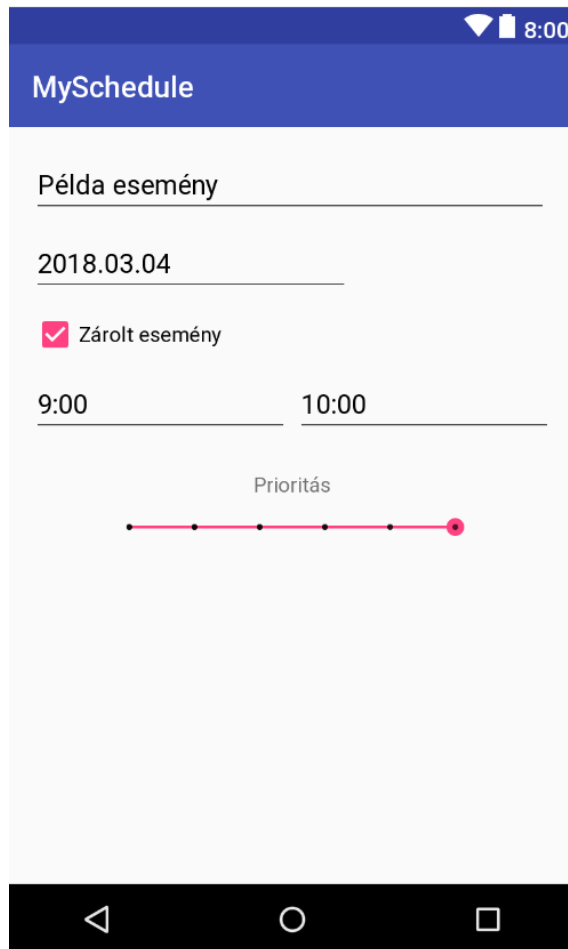
Az alkalmazásba kétféleképp vihetünk be eseményeket. Az egyik módszerrel kézilleg lehet megadni az esemény adatait. A másik lehetőség, hogy csatlakozva Google fiókunkhoz a naptárat betöltjük.



1. ábra: Az alkalmazás kezdőlapja

2.3. Elrendezés

A bevitel után a szoftverben megadhatjuk a rendezési szempontokat. Az egyes eseményeinket zárolhatjuk egy adott időponttal, így a rendezés során ezek az időpontok nem fognak változni, emellett lehetőségünk van megadni minden eseménynek (kivéve a zároltakat) egy prioritást, amely a rendezés alapját adja. A feladatoknál megadhatjuk, hogy egyszerre kívánjuk-e elvégezni azt vagy több részre bontva. Ha több részre bontjuk, akkor megadhatjuk, hogy ugyanazon a napon kisebb pihenőkkel vagy esetleg több nap alatt szeretnénk megcsinálni. Beállíthatjuk ezek mellett az is, hogy mikor kezdődjön a napirendünk, illetve meddig tartson. Ekkor a program ehhez igazodva csak ezen az időtartamon belül helyezi el az eseményeket.



2. ábra: Az esemény módosító felülete

2.4. Az elrendezés technikai háttere

A szoftver elsődlegesen mindig a zárt eseményeket nézi, mivel ezek időpontját nem változtathatja. Majd ezek után a megadott felbontható események részekre bontását elvégzi a program, így több azonos tulajdonságú eseményként kezeli tovább az alkalmazás. Miután kialakult a végleges eseménytömb, megkezdődik a prioritások szerinti sorba rendezés. Egy adott prioritáson belül véletlenszerű elrendezést alkalmaz, ezzel lehetővé téve több kombináció előállítását. Ha nem tetszik egyik kombináció sem, akkor pedig átállíthatjuk bármely esemény prioritását. Ekkor már csak az előre megadott pihenőidőket kell hozzáadni, hogy elkészüljön a végleges sorrend.

2.5. Exportálás, avagy a naptár használata

Az elkészült napirendünket exportálhatjuk a Google Naptárba. Választhatunk, hogy az eseményeket másoljuk vagy áthelyezzük azokat. Az előbbi esetben új eseményeket hoz létre a naptárban a meglévők mellé, így az importálás előttiiek is láthatóak. Míg az áthelyezés esetén a naptárban levő eseményeket felülírja az alkalmazás.

3. Elért eredmények

A szoftver használatával a felhasználó jobban be tudja osztani az idejét és így hatékonyabban tudja a munkáját elvégezni, sőt még pihenésre is lesz lehetősége. A program felajánlhat többféle napirendet, amelyekből kiválaszthatjuk a számunkra legmegfelelőbbet. Többen is kipróbálták alkalmazásunkat és elégedettek voltak, mert kevesebb időt kellett tölteni a napirendkészítéssel, és csökkent az üresjáratok hossza, amit a rossz szervezés miatt semmire se lehetett kihasználni.

e-Gradu

SootSoft

Korom Richárd

Felkészítő tanár: Fási Szilvia, Lukács Magdolna

*Gyulai SzC Kossuth Lajos Szakgimnáziuma, Szakközépiskolája és Kollégiuma
5900. Orosháza, Kossuth tér 1.*

1. Bevezetés

A mai kultúra tudás alapú társadalomra épül, melyben egyre inkább előtérbe került és felértékelődött az informatikai eszközök használata. Tanárainkkal és diáktársaimmal beszélgetve arra a következtetésre jutottam, hogy a tananyagtartalom növekedése és az idő szűkössége miatt a tanulók többsége a képességeihez mérten alul teljesít. A fent említett tényezőket figyelembe véve álmodtam meg egy olyan tanulást segítő szoftvert, amely időtől és helytől függetlenül a tanulók rendelkezésére áll. A program segítségével a diákok a hagyományos papír alapú dolgozatok helyett digitális tesztek, jegyzeteket illetve esszéket készítenek, mellyel a környezettudatosság kialakítását is figyelembe vettem, hiszen általa csökken a papírhulladék mennyisége.

A tervezésénél három alapelvet fogalmaztam meg:

- A tanulói teljesítmény támogatása, megsegítése
- Az időhatékonyság optimalizálása
- A tanári munka leegyszerűsítése

2. A probléma megoldásának menete

A szoftver tervezésének egyik nehezített eleme a „kettősség elve”, amely szerint a programnak egyrészt a tanár elvárásait, másrészt pedig a tanulók gyakorlati jellegű igényeit kell kiszolgálnia, úgy, hogy egyaránt tudja használni mindkét fél a pontosság, a megbízhatóság és a kezelhetőség szellemében.

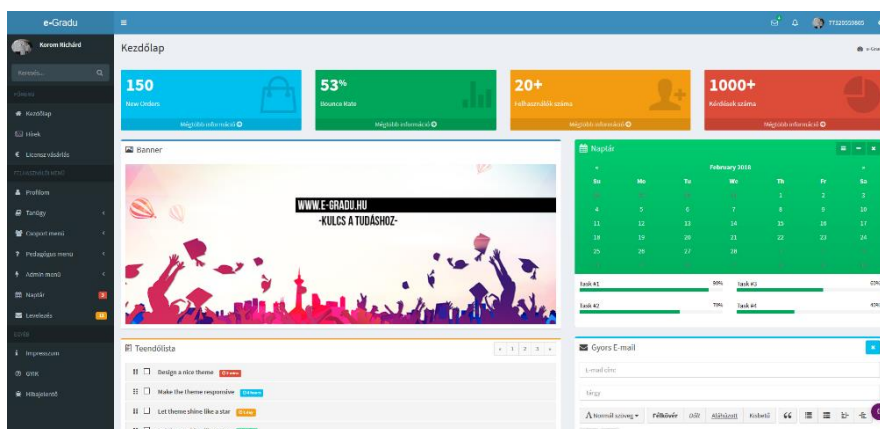
2.1. A szoftver kivitelezése

A fejlesztés során számos különböző programozási nyelvet használtam, így az egyik kihívás, ezek egybefüggő, összehangolt működésének megoldása volt.

A legfontosabb kritérium a biztonság megalapozása és a felhasználók adatainak védelme. Minden egyes beviteli mezőt szigorú ellenőrzésnek vet alá a rendszer, hogy semmiféle behatolás, adatlopás ne történhessen. A kommunikáció kizárólag SSL titkosítással engedélyezett a „lehallgatások” kiküszöbölésének érdekében. Az adatvédelem szempontjából, egyik hasznos

elem az időzár, amely 5 perc inaktivitás után, újbóli jelszó megadását kéri, hogy illetéktelen személy ne tudjon az adott felhasználó fiókjához hozzáférni.

Szintén fontos szempont volt a platformfüggetlenség megteremtése, hogy a szoftver operációs rendszertől függetlenül, bármilyen számítógépen, telefonon, vagy akár egyéb eszközön is elérhető legyen. A fentieket figyelembe véve, egy webalapú alkalmazás elkészítése mellett döntöttem. Külön figyelmet fordítottam a mobil optimalizálásra, így könnyedén áttekinthető bármely készüléken a futtatott program, amit az 1. ábra szemléltet.



1. ábra: Kezdőlap

2.2. A szoftver funkciói

Az elmúlt fél év alatt, közel 10.000 új elem került az adatbázisba, mely tartalmazza a tanárok által felvitt teszt jellegű kérdéseket és a felhasználók által rögzített esszéket, jegyzeteket. A lekérdezések hatékonyságát a maximálisra kellett növelni, a szelektálás erőforrásigényét pedig a minimumra kellett csökkenteni.

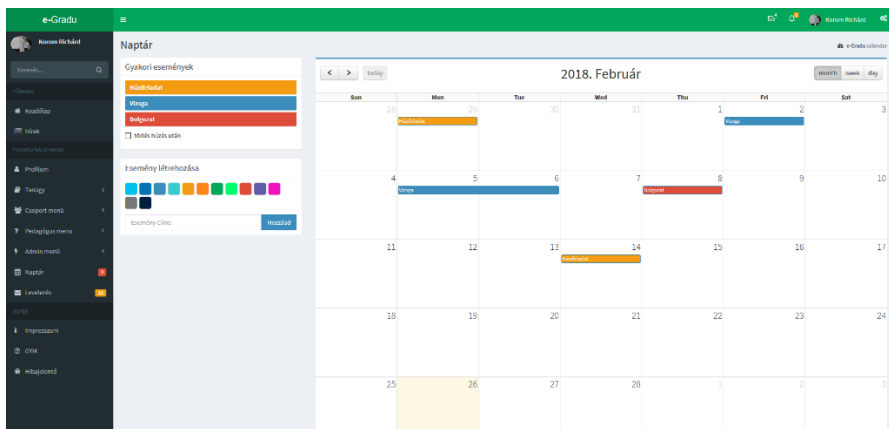
Az érettségi tesztek során a tanulók kiválaszthatják azt a tárgyat, amelyet gyakorolni szeretnének, ezen belül választhatnak egy adott témát, vagy akár egy összesített tesztet témakörből függetlenül. A teszt kitöltése után, a rendszer automatikusan értékeli, majd megmutatja a felhasználó számára a helyes, illetve helytelen válaszokat, ezek arányát, százalékát, a megoldásra fordított időt és egyéb statisztikai adatokat. Ezen felül a hibás válaszok tárolásra kerülnek, amelyeket a következő teszt során a rendszer preferál a tanuló számára, segítve ezzel a helyes ismeretek elsajátítását.

Esszé/fogalmazás/levél készítése során a felhasználóknak lehetősége nyílik egy felületen, hogy tesztelhessék önálló fogalmazási képességüket, majd

az általuk elkészített dokumentumot elküldjék egy szaktanárnak, aki a rendszeren keresztül tudja ezt pontozni. Természetesen az egész folyamat név nélkül zajlik.

Az e-Jegyzet funkció használatával a felhasználóknak, legyen szó akár diákról, akár tanárról, lehetősége nyílik elektronikus dokumentumok készítésére. A különbség egy hasonló programhoz képest az, hogy az itt elkészített jegyzet felhő alapon kerül tárolásra, így bármikor, bármilyen eszközről megtekinthető, szerkeszthető. Személyes beállításként testreszabható, hogy a dokumentum privát, vagy publikus jellegű, így korlátozható a hozzáférés, illetve egy adott tantárgyhoz, témakörhöz is köthető.

A tanulás megkönnyítése érdekében, bevezetésre került a csoport rendszer, mellyel a felhasználók saját maguk tudnak osztályok, kisebb, nagyobb közösségeket létrehozni. Lehetőség van egymás között különböző fájlok megosztására, beszélgetés indítására, események létrehozására, melyet a 2. ábra szemléltet.



2. ábra: Naptár funkció

2.3. Innováció

A program kidolgozásánál arra törekedtem, hogy a különböző tanulási profillal és egyéni tanulási stílussal rendelkező tanulók igényeinek is megfeleljen. Újításként így került bele egy egyedi igényt kielégítő eszköz gyengénlátó tanulók részére, melyet gyógypedagógussal egyeztetve készítettem el. A fejlesztés lényege, hogy a speciális ikonra kattintva a képernyő kontrasztosabb üzemmódra kapcsol, megnagyobbítva a tartalmat, ezzel is átláthatóbbá téve az oldalt és támogatva a sérült látásszervet.

A másik újítás a teszt jellegű feladatok (melyek közvetlenül a gondolkodási képességet és a felidézés gyorsaságát fejlesztik) mellett, hogy párhuzamosan kialakításra kerültek az esszé jellegű feladatok, aminek segítségével a tanulók szövegalkotási és szövegértési képességeik fejleszthetők.

3. Elért eredmények

A szoftver alapjait 2017. szeptemberében készítettem el. Már a tervezés stádiumában is több intézmény érdeklődéssel fogadta kezdeményezésemet, ez pedig további fejlesztésekre inspirált. A visszajelzések alapján jelenleg 3 intézmény több száz tanulója próbálta ki a program adta lehetőségeket. A jelzett hibákat folyamatosan javítottam és további fejlesztéseket végeztem. Továbbfejlesztésként a szakmai tanárok bevonásával a szakmai tárgyak használatának bevezetése kerül megvalósításra.

Végzős tanulóként jövőbeli célom, hogy az egyetem falai között tovább mélyítsem tudásomat, mellyel nem csak az oktatási szférában, hanem az élet bármely területén tudok professzionális színvonalú programokat készíteni.

SentiMentality – karnyújtásnyira a közvélemény

Cyber-sentiment

Kónya Leon, Sztarek Norbert

Felkészítő tanár: Kőrösi Gábor

Bolyai Tehetséggondozó Gimnázium és Kollégium, 24400 Zenta, Posta utca 18.

1. Bevezetés

Manapság rohamosan szaporodik a felhasználó által generált dokumentumok mennyisége közösségi oldalakon, hálózatokon. Ilyen oldalak például a Twitter, Facebook, Reddit, közösségi hírportálok kommentjei, politikai vitafórumok és sok más. Ezek a dokumentumok sok esetben valamilyen véleményt fejtenek ki egy témáról, érzelmet fejeznek ki.

Sok vállalkozás vagy politikai kampány sikeressége, bármekkora méreteken, függ a felhasználói, fogyasztói visszajelzéstől. Fontos számukra, hogy a visszajelzés tükrében helyes irányba tudják fejleszteni anyagi vagy eszmei terméküket, marketingjüket javítani.

A fenti gondolatmenet alapján állíthatjuk, hogy előnyös lenne számos szervezetnek statisztikai mennyiségben kinyerni a közösségi hálózatokon rejlő közvéleményt, digitalizálni, megmérni az emberek érzelmeit.

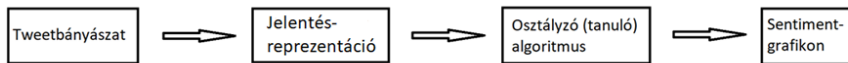
A szövegbányászat és a közvélemény-kutatás már eleve létező iparágak és kutatási területek, amelyek ezzel a témakörrel foglalkoznak, ám tapasztalataink szerint mindazon emberek számára, akik nem rendelkeznek egy előrehaladott vállalkozással vagy programozási háttérrel, nehézségek merülhetnek fel céljuk elérésében.

Ez motivált bennünket arra, hogy elkészítsünk egy szoftvert, amely vizsgálja a pozitív és negatív hangnemű felhasználói bejegyzések arányainak változását Twitteren. Lehetővé teszi programozási tapasztalattal nem rendelkező emberek számára, hogy következtetéseket vonjanak le az adatból, szociológiai kutatásokat végezzenek, feltörekvő vállalkozásuk termékének hírnevét követni tudják, vagy csupán csak kielégítsék kíváncsiságukat.

2. Probléma megoldásának menete

Az elképzelésünk az volt, hogy egy olyan szoftvert készítsünk, amely képes ábrázolni egy grafikonon a Twiterről kinyert pozitív és negatív hangnemű szövegek arányának a változását. A grafikonok szelekcióját úgy képzeltük el, hogy a felhasználó megadja az időintervallum kezdetét és végét (keltezéseket), valamint egy Twitter témát (hashtaget), hogy a szoftver e témán belül az adott időszelést „szentiment-grafikonját” rajzolja ki.

A munkánk során 4 főbb lépést különítettünk el, amelyet ábra 1 szemléltet.



1. ábra: munkánk folyamatábrája

2.1. Tweetek kibányászása

A megadott követelmények alapján kérvényt lehet küldeni a TwitterAPI felületnek, amely tweetszövegeket ad vissza illetve azok további adatait. Egyetlen hátránya ennek az eljárásnak az, hogy vannak időkorlátai, így bizonyos időnél régebbi tweeteket már nem vesznek figyelembe. Erre használtunk egy TwitterScraper alkalmazást, amely a felhasználó görgetését szimulálja JavaScript eljárásokkal, így elérhetővé teszi a korlátlan tweetkinyerést. A felület kezelése Python nyelven történt.

2.2. Tweetek feldolgozása

Ahhoz, hogy szöveganalízist tudjunk végezni egy szövegen, valamilyen számszerű formába át kell alakítani. Mi erre a Doc2Vec eljárást alkalmaztunk, amelyhez át kellett néznünk pár publikációt az eljárás matematikai hátteréről, valamint a Python gensim könyvtárát, amely a vektorizációs eljárásokat biztosítja.

Maga a vektorizáció úgy történik, hogy első alkalommal, amikor végigment a program a tweeteken, felvesz minden szóra véletlenszerű értékeket, majd amikor következő pár (a mi esetünkben 50) alkalommal ez megtörténik, kontextus alapján változtat a szavak vektorjain (tehát ha hasonló kontextusban vannak jelen, akkor az értékeket egymáshoz közelebb rajta). Ezt az eljárást Doc2Vec-nek nevezzük, vagy Paragrafus vektornak (egy egyszerűsített példát láthatnak a 2. ábrán).

Miután van számszerű adatunk, készek vagyunk arra, hogy szabályszerűségeket keressünk egy algoritmus segítségével az adatban. Ez úgy történik, hogy egy előre osztályzott, valamint a Doc2Vec-es eljárás segítségével vektorizált tweeteket az n (a mi esetünkben 400) dimenziós vektortérben elhelyez, majd egy $n-1$ dimenziós függvényt rajzol, amely 2 részre osztja a teret, a pozitív és negatív részre. Az alapján, hogy az új tweetek hol helyezkednek el, tudja a program meghatározni az \hat{o} számban kifejezett érzelmüket. Ezt az eljárást Logistic Regression-nek nevezzük (egy egyszerűsített példát láthatnak

a 3. ábrán). Ezeket a metódusokat Python programozási nyelvben írtuk meg, főként a scikit-learn, valamint a gensim könyvtárak segítségével.

2.3. Statisztikai feldolgozás

Miután meg lett minden egyes szöveg szentimentje (az, hogy 1 vagy 0) határozva, a grafikon kirajzolásához összegeznünk kellett az eredményeket. A szervergép, amely az egészet lekérte a weboldalról, ismét felküldi a weboldalnak PHP oldalakon keresztül, amely kiszámítja a „szentiment-grafikont”. Ábra 4-en látható a felhasználói felületén kirajzolt grafikon, melynek elemzése a 3. fejezetben következik.

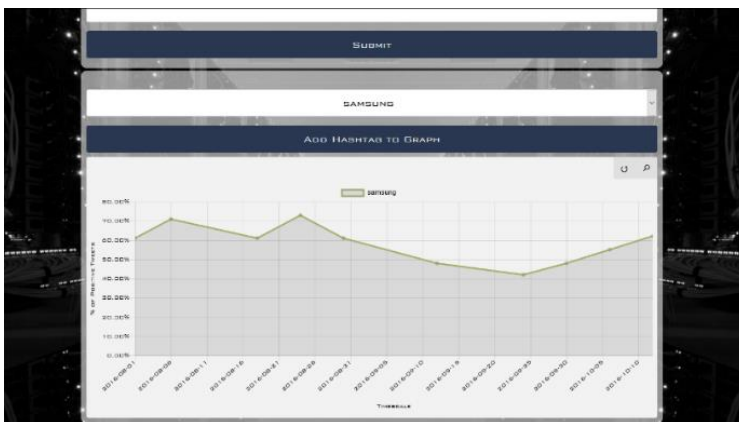
3. Elért eredmények

Ami a Samsung 2016-os évét illeti, számára nem volt túlságosan sikeres. Ha megvizsgáljuk a grafikon, egy emelkedést figyelhetünk meg augusztusban, pont abban az időben, amikor a Note 7-es telefon piacra adását bejelentette a cég. Miután piacra került a telefon, egy gyári hiba következtében világszerte elkezdtek felrobbani az akkumulátoraik, ami várhatólag a Samsung hírnevének csökkenését eredményezte.

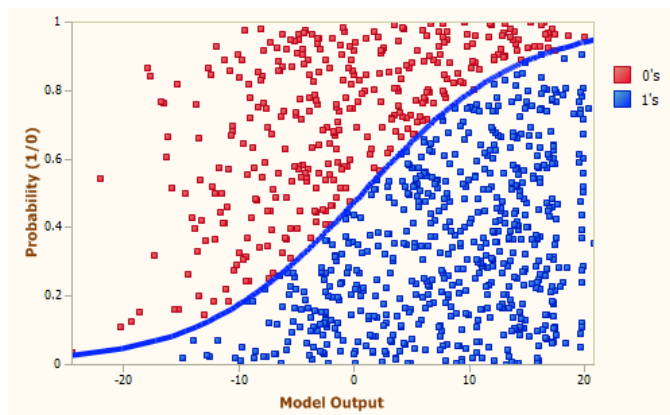
Ezt a feltételezést igazolja a grafikonunk, amely augusztus vége felé komoly süllyedést jelez.

Ez a példa, illetve tétel alapján állíthatjuk, hogy az általunk használt eljárás valóban tükrözi a valóságban történő eseményeket, vagyis megbízható, közvélemény-kutatásra alkalmas szoftvert hoztunk létre.

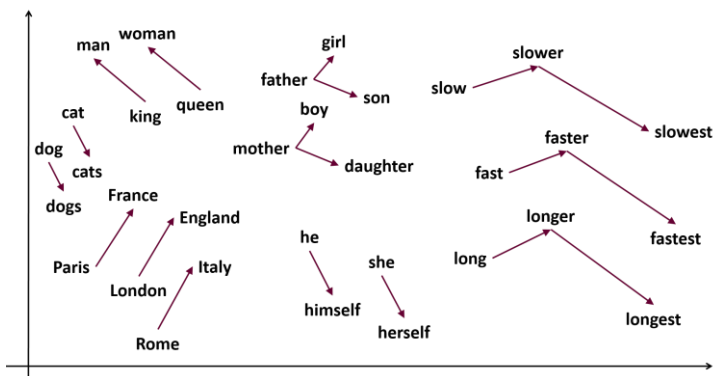
Mivel a kezdetekben kikötött célunkat elértük, felmerül a kérdés, hogy hogyan tovább. Saját, központi szervergép igénybe vétele aránylag költséges lenne, így egy félig decentralizált paradigma kidolgozásán gondolkodunk. Alapjában véve minden felhasználó, aki a standardnál nagyobb mennyiségű adathoz szeretne hozzáférni, annak hozzá kell járulnia a „közvélemény kibányászásához”, olyan formában, hogy processzorerejének egy részét felajánlja a rendszer számára, és egy ún. proof of work-öt számít ki, amelyet ellenőrizhet bárki. Ez valójában olyan, mintha egy állandó függvény eredményét számítaná ki a felhasználó, bemenetként véve egy időintervallumot és egy témanévet.



2. ábra: Samsung 2016-os grafikonja a felhasználói felületen



3. ábra: bináris klasszifikáció szemléltetése



4. ábra: vektoros jelentésrepresentáció szemléltetése

Memeium & Memeium Services

Voidloop

Rátki Barnabás

Felkészítő tanár: Kürtösi Balázs

Árpád Gimnázium, 1034 Budapest, Nagyszombat u. 19.

1. Bevezetés

A Memeium egy blockchain rendszeren működő kriptovaluta és az ehhez tartozó szolgáltatásokat biztosító oldal a Memeium Services. A Memeium egy P2P hálózaton üzemel, mindenki futtathatja a saját teljes értékű szerverét (ún. teljes node-ot). A teljes node applikáció C#-ban megírva, dotnet 2.0 fordítóval lefordítva, hogy minden változtatás nélkül futtatható legyen Windows, Mac és Linux rendszereken. A weboldal ASP.NET Core-ban van elkészítve és jelenleg a Microsoft Azure-on van hosztolva.

2. Probléma megoldásának menete

A kitűzött cél az volt, hogy létrehozzak egy olyan kriptovalutát amit én, és más emberek könnyen tudjunk használni, mint egy digitális token. Nem is volt semmi olyan célja, hogy igazi gazdasági értékkel rendelkezzen, inkább csak jelképes, a felhasználók által nyeri értékét.

Cél volt az is, hogy a token egyszerűen használható legyen és a mozgatása egyik tulajdonostól a másikig relatív kevés időt vegyen igénybe. Másodlagos cél volt, hogy a mai teenagerek világában nagyon elterjedt úgynevezett „meme”-eknek (a „meme” valami vicces kép az interneten) az originalitását díjazni lehessen. Erre készült a szolgáltatások oldal, aminek felhasználói ingyenesen kezelhetik tokenjeiket.

2.1. Memeium protokoll

A Memeium protokoll egy elég standard blockchain rendszer. A hálózat P2P, tehát a teljes node-ok egymással közvetlenül kommunikálnak le minden fő adatot. Ezzel is biztosítva, hogy semmilyen fél nem tudja egyoldalúan befolyásolni a rendszert. (Ha egy központi adatbázisban lenne tárolva mindenkinek a tokenjeinek mennyisége, az adatbázis fentartója csak úgy megváltoztathatná azt). A kommunikáció UDP protokollon történik, akármelyik node ha offline állapotba kerül, akkor sem lesz semelyik adat sem elérhetetlen (már persze ha a hálózaton 1-nél több teljes node van). Minden adat minden teljes node-on el van tárolva.

2.2. Protokol tulajdonságok

Mint minden blockchain rendszer, a Memeium is egy kizárólagosan írható nyíltan elérhető nyilvántartás („Append only public ledger”), ami csak annyit takar, hogy minden teljes node aki betartja a hálózat „szabályait”, olyan adatot tud generálni, amit a többi node a hálózaton igaznak fog elfogadni és senki sem tud törölni semmilyen adatot (csak lokálisan). A blockchain lényege (a neve is innen ered) az, hogy az adatot blokkokra bontjuk le (egy blokkba fogjuk az új tranzakciók adatait) és minden blokk az előtte lévő blokkra épít. Ezeket az adatsomagokat kell minden node-hoz eljuttatni ahhoz, hogy a hálózaton úgynevezett konszenzus alakuljon ki (egyetértés). Viszont ha mindenki tud generálni újabb adatokat, akkor, hogy döntjük el, hogy mégis kinek írhatja „fel” az általa generált legújabb blokkot a nélkül, hogy kavardás alakulni ki (egyszerre több node is ugyan arra a blokkra építene egy újabbat). Ennek a megoldására egy általános konszenzus algoritmust alkalmazunk. Aminek az az alapja, hogy az SHA 256 hash algoritmus kimenete pseudo-random, és minden node meg tudja állapítani, mennyi lehet a hálózat össz „teljesítménye” a blokkok kiadásának időközéből. Ennek a működését elmagyarázni kevés lenne négy oldal. Fontos még megjegyezni, hogy aki sikeresen megnyeri magának az új blokk felírásának lehetőségét, megnyer 4.2 token-t is (1 token = 1 MIM (Memeium) = 100 000 RIP (A legkisebb kifizethető egység 1 RIP)).

2.3. Kriptográfia

A Memeium hálózaton minden „pénztárcát” a címe alapján azonosítunk be. A tranzakciók eredetiségét úgy tudjuk bizonyítani, hogy mindenkinek van egy RSA kulcs párja és a privát kulcsával a nyíltan elküldött adatokat még egy példányban aláírva mellékeli a tranzakcióhoz, azzal együtt, hogy megadja a nyílt kulcsát is. Viszont mivel a címe egy tárcának csak a nyílt kulcsának az SHA 256-os hash-e ezért gyakorlatilag csak a cím tulajdonosa tud küldeni tranzakciót, hiszen csak ő ismeri a privát kulcsot. Persze akárki a nyílt kulcs birtokában meg tudja erősíteni, hogy az aláírást csak a nyílt kulcs tulajdonosa készíthette el.

2.4. Szolgáltatások

A szolgáltatások oldal (memeium.ml) elérhető mindenki számára. Felhasználói rajta tárolhatják biztonságosan a kulcsaikat*. Ezekkel új tranzakciókat hozhatnak létre. Az oldalon lehetőségük van tranzakció kérés létrehozására is. 1. Ábra: a QR- kódot ha eljuttatják a másik félhez és ő beolvassa, majd meglátogatja a benne tárolt linket (akár ezt is el lehet közvetlen küldeni) rögtön a megfelelő címre küldhet egy tranzakciót a kért

összeggel. *(Azt gondolhatnánk, hogy ez elrontja a blockchain lényegét, de valójában a teljes node-ról közvetlen kezelhetjük a címünket és küldhetünk tokeneket, az oldal csak azoknak hasznos akik nem akarják maguk futtatni a node applikációt és megbíznak egy nyílt forráskódú oldalban.)

Még fontos funkció az is, hogy minden nap indul egy verseny, ahova a felhasználók feltölthetik memeeiket és mindenki szavazhat, hogy szerinte melyik a legjobb. A nyertes megnyeri a többi ember által felajánlott token összeget. Persze az oldalon közvetlen adhatunk tokenet annak az embernek, akinek a képe elnyerte a tetszésünket.

Request Payment

Create the transaction request below:

Select Wallet :

Lols 0 MIMs

Lols

Balance: 0 MIMs

Address: c3e6ZSzmAdriLPPzLDkTy+vzi2b7+0tMlJeymzMKswY=

Personal Message:

Hello fellow memer!

Transaction Message:

This will be saved to the blockchain.

Amount :

0

Make Request

Hello fellow memer!

This will be saved to the blockchain.

To Address : c3e6ZSzmAdriLPPzLDkTy+vzi2b7+0tMlJeymzMKswY=

Send this QR code or link to someone so they can pay you.)

<http://memeium.azurewebsites.net/Home/PlayRequest?msg=Hello%20fellow%20memer!&msg%20will%20be%20saved%20to%20the%20>



1. ábra: Tranzakció kérés példa

3. Elért eredmények

A pályamunkám eredetileg nem a pályázatra lett közvetlen fejlesztve, hanem csak saját magam és barátaim szórakoztatására, de azért is, mert nagyon érdekelt az alatta lévő technológia. Az ötlet szerintem elég kreatív de persze más szempontból nem az, mivel nem végeztem új kutatásokat a blockchain technológiával kapcsolatban, inkább csak bevált módszereket ötvöztem. A probléma nélkülöz minden fontosságot hiszen, célja inkább a szórakoztatás. Minden kód fent van githubon: [Memeium Github](#). A teljes node applikáció ~4250 sor. A weboldal ~3440 sor. (Csak a saját kódom számolva.) Megjegyzés: nem mindig van futó teljes node, és olyankor a weboldal minden egyenleget nullának mutat.

Felhasznált könyvtárak és források:

- [Bitcoin Wiki](#)
- [EmbedIO \(Embeded Rest server\)](#) * A teljes node-on fut egy REST server de nem ezen keresztül kommunikálnak a node-ok, csak a nyílt REST API megy rajta.
- [Newtonsoft.Json](#)



2. ábra: A Memeium logó (Szabó Bianka munkája)

Arcfelismerés alapú beléptetőrendszer

csapatnév¹

Suszter Ágnes Eleonóra, Robotka Adrián

Felkészítő tanár: Csató Endre

Földes Ferenc Gimnázium, 3525 Miskolc, Kelemen Didák utca 5

1. Bevezetés

Pályamunkák egy arcfelismerésen alapuló beléptető rendszer megvalósítását tárgyalja. Az általunk megoldani kívánt problémát a Microsoft Imagine Learn to Win Contest nevű országos versenyen kaptuk.

Elsődleges célunk, hogy bizonyos rendezvények (pl. szakmai konferenciák) vendégeinek beléptetését egyszerűbbé és gyorsabbá tegyük. A beléptetés előfeltétele a regisztráció, amit egy mobil alkalmazás biztosít. Ehhez néhány személyes adat megadása és több fotó szükséges a regisztrálandó személyről. Az adatokat egy web API tárolja, ami hozzáférést is nyújt a külső arcfelismerő és azonosító szolgáltatáshoz. A beléptetéseket egy kliens alkalmazás végzi a helyszínen, amely egy kamera élőképet használja adatforrásként a felismeréshez. Megoldásunk készítése során igyekeztünk minél jobban lefedni azokat a problémákat, amik felmerülhetnek egy ilyen környezetben.

2. Probléma megoldásának menete

2.1. Fejlesztés menete

Amikor neki láttunk megoldani a problémát, akkor még nem volt elképzelésünk rendszerünk várható működéséről, így proof of concept² módszerrel igyekeztünk kialakítani egy működési modellt.

Lépésről lépésre tudtuk meg mi is a járható út. Ilyen volt az élő videófolyam feldolgozása és az abból történő arcfelismerés, továbbá az ezen rendszer köré épített eseménykezelő rendszer, ami a megfelelő visszajelzéseket és az alkalmazást irányítja. Újabb és újabb fejlesztési ágakat nyitottunk Git verziókezelő rendszerünkben. Volt hogy 1-2 vakvágánynak bizonyult, de ez a módszer több sikeres funkcionalitással gazdagította a végeredményt.

Fejlesztés közben igyekeztünk minél több esetvizsgálatot tenni, hogy „bomba biztos” legyen a munkánk. Így került bele a több képernyős mód is. Egy képernyő a beléptető asszisztensnek, egy másik a vendégnek. Erre azért volt szükség, hogy csak releváns információkat jelenítsünk meg a feleknek.

¹ A csapatnevünk tényleg a következő: csapatnév

² Kísérleti projekt; ötletek kipróbálása, hogy működőképesekek-e

A programjaink további fejlesztéséhez agilis módszertant választottunk. Ezzel folytatni kívántuk az előbbieken említett elképzeléseinket, ugyanakkor produktívabbá tettük a fejlesztést.

Mivel objektum-orientált nyelveket használunk, így megfelelő módon tudtuk implementálni az architektúrális, szerkezeti és létrehozási tervezési mintákat, amelyek segítségünkre voltak a megvalósításban.

2.2. Fejlesztés közben felmerült problémák

Az arcfelismerés és azonosítás alapkövei munkánknak. Fontos volt a minél megbízhatóbb azonosítás biztosítása. Ehhez a Microsoft Cognitive Services Face API-t választottuk. Ez egy pénzdíjas, de kiváló szolgáltatás. Mi az ingyenesen igénybe vehető API lekérdezések keretei közé szorultunk a továbbfejlesztés alatt.

Az első tesztelések alkalmával szembesültünk a webkameránk szűk alkalmazhatóságával, ezért megkerestük a Canon európai terjesztését, hogy részt vehessünk a Canon developer programjában. Mivel sikeres elbírálást kaptunk így lehetőségünk nyílt a csapat tulajdonában lévő 700D típusú DSLR fényképezőt webkameraként használni, de ehhez szükség volt többletkódolásra és a Cannon EOS SDK-ra.

3. Elért eredmények

A Java nyelvet és az Android platformot választottuk mobilalkalmazásunkhoz. Ezzel a többnyelvű alkalmazással lehet jelentkezni az eseményre. A tájékoztató képernyő után a regisztrációs képernyővel (lásd: 1. ábra) találkozik felhasználó, ha rendelkezik internet kapcsolattal. Az alkalmazás álló és fekvő orientációra is fel van készítve.

1. ábra: Regisztrációs felület

Figyeltünk a felület felhasználó barát kialakítására, így a jól megszokott vezérlő elemeket használtuk. Ahol tudtunk ott a még kényelmesebb formáját valósítottuk meg az adatbevitelnek, például a dátumválasztáshoz dedikált ablak jelenik meg. Adatrögzítés előtt ellenőrizzük, hogy az adatok megfelelnek-e a formai követelményeknek. Ennek része az egyedi ímélcím megszorításnak való megfelelés vizsgálata is. Az alkalmazásban 3 db képet kell feltölteni a személyről (ez az API-n beállítható), amelyeken nem lehet több arc. Ezt ellenőrizzük is. A fotók kiválasztására a kamera és a képgaléria áll rendelkezésre. Az alkalmazás előző bemutatása óta lecseréltük a kezdő és sikeres regisztráció képernyőt, valamint segítséggel láttuk el a képfeltöltést.

API URL-ek		
HTTP metódus	URL path része	Leírás
POST	/detect/	Arc(ok) felismerése
POST	/identify/	Arc(ok) azonosítása
POST	/participant/	Résztevő létrehozása
GET	/participant/{participant_id}	Adatok lekérdezése
POST	/participant/{participant_id}/image	Kép feltöltése
GET	/participant/checkEmail/{email}	Email ellenőrzése

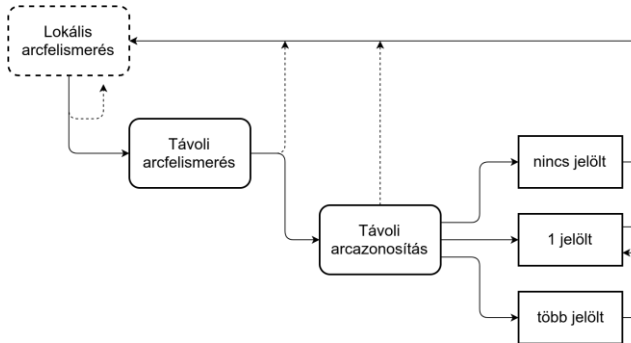
1. táblázat: API URL-ek

Az adatok tárolásához PHP nyelven készítettünk RESTFUL API-t³. Az API funkcionalitásáról áttekintést nyújt az 1. táblázat. Ehhez Laravel keretrendszert használtunk és ORM⁴ rendszeren keresztül MySQL adatbázis kezelővel üzemeltetjük. Ügyeltünk a biztonságra: kötelező az API kulcs továbbítása, figyelmet fordítottunk az XSS, sql injection, overflow jelenségek kiküszöbölésére. A keretrendszernek köszönhetően támogatja a gyorsítótárazást és könnyen skálázható az API. Szabadon választható az adatbázis kezelő rendszer és a helyi fájlkezelésen felül felhő alapú tárolást is be lehet állítani. Mivel az arcfelismerést egy külső API valósítja meg, tervezési, üzemeltetési és biztonsági szempontok alapján úgy gondoltuk, hogy az alkalmazás csak a mi API-nkon keresztül érheti el az arcfelismerés szolgáltatást. Így a mi API-nk adja hozzá az azonosítási kulcsot a lekérdezéshez. Az előző bemutatás óta weboldalként is funkcionál az API. Bejelentkezhetnek

³ REST: REpresentational State Transfer; kihasználjuk a get, post, put, delete HTTP metódusok lehetőségeit.

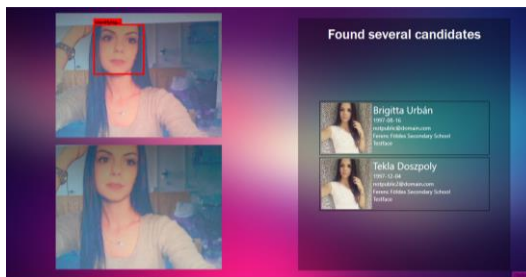
⁴ ORM: Object-releational mapping; az adatbázishoz objektum szinten férünk hozzá absztrakciókon keresztül

a szervezők és kezelhetik a regisztráltakat valamint a beléptető személyzetet, akiknek be kell lépniük ezen túl a beléptető kapuknál az alkalmazásba. A résztvevőket osztályokba csoportosíthatják (pl: VIP, normál, stb.) Ezen túl élő belépési képernyő is elérhető az oldalon.



2. ábra: Felismerési folyamat főbb esemény forrásai

C# nyelven írt kliens alkalmazásunk inicializációs ablakkal indul. Ebben ki kell választani a kamerát, be kell állítani, hogy működjön-e a lokális arcfelismerő, ami előfeltétele a távoli arcfelismerésnek, és hogy megjelenjen-e a látogatói ablak. A 2. ábrán az arcfelismerési folyamat főbb állomásai láthatóak. Itt az események vízésés modell szerint érhetőek el. A kezelhetőség javítása érdekében ezt a ciklust újraindulása előtt különböző várakoztatásokkal láttuk el. Elképzelésünk szerint egy pontos egyezés után kevesebb ideig kell a visszajelzést a képernyőn tartani, kisebb ideig tartva fel ezzel a következő azonosítást, mint amikor egy ikerpárból ki kell választani, hogy pontosan melyik lépett be (lásd: 3. ábra). Az előző bemutatás óta bejelentkezési képernyővel bővült az alkalmazás. Megjeleníti a résztvevők csoport tagságát, jelentést küld amikor több lehetséges személy közül választották ki a beléptetettet. Néhol a belső működésen is javításokat eszközöltünk.



3. ábra: Több azonosítási találat

Sapphire

Mészáros Bálint

Felkészítő tanár: Fekete Balázs

*Debreceni SzC Mechwart András Gépipari és Informatikai Szakgimnáziuma,
4025 Debrecen, Széchenyi utca 58.*

1. Bevezetés

Napjainkban az információtechnológiai eszközök és megoldások egyre inkább életünk szerves részét képezik. Lassan nemcsak a mobiltelefonunk és a számítógépünk fog hálózatba kapcsolódva kommunikálni a hálózatunk vagy az Internet többi részével, de a háztartási eszközeink, sőt akár a világítótesteink is. Az informatika megállíthatatlanul fejlődik, és vele együtt fejlődik a környezetünk, a világ, ahol élünk.

Egy ICT világban otthonosan mozgó fiatal azonban nem, vagy nem mindenütt tud teljesen azonosulni ezzel a gondolattal. Gondoljunk csak bele: ma Magyarországon buszra szállva papír alapú jegyet kapunk, őrizgetjük a papír alapú számláinkat, sok ügyet pedig személyesen intézünk a digitális út helyett.

Nem volt ez a kollégiumomban sem másképpen: a digitális megoldások helyett a legtöbb nyilvántartás külön, papíralapon történt vezetésre, a leltártól kezdve, a kimenőkön át, az étkeztetésig szinte minden terület visszamaradt a fejlődésben. Biztos, hogy ennek 2018-ban és a XXI. században így kell lennie?

A helyes válasz: nem! A technika rendelkezésünkre áll, nekünk csak használni kellene, és használjuk is. A kifejezetten kollégiumokra szabott szoftver képes az intézmény mindennapjainak valamennyi területén jelentős segítséget adni, legyen szó a diákok vagy a kollégiumi ingóságok nyilvántartásáról, beléptetéséről, étkeztetéséről vagy akár az IT eszközök hozzáféréseinek szabályozásáról.

2. Probléma megoldásának menete

Célom volt a szoftver megalkotása során, hogy egy olyan korszerű alapokra építkező integrált szoftver-, és szolgáltatáscsomagot tudjak a kollégiumok részére bocsátani, mely egyszerűen, hatékonyan és minél kevesebb kellemetlenséggel képes átvenni a papíralapú nyilvántartások jelentős hányadát amellet, hogy a kollégiumi dolgozók és a diákok számára innovatív szolgáltatásokat nyújt.

Célom volt továbbá, hogy a szoftver kiszolgálói és adattároló része nem elosztva, hanem központosítottan szolgálja ki az intézményeket, mely

egyrésről jelentős karbantartásból fakadó terhet vehet le az intézmény válláról, másrészt így biztosítható, hogy az intézmények közötti átjárás megvalósulhasson.

2.1. A szoftvercsomag fejlesztésének kezdete

A szoftvercsomag fejlesztését a HBM-i Általános Iskola, Gimnázium és Kollégium Györffy István Kollégium Intézményegységének diákönkormányzatának kérésére kezdtem meg. A diákönkormányzat (és a kollégium nevelőtestületének) elvárása az volt, hogy egy olyan okoskártyákkal működő és a diákok épületből történő be-, és kilépését naplózó rendszer jöjjön létre, mely a meglévő kollégiumi infrastruktúrába és nyilvántartási rendszereibe egyszerűen integrálható, emellett pedig a számítástechnika világában kevésbé otthonosan mozgó kollégiumi alkalmazottak is probléma nélkül tudják kezelni azt.

A követelményrendszer teljesítéséhez egy szoftvercsomag készült, mely több komponensből is áll: egy PHP7 háttérrendszeren nyugvó szerverprogramból, és egy .NET C#-alapú kliensprogramból, mely egyik alkotórésze a szerver és a kliens közötti kommunikációt, míg a másik a kliens és a hardvereszközök közötti kommunikációt biztosítja.

2.2. Az ötlet tovább gondolása

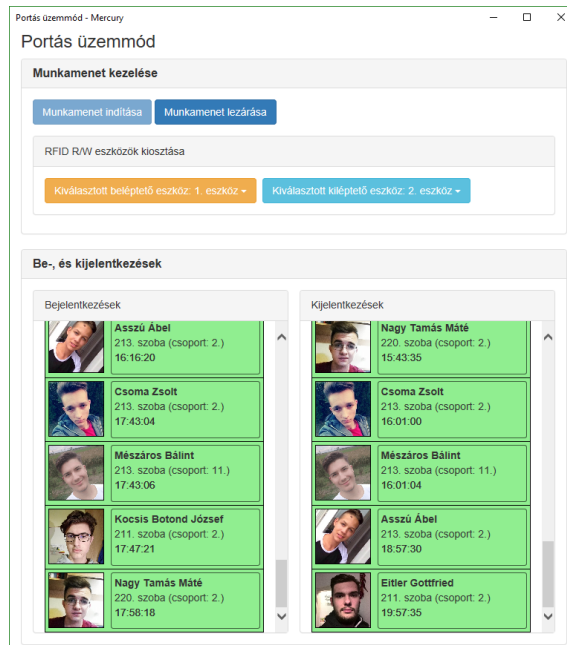
A kollégium által elvárt funkciók helyes és ésszerű működéséhez a szoftvernek tárolnia kellett a tanulók alapadatait (ideértve az igazolványképet is), a tanuló jogviszonyának adatait (beleértve a tanuló szobaszámát és csoportját is), a kollégiumi alkalmazottak intézményhez való viszonyát és a különféle szintű események naplóját is. Ezek az adatok (még ha nem is ilyen részletességgel) lefedik a kollégiumi nyilvántartások egy jelentős részét, így redundancia alakul ki a papíralapú és a digitális adattárolási módszerek között az adatok tekintetében.

A szoftvert ezért továbbfejlesztettem: az alapfunkciók mellett a program alkalmas arra, hogy a kollégiumban található (és/vagy a kollégium tulajdonában lévő) ingóságokat nyilvántartsa, a kimenőket (engedélyeket) kezelje, az étkeztetés lebonyolítását segítse és az információs infrastruktúra egyes elemeihez történő hozzáférést vezérelje.

2.3. Be és kiléptetés, a kollégisták aktuális állapotának nyilvántartása

Ez a szoftverem alapfunkciója. A portán elhelyezett két RFID R/W eszközt használva (egy a kilépéshez, egy a belépéshez) és a kollégisták egyedi okoskártyájának segítségével a kollégisták épületbe történő be és kilépése

naplózható, a diákok aktuális állapota (belépve vagy kilépve) pedig lekérdezhető. A szoftver ezeket az adatokat grafikusan, szintekre és szobákra tagolva, színekkel segítve jeleníti meg a jobb áttekinthetőség érdekében.



1. ábra: A portás üzemmód felülete a be-, és kiléptetéshez

2.4. Leltárnyilvántartás

A kollégiumban számos ingóság található: számítógépek, televíziók, hálózati eszközök, bútorok, háztartási gépek és más eszközök, melyek megfelelő nyilvántartását egy felelősen működő kollégiumnak biztosítani kell. A vagyonelemek ugyanis különféle tulajdoni viszonytal rendelkezhetnek: egyes vagyontárgyakat kollégisták, másokat a diákönkormányzat birtokolhat, de természetesen a kollégiumnak és (annak jelenlegi, illetve korábbi) fenntartójának vagyonelemei is nyilvántartásra szorulnak. A szoftver nemcsak a különféle tulajdoni viszonyokkal rendelkező ingóságokat tartja nyilván, de segít a leltározás okozta megpróbáltatásokat enyhíteni is.

2.5. Étkeztetés

A legtöbb kollégium működtetett menzát is, amelyek több szempontból is sokszor idejélműltnek tekinthetők. A diákoknak étkezési jegyeket kell felvenniük és magukkal hurcolniuk, a befizetést a legtöbb helyen még mindig

csak készpénzben lehet rendezni, holott mint tudjuk, ez nem csak jelentős kényelmetlenséggel de veszéllyel is jár.

A szoftver több területen is jelentősen leegyszerűsíti az élelmezés folyamatát: a diákok a rendelésüket digitálisan adják le, melyeket az élelmezésvezető dolgoz majd fel. A kiállított számlákat a diák (vagy szülő) bárholnan kényelmesen tudja rendezni a bankkártyájával sorban állás nélkül. A diákok a már leadott rendeléseket bárholnan figyelemmel követhetik, betegség vagy egyéb váratlan esemény esetén pedig kérhetik a rendelésük módosítását is – digitálisan.

A diákok az étkeztetést az okoskártyájuk terminálhoz történő közelítésével vehetik igénybe. Ebben az esetben nemcsak az étkezési jogosultsághoz tartozó információk villannak fel (a diák alapadatai, a rendelt menü), hanem a tanuló igazolványképe is. A megoldás megelőzheti az étkezési jegyek jogosulatlan felhasználásából keletkező károkat.

2.6. Az információs infrastruktúrához történő hozzáférés vezérlése

Számos kollégiumban komoly problémát jelent az, hogy a vezeték nélküli hálózathoz az előre megosztott kulcs birtokában bárki hozzáférhetett, forgalmát pedig nem lehetett semmilyen személyhez kötni.

Ezen szeretnék segíteni egy, a szoftverbe épített megoldással, mely – az IT hálózat többi elemével összehangolva – biztosítani tudja a hozzáférés vezérlését és naplózását is. A hálózathoz történő csatlakozáskor a rendszer kérni fogja a felhasználó egyedi azonosítóját és jelszavát is. A felhasználó hitelesítése után a rendszer alkalmazhatja a munkamenethez kapcsolódó egyedi korlátozó szabályokat, így a tiltott és szűrt tartalmakhoz csak az erre jogosultak férhetnek hozzá.

Amennyiben a kollégium kiterjedt vezetékes infrastruktúrával rendelkezik, az ahhoz történő hozzáférés szabályozása is megoldható a szoftver segítségével, a vezeték nélküli hálózatokhoz hasonló módon (például az IEEE 802.1x alkalmazásával).

3. Elért eredmények

A kollégium alkalmazottjai és diákjai mintegy három hete használják próbaüzemben a szoftver alapfunkcióit, nagy meglepéssel. A szoftver felhasználói – számos visszajelzés biztosítása mellett – kiemelték a szoftver felhasználóbarát és reszponzív jellegét, és kiemelten dicsérték a megoldás egyszerűségét is. A visszajelzések alapján hiszem, hogy a folyamatos fejlesztéseknek köszönhetően egyre több kollégium lépheti át a XXI. század küszöbét, és így egyszerűbbé teheti a diákok és munkatársak mindennapjait.

Rubik kocka kirakó program

fsociety

Bakos Kristóf Nándor; Nagy Ádám; Hunkó Dániel

Felkészítő tanár: Török Attila

*Miskolci Szakképzési Centrum Bláthy Ottó Villamosipari Szakgimnáziuma,
3525 Miskolc Soltész Nagy Kálmán utca 7.*

1. Bevezetés

A mi feladatunk a logikai problémák ezen belül a 3 dimenziós logikai játékok körébe tartozik. Mióta Rubik Ernő megalkotta a bűvös kockát, azóta rengeteg hasonló térbeli játék került a boltokba, melyek kirakása rendre komoly fejtörést okoz az átlag emberek számára. Rubik kocka szinte minden ember polcán van, de csak kevesek tudják kirakni. Úgy gondoltuk, hogy ehhez adunk a programunkkal segítséget.

Ezen a téren ismereteink-csak úgy mint sok más emberé- elége szűkek, melynek következtében sok akadállyal néztünk szembe, mely végeredményben pozitívan járult hozzá mind problémamegoldó-képességünk, mind kooperációs képességeink fejlődéséhez.

2. Probléma megoldásának menete

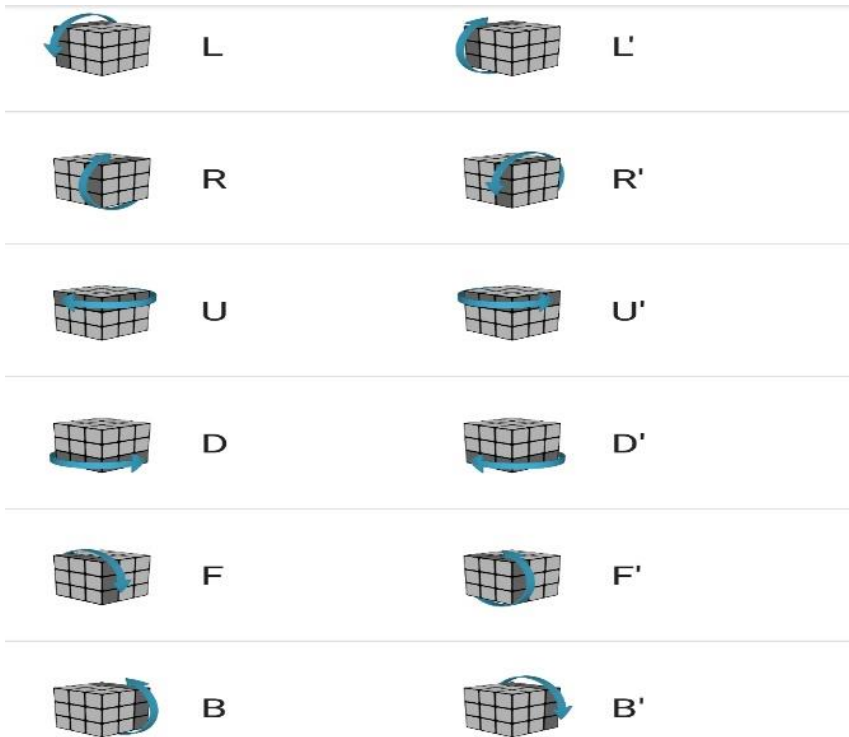
Kezdetben több személyt is megkérdeztünk a probléma gyakoriságának kiderítése érdekében, és végezetül arra jutottunk, hogy eléggé gyakori problémaként jelenik meg az, hogy van otthon egy Rubik kockája valakinek, de képtelen annak megoldására. Ennek következtében feladatunkban egy olyan célt tűztünk ki magunk elé, amelynek sikeres megvalósításához vezető út során számos weboldalt volt szükséges átkutatnunk és megannyi tananyagot elolvasnunk ahhoz, hogy viszonylag tökéletes megoldást találjunk az olyan személyek számára – mint mi is vagyunk -, kiknek fogalmuk sincs egy ilyesfajta Rubik kocka kirakásáról.

Miután elhatározásra jutottunk a feladatunkban a megoldás módját kieszelve neki is kezdtünk a probléma megoldásának. Először (két hónappal ezelőtt) azt gondoltuk, hogy nem okozhat akkora nehézséget hiszen megoldhatónak véltük egy back-track algoritmus megoldásával, azonban tévedtünk és minél jobban fejlesztettük programunkat, annál több probléma tárult a szemünk elé.

A megoldási kísérleteink kronologikus sorrendje a következő volt:

- 12 operátort használunk és az algoritmus a 12 operátoron lineárisan halad végig a végső megoldásig.
- 12 operátort használunk azonban az algoritmus egy bizonyos heurisztikát követve halad a végső megoldásig.
- Rájöttünk, hogy képesek vagyunk a kirakásra „24 lépésből” (itt egy lépés egyenlő egy részmegoldással).
- Arra lettünk figyelmesek, hogy megoldásunk a mélységi korlátok és a back-track miatt nem volt elég effektív, így akár több órán át is várhattunk volna a számítógép számolására. Ezért a részmegoldásokat tovább bontottuk egy részmegoldás előtti állapotra, amelyet egy manuális forgatássorozattal eljuttatunk a részmegoldásba.

Az ezen úton elért végső megoldásunkat szintén az emlegetett 12 operátorra alapozzuk, amelyek jelölése megegyezik a szabványos forgatások jelölésével (lásd 1. ábra).



1.ábra: A forgatási szabványok

Fontos kiemelnünk, hogy mi egy fix helyzetben lévő kockával dolgozunk, akárcsak sok hasonló algoritmus. A kockának a szabályos színei vannak: fehér, citromsárga, piros, narancssárga, kék és zöld. A kockára mi úgy tekintünk, hogy a fehér lapja van felfelé és a kék lapja van felénk. Ennek következtében a zöld és kék oldalon áthaladó vízszintes tengely szerint forgatjuk a kockát, míg vissza nem érünk az alapállásba (fehér [0. lap] -> narancs [1. lap] -> citromsárga [2. lap] -> piros [3. lap]). Az alapállásba való visszatéréskor a vízszintes tengely a narancssárga és a piros oldalon halad át végül, így téve a kék lapot a negyedik lappá, a zöldet pedig az ötödikké. A színeket számokkal jelöltük: fehér:1, narancs:2, citromsárga:3, piros:4, kék:5, zöld:6 (adott szín=adott lap száma+1). Minden lap egy 3*3-mas tömbből áll. Az adott lapra rátekintve a megfelelő tengely alapján nekünk a bal felső sarokban lévő lapocskák a [0,0]-ás helyzet, míg a jobb felső a [0,2]-es. Értelmszerűen a [2,0]-ás helyzetű így a bal alsó lapocskák, a [2,2]-es pedig a jobb alsó sarokban lévő.

A bekérés színek alapján történik, amelyet a vizuális megjelenítési felületen a jobb felső sarokban lehet kiválasztani. A kocka lapjait mi színezzük be az egyes négyzetekre való kattintással, amilyenre csak szeretnénk, feltéve, ha az egy valódi 3x3-as Rubik kockáról bevitt potenciális állás. (Ezt csak egy rövid ismertetőnek szántuk a szemléletünk megértése érdekében.)

A megoldás után való kutatás során sok problémába ütköztünk, amelyeket hosszabb-rövidebb idő alatt, de sikeresen megoldottunk. Többek között akadály volt számunkra a három dimenziós rendszerben való eligazodás, hiszen nem volt, ha más irányból nézzük a kockát, akkor más kerül Pl. a [0,0]-s helyre, de egy kirakott oldal esetén ezt nem feltétlenül venni észre. Mivel az általunk megfigyelt weboldalakon a forgatási algoritmusokat az általunk megszabott alapálláshoz képest fejfelé állított kockán mutatják be, ezért az ott leírt operátorokat át kellett konvertálnunk a saját rendszerünkbe. A bemutató videókban pedig 12-nél több operátort használtak (Pl. a középső síkot is forgatták). További probléma volt, hogy a manuális lépéseknél az újabb állapotok nem jól kerültek be a listába, ezért mikor kirakta a kockát nem volt meg az odavezető útvonal.

3. Elért eredmények

Úgy tűnik sikeresen vettünk minden akadályt, melynek következtében a programunk képes kirakni egy valódi 3x3-mas Rubik kockát akármilyen létező állásból. A program futásának az ideje pedig a tapasztalatok alapján mindössze néhány tized másodperc. Miután lefut a program, egy fájlba beleírja a kirakáshoz szükséges lépéseket.

Ezen projekt fejlesztése során fejlődött a programnyelvbeli tudásunk, valamint általános térbeli látásunk is. Az elmúlt hetekben éjjel-nappal az elkészítésén fáradoztunk a 2 hónappal ezelőtt elkezdett, magunk elé bátran kitűzött feladatunkon. Rendszeresen maradtunk késő délutánig az iskolában a teljes kooperatív munka érdekében. Természetesen, ami a legfontosabb, nem csak tudásunkat bővítettük, hanem sok kedves élményre is szert tettünk a közös munka és a logikai problémák megoldása során a csapatunk tagjaival.

Mindent összefoglalva hisszük, hogy a projektünk felkeltette az érdeklődésüket, és hogy sok embernek megoldódhat így azon problémája, melyet a polcukon porosodó Rubik kocka okozott. Kívánjuk, hogy a program használóinak ugyanakkora öröme teljen a használatában, mint nekünk az elkészítésében volt.

Ne szokj rá a dohányzásra!

My8it

Cseri Gábor, Kocsis Kamilla, Kiss Gergő Csaba

Felkészítő tanár: Brósch Éva

*Berettyóújfalvai Szakképzési Centrum Közgazdasági Szakgimnáziuma,
4200 Hajdúszoboszló, Gönczy Pál u. 17.*

1. Bevezetés

Mai világunk egyik problémája a dohányzás, amire könnyű a rászakas, viszont leszokni sokkal nehezebb. A dohányzás egészségtelen, mivel sok rákkeltő anyagot tartalmaz. A dohányzó emberek naponta több szál cigarettaszálra is rá gyújtanak ezzel kockáztatva életüket. Legjobban úgy előzhetjük meg, hogy rászakjunk a dohányzásra, ha rá se gyújtunk. Mivel a legtöbb cigarettázó ember már ifjúkorában szokott rá a dohányzásra, ezért a fiatalabb korosztályt szeretnénk megcélozni a játékunkkal, amit elkészítettünk.

2. Probléma megoldásának menete

A célunk egy olyan játék elkészítése volt, ami rámutat, hogy milyen gyorsan rá lehet szokni a dohányzásra már egy-két szál elszívása után. A játékot Java programozási nyelvben írtuk meg. A játék alapja az akasztófa, csak itt egy ember van, aki minden rossz válasz után beleszív a cigarettájába és ezzel nő a függőség skálája. A helyes betű kitaláláskor a betű felrajzolódik a vonal felé, ahol éppen a betű előfordul a program által kigondolt szóban. A felhasználó célja, hogy mielőtt a játék karaktere ráfügg a dohányzásra, kitalálja a szót. Minden játék végén van egy dohányzással kapcsolatos rövid információ, amivel próbáljuk elvenni a játékosok kedvét a dohányzástól. A játék elindulás előtti felülete az 1. ábrán látható.



1. ábra: A játék elindulás előtti kinézete

2.1. A játék megírása

A játékot kisebb részekre bontottuk fel: random szót kisorsoló osztály, grafikus osztály és az ablakmegjelenéséért, gombokért felelős osztályra. A cigarettaszívás menetét képsorozattal állítottuk össze, valamint a skála növelését is.

2.2. Random szó kisorsolása

Létrehoztunk egy függvényt, ami betölt egy fájlt, ahol a szavak találhatóak. Kisorsolunk egy random számot és a random szám helyén található szót visszaküldjük a grafikus felületnek és a program fő algoritmusainak.

2.3. A grafikus osztály

A grafikus osztály felelős egy metódus segítségével, hogy megszámolja a kisorsolt szó hosszát és felrajzolja a képernyőre a vonalakat, valamint megjelenjenek a játék elemei. A képernyőn minden betűről van egy gomb, amit a felhasználó egyszer tud benyomni. Egy metódus segítségével minden gombnál eldöntjük, hogy benne van-e az adott betű a szóba, ha nincs, meghívjuk a cigarettaszívás metódusát, ha pedig benne van, akkor felrajzoljuk az adott vonal felé a betűt. A játék menet közbeni felülete az 2. ábrán látható.



2. ábra: A játék menet közben

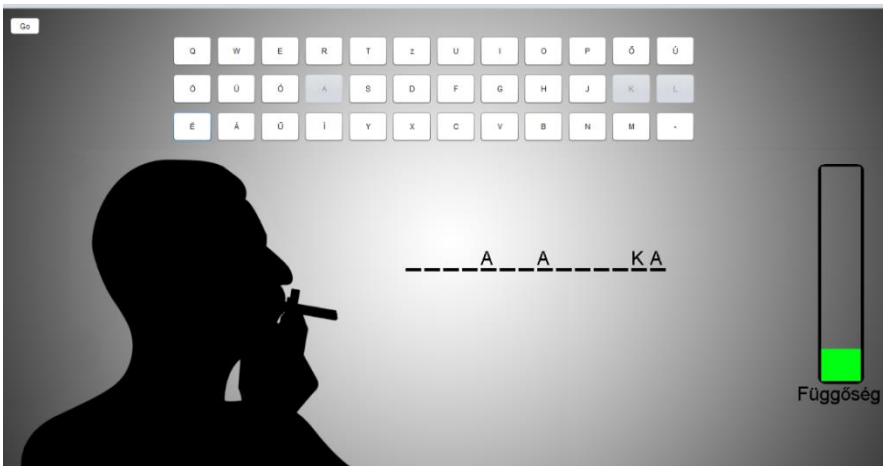
2.3.1. A cigarettaszívás metódusa

Amikor beleszív a cigarettába, egy for ciklusban mindig meghívjuk a következő képet, ahol egyre közelebb van a szájához a dohányzás és felrajzoljuk a képernyőre, majd ugyanígy visszafelé is felrajzoljuk a képeket. Itt belefutottunk egy olyan problémába, hogy az újabban felrajzolt kép mellet még

az előtte felrajzolt kép is látszódott. Ezt a problémát úgy tudtuk kiküszöbölni, hogy a megrajzolt kép kapott egy hátteret, ami lefedi az előzőleg felrajzolt képet.

Következő dolgunk az volt, hogy jó válasz esetén a megfelelő vonal felé rajzolja a kiválasztott betűt, amit úgy oldottunk meg, hogy a programban egy ciklus segítségével végig megyünk a kitalált szón és ahol megegyezik a betű a felhasználó által kiválasztott betűvel, ott meghívunk egy metódust, ami átvesz két paramétert: egy számot, hogy hányadik helyen van a betű és egy karakter típusú változót a kitalált betűvel. Az első vonal helyét megszoroztuk azzal a paraméterrel, ami azt a számot adja vissza, hányadik helyen van, és ott felrajzoltuk az adott betűt.

A felhasználó 12-szer adhat rossz választ. Minden rossz válasz esetén számoljuk hányszor adott eddig rossz választ, és növeljük a függőség skáláját. Ahogy elérjük a tizenkettőt, a skála maximumon lesz, és a felhasználó kap egy üzenet, hogy sajnos nem sikerült: rászokott a dohányzásra. A rossz válasznál történő változás a 3. ábrán látható.



3. ábra: Rossz válasznál

Jó válaszok esetén a megrajzolt ember nem szív bele a cigarettába, és így nem nő a függőség skála. Ha sikerül kitalálni a szót, akkor a felhasználó nem függött rá a dohányzásra, ezáltal nyert.

A játék ugyanakkor kapott egy menüt is, ahol elindíthatjuk a játékot, vagy kiléphetünk belőle. A játék menüpontja a 4. ábrán látható.



4. ábra: Menü

3. Elért eredmények

A játék megírása számunkra egy nagy feladat volt, mivel ilyen összetett feladattal még nem találkoztunk. Ez a program megírása segített megérteni miért is érdemes külön objektumokat, függvényeket, metódusokat használni. Amikor változtatni kellett például a szót vizsgáló algoritmuson, nem kellett minden betűnél megtenni, csak egy metódusnál, amit a gombok meghívtak. Sok új hasznos információt is szereztünk, mint például a javában beépített grafikus osztály milyen lehetőségeket kínál nekünk képeink megjelenítésére. A JFrame osztályt ismertük, de most a program írása közben ismereteinket tovább gyarapítottuk a játék elkészülése érdekében. Többek között elsajátítottuk a grafikus osztály drawImage metódus alkalmazását. Ennek érdekében szükséges volt utána olvasni bizonyos osztályok, metódusok működési elvének, szintaktikai és szemantikai felépítésének. Egyéni eredményünk volt az is, hogy egy igazi csapatként dolgozhattunk. Kiosztottuk egymás között a feladatokat, és amikor valamelyikünk elakadt, segítettük egymást a közös cél érdekében.

Együttműködő partnereink

Együttműködő partnereink



National Instruments
hungary.ni.com



NNG
www.nng.com



Optin Kft.
www.optin.hu



TEConcept Hungary Kft.
www.teconcept.hu



ELI-ALPS
www.eli-alps.hu



Digital Kft.
www.digital.co.hu



Creativ_IT
www.creativit.hu



LogMeIn
www.logmein.hu



CAS Software
www.cas-software.hu



EGYÜTT
A JÖVŐ MÉRNÖKEIÉRT
SZÖVETSÉG



**Együtt a Jövő Mérnökeiért
Szövetség**
www.ejmsz.hu

**SZTE Természettudományi
és Informatikai Kar**
www.ttik.hu



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

A program a "A Szegei Tudományegyetem készségfejlesztő és kommunikációs programjainak megvalósítása a felsőoktatásba való bekerülés előmozdítására és az MTMI szakok népszerűsítésére" című pályázat (Pályázati azonosító: EFOP-3.4.4-16-2017-00015) keretein belül kerül megvalósításra.

SZÉCHENYI 



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

SZTE TTIK Informatikai Intézet

Intézetünk 1990-ben alakult Informatikai Tanszékcsoporthoz 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 tanulmányaik 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. Külföldi partnerekkel folytatott közös projektjeink révén számos hallgatónk számára tudunk biztosítani lehetőséget külföldi részképzésben való részvételre. 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 a teljes képzési skálát lefedi.



Kiadta: SZTE TTIK Informatikai Intézet

Készítette: Vadai Gergely

Design: Dr. Németh Gábor

Nyomda: Innovariant Nyomdaipari Kft.

Készült: 150 példányban



Együttműködő partnereink:



EGYÜTT
A JÖVŐ MÉRNÖKEIÉRT
SZÖVETSÉG



SZTE TTK
INFORMATIKAI INTÉZET

