

Szegedi Innovatív Informatika Verseny

2017



Program

Időpont	Műszaki informatika szekció (Informatórium)	Informatika szekció (Kamaraterem)
8:30-9:30	Regisztráció	
9:30-11:20	Előadások I.	
9:30-9:40	Megnyitó	Megnyitó
9:40-10:00	Rakéta Csapat: <i>Vízzel hajtott rakéta valós idejű adatgyűjtő és kiértékelő rendszerének fejlesztése</i>	A három grácia: <i>Sophie</i>
10:00-10:20	Lost Theory: <i>Marsjáró robot szimuláció Mindstorms NXT robotokkal</i>	Hiperkockák: <i>Online Betegkarton</i>
10:20-10:40	ERROR: <i>Smart Truck Swarm modell</i>	Bencsik Blanka: <i>Gépjárművezetés biztonság növelése képfeldolgozás alkalmazásával</i>
10:40-11:00	SZVIIT: <i>Next Generation Switch</i>	SootSoft: <i>Company Manager</i>
11:00-11:20	COHESION: <i>A környezeti hatásokra reagáló automatizált intelligens ház LEGO® Mindstorms® EV3 robotokkal</i>	Ali Arsen: <i>Safe Kid Zones</i>
11:20-11:40	Kávészünet	
11:40-13:30	Előadások II.	
11:40-12:00	ESZI szakkör: <i>Intelligens fűtési rendszer</i>	Virtual Entertainment: <i>VR Escape Room</i>
12:00-12:20	Ardudroid: <i>Intelligens szénmonoxid érzékelő és vészhelyzeti segélyhívó</i>	DevQuarter: <i>Fogyasztásmérő</i>
12:20-12:40	"Jelbontók": <i>1/3 oktáv felbontású hangszínekép-mérő (Audió Spektrum Analizátor)</i>	Verseghy Coders: <i>PlatStreamer</i>
12:40-13:00	RC Cloud: <i>RC Cloud: egy léghajó alapú drón</i>	Kébel Zsolt: <i>A TE szótárad</i>
13:00-13:30	Eszközök bemutatása	Szoftverek bemutatása
13:30-14:30	Ebéd	
14:30-15:00	Plenáris előadás (Informatórium)	
15:00-15:30	Eredményhirdetés (Informatórium)	



Tartalomjegyzék

<i>Program</i>	1
<i>Előszó</i>	5
<i>Programbizottság</i>	6
<i>Műszaki informatika szekció</i>	11
Járó Ádám, Jakab István, Gilicze Kristóf: Vízrel hajtott rakéta valós idejű adatgyűjtő és kiértékelő rendszerének fejlesztése	13
Mihalik Dániel, Nyíri Anna Eszter: Marsjáró robot szimuláció Mindstorms NXT robotokkal.....	17
Finta Imre, Szakali Benedek: Smart Truck Swarm modell.....	21
Szvoreny Viktor, Illés Illés Tamás: Next Generation Switch.....	25
Varga Ákos, Hegedűs András: A környezeti hatásokra reagáló automatizált intelligens ház LEGO® Mindstorms® EV3 robotokkal	27
Prantner József, Soós Tamás: Intelligens fűtési rendszer.....	31
Kocsis Zoltán Soma, Putz Koppány Norbert, Nánai Ádám: Intelligens szénmonoxid érzékelő és vészhelyzeti segélyhívó	35
Kardos Csaba, /*Munkácsi Mátyás*/, Koncsik Alex: 1/3 oktáv felbontású hangszínkép-mérő (Audió Spektrum Analizátor)	39
Sztojka Áron, Hargitai Benke, Molnár Barnabás: RC Cloud: egy léghajó alapú drón.....	43
<i>Informatika szekció</i>	49
Német Ágnes Sára, Hunyadi Anna, Ruzsa Kata: Sophie.....	51
Vámosi Tamás, Király Máté: Online Betegkarton.....	55
Bencsik Blanka: Gépjárművezetés biztonság növelése képfeldolgozás alkalmazásával.....	59
Korom Richárd: Company Manager	63
Ali Arsen: Safe Kid Zones	67
Gál Patrik: VR Escape Room	69
Vida Imre: Fogyasztásmérő	73
Kovács Marcell, Szepesi Zoltán: PlatStreamer	77
Kébel Zsolt: A TE szótár.....	81
<i>Informatikus képzés a Szegedi Tudományegyetemen</i>	85
<i>Támogatóink</i>	911

Előszó

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

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

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

Az **Informatika** szekcióba olyan pályamunkákat 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 végül a két szekció zsűrije összesen 12 belföldi és határon túli település iskoláiból érkező 18 csapatot válogatott be a szegedi döntőbe, mely 2017. április 7-én került megrendezésre a Szent-Györgyi Albert Agórában.

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

Szeged, 2017. 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. Kincses Zoltán
adjunktus
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
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.

Szervezők



Vadai Gergely
tanársegéd
SZTE Műszaki Informatika Tanszék



Dr. Kincses Zoltán
adjunktus
SZTE Műszaki Informatika Tanszék



Makan Gergely
tudományos segédmunkatárs
SZTE Műszaki Informatika Tanszék



Dr. Németh Gábor
adjunktus
SZTE Képfeldolgozás és Számítógépes Grafika Tanszék



Bán Dénes
PhD hallgató
SZTE Szoftverfejlesztés Tanszék



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

Műszaki informatika szekció

Vízzel hajtott rakéta valós idejű adatgyűjtő és kiértékelő rendszerének fejlesztése

Rakéta Csoport

Járó Ádám, Jakab István, Gilicze Kristóf

Felkészítő tanáraink: Gilicze Tamás - Méréstechnika

Giliczené László-Kókai Mária – Informatika

Szabó László - Fizika

Csongrádi Batsányi János Gimnázium, Szakgimnázium és Kollégium, 6640

Csongrád, Kossuth tér 1.

1. Bevezetés

A vízzel hajtott rakéta projekt eleinte egy kísérleti projektnek indult fizikából. Olyan rakétát építettünk, melyhez a hozzávalókat egy barkács-áruházban is akár be lehet szerezni.

A rakétát 1/3-ad részéig töltöttük vízzel, majd csatlakoztattuk a kilövő állványhoz. A pumpával levegőt fújtunk a palackba, így ott a nyomás 5-6 Bar lett. Ha a spárgát meghúztuk, akkor a gyorscsatlakozó kioldott és a nyomáskülönbségnek köszönhetően nagy sebességgel kifelé áramlott a víz.

Az impulzus-megmaradásnak köszönhetően a rakétánk nagy sebességgel elindul felfelé. A test és a kilövő állvány, a sérülésmentes visszaérkezést biztosító ejtőernyő még az elmúlt tanévben elkészült Jakab István és több diáktársunk közreműködésével.

Az elsődleges tesztek, manuális mérései és a rakéta ejtőernyőének nyomáskülönbségen alapuló nyitása nem bizonyult megbízhatónak.

Csoportunk ekkor határozta el, hogy az ejtőernyő nyitásához egy barometrikus magasságmérő szenzorral összekötött szervomotort használunk. Ha a szenzor a kilövést követően nyomásnövekedést azaz a magasság tartós csökkenését észleli, akkor a szervomotor kioldja az ejtőernyőnket.

Gilicze Kristóf azon dolgozott, hogy a rakétával együtt felküldhető legyen egy mini szenzoros mérőállomás, ami kétféleképpen képes a szenzorok adatait szolgáltatni SD kártyán rögzítve, így utólag olvasható ki vagy valós időben adóvő berendezéssel.

Járó Ádám idén csatlakozott a csapathoz és a földi állomás szoftverének fejlesztésén dolgozott főleg a föld-rakéta kommunikáció, adatátvitel megvalósításán fáradozott.

A célkitűzés, a részterületek fejlesztésének összehangolása, a tesztelés és a szerkezet tökéletesítése csapatmunkában történt.

2. Probléma megoldásának menete

2.1. A pályamunkánk elkészítésének főbb állomásai

2016. március-április: megépül a rakéta test, a kilövő állvány, elkészül az ejtőernyő.

2016. május: több alkalommal tesztelés szabad térben.

2016. májustól szenzoros megfigyelés tervezése, prototípus elkészítése tesztelése szimulált helyzetekben.

2016. június-augusztus: SD-kártyás módszer hibakeresés, tökéletesítés.

2016. szeptember-december: informatikai, elektronikai ismeretek bővítése

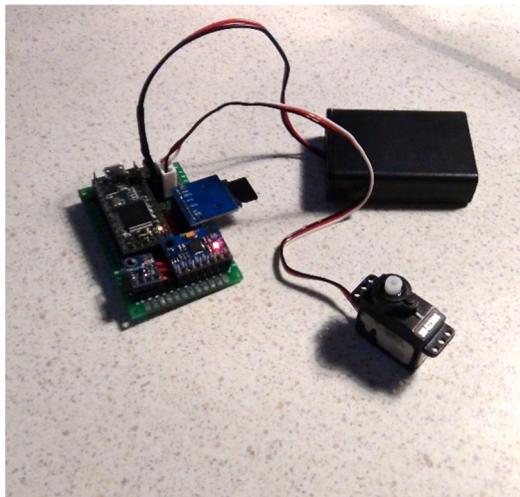
2017. január-február: informatikai fejlesztőcsapat megalakulása, célkitűzés, munkamegosztás, kommunikációs technológia kiválasztása, hardverfejlesztés, programozás, nem valós helyzetekben tesztelés.

2017. március: a pályamunka tesztelése, fejlesztése, dokumentálása.

2.2. Az adatgyűjtő modul fejlesztésének leírása

2.2.1. Első prototípus: adatrögzítés SD kártyára

A szenzormodul egy MPU 6050-es, amely I2C protokollon keresztül gyorsulás és giroszkóp adatokat olvas ki. A BMP 180 barométer adatait a barometrikus magasságmeghatározáshoz használjuk fel. Az adatokat a Teensy mikrokontrollerrel dolgozzuk fel, majd SPI protokoll felhasználásával microSD kártyára rögzítjük. Ebben a szakaszban különböző rádiós megoldások (RF24, ESP8266, HC05) tesztelése is zajlott. Az ejtőernyőt egy szervo motorral nyitjuk, itt felhasználjuk a mért adatokat.



1. ábra. Az adatgyűjtő első verziója

2.2.2. Második prototípus: valós idejű adatfeldolgozás

Ha a rakéta mozgását valós időben szeretnénk pl. egy monitoron követni, akkor az adatokat a földi bázisra kell elküldeni. A rakétán hordozott vezérlő a magasság adatokat meghatározza a mért nyomásadatokból.

A Teenseyhez illesztett wifi modul közvetlenül is kapcsolódhatna a földi számítógéphez, de a hatótávolság növelése érdekében egy routert iktattunk közbe. A nyers adatokat a földi számítógép eltárolhatja későbbi adatfeldolgozás céljából vagy valós időben megjelenő grafikon kirajzolására felhasználhatja.

2.2.3. Harmadik prototípus: saját készítésű NYÁK-lap

A harmadik prototípushoz a nyomtatott áramkört saját magunk készítettük el. Két technológiát próbáltunk ki: CNC géppel és kémiai módszerrel állítottuk elő az áramköri lapot.

2.2.4. A szoftver

Az adatgyűjtő mikrokontroller kódját C++ nyelven írtuk. A földi feldolgozóegységet egy Python script vezéri. Az összes szenzorral, és programmodullal a többi részegységtől függetlenül, külön ismerkedtünk meg.

A kommunikáció végleges megoldásra egy ad-hoc wifi hálózatot létesítettünk, amin TCP csomagokkal küldünk a rakétáról valós idejű adatokat a számítógépre.

A számítógépen ezeket az adatokat fogadjuk és különböző módon feldolgozzuk és grafikus úton is elérhetővé tesszük a „rakéta-mérnökeinknek”.

3. Elért eredmények

A rakéta által hordozott mérési adatgyűjtő eszköz három fejlesztési szakaszt is megélt. Elkészült a működtető szoftver és folyamatban van a mozgást vizuálisan is szemléltető felület fejlesztése, működik az ejtőernyő nyitó szerkezet.

A fejlesztőmunka során adódtak nehézségek, amelyeket idővel megoldottunk. Ilyen volt pl. az SD kártyára írás kezdeti problémái (nem megfelelően választott áramforrás okozta), a grafikont rajzoló Python modul lassúsága (másik modul választásával oldottuk meg).

Időközben több új ötlet is eszünkbe jutott. Ha a palack belsejében a nyomás-változást is mérjük szenzorokkal, akkor az így kinyert adatok elemzése a fizika órákon hasznos lehet. Pl. kitapasztalható, hogy a palackban mérhető kezdeti nyomás és az emelkedési magasság között milyen kapcsolat van. Optimalizáláshoz is felhasználható: adott nyomás esetén mennyi vizet töltünk a palackba, hogy maximális legyen az emelkedési magasság?

Ha a mérőberendezés „túléli” a küldetést, akkor újra felhasználható.

Sem a vizesrakéta projekt, sem a pályázatban leírt műszaki-informatikai fejlesztésünk még nem került bemutatásra nagyobb nyilvánosság előtt.

Marsjáró robot szimuláció Mindstorms NXT robotokkal

Lost Theory

Mihalik Dániel, Nyíri Anna Eszter

Felkészítő tanár: Kiss Róbert

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

1. Bevezető

Projektünk szorosan kapcsolódik a térinformatikához, annak modellezés részéhez. A térinformatika a GIS, a Geographical Information System (tkp. földrajzi információs rendszer) magyar elnevezése. Ez olyan számítógépes rendszer, amelyet földrajzi adatok tárolására, elemzésére, a földrajzi jelenségek megfigyelésére alakítottak ki. Három fő tudományág együttes használatát foglalja magába, a térképészetét, a geológiát és az informatikát. A geoinformatika rendkívül nagy jelentőséggel bír a természeti erőforrások kutatásában, állapotának figyelésében; a közigazgatásban; ugyanakkor a közlekedési, szállítási, honvédelmi feladatok megoldásában. Egy területről készült fényképet például képi, domborzati, közlekedési, lakossági, politikai viszonyai alapján is elemezhetünk. A rendszer felhasználói közé a 21. században már nem csak a tudósok és érdeklődők szűk köre tartozik, a hétköznapi ember is mindennap találkozik vele. Modellünk egy térinformatikai probléma emberi beavatkozás nélküli megoldását mutatja be.

2. A projektfeladat és a megvalósítás

2.1 A szimuláció koncepciója

A projekt a következő szituáció alapján készült: Tegyük fel, hogy egy robot egy ember számára járhatatlan környezetben (pl. a Mars felszínén) elakad, vagy nem talál a cél eléréséhez megfelelő útvonalat. Ilyen esetben egy műhold feltérképezheti a robot közelében lévő területeket, például fotót készíthet a felszínről, majd a fotó elemzése alapján megtervezhető az útvonal. A műhold megfelelő informatikai kapacitás hiányában nem feltétlenül képes az adatok feldolgozására, illetve szerencsésebb, ha különválasztjuk a képkészítést és az elemzést végző hardvert, hiszen távoli, idegen környezetben a meghibásodás esetén így csak az egyiket kell pótolni. Az elkészült képet tehát el kell küldenie például a felszíni központhoz, ahol a kép elemzéséből nyert információk alapján utasításokat készítenek a robot útvonalát illetően. A parancsot és az útvonalat a bázis továbbítja a terepen mozgó robotnak, amely így már végigmehet a kijelölt útvonalon és teljesítheti feladatát.

2.1 A kivitelezés

A marsjáró robot mozgását és a műholddal, illetve a központtal való kommunikációját modelleztük. A bázist, a műholdat és a terepen lévő robotot 3 Mindstorms NXT téglá helyettesíti. A robotok Bluetooth-on keresztül kommunikálnak egymással, a rajtuk futó programok NXC, C-hez közeli karakteres programnyelven íródtak.

2.2 A modell hardver része

A szimulációban a hardver három elemből áll: a műholdból, a bázisból és a marsjáró robotból (rover). A terepen lévő robot konstrukcióját különlegesre terveztük, mert tapasztalataink alapján tudjuk, hogy robotjaink egyik legnagyobb hibája a pontatlan fordulás, valamint a szenzorok méréseinek pontatlan értékei. Erre a problémára jelent megoldást a robot két keretből álló váza. Erre a két keretre kerekeket szereltünk úgy, hogy a különböző vázak kerekei egymásra merőlegesen álljanak. Így a robot négy irányban képes haladni (fordulásai mindig 90°-osak) attól függően, hogy melyik keret éri a földet, és hogy az azon lévő kerekek merre forognak. A konstrukcióról készült fényképek az 1. ábrán láthatók. Mivel a rover az utasításokat Bluetooth-on keresztül kapja, a tájékozódáshoz nincs szüksége szenzorokra. A robot feladata, hogy fogadja a bázis üzenetét és az utasításoknak megfelelően haladjon végig az útvonalon. A műhold szerepét szintén egy MINDSTORMS NXT téglá tölti be. Feladata, hogy feltérképezze a rover környezetét, majd ezen adatokat a bázisnak továbbítsa. A központot jelképező téglá feladata az adatok feldolgozása, az elemzést követően a legoptimálisabb útvonal kiválasztása, majd ezek továbbítása a marsjárónak.

2.3 A modellezés menetének részletes ismertetése

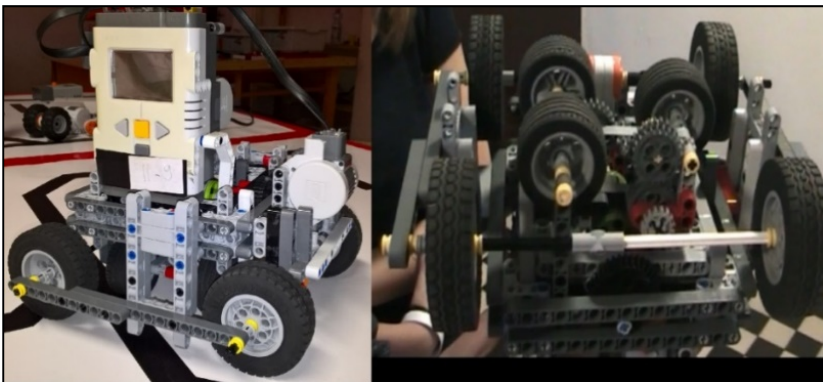
A kép készítését manuális adatbevitellel modellezzük, mivel az NXT téglá nem rendelkezik kamerával, illetve a hardver nem tartalmaz megfelelő méretű memóriát a kép tárolásához. A műholdat jelképező téglára 4 touch szenzor (érintés-ütközésérzékelőt) helyeztünk fel, amelyek a gombok szerepét töltik be (lásd 2. ábra). A program elején a képernyőre egy 5x5-ös négyzetrácsot rajzolunk ki, amely a rover környezetét szimbolizálja (a négyzetrács természetesen tovább bővíthető). A négyzetrácson egy kör jelzi a kurzort, amit a fent említett touch szenzorokkal mozgatunk (a négy gomb a jobb, bal, fel, le irányoknak felel meg). Elsőként azokat a területeket kell megjelölni, amelyekeken akadály van (pl. földrajzi képződmények, hegyek, szakadékok), tehát a marsjáró nem tud rajtuk áthaladni. Ezeket a képernyőn téglalapok jelzik. Mikor az összes akadály helyzetét meghatároztuk, véglegesítenünk kell azokat. Ezt a téglán lévő balra gomb megnyomásával tehetjük meg. Továbbá meg kell adnunk a rover jelenlegi helyzetét és a célmezőt. Ezt is a fenti módszerrel tehetjük meg. Mikor ez is megtörtént, a műhold az információkat

átalakítja egy könnyen továbbítható formátumba, és átküldi a földön lévő bázisnak. A kapott adatokat feldolgozva a központ kiszámítja a legoptimálisabb útvonalat. Ezt az ún. flow-algoritmussal teszi, amely jelen esetben 3 lépésből áll. Ezek közül az első, hogy a négyzetrácsban szereplő mezőket beszámozza a starttól való távolság alapján, azaz hogy az adott pontba való eljutáskor legkevesebb hány mezőt kell érinteni. Másodsor, a számok alapján meghatározza a legrövidebb útvonalat. Ez alapján ír egy kódot, amelyben a rover mozgásához szükséges számokat tárolja. Ebben a kódsorban három különböző szám lehet, 1, 2 és 3. Az 1 jelentése, hogy a robot balra vagy felfele mozogjon, a 2 pedig a jobb és le irányokat jelzi attól függően, hogy éppen melyik keret van a földön. Ha a következő szám a 3, a rover keretet vált, vagyis az előzőhöz képest másik keréksoron fog közlekedni, így haladási iránya az előzőhöz képest merőleges lesz. Miután ezt a három lépésből álló feladatot elvégezte, a bázis átküldi a kódsort a marsjárónak, amely ezen kód alapján mozog, tehát nincs szüksége szenzorokra. A teljes rendszer működését modellező terepről készült fénykép a robotokkal a 3. ábrán látható.

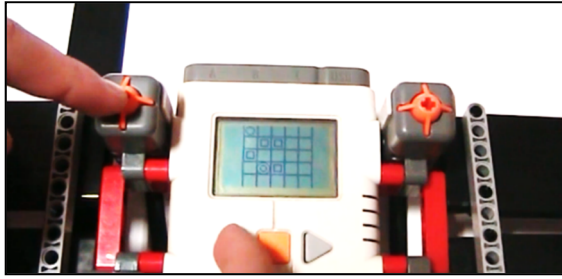
2.4 A kommunikáció

A modellezés alapeleme a kommunikáció, hiszen nélküle nem jöhetne létre információcsere az objektumok között. A robotok 2 Bluetooth csatornán keresztül küldik egymásnak az információkat. A bázis áll kapcsolatban a műhoddal és a roverrel is, utóbbi kettő között csak közvetett kommunikáció van. A felépített Bluetooth kapcsolat master-slave elvű, tehát van egy kitüntetett eleme a kommunikációs láncnak (master), amelyen keresztül a többi eszköz üzeneteket tud küldeni. A kommunikáció az egyes partnerek között egyirányú. A műhold küld üzenetet a bázisnak, majd a bázis elvégezve a számításokat továbbítja az adatokat a rover-nek.

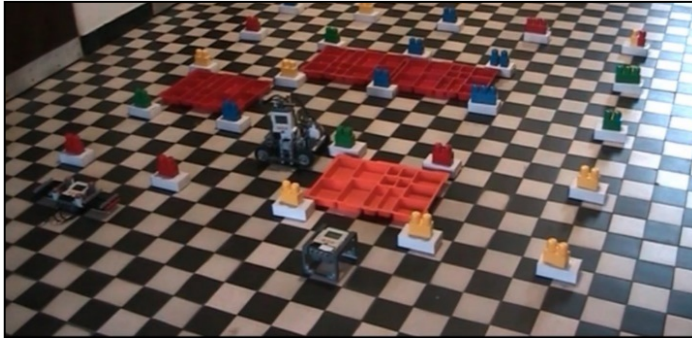
2.5 Ábrák



1. ábra. A marsjáró konstrukciója



2. ábra. A környezeti adatok manuális megadása



3. ábra. Az elkészült modell működése a terepen

3. Eredmények

Az ismeretlen környezeti viszonyok közötti mozgás egyik lehetősége a szenzorokkal történő tájékozódás, a másik lehetőség pedig a térinformatika általunk modellezett része, tehát fényképek alapján történő adatelemzésre épülő tájékozódás. Ez utóbbi esetben az egyes egységek közötti kommunikáció nehezen elkerülhető.

A projektben sikerült megvalósítanunk egy olyan modellt, amely működőképes, mind hardver, mind szoftver, mind kommunikációs területen. A megépített konstrukció képes volt bejárni az utasításoknak megfelelő útvonalon a területet. Az egyszerű térkép alapján az adatelemzést végző eszköz minden esetben megtalálta a legoptimálisabb útvonalat. Természetesen bemenő adatként valós fényképet használva komolyabb hardver és szoftver háttérre lenne szükség, de a képfeldolgozás is történhet hasonló elven, csak jóval nagyobb mátrixban kell tárolni az egyes adatpontok tulajdonságait.

Projektünk első változatát bemutattuk a VII. Mobilrobot Programozó Országos Csapatverseny egyik részterületeként, ahol összetett 2. helyezést értünk el 2017 februárjában.

Smart Truck Swarm modell

ERROR

Finta Imre, Szakali Benedek

Felkészítő tanár: Kiss Róbert

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

1. Bevezetés

Napjaink legelterjedtebb szárazföldi szállítóeszközei kétségtelenül a kamionok. Gyorsaságuk és rugalmasságuk kitűnő eszközzé teszik őket ezen feladatok elvégzésére, viszont van egy komoly hátrányuk: magas fogyasztásuk miatt fenntartásuk drága, ráadásul a nagy károsanyag-kibocsátás a környezetet is komolyan veszélyezteti. Ezért a szállítmányozó cégek és kamiongyártók legfontosabb preferenciái közé tartozik a kamionok fogyasztásának csökkentése.

A fogyasztás csökkentésére több módszer is kínálkozik. Az elmúlt években számos gyártó kísérletezett már alternatív üzemanyagokkal és jobb hatásfokkal rendelkező motorokkal, de talán a legnagyobb potenciál a járművekre ható légellenállás csökkentésében rejlik. Egyes kutatások szerint a kamion üzemanyag felhasználásának akár 50%-át is a légkör ellenében kifejtett munka emésztheti fel, ennek csökkentésével tehát javíthatóak a fenntartási költségek.

A légellenállás csökkentésére irányuló kutatások manapság két fő irányba tartanak. Az egyik lehetőség a kamion formájának megváltoztatása. A nagy, szögletes dobozforma rendkívül rossz légellenállási mutatókkal bír, az elmúlt évtizedekben mégsem történt számottevő változás a járművek külalakjában. A másik lehetőség az érték javítására kamionrajok kialakítása. Ezen módszerrel a kamion az előtte haladó szélárnyékába kerül, így jelentősen csökkenti saját fogyasztását. A módszer hátránya, hogy az ezt használó sofőrök nem tudnak megfelelő követési távolságot tartani, így alkalmazása igen balesetveszélyes.



1. ábra. A Smart Truck Swarm modell robotkamionjai

Mi ez utóbbi módszert választottuk projektünk témájául. Az általunk létrehozott Smart Truck Swarm modell három, autonóm működésre képes kamionból álló robotraj, amelynek tagjai érzékelők és a köztük felépített kommunikációs hálózat alapján képesek követni egymást. A kamionokat az 1. ábra mutatja.

2. Probléma megoldásának menete

2.1. Projekt kiválasztása

A megvalósítás első lépése a megoldás megtervezése és az elméleti háttér kialakítása volt. A rendszer ideális kialakításához elsősorban kutatómunkát végeztünk. Utánanéztünk a kamiongyártók jelenleg e téren folytatott kísérleteinek és az általuk használt módszereknek. A Mercedes 2016-ban indított útnak három rajba állított kamiont Stuttgartból Rotterdamba (a kamionok a 2. ábrán láthatóak). A kamionok automatikusan követték egymást az autópályán, és sikeresen eljutottak céljukhoz. A Mercedes által használt megoldás inspirálta a saját kommunikációs hálózatunk kialakítását.

A kutatás után az előkészítés következő lépése volt saját modellünk megtervezése. Az eredeti terv szerint a kamionok ultrahangos távolságmérővel határozták volna meg az előttük haladó távolságát és a kanyarodás irányát. A kommunikációt peer-to-peer hálózat formájában képzeltük el, így minden kamion csak az előtte- és mögötte haladóval kommunikál. Ez rugalmas és – mivel nincsenek benne kitüntetett csomópontok – stabil rendszert jelent. A hálózaton keresztül a robotok szinkronizálják sebességüket és állapotukat.



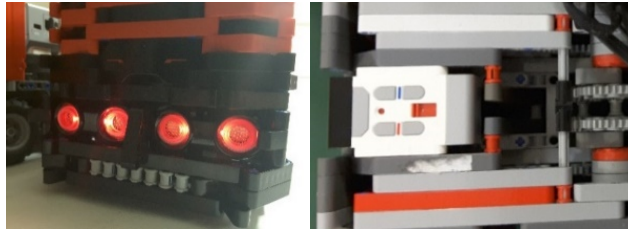
2. ábra. A Mercedes kísérlete

2.2. Modellek

A koncepció megtervezése után következett a kamionmodellek elkészítése. A járműveket LEGO-ból építettük meg, és különösen ügyeltünk a minél élethűbb paraméterekre a minél realisztikusabb szimuláció érdekében. A modellek így fejlett kormányművel, háromtengelyes kialakítással és két hajtott tengelyt biztosító hajtáslánccal rendelkeznek. A hajtásért két EV3 Large motor a felelős, a kormányzást egy harmadik, EV3 Medium motor végzi.

Minden modellbe került két ultrahangos érzékelő az előttük haladó mozgásának követésére. Mivel azonban ezek nem bizonyultak elég pontosnak

a kanyarok érzékelésére, egy-egy infravörös érzékelőt is elhelyeztünk a modellek elején, a hátuljukon pedig egy-egy infravörös jeladót. Ez utóbbiak ráadásul egyedi csatornán sugároznak, így a kamionok követése mellett azok azonosítására is lehetőséget adnak. A szenzorokat a 3. ábra mutatja be



3. ábra. Ultrahang- és infraszenzorok illetve infravörös jeladó a kamionokon

2.3. Működés és programok

A robotraj első tagját – a sofőrt szimulálva – távirányítóval irányítjuk. A mögötte haladó kamionok ultrahangszennozorral mérik meg az előttük haladó távolságát. A járművek az előttük haladó sebességéből és távolságából kiszámolják saját sebességüket, amely az ideális követési távolság tartásához szükséges, a kanyarodás mértékét pedig a két ultrahangszennozor által mért értékek különbségéből, majd ezt pontosítják az infravörös szennozor mérése alapján, amely az előtte haladó kamion hátulján elhelyezett jeladó irányát határozza meg. A korrekcióra azért van szükség, mert az ultrahangszennozorok – az eredeti elképzeléssel ellentétben – nehezen érzékelik az ilyen kis távolságkülönbséget, valamint csak kis látószöggel rendelkeznek, így könnyen szem elől tévesztik a követendő kamiont.

A koncepció rugalmassága érdekében eredetileg minden roboton azonos program futott volna. Ez végül a kommunikáció nehézségei miatt nem valósulhatott meg, de így is csupán az elől haladó jármű programkódja tartalmaz extra parancsokat a kommunikáció vezérlése és a távirányítás céljából. Minden program két párhuzamos programszálon fut, az egyik a kommunikációt kezeli, a másik a motorok vezérlését látja el. A szimultán működés lehetővé teszi, hogy a robot vezérlése folyamatosan, valós időben frissülő adatok alapján történjen.

2.4. Kommunikáció

Az eredeti elképzelés a kommunikációs hálózat felépítésére egy Peer-to-Peer (P2P) alapú hálózat volt, ahol minden robot csupán az előtte és mögötte haladóval kommunikál. Az EV3-as robotok szoftvere azonban csak a master-slave alapú piconet hálózatok létrehozását támogatja, ezért az elől haladó master robot programját az üzenetek továbbításához szükséges kódsorokkal kiegészítve egy szimulált P2P hálózatot hoztunk létre. Bár az üzenetváltás sebessége és a rendszer dinamikussága szempontjából ez nem az elérhető

legjobb megoldás, de a modell továbbfejlesztése és esetleges platformváltás esetén ez a felépítés biztosítja a lehető legszélesebb mozgásteret.

A kommunikáció során minden robot két adatot küld, és ugyanilyen két adatot fogad. Az egyik adat az előtte haladó kamion sebessége. Ez alapján határozza meg saját sebességét a másiktól mért jelenlegi távolságának függvényében. Ezután saját sebességét a mögötte haladónak továbbítja. A másik fogadott adat a mögötte haladó robot állapota. Amennyiben egy robot szem elől veszti az előtte haladót, egy STOP állapotjelzést küld neki, ami azt tovább küldi az őt megelőzőnek stb., így a leszakadt rész előtt haladó robotok mind STOP jelzést kapnak. A STOP jelzést kapott robotok megállnak, ezzel szétszakadás esetén a lánc elöl járó részei leállnak, hogy bevárják a leszakadt részt.

3. Elért eredmények

Az elkészült projekt egy-két (fentebb említett) kivétellel sikeresen valósította meg az eredeti elképzelést. A legfontosabb elért eredmények:

- Valóságghú, a lehetőségekhez mérten pontos modellek elkészítése
- Szenzormérésekre és megosztott adatokra támaszkodó vezérlőrendszer kifejlesztése a járművekhez
- A rendszer működik, a kamionok hatékonyan képesek követni egymást
- A modell egy változatát bemutattuk a VII. Mobilrobot Programozó Országos Csapatversenyen, ahol elnyertük a Legjobb kommunikációs robotprogram díjat

A téma iránt napjainkban mutatkozó kiemelt érdeklődés is mutatja, mekkora potenciál rejlik a területre irányuló kutatásokban. Az általunk létrehozott rendszer kitűnő alternatívát nyújt a felvázolt probléma megoldására, a felhasznált szenzorok lehetőségeinek nagyarányú kihasználása pedig segíthet a technológia költségeinek csökkentésében, hiszen így olcsóbb szenzorok használatával is ugyanolyan, vagy akár jobb eredmények érhetőek el, mint drágább társaikkal. A modell továbbfejlesztése révén pedig nem csak a kamionos szállítmányozás költségei csökkenthetőek jelentősen, de az ezzel elért üzemanyagfogyasztás-csökkenés kisebb károsanyag-kibocsátással jár, ami környezetvédelmi szempontból is fontossá teszi az ilyen irányú fejlesztéseket.



4. ábra. A Smart Truck Swarm működés közben

Next Generation Switch

SZVIIT

Szvoreny Viktor, Illés Illés Tamás

Felkészítő tanár: Kőrösi Gábor

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

1. Bevezetés

Manapság egyre inkább teret kapnak az okos eszközök. Ennek egyik ékes példája az okos LED izzók rohamos terjedése. Számos előnnyel jár, ha otthon ilyen fajta lámpákat használunk, mivel hosszabb az élettartama, gazdaságosabb, környezetbarátabb, minőségesebb, és okosabb mint egy átlagos izzó. Emellett azonban nyilvánvaló, ha van egy ilyen izzónk, akkor azt szeretnénk valamivel irányítani, ki-és bekapcsolni. Mely értelemszerűen egy villanykapcsoló kellene, hogy legyen.

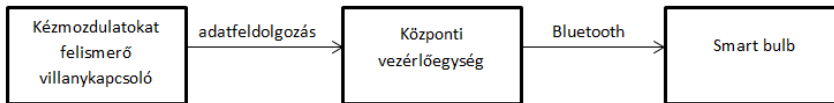
Az első villanykapcsolót 1884-ben telepítették Newcastle-ben. Azóta változott bármennyire is, néhány finomítást és designt leszámítva? Nem igazán, ennek az eszköznek a falon az a funkciója, hogy megszakítsa az áramkört kedvünk szerint, esetleg passzív elektronikai eszközökkel fojtsa annak áramerősségét, a világítás erősségét. Alapvetően még a fény kibocsájtást biztosító eszközök futurisztikusak lettek, annak vezérlése mit sem változott. E gondolatmeneten haladva döntöttünk, úgy hogy az okos égőkhöz okos villanykapcsolók fejlesztésébe kezdünk.

2. Probléma megoldásának menete

Amennyiben szeretnénk lecserélni a régi, unalmas „on-off” típusú egyszínű égőinket, akkor az interneten számos olyan izzót találunk, amelyet egy egyszerű, Bluetooth-alapú applikációval tudunk működtetni, vezérelni. Azonban ezek ilyen irányú kezelése nem túl praktikus, mivel megtörténhet az, hogy nincs nálunk mobiltelefonunk, illetve egy-egy applikáció beindítása elepszteően sok időt vehet igénybe.

Az általunk fejlesztett villanykapcsoló a mai okos fénytestek és applikációk ötvözeteként okos vezérlést kíván biztosítani, mely az eddigi körülményes irányítást hivatott felváltani. Eszközünkkel a történelmi „on-off” kapcsoló utódjaként képesek vagyunk számtalan további utasítás kiadására, melyhez csupán kézmozdulatokra van szükségünk. Az elkészült kapcsolóval mindennemű fizikai érintkezés nélkül fel-és lekapcsolhatjuk izzónkat, megváltoztathatjuk annak színét és intenzitását, melyhez egyfajta „Non-physical contact” technológiát és a Bluetooth technológiát vettük alapul.

A fejlesztésünk célja az volt, hogy egy innovatív hozzáállással egy hétköznapi eszköz újragondolásával megkönnyebbítsük a mindennapi életnek e kicsi, ám nem elhanyagolható részét. Az eszköz további előnyeként említésre méltó az a nem elhanyagolható tény, hogy az NFC technológia által a villanykapcsolók higiénikusabbak lennének, mely túlmutathat az otthoni, hétköznapi felhasználás területén.



1. ábra. Eszközünk algoritmus

3. Elért eredmények

Tehát, jelenlegi alapkoncepcióban adott egy LED „okosizzó” és egy kézmozdulatokkal irányított kapcsoló, melyek Bluetooth-on keresztül kommunikálnak. Ezt, tervünk szerint, szeretnénk bővíteni fotorezisztorokkal. Ezeknek a funkciója egyszerű: egy adott helyre (pl. ablakba) kihelyezve méri a fényességet, és az alapján szabályozza izzónk fényének intenzitását. Ez elég kézenfekvő, mivel a helyiségben mindig olyan fényerő lesz, ami egészséges szemünknek.

Ezután egy nagyobb rendszert képzeltünk el, amelyben több izzót is kezelhetnénk egy kapcsolóval, szintén néhány kézmozdulattal. Ez egy olyan menürendszert igényel, amely egyszerű, a felhasználó számára könnyen megtanulható és nem utolsósorban gyors is.

Egy olyan ötletet is kilátásba helyeztünk, amely bizonyos ideig elmentené napi, heti világításhasználatunkat. Később pedig ehhez igazodna magától, ennek köszönhetően pedig, egy futurisztikus, önműködő villanykapcsolót kaphatnánk, ami hozzánk igazodik. Évszakonként lehetne változtatni ezeket a beállításokat, naplementétől és napkeltétől függően.

A környezeti hatásokra reagáló automatizált intelligens ház LEGO® Mindstorms® EV3 robotokkal

COHESION

Varga Ákos, Hegedűs András

Felkészítő tanár: Kiss Róbert

Bányai Júlia Gimnázium, 6000 Kecskemét Nyíri út 11.

1. Bevezetés

A projekt célja egy olyan automatizált ház (Smart Home) és az azt üzemeltető rendszer létrehozása, amely emberi beavatkozás nélkül önműködően képes irányítani belső rendszereit, mint a fűtést, a hűtést, a világítást, a légtechnikát (például: szellőztetés) és a biztonsági rendszereket.

A Smart Home-ok a hagyományos családi házaktól eltérően energia- és pénztakarékosabbak. Egy intelligens házban lakók a fűtésre költött összegnek 12%-át, míg a hűtésre költött összegnek 15%-át tudják megspórolni.¹ Ezen túl az intelligens házak további előnye, hogy a lakók kényelmesebben és időtakarékosabban élhetnek benne, mert nem kell a ház rendszereit manuálisan kezelni.

Emellett a Smart Home elkerüli az emberi figyelmetlenségből bekövetkező károkat. Például egy szeles időben nyitva hagyott napellenző könnyen megrongálódhat.

Ilyen problémák megoldására hoztuk létre ezt a projektünket.

2. Probléma megoldásának menete

2.1. A legnagyobb energia megtakarítás elérése

A projektünkben egy olyan rendszert szerettünk volna létrehozni, ami a lehető legjobban irányítja a belső rendszereit és a lehető legnagyobb energia és pénzmegtakarításra képes.

Sok készülék létezik már, amivel a lakók életét próbálják kényelmesebbé tenni, mint például mozgásra aktiválódó világítás, vagy robotizált tisztítóeszközök. Ezek használata viszont még nem vezet nagymértékű energia megtakarításhoz.

Felismertük, hogy a legtöbbet úgy lehet megtakarítani egy Smart Home-mal, ha a ház az aktuális időjáráshoz alakítja működését, azaz az intelligens ház pontos meteorológia adatokkal dolgozik. A megfelelő időben és csak a megfelelő mennyiségben fogyasztaná az energiát a fűtésre, a hűtésre,

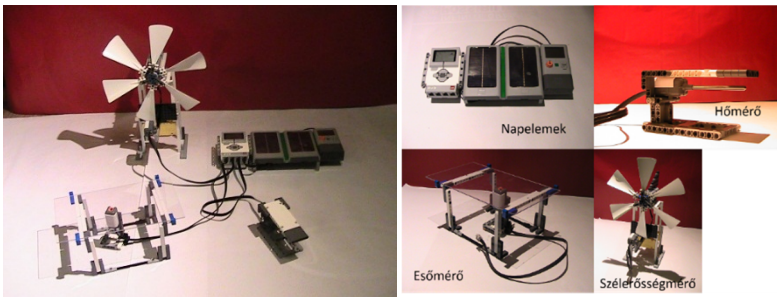
¹ A számítást a Consumer Technology Association tanulmányai alapján készítettük.

légkondicionálásra, világításra és tudná azt, hogy mikor alkalmas a kinti idő a szellőztetésre. Emiatt kezdtük el építeni az intelligens házat, ami az előbb leírt elvek figyelembe vételével működik. Ennek megvalósításához LEGO® Mindstorms® EV3 robotokat használtunk.

2.2. A rendszer egységei

2.2.1. A meteorológiai állomás

Ahhoz, hogy az intelligens ház működés közben folyamatosan tudja mérni a külső időjárási adatokat, egy meteorológiai állomást építettünk. A meteorológia állomás az 1. ábrán látható. Mérőeszközeivel (szenzorjaival) méri az időjárási értékeket. Az általunk készített állomás képes meghatározni a fényviszonyokat, a hőmérsékletet, a szélereősséget és azt, hogy esik-e az eső. A szenzorok az 1. ábrán láthatók.



1.ábra: A meteorológiai állomás és szenzorai

A meteorológiai állomáson található napelemek a napszak meghatározásáért felelősek. A két napelem panelt egy feszültségmérőhöz kötöttük. A feszültségmérő a meteorológiai állomás vezérlőjének (a téglának) továbbította a mért feszültséget. Nappal 1 V - 1,5 V, este pedig 0 V-ot feszültséget mértünk. Így határoztuk meg, hogy nappal van-e vagy éjszaka.

A hőmérővel 3 intervallumban mértük a hőmérsékletet. 20°C alatt hidegnek, 20°C-28°C közötti intervallumban kellemes hőmérsékletnek, míg 28°C felett meleg definiáltuk a hőmérsékletet. A szenzort a közvetlen napfénytől való védelem miatt egy LEGO® elemmel fedtük le.

Az általunk épített esőmérő szenzor hasonló módon működik, mint az autók szélvédőjén található esőérzékelő, ami az ablaktörlőket aktiválja. Egy átlátszó plexi lap alatt egy ún. Colour Sensor-t helyeztünk el, ami Reflected Light Intensity módban méri a saját fényének a plexiről visszaverődő százalékos értékét. Amint a plexire, a fényszenzor fölé egy csepp vizet rakunk (ami az esőt szimulálja) a fénynek egy nagyobb része az esőcsepp fénytörése miatt távozik, így a szenzor már kisebb értéket mér. Problémaként merült fel az, hogy a kinti fény megzavarja a szenzort, ami a saját fényének a szenzorba

visszaérkező százalékos értékét méri. Ezt a problémát úgy tudtuk megoldani a robot bizonyos időközönként újrakalibrálja a szenzort az aktuális fényviszonyokra.

A szintén saját készítésű szélmérő szenzor az alábbi elven működik: A fehér szellapátok mögé egy fekete LEGO® lapot helyeztünk. A szellapátok elé pedig egy Colour Sensor-t Reflected Light Intensity módban. Ha szélcsend van, akkor a lapátok egy helyben állnak. Ilyenkor a színszenzor előtt vagy a fekete lap, vagy a fehér lapátok állnak. Szeles időben viszont a szélkerék forgása miatt váltakozva van előtte a fehér és a fekete szín. A robot 5 másodpercen keresztül 0,1 másodpercenként méri a fényt. Amikor szélcsend van, akkor a mért értékek átlaga megegyezik a fehér, vagy a fekete színén mért értékkel. Szeles időben az 5 másodperc alatt a fehér és a fekete szín fényértékeit is méri, így ezek átlaga nagyban eltér, a feketén, vagy a fehéren mért értékektől. Ezek alapján a meteorológiai állomást irányító robot meg tudja állapítani, hogy éppen fúj-e a szél, vagy sem.

2.2.2.A Smart Home

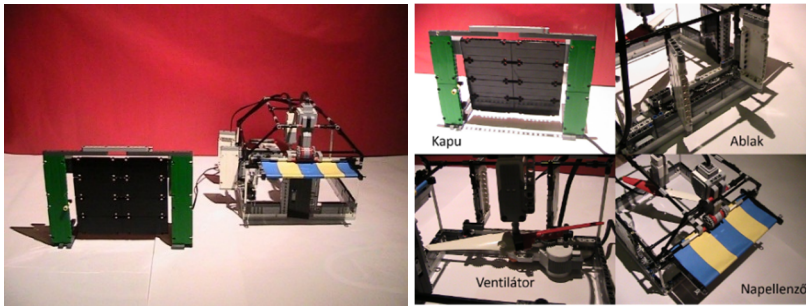
A rendszer másik fő elme a ház. A ház a 2. ábrán látható. A házat vezérlő robot megkapja az időjárásra vonatkozó információkat kódolt formában. Ha a ház egyik automatizált részének összes időjárástól függő működési feltétele egyszerre bekövetkezik, akkor ez a rész aktiválódik. Ez emberi beavatkozást nem igényel. Az otthon a következő önműködő részekkel rendelkezik. A részekről készült képek a 4. ábrán találhatóak.

A ház kapuval rendelkezik. A házban lakók autójának megérkezésekor az autó ultrahangos távolságérzékelőjével (Ultrasonic Sensor-jával) érzékeli a kaput, majd jelet küld a háznak, hogy nyissa ki azt. Mikor sikeresen beállt, az autó újra jelet küld a háznak, hogy becsukhatja a kaput. Hasonló elven működik, amikor az autó elhagyja a házat.

Az ablak a ház szellőzését és természetes megvilágítását szolgálja. Mindig nyitva van, amikor otthon vannak a lakók, valamint nincs szél, eső vagy túl meleg. Az ablak csak nappal van nyitva.

A ventilátor feladata a ház hűtése, amikor nagyon meleg van függetlenül attól, hogy esik-e az eső, fúj-e a szél, vagy, hogy nappal van-e. Ezen berendezés energiatakarékossági okokból csak akkor működik, amikor a lakók otthon vannak.

A napellenző feladata az árnyékolás. Ez a falakat és az alatta tartózkodókat védi a napsütéstől. Akkor működik, amikor nagyon meleg van. Ha a fúj a szél, vagy esik az eső akkor automatikusan behúzódik, hogy meggátolja a károsodását.



2. ábra. A Smart Home és részei

2.2.3. Az autó

A robot autóval a lakók otthonlétét, vagy éppen távollétét szimuláltuk. Az autó (és egyben a lakók) távollétében a ház energiatakarékos üzemmódba kapcsol. Kikapcsolódik a ventilátor, az ablak és a napellenző behúzódnak. Amikor a lakók hazaérnek a ház rendszeri automatikusan bekapcsolódnak.

2.3. Az egységek kommunikációja

A Smart Home működéséhez elengedhetetlen az egységek kommunikációjának programozása. A Bluetooth alapú kommunikáció az alábbi elven működik. A program elindításakor a Smart Home automatikusan felveszi a kapcsolatot a meteorológiai állomással és az autóval. A sikeres kapcsolódás után a meteorológiai állomás kódolja az időjárásra vonatkozó adatokat és továbbítja azokat a Smart Home-nak. A Smart Home értelmezi a kódot és ehhez alakítja a működését.

3. Elért eredmények

Így bizonyítást nyertünk arra, hogy a jelenleg folyamatosan nagy ütemben fejlődő okos ház technológiák még több előnnyel ruházhatók fel, ha a házak közvetlenül a környező meteorológiai mérő állomások adataival dolgoznak. Ehhez, mint az a projektünkéből kiderült elengedhetetlen a vezérlő és mérőegységek szoros, emberi beavatkozást nem igénylő kommunikációja. Ez a két egység esetleges nagy távolsága miatt interneten keresztül valósulhat meg.

A projektünk jól példázza, hogy az így létrehozható rendszerek nem csak a lakók kényelmét szolgálják, hanem pénz és energiatakarékosak is.

Környezetünk védelme egyre nagyobb szerepet kap a jövőbeni tevékenységeink során ezért nagyon fontos szerepet kapnak az energiatakarékos és költséghatékony rendszerek.

Előadásunkat bemutattuk a VII. Mobilrobot Programozó Csapatverseny egyik részfeladatáént 2017. februárjában.

Intelligens fűtési rendszer

ESZI szakkör

Prantner József, Soós Tamás

Felkészítő tanár: Manzinger József

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

1. Bevezetés

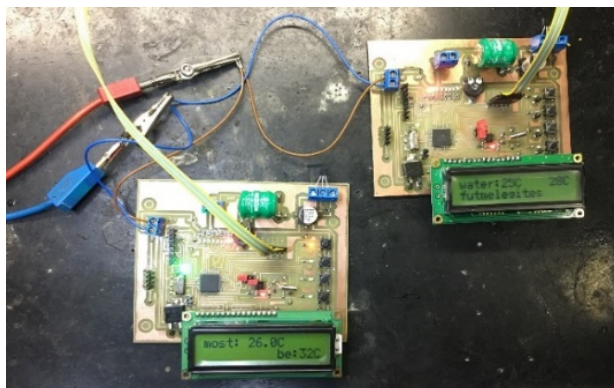
A projektünkben egy olyan fűtési rendszert szerettünk volna megalkotni, mely egyszerűen kezelhető, hatékony és flexibilis. Fontosnak tartottuk, hogy megoldásunk minél különbözőbb szituációkban is helytálljon.

2. Probléma megoldásának menete

Az általános problémában van N darab fűthető, egymástól esetlegesen elzárható helyiség, melyeket az általunk beállított hőmérsékletekre szeretnénk felfűteni. A megoldásnak függetlennek kell lennie a szobák fizikai tulajdonságaitól és a fűtési rendszer fajtájától. A cél elérését azonban gátolhatják ezen tényezők, így ilyenkor külső beavatkozást kell igényelni a felhasználótól, ha lehetséges.

2.1. Hardver

A paneleket saját kezűleg terveztük és készítettük el. Kialakításuknál egyik fő szempontunk volt az egyszerűség. A projektünket egy ATmega 32 mikrovezérlőre alapoztuk. A hőérzékelő egy DS18B20 digitális hőszenzor, melyet onewire kommunikációs módban csatlakoztattunk a mikrovezérlőhöz. Első sorban két féle panelt készítettünk. Az egyik maga a termosztát, ami szabályozza a szelepeket és így állítja be a kívánt hőmérsékletet. A másik panel pedig magát a kazánt szabályozza.

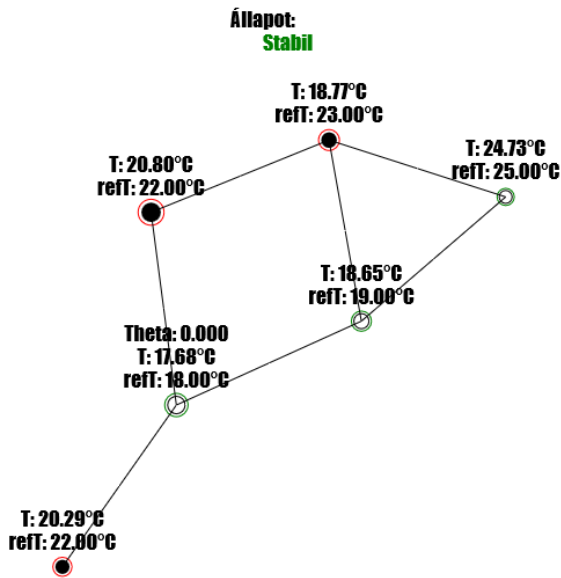


1. ábra. Panel

Szükség esetén bekapcsolja, illetve szabályozza a rendszer teljesítményét az optimális működés elérése érdekében. A továbbiakban szeretnénk valamiféle külsőt is adni az eszköznek.

2.2. Szoftver

Egy ház a fűtés szempontjából modellezhető egy egyszerű gráffal. A pontok a fűthető légtereket jelölik, a kapcsolatok a helyiségek közötti nyílások, átjárók következtében fellépő hőáramlást jelképezik.



2. ábra. A szimuláció JavaScript-ben

A modell dinamikáját az (1) egyenlet írja le. Ez alapján teszteltük a modellt egy szimulált környezetben (2. ábra),

$$dT = c_{fajhő}(\theta_{szelep}P + \sum_i^n k_i(T_i - T) - r(T - T_{környezet}))dt \quad (1)$$

A vezérlés egy módosított PI kontrollert. Alapvető viselkedését a (2) egyenlet írja le. Minden egység külön dolgozik.

$$\theta = \text{Sigm}(K_p e + K_i \int_0^t e dt) \quad (2)$$

A stabil állapotban a rendszer lassú oszcilláláson megy keresztül. A megfelelő hőmérséklet elérésekor a fűtés lekapcsol, enged félfokos visszaesést,

majd újra fűt. A ciklusok közben az algoritmus folyamatosan közelíti a szelepet az optimális értékéhez.

Ha a hőmérséklet valamilyen oknál fogva nem elérhető a rendszer figyelmezteti a felhasználót, esetlegesen az ajtók, ablakok becsukását ajánlja fel. Azonban, ha a probléma forrása a fűtési teljesítmény, a hibát érzékelő egység a teljesítmény növelését igényeli. A vezérlést egy evolúciós algoritmus segítségével optimalizáltuk a maximális költséghatékonyság elérése érdekében.

3. Elért eredmények

Projektünkkel sikerült egy olyan fűtési rendszert létrehozni, mely könnyen beépíthető bármilyen házba, egyszerűen kezelhető és hatékony. Elkészült egy fűtési megoldástól függetlenül alkalmazható vezérlés, illetve a bemutatásához szükséges hardver. A továbbiakban szeretnénk kibővíteni a rendszert internetes lehetőségekkel, valamint kísérleteznénk a termosztátok kommunikációjával.

Intelligens szénmonoxid érzékelő és vészhelyzeti segélyhívó

Arduroid

Kocsis Zoltán Soma, Putz Koppány Norbert, Nánai Ádám

Felkészítő tanár: Zsigri Attila

Tatabányai SZC Szent Imre Gimnáziuma és Szakgimnáziuma - Esztergom, 2500

Esztergom, Főapát u. 1.

1. Bevezetés

Korunk fűtésrendszereinek felismert hibái miatt számtalan hír szól a fűtési szezonban arról, hogy rosszulétek, halálos mérgezések történnek látszólag váratlanul. A fűtésrendszerek korszerűsítése, fejlesztése lehet az egyik út, ami megálljt parancsolhat a gyilkos gáznak. A lég és szelőzés technika fejlesztése kimondottan sokat jelentene a biztonságunk mértékének növelése érdekében. Minden beruházás ez is költséggel jár, és a kiépítésben is lehet hibát vétetni. Úgy gondoljuk, hogy egy aktív biztonsági eszköz elterjedésével csökkenthető a CO által okozott egészségügyi problémák száma. Létezik olyan riasztó eszköz, amely hangos sípolással jelzi a szénmonoxid magas szintjét, azonban nem lehet tudni a valódi koncentrációt. Azt sem tudhatjuk a legtöbb eszköznél, hogy valóban működőképes e, mivel az adatvisszajelzést nem támogatja a legtöbb eszköz. Kérdéses, hogy a sípolásra képes e reagálni védendő egyén. Előfordulhat, hogy már magatehetetlenné válik, mire a jelzést megkapja. Van megoldás a biztonságunk növelésére és az információéhségünk csökkentésére, - ráadásul kényelmesen - ha kihasználjuk az IT rendszerek gyors döntéshozási képességét és felvértezzük azokat a megfelelő döntési algoritmussal.

A projekttel a célunk, egy olyan berendezés építése mely folyamatosan felügyeli a fűtésrendszer működését (a légtérben lévő CO-gáz koncentráció mérésével), veszély esetén értesíti a felhasználót, illetve a közvetlen veszély elhárítását is megkezdi.

Olyan mérő és vészhelyzeti riasztó fejlesztése a cél, amely:

- Mért értéket (gáz koncentrációt, ppm) megjeleníti egy dedikált kijelzőn,
- Az adatokat elérhetővé teszi a hálózaton valós időben (webes felületen, pc-ről, mobil eszközökről),
- Statisztikát készít a korábbi mért értékekből, így láthatjuk a fűtéssel összefüggésbe hozható gáz koncentrációváltozásokat, következtethetünk a fűtési rendszerünk állapotára,
- A szénmonoxid koncentráció veszélyes szintje esetén leállítja a kazánt, illetve megkezdi a helyiség szellőztetését,

- Beépített Li-ion akkumulátora áramkimaradás esetén is biztosítja a folyamatos tápellátást,
- **Fő funkció:** beállított szénmonoxid szintek (figyelmeztetés: 300 ppm, vészhelyzet: 1000 ppm) elérése esetén figyelmeztető fénnel és erős hangjelzéssel jelez és automatikusan segítséget is hív a beépített GSM modulon keresztül előre beállított telefonszámokra küldött SMS-el. A vészhelyzet elhárítása érdekében a kazánt leállítja (áramellátás megszakításával), ezzel egyidejűleg a helyiség szellőztetését (ventilátor működtetésével) megkezdi.

2. Probléma megoldásának menete

A bevezetésben felvázolt probléma legegyszerűbben így demonstrálható:

- Normál gázkoncentráció esetén egy informatív, adatgyűjtő központként működik az eszköz.
- Vészhelyzetben működésbe lép a beépített GSM modul és SMS-ben segítségkérést küld a beállított számokra. Fontos előre egyeztetni az értesítettekkel, hogy az üzenet megérkezése esetén milyen teendőket szükséges megtenni. Például telefonos hívást kezdeményezni, a helyiséget azonnal elhagyni, katasztrófavédelmet értesíteni. Megfelelő törvényi szabályozással/infrastruktúra kialakítása esetén erre maga az eszköz is képes.

Megvalósításhoz szükséges hardver eszközök (lásd: Ábrák fejezet):

- Arduino Mega (1. ábra, bal) – 54 db digitális ki/bemenet, 16 analóg bemenet, 16 Mhz-es processzor, 128 kbyte flash, 8 kbyte SRAM) – Ez a „mikro vezérlő” hajtja végre a létrehozott algoritmus alapján a mintavételt, kiértékelést, a kommunikációt és a beavatkozókat is ez működteti,
- W5100 Ethernet shield – a WEB-es kezelőfelület szervere (1. ábra, középen),
- LCD kijelző– 16*2 karakter, I2C csatlakozási felülettel Arduino-hoz (1.ábra, jobb),
- MQ-7 Elektrokémiai szénmonoxid szenzor (2.ábra, bal),
- GSM modul, SIM800L – GPRS adatkommunikációhoz (2.ábra, jobb),
- Piezo hangszóró – figyelmeztető jelzéshez,
- Li-ion akkumulátor és töltésvezérlő áramköre,
- Hálózati adapter (12V, 1,5A).

3. Szoftveres és hardveres feladatok, nehézségek

A projekthez szükséges C++ nyelvhez hasonló Arduino fejlesztői környezethez további függvénykönyvtárakat kell hozzárendelni: MQ7.h, SPI.h; Ethernet.h; Wire.h; LiquidCrystal_I2C.h; SD.h. A WEB alapú irányítófelület kialakításához a HTML/CSS/PHP leíró és szkript nyelveket használjuk fel.

Nehézségek

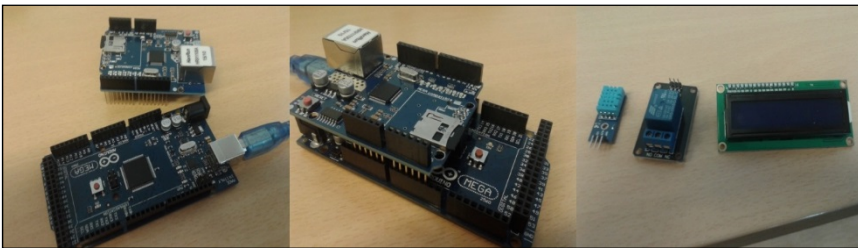
Legnagyobb nehézséget a modulok összeépítése, a függvény könyvtárak megtalálása és célszerű használata valamint az ergonomikus kezelőfelület (nyomógombos kezelés, netes felület) megalkotása okozta.

További nehézséget okozott a szabatos fizikai fogalmak alkalmazása. Az Arduino-hoz kapcsolt szenzor, az LCD-kijelző és relé modul bonyolult vezetékezési és programozási feladatot rótt ránk, mivel mindegyik függvénykönyvtárát meg kellett ismerni. A WEB-es irányítófelület kialakítása is egy újabb programozási nyelv elsajátítását igényelte.

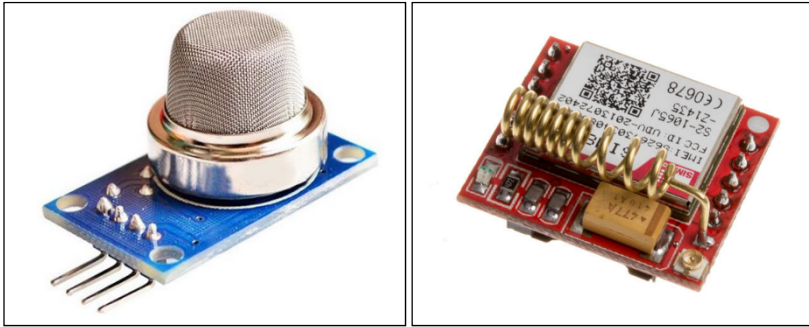
Megvalósítás

A hardver és szoftverkövetelmények számba vétele után, a bemeneti információk rendszerezése és megjelenítésének algoritmizálása következett. A GSM modul megismerése után elkészült a döntési algoritmus, amely a segítségkérést kezeli magas koncentráció esetén. Az Arduino fejlesztőkörnyezetében le kellett kódolni az algoritmust. Ezek után az LCD paneles vezérlés megvalósítása következett, aminek analógiájára egy WEB-es - ergonomikusabb - felület is elkészült, mely a távolról - online - történő irányítást és információkinyerést is lehetővé teszi. Miután a szoftver és hardverszerkezet kialakult, már csak egy tesztelés volt hátra.

4. Ábrák



1. ábra. Arduino Mega 2560, W5100 Ethernet shield, kijelző és relé modul



2.ábra. MQ7 gázérzékelő szenzor, SIM800L GSM modul

5. Elért eredmények

A megvalósítás során sikerült egy megalapozott döntést hozó készüléket megalkotni. A berendezés leglényegesebb előnye, hogy használatával az életveszély elkerülésének esélye megnőtt. A bizonytalanság csökkent, mivel nem egy ellenőrizhetetlen, visszajelzés nélküli „dobozka” van elrejtve a szekrény tetejére 2 AA akkuval, hanem egy központi helyen lévő hasznos állomás, amelynek bármely meghibásodása azonnal figyelmet von magára. Továbbá felhasználója a webes felületen keresztül képet kap fűtési rendszere helyes működéséről és a várható meghibásodásról (határértéket még el nem érő, de folyamatosan növekvő szénmonoxid szint) is.

A szakember általi hitelesítése igazán egyszerű, mert a valóságban megmért adatokat lehet összehasonlítani a referenciával. Álláspontunk szerint egy a valós életben is igazán hasznos készüléket sikerült megépítenünk.

1/3 oktáv felbontású hangszínek-mérő (Audió Spektrum Analizátor)

“Jelbontók”

Kardos Csaba, /*Munkácsi Mátyás*/, Koncsik Alex

Felkészítő tanár: Sándor Péter

Miskolci Szc Kandó Kálmán Szakgimnáziuma, 3525 Miskolc, Palóczy u. 3.

1. Bevezetés

A csapatunk közös csatlakozási pontja a zene és a hangtechnika iránti szeretet. Szerettünk volna létrehozni egy olyan eszközt, amely zenei téren elektronikával is kapcsolatos és látványos. Így jött az az ötlet, mi lenne, ha vizualizálnánk a hangokat! A választásunk egy hangfrekvenciás spektrum analizátorra esett.

2. Probléma megoldásának menete

A pályamunka megvalósítása nagy lendülettel indult és már szeptemberben egy-két hét alatt eljutottunk a számításoktól a szimulációs programok használatáig és felosztottuk 32 sávra a hangfrekvenciás tartományt. A középfrekvencia 1000Hz, amelyet az (1) egyenlet tartalmaz. A 1000 Hz-hez képest kerestük az alsó (16Hz) és a felső (20kHz) határfrekvenciákig a harmonikusokat, amelyekre a (2) és (3) képleteket használtuk. Az így kiszámított sávoknak az alsó és a felső határfrekvenciáinak meghatározása után, amelyet a (4) és (5) egyenlet mutatja be, az eredményeket szabványos értékekre kerekítve egy táblázatban rögzítettük (1. táblázat). A szűrők méretezése után további tervezések következtek.

$$f_{19}^{ctr} = 1000 \text{ Hz} \quad (1)$$

$$f_{n-1} = f_n / 2^{1/3} \quad (2)$$

$$f_{n+1} = 2^{1/3} * f_n \quad (3)$$

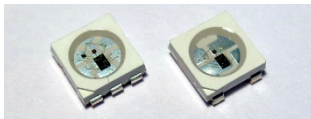
$$f_n^{low} = f_n / 2^{1/6} \quad (4)$$

$$f_n^{high} = 2^{1/6} * f_n \quad (5)$$

Frekvencia értékek (Hz)															
16	20	25	31,5	40	50	63	80	100	125	160	200	250	315	400	500
630	800	1k	1,25k	1,6k	2k	2,5k	3,15k	4k	5k	6,3k	8k	10k	12,5k	16k	20k

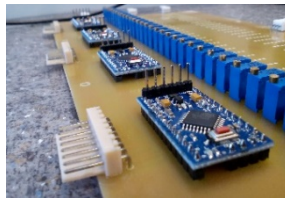
1. táblázat. A szabványos értékekre kerekített frekvencia értékek

A probléma a kijelzés módjával kezdődött. Abban biztosak voltunk, hogy LED-eket szeretnénk, és ha már lúd, akkor legyen kövér, egyből 32x16 RGB LED-et. Igen, de ezt vezérelni is kell valahogy és 512 db LED-hez vezetősávok tömkelegére lesz szükségünk. A megoldást egy nagyon speciális WS2812B - típusú LED jelentette (1. ábra). A kínaiak minden egyes pixel LED-be beintegráltak egy kis jelfeldolgozó és meghajtó egységet a három alapszín (piros, zöld, kék) keveréséhez. A fantasztikus ebben az, hogy külön áramgenerátort is tartalmaz, így nincs probléma a nyitófeszültségből adódó fényerősség különbségek kiküszöbölésére sem. A vezérlést egy közös adat és órajel vezetéken keresztül kell küldeni, megfelelő időzítéseket alkalmazva.



1. ábra. WS2812B - típusú LED

A projektben négy darab Arduino Pro Mini dolgozza (2. ábra) fel a bemenetről érkező jelet, majd 128 LED-es blokkokra osztva tölti fel adattal a meghajtó chip-eket.



2. ábra. Arduino Pro Minik az alaplapunkon

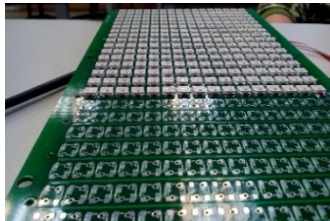
Az elképzelésünk az volt, hogy a kijelző frissítése a lehető legrövidebb időn belül megtörténjen, és minél precízebb műszert építsünk. A választás a nem túl költségkímélő, de annál jobb megoldáshoz vezetett. Ötvöztük az analóg technika előnyeit a digitális technikával. A bemeneten egy speciális, logaritmikus léptékekben digitálisan vezérelhető analóg hangerőszabályozó áramkör foglal helyet. Ezt követi 32 darab keskenysávú sáváteresztő szűrő, aminek a kimenetén 32 darab teljes hullámú precíziós egyenirányító található (3. ábra). Ezek a kártyák gondoskodnak a jelfeldolgozás elősegítésében.



3. ábra. Sáváteresztő szűrő (balra), precíziós egyenirányító (jobbra)

Innen már csak az analóg digitális átalakító feladata a minták számokká alakítása. Az ATMEGA328P mikrovezérlő SMD verzióban pontosan 8 darab analóg bemenetet tartalmaz, ami a teljes spektrum analízátor negyedét frissíti egy blokkban. Ezzel a felosztással rengeteg időt megspórolunk. A párhuzamos adatfeldolgozást egy gyors algoritmussal megtámogatva különböző színekre színezzük a kijelzőt. Nincs szükség multiplex vezérlésre, mert mindaddig tárolódik a LED saját chip memóriájában az adat, amíg felül nem írjuk. Ezáltal nincs vibrálás sem.

A következő problémát az 512 darab LED áramellátása okozta. Teljes fényerőn 30,72 Amper áramerősségre van szüksége 5V tápfeszültség betáplálás esetén. Erre ideálisnak egy 450W-os PC tápegységet tartottunk, ami több szál vezetékkel kapcsolódik a LEDPANEL-hez (4. ábra). Az analóg áramkörök földfüggetlen tápellátását saját tervezésű (ZVS) inverter kapcsolással oldottuk meg.



4. ábra. A készülő LEDPANEL

A kezelőszerv része 2 darab nyomógombos Rotary Encoder és a hozzá tartozó szabványos 16x2-es LCD kijelző. Ezek segítségével vezérelhető a készülék, valamint a jelszintek és egyéb más paraméterek állíthatók. A menüben kényelmesen lépkedhetünk a forgatógombok segítségével. A berendezés belsejében helyett kapott még egy valós idejű óra chip (DS3231) és egy hőmérséklet mérő (DS18B20+) szenzor is.

Az analóg jélbemenet lehet RCA, JACK, vagy XLR dugó. A kimenet megvalósításánál még arra is gondoltunk, hogy egy relés útválasztó és előerősítő kapcsolást készítünk (6. ábra), ami természetesen szintén a digitálisan vezérelhető analóg hangerőszabályozó áramkörre épült. Ezzel lehetőség adva bármilyen erősítési láncba való beillesztésre.



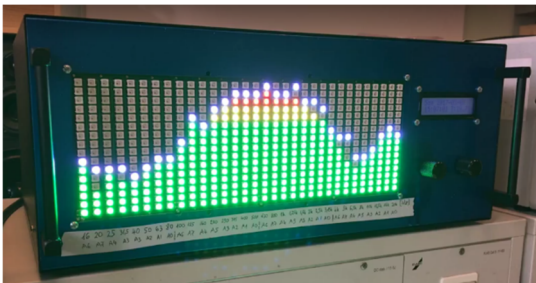
6. ábra. Bemeneti előerősítő, relés útválasztó

3. Elért eredmények

A pályamunka az elméleti terveinket beigazolta, remekül működő és így ezzel egy egyedülálló konstrukciójú precíz hangfrekvenciás spektrum analizátort hoztunk létre (7. és 8. ábra). Természetesen léteznek PC alapú szoftverek, amik egy hangkártya segítségével még ettől is nagyobb felbontásra képesek, de ez akkor is csak erre a célra kifejlesztett hardver lett. A piacon egyetlen terméket találtunk, ami hasonló a miénkhez, de az is képességeit tekintve korlátozott, az áráról nem is beszélve. Összességében a hangszínkép-mérő műszerünk akusztikus mérési feladatokra, zajanalízisre, hangszugárzók telepítésénél és összehangolásánál kiválóan alkalmazható egy mérőmikrofon segítségével.



7. ábra. Az Audió Spektrum Analizátor belülről



8. ábra. Az elkészült Audió Spektrum Analizátor működés közben

RC Cloud: egy léghajó alapú drón

RC Cloud

Sztojka Áron, Hargitai Benke, Molnár Barnabás

Felkészítő tanár: Lang Ágota

Soproni Széchenyi István Gimnázium, Sopron, Templom utca 26.

1. Tervek

Az utóbbi években egyre nagyobb szerephez jutottak a drónok különböző kutatásokban. A technikának ez a területe gyors fejlődésnek indult, ámde a legtöbb drón képtelen hosszabb ideig levegőben maradni. Ennek oka főként az, hogy rengeteg energiát használnak arra, hogy egyáltalán levegőben maradjanak. A multikopterek egyhelyben lebegve is rohamosan fogyasztják az energiát, nem is beszélve a folyamatosan haladó merevszárnyú modellekről. A probléma megoldásáért kicsit vissza kellett nyúlnunk időben az egykori Zeppelinekig. Ezek a járművek könnyebbek voltak a levegőnél, egyáltalán nem került energiájukba a fennmaradás. Eredetileg a mi projektünk is egy Zeppelin (merev vázú léghajó) lett volna, de kis számolás után rájöttünk, hogy ez a mi kereteink között nem megvalósítható. Ekkor fordultunk a Zeppelinekhez hasonló Blimpek (nem merev vázú léghajók) felé. Sok különböző terv után végül eljutottunk a jelenlegi projektig, amely már megvalósíthatónak tűnik. Célunk tehát egy távirányítható blimp megépítése, amelyen azonban egy Raspberry Pi is repül. Ennek segítségével tudunk légifelvételket készíteni illetve időjárási adatokat gyűjteni.

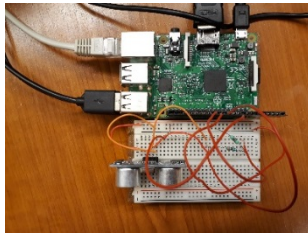
2. Az RC Cloud felépítése és programozása

A test alapját egy közel 2 méter (5,5 láb) átmérőjű ballon képezi, a ballont egy szúnyoghálóból, vagy vékony, erős szövetből varrott csőben helyezük el. A ballon különböző üzletekben megtalálható, éppen ezért választottuk, hogy könnyen és olcsón cserélhető legyen. A cső átmérője kisebb, mint a ballonné, ezért az megnyúlik benne és eléri az általunk kívánt szivarformát. A csőre kívülről kis fülek vannak varrva, ezekre lehet erősíteni különböző eszközöket. A gondola ugyanilyen módon van rögzítve. A szúnyoghálót keskeny, erős szalagokkal meg is erősítjük, hogy biztosan bírja a ballon feszítő erejét. A gondola 3mm vastagságú rétegelt lemezből áll, ez könnyű, hozzáférhető, erős és könnyen megmunkálható anyag. A bontható kötések csavarokkal, a bonthatatlanokat pedig ragasztással oldottuk meg.

Alapfelszerelés:

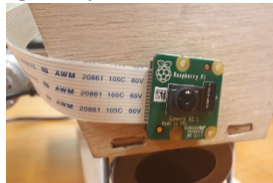
- Raspberry Pi 3 model B; Lásd: Fedélzeti számítógép
- Raspberry Pi Camera V2 Video Module; Lásd: Kamera
- NiMH Akkupack 7,2V 5000mAh (Kb 1h repülési idő)
- 2 db. X-Fly 480 elektromotor (7,2V)

Fedélzeti számítógép: A fedélzeti számítógép szerepét egy Raspberry Pi 3 model B látja el. Feladata, hogy a kamera, és a csatlakoztatott szenzorok működését szabályozza, adataikat feldolgozásra előkészíti, elmenti. A tökéletes megoldás az lenne, ha "előben" is tudná ezeket közvetíten interneten keresztül. Az irányításhoz/mozgatáshoz szükséges szervókat és motorokat is a számítógép kezeli, de a parancsokat a távirányító vevőjétől kapja.



1. ábra. A fedélzeti számítógép

Kamera: A hajóra a Raspberry Pi saját kameráját, egy V2 Video Module-t szereltük fel. Ez méretéhez, és főleg árához képest nagyon jó minőségű felvételeket tud készíteni, 30 fps mellett Full HD, 90 fps mellett pedig 640*480 pixel felbontásban dolgozik. Rögzíteni csavarokkal fogjuk a gondola aljára (a jelenlegi megoldás csak ideiglenes).



2. ábra. A fedélzeti kamera

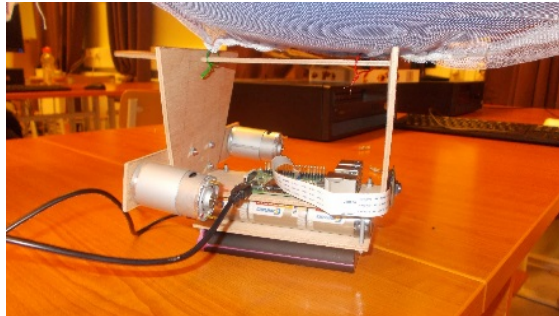
Irányítás: A hajót először úgy egyensúlyozzuk ki, hogy a levegőben lebegjen. A függőleges és vízszintes kormányzásra keresztvezérsíkot



3. ábra. A motorok

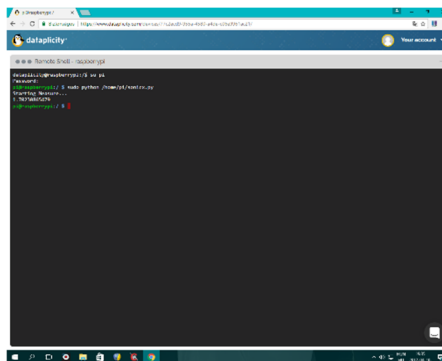
alkalmazunk. Ezt a meghajtásra használt motorok mögé helyezük el, a légáram útjába. Az irányítás alapvetően távirányítóval történik. A vevőt a fedélzeti számítógépre kötjük rá, és az kezeli a szervókat, motorokat. A meghajtást két kisebb teljesítményű motor adja, ezek egyenként 100 grammot nyomnak. 7,2 voltos feszültséggel működnek, ezekhez méreteztük az akkumulátort is. A motorok egy-egy 5 inch, azaz 12,7 cm átmérőjű rotort kapnak a megfelelő meghajtás érdekében.

Jelenleg a képen látható formában tudjuk szemléltetni, milyennek is tervezzük az RC Cloudot, mert a hélium beszerzésére még nem találtuk meg az árban legideálisabb megoldást.



4. ábra RC Cloud az asztalon

Programozás: A vezérlőegység, a Raspberry Pi alapvetően python nyelven programozható, egyszerű szöveges kódot futtatunk le a Raspberry-n. Mindezt interneten keresztül is meg tudjuk tenni a dataplicity.com weboldal segítségével.



5. ábra Képernyőkép a Dataplicity szolgáltató honlapjáról

Ezzel a weboldallal el tudjuk érni hogy másik eszközről is le tudjuk futtatni az adott programunkat. Ez például a fényképezés vagy mérés indításánál előnyös.

Szenzorok: A Raspberry Pi csak digitális bemenetekkel rendelkezik, így szenzorokat is ennek megfelelően kellett megválasztanunk. Szerencsénkre találtunk egy olyan komplex szenzort (BMP180-M), ami tud hőmérsékletet, nyomást valamint magasságot mérni (illetve utóbbit feltehetően a nyomásból számítja).



6. ábra Az RC Cloud szenzora

3. Összefoglalva:

Célunk egy viszonylag olcsó drón létrehozása, mely testreszabható, sokféle feladat elvégzésére alkalmas, hosszú üzemidővel bír, és rendelkezik a blimppek pozitív tulajdonságaival.

Drónunk előnyei:

- kis energiafogyasztása, nagy akkumulátora miatt viszonylag hosszú ideig üzemel
- a multikopterekhez hasonlóan képes egy helyben lebegni, és kis területen is könnyen leszáll
- nagy hatósugárral rendelkezik hosszú üzemideje és viszonylag nagy sebessége miatt
- könnyen vezethető 4 csatornás távirányítóval
- testreszabható, azaz különböző feladatokra specializálható

Alkalmazható pl.:

- Légköri adatok mérésére hosszabb időn keresztül
- Domborzat 3 dimenziós feltérképezésére
- Légifotózásra
- Hobbicélú repülésre

Informatika szekció

Sophie

A három grácia

*Német Ágnes Sára, Hunyadi Anna, Ruzsa Kata
Felkészítő tanár: Rozgonyi-Borus Ferenc
SZTE Gyakorló Gimnázium és Általános Iskola,
6722 Szeged, Szentháromság utca 2.*

1. Bevezetés

Hogyan tegyük szórakoztatóvá a matematikát a kis –lány - gyerekek számára? Ez a játék logikus gondolkodásra ösztönöz, miközben egy nehéz labirintuson mennek keresztül. A játékkal ráébreszthetjük az alsós lányokat, hogy számukra is fontos lesz az életben a matematika, ezért életszerű problémákat is felvetettünk. Ezalatt felidézük a Hamupipőke történetét és igazi hercegnővé válhatnak.

2. Probléma megoldásának menete

A Scratch programozási nyelvet választottuk a probléma megoldásához.

Olyan játékot készítettünk, amibe beleszóttuk a matematikát, mert egy igazi hercegnőnek műveltnek kell lennie.

A játékszabályokat a két főszereplő dialógusából ismerhetjük meg a játék elején. Lásd a 2. ábrán.

A mesében a kisállatkák próbára teszik a játékost. Ehhez olyan kérdéseket kellett kitalálnunk, ami az adott korosztálynak érthető, megoldható, és a történettől sem áll messze. A kérdéseket alsós tanulók segítségével teszteltük. Ezekre adott idő alatt kell válaszolni, ezt is a segítségükkel állítottuk be. Az idő a 4. ábrán látható, a bal felső sarokban.

A játékos miközben eljut az állatkákhoz néhány segítségbe illetve nehézségbe ütközhet. Ha tökört érint útközben, időt nyer, ha fáklyát érint, újra láthatja a pálya egészét. Ez az 5. ábrán látható.

Amikor a játékos helyes választ adott, új pályára léphet.

Emellett látványossá kellett tennünk a játékot, így magunk rajzoltuk a szereplőket. Ez a 1. képen látható. A háttér egy témához illő kép, amely néhány pályánként változik, és rajta a labirintus, amelyet a „Word 2013” program segítségével, táblázatokkal rajzoltuk meg. A hangulathoz hozzájárul, hogy hanggal is kibővítettük a programot.

A játék végén a főszereplő/játékos eljut a herceghez a palotába, hiszen ez volt a fő célja is, és bebizonyította, hogy erre érdemes.

2.1. Ábrák

Az 1. ábrán látható Sophie kinézete a játék elején és miután a játékos végigment a pályán. Minden egyes jó válsz után egy új ruha az ajándék.



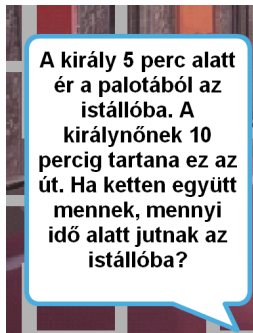
1. ábra. Sophie átváltozása

A 2. ábrán egy párbeszéd részlet látható a játék elejéről.



2. ábra. A párbeszéd

A 3. ábrán egy kismadár által feltett kérdés van.



3. ábra. Egy feltett kérdés.

A 4. ábrán a labirintus egy részlete van.



4. ábra. Labirintus

A 5. ábrán a fáklya és tök látható.



5. ábra. A segítő lehetőségek.

3. Elért eredmények

Az iskolánknak van általános iskola része is, és ez inspirált minket egy kisgyerekeknek szóló játék megtervezésére. Az elkészült programot sok kisgyerekekkel kipróbáltattuk, ami hatalmas sikert aratott.

Online Betegkarton

Hiperkockák

Vámosi Tamás, Király Máté

Felkészítő tanár: Vári Ágnes, Pásztor Zoltán

*Hódmezővásárhelyi SZC Pollák Antal Szakgimnáziuma és Szakközépiskolája,
6600 Szentes, Apponyi tér 1.*

1. Bevezetés

Az Online Betegkarton egy olyan megoldás, melynek célja az egészségügyi „tulajdonságaink” egyszerű, könnyen elérhető és átlátható rendszerezése. Ebben a rendszerben a felületre a felhasználók regisztrálják magukat (lásd 1. ábra), majd az orvosok (akik „Admin” jogosultsággal rendelkeznek) tölthetik fel a felhasználók (Kliensek) adatait.

Mi úgy gondoljuk, hogy a legnagyobb segítséget az Országos Mentőszolgálat munkatársainak jelentené, ugyanis egy baleset esetén az orvos körülbelül másfél perc alatt kideríthet minden lényeges információt (lásd 2. ábra) a balesetet szenvedő személyről, ha tudja a TAJ-számát, vagy a személy egyedi azonosítóját (amit regisztrációkor kap meg a szervertől).

Az Online Betegkarton egy weboldalból és egy mobil alkalmazásból áll (jelenleg az alkalmazás csak Android rendszeren érhető el). Mindkét felületen ugyanazok a funkciók találhatók, kevés eltéréssel. Lehetőségünk van az orvosi leleteinkről készült képek feltöltésére (lásd 3. ábra, 4. ábra). Később ezeket a képeket megtekinthetjük, vagy újra letölthetjük. Természetesen lehetőségünk van képeink törlésére is. Továbbá a felhasználók számára kialakítottunk egy mezőt, ahol a saját megjegyzéseiket adhatják meg (pl.: „Az elmúlt héten „X” gyógyszerert szedtem mert fáj a fejem, de azt nem az orvosom írta fel.”).

2. Probléma megoldásának menete

A programozás során rengeteg megoldandó feladatunk volt, ezek nagy része abból adódott, hogy a csapat egyik tagja írta meg a szerver oldali PHP kódot, a másik tag pedig a kliens oldali Java alkalmazást. Ebből adódott az, hogy a változók neveit különböző módon írtuk le, így az adatcsere nem tudott végbe menni. Továbbá azt is érdemes megemlíteni, hogy a „java.lang.NullPointerException” kifejezés a csapaton belül már szállóigévé vált.

Szerver rész: A szerveren fut egy Apache és egy MySQL szerver. Az oldal funkcióit és a mobil alkalmazás kérelmeit PHP Scriptek hajtják végre (Példa PHP kód a felhasználók saját megjegyzésére 5. ábra).

Kliens rész: A Kliens funkció részben az oldalon található PHP Scriptekből áll, részben a mobil alkalmazásból. Ez azért szükséges, hogy a saját fiókunkhoz

hozzáférhessünk mind a webes felületen, mind a mobil alkalmazásból. Itt egyszerű felhasználóként megtekinthetjük adatainkat, szerkeszthetjük a saját megjegyzéseinket, leleteinkhez tölthetünk fel képeket címmel ellátva és törölhetjük a már fent lévő képeinket. „Admin” jogosultságokkal mindezen felül kereshetünk a felhasználók között (TAJ szám vagy egyedi azonosító alapján), és a webes felületen még lehetőség van adatok feltöltésére és módosítására is.

2.1. Ábrák



The image shows a screenshot of a web browser window. The browser's address bar displays the URL "onlinebetegarton/register.php". The main content area of the browser is light blue and contains a registration form. The form is titled "Regisztráció:" and includes three input fields: "Felhasználónév:", "E-mail cím:", and "Jelszó:". Below these fields is a "Regisztrálok" button.

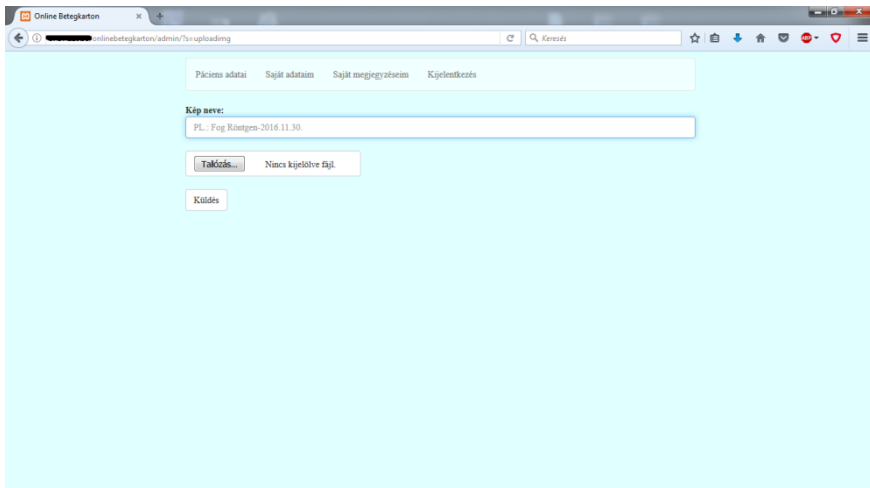
1. ábra - weboldalas regisztráció



2. ábra - egy felhasználó adatai



3. ábra - Képfeltöltés az alkalmazásban



4. ábra - Képfeltöltés a weboldalon

```

<?php
if(isset($_SESSION["id"]))
{
    if(isset($_POST["submit"]))
    {
        $sql = "UPDATE `data` SET `owncomments` = '$_POST[comment]' WHERE `ID` = $_SESSION[id]";
        if($result = $mysqli->query($sql))
        {
            echo "<div class='alert alert-success'>
                <strong>Frissítés sikeres volt!</strong></div>";
        }
        else echo "<div class='alert alert-danger'>
                <strong>Frissítés sikertelen volt!</strong></div>";
        }
        $sql = "SELECT `owncomments` FROM `data` WHERE `ID` = $_SESSION[id]";
        if($result = $mysqli->query($sql))
        {
            $user = $result->fetch_assoc();
            echo "
            <div>
                <form method='POST'>
                    <div>
                        <div>Saját megjegyzés:</div><br>
                        <input type='text' class='form-control' rows='9' cols='75' placeholder='Megjegyzés' name='comment' value='$_SESSION[owncomments]'></div><br>
                        <input type='submit' name='submit' value='Frissítés' class='btn btn-default'>
                    </div></div>
                </div><br>
                <a href='?#uploadimg' class='btn btn-default'>Leletek képeinek feltöltése</a>
                <a href='?#listimg' class='btn btn-default'>Leletek kilistázása</a>
            ";
        }
        else echo "Kérem lépjen be!";
    }
}
}

```

5. ábra - PHP kód a felhasználók saját megjegyzésére

3. Elért eredmények

Magyar Elektrotechnikai Egyesület és a Budapesti Műszaki Szakképzési Centrum által meghirdetett MŰKÖDJ! pályázat döntőjében mutathattuk be rendszerünket. Az Online Betegkarton tesztelés alatt áll jelenleg. Akik eddig használták elégedettek voltak mindkét felülettel, a hibákról szóltak, azokat kijavítottuk

Gépjárművezetés biztonságnövelése képfeldolgozás alkalmazásával

Bencsik Blanka

Felkészítő tanár: Kőrösi Gábor

Bolyai Tehetséggondozó Gimnázium, Posta u. 18., Zenta 2440 Szerbia

1. Bevezetés

A világon rengeteg ember veszti életét közúti balesetben az emberi hibáknak betudhatóan. Ez motivált arra, hogy létrehozzak egy programot, mely könnyíti az autóvezető számára a vezetést, s ezáltal csökkentheti a közúti baleset bekövetkezésének valószínűségét. Az elkészült program képes megállapítani a gépkocsira rögzített kamera által készített felvételtől, hogy az úton hol vannak az sávjelző vonalak, hol halad az autó, helyesen tartja-e a sávot, merre kanyarodik az út, és, hogy szaggatott-e a vonal. Mindezekről a vezető grafikus formában kap értesítést, valamint, ha túlzottan kitér a sávból, vagy telt vonalnál előzésbe kezd, akkor figyelmeztető hangjelzést ad. A program elkészítéséhez a Microsoft Visual Studio 2010 szoftvercsomagot, valamint az OpenCV képfeldolgozási könyvtárat használtam.

2. Probléma megoldásának menete

A program a videót képkockákra bontja, s minden képen elvégzi a programkódba foglalt műveleteket.

2.1. Előfeldolgozás

Előfeldolgozás során javítjuk a kép minőségét, előkészítjük, alkalmassá tesszük a feldolgozásra. A színek ebben az esetben nem fognak különösebb szerepet játszani, ezért a képet szürkévé alakítjuk.

2.2. Szegmentálás

A szegmentálás az értékes részek elválasztása a háttértől. Jelen esetben adaptív küszöbölést alkalmazunk (adaptive thresholding), amely nem az egész képre, hanem külön-külön annak kisebb tartományaira keres küszöbértéket. Adaptív küszöböléskor az adott tartományba eső képpontok közül megkeressük a legkisebbet és a legnagyobbat. A két érték különbségének meg kell haladnia egy meghatározott differenciaértéket, különben az eltérés csupán szennyeződésnek tudható be. Ekkor az adott tartományba eső képpontok mindegyike 0 értéket kap (fekete). Ha lokális tartományon belüli maximum és minimum különbsége eleget is tesz a feltételnek, egy-egy képpont csak akkor nyilvánítható fehérnek, ha értéke közelebb van a maximumhoz, mint a minimumhoz. [6] [7] [8]

A küszöbölést csak a kép alsó felén végezzük el, feltételezve, hogy a kép felső felében égbolt, azaz jelentéktelen információ van, azt egyszerűen lefeketítjük.

2.3. Élkiemelés

Sávfelismeréskor a sávjelölő vonalak belső széléhez illesztett vonalat keressük. Vonalfelismerést csak éleket tartalmazó képen lehet eredményesen véghezvinni, ezért a képen élkiemelést (edge detection) kell végezni.

Digitális képen élek ott találhatóak, ahol a szomszédos pixelek között nagy az intenzitáskülönbség. A gradiens számításán alapuló élkiemelési módszerek közös jellemzője, hogy két, egymásra merőleges irányban számol gradienst, mindkettőnek megvan a saját operátora. [1] [3][6] [7] [8] Gradiens:

$$\nabla f = \frac{\partial f}{\partial x} \vec{i} + \frac{\partial f}{\partial y} \vec{j} \quad [2]$$

, ahol \vec{i} és \vec{j} az egységvektorok, x és y pedig az irányok.

A két egymásra merőleges irányú differenciából a következőképp határozható meg a gradiens nagysága és iránya:

$$G = \sqrt{G_x^2 + G_y^2}, \quad \varphi = \arctg \frac{G_x}{G_y} \quad [3]$$

2.4. Sávtartás

Bármiféle következtetés levonása előtt ki kell nyerni a képből azt az információt, hogy hol halad az autó, és a meghatározott útsávot grafikusán kijelölni a képen. Az első meghatározásához a sávjelző vonalakhoz illesztett egyeneseket keressük az élkiemelt képen. A vonalfelismeréshez Hough-transzformációt használunk, amely adott kép objektumainak olyan részhalmazát határozza meg, amelyekre közös egyenes illeszthető. [4] Polárkoordináta-rendszerben dolgozunk, mely a sík minden pontját egy szög és egy távolság adattal látja el. Egy egyenes egyenlete így írható le polárkoordináta-rendszerben:

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right) \Rightarrow r = x\cos\theta + y\sin\theta \quad [5]$$

, ahol r a Descartes-féle koordináta-rendszerben az egyenesre az origóból húzott merőleges, θ pedig r és az x -tengely között bezárt szög.

A várt sávjelző vonalak dőlési szögét bizonyos határok közé tudjuk szorítani a kamera állásának alapján, így a több felismert vonal közül ki tudjuk választani azt a kettőt, ami valóban a sávjelző vonalakat jelöli. A vonalnak polárkoordinátában használatos paramétereit ismerjük, x és y koordinátájának meghatározásához analitikus geometriai számításokat kell végeznünk.

Az applikáció egyik szolgáltatása, hogy megállapítja, a gépjármű helyesen halad-e az útsávban. A kamera úgy van felszerelve a kocsihoz, hogy a sávtartás akkor tökéletes, ha a kép függőleges középvonala egybeesik az útsáv közepével. A program az ettől való százalékos eltérést számolja, a következőképp: $\frac{\Delta x}{h} * 100$, ahol Δx a kép függőleges középvonala és a beazonosított haladási sáv közepe közti differencia, h pedig a beazonosított haladási sáv szélességének fele. Ha a kitérés meghaladja a 20 %-ot a program figyelmeztető jelzést ad a vezetőnek.

2.5. Kanyar felismerése

A kanyar megállapítása perspektivikus képen nehézkes, ezért a képet felülnézetivé transzformáljuk. Az átranzformált képen az út jobbra kanyarodásakor a sávjelző vonalak jobb irányba, balra kanyarodásakor bal irányba dőlnek, egyenes haladása esetén pedig a két oldalsó sávjelző vonal vagy párhuzamos egymással, vagy kissé egymás felé dől. A program egy vágást végez a kép magasságának háromnegyedén, majd oszloponként összeszámolja a fehér pixeleket, külön a kép magasságának felétől a vágásvonalig, és a vágásvonaltól a kép aljáig terjedő tartományokban. Ezután az alsó tartományban meghatározott értékekből kivonja a felső tartomány értékeit. Ekkor a vágásvonal mentén ahol a sávjelző vonal áthalad rajta, egy hullámot kapunk. A hullám negatívba nyúló része a vágásvonal alatti, pozitívba nyúló része pedig a vágásvonal feletti fehér pixeleket mutatja. A sávjelző vonal haladási iránya abból határozható meg, hogy a hullám melyik tartományba nyúló része következik előbb, vagyis a hullám szélsőértékei között a függvény növekszik vagy csökken. Csökkenéskor balra, növekvéskor jobbra kanyarodik. Az út kanyarodási irányát a program 10 képkockánként határozza meg az addig összeszámolt hullámokból.

2.6. Szaggatott vonal felismerése

A sávjelző vonal folytonosságának vizsgálatát a szegmentált képen végezzük az előzőleg Hough-transzformációval azonosított vonal mentén, 100 pixeles környezetben, a sávjelző vonalak metszéspontjától a kép aljáig. A program összeszámolja, hogy adott magasság szinteken a vonal 100 pixeles környezetében hány fehér képpont található. Ezután meghatározzuk a küszöbértéket, ami azt mutatja, legkevesebb mennyi fehér képpont szükséges ahhoz, hogy azt mondhassuk, adott magasság szinten megjelenik a sávjelző vonal.

$$h = \frac{1}{2 * db} \sum_k^n f_k$$

ahol db a fehér képpontokat tartalmazó magasságszintek száma, k a magasságszint, ahol a vizsgálat kezdetét veszi, n a magasságszint, ameddig a vizsgálat tart, f_k pedig a fehér képpontok száma adott magasságszinten.

A küszöbértéken felül eső értékek az mutatják, hogy adott magasságszinten megjelenik e sávjelző vonal. Ezek új, 1-es értéket kapnak, a maradék 0-át. Ezután a program eldönti, hogy a vizsgált vonal folytonos-e. Megszámolja, hogy az egyenes mentén hány olyan magasságszint van, ahol nem jelenik meg a sávjelző vonal. Ha ez szakasz nagyobb, mint a teljes vizsgált szakasz 35 %-a, akkor az sávjelző vonal szaggatottnak, ellenkező esetben teltnék minősül. A felhasználó számára az eredmény a kijelzőn válik láthatóvá: szaggatott vonal esetén egy zöld színű, telt vonal esetében pedig egy piros vonal jelenik meg.

3. Probléma megoldásának menete

Az eredmény egy hatékony, látványos program, mely használata jelentősen megkönnyítheti a gépjárművezető dolgát, s akár a baleset bekövetkezését is megelőzheti. Megalapozott képfeldolgozási eljárásokkal és összetett matematikai számításokkal dolgozik, mely leírása 4 oldalban szinte lehetetlen. Munkám során rengeteg tapasztalatot szereztem. Az itt bemutatott applikáció elkészítése rengeteg munkát és energiát ölel magába, és mégis csak egy része a végleges elképzelésemnek. Mint minden más informatikai fejlesztés, ez az applikáció is, hatékonysága ellenére természetesen bővíthető, továbbfejleszhető a mesterséges intelligencia és gépi látás szakterületére irányulva, s ennek végrehajtása jelen is van a jövőbeli terveim közt.

4. Felhasznált irodalom

- [1] Czap László: Képfeldolgozás., Miskolci Egyetem.
- [2] Wayne Niblack: An Introduction to Digital Image Processing, Prentice-Hall International Ltd, UK, 1986.
- [3] www.inf.u-szeged.hu/~gnemeth/kurzusok/kepfel1/7_gyakorlat.pdf
- [4] Fazekas Attila, Kormos János: Digitális képfeldolgozás matematikai alapjai., Debreceni Egyetem Informatikai Intézet.
- [5] OpenCV 2.4.13.2 documentation. <http://docs.opencv.org/2.4/index.html>
- [6] Henning Bassmann, Philipp W. Besslich: Ad Oculos Digital Image Processing., International Thomson Publishing Company, Sheffield, 1995.
- [7] Theo Pavlidis: Algorithms for Graphics and Image Processing., Computer Science Press, Rockville, 1982.
- [8] Anil K. Jain: Fundamentals of Digital Image Processing., Prentice-Hall, Inc. New Jersey, 1989.

Company Manager

SootSoft

Korom Richárd

Felkészítő tanár: Szöllősi Béla

*GYSZC. Kossuth Lajos Szakképző Iskolája és Kollégiuma, 5900. Orosháza,
Kossuth tér 1.*

1. Bevezetés

Rohanó világunkban, a kis- és középvállalkozások életében egyre nagyobb terhet jelent a statisztikai adatok naprakész nyilvántartása, a dolgozók munkájának nyomon követése, a hibalehetőségek minimálisra csökkentése vagy kiküszöbölése. A fent említett tényezőket figyelembe véve célom az volt, hogy kifejlesszek egy olyan vállalatmenedzselő programot, amely önmagában képes kezelni a cég teljes adatbázisát beleértve a raktárkészlet nyilvántartását, a dolgozók juttatását a különböző biztonsági rendszerek (riasztó, kamera) működtetését. Tájékozódásom során megállapítottam, hogy több hasonló szoftver is létezik, amely csak egy – egy rész területet (ez általában a könyvelés) kezel, azonban olyanról nincs tudomásom, ami az általam megfogalmazott komplex elvárásoknak megfelelne. Szüleim biztatására úgy döntöttem, hogy a családi vállalkozás keretén belül kifejlesztsek egy globális rendszert, mely a folyamatos újításoknak köszönhetően naprakészen átfogja a vállalkozás egész tevékenységét.

A program tervezésénél három alapelvet emeltem ki és fogalmaztam meg

- a vállalkozás nyilvántartási rendszerének fejlesztése
- a környezetvédelmi tudatosság növelése
- az időhatékonyság optimalizálása

A program segítségével a hagyományos papír alapú nyilvántartás helyett életbe lépett a digitális adatkezelés, amely a másik két tényező előre mozdulását is elősegítette. A környezetvédelmi tudatosság kialakításának elengedhetetlen feltétele a papír hulladék csökkentése, illetve a gyorsabb adatszolgáltatás, melynek segítségével időt takarítunk meg.

2. Probléma megoldásának menete

A szoftver tervezésének egyik nehezített eleme a „kettősség elve”, melynek egyrészt a cégvezető elvárásainak, másrészt pedig a jó szakmunkás gyakorlati jellegű igényeinek kell megfelelnie, úgy, hogy egyaránt tudja használni mindkét fél a pontosság, a megbízhatóság és a kezelhetőség szellemében.

2.1. A vállalkozás nyilvántartási rendszere

A program alapvetően 2 részre osztható, melyek a felhasználói, illetve a cégvezetői oldalakból tevődnek össze. A felhasználói felület egyszerű letisztult forma, épp a könnyű kezelhetőség miatt. A felhasználói adatokat a rendszer automatikusan feldolgozza, mellyel leegyszerűsíti a további adatbevitelt.

A stílus elemek megvalósítása meglehetősen időigényes volt. A kis- és középvállalkozások nyilvántartásában már nem elegendő, hogy egy program „csak” működőképes legyen. Szükséges feltétel az átláthatóság, kezelhetőség, egyszerűség és a gyorsaság valamint a cégvezetői felület zsúfoltságának az elkerülése, ami a legfőbb problémát okozta. A legtöbb információ ismertetésén túl, átláthatóvá kellett tenni az egyes felhasználókat és a különböző napok adatait.

2.2. A szoftver kivitelezése

A kivitelezés egyik legfontosabb tényezője az volt, hogy minden tartalomnak dinamikusan kell változnia. Ezek mellett törekedni kellett az erőforrások használatának optimalizálására. A rendszer számos különböző programozási nyelvet használ, így ezek egybefüggő, összehangolt működése elengedhetetlen.

A legfontosabb kritérium a biztonság megalapozása és a munkatársak 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, talán az egyik leghasznosabb elem, a jobb felső sarokban található időzár, amely 5 perc inaktivitás után, automatikusan kilépteti a felhasználót, hogy illetéktelen személy ne tudjon a fiókjához hozzáférni.

Természetesen ebből kifolyólag a bejelentkezésnél sem lehet eltárolni a felhasználó nevét, illetve jelszavát, hogy elkerüljük a későbbi visszaéléseket. Ezt az elvet követte a felhasználói lokalizáció, melynek értelmében a munkavállalók csak a cég belső hálózatára csatlakozva tudják használni a szoftvert.

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 szoftver elkészítése mellett döntöttem. Külön figyelmet fordítottam a mobil optimalizálásra, így könnyedén áttekinthető az adott készüléken a futtatott program.

A rendszer egyszerre többfajta és nagymennyiségű adatot kezel, ezért kiemelten kellett foglalkozni az adatok kinyerésével, legyen az papíralapú (nyomtatás) vagy akár a Microsoft Office által széleskörben használt Excel programba való exportálási lehetőség is.

2.3. A szoftver funkciói

Két hónap alatt, közel 1000 új elem került az adatbázisba. A lekérdezések hatékonyságát, a szelektálás erőforrásigényét a minimumra kellett csökkenteni. Szinte minden egyes oldal, az összes táblát használja 1-1 lekérdezési folyamat során.

Az elkészült naptárrendszer arra szolgál, hogy tájékoztassa a cég vezetőségét azokról a munkatársakról, akik szabadságot kérvényeznek egy adott napra, illetve információt nyújt az azonos szakterületen dolgozók számára és létszámkorlátozásokat is végez. A szabadnapokon felül, a rendszer tájékoztatást ad az ünnepnapokról is.

A fizetési összesítés a munkalapok alapján készül. A program automatikusan létrehoz egy táblázatot melyen tökéletesen látható, hogy melyik munkatárs, mikor és hány órát dolgozott. Ezen felül a rendszer számolja a dolgozók órabérét, az összes ledolgozott órát, összesíti a személyek munkabérét, nyilvántartja az alapanyagköltségeket, illetve a cégvezetőknek tájékoztatást ad a várható kiadások összegéről.

2.4. Időhatékonyság növelése

A program egyik jelentős újításaként bevezetésre került a QR-kód szabvány használata, melynek segítségével a dolgozók mobiltelefonjaikkal az alkatrészeken elhelyezett kódokat beolvasva jutnak a szükséges információk birtokába. Ezzel a munkafolyamattal lényegesen lerövidítik az adatok rögzítésének idejét. A kód használatát a 1. ábra szemlélteti.



1. ábra. QR-kód olvasó használata munka közben

3. Elért eredmények

A szoftver kidolgozása 2016 novemberében kezdődött azzal a céllal, hogy a családi vállalkozás keretén belül egy jól működő vállaltmenedzselő programot hozzak létre megkönnyítve ezzel a vezetők és a dolgozók munkáját. A program azóta is folyamatos fejlesztés alatt áll, igazodva a növekvő igényekhez és változásokhoz. A tesztelések során felmerülő problémák folyamatos javítása mellett törekedtem a teljességre. Az általam készített program gyakorlati alkalmazása az idei év januárjában megkezdődött. A használat során bebizonyosodott a „kettősség elvének” működése, hiszen mind a munkáltató és a mind a munkavállalók kölcsönös megalégedéssel és bizalommal használják ki a program kínálta lehetőségeket. A pozitív visszajelzések arra ösztönöznek, hogy további fejlesztéseket végezzenek, hogy a kis- és középvállalkozások minél szélesebb körben használni tudják.

A szoftvert további két területen szeretném bővíteni a jövőben. Az egyik, a kommunikációs terület (partneri felület létrehozása) - segítségével a cégvezető további vállalkozásokkal és üzleti partnerekkel kapcsolatot tud létesíteni a megrendelések leegyszerűsítése érdekében. A másik, a személy- és vagyonvédelmi terület kiépítése, mely során a kamera rendszerek működtetését és a beléptető rendszer (mágneskártyás beléptetés) irányítását is a szoftver fogja kezelni.

Safe Kid Zones

Szerző: Ali Arsen

Felkészítő tanár: Kőrösi Gábor

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

1. Bevezetés

A szülők mindig aggódnak gyerekeik miatt, hiszen szeretnék tudni, hogy biztonságban vannak-e, ezért van, hogy naponta többször is felhívják őket ellenőrzés képen. Ez lehet a gyerekeknek idegesítő, ám belőről érzik, hogy ez valójában a saját biztonságuk miatt is fontos lehet. Hogy áthidaljam a szülői felügyelet és a „zaklatás” közötti átmenetet megoldásként egy erre a célra előrukkoltam egy új innovatív ötlettel. Az általam készített Safe Kid Zones megoldással a szülők előre meghatározhatnak bizonyos helyeket, ahová nem szeretnék, hogy menjen a csemete, illetve baj esetén a gyermek egy riasztó gombot használhat, mely megnyomásával értesíti a szülőt, ismerőst, hogy esetleg elrabolták vagy bajba került, így a gondviselők szinte azonnal tehetnek valamit, esetleg riaszthatják a rendőrséget.

2. Pályamunka megvalósítás

Az alap elgondolásomban egy alkalmazás és egy központi weboldal képezne a Safe Kid Zones megoldás kulcspontjait. Az alkalmazás az eszköz (gyermek) követést biztosítaná, még a weboldal a szülői felügyeletet tenné lehetővé.

Mivel manapság minden fiatal magánál tartja az okostelefonját, és ez be is határolható. E tulajdonság használatával számtalan „táv felügyeleti” megoldással rukkolhatunk elő, mellyel könnyen megtudhatjuk, ha a gyermekeink olyan helyen vannak, ahol nem szeretnénk, hogy legyenek. Ilyenkor a szülő kaphat egy értesítést (egyelőre SMS-ben) a gyerek tartózkodási helyével, így akár el is mehet érte, vagy csak felkeresi telefonon. Olyan eset is megtörténhet, hogy a gyerek eltűnik, elrabolják és ebben az esetben a gyerek egy gombnyomással jelezhet a szülőnek, aki végig követheti az útvonalát. Illetve a szülő, ha szeretné megtekintheti a gyermek aktuális helyzetét, és az eszköz töltöttségi szintjét a weboldalon.

Először, a program felépítését kellett megtervezni, hogy mi hogy nézzen ki. Azonban ez a készítés során nagyon megváltozott. Majd az algoritmust kellett megírni, amivel meghatározhatjuk, hogy az övezetben, vagy azon kívül van az eszköz, és értesítést küld a szülőnek. Kellott egy web szerver SQL adatbázissal, és megírni a web oldal adatbázisát, és működését. Miután ez elkészült, magát az alkalmazást kellett elkészíteni. Ahhoz, hogy az adatbázisból az adatokat eltároljuk a telefonon, kellett egy kommunikációs csatorna, amin keresztül

ezeket az adatok lekérdezzük, és eltároljuk az eszközön. Minden működött, és már csak az maradt hátra, hogy felhasználó barát legyen az oldal. Ezt egy ingyenes template-tel meg is oldottam és most már a célközönség számára is használhatóvá vált.

VR Escape Room

Virtual Entertainment

Gál Patrik

Felkészítő tanár: Tekse Marianna

Verseghy Ferenc Gimnázium, 5000 Szolnok, Tisza park 1.

1. Bevezetés

2016 a VR éve volt. Ebben az évben adták ki az első virtuális valóság eszközt hivatalosan. Ezeknek az eszközöknek a célja főként a szórakoztatás, azonban az újszerűsége miatt még nagyon kevés játékot készítettek ezekre az eszközökre. Az új betöltésére, a HTC Vive Virtual Reality eszközre fejlesztettem ezt a játékprogramot. A játék egy escape room, azaz magyarul egy szabadulószoba. A valóságban is létező szabadulószobák analógiájára - ahol legtöbbször az a cél, hogy egy adott időn belül logikai vagy ügyességi feladatok segítségével kijusson, kiszabaduljon a játékos a szobarendszerekből - készítettem el a programomat. Ezt az élményt akartam átadni VR-ban. Így a játékos akár saját lakásában is próbára tudja tenni logikáját ezzel az escape room-mal.

2. A VR Escape Room játékmenetének a bemutatása

A játék angol nyelvű, én magyar fordításban közlöm. A játékos először a kezdőszobába jut be, ahol megismerheti az alaptörténetet, és megláthatja az „indítás” gombot. A játék alaptörténete: „Tegnap téged elfogtak egy sötét sikátorban, és begyömöszöltek egy furgonba. Elkábítottak. Az utolsó dolog amire emlékszel, hogy egy mély hangú ember a füledbe suttog. Azt mondta, hogy egy órád van kiszabadulni, mielőtt elfogyna a levegőd.” Miután megnyomta a gombot, betöltődik a következő pálya, a szobarendszer, amiből ki kell jutni. A szobarendszer négy helyséből áll, amiket a következő pontokban részletezek. Egy jelenet a játékból és a szobarendszer az 1. ábrán látható.



1. ábra. Egy jelenet a játékból és a szobarendszer

2.1. Az első szoba

Ebben a szobában félhomály van, az egyetlen fényforrás egy ablakból beszűrődő fény. A játékosnak meg kell találnia a lámpa kapcsolóját, és ezáltal sokkal többet fog látni. Látni fog három darab ajtót, ami további szobákba vezet. A szoba tartalma: egy ablak, ami a londoni Big Benre néz, egy fali kép, amin az egyik zár kombinációja található, egy visszaszámláló, ami 1:00:00-tól számol vissza nulláig, és jelzi a játékosnak, hogy mennyi ideje van még hátra. Van egy szekrény, aminek van három fiókja, és ezekben különböző tárgyak vannak számmal ellátva (ezekre később rátérek), van egy EXIT felirat, ami mutatja a játékosnak, hogy melyik ajtó a cél, amelyikkel ki tud jutni a szobarendszerből. Továbbá található még egy ventilátor, ami szén-dioxidot pumpál a levegőbe. Ez kiszorítja a szobákból a belélegezhető gázokat, ez egy órán belül eléri a kritikus szintet, amibe „belehal” a játékos. Az ajtók előtt vannak lábtörlők, és az egyik rejti maga alatt a második szobának a kulcsát. Ezt be kell tenni a zárba, el kell fordítani azt 90 fokkal, és a kilincset lenyomva bekerülhet a következő szobába.

2.2. A második szoba

Itt még több bútort fog találni a játékos, mint az elsőben. Fiókos asztal, dupla-fiókos szekrény, akasztós szekrény, falikép, álló lámpa és egy ruhafogas. Mindegyiken és mindegyikben található egy számmal ellátott tárgy, ugyanúgy, mint az első szobában. Ezek a tárgyak a harmadik szoba ajtajának a kinyitásához szükségesek. Van még egy projektor elrejtve a szobában (nincs számmal ellátva), azonban ennek még később lesz szerepe.

El vannak még rejtve úgynevezett memória lapok is. Ezekből a memória lapokból három darab van, témakör szerint: konyhai eszközök, ruhadarabok és dísz tárgyak. Konyhai eszközök: sodrófa, tányér, serpenyő és kés; ruhadarabok: ing, kesztyű, sapka és cipő; dísz tárgyak: falikép, váza, cserepes növény és lámpaernyő. A lapokon matematikai egyenletek vannak, azonban a számok helyett az adott tárgyak sematikus képei láthatók. Egy példa a 3. ábrán látható. A lapokon látható tárgyak számaival a játékos elvégezi az egyenleteket fejből, és a lapok segítségével elmenti a végeredményeket.

$$\frac{(\ / + \bullet)}{(\bullet - \backslash)} = \blacksquare$$

2.ábra. Egy memória lap

2.3. A harmadik szoba

A játékos elvégezte az összes egyenletet, és kapott három végeredményt. Vissza kell mennie az első szobába, ahol megtalálja a harmadik szoba ajtaját, amin egy láncsal ellátott számszáras lakat van. Észre kell vennie a játékosnak,

hogy három számjegyet mentett el, és azt a három számot kell bepörgetni a lakatba. Azonban nem tudja a sorrendet. Ehhez ad segítséget az első szobában (amiben éppen van) a megtalált falikép, amin éppen a tárgyak témaköreit ábrázoló kép látható sorban, nyíllal ellátva. A lapokon elmentett számokat beüti a képen látható sorrendben a lakatba, és az a lánccal együtt leesik az ajtóról. Lenyomja a kilincset, és azonnal be tud menni. Egy tiszta fehér szobát fog megpillantani, amiben található: egy magas asztal; egy konnektor, mellette egy kábel; öt darab óra, melyek különböző időzónákat mutatnak, alattuk városnevek táblái és a falon egy falfirka. Rá kell jönnie a játékosnak, hogy a konnektor és a kábel a második szobában meglelt projektort fogja bekapcsolni. Átviszi ebbe a (3.) szobába. Bedugja mind a két oldalt, és a projektor egy térképet fog kivetíteni. Ezt a projektort úgy kell az asztalra tenni, hogy a térképen és a falon található illesztési pontok illeszkedjenek. Így a falon egy körülvárosi útvonal jön létre, ami hat városon megy át: New York, Rio de Janeiro, London, Sydney, Peking és Moszkva. Ennek a hat városnak az órái vannak a falon, egyet kivéve, a londonit, mert azt már a játékos az első szoba Big Benjéről beazonosíthatja. Ez a hat óra hat számkódot fog rejtetni, ami a negyedik, végső szobába fog vezetni. Ezt a hat számot be kell ütni a körülvárosi útvonal sorrendjében, a negyedik szoba ajtaja mellett lévő elektronikus számszárba. Meg kell nyomni az ENTER-t, lenyomni a kilincset, és máris bejut az ember a negyedik szobába. Amint bejutott, a program azonnal teleportálja a végszobába.

2.4. A végszoba

A végszoba akkor érhető el, ha bejut a negyedik szobába egy óra alatt, azonban, ha a visszaszámláló elér a nulláig, akkor ugyanígy bekerül a végszobába, csak másik feliratot fog látni. Ha nyert, akkor: „Gratulálok! Kiszabadultál x perc és y másodperc alatt.”, ha pedig vesztett, akkor: „Vesztettél! Meghaltál, mert elfogyott a levegőd. Újra próbálsz?”. Mind a két esetben a játékos előtt lesz két gomb: egy „újra” és egy „kilépés”. Ezzel a két gombbal újra próbálhatja a játékot, vagy csak egyszerűen kilép. Az egész játékmenetet meg lehet tekinteni ezen a Youtube linken: <https://www.youtube.com/watch?v=wgC1dL5y6Ys>

3. Probléma megoldásának menete

A játék Unity játékmotorra lett fejlesztve. Felhasználtam a Valve által kiadott ingyenes Asset-et, a SteamVr plugint. A HTC és a Valve Corp. által fejlesztett Vive-ra készítettem a játékomat. A Unityn belül az összes scriptet C#-ban írtam. A játékban található összes modellt és játékelemet a SketchUp modellező és szerkesztő programmal készítettem.

3.1. Irányítás és tárgyakkal való interaktálás

Először be kellett állítani a Vive játéktérületét, amin belül a játékos mozogni fog. Én ezt a legminimálisabbra beállítottam, ami 2mx1,5m és így a legkisebb

szobamérettel rendelkező felhasználó is tudja játszani. Felhasználtam a Vive-nak a headsetét és mind a két controllerét. A headset felelős a képi megjelenítésért és a hangért. A játékban még csak két hang van (azonban ez a jövőben változni fog): a ventilátor zúgása és az „utcaról” beszűrődő hang. Mind a kettő 3D hatású. Felhasználtam a kontrollereknek a trigger gombját és a touchpadot. Ha a játékos egy megfogható tárgyhoz közelíti a kontrollert (a tárgy zöldszínű lesz), és meghúzza a triggert a mutatóujjával, akkor az adott objektumot szabadon tudja forgatni collider nélkül és például meg tud rajta találni egy számot. A triggerrel kell megfogni a különböző forgatható vagy húzható, de rögzített tárgyakat. Ilyenek például a fiókok, a kilincsek, az ajtók és a már berakott kulcs. Ezek azonban nem gombnyomásra működnek, muszáj fizikai munkát befektetni (pl.húzni). A touchpad fontos a memória lapoknál és a számszáras lakatnál. A lapoknál jobbra vagy balra lehet csúsztatni a hüvelykujjunkkal a számokat 0-tól 9-ig. A lakatnál ugyanez a csúsztatás van 0-tól 9-ig, és a forgó korongok között le- vagy felkattintással lehet váltani. Az elektronikus számszár gombjainál nem használtam triggert, ott egyszerűen be kell nyomni a kontrollert a gombokba és benyomódnak. Azonban a indítás, újra próbálás és a kilépés gomboknál meg kell nyomni a triggert. A szobák közti teleportációt is triggerrel oldottam meg. Amint egy ajtó kinyílik, egy zöld felületet megnyomva tovább megy a játékos a kívánt szobába.

3.2. Fontosabb játékelemek

A második szoba ajtaja kulccsal nyílik, de ha a játékos beteszi a kulcslyukba, utána már nem lehet kivenni, csak elforgatni. A kilincset 45 fokban le tudja nyomni, és csak ezt megragadva tudja magát az ajtót is kinyitni. A ruhafogason, a vállfán és a lámpán lévő tárgyakat le tudja venni. Ha vissza akarja rakni, annak a helye sárga lesz, és egy animáción belül visszarakódik az eredeti helyére. A kábel két vége is animációval kerül be a megfelelő helyre, utána már nem vehető ki, a projektor el kezd világítani. A hármas és a négyes szoba számára ellenőrzi, hogy a betáplált kód egyezik-e a beírt kóddal, és csak ezután lehet kinyitni a kilincseket. A ventilátorban folyamatosan pörög egy propeller és particle effektek vannak körülötte.

4. Elért eredmények

Mielőtt elkezdtem volna az első játékomat, a programozási képességeim kimerültek a LEGO NXT blokk alapú rendszerében. 4 hónapja, a projektem kezdete óta, fórumokból és videókból szerzett tudásommal a C# nyelvet végre jól tudom alkalmazni. Bízom benne, hogy a játékommal sok VR rajongónak okozok örömet, és sok felhasználó logikai képességét tudom fejleszteni. Ezzel a játékkal a VR világának hiátusát szeretném betölteni.

Fogyasztásmérő

DevQuarter

Vida Imre

Felkészítő tanár: Fekete Balázs

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

1. Bevezetés

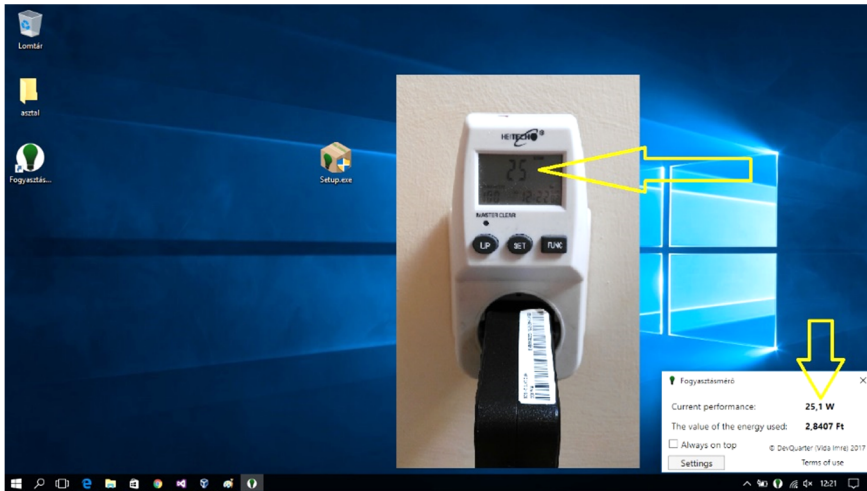
Manapság az emberek, és a cégek egyre nagyobb hangsúlyt fektetnek az energiahatékonyságra, és a környezettudatosságra. A sok számítógépet használó cégek energiafelhasználásának nagy részét a számítógépek, és az azokhoz csatlakoztatott eszközök teszik ki. Az egyes eszközök energiafogyasztásának mérése a cégek számára egy nagyon bonyolult és időigényes folyamat lenne a hagyományos fogyasztásmérő berendezésekkel. Ezért volt szükség egy olyan szoftveres eszköz, a Fogyasztásmérő program megalkotására, amely képes az egyes eszközök fogyasztását mérni, a felhasznált energia mennyiséget naplózni, és a felhasznált energia értékét is kiszámolni.

2. Probléma megoldásának menete:

A probléma meghatározása után a következő lépés a megoldás megtervezése volt. Első lépésként felvettem a kapcsolatot olyan emberekkel, és cégekkel, akik találkozhattak ezzel a fentebb leírt problémával, akik megerősítették, hogy a probléma valódi, hiszen a cégeknek nagyon nehéz dolguk van, amikor az energiahatékonyság növelését tűzik ki céljukként. Ezután kiválasztottam a szerintem erre a célra legalkalmasabb programozási nyelvet, a C#-ot, és elkezdtem a kódolást. A programozás során sok problémába ütköztem, hiszen egy ilyen alkalmazásnak nagyon sok mindent kell egyszerre figyelnie, a pontos mérés érdekében, de eközben nem terhelheti le, nem lassíthatja a futtató eszközt, és a fő szempont, hogy nem akadályozhatja a munkavállalókat az adott számítógépen végzett munkában. Ezen kívül a számításokhoz szükséges helyes képlet megtalálása is nagy kihívást jelentett. A program több hónap kemény munka és optimalizálás után kezdte megbízhatóan ellátni fő feladatait, és közel hibátlanul nyomon követni a számítógépek fogyasztását. Az első verzió képes volt nyomon követni a számítógépek fogyasztását közel $\pm 30\%$ -os pontossággal. Ekkor a program még a tesztelésre szánt számítógépet lelassította, a processzoridő nagy százalékát felhasználta, és körülbelül 50Mb RAM memóriát igényelt a futtatáshoz. A további optimalizációknak hála a program jelenlegi verziója nem lassítja a számítógépet, nem befolyásolja a

számítógépen történő munkavégzés hatékonyságát, mert lehetőség van az automatikus, háttérben történő futtatásra. Ezen felül a program jelenlegi verziója közel 20Mb memóriát igényel, és a pontossága a beállítások pontosságától függ. A helyes adatok beírása után $\pm 5\%$ -os pontossággal képes mérni, mint egy konnektorba dugható, hardveres fogyasztásmérő eszköz.

2.1. Ábrák



1. ábra. A program működés közben (mellette fogyasztásmérő eszköz, ami ugyanannak a számítógépnek a fogyasztását méri.)

2.2. Táblázatok

Az alábbi táblázat a program több számítógépen történő tesztelésének adatait mutatja, összehasonlítva az adott számítógép energiafogyasztását mérő külső fogyasztásmérő eszközzel.

Eredmények		
	A külső fogyasztásmérővel mért adatok:	A programmal mért adatok:
Laptop	25 W	25,1 W
Szerver	138 W	139 W
Laptop2	70 W	69,8 W
Asztali PC	215 W	215,2 W

1. táblázat. A program pontosságát bizonyító összehasonlító mérések

3. Elért eredmények

A program jelenlegi verziója hosszú kutató, és programozó munka után az alábbi funkciókat tartalmazza: A termék a megfelelő, minimális, a rendszerből le nem kérdezhető adat beírása után képes közel tökéletesen figyelemmel kísérni a számítógép hardware eszközeinek fogyasztását (ezt az 1. táblázat mutatja be), és a felhasznált energia értékét forintban kiírni. Ezzel lehetővé válik az adott eszköz energiafogyasztásának, ezáltal gazdaságosságának folyamatos nyomon követése külső fogyasztásmérő eszköz, illetve egyéb hardveres változtatások nélkül. A program alkalmas lehet szervergépek fogyasztásának nyomon követésére és a szerver fenntartási költségeinek kiszámítására is. A program használata nem csak magánszemélyek számára lehet hasznos, hiszen az egyszerre több számítógépet működtető cégek a program segítségével figyelemmel kísérhetik egyes eszközeik fogyasztását. Ez segítheti őket abban, hogy kiválasszák azokat a számítógépeket, amelyeket célszerű lenne korszerűsíteni, lecserélni az energiafelhasználás optimalizálása érdekében. A döntésüket segítheti egy olyan beépített döntéstámogató rendszer, amely az eddig mért adatok alapján előrejelzi az adott számítógép, vagy akár az összes eszköz várható fogyasztását adott időszakokra. A programot alkalmazó cégek vezetői és rendszergazdái saját számítógépen vagy okostelefonon futtatott alkalmazásukkal nyomon követhetik vállalatuk számítógépeinek pillanatnyi összefogyasztását, illetve heti, havi és éves statisztikáikat is megtekinthetik. A kliensprogramok által gyűjtött adatok egy külső kiszolgálón lévő adatbázisban kerülnek tárolásra, ennek meghibásodása esetén pedig a számítógép maga menti el az adatokat. Ezzel az információval más eszközök energiahatékonysági hibái is kiszűrhetők, hiszen, ha a számítógépek fogyasztása alacsony, de az adott hónapban felhasznált energia mennyiség nagyon magas, akkor egyértelműen következtethetünk arra, hogy nincs szükség informatikai korszerűsítésre, de annál inkább szükség van az egyéb eszközök (például: világítás, hűtőszekrény) energiahatékonysági felülvizsgálatára. Az alkalmazás használata nem gátolja a dolgozókat a számítógépes munkavégzésben, hiszen a program végig rejtve marad, a háttérben fut. A program telepítése és beállítása sem igényel különös informatikai tudást, hiszen a mellékelt használati utasítás lépésről lépésre részletesen leírja ennek menetét, és a teendőket. A program alacsony rendszerigényű, így az nem terheli le a futtató eszközt, valamint az eszköz energiafogyasztását sem befolyásolja.

A további fejlesztési célok közé tartozik:

- Cégvezető, vagy rendszergazda figyelmeztetése a beállított összeg elérésekor.
- Adott művelet végrehajtása a beállított összeg elérésekor (pl.: leállítás, energiatakarékos mód bekapcsolása)

- a felhasznált energia hardver eszközönkénti megtekintése
- széleskörű tesztelés (több nagyobb cég, pl. a National Instruments Hungary Kft. képviselőivel is vannak konzultációk)

PlatStreamer

Verseghy Coders

Kovács Marcell, Szepesi Zoltán

Felkészítő tanár: Tekse Marianna

Verseghy Ferenc Gimnázium, 5000 Szolnok, Tisza park 1.

1. Bevezetés

A videójátékok legfőbb célja az interakció, a játékosok ezáltal kerülnek kapcsolatba a programmal. Mi tettünk egy merész lépést, hogy ezt az interakciót egy magasabb szintre emeljük és már ne csak a játékos, hanem a nézők, vagy akár esportok esetén a szurkolók is bekapcsolódhassanak, így jött létre a PlatStreamer. A nézők interakcióba való bevonását, mi a mostanra nagyon népszerűvé vált streamekkel képzeltük el, de, ez csak egy felület, ami könnyedén lecserélhető. Mint ahogy a játék nevében található szójátékból kikövetkeztethetjük, a program alapjaiban egy platformer játék, mivel ezt a műfajt találtuk legalkalmasabbnak terveink megvalósítására, amire egy gyors (fast-paced) játékokat tudtunk teremteni.

A játék célja, hogy egy mini gamekkel teli platformert játszanak a felhasználók. Ez önmagában nem lenne meglepő, ám itt a játék menetét a Twitch chat segítségével befolyásolhatják a nézők is. Úgy gondoljuk, hogy ez a videójátékok és remélhetőleg az e-sportok jövője is, mivel a játékok interakciót biztosítanak nem csak a játékosokkal, hanem a nézőkkel is.

A nézők a játékmódot szavazhatják, bónuszokat adhatnak a játékosoknak (ezen utóbbi funkció még nem tökéletes, további fejlesztést igényel).

2. Probléma megoldásának menete

A négy játékost jelenleg egy számítógépről lehet irányítani (a hálózati irányítás még fejlesztés alatt áll). WASD, IJKL, nyilak és a numpaden található 8456 gombokkal tudunk mozogni. Lőni a Q, az U, a space és a numpad 7-es gombjával lehet (amennyiben ilyen a gamemode). Az első játékmód mindig a rifle elnevezésű játékmód utána pedig véletlenszerűen követik egymást, valamint 1-a 4-hez az esélye, hogy minden mód után szavazásra bocsátják a twitch chatben, hogy mi legyen a következő. Ha szeretnénk az F2 gomb segítségével képernyőképeket is készíthetünk a játék folyamán, ezek a screenshots/mappába automatikusan mentésre kerülnek.

Pályák egyszerűen létrehozhatók, a /resources/maps/ mappába rakhatunk pályákat vagy módosíthatjuk a jelenlegit, ezekhez saját textúrákat is rakhatunk a resources/textures/ mappába a pálya nevének megfelelő mappába rakva a textúrákat. A pályákat .map formátumban kell létrehozni (a

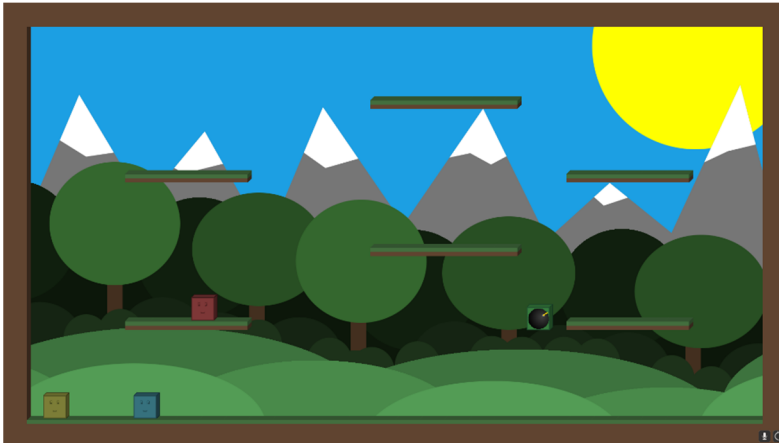
példa map1.map alapján, ezt használja most a program), a textúrákat, pedig .png formátumban.

A map fájlok először a játék verziószámát tartalmazzák, majd minden új sorban a platformok részeinek koordinátáit találhatójuk, ezt egy kettőskereszt (#) követi és az ez alatti sorok a platformok hitbox-ait tartalmazzák.

Jelenlegi Játékmódok:

- **Bomb Tag** (szavazásnál „bombtag”)

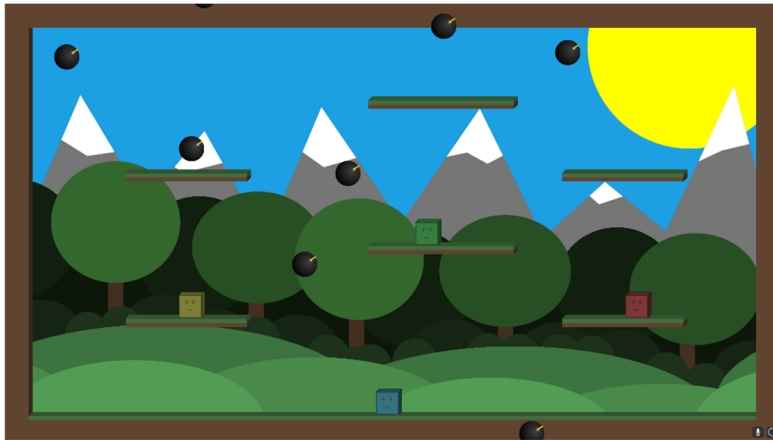
A Bomb Tag játékmód lényege, hogy a négy játékos egyike véletlenszerűen megkapja a bombát, ami egy 10 és 15 másodperc közötti intervallumon belül fel fog robbanni. Amennyiben hozzáér egy másik játékoshoz átadja annak a bombát és fél másodperc védelemért kap, hogy ne lehessen neki azonnal visszaadni. Amint a bombát tartó játékos felrobban, a játékos kiesik és újra véletlenszerűen valakihez kerül a bomba, ez addig ismétlődik amíg csak egy játékos marad a pályán.



1. ábra

- **Bomb Rain** (szavazásnál „bombrain”)

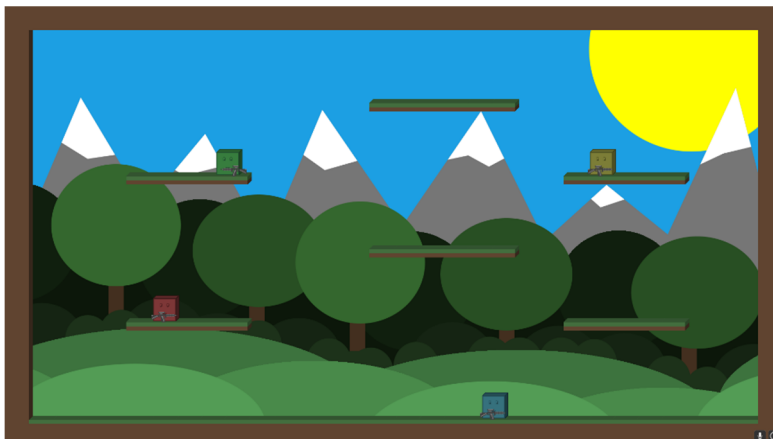
A Bomb Rain nevű játékmódban egy bomba „esőben” található magunkat, itt az a cél, hogy kikerüljünk a zuhanó bombákat. Amennyiben valakit eltalál egy bomba, a bomba felrobban, a játékos pedig kiesik a körből. A játék addig tart, amíg csak egy játékos marad a pályán.



2. ábra

- **Rifle** (szavazásnál „rifle”)

A rifle játékmódban minden játékos rendelkezik egy mesterlövész puskával, ezeknek a puskáknak nagy előnye, hogy egyetlen egy lövéssel képesek bármelyik másik játékost megölni, de hihetetlenül sokáig - 5 másodpercig - tart újra tölteni. Ez a felállítás négy játékosal még gyorsabbá és élvezetesebbé teszi a játékot, de itt már komoly szerepe van a taktikának is. A játékmód addig tart még egyetlen egy játékos marad életben a pályán.



3. ábra

2.1. Technikai kivitelezés

A játék C++-ban íródott az SDL és annak alkönyvtáraival, az SDL_Net-el (TCP), curl-el (Twitch token ellenőrzés)

A fejlesztésben használtuk az egyik csapattag (Kovács Marcell) által írt C++ alapú játékmotort, amely dll-ként van csatolva, de nyílt forráskódú és elérhető a következő oldalon: <https://bitbucket.org/kovacsmarcell99/mengine-sdl>

A fizikai szimulációkhoz semmilyen könyvtárat nem használtunk, azok a program részei.

Igyekeztünk a C++11 és a C++14 szabvány adta lehetőségeket is kihasználni.

A programkód felépítésének és a saját játékmotornak köszönhetően a további játékmódok hozzáadása egyszerűvé vált a GameMode class egy child osztály létrehozása után az Update és Draw függvényeket kell implementálni.

A forráskódban egy twitchAPI mappába helyeztük el a Twitch integrációhoz írt kódjainkat, melyik szintén saját készítésűek.

A fejlesztés a Visual Studio 2015 segítségével történt. A modern projekteknél már megszokott eljárásokat is használtunk, mint például a source control (git). A forráskódot, így egy online git repository-ban tudtuk tárolni és bármilyen módosítást is végeztünk a kódon azt visszaállíthattuk a korábban már tesztelt és megírt forráskódokra, ami nagyban megkönnyítette a munkát.

A programozás nagy részét Kovács Marcell végezte, mivel ő már gyakorlottabb a C++ programozásban és otthonosan mozog ebben a témában, még a grafikák a pályák, valamint a Twitch-el való kommunikáció kódjának egy részét Szepesi Zoltán írta, mivel ő már rendelkezett ezzel kapcsolatban tapasztalatokkal (javascript-el már használta régebben).

A játékban látható összes grafika saját készítésű. A hangeffektek és a zene vagy Creative Commons Attribution 3.0 License alatt elérhetőek vagy public domain-ek, szükség esetén, a játékhoz csatolt fájlban megneveztük a forrást.

A programot, a saját grafikákat és a forráskódot szerzői jogok védik. A program üzleti célú felhasználása, továbbítása, vagy módosítása a készítők engedélye nélkül tilos.

3. Összegzés és további tervek

A játék alapjai készen vannak és nagyon biztosak, már csak néhány kisebb hibajavításra van szükség és néhány egyszerűbb funkció hozzáadására, hogy minden teljesen üzemeljen. A játékmódok bővítése viszonylag egyszerűen megvalósítható. Amire még szükség lehet az a nagyobb tartalom: több játékmód, pályák, több nézői interakció. Továbbá a hálózaton történő játék teljes implementálása.

Ezek közül elsődleges tervünk, amely egyben a célunk is a játékkal, a nézői interakció minél nagyobb kibővítése.

A TE szótárad

Kébel Zsolt

Felkészítő tanár: Kelemen Mihály, Bordi István

Tóth Árpád Gimnázium, 4024 Debrecen, Szombathi István utca 12.

1. Bevezetés

„Az igazi határokat a beszélt nyelvek határozzák meg.”¹ Szerintem senkit nem kell győzködni arról, hogy a nyelvtudás mennyire meghatározza jövőbeli lehetőségeinket. Éppen ezért én is igyekszem elsajátítani néhányat a rengeteg idegen nyelv közül. Azonban ahhoz, hogy sikeresen tudjunk használni egy nyelvet mindenek előtt az adott nyelv szavait és kifejezéseit kell megtanulni. Erre a célra egy szótárfüzet a legalkalmasabb.

Év elején támadt egy ötletem: Mi lenne, ha ez a szótárfüzet mindig ott lenne velünk, nem szakadnának el a lapok benne egy rosszul sikerült táskába hajítás után, nem lennének benne utólag olvashatatlan a sietség következtében összerántott betűkkel leírt szavak, esetleg még magától ki is kérdeznék a tanulandó szavakat?

Így körvonalazódott bennem ez az Android applikáció, melynek nemes egyszerűséggel a *MyDictionary* nevet adtam.

2. Probléma megoldásának menete

A program fejlesztése rengeteg időbe telt/telik, jelenleg is fejlesztés alatt áll. Lassan három-négy hónapja dolgozom rajta. Az ábrákon **(1)** jól megfigyelhető az egyre letisztultabbá válása.



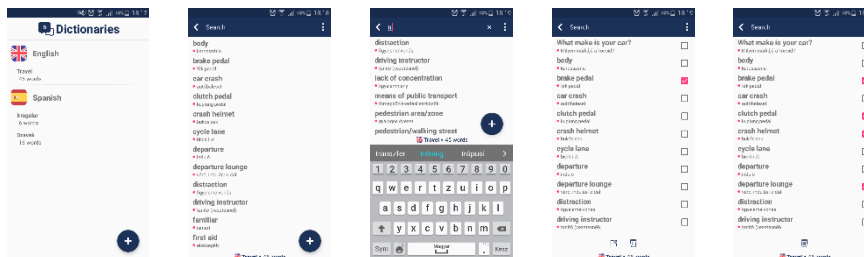
1.ábra. A program felhasználói felületének a fejlődése

Az adatbázis kezelés része volt a legtöbb fejtörést igénylő feladat, mivel ezeket az adatokat (a szótárakat és szavakat) egy a telefonon létrehozott adatbázisban tárolja a program.

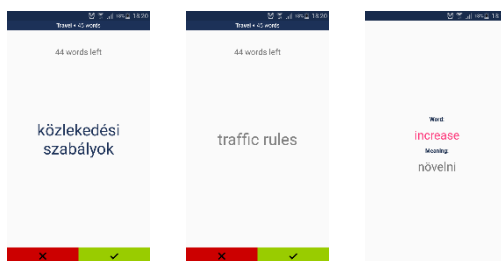
Kapott egy plusz funkciót a hagyományos szótárfüzetekhez képest. Amikor belépünk egy témakörbe az abban szereplő szavakat kikérdeztethetjük. Ezzel tanulhatjuk a szavakat vagy ellenőrizhetjük tudásunkat.

3. Elért eredmények

A program fejlesztése során rendszeresen figyeltem a visszajelzésekre tanáraitótl, és főleg osztálytársaimtól, akik közül néhányan tesztelik az applikációt.



A Tanulás, avagy „Learn” funkciót előszeretettel használják a jelenlegi felhasználók. Hasznosnak találják és néhányan már készültek fel ezzel szódolgozatra, ami a funkció eredményességét és gyakorlati hasznát mutatja. Az app jelenlegi verziójáról látható ábra (2).



2.ábra. Az app jelenlegi kinézete

¹ idézet Jean-Jacques Annaud-tól

Informatikus képzés a Szegedi Tudományegyetemen

SZTE Informatikai Intézet

Intézetünk 1990-ben alakult Informatikai Tanszékcsoport 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 részvétellel. Természetesen a hallgatóink bekapcsolódhatnak a tanszékeken folyó kutatásokba is. Ennek eredményeként hallgatóink a mai napig is számos színvonalas kutatási eredményt mutattak be a rendszeresen rendezett Helyi, és Országos Tudományos Diákköri Konferenciákon.

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

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

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

Képzéseink

Programtervező informatikus

Számtalan olyan eszközzel, berendezéssel találkozhatunk, amelyek alapvető működését valamilyen program (szoftver) biztosítja. A programtervező informatikus képes szoftverek létrehozására, továbbfejlesztésére, karbantartására egyénileg, vagy csoportmunkában. A programtervező informatikus szakon hallgatóink elmélyíthetik matematikai ismereteiket, megtanulják a számítástudomány és a logika alapjait, megismernek többféle programozási nyelvet, és megtanulják hogyan alkalmazzák ezen ismereteket az optimalizálás, a mesterséges intelligencia, a szoftverfejlesztés és a képfeldolgozás területein. Hallgatóink számára lehetőséget biztosítunk a következő specializációk elvégzésére: szoftverfejlesztés, képfeldolgozás, informatikai modellalkotás.

Gazdaságinformatikus

A gazdaságinformatika az informatikai és közgazdász szakterületek kombinálása, különös tekintettel az informatika alkalmazásaira és az elemző munkára. A Gazdaságinformatikus képzésünkben nagyon fontos szerepet játszanak a matematikai tantárgyak, ezen kívül a hallgatók gazdasági ismereteket is szereznek (többek között mikro – és makroökonómia, menedzsment, marketing), hogy informatikai tudásukat elsősorban gazdasági területeken, optimalizálásban, mesterséges intelligenciával támogatott döntéshozatalban, vállalati információs rendszerek kezelésében és fejlesztésében kamatoztathassák.

Mérnökinformatikus

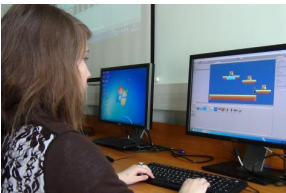
A mérnökinformatika az informatikai és villamosmérnöki területek kombinálása, az informatika műszaki alkalmazása. Képzésünkön a hallgatók komplex fizikai, matematikai és informatikai tudásra tesznek szert. Megismerkednek a robot technológiával, az áramkörtervezéssel, szenzorhálózatokkal. A kurzusok többsége jól felszerelt laborjainkban zajlik, ahol, gazdag választékot kínálunk a vezérlőrendszerek programozásától, a grafikus processzorok programozásán keresztül a mérnöki fejlesztések orvosi informatikai, valamint gazdaságinformatika alkalmazásáig. Hallgatóink számára lehetőséget biztosítunk a következő specializációk elvégzésére: szoftverfejlesztés, képfeldolgozás, ipari informatika, intelligens műszaki rendszerek, beágyazott műszaki rendszerek.

Diákoknak kínált programjaink

Felsőoktatási tevékenységünkön túl eltökélt célunk az informatika, mint tudományág megismertetése a fiatalabb generáció számára is. Természetesen nem a tételek és bonyolult matematikai összefüggések irányából tesszük mindezt, hanem a diákok érdeklődését felkeltve, látványos bemutatókkal, érdekes szakkörökkel igyekszünk újra és újra előrukkolni.



Egyik, talán leglátványosabb fórum a tudomány iránti figyelem felkeltésre a **Kutatók Éjszakája** programsorozat. Ezen a rendezvényen intézetünk már 2013 óta rendszeresen részt vesz és mindig új témákkal készül.



Az utóbbi időben egyre színesebb programokat kínál az érdeklődőknek a **Lányok Napja** rendezvénysorozat, ahol egy-egy szakma képviselői keltik fel az érdeklődést szakmájuk iránt. Az utóbbi két évben, mi is tartottunk interaktív bemutatókat a program keretén belül.



Előzetes egyeztetés alapján iskolás csoportok számára **laborlátogatással** egybekötött bemutatókat tartunk, amelyeken munkatársaink ritkán látott eszközökkel és technológiákkal ismertetik meg a látogatókat. Ezen kívül meg kell említenünk **szakköreinket**, amely egy féléven keresztül tartott különórákból állnak.

Az itt felsorolt programokon túl várunk minden kedves érdeklődőt Nyílt napjainkon is!

Támogatóink

Támogatóink



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



MinervaTel
www.minervatel.com/hu



CryptTalk
Hívó- és üzenetköltség

CryptTalk
www.crypttalk.com/hu



CAS Software
www.cas-software.hu



**3i Fejlesztő és Szolgáltató
Kft.**

www.3i.hu



ICBTech

www.icbtech.rs



Szent-Györgyi Albert Agóra

www.agoraszeged.hu



**SZTE Idegennyelvi
Kommunikációs Intézet**

www.lingo.u-szeged.hu



**SZTE TTIK Informatikai
Intézet**

www.inf.u-szeged.hu

A rendezvény az NTP-TV-16-0010 azonosító számú pályázat keretében, az Emberi Erőforrások Minisztériuma és az Emberi Erőforrás Támogatáskezelő, Nemzeti Tehetség Programjának támogatásával valósult meg.



EMBERI ERŐFORRÁSOK
MINISZTERIUMA



EMBERI ERŐFORRÁS
TÁMOGATÁSKEZELŐ



Nemzeti
Tehetség Program



Kiadta: SZTE Informatikai Intézet

Készítette: Makan Gergely és Vadai Gergely

Design: Németh Gábor

Nyomda: Innovariant Nyomdaipari Kft.

Készült: 100 példányban



CryptTalk
Hír- és kommunikáció

