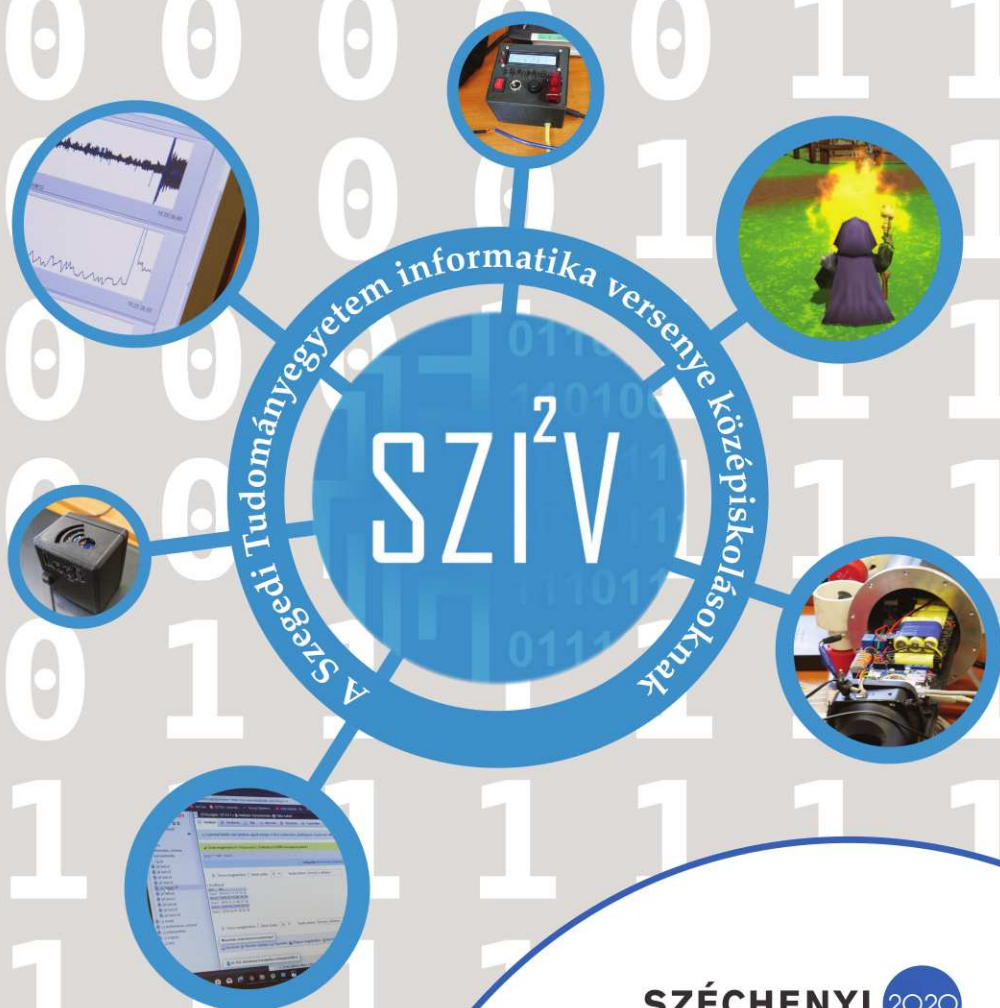


Szegedi Innovatív Informatika Verseny

2020



SZÉCHENYI



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

SZIIV Program (2020. november 6. – péntek)

Időpont	Műszaki informatika szekció	Informatika szekció
9:00–9:30	Megnyitó	
9:40–12:00	Előadások I.	
9:45–10:10	Dudás Miklós: BOTBlocks	Do I know it?: I Know It!
10:10–10:35	Janus3D_JM2020: JanusMeter 2020	Dz: Caley
10:35–11:00	FT-PD tech: Pókrobot és annak irányítása	Tűzvarázsló: Új generációs receptoldal
11:00–11:25	Informer: Házi készítésű CNC gép	TODO: Change name: Menetrend
11:25–11:50	Irinyi: Vezeték nélküli kesztyű	Házizz velünk: ParkMyst
13:00–15:30	Előadások II.	
13:00–13:25	Szixi: Fröccsgép	INeedMoreFreeTime: GraphChat
13:25–13:50	Irány a RALLYE!: Házilag készített autószimulátor versenyjátékokhoz	Janus3D_BuildUp: BulidUp
13:50–14:15	Robot TÉP 2020: AGV Robot készítése	MilanCodes: MyStudiez - Órarend alkalmazás diákoknak
14:15–14:40	Erdélyi Innovációs Műhely: Rusty	MakulátLANok: Communit.me
14:40–15:05	SunFlower: Pyrheliométer Nap-követő rendszerrel	Team Face: Face Trace

KTPK Program

2020. november 6. – péntek
A három verseny döntője, záró rendezvénye
Szegedi Innovatív Informatika Verseny
Polygon Pályázat Matematikából Középiskolásoknak
Játsszunk fizikát! Öveges József Emlékverseny
2020. november 7. – szombat
Interaktív előadások és játékos versenyek – Együtt a Jövő Mérnökeiért Szövetség, Eredményhirdetés



Tartalomjegyzék

<i>SZIIV</i>	1
<i>KTPK</i>	2
<i>Előszó</i>	5
<i>Programbizottság</i>	6
<i>Műszaki informatika szekció</i>	9
Dudás Miklós:	
BOTBlocks	11
Kis-Bogdán Kolos, Hegedüs János:	
JanusMeter 2020	15
Papp Tamás, Füstös Dániel:	
Pókróbot és annak irányítása	19
Piri Pál, Zabos Péter:	
Házi készítésű CNC gép	23
Sonkoly Gergő, Hentes Kornél:	
Vezeték nélküli kesztyű.....	27
Viktor Kristóf, Petrovics Dávid:	
Fröccsgép	31
Csizmarik Dániel:	
Házilag készített autószimulátor versenyjátékokhoz.....	35
Beleznai József, Pap Dániel Péter, Győző Róbert:	
AGV robot készítése.....	39
Kovács Nóra-Anna:	
Rusty	43
Bejő Mátyás, Hargitai Benke, Sztojka Áron:	
Pyreheliométer Nap-követő rendszerrel	47

<i>Informatika szekció.....</i>	<i>51</i>
Tóth Tamás:	
I Know It!.....	53
<i>Fejes Dániel, Várkonyi Dániel:</i>	
Caley	57
Durucz Ádám István:	
Új generációs receptoldal	61
Futaki Bálint, Vajda Simon:	
Menetrend	65
Horváth Gergely, Stefán Kornél, Vad Avar:	
ParkMyst.....	69
Laczkó Örs:	
GraphChat.....	73
Rendes Botond, Kis-Bogdán Kolos:	
Build ^{Up}	77
Herke Milán:	
MyStudiez – Órarend alkalmazás diákoknak	81
Szijártó Bence, Cselényi Dániel, Csorba Ákos Bence:	
Communit.me	85
Horváth István:	
Face Trace	89
<i>Együttműködő partnereink.....</i>	<i>93</i>

Előszó

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

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

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

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

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

A versenyre a két szekció zsűrije 14 belföldi és határon túli iskolából beérkezett 23 pályamű alapján összesen 20 csapatot válogatott be a szegedi döntőbe, mely 2020. tavasza helyett végül 2020. novemberében kerül megrendezésre rendhagyóan online módon.

Ez a kiadvány a döntőn bemutatásra kerülő 20 pályamunkát foglalja egy kötetbe.

Szeged, 2020. november

Dr. Vadai Gergely
a SZIIV verseny főszervezője

Programbizottság

Műszaki informatika szekció

A zsűri elnöke:



Dr. Pletl Szilveszter
főiskolai 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
intézetvezető, tanszékvezető egyetemi docens
SZTE Képfeldolgozás és Számítógépes Grafika Tanszék

A zsűri tagjai:



Dr. Vinkó Tamás
egyetemi docens
SZTE Számítógépes Optimalizálás Tanszék



Dr. Kertész Attila
egyetemi docens
SZTE Szoftverfejlesztés Tanszék



Fritz Zsombor
fejlesztési központ vezető, Szeged
Lufthansa Systems Hungária Kft.



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

Szervezők



Dr. Vadai Gergely
adjunktus
SZTE Műszaki Informatika Tanszék



Dr. Németh Gábor
adjunktus
SZTE Képfeldolgozás és Számítógépes Grafika Tanszék



Dr. Jász Judit
adjunktus
SZTE Szoftverfejlesztés Tanszék



Dr. Kincses Zoltán
adjunktus
SZTE Műszaki Informatika Tanszék



Tóth-Ságiné Kun Eszter
ügyintéző
SZTE Informatikai Intézet



Schäffer László
tanársegéd
SZTE Műszaki Informatika Tanszék

Műszaki informatika szekció

BOTBlocks

Dudás Miklós

Felkészítő tanár: Harangozó Zsolt

BMSzC Bláthy Ottó Titusz Informatikai Szakgimnáziuma, Budapest

1. Bevezetés

A BMSzC Bláthy Ottó Titusz Informatikai Szakgimnáziumának diákja vagyok és az iskola Arduino Szakkörének (BOTDuino) tagja. Iskolánkban szoftverfejlesztő és rendszerüzemeltető szakokra készítenek fel a közismereti tanulmányaink mellett, ezért nem is tanulunk elektronikai ismereteket. Ennek hiányában nagyon sok fennakadást okozna egy-egy kapcsolat megépítése, eleinte kevés sikerélményt adna, ezzel drasztikusan csökkenve az érdeklődők lelkesedését.

Az elektronikai akadályokkal szemben viszont motiváló, amikor a programozás tanulatása közben nem csak a képernyőn jelenik meg valami, hanem akár a környezetünk manipulálására, érzékelésére is van lehetőségünk.

Felmerült az ötlet, hogy készíthetnék egy elektronikai eszközökből álló szettet, amelynek segítségével a tanulókat megkíméljük (legalábbis a kezdetekben) a kapcsolások rejtelmétől, így csak a szoftvert kell elkészíteniük egy-egy működő projekthez.

2. Probléma megoldásának menete

A tervezés során több különféle mikrovezérlő megvizsgálásra került, de végül az Arduino alapú megoldás mellett döntöttem, mivel ez az eszköz az eddigi projektjeim alapján megbízhatónak minősült, illetve kifejezetten felhasználó barát a programozási környezete, valamint elterjedt eszközzől beszélünk, ami mögött nagy közösség áll.

2.1 Az alap koncepció

Az alap koncepció az volt, hogy készítssek egy olyan készletet, amely modulokból, úgynevezett Block-okból épül fel, így a téves bekötéseket kiküszöbölhetjük, illetve használatukhoz elektronikai ismeretekre sincsen szükség. Nehéz döntés volt, hogy melyik Arduino paneleket használjam, de az elterjedtségük miatt az Arduino UNO és Mega panelekhez fejlesztettem shield-eket, majd elgondolkoztam, hogy valami Wifi-s mikrovezérlőre is szükségünk lehet - például, egy IoT projekt bemutatására - így kibővítettem az eddigi tervezetet még egy Wemos D1 mini shielddel.

2.2 A Block-ok megtervezése

A második nehezebb döntés az volt, hogy milyen Block-okat készítsék. Itt többféle modul megtervezésre került, de végül az 1-es táblázatban látható Block-ok kerültek legyártásra. A nyákok tervezése során fontos szempont volt a rugalmas felhasználhatóság. A következő probléma az volt, hogy milyen típusú vezetékekkel lehessen összekapcsolni a Block-okat, illetve a Shield-et. Itt is többféle csatlakozó típust szemügyre vettem, de végül a 10-eres szalagkábel mellett döntöttem, annak megbízhatósága és strapabírósága miatt.

Az elkészült Block típusok	
Led modul	Szükség szerint szerelhető SMD, illetve THT leddel.
Gomb modul	A nyák alkalmas nyomógomb, DIP kapcsoló, illetve billenő kapcsoló beforrasztására is
Analóg modul	Analóg Potenciométerrel szerelhető.
Wire modul	DS18B20 típusú hőmérővel kompatibilis.
Led Mátrix modul	25darab WS2812B típusú címezhető színes ledből álló mátrix, ami különféle effekteket, illetve karaktereket (Emojikat) tud megjeleníteni.
7 szegmenses modul	A modul 1db 7 szegmenses kijelzőt tartalmaz, amit egy 74HC595 típusú shift regiszter hajt meg.
Kör alakú led modul	Ezen a modulon 8db led helyezkedik el kör alakban, amelyek vezérlését szintén a 74HC595 típusú shift regiszter végzi.
Breadboard modul	Egy 10pin-es tűkesorra ki lett vezetve a szalagkábel 10 pinje, s ez alatt helyezkedik el egy 170 pin-es breadboard.

1. táblázat: A Block típusok

2.3 A könyvtár megírása

A következő lépés az volt, hogy írjak egy olyan könyvtárat, amely meghatározza, hogy az adott csatlakozóban az adott csatlakoztatott modulnak az Arduinon lévő kivezetését, illetve, hogy az összes modul 1 vagy 2 utasítással használható legyen így nincsen szükség specifikus hardver tudásra, ami a kezdőknek jelentősen megkönnyíti az elindulást.

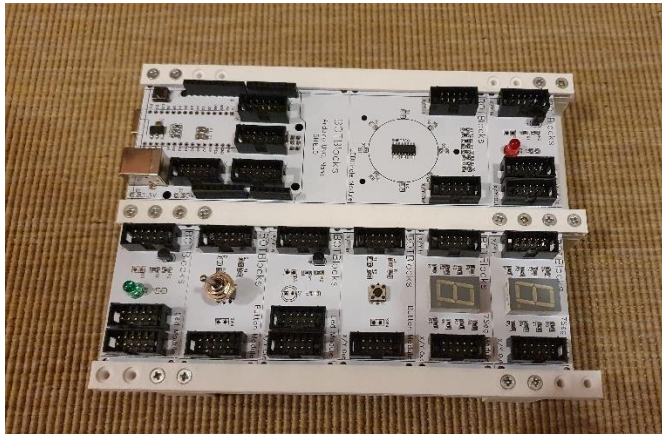
2.4 Az inicializáló kód generátor, a részletes modul leírások, illetve a tanmenet elkészítése

A fejlesztés alatt álló online felületen lehetőség van a könyvtárat inicializáló kód létrehozására, ehhez a felhasználónak csak ki kell választania, hogy milyen shieldet használ és hogy milyen modulokat csatlakoztatott hozzá, majd rányomni a megnyitás új ablakban vagy a letöltés gombra, ami egy kész INO fájlt fog letölteni, ez tartalmazza a legenerált kezdő kódot. Az elkészült kódgenerátor látható a 2. képen. A weboldalon megszerkesztésre került az összes modulhoz egy-egy részletes leírás, illetve egy-egy kezdő kód, illetve annak a magyarázata.

Emellett folyamatban van egy 12 alkalomból álló tanmenet is, ami különböző feladatokat tartalmaz majd, amelyeket egy esetleges órán be lehet mutatni. A fejlesztés alatt álló weboldal a <http://www.botblocks.hu/> címen bármikor megtekinthető.

2.5 A tervek a jövőre nézve.

- Saját vezérlő panel elkészítése, amelyen FT232 típusú programozó IC kapna helyet, illetve egy ATmega328-AU típusú mikrovezérlő.
- Lehetőség szerint bevezetni az oktatásba, akár egy szakkör keretei között az 1. képen látható készletet.
- A könyvtár publikálása, hogy az Arduino IDE-ből könnyen telepíthető legyen.
- Még több típusú Block legyártása, hogy több különböző projektet lehessen elkészíteni a készlet segítségével.
- Könyvtár továbbfejlesztése a Mega és a Wemos mikrovezérlőkhöz, illetve később a saját vezérlőpanelhez is.
- A tananyag elkészítése.



1. ábra: Egy szett

The code generator

UNO	X	Button	DS18B20	WS2812B
	X	DS18B20	Led	Button
	A	Breadboard		
	X	Button	Button	Led

[DOWNLOAD CODE](#)
[OPEN CODE IN NEW TAB](#)

2. ábra: A kód generátor

3. Elért eredmények

- 5 darab komplett készlet legyártásra került.
- Arduino, Wemos kompatibilis shieldek tervezése.
- Több különböző block elkészítése, tervezése.
- Nyomtatott áramkörök legyártása, illetve összeállítása.
- Folyamatosan bővülő könyvtár létrehozása.
- Használatot támogató webes felület létrehozása, folyamatos fejlesztés.

JanusMeter 2020

Janus3D_JM2020

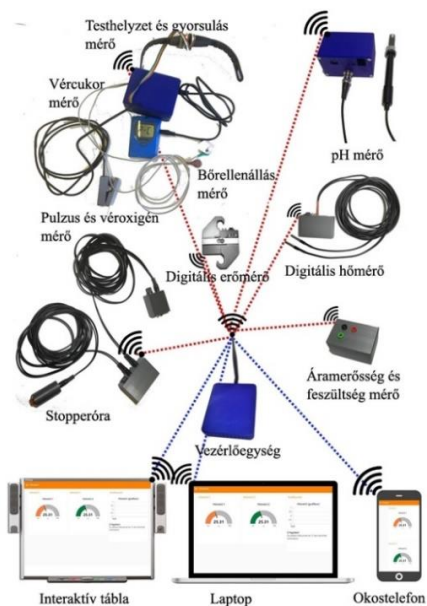
Kis-Bogdán Kolos, Hegedüs János

Felkészítő tanár: Dlusztus Péter

Pécsi Janus Pannonius Gimnázium, Pécs

1. Bevezetés

Célunk egy oktatást segítő komplex mérőműszer (rendszer) kialakítása volt, elsősorban fizika, kémia, biológia és egészségtan órákhoz. A termék előállításához és használatához elengedhetetlenek a megfelelő matematikai és informatikai ismeretek. A műszer segítségével a tanórán végzett kísérletek szemléletesebbek, gyorsabbak és pontosabbak. Továbbá, az egészségügyi mérőeszköz használata előtérbe helyezheti az egészséges életmódra való nevelést, és annak fontosságát.



1. kép: Rendszerábra

2. Probléma megoldásának menete

A JanusMeter egy univerzális digitális mérőrendszer természettudományos órákhoz (1.kép: Rendszerábra). A mérésekhez szenzorokat használunk. A mérőegységek a különféle mért adatokat WiFi-n

keresztül továbbítják egy vezérlőhöz, ami továbbítja a megkapott értékeket a számítógép és a diákok okostelefonjai felé. A számítógépen futó saját fejlesztésű megjelenítő program pedig MQTT-n (mqtt.org) keresztül kommunikál a vezérlővel, ami kapcsolatban van az összes eszközzel. A program a mért adatokat kiírja és grafikonon ábrázolja. Minden mérőegységben van egy WeMos D1 Mini (wemos.cc) mikrovezérlő, ami kiolvassa az érzékelő adatát, feldolgozza, majd elküldi a vezérlőnek WiFi-n keresztül. A mérőegységek lítiumos akkumulátorral üzemelnek, és saját fejlesztésű program fut rajtuk. A mérőeszköz dobozai saját tervezésűek és készítésűek, 3D nyomtató segítségével készültek (goo.gl/jKcRV5). A mérőműszerek kalibrálását is saját magunk végeztük.

Az "AGY", azaz a vezérlőegység: A JanusMeter "agya" egy egyedileg tervezett 3D nyomtatóval készített doboz, melyben található egy Raspberry Pi (raspberrypi.org) számítógép, és egy TP-link router.

A felhasználói felület: A felhasználói felületet egy Node-Red (nodered.org) nevű, okos eszközök vezérlésére, és azok adatainak feldolgozására kifejlesztett programmal készült. A Node-Red egy folyamatábra alapú programozási nyelv, bárki számára megérthető és egyszerűen módosítható a program, így akár a diákok is bővíthetik a már meglévő komplex rendszert, hozzáadhatják saját számítási folyamataikat. A mérések során keletkezett adatokat, a rendszer naplózza, azaz másodpercenként elmenti és azokat dátum szerint rendezi.

Hőmérő: A hőmérőegység egy vagy több digitális hőmérőből és egy vezérlőből áll. A vezérlőben található egy WiFi-s számítógép és egy lítiumos akkumulátor, így nem kell konnektorhoz csatlakoztatni az eszközt a működéshez. A digitális hőmérők vízállóak, és -55°C és 125°C között képesek mérni körülbelül 0,1°C pontossággal, ami szinte minden fizika órán előforduló kísérlethez megfelelő. Például kiválóan alkalmazható kalorimetria mérésekhez, ahol több folyadék hőmérsékletét kell egyszerre mérni.

Erőmérő: Digitális erőmérő, működése megegyezik egy hagyományos rugós erőmérőével, csak sokkal nagyobb pontossággal lehet vele mérni. Méretben nem sokkal nagyobb egy hagyományos előmérőnél. Az eszköz áll egy mérőcellából, ami méri az erőt, és egy elektronikából, ami a mért adatokat WiFi-n keresztül továbbítja az "Agy"-nak. Ez az eszköz szintén akkumulátorról megy. A maximálisan mérhető erő az 50N, ami körülbelül 5kg-nak felel meg. Ezt 0,001N pontossággal képes elvégezni. Kiválóan alkalmas szinte minden rugós erőmérővel végzett kísérletre.

Stopperóra: Digitális stopperóra, mely nagy pontosságú időmérést tesz lehetővé. Egy hagyományos stopperórával ellentétben ezt nem csak

nyomógombbal lehet elindítani és megállítani, hanem közelség és hangerősség érzékelővel is. Ennek az az előnye, hogy a mérés pontosságát nem befolyásolja az ember reakció ideje. A különböző indítási és megállítási lehetőségeket a stopper vezérlőhöz 3,5mm-es jack kábellel lehet csatlakoztatni, és minden egységet lehet megállításra és indításra is használni. A 3,5mm-es jack kábelnek az az előnye, hogy ha megsérül a kábel, akkor az olcsón és egyszerűen cserélhető. A stopperóra vezérlője szintén tartalmaz egy akkumulátort és egy WiFi modult, így a méréseket azonnal küldi a vezérlőegységnek, ami ezeket feldolgozza és közli a felhasználókkal. Az időmérés az eszközön belül történik, így 0,001 másodperces pontosságot is el lehet vele érni a mérésekkel. A mért időnek felső limitje szinte nincs (1,5 hónap).

Áramerősség és feszültségmérő: Ez az eszköz áram és feszültség egyidejű mérését teszi lehetővé. A készülék áramkörbe kötése 3db banándugóval történik, ami lehetővé teszi a gyors bekötést. Egy hagyományos multiméterrel ellentétben itt nem kell méréshatárt kiválasztani, így el lehet kerülni a félreértéseket mérés közben. Az eszköz házában megtalálható egy lítiumos akkumulátor, egy WiFi-s számítógép és egy 16 bites ADC is, ami biztosítja az eredmény pontosságát. Feszültséget 0 és 20 V között képes mérni 0,001 V pontossággal, míg áramerősséget -5 és 5A között 0,01 A pontossággal. Kiválóan alkalmas minden fizika órán végzett kísérlethez.

További mérőegységek:

- PH mérő: Oldatok PH értékének mérése digitálisan. Kiválóan alkalmas olyan kísérletekhez, ahol az oldat pH értéke a folyamat során változik.
- Vércukorszintmérő: Működése a hagyományos mérőéhez hasonló, azaz az ujjbegyet kell megszúrni, és egy csepp vért a tesztcsíkra helyezni.
- Pulzoximeter: Pulzust és véroxigén szintet mér, az ujjunk behelyezésével a műszerbe.
- Bőrellenállás mérő: A bőrellenállás, amelyet galvános bőrreakciónak hívnak (GBR), a bőr elektromos ellenállás változásainak mérésén alapul. A stresszhelyzetek vagy az izgalmi állapot verejtékezésre készíti a bőrt, ezt azonban közvetlenül nem vagyunk képesek érzékelni.
- Testhelyzet figyelő (testhelyzet monitor): Speciális eszköz, amely a tárgyak, akár emberi test aktuális helyzetét tudja mérni, megmutatni. Felhasználása különböző gerincvizsgálatokra és testtartásfejlesztésre ajánlott.



2. kép: Testhelyzet monitor Unity-ben

3. Elért eredmények

- Univerzális, több tudományterülethez, tantárgyhoz (fizika, kémia, biológia, egészségügy) használható Smart mérőrendszer.
- Komplexitásából adódóan is innovatív, mert a jelenlegi piacon nem található ilyen összetett méréseket végző oktatási eszköz.
- A mérőműszereink a piacon kapható mérőműszerekhez képest hasonló pontossággal mérnek, de sokkal kedvezőbb árral bírnak, továbbá nagy részük saját fejlesztésű.
- A méréseket valós időben ki lehet írni akár digitális táblára, valamint azok a felhasználók okoseszközeiken is nyomon követhetők.
- Használata mindenki számára könnyen elsajátítható, a kezelőfelület egyszerűségéből kifolyóan.
- Bővíthető keretrendszer!!! Új mérőegységgel egyszerűen és gyorsan bővíthető a rendszer moduláris felépítése miatt.
- Előállítása iskolák vagy bárki számára megoldható.

A JanusMeter 2019-ben készült, melyhez 2020-as fejlesztésként a testhelyzet figyelőhöz elkészítettünk egy szimulációs programot (2. kép), amely jól érzékelhetővé teszi a test állapotát. A program Unity-ben készült, és a vezérlőegységtől kapja a forgási adatokat. Ezeket az adatokat egy 3D "ember" modellre vetítjük, amely a törzsénél forog, tehát különböző hajlásokat lehet vele szemléltetni. További fejlesztés, hogy most már a kezelői platform angol nyelven is elérhető.

Pókrobot és annak irányítása

FT-PD tech

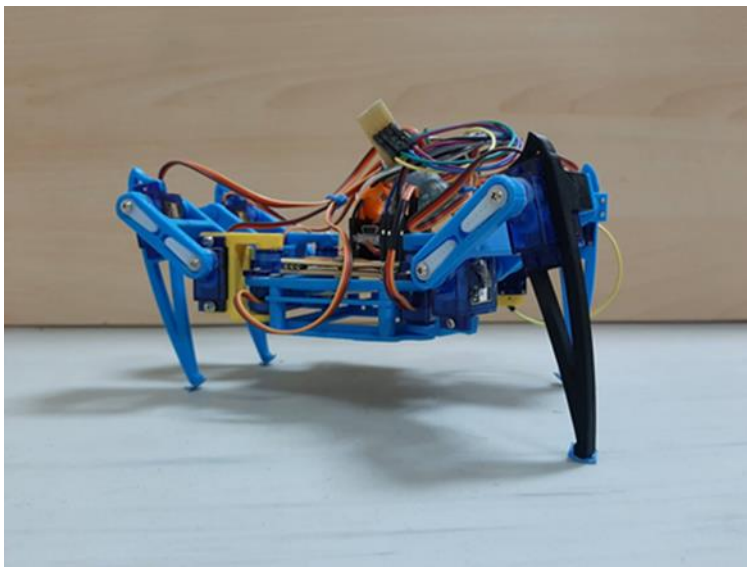
Papp Tamás, Füstös Dániel

Felkészítő tanár: Kőrösi Gábor

Bolyai Tehetséggondozó Gimnázium és Kollégium, Zenta

1. Bevezetés

Motiváció, hogy a robotot megépítsük onnan jött, mivel egyikünk nagyapája a Jugoszláv polgárháborúban vett részt, és a későbbi bombázásokat is látta, és elmesélte, hogy milyen nehéz a törmelék között akármit megtalálni. Reméljük ez a mi projektünk professzionális megoldása megmenthette volna sok ember életét! A robotot azért terveztük így, mert a pókok mozgékonyak, stabilak és egyszerű felépítésűek.

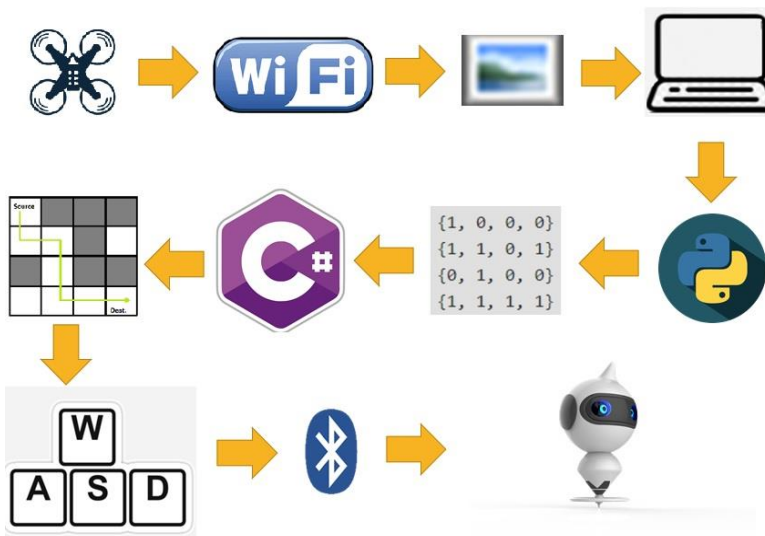


1. ábra: Pókrobotunk

2. Probléma megoldásának menete

Ezt a robotot 3 dimenziós Wanhao D6 duplikátor nyomtatóval nyomtattuk ki PLA anyagból. Azért választottuk a PLA anyagot mivel ipari melléktermék, valamint megújuló nyersanyagokból is előállítható. PLA a természetben relatív gyorsan lebomlik megfelelő körülmények között. A Pókrobot nevet arról kaptam, hogy lábai három fő részből állnak, igaz egy igazi póknak nyolc lába van, és a mi robotunknak csak négy, de kinézete pókformát ölt, ahogyan az 1. ábrán

is látszik. Most már megvan a pókunk, az irányítást egy DJI Tello drón, valamint egy központi számítógép végzi. A drónban jelen van egy 2MP-es kamera, amely erről a terepről fényképet készít és Wi-Fi-n keresztül ezt a jpg fájlt elküldi a rendszer “agyának”, a számítógépnek. Mi Bluetoothon hozzá vagyunk a robotoz csatlakoztatva. A drón elküldi a képet, amit egy Python program fogad, és egy bináris mátrixot generál belőle. Ez a program, ezt a mátrixot, egy C# programba helyezi át, amely gráfelmélet segítségével megkeresi a legrövidebb utat az (1,1) koordinátától az (n, m) koordinátáig. Mikor kigenerálja ezt az utat, a “W”, “A”, “S”, “D”, “Q”, “E” billentyűknek parancsokat ad, és ezeket a Bluetooth modul segítségével elküldi a robotunknak, amely aztán vakon követi a feladatokat. Ezt a folyamatot a 2. ábrán vizualizáljuk.



2. ábra: Vizuális magyarázat a működési elvről

Mivel a mélységérzékelő-kamerához nem tudunk hozzáférni, mert a gyártó nem támogatja azt, hogy bármit is tegyünk vele, így mélységet nem tudunk normálisan mérni, ezért kicsit nehezebb a terepet megjárnia, de megvalósítható. Elhelyezünk egy tükröt a főkamera elé, és egy képzeletbeli háromszöget csinálunk. A háromszög három csúcsa a drón, a robot, és a legutolsó pont, amelyet a főkamera érzékel. Szögfüggvények segítségével meghatározzuk a robot pozícióját a drónhoz képest, valamint a végpont és a drón közötti távolságot. Ezt a kettőt összeadva megkapjuk egy bizonyos egységben ezt az oldalt. Ezt később átalakítjuk póklépésbe, amellyel

megkaptuk, hogy mennyi ideig kell neki ezt megcsinálni, és hogy körülbelül mennyi lépést kell neki megtenni ahhoz, hogy a jelenlegi végpontig elérjen.

3. Elért eredmények

Munkánkat elkezdtük lassan fél esztendeje, több verzióját készítettük el. Egymásra építettük a fejlesztéseket. Az első verzió volt számunkra a legnehezebb, mivel rengeteget kellett kísérletezni a lábak helyes mozgásával és pozicionálásával, és csak sétálni tudott. Második verziónk két infravörös-fényérzékelő segítségével követett egy fekete csíkkal kijelölt útvonalat. Most harmadik verziót állítjuk szemük elé. Itt a megoldhatatlan probléma tárult elénk, amit még az internet sem tudott megválaszolni. Szégyenlem bevallani, hogy csak a fordítóprogram nem megfelelő verzió volt, Python 2.7 helyett Python 3.5-ös volt.

Mivel a projekt korábban előadható állapotba került, mint gondoltuk, ezért úgy döntöttük, versenyeztetjük a Neumann informatika versenyen. Tovább jutottunk a nemzetközi döntőre, Szekszárdra.

Házi készítésű CNC gép

Informér

Piri Pál, Zabos Péter

Felkészítő tanár: Esztelecki Péter

Bolyai Tehetséggondozó Gimnázium és Kollégium, Zenta

1. Bevezetés

A projekt címe magába foglalja szinte teljesen, miről fog szólni ez a dolgozat: saját kezűleg elkészített CNC gép.

A projekt elkészítése több mint egy évet vett igénybe. A kezdetiek voltak a legnehezebbek, a megtervezés és az elmélet alkalmazása gyakorlatban. Két nagyobb részre bonthatjuk a folyamatot: első a megtervezés és megépítés, a második pedig a beprogramozás valamint az elektronikai rész elkészítése. A szoftver fejlesztése sokkal hamarabb megindult, mint még ahogy akármelyik alkatrészrel rendelkezünk volna. Saját módszerekkel próbálkoztunk megoldani a problémákat. Kijelenthetem, hogy a munka folyamata során különböző problémával találkoztunk, amikre saját magunknak kellett megoldást találnunk.

A projektmunkánk valós életben is használatos, mivel különböző dekoratív rajzokat lehetséges vele kimarni, ami díszként szolgálhat, vagy akár alkatrész gyártásra is használható. Több szempontból is univerzális, mivel a marógép éle is cserélhető különböző méretűekre. Valamint akár a módszer is. Lehet vele marni, gravírozni,...

2. Probléma megoldásának menete

Az első fő problémát a szerkezet elkészítése jelentette. Saját kezűleg nekifogtunk, de mivel nem voltak megfelelő eszközeink és tudásunk ehhez e folyamathoz, ezért egy esztergályos mester segítségét kértük. 3 tengellyel rendelkezik. Az alábbi ábrán látható a fém keret a marógéppel. A következő lépés az elektronika beépítése és a szoftver megprogramozása volt.



1. ábra: A keret

3. Szerkezet

A tengelyek mozgatása léptető motorokkal történik. Jóval a fém keret elkészülése előtt már rendelkezünk a motorokkal, mivel előre tudtuk, hogy ilyenekre lesz szükségünk. 3 darab E Series Nema 23 Bipolar 1./deg 3.0 Nm(425oz.in) 4.2A 57x57x13mm 4 Wires [2]. Ezek a motorok 2 fázisúak, 1.8 fokot forognak lépésenként, 3 Nm forgatónyomatékúak, valamint 4.2A áramerősséggel működnek. 57 mm szélességűek és 57mm magasak, hosszuk pedig 113 mm. 4 darab dróttal lehet hozzákötni őket a driverhez. Legkisebb egység, amit lehet vele mozdtítani az 1/10-ed milliméter.

Vezérlőegység alatt a gép elektronikai részét kell érteni. Itt történik az adat fogadása a számítógéptől és ennek a továbbítása a motorokhoz, valamint az áram transzformálása. Mindezt egy saját kezűleg készített préselt falemezből álló dobozban helyeztük el. Részei az Arduino UNO, a léptető motor driverok, a tápegység, kapcsolók valamint a drótok, amivel össze vannak kötve.

A léptető motorok fontos kisegítő eszköze a meghajtójuk. Az általunk használt Nema 23-as léptető motorokhoz a TB6600 jelölésű meghajtó (driver) tartozik. Motoronként szükséges egy darab. Ez a készülék látja el a motorokat árammal. Megfelelő feszültség küldésével a motor forogni kezd a megadott irányba (pozitív vagy negatív). 12 darab bemenet/kimenet található rajta: az első 2 a +12V áram és a földelés helye, a következő 4 a motorok kebeljeinek szolgál, majd a következő 6 az Arduinoval való kapcsolat (ebből 3 darab földelés szintén).



2. ábra: A CNC gép

Árammal van ellátva: az Arduino, a léptető motor meghajtók, valamint itt található áramelosztó a marógép és porszívó számára is. Az Arduino állandó kapcsolatban áll a számítógéppel, ezért nem szükséges számára külön 12V-os tápegységet biztosítani. A vezérlőegységbe érkezik áramkábelen keresztül 220V, ez elágazik 2 részre: egy áramelosztóra (biztosítékkal), és egy kapcsolón keresztül az ATX tápegységre. Az áramelosztón 3 hely található, 1 a marógépnek, 1 a porszívónak. Az ATX tápegység egy asztali számítógépből kiserelt darab. A tápegységből használtunk 3 darab földelés (fekete) drótot és 2 darab 12V-os kábelt, a többire nem volt szükség. A biztosítókkal áramtalanítani lehet az áramelosztót, a kapcsolóval pedig a tápegységet.

4. Működési mechanizmus

Az információ árama a számítógéptől a motorokig. A Vezérlőegység alcímben felsorolt eszközök: Arduino UNO, TB6600 léptető motor meghajtó szerkezeti elemek és ezen kívül 2 darab nyílt forráskódú programból áll (Universal G-code Sender, GRBL).

A folyamat a számítógépnél kezdődik és a motoroknál ér véget. A számítógépen fut az Universal G-code Sender nevű program, ezzel beolvasunk egy g-code file-t. A program az utasításokat USB kábelon keresztül küldi az Arduino UNO-nak (az információ ekkor kerül a vezérlőegységbe). Az Arduinon található egy GRBL nevezetű szoftver. Ez képes egy g-code értelmezésére és tovább küldi a parancssort a léptető motor meghajtói felé egyszerű kábeleken keresztül a soros pinjeiből (mindegyik pin megfelelő motort mozgat, ez standard beépített a GRBL programkódjába). A meghajtókat bekötési diagramm szerint kötöttük be, melyik pin melyik motorhoz kapcsolódik. Az utolsó lépés pedig az, amikor a meghajtók a fogadott információ alapján megfelelő feszültségű árammal látják el a motorokat, és ez által a megadott irányba meghatározott számú fordulatot tesznek meg. A folyamat végeredménye, hogy a marógép feje az általunk kívánt pozícióba kerül és kimarja a fa anyagból a meghatározott mintát.

Az Universel G-code Sender nevű program bizonyult a legmegbízhatóbbnak és használhatónak a munkánk elkészítése során. Ez a program teljes g-code funkcionalitású, amelyet fel lehet használni fejlett CNC vezérlőknél, mint például a GRBL és a TinyG. Az Universal G-code Sender egy önálló Java alkalmazás, amely magába foglalja az összes külső függőséget, azaz ha van Java Runtime Environment (Java futtatási környezet) telepítve, akkor a program biztosítja a többi. Tartalmaz 3D-s g-code vizualizálót, ami valós időben mutatja a marógép pozícióját, valamint különböző színnel jeleníti meg már a megtett és a még hátralévő utat. A marás elkezdéséhez a file üzemmódban beolvasunk egy g-code file-t. Ennek a kiterjesztése lehet .gcode, .ngc. Itt tekinthetjük meg a vizuális ábrázolását a marásnak.

GRBL egy nyílt forráskódú program, amely az Arduinon fut. Ez biztosítja a kommunikációt a szoftver és a motorok között. Tulajdonképpen ez egy előre elkészített könyvtár, ami képes értelmezni a g-code file utasításait. Egy külön file-ban találhatóak meg a motorokra vonatkozó beállításai. 0-tól 132-ig terjedő számozású beállításokkal rendelkezik. Mindegyik másért felelős, és \$ jelöléssel van ellátva. A legnagyobb része alapbeállítás, melyet nem szükséges megváltoztatni, de van néhány, amelynek beállítása elengedhetetlen.

A \$2 és \$3 beállítás arra szolgál, hogy a motorok mozgásának irányát beállítsuk anélkül, hogy a kábeljeit fizikailag felcserélnénk.

A \$100,\$101 és \$102 (X,Y és Z koordináta) utasításokkal beállíthatóak a motorok lépés száma, egy megadott távolság esetén. A következő értékeket is szükségyszerűen változtatni kell: \$110, \$111 és \$112. (X,Y és Z koordináta). Ezek a motorok maximális sebességét határozzák meg. Hasonló módon állítható a \$120,\$121,\$122 (X,Y és Z koordináta) változókkal, amik a tengelyek gyorsulásáért felelős, azonban ez nem annyira befolyásolja a marást.

Annak ellenére, hogy a g-code file-ban a kép méretei nem lépik túl a gépnek a határait, ajánlatos beállítani maximális mozgási korlátokat. Azaz meghatározzuk, melyik a legszélső pont, ameddig elmehet a gép. A \$130,\$131,\$132 (X,Y és Z koordináta) értékekkel tehetjük meg ezt.

G-code utasítás fájl, az Inkscape nevű ingyenes rajzprogramban készíthetünk. Az kép éleinek beazonosításával útvonalat tervez, amiket utasításokká alakít át.

5. Kiegészítő ötletek

A gépet elláttuk végkapcsoló rendszerrel, ami hiba esetén, amikor a gép a tengely legszélső pontjára jut, akkor áramtalanítja a gépet. Ilyen módon nem sérülhet a tengely és a motorok sem éghetnek le.

A vezérlőegységet leválasztható ATX kapcsolóval oldottuk meg. Tehát külön lehet szállítani a szerkezettől függetlenül.



3. ábra: Kimart macska fej

Vezeték nélküli kesztyű

Irinyi

Sonkoly Gergő, Hentes Kornél

Felkészítő tanár: Pere Csaba

Irinyi János Református Oktatási Központ, Kazincbarcika

1. Bevezetés

Műszaki informatika kategóriában pályázunk. Habár jelenleg Villamosipar- és elektronika ágazati képzésben tanulunk, elektronikai technikus szakmában. Tizenkettedikes diákként már folyamatosan két éve ismerkedünk az irányítástechnikában alkalmazott speciális eszközökkel (nyúlásmérők, szenzorok), amelyek megmozgatják a fantáziánkat. Emellett szabadidőnkben egyszerű mikrovezérlők, Arduino programozásával foglalkozunk, és tapasztalatainkat rendszeresen megbeszéljük egymással. A gyakorlati órákon lehetőségünk nyílik arra, hogy néhány ötletünket a valóságban is kipróbáljuk. Már van néhány közös alkotásunk és ez a verseny remek alkalom arra, hogy szélesebb körben is bemutatthassuk találmányunkat, ami a Helping-hand (segítő kéz) fantázianevet kapta.

2. Probléma megoldásának menete

Elképzelésünk alapja az ember által minden nap igénybe vett testrészére, a kézre fókuszál. Találmányunk egy nagyon egyszerű ötleten alapszik, ami tulajdonképpen egy speciális kesztyű. Ezzel a kesztyűvel az a célunk, hogy a hétköznapi ember mindennapjait könnyebbé tegyük, segítsük a mozgáskoordinációs betegségben szenvedőket, illetve egy-egy nagyobb volumenű prezentációt látványosabbá tegyünk az előadók és oktatók számára. Úgy gondoljuk, hogy ebben a kesztyűben több lehetőség rejlik és szinte csak a képzelet szabna határt, hogy mire alkalmazhatnánk még. Azok az emberek, akik valamilyen betegségben élnek (pl. mozgássérültek, tartósan ágyhoz kötöttek), ezért mindennapjaik nehezebbek az átlagos emberénél. A mi eszközünk, ami egy elektronikus kesztyű ezeknek az embereknek a mindennapi nehézségeit próbálja egyszerűbbé tenni. Itt szeretném kiemelni, hogy olyan mozgáskoordinációs betegek a kesztyű célcsoportjai, akik a kezüket képesek használni. Úgy véljük, hogy egy ilyen kesztyű nagy segítség lehetne ezeknek az embereknek, hiszen könnyen hordozható, kényelmes és vezeték nélkül tud kommunikálni bármilyen Bluetoothos eszközzel egy előzetes programozás után, amit természetesen mi programoznánk fel. Így akár egy fekvő beteg is a keze mozgásával tudja otthonában irányítani a

világítást vagy szórakozását elősegítő okos TV-jét, és még számtalan más elektronikai eszközt, ami rendelkezik USB foglalatral vagy Bluetooth vezeték nélküli csatlakozási lehetőséggel. Ezen felül, azok is használhatják, akik nagyobb kényelemre vágnak okos otthonaikban.

A kesztyű olyan elektronikai alkatrészeket tartalmaz, mint például a nyúlásmérő, giroszkóp, Bluetooth vagy mikrokontroller. A nyúlásmérők segítségével tudunk egyes ujjbehajlításokhoz különböző funkciókat hozzárendelni. A giroszkóppal viszont több kézállásban is lehetőség nyílik további ujjbehajlítások hozzárendelése. A kéz mozgását követő állapotokat és ujjbehajlításokat a használó képes lenne az egyéni kényelme szerint beállítani és használni. Például, ha az illető a nyújtott keze állásában a mutató ujját behajlítja, felkapcsol a lámpa vagy bekapcsol a TV. A kesztyű Bluetooth segítségével kommunikálna a műszaki berendezésekkel.

Az általunk elkészített elektronikus kesztyű az alábbi részekből áll:

- normál textil anyagú kesztyű,
- 3D nyomtatóval nyomtatott tárolódoboz,
- 2db Bluetooth(Master, Slave),
- Akkumulátor (1200 mA-es),
- Arduino (mikrovezérlő áramkör),
- DC-DC konverter,
- 5 db nyúlásmérő ellenállás.

A kesztyű működési technológiája a nyúlásmérőkön alapszik. Ujjunk behajlításával a nyúlásmérő alap Ohm-os ellenállás értéke megváltozik, amit egy Arduino nevezetű mikrovezérlő áramkör folyamatosan monitoroz. Ez az egység úgy lett programozva, hogy egy bizonyos mértékű behajlítás során jelet küldjön az irányítandó eszköz számára. Például a lámpát a kézfejem ökolbe szorításával bekapcsolom, hiszen a nyúlásmérők ellenállás értéke megváltozik, és ezt érzékeli az Arduino. Ennek hatására jelet küld a vezérelni kívántegységnek, ami értelmezi azt. Az Arduino felprogramozása a mi feladatunk volt, ami körülbelül mintegy 200 munkaórát vett igénybe.

A kesztyű egyik fő része az akkumulátor teszi lehetővé a vezeték nélküli hordozhatóságát, ami jelenleg egy 1200mAh-ás 3,7V-os akkumulátor, amivel az eszközünk tartós használat mellett olyan 15 óráig képes működni. Ez az akkumulátor a 1. ábrán látható. Ahhoz, hogy az Arduinonk is megfelelően működjön, szükségünk van egy DC-DC konverterre (átalakító) ami 3,7V feszültségből 7V-ot állít elő. A használat közben felmerülő hibákat, összeakadásokat egy egyszerű programsorbéli változtatással orvosolni tudjuk. Így újra működésre bírjuk az eszközünket. Új funkció beillesztése esetén a

kesztyűt előzetes próbavizsgálatoknak vetjük alá, hogy megfelelően működik az új felprogramozott funkció. A kesztyű további funkciókkal egészül ki. Ilyen a *személyre* szabható hangulatvilágítás, melyet működtethetünk egy 5V-os tápegységről vagy a számítógépünkről. Lényege, hogy különböző okos otthonaink teljes világítás tartományát bárhol tudjuk irányítani, úgy ahogyan azt mi akarjuk. Alkalmazása lehet okos otthon, ahogy azt említettük vagy kisebb üzemszerek megvilágítása.



1. ábra: Az általunk készített eszköz, a vezeték nélküli kesztyű

A harmadik általunk kitalált funkció esetében egy konnektort alakítottunk át, melybe bele építettünk egy bluetooth érzékelőt. A munka ezen szakaszában az ember számára veszélyes hálózati feszültséggel (AC 230V) dolgoztunk, ezért mindent úgy alakítottunk ki, hogy biztonságtechnikai szempontból megfeleljen az előírásoknak. Az átalakítani kívánt konnektor kiválasztásánál fontos volt a gyermekvédelem is, ezért egy biztonsági konnektort vettünk. Az átalakítás bonyolult volt, mert állandóan gondot okozott az áramkörök fizikai terjedelme, de végül is sikerült mindennek megtalálni a helyét. Elsőnek a vezérlő egység helyét alakítottuk ki, majd ezt követte a bluetooth utána pedig egy 4-es relé modul kapott a dugaszolóaljzatban helyet. Mivel ezeknek az eszközöknek a működéséhez törpe feszültségre van szükség, még elhelyeztünk egy transzformátort is, ami a 230V-os váltakozó feszültséget 5V-os egyenfeszültségre csökkenti. Így egy multi-funkciós konnektort készítettünk, amit széles körben alkalmazhatunk. Ez a konnektor a 2. ábrán látható. Aminek szerepe annyival bővül ki, hogy a rákapcsolt eszközöket távolról is vezérelhetjük és még a kényelmes székünkben se kell felkelnünk.



2. ábra: Multi-funkciós átalakított dugaszolóaljzat

3. Elért eredmények

Eddig 3 különálló eszközt készítettünk hozzá, melyek tökéletesen bemutatják a kesztyű felhasználási funkcióit. Az USB-vel rendelkező szerkezet, amely képes elektronikai termékeinket irányítani, ha azok rendelkeznek képmegjelenítési és USB csatlakoztatási funkcióval. Felhasználását különféle okos otthonokban javasoljuk, valamint valamilyen oktatási céllal létrejött épületekben. Kesztyűnk nagy sikerrel tudja kezelni a számítógépes kurzort. Képes a kattintásra, görgetésre, kijelölésre. Előre beprogramozott szavakat, mondatokat képes beírni a böngészőbe. Tudja kezelni a PowerPoint diavetítési funkcióját, hangulatvilágítás színválogatását. A PowerPoint diavetítési funkciót egy bemutató tanórán keresztül ismertettük meg a tanárokkal még tavaly decemberben, akik nagyon pozitívan fogadták a vezeték nélküli kesztyű ilyen fajta felhasználhatóságát. PowerPoint diavetítési funkcióra, de az egész aktív tábla kurzorának kezelésére jól működött. A második funkció alkalmazásához az osztálytermünkben építettünk ki LED-es hangulatvilágítást, amit a kesztyűvel vezérletünk. Ennek a világítástechnikai ötletnek a megvalósítása nagyon egyszerű dolog volt számunkra miután elkészítettük a kesztyűt, és a működése is teljesen probléma mentes, valamint nagyon hatékony lett. A harmadik funkció irányát már a kesztyű megalkotásakor kijelöltük és ehhez sikerült találnunk egy személyt, az egyik osztálytársunk nagypapáját, aki térdműtéten esett át és hajlandó volt kipróbálni a kesztyűt. A lényeges szempont nemcsak az volt, hogy a mindennapokban hogyan tudja használni, hanem az is, hogy milyen gyorsan tudja elsajátítani a működtetését és mennyi idő múlva alkalmazza önállóan. Nagy örömünkre a kesztyű ezt a próbát is kiállta, és visszaigazolta az eddigi elmélkedésünket. A multi-funkciós konnektorról külön-külön eszközöket működtettünk a fekvő beteg páciensünkkel. Először az éjjeli lámpát üzemeltük be a dugaszolóaljzatba. Nagyon jól jött éjszaka a betegünknek, aki nem tudott aludni és fekvő helyzetbe felkapcsolta a lámpát, hogy a maga mellé készített újságot olvashassa. Aztán napközben nagyon nagy szolgálatot tett a rádió beüzemelése a konnektorba, mert betegünk elmondta, mennyire szeret rádiót hallgatni és ez számára könnyen megoldható. Ezen kívül még egy eszközt, a ventilátort is kipróbáltuk, mert betegünk arra panaszkodott, hogy régebben a nagy nyári melegben milyen jól lett volna ki-be kapcsolgatni, amikor szintén egy műtétje az ágyhoz kötötte. Összegezve arra a következtetésre jutottunk, hogy ennél a kísérletnél az általunk kitalált eszköz egy nem várt eredményt is produkált, javította a beteg közérzetét növelve ezzel a gyorsabb gyógyulási esélyeit.

Fröccsgép

Szixi

Viktor Kristóf, Petrovics Dávid

Felkészítő tanár: Hegedűs Tamás

*Tatabányai SZC Széchenyi István Közgazdasági és Informatikai
Szakgimnáziuma, Komárom*

1. Bevezetés

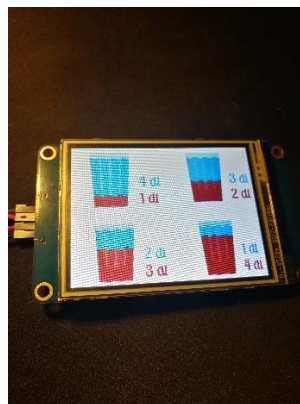
Az alap ötlet nyáron született meg egy grillezés alkalmával, hogy milyen jó lenne építeni egy fröccsgépet. Cél a gép megépítésekor az volt, hogy egy minél kevesebb alkatrészből álló szerkezetet hozzunk létre, ami nem csak fröccsöt tud keverni, de képes hidegen tartani is azt. Az ötletet hallva borászok és kocsmárosok is jelezték, hogy örülnének egy ilyen gépnek.

2. Probléma megoldásának menete

A projekt fő szempontja egy olcsó és egyszerű eszköz megépítése volt, amit hosszas tervezés és kutatás előzött meg.

2.1 Kijelző

A kijelzőnek fontos szerepe van, mivel a használó ezen tudja kiválasztani az italt. A választás egy Nextion érintőképernyős LCD kijelzőre esett, mert a panel grafikus tervező felülete egyszerű és gyors tervezést tett lehetővé, illetve a későbbiekben is megkönnyíti az esetleges javításokat vagy változtatásokat. A kijelző az 1. ábrán látható.



1. ábra: Nextion kijelző

2.2 Relé és szivattyú

A folyadékok adagolását kettő akvárium szivattyú végzi, ami egy 2-t csatornás relébe van kötve, amit a mikrokontroller vezérel. A relé és a szivattyúk a 2. ábrán láthatóak.



2. ábra: Relé és szivattyúk

2.3 Ultrahang szenzor

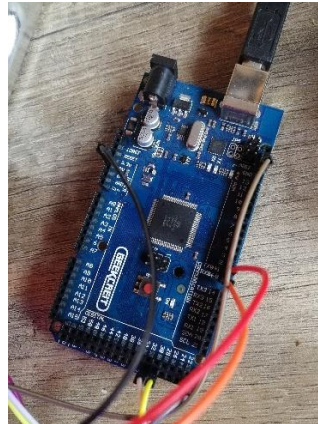
Az esetleges hibák elkerülése végett beépítésre került egy HC-SR04 típusú ultrahang modul is, amely a pohár felett kapott helyet, mérve annak folyadék szintjét, megakadályozva így a túlfolyást. Az ultrahang modulról kép a 3. ábrán látható.



3. ábra: Ultrahang szenzor

2.4 Mikrokontroller

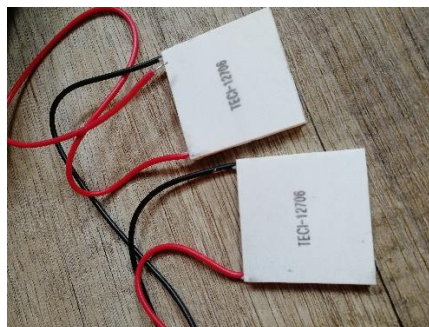
A kiválasztáskor több szempontot is figyelembe kellett venni, elsősorban, hogy elég számú digitális, illetve analóg be- és kimenettel rendelkezzen az eszközökkel való megfelelő kommunikáció miatt, illetve a későbbiekben a továbbfejlesztésre is alkalmas legyen. Választásunk pedig egy Arduino Mega lett, mivel a feltételeknek tökéletes megfelel mindemelett ára is kiváló. Az Arduino a 4. ábrán látható.



4. ábra: Arduino Mega

2.5 Hűtés

A legnagyobb fejtörést a hűtés okozta, hosszas keresés után úgy döntöttünk, hogy kettő Peltier elemmel oldjuk meg ezt a problémát. Az elemekről kép az 5. ábrán látható.



5. ábra: Peltier elem

3. Elért eredmények

Ez a szerkezet volt az első projektünk, amit valaha is csináltunk, mindent sikerült kiviteleznünk. Végeredményben tudtunk építeni egy fröccsgépet, ami képes hűteni is. A feladat elvégzése alatt pedig tudásunk is fejlődött. A továbbiakban pedig azon gondolkozunk, hogy hogy lehetne fejleszteni a szerkezetet.

Házilag készített autószimulátor versenyjátékokhoz

Irány a RALLYE!

Csizmarik Dániel

Felkészítő tanár: Bakti András Tamás

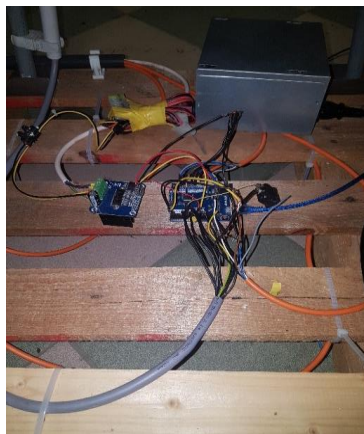
*Nyíregyházi Szakképzési Centrum Széchenyi István Közgazdasági és
Informatikai Szakgimnáziuma és Kollégiuma, Nyíregyháza*

1. Bevezetés

Mint minden fiatal, én is szeretek számítógépes játékokkal játszani, elsősorban autóversenyes játékokkal. Az elvárásaimnak megfelelő kormány nagyon drága lenne, ezért utánanéztem az interneten, hátha van egy olcsóbb alternatíva. Ekkor találtam egy leírást és hozzá egy videót, ami bemutat egy házilag elkészített, elkészíthető kormány, váltó pedálok kombinációt. Mivel szoftverfejlesztőnek tanulok, és az édesapám révén érdekel az elektronika, valamint járok a Széchenyiben a Robot TÉP szakkörre, felmerült bennem, hogy miért ne építhetnék én is egy ilyet. Szakmai kihívásnak tekintettem, valamint jó lépcsőfoknak a további céljaim megvalósításának elérése érdekében.

Tehát az eszköz, aminek az építésébe belefogtam egy kormányból, egy H váltóból, és 3 pedálból áll. Később karosszériát is kaphat, és lehet belőle egy autószimulátor.

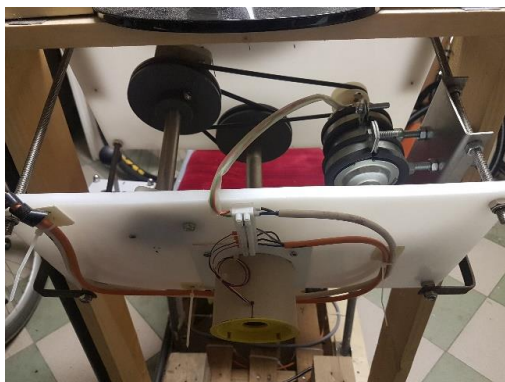
Az alkatrészek nagy része az apa munkahelyén meglévő hátramaradt anyagok, melyek segítségével sikerült a főbb elemeket összeépíteni. A



1. ábra: Vezérlés

vezérlésről egy Arduino Leonardo (1. ábra) gondoskodik. Probléma megoldásának menete

A kormány szerkezet (2. ábra) egy autóbontóból szerzett „kispolszki” kormányból, egy 24V-os DC motorból, és áttételekből áll, amelyeket egy esztergályos készített. A DC motor a kormány visszatekeréséért felel, az áttételek pedig a visszatekerés sebességét szabályozzák. A főtengelyen található továbbá egy encoder, amely egy selejtes nyomtatóból származik. Az az encoder érzékeli az elfordulást.



2. ábra Kormány szerkezet

A pedál rész (3. ábra) 3 db pedálból áll, amelyek házilag készített teleszkópokon vannak, amelyben a nagybátyám segítségét kértem. Különlegessége hogy állítható az útja a teleszkópnak és ezáltal az erősségét is lehet befolyásolni. A benyomás mértékét potméterekkel méri, ezen értéket dolgozza fel a vezérlés.



3. ábra: Pedálok

A váltó (4. ábra) egy 7 sebességes H váltó, amelyben található 7 db végálláskapcsoló. Ezek érzékelik, hogy hányadik sebességben van a váltó (ha egyik potmétert sem nyomja be, akkor üresben van). Továbbá két oldalról egy-egy húzórugó tartja középen a rudat.



4. ábra: Váltó

Ez az egész szerkezet pedig egy raklapra van rászerveelve, amit a kertben tartogattunk. Az ülés egy használaton kívüli szék, ami a garázsban porosodott. A szerkezet építésénél igyekeztem minimalizálni a költségeket, ezért vannak benne újrahasznosított szerkezeti elemek. A megvalósítás során sokat tanultam programozás, és az elektronika területén. Sokat köszönhetek édesapámnak, aki mindvégig segített, támogatott a projekt megvalósítása során a gyakorlati dolgok kivitelezésében.

2. Elért eredmények

Mivel a megvalósítás eredménye nagy méretű, ezért a versenyre nem fogom tudni elszállítani, de egy videófilmmel be tudom majd mutatni a működését.

AGV robot készítése

Robot TÉP_2020

Beleznai József, Pap Dániel Péter, Győző Róbert

Felkészítő tanár: Bakti András Tamás

*Nyíregyházi SzC. Széchenyi István Közgazdasági, Informatikai Szakgimnáziuma
és Nyíregyháza*

1. Bevezetés

A versenyre a Robot TÉP szakkört vezető tanárunk inspirálására jelentkeztünk. Mivel szoftverfejlesztőnek tanulunk, és érdekel bennünket a robotika, ezért elkezdtünk járni a szakkörre. Itt arduino és raspberry vezérlőkkel felszerelt robotokat tervezünk, építünk, és programozunk. (Innen a TÉP a nevünkben.) Eleinte csak 3D nyomtatóval készített és lézervágóval vágott alkatrészekből épített robotokkal foglalkoztunk, amelyek érdekesek voltak, de valódi gyakorlati hasznuk a tapasztalatszerzésen kívül nem volt.

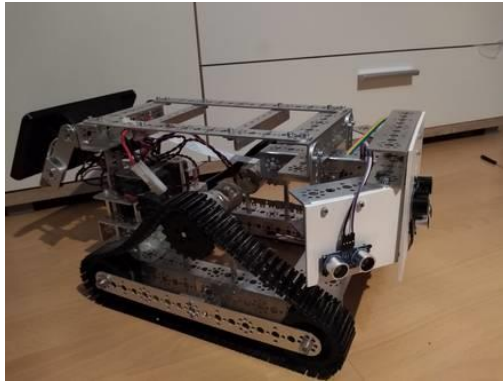
A tanárunk révén azonban kaptunk egy felkérést a Jász-Plasztik Kft. tulajdonosától, hogy építsünk első körben egy AGV (Automatic Guided Vehicle) robotot, amely képes a gyár területén A pontból B pontba eljuttatni az alkatrészeket, összehordani a részegységeket az összeszerelő üzembe, stb. A tulajdonos és a nyíregyházi gyáregység gyárigazgatójának elmondása alapján komoly problémát jelent, hogy miközben a fröccsöntő gépeket automatizálják, a leszedő egységek automatizálhatók, emellett a szállítás automatizálására is nagy szükség van. Az emberek manapság nem szívesen mennek el anyagmozgató munkakörbe, ezért kellenek az AGV és egyéb robotok, megoldva ezzel az anyagmozgató munkások hiányából adódó problémákat. És mivel kapcsolódik az érdeklődési körünkhöz egy ilyen projekt megvalósítása, nagy örömmel vetettük bele magunkat a munkába.

Bár már számunkra is jól ismert korlátai vannak egy AGV-nek, a projekt során szerzett tapasztalatok hasznosnak bizonyultak számunkra. Mivel egy AGV robot nem képes alkalmazkodni a környezet változásaihoz, például ha megváltozik az épületen belül a gyártósor, máshová kell szállítani a dolgokat, egy AGV robot esetén ez csak újraprogramozás segítségével oldható meg. Ez jelentős időkiesés, ezáltal a termelékenység rovására megy. Ezért távolabbi cél egy AMR (Autonomous Mobile Robot) megépítése, ami lézeres távolság érzékeléssel, térképezéssel működik. Ezen cél eléréséhez egy lépcsőfok volt ez a projekt.

2. Probléma megoldásának menete

A robot alvázát a tavalyi évben a National Instruments Kft.-től ajándékként kapott Tetrix robotépítő készlet fém építő elemei adják. Későbbiekben ki kell dolgozni egy olyan szerkezetet, amiből több is készíthető, és ami le van tesztelve méretek, teherbírás, praktikusság tekintetében. Az 1., ábrán a prototípus fém szerkezete látható.

Az irányítás egy Raspberry Pi3 segítségével van megoldva, melyhez egy 7"-os érintőképernyő csatlakozik, melyről indítható a robot programja. A



1. ábra: A robot fém alváza

meghajtást a Tetrix DC motorok végzik, egy L298N motorvezérlővel és lánctalpak segítségével. A 2. ábrán a robot hátulja látható, ahol van a kezelőfelület, ahonnan elindítható a program, melyet Python-ban írtunk.



2. ábra: A robot kezelőfelületének érintőképernyője

A három darab ultrahangos távolságszenzorral (HSR-04) „tájékozódik” a robot, hogy nincs-e akadály az útjában. Akadály esetén megáll, elfordul és

próbálja kikerülni az akadályt. A Raspberry Pi kamera segítségével egy okos telefonról nyomon követhető hol jár éppen a robot, és szükség esetén a kezelő beavatkozhat. A 3. ábrán az éjjellátó Rpi kamera és az ultrahangos távolságérzékelők láthatóak.



3. ábra: A kamera és a távolságszenzorok

A teherszállítás kipróbálásához egy műanyag ládát használtunk, ami természetesen lecserélhető a szükségletnek megfelelően. A 4. ábrán a szállító láda látható, amely a szállítandó anyag tulajdonságainak figyelembe vételével szintén további tervezést igényelhet. Mint korábban említettük ezen projekt nem a műszaki tervezésről szólt, hanem a robot programozása volt az elsődleges feladat. Ezért a szerkezeti elemek további finomításra szorulnak.



4. ábra: Az AGV robot szállítólárával

3. Elért eredmények

Rájöttünk, hogy hasznosabb lehet a kamera mozgathatóságának megoldása szervó motorok segítségével, hogy elakadás esetén a távolról körbe

tudjunk nézni a probléma feltárása, és megoldási javaslat kidolgozása érdekében.

Mivel a robotika egyre elterjedtebb, a jelentős munkaerőhiány miatt már kisebb cégek is az automatizálás irányába indultak, egyre nagyobb szükség lesz hasonló, önjáró robotokra. Az építés során szerzett tapasztalatokat pedig remélhetőleg a későbbiekben mi is tudjuk majd kamatoztatni további tanulmányaink során, és majd leendő munkahelyünkön. Jelentős előrelépés számunkra, hogy a Robotika Kabinet és az Iskola falain kívül, valós, ipari környezetben is van lehetőségünk kipróbálni, tesztelni, továbbfejleszteni a projektünket.

Ezen kívül annak kapcsán, hogy több bemutatón és rendezvényen részt vettünk korábban készített robotokkal, sikerült jó kapcsolatokat kiépíteni. Kezd beérni az eddigi munkánk gyümölcse, kezd terjedni az iskolánk és a szakkör jó híre. Például megkeresett minket a tiszalöki Hot & Cold Therm Kft. is, ahol szintén bekapcsolódhatunk robotikai fejlesztésekbe. Itt szintén valós problémák megoldásában szerezhethetünk tapasztalatot az ipari robotkarok programozása területén. Ezen kívül lézeres biztonsági kapuk építésére kaptunk felkérést, amelyek leállítják a robotkart, ha valami, vagy valaki a robotkar útjába kerül.

Összességében biztosak vagyunk benne, hogy a sok-sok idő, amit ráfordítottunk, az energia, amit befektettünk, mindenképpen meg fog térülni hosszútávon.

Rusty

Erdélyi Innovációs Műhely

Kovács Nóra-Anna

Felkészítő tanárok: Kovács Róbert-Jenő; Gyenge L. Ervin; Tamás Levente

János Zsigmond Unitárius Kollégium, Kolozsvár

1. Technológiai tükör egy tisztább környezetért

A technológiai újítások egyik fő feladatköre, hogy megkönnyítsék az emberi munkavégzést, hozzájárulva civilizációnk fejlődéséhez. Az első ipari forradalomtól napjainkig, a létszámában egyre csak növekvő emberiség óriási hitelt kölcsönzött környezetétől, hogy a több milliárd főre emelkedett népesség hétköznapi életvitelét, egzisztenciáját fent tudjuk tartani. Nem csoda tehát, hogy napjaink innovációi olyan célokat tűznek ki, amelyek a megváltozott környezeti problémákra szeretnének megoldásokat biztosítani.

A 20. század folyamán a gyorsan növekvő népesség egyre nagyobb része költözött a városokba, ezzel egy időben a szemét mennyisége a tízszeresére nőtt. A cigarettacsikkektől a sörös dobozokon át a műanyagpalackig kéthavonta dobunk ki a testsúlyunknak megfelelő szemét mennyiséget. A hulladék halom nagyságáról kevés fogalmunk van, pedig világszerte a szemét mennyisége nő a leggyorsabban az összes környezetszennyező anyag közül, beleszámítva az üvegházhatású gázokat is. A szemétermelődés leginkább az életminőség alakulástól és a városi lakosság számától függ, hiszen fogyasztói társadalmunk egyre többet és többet vásárol, több csomagolást használ, ezért jóval több újra nem hasznosítható vagy lassan lebomló hulladék marad utánunk.

A tudatos hulladékgazdálkodás egyik fontos állomása lenne, ha közvetlen környezetünk, köztereink tisztántartását olyan módon képesek lennék biztosítani, amely során a közterület-fenntartás nem igényel nagy volumenű emberi munkavégzést. Ugyanakkor az iparág robotizációja csak a probléma kezelését jelentené. A tényleges probléma felszámolása, a városfejlesztési megoldásokon túl, az emberi helytelen szeméthasználat minimalizálásával valósítható meg, továbbá hosszútávon ennek óriási a társadalmi, környezeti és morális jelentősége lenne. A környezetvédelmi problémák megoldásában gyakran nem számolunk az azt kiváltó emberi tényezővel, és kiváltképp nem helyeződik hangsúly az emberek irányába történő negatív visszacsatolásra. Munkám során igyekeztem egy olyan technológiai tükör koncepcióját megalkotni, amellyel reményeim szerint, képes leszek az emberek figyelmét

felhívni a közterületeken elszórt hulladékokra, illetve azok eltakarítására. Úgy vélem, hogy egy robot által mutatott társadalmi tükör képes megszólítani a mindennapokban a városi közösségek jelentős részét, így elősegítve a közterületek hulladékmentességét. Továbbá abban bízom, hogy Rusty a fiatalabb generációk részére egy környezetvédelmi szempontból edukatív eszköz szerepét is betöltheti.

2. Probléma megoldásának menete

Kutatómunkám célja, hogy tudatos szemtelésre ösztönözzem az embereket és, hogy rámutassak arra, hogy tényleges életminőség változásra van szükség a környezetünk megvédéséért és ezt egyfajta technológiai példamutatással szeretném elérni. A Rusty fejlesztésem egy olyan robotot idéz meg, amely képes a környezetében lévő szemét felismerésére gépi tanulás (machine learning) segítségével.

A szemétfelismerés követően Rusty kapcsolatba lép a környezetében elhaladó emberekkel hang és fényjelzések segítségével, melynek célja, hogy felhívja a figyelmet az eltakarítandó szemétre. Fejlesztéssel azokat a társadalmi csoportokat kívánom megcélolni, amelyek a környezettükkel szorosabb harmóniában szeretnének élni, valamint jól rezonálnak egy robot által felmutatott társadalmi tükör technológia jelenségére (social mirror tech.). Ez a technológiai tükör nagyban hozzájárulhat a társadalmi önreflexió felélénküléséhez és ahhoz, hogy élettereink autonóm módon tisztábbak és a természettel harmonikusabbak legyenek.

2.1 A megvalósítás főbb állomásai

A közterületi szemét típusainak felmérése során könnyen megállapítható, hogy egyes hulladék típusok időben és térben, valamint szezonálisan, sőt akár napszakonként is változhatnak. A nagyfokú diverzitás és variancia kiküszöbölésének érdekében az előforduló hulladékokról szintén változatos fényképgalériát kell biztosítani az adatbázisban az optimális képfelismerési hatékonyság eléréséhez. Lévén, hogy az adatmennyiség optimalizáció kiemelten fontos részét képi a fejlesztési tevékenységeknek, ezért ebben az esetben is hangsúlyos szerepet kapott az elégséges fényképmennyiség meghatározása. Előzetes szakirodalmi és szabad felhasználású kódforrások, illetve tárhely takarékosági, energiafelhasználási szempontok szem előtt tartása, továbbá a precizitás igényének fenntartásával a minimális képmennyiséget 20-120 fénykép/ hulladéktípusban lett meghatározva.

Az így létrejövő adatbázisból, Rusty egy Jetson TX2-es alaplap által gépi tanulás segítségével dolgozza fel az adatokat, mely révén gyors és hatékony

hulladék felismerés hajtható végre. A Jetson TX2 jelenleg az egyik leggyorsabb és leghatékonyabb beágyazott AI számítástechnikai eszközöknek. Az alaplapon Ubuntu (Linux disztribúció) operációs rendszer fut, mely lehetővé teszi az AI programok futtatását, valamint az internetes adatforgalmat, illetve az emberek irányába történő kommunikációt is

Rusty megvalósításához nyílt forráskódú könyvtárak kerültek implementálásra. A PyTorch, amely olyan alkalmazásokhoz használható, mint például a számítógépes látás, a természetes nyelv feldolgozása és a NumPy, amely a számítástechnika egy alapvető csomagja. A valós idejű számítógépes látáshoz felhasználok az OpenCV-t és a YOLO csomagokat is. A kamerának be kell ágyaznia egy technológiát, amely képes az arcok, a ruhák és a személyek testének elmosására, illetve a szemét még pontosabb körvonalazására. Mindezt pedig a Nvidia által kifejlesztett CUDA technológiájával érhetjük el, amely több száz CUDA mag (CUDA core) segítségével képes pillanatok alatt az adatforgalom megvalósítására.

2.2 Főbb kihívások

Rusty képfelismerési tulajdonságának fejlesztése során fokozott figyelmet igényel, hogy a közönségesnek hitt szeméttípusok morfológiája is igen változatos lehet, így a betanítás során, lényeges a felismerésre szánt objektum különböző perspektívákból való bemutatása. Ebben jelenthet megoldást az adatgyűjtemény (dataset) komplexitása.

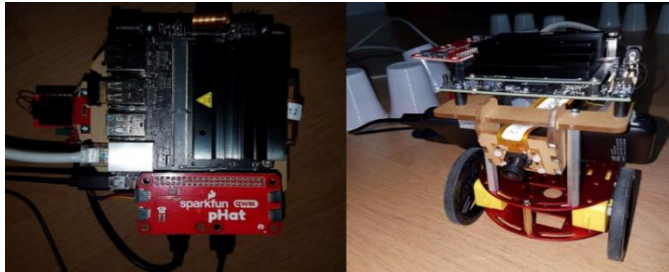
A robot összeállításánál további hardware és etikai problémák is fennálltak. Ennek egyike nem más, mint az emberekkel való ütközés. Elkerülésére, a kameramodul mellett alkalmazásra kerül egy ultrahangos távolság érzékelőből álló rendszer is, valamint a váz felépítése során a mechanikai terhelés elsődleges paraméterként szolgál. Emellett az emberek arcainak elhomályosítása szintén adatbiztonsági tényező, amely újabb szoftver csomagokat követel meg.

Egy másik jelentős kihívást képvisel Rusty és a társadalom közötti minőségi, pozitív kapcsolatrendszer kialakítása. Ennek érdekében az emberközeli, barátságos, gyerek centrikus formatervezés nélkülözhetetlen. További előnyös szempont lehet, hogyha a robot által kijelölt munkaterület mikrokörnyezetével összhangban álló színvilággal látjuk el a robotot.

3. Elért eredmények

Kutatásom során sikerült egy olyan adatbázist kiépítenem, amely képes megfelelő mennyiségű és minőségű információval ellátni, a képfeldolgozást végző modult, így számos hétköznapi hulladék felismerésére alkalmazható

Rusty. Sikerült implementálni az említett modulokat, így a képfelismerés után a robot képes hang és fény jelzések kibocsátására. Megépítésre és beprogramozásra került egy Jetson nano robotot (**1. ábra**), mely képes a gépi tanulás (deep learning) segítségével a tájékozódásra, illetve a különböző nyugalomban lévő tárgyak és objektumok felismerésére és kikerülésére, mint például a közterületeken lévő padok. A jelenlegi fejlesztés fázisában Rusty képes egyszerű háttér jelenlétében, néhány hulladék (pl. sörös doboz, pillepalack stb.) felismerésére.



1. ábra: SparkFun JetBot AI Robot Kit

3.1. Fejlesztési tervek

Annak érdekében, hogy Rusty munkássága még nagyobb impaktot érjen el, egy applikáció fejlesztésén és egy ecobuisness modellen is folyamatban van. Az applikáción keresztül a hulladékot felszedő személyek kommunikálhatnak Rustyval, így a közérdekű munkájukért digitális pontokat, Trashcoint gyűjthetnek. Az ecobuisness program célja, hogy olyan gazdasági szereplők kapcsolódjanak össze hálózatba, akik zöld termékekkel és szolgáltatásokkal dolgoznak és partnerségben Rustyval, a kiosztott pontokért kedvezményt kínálnak a Trashcoint begyűjtő személyeknek.

Egy másik jelentős célkitűzés az, hogy Rusty ne csak azonosítani tudja a szemetet, hanem kategorizálni is, még hozzá szelektíven. Ezáltal pontos útbaigazítást tudna adni a hulladékot begyűjtő embereknek, így erősítve Rusty fenntarthatóságra nevelő tulajdonságát. Valamint hosszútávon nyílt forráskódú statisztikai elemzéseket is képes lesz a nyilvánosság elő bocsátani, megosztani. Hosszú távú célkitűzés, hogy Rusty által egy egész város tisztaságának monitorizálására alkalmas rendszer jöhessen létre, amely IoT rendszerek moduljaként funkcionálhatna.

Felhasznált források:

https://drive.google.com/file/d/1W7Jo_HrsvjDZNcub2aThkG3NkuqmM1Xu/view

Pyrheliométer Nap-követő rendszerrel

SunFlower

Bejó Mátyás, Hargitai Benke, Sztojka Áron

Felkészítő tanár: Lang Ágota

Széchenyi István Gimnázium, Sopron

1. Bevezetés

A Magyar Asztronautikai Társaság ebben az évben „Csillagunk, a Nap” címmel hirdette meg diákpályázatát. Csatunk 2/3 része ezen is indult, és azt a kérdést járta körbe, hogy mennyire függünk itt a Földön a Naptól. Ennek során találkoztunk a napállandó fogalmával, így vetődött fel az ötlet, hogyan lehetne ezt az értéket megmérni. Utánanéztünk, és egy pyrheliméter nevű eszközt találtunk erre a célra. Utána olvasva, úgy döntöttünk, hogy készítünk egy saját példányt, hozzá pedig egy Nap-követő rendszert. A pyrheliméter gyakorlatilag egy fél méteres cső, ennek a végében helyezkedik el egy szenzor. Mivel a Nap irányából érkező direkt sugárzást méri, ezért a szenzor felületére merőlegesen kell beesnie a napsugaraknak, emiatt szükséges a Nap követése.

2. Eszközeink felépítése

2.1 Pyrheliméter

A vezérlést egy Arduino Mega végzi, mivel sok pinre van szükségünk egyrészt a szenzorokhoz – ebből hármat használunk -, illetve a két motor valamint az elektronika számára. A hőmérséklet mérésére egy TMP006 típusú, érintkezést nem igénylő infra termopile szenzort alkalmazunk. (Ez egy olyan eszköz, ami a hőt elektromos energiává és így elektromos jellé alakítja.) A szenzorral egy fekete felületű kis tömegű alumíniumlemez hőmérsékletét mérjük, amely a nap sugárzásának van kitéve. Hogy az égfelület többi részéről



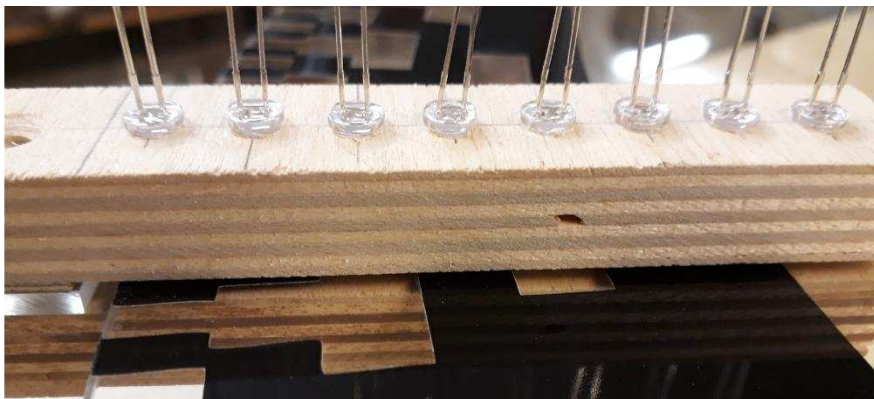
1. ábra: A pyrheliméter háza még 2 darabban és a termopile szenzor

érkező szórt fényt kiszűrjük, egy 5° látószögű 3D nyomtatott árnyékoló végében helyeztük el. (1. ábra) Ebből következően fontos az eszköz pozícionálása, $\pm 0,5^\circ$ pontossággal kell ráállítani a napra. A lemez hőmérsékletének a változásából következtethetünk a beeső teljesítményre tehát a napállandóra.

2.2 Nap-követő rendszerünk

Ehhez a Nap pozíciójára állást használjuk (fényerősség-mérés helyett). A szükséges képletet ezen a honlapon találtuk meg: <https://www.pveducation.org/pvcdrom/properties-of-sunlight/the-suns-position> Ezek az összefüggések a dátummal, helyi idővel és a földrajzi koordinátákkal számolnak, tehát ezeket az adatokat kell tudnunk. Ennek lekérdezésére egy Adafruit GPS modult alkalmaztunk.

Ha megvannak a Nap égi koordinátái (azimuth és magasság (altitude/elevation)), már „csak” be kell lőni rá a pyrheliométert. Az azimuthra állás iránytű-szenzor segítségével történik: egy léptetőmotor elforgatja kerekein a kis asztalt a teljes eszközzel, míg a másik az elevation tengelyen már csak a műszert. A magasság szögének a beállításához egy 8-bites encodert használunk. A kód egy negyedkör alakú plexilemezen helyezkedik el, a „mintát” AutoCAD-ben terveztük meg. A minta fekete csíkjai jelentik az 1-es számot, a fehér/átlátszó részek pedig a 0-át jelölik. A hagyományos egyesével növekvő beosztással ellenétben olyan módszert alkalmaztunk, hogy szomszédos számok csak 1 bitben térjenek el egymástól, ezt hivatalosan Gray kódnak nevezik. A lemez egyik oldalán 8 infra-LED, másik oldalán velük szemben 8 fototranzisztor helyezkedik el. (2. ábra) Ezek „olvassák” a kódot. Ez azt jelenti,



2. ábra: Az encoder a szerelőasztalon, egyik oldalán a LED-ek, másikon a fototranzisztorok, még beforrasztás előtt

hogy a 90 fokot 28=256 részre osztottuk, ilyen pontossággal tudjuk a magasság szögét beállítani egy másik léptetőmotorral, amelynek az encoder mondja meg, mennyit kell elfordulnia.

3. A program

Első lépésként a program lekérdezi a GPS-modultól a szükséges adatokat. A GPS-el való kommunikációhoz saját arduino könyvtárat fejlesztettünk, amellyel \$GPRMC típusú mondatokat dekódolunk.

Ezután kiszámoljuk a Nap helyzetét a fentebb említett honlapon található egyenletekkel a földrajzi pozíció és az aktuális dátum és idő alapján. Először a pontos helyi napidőt határozzuk meg, amelyhez egy listában eltároljuk az egyes hónapok napszámát. Majd különböző korrekciós értékek segítségével a Nap égi koordinátáit, vagyis az azimut és magasság szögeit számítjuk ki. (3. ábra)

```

84 void Calc() {
85   if (Year%4 == 0) {
86     for (int i = 0; i < Month-1; i++) d += ly[i];
87     d += Day;
88   } else {
89     for (int i = 0; i < Month-1; i++) d += nly[i];
90     d += Day;
91   }
92
93   dTgmt = Long/15;
94   b = 360/365*(d-81);
95   EoT = 9.87*sind(2*b)-7.53*cosd(b)-1.5*sind(b);
96   LT = Hour*Min/60+Sec/3600+dTgmt;
97   LST = LT + EoT/60;
98   HRA = 15*(LST-12);
99   ang = 23.45*sind(360/365*(d-81));
100  Elevation = asind(sind(ang)*sind(Lat)+cosd(ang)*cosd(HRA));
101  Azimuth = acosd((sind(ang)*cosd(Lat)-cosd(ang)*cosd(HRA))/cosd(Elevation));
102 }

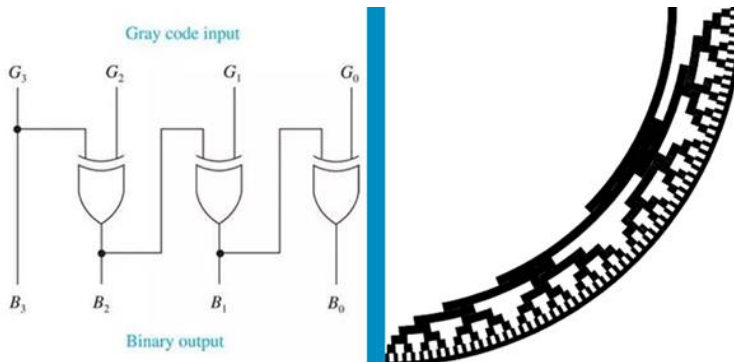
```

3. ábra: Kódrészlet a számításokból

Ezután a program átadja ezeket az értékeket a motorok számára. Az első steppermotor addig forgatja a berendezést, amíg az iránytűszenzor alapján rááll a megfelelő azimutra.

A másik steppermotor számára a Gray-kód ad információt arról, hogy mekkora magassági szögnél jár a műszer. A Gray-kód olvasásához először a szomszédos beosztásban lévő skálát vissza kell alakítani egyenletesen növekvő kettes számrendszerbeli beosztásra. Ezt a feladatot egy olyan logikai kombinációs hálózat látja el, mint ami az ábrán is látható - annyi különbséggel,

hogy a miénk 8 bitet is tud kezelni. Ezt az átalakítást a program 7db XOR logikai művelettel hajtja végre. (4. ábra)



4. ábra: A logikai műveletek illetve az encoderen található minta

```
int getPos(int pin7, int pin6, int pin5, int pin4, int pin3, int pin2, int pin1, int pin0) { //MSB-----LSB
    byte g7 = !digitalRead(pin7); //MSB
    byte g6 = !digitalRead(pin6); // ! boolean negation
    byte g5 = !digitalRead(pin5); // Tanúsítványok 1 értéke a logikai 0.
    byte g4 = !digitalRead(pin4); //
    byte g3 = !digitalRead(pin3); //
    byte g2 = !digitalRead(pin2); //
    byte g1 = !digitalRead(pin1); //
    byte g0 = !digitalRead(pin0); //LSB

    byte bin[8] = { 0 }; //bin[7] MSB----- bin[0] LSB
    //-----
    bin[7] = g7;
    bin[6] = bin[7] ^ g6;
    bin[5] = bin[6] ^ g5;
    bin[4] = bin[5] ^ g4;
    bin[3] = bin[4] ^ g3;
    bin[2] = bin[3] ^ g2;
    bin[1] = bin[2] ^ g1;
    bin[0] = bin[1] ^ g0;
    //-----
    return ((((((bin[7] * 2 + bin[6]) * 2 + bin[5]) * 2 + bin[4]) * 2 + bin[3]) * 2 + bin[2]) * 2 + bin[1]) * 2 + bin[0]));
}
```

5. ábra: Programrészlet a dekódoláshoz

4. Hol tartunk?

A pillanatnyi helyzet szerint kész van a program, amit lehetett, azt leteszteltünk (pl. külön az encoder beforgatását), azonban a részek összeillesztése egészéig még hátravan. Annyi tapasztalattal már rendelkezünk, hogy az eddigi tesztek eredményei csak kisebb optimizmusra adjanak alapot, még meg kell küzdenünk azért, hogy a teljes rendszer működjön.

Azonban ha összeáll minden, akkor precíz adatokkal rendelkezünk majd a napállandóról, amelynek értéke jelentősen befolyásolja mindennapjainkat és jövőnket is.

Informatika szekció

I Know It!

Do I Know It?

Tóth Tamás

Felkészítő tanár: Haramia László

SZTE Gyakorló Gimnázium és Általános Iskola, Szeged

1. Bevezetés

Eddigi tanulmányaim során, többször tapasztaltam, hogy a tanároknak olykor kevés vagy egyáltalán nincs idejük dolgozatokat javítani, és így a megírástól számítva a diákok több hónapon keresztül nem látják a felmérőt. Továbbá az is előfordult, hogy az oktató a tanuló kézírását nem tudta elolvasni, természetesen a dolgozat rovására. Éppen ezért találtam ki, és valósítottam meg az „I Know It!”-ot.

Az „I Know It!” egy webes szoftver, melynek lényege, hogy egy könnyen kezelhető felülettel lehessen dolgozatokat készíteni, osztályozni és megírni. Maga a dolgozatkészítő moduláris, bármikor hozzáadhatunk újabb modulokat, melyek több és több feladattípust adnak hozzá a generátorhoz. Ezekből a dolgozatfelépítő blokkokból a tanárok egy WYSIWYG szerkesztőben a „fogd és vidd” technikával egyszerűen tudnak dolgozatokat létrehozni, majd a felmérést könnyedén ütemezni és javítani.

2. Probléma megoldásának menete

A projekt elkészítéséhez először egy felhasználóbarát felületet készítettem el, amelynek elemeit egy korábbi projektből másoltam át. Az egyszerűség kedvéért több library-t használtam (pl. Bootstrap, JQuery, DataTables). A következő problémák/kihívások merültek fel a projekttel kapcsolatban:

2.1 A bejelentkezés/regisztrálás

Bejelentkezésnél és regisztrálásnál jelszavakat adunk meg, ezért minél biztonságosabb eljárást kerestem; a jelszavakat SHA-256 algoritmussal hash-elem még kliens oldalon. Brute force támadások ellen két védelmet állítottam fel:

- több sikertelen bejelentkezés után a felhasználó le lesz tiltva 2 órára
- a jelszavakat az adatbázisban egy véletlenszerű salt-tal tárolom.

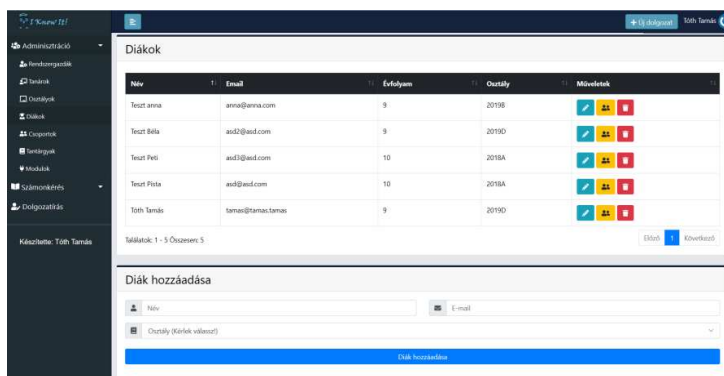
A php sessionnek a beállításait átállítottam, így biztonságosabbá téve azt.

A regisztrációt a rendszergazda végzi, majd a szoftver egy e-mailt küld a felhasználónak, amely tartalmazza a belépési adatait. A jelszót a rendszer generálja, amelyet minden belépés után könnyen meg lehet változtatni.

2.2 Adminisztráció

A rendszergazdák a szoftver beállításához, karbantartásához szükséges opciókat az adminisztráció fül alatt találják:

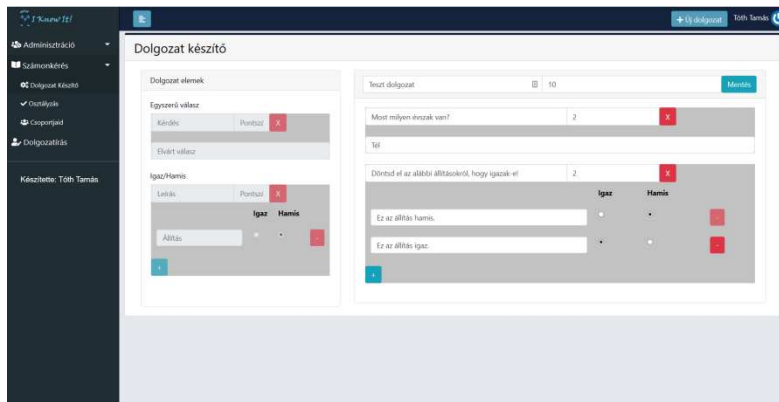
- **Rendszergazdák** - Rendszergazdák hozzáadása és törlése
- **Tanárok** – Tanárok hozzáadása és törlése
- **Osztályok** – Osztályok hozzáadása és törlése
- **Diákok** – Diákok hozzáadása, csoportba beosztása és törlése (1. ábra)
- **Csoportok** – Csoportok hozzáadása és törlése
- **Tantárgyak** – Tantárgyak hozzáadása és törlése, modulok hozzárendelése
- **Modulok** – Modul „.php” fájlok feltöltése



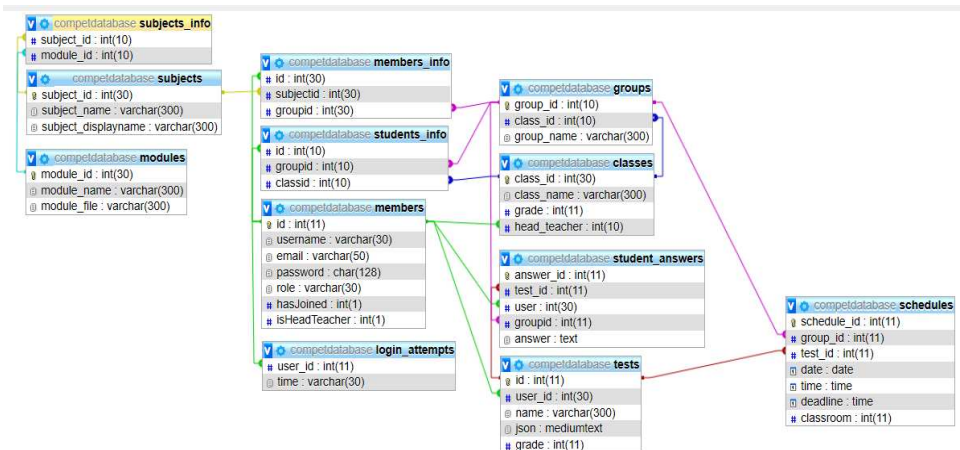
1. ábra: A diákok kezelő felülete

2.3 Dolgozatkészítő

A dolgozatkészítő (2. ábra) a saját „fogd és vidd” kódot használja. A generátor a dolgozat nevét, a célzott évfolyamot, a feladatok kérdéseit, az elvárt válaszokat, az elérhető pontszámokat és egyéb modulfüggő adatokat JSON formátumban menti el egy MySQL adatbázisba (3 ábra.), amelyet később a javítószerver értelmez. Ezeket az elmentett dolgozatokat később a tanár bármikor felhasználhatja és ütemezheti egy kiválasztott csoportra.



2. ábra: Dolgozatkészítő



3. ábra: Adatbázis szerkezete

2.4 Számonkérés és javítás

A számonkérés és a javítás a gerince az egész alkalmazásnak. A felmérés ütemezésekor a tanárnak meg kell adni a dolgozatírás napját, kezdési idejét, valamint időtartamát. Továbbá lehetősége van a megmértetettést bejelentettnek jelölni, így a diákok automatikusan kapnak egy email-t a közelgő dolgozathoz kapcsolódó információkról.

A számonkérés fül az adatbázisból, az elmentett dolgozathoz kapcsolódó JSON bejegyzésekből olvassa ki a feladatokat és generálja a diákoknak az űrlapot. Ez az oldal a számonkérés alatt folyamatos figyelmeztetést küld a hátralévő időről. Puskázás ellen a szerveren egy nyílt API-t helyeztem el,

amelyet az internetes oldalak letiltására alkalmazhatnak, mivel le lehet kérni a teremszámokat, ahol éppen dolgozatírás folyik.

A javítást a modulok végézik, majd adnak egy pontszámot, a feladatra; a standard modul levenshtein algoritmust használ az egyszerű válasz javítására, és if-else szerkezetre épülő egyszerű ellenőrzést az igaz-hamis feladatokra. Ha a tanár úgy érzi, felülírhatja a gép döntéseit a javítás után. A rendszer a dolgozat összpontszámának függvényében kiírja a százalékot, így tovább segítve az osztályzást. A javítást követően a dolgozatválaszok nem vesznek el, később ezeket meg lehet tekinteni.

3. Elért eredmények

Ez a projekt egy kihívásokkal teli fejlesztési feladat volt. Megismerkedtem az információbiztonsággal, valamint a komplexebb web-fejlesztéssel. Az alkalmazás továbbfejlesztését és karbantartását a későbbiekben is folytatom (<https://github.com/DevTOTHOMI/IKnowIt>) majd.

Azt gondolom, hogy a felületet sikerült elég intuitívvá tenni, hiszen tanárokkal dolgoztam együtt, akik felhívták a figyelmemet a lehetséges félreértésekre, azaz mit nem szoktak érteni ők és mit nem a diákok. Továbbá a modularitás miatt, és mivel open-source a projekt, bárki bármikor bármilyen dolgozatmodult tud készíteni, amelyet hozzá is tud adni a szoftverhez. A szoftver tehát bővíthető, fejleszthető.

Ugyanakkor fontos kérdés az is, hogy mennyi időt lehet a használatával megtakarítani? Egy egyszerű „röpdolgozatot” készítettem mintegy két perc alatt, amit diákként is kitöltöttem. Tapasztalatom szerint több idő maradt a részleteket átgondolni és nem kellett felesleges papírt sem pazarolni.

A szoftver az alábbiak szerinti támogatást nyújtja:

- a tanároknak részére időmegtakarítást, újrafelhasználhatóságot,
- egységes értékelést, amely ellenőrizhető,
- a diákoknak várhatóan sikeresebb dolgozatok írását,
- felesleges papírfelhasználás kiküszöbölése,
- közelebb kerülni az okos iskolákhoz:

I Know It! – „A jövő iskoláinak!”

Elérhető egy demó oldal a verseny idejére a következő helyen:
<https://compet.iceyleagons.net/> felhasználónév: demo, email:
demo@iknowit.demo, jelszó: dEm0_1234#

Caley

Dz

Fejes Dániel, Várkonyi Dániel

Felkészítő tanár: Gyuris Csaba

SZSZC Vasvári Pál Gazdasági és Informatikai Szakgimnáziuma, Szeged

1. Bevezetés

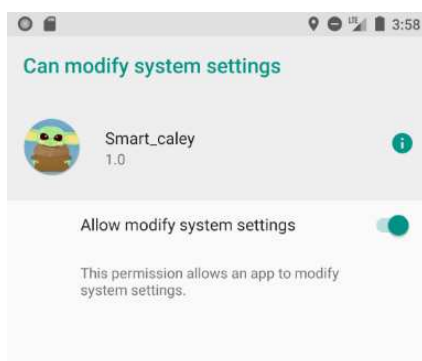
A rohanó világunkban egyre kevesebb idő marad az apróbb feladatok elvégzésére. Ezek a problémák lehetnek akár egy kapcsoló átállítása vagy a kézfék behúzása. Az utóbbira sajnos nem képes egy telefon, viszont egy folyamat automatizálására igen. A zsebünkben lévő miniszámítógép majdnem minden funkcióját képes vezérelni egy alkalmazás. A csapatunk célja az, hogy a mindennapi, szinte lényegtelennek tűnő feladatokat ütemezze. Ezek a feladatok kimerülhetnek akár az internetelés ki- és bekapcsolásában, vagy a hangprofil globális változtatásában. Ilyen egyszerű módosításokkal egy sokkal problémamentesebb, gördülékenyebb és okosabb készülék lehet a kezünkben.

2. Probléma megoldásának menete

Az első probléma a beállítások eléréséből adódóik. Habár open source licenc alatt fut az Android, jó ideje nem engedélyezi, hogy bizonyos beállításokhoz hozzáférjünk egy alkalmazás segítségével. Ezeket csak úgy lehetne alkalmazni egy ilyen programban, ha egy külön változatot készítünk az operációs rendszerből. Emiatt és más, már elavult funkciók miatt az elérhető akciók tárháza igen kicsi, de még így is nagyobb, mint amit bármilyen más jelenleg hozzáférhető programban találhatunk.

2.1 Engedélykérések

Még azon funkciók között is szükségünk van felhasználói megerősítésre, amiket elérhetővé tudunk tenni a felhasználó számára. Ezeknél a felhasználónak egy külön beállítási fülön kell engedélyeznie az alkalmazást a működés érdekében. Ezt úgy orvosoltuk, hogy az alkalmazás elindulása előtt leellenőrzi, hogy megvannak-e a szükséges engedélyei a rendszertől. Ha valamely engedély hiányozna, akkor indulás után elnavigálja a felhasználót ahhoz a beállítási fülhöz, ahol ezt a hiányosságot be tudja pótolni (1. ábra).

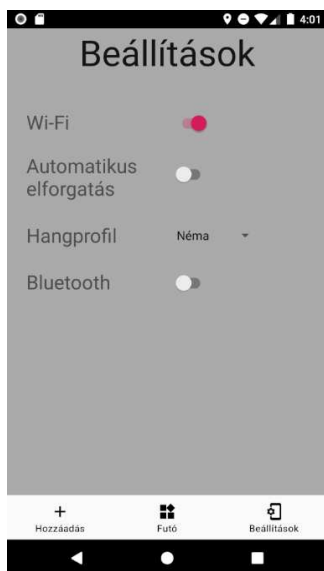


1. ábra: Az engedélykérő menü megjelenítése

2.2 A funkciók ki-bekapcsolása

Minden funkció alapjáraton automatikusan hajtódik végre, de a felhasználó személyre szabhatja, hogy és milyen körülmények között fusson le. Azonban az alkalmazásunkkal ezeket a funkciókat aktiválhatja külön is.

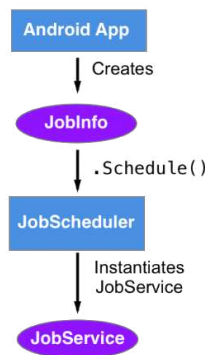
A beállításokat külön objektumokként érhetjük el a rendszerből, amit majd kapcsolóhoz (3. ábra), vagy akár egy ütemezéshez kötve tudunk irányítani. Azt is meg kell figyelni, hogy éppen be van-e kapcsolva az a funkció a felhasználó telefonján, nehogy véletlenül a rossz tevékenységet futtassa le a program.



2. ábra: A beállítások menüpont

2.3 Ütemezés

Az Android a megadott feladatokat ún. Job-okként kezeli. Egy alkalmazás ilyen feladatokat tud hozzáadni várakozásra és meg tudja adni, hogy mikor kerüljön végrehajtásra. Ennek segítségére van a `JobInfo` és a `JobScheduler` osztály (3. ábra). Minden ilyen feladatot a `JobInfo.Builder` alosztállyal lehet inicializálni. Ezeket a rendszer eltárolja, és csak akkor futtatja le, hogyha a megadott feltételek teljesülnek. Ilyen feladatok kombinációjával elérhetjük, hogy a felhasználó saját igényei szerint személyre szabhatja a telefonja reakcióit a környezeti hatásokra, mint például helyváltozás.



3. ábra: A `JobInfo` működése

3. Elért eredmények

Végeredményül egy olyan programot sikerült írunk, amely tudja figyelni és ki-bekapcsolni a támogatott beállításokat, akár ütemezni is tudja őket. Egy egyszerű, mégis tartalomban gazdag dizájnnal sikerült előrukkolni, ami nem fog a felhasználó számára fejfájást okozni. Habár eleinte több funkciót szerettünk volna a programban tudni, a limitációk miatt csak ennyi sikerült.

Mivel iskolai körülmények között ekkora mértékű kihívással nem találkozhattunk, így hatalmas erőfeszítést okozott a legegyszerűbb funkciók kidolgozása. A teljes környezetváltozás miatt az idő volt a legnagyobb ellenségünk. Viszont optimistán tekintünk előre, hogy mit tudunk kihozni ezen a kiaknázatlan területen, és teljes erőbedobással törekszünk tudásunk bővítésére és a projektünk legjobb kivitelezésére.

Néhány kép az alkalmazásból:



Új generációs receptoldal

Tűzvarázsló

Durucz Ádám István

Felkészítő tanár: Kovács Gábor

Szolnoki SzC Jendrassik György Gépipari Szakgimnáziuma, Szolnok

1. Bevezetés

Napjainkba, a felgyorsult világ következtében nagymértékben megváltoztak az emberek mindennapi szokásai. Már teljesen természetesnek vesszük, hogy egy pillanat alatt elérünk bármilyen információt az internet segítségével. Ez nem meglepő, hiszen az információ korszakát éljük. A megváltozott világ igényeinek viszont jó pár alkalmazás nem felel meg. Néhány alkalmazás, ami még pár évvel ezelőtt kielégítette a felhasználók igényeit az lehetséges, hogy manapság már elavult.

A beküldött pályaművemmel egy olyan probléma megoldást tűztem ki célul, amivel segíteni és megkönnyíteni szeretném a felhasználók mindennapi életét. Mivel azt tapasztaltam, hogy az online elérhető receptoldalak elavultak, ezért hoztam létre egy mai világ igényeinek eleget tevő webalkalmazást.

Mivel a felhasználó sok esetben azonnal akar mindent, ezért egy étel elkészítése során nem a boltba akar menni. hanem az otthon lévő alapanyagokból gyorsan el akarja készíteni a számára megfelelő finom ételt. Az alkalmazásom azt a célt szolgálja, hogy a felhasználónak idejét feleslegesen a boltba töltenie, hanem az otthon lévő alapanyagokból gyorsan el tudja készíteni a számára megfelelő ételt. Az ötletemmel az emberek felesleges pénzköltést és időt is takaríthatnak meg egyaránt.

Emellett arra is kiváló felületet biztosít, hogy a receptjeinket megosszuk másokkal.

2. Probléma megoldásának menete

Úgy gondolom a problémát a legjobb tudásom szerint sikerült megoldani. Mivel egy problémára sok különböző megoldás létezik, ezért nem volt könnyű megtalálnom az általam vélt legjobbat. Az alkalmazásom egy Apache szerver szolgálja ki. Az adatbázis választás során a relációs adatbázisokat találtam legalkalmasabbaknak, ezen belüli is MySQL választottam, mivel ez teljesen jól ki tudja szolgálni a szerveremet. Mivel az oldal teljes mértékben dinamikus, ezért szükségem volt egy szerveroldali programozási nyelvre. Ennek eldöntése nem volt könnyű dolog, mert nyelvnek van előnye és hátránya. Én a PHP-t szerveroldali szkriptnyelv választottam, mivel az a nagy előny szólt a neve

mellett, hogy HTML oldalba ágyazható. Az alkalmazás frontend részén JavaScriptet és CSS-t alkalmaztam. Az alkalmazás versenyre történő beküldés érdekében egy webszerverre töltöttem fel, hogy könnyen elérhető legyen(<http://34.65.105.234/>).

2.1 Kezdő oldal

Az oldalra lépve egy leegyszerűsített dizájnnal találjuk szembe magunkat. Ez igazából azért előnyös, mert könnyű navigálni az oldalon. Viszont ezt a megjelenést még lehetne fejleszteni, hogy szebb legyen. Ezután ki tudjuk választani egy kereső segítségével, hogy milyen élelmiszer alapanyagok találhatók meg otthon. Ki kell választani az alapanyag mennyiségét, ezután pedig rákattintani a hozzáadás gombra. Ha minden alapanyagot hozzáadtuk, akkor rámehetünk a receptek kilistázása gombra. Továbbiakban két dolog történhet, a program vagy kilistázza a megadott hozzávalók alapján milyen ételt tudunk elkészíteni, vagy kiírja, hogy ilyen recept még nem található az adatbázisban és lehetőségünk van feltölteni receptet. Ha a kilistázott receptre kattintunk, megtudhatunk minden információt annak elkészítéséről. Az oldalt úgy lehetne még tovább fejleszteni, hogy minden recepthez lenne egy hozzá csatolt kép, mert jelenleg a felhasználók nem látnak képet ez elkészítendő ételről.

2.2 Recept feltöltés

Az oldalra lehetőségünk van receptek feltöltésére is. Ezt a felső navigációs sávba recept feltöltésnél érhetjük el. Egy főoldalhoz hasonló oldalon ki kell választanunk az általunk feltöltésre kívánt recept hozzávalóit. Ezután kattintsunk az Alapanyagok kiválasztva gombra. Továbbiakban meg kell adni a recept nevét és elkészítésének menetét. Ha minden adatot helyesen adtunk meg és rámentünk a feltöltés gombra akkor az oldal jelzi számunkra, hogy sikeres volt a feltöltés. Minél több feltöltött recept található az oldalon, annál használhatóbb és jobb.

2.3 Adminisztrációs felület

Azonban a feltöltött receptek nem kerülne azonnal publikálásra. A recepteket egy adminisztrátornak ellenőriznie kell. Erre azért van szükség, hogy nehamis recepteket töltsenek fel a felhasználók. Ezt a felső navigációs sávban a beépítés gombnál tudjuk megtenni. Előre létrehozott accounttal lehetőségünk van belépni az adminisztrátori felületre. Felhasználónév: admin, jelszó: admin.

Ezután láthatjuk a feltöltött recepteket, amiket meg tudunk tekinteni, illetve bírálni. Ha az elfogadásra megyünk a recept nyilvánossá válik és bárki elérheti az oldalon. Viszont, ha elutasításra megyünk, mivel valaki hamis receptet töltött fel akkor a recept törlésre kerül.

3. Elért eredmények

Úgy gondolom a program megírása során sokat fejlődött a probléma megoldó képességem. Mivel az alkalmazást egyedül csináltam, ezért csak magamra számíthattam, ha valami nem működött vagy nem sikerül. Ez viszont nagy mértékben megnehezítette a munkámat. De a kitűzött célomat sikerült elérnem, megoldani az általam kiválasztott problémát. A legjobb tudásom szerint csináltam és próbáltam a legtöbbet kihozni az oldalból.

Menetrend

TODO: Change name

*Futaki Bálint, Vajda Simon
Felkészítő tanár: Kutas Tibor*

Kecskeméti Bolyai János Gimnázium, Kecskemét

1. Bevezetés

A tömegközlekedés sokak számára az elsődleges megoldás a mindennapi ingázásban, utazásokban. Azonban elég frusztráló, amikor nem tudjuk pontosan, hogy mikor és melyik járatra kell felszállnunk, hogy a leggyorsabban odaérjünk a kívánt úticélhoz. A forgalmasabb városokban az üzemeltetők honlapja vagy a Google Maps segítséget nyújt az útvonal tervezésében, de sajnos ez nem minden városban elérhető. Az mi lakóhelyünkön, Kecskeméten ugyan rendelkezésre áll egy online menetrend, de útvonalat nem tud tervezni, ahogyan a Google Maps sem. Erre a problémára nyújt segítséget a Menetrend alkalmazás.

2. Probléma megoldásának menete

Egy város összes megállóját és járatát figyelembe véve megtalálni a leggyorsabb útvonalat egy embernek már komoly feladatot jelent. Szerencsére a megfelelő eljárásokkal egy számítógép ezt gyorsan el tudja végezni. A probléma megoldásához a város buszhálózatát egy gráfnak tekintettük, melyben a megállók a gráf pontjai, és a megállók között közlekedő buszok a gráf élei.

2.1 Az adatok rendszerezése

Mielőtt nekiláthattunk volna a legrövidebb útvonal kereséséhez, szükségünk volt egy rendszerezett adathalmazra. Először megterveztük, hogy az útvonalkereséshez milyen adatbázis táblákban kell eltárolni az információkat, majd megkerestük, hogy honnan tudnánk ezeket az adatokat beszerezni. Szerencsére a menetrend.kecskemet.hu webhelyről HTTP lekérdezésekkel le tudtuk tölteni az adatokat JSON formátumban, és néhány sor kóddal rendszereztük őket a megfelelő táblákba egy SQLite adatbázisban.

2.2 Útvonal kereső algoritmus

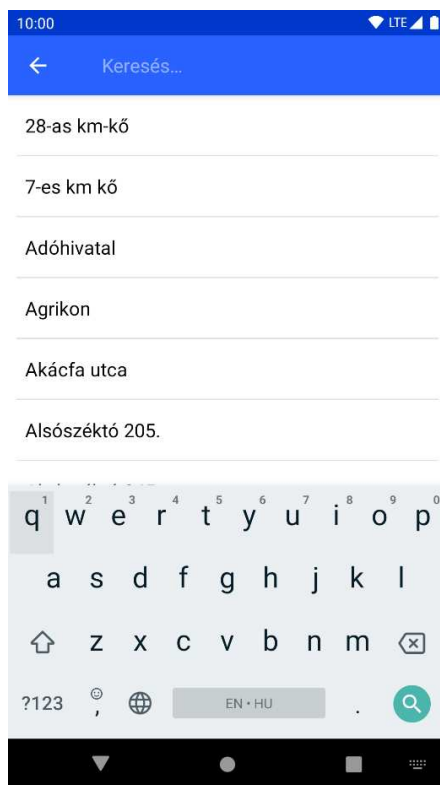
A buszhálózatot ábrázoló gráfban az élek irányítottak, mivel minden járat egy irányban közlekedik, és súlyozottak is. Két megálló közötti él súlya függ a menetidőtől és a várakozási időtől. Némi kutatómunka után megtaláltuk, hogy

melyik bejárési algoritmus lenne a leoptimálisabb ehhez a problémához: Dijkstra algoritmusára esett a választás. Az útvonaltervezés alapvetően erre épül, némi változtatásokkal. A súlyok meghatározásánál ugyan a menetidőt előre tudjuk, viszont a várakozási idő attól függ, hogy mikor ér a megállóba az előző járat, és mikor indul a következő. Így tehát minden él súlyát futásidőben számolja ki a program.

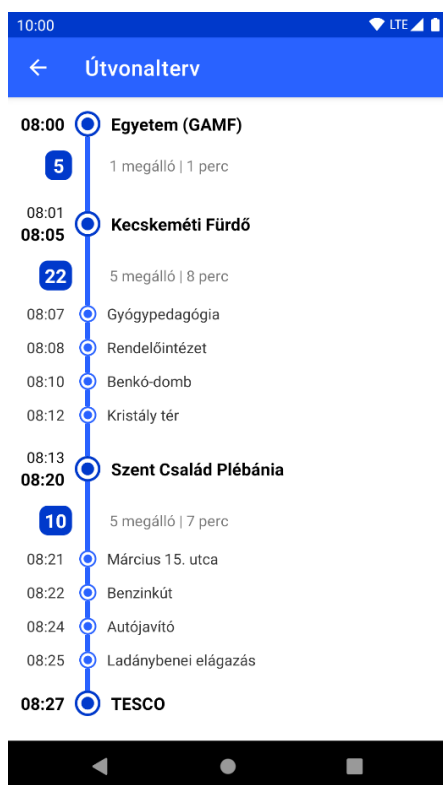
2.3 Alkalmazás, és felhasználói felület

Az algoritmus egyszerű, és a jövőben akár szélesebb körű felhasználását egy androidos alkalmazás teszi lehetővé. Az adatbázis az alkalmazásban van, így az offline használható, nem szükséges hozzá internet kapcsolat. A felület egyszerű és rendezett. A főképernyőn, ami az 1. ábrán látható, megadható a kiindulási és cél megálló, valamint az indulás időpontja. A megállókat egy listából lehet kiválasztani, amit egy kereső mező könnyít meg (2. ábra). A megtervezett útvonal pedig a 3. ábrán látható.

1. ábra: Főképernyő



2. ábra: Megálló választása



3. ábra: Útvonalterv

3. Elért eredmények

Az alkalmazás erre a versenyre készült, máshol még nem volt bemutatva. Úgy gondoljuk, hogy egy ilyen jellegű program sokak mindennapi közlekedését könnyítheti meg. Habár eddig csak a kecskeméti menetrend érhető el, és a közel jövőben az alkalmazás funkcióinak bővítése és tökéletesítése a célunk, egy hasonlóképpen rendszerezett adathalmazon ugyanúgy működik az útvonaltervezés. Ez lehetővé teszi, hogy a későbbiekben akár más városok menetrendjét is kezelni tudja a program.

ParkMyst

Házizz velünk

Horváth Gergely, Stefán Kornél, Vad Avar

Felkészítő tanár: Répásné Babucs Hajnalka

BMSZC Neumann János Számítástechnikai Szakgimnáziuma, Budapest

1. Bevezetés

Történetünk egy szerdai osztályfőnöki órával kezdődött, amikor a négy nap múlva kezdődő osztálykirándulásunk kalandját terveztük. Felmerült a gondolat, hogy lehetne egy kincsvadászatot tartani, amiben az osztály csoportokra oszlik, ide-oda mászkál, és feladatokat old meg. Erre viszont létező és számunkra is elérhető megvalósítást nem találtunk. Az elkövetkező 4 hosszú nap emiatt azzal telt, hogy kifejlesztettünk egy prototípust, ami ezt az egy alkalmat kiszolgálta. Miután kisebb veszteségek árán sikert aratott a projekt, felmerült bennünk, hogy ezt az eszközt továbbfejlesztve még mi mindenre lehet alkalmas.

Célunk egy olyan rendszer megalkotása volt, ami lehetővé teszi interaktív, történet alapú élmények létrehozását, amelyek segíthetnek a játékosok érdeklődésének felkeltésében egy adott témakörben, egy rendezvény lebonyolításában, vagy egyszerűen az osztálykirándulások tartalmassá tételében.

2. Fejlesztési szempontok

Az alábbi elvekben határoztuk meg elvárásainkat a végső szoftverrel kapcsolatban:

- Felhasználói élmény centrikusság, a játékosok és játékmesterek igényeire optimalizálva.
- Mély és összetett játékok készítése programozási ismeretek nélkül.
- Könnyen bővíthető keretrendszer kialakítás az egyéni igények kielégítésére - programozási ismeretekkel.
- Univerzális, önmagában használható szoftver elkészítése.

3. Probléma megoldásának menete

A probléma megoldásához több programot fejlesztettünk ki, amik együttes működésével érjük el a célunkat.

3.1 Kliens

A felhasználói felület elkészítéséhez a web technológiák világát választottuk, és azon belül a React könyvtárat, a modern és könnyen bővíthető kialakítása végett, ami megkönnyíti a jó felhasználói élmény létrehozását.

A felhasználók egy újragondolt jelszórendszer (1. ábra) használatával jelentkezhetnek be a játékba, hét piktogramot kell kiválasztaniuk az előre megadott lehetőségek közül. Ezzel a módszerrel el lehet kerülni olyan gyakori problémákat, mint például a hibás jelszó bevitel.

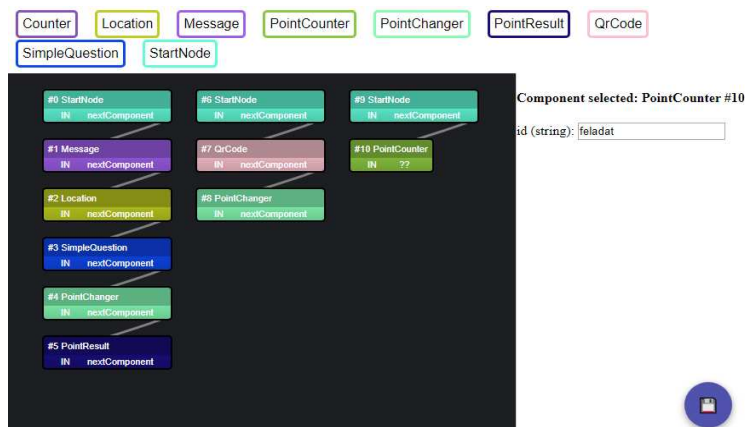


1. ábra: Bejelentkezési felület telefonon

Ha már létezik elindított játék, akkor belépés után a játékosok egyből láthatják az aktív játékokat. Ez tartalmazza például az üzeneteket és a feladataikat, illetve egy térképet, ahol láthatják, hogy hova kell menniük a következő feladatért.

Mindezt az adminisztrátorok hozhatják létre, a játékosokkal és csoportokkal együtt. A játékosok jelszavát az adminisztrátori felületen gombnyomásra ki lehet exportálni - így könnyen oda lehet adni ezeket a

játékosoknak. Emellett játszani könnyedséggel készíthetőek játékok, az alaphoz elérhető „építőkövek” felhasználásával, mint például egy „egyszerű kérdés” vagy egy „menj ide” komponenssel. Az elemeket csak be kell húzni és összekötni, majd megfelelően konfigurálni, és már kész is a játék.

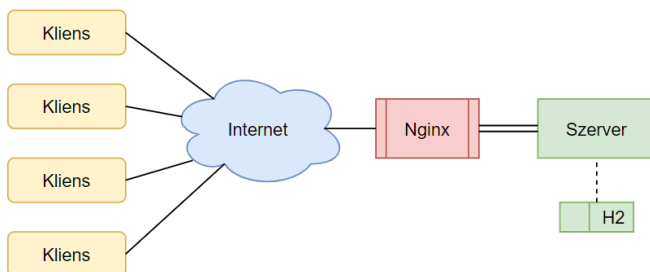


2. ábra: Egyszerű játék a szerkesztőben

A kliensen továbbá lehetőségük van a folyamatban lévő játékba is belenézni. Itt valós időben láthatóak a játékban történő események, maga a játék állapota, illetve egy térképen követhető a játékosok mozgása. Szükség esetén ugyanitt be is lehet avatkozni a játék menetébe megszakítás nélkül.

3.2 Kiszolgáló

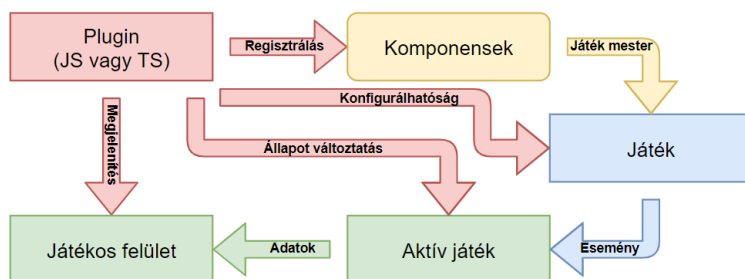
A szerver Java nyelven íródott, Spring Boot keretrendszert használ a kiszolgálására. A szerver továbbá rendelkezik egy beépített H2 adatbázissal, amiben az adatokat tárolja. A szolgáltatás minősége érdekében és a támadási felület csökkentése érdekében a szerver egy Nginx kiszolgáló mögött található.



3. ábra: Kiszolgálás modell

3.3 Testreszabhatósági rendszer

A könnyű bővíthetőséget egyrészt a komponens rendszer használatával érjük el, amiben az adminisztrátoroknak lehetősége van az alap játékot bővíteni JavaScript vagy TypeScript nyelvben íródott pluginokkal. Ezek a pluginok hozzáadhatnak a fentebb említett játék szerkesztőhöz további építőköveket, amivel olyan funkciókat lehet rakni egy játékba, amivel alapból a szoftver nem volt ellátva. Továbbá ezek a komponensek a játék létrehozásánál egyedileg konfigurálhatóak mind funkciójukban, mind megjelenésükben, így változatosságot érhetünk el további fejlesztés nélkül.



4. ábra: Plugin feladatai

4. Elért eredmények

A 2019. május 1-i ötlet első, koncepció verzióját május 5-re készítettük el, és ezt 7-én az egész osztályunk bevonásával ki is próbáltuk.

Az ebből levont következtetések alapján, egy kisebb pihenő után, 2019. december 7-én nekiláttunk, hogy kibővítsük és tökéletesítsük a ParkMyst projektet.

2020. január 31-én már egy jóval tartalmasabb és stabilabb verziót teszteltünk le. Eddigre elkészült az első bétának mondható verzió.

A szoftver használatával tervezzük lebonyolítani a 2020. március 15-ei megemlékezésre szervezett iskolai programot, melynek egy egész osztályos főpróbája március 5-én lesz.

GraphChat

INeedMoreFreeTime

Laczkó Őrs

Felkészítő tanár: Kovács Irén

Kerék Általános Iskola és Gimnázium, Budapest.

1. Bevezetés

Egy korábbi versenyre készített adatvizualizációs programom szolgált inspirációként a GraphChat ötletéhez. Az alapgondolat egy olyan weboldal volt, ami egy számítógépen tárolt .txt kiterjesztésű file-t, vagy egy szövegmezőbe írt számsorozatot olvas be, majd egy animációval megjeleníti, így segítve azokat, akik hosszú távon szeretnének valamilyen tevékenységet mérni (pl.: testmozgás, kalóriabevitel, testmagasság, testtömeg). Azonban hamar be kellett lássam, hogy nem túl felhasználó barát, hogy az egyetlen mentési mód egy programon kívüli file szerkesztése. Mivel a weboldal fejlesztése során még nem lehetett közvetlenül a számítógépen tárolt fájlokba tartalmat menteni, más megoldás után kellett nézmem. Így jött a képbe a Google Cloud platformra épülő Firebase, a maga számtalan és folyton bővülő - kezdetben ingyenes-micro service-eivel, ahol a legmodernebb biztonságtechnikákkal vannak ötvözve különböző szerver oldali szolgáltatások (pl.: adatbázisok, file elosztók, hosting, stb.). A Firebase különböző szolgáltatásai óriási inspirációként hatottak, hogy hogyan bővíthetném az alapötletet. Ahogy egyre több micro service-szel ismerkedtem meg, úgy jöttek az ötletek, hogy mivel bővíthetném a programot. A jelenlegi elképzelésem, egy olyan webes applikáció, ahol az azonos érdeklődési körű felhasználók megoszthatnak egymással hasznos tippeket történeteken keresztül (pl.: hogy mit hol érdemes megvenni), tarthatják egymással a kapcsolatot (az applikációba épített csetnek köszönhetően - 1. ábra), vagy megoszthatnak egymással fontos adatokat privát üzenetben vagy csoportos üzenetben (pl.: egy edzőteremben, egy edzőcsoport a heti teljesítményüket). Emellett még számos apró beépített funkciót valósítottam meg, vagy vár még megvalósításra.



1. ábra: Az applikációba épített cset

2. Probléma megoldásának menete

2.1 React

Probléma megoldásának menete Mikor az alapötletből kiindulva elkezdtem a GraphChat-et fejleszteni, hamar rájöttem, hogy ekkora mennyiségű HTML-kódot összefésülni mind a szerver- mind a kliensoldali logikával rendkívül nehéz lenne. Nem beszélve arról, hogy mivel webes applikációról van szó, fontos, hogy az URL az adott tartalomra mutasson, és ez vagy azt jelentené, hogy az applikáción belül minden új felhasználói felületnél lenne egy plusz 1-2 másodperces várakozási idő, amíg a szükséges file-ok megérkeznek a szerverről, majd egy újabb 1-2 másodperces várakozási idő (készüléktől függő) amíg a tartalmat a böngésző rendereli, vagy azt, hogy egy külön JavaScript könyvtárat kéne ahhoz használnom, hogy az URL megváltozásakor, amíg az oldal domain nevén van a felhasználó, megakadályozza a re-renderelést, és ez után a DOM (document object model) felülírásával változtassa meg a felhasználói felületet. Egyik variációt se szerettem volna, így döntöttem végül egy JavaScript framework, - a Facebook által fejlesztett - ReactJS (2. ábra) mellett, ami amellett, hogy saját routing-gal rendelkezik - így megoldva az URL-s problémám -, úgy nevezett RxJS (Reactive Extensions for JavaScript) szintaxot használ, ami egyrészt lehetővé teszi a felhasználói felület komponensekre bontását, másrészt lehetőséget biztosít, arra, hogy közvetlenül a HTML kódba lehessen logikát és JavaScript változókat használni. További előny, hogy a komponensekre bontott felhasználói felületnek a részeit újra fel lehet használni, (például elég egyszer

implementálni az avatár kódját, és azt fel lehet használni mind a történetek, mind a cset beszélgetések során). A ReactJS jelenlegi verziója úgynevezett Virtual DOM-ot használ, ami azért előnyös, mert ha új tartalmat kell megjeleníteni - mert például érkezik egy új üzenet - először a Virtual DOM-ot frissíti, majd összehasonlítja a Virtual DOM-ot a HTML DOM-jával, és csak azt a részt frissíti, ami eltér, ezzel jelentősen csökkentve a gép terhelését.



2. ábra: ReactJS

2.2 Material-UI

A felhasználói felület fejlesztéséhez, a native hatás elérés érdekében egy ReactJS-re épülő keretrendszert, a Material-UI-t használtam. A Material-UI egy nyílt forráskódú, előre definiált komponensekből álló könyvtár, amely keretrendszer-szerű funkciókkal rendelkezik (pl.: Hook API, CSS stílusokhoz).

2.2 Firebase

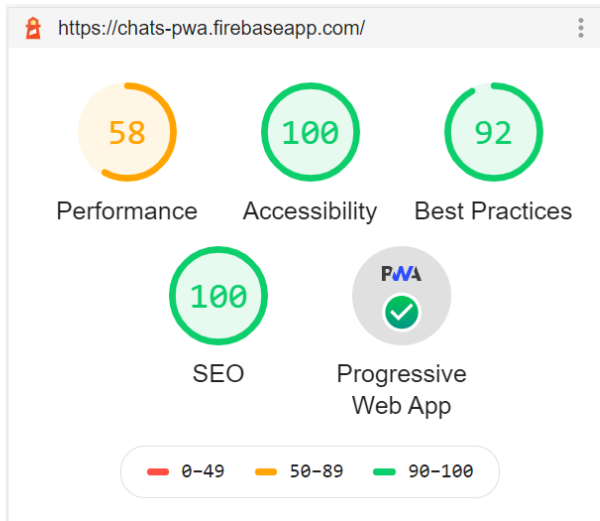
A szerver oldali kiszolgáláshoz és a weboldal hostinghoz Firebase-t (3. ábra) használtam. A program jelenlegi állapotába számos Firebase szolgáltatást vesz igénybe: Firebase Authentication, a felhasználók bejelentkeztetéséhez és kliens és szerver oldali megjegyzéshez. Firebase Realtime Database, a gyakran olvasott és írt adatok tárolásához (pl.: cset üzenetek, grafikonok), Firebase Firestore, a ritkábban olvasott, nagyobb tárhelyigényű adatok tárolásához (pl.: felhasználó profil), Firebase Analytics, a felhasználói élmény javításához és még sok más.



3. ábra: Firebase

2.3 Service Worker

Azért, hogy teljes képernyős, native-hatású applikációként, kezdőképernyőről indítva is lehessen használni, Service Worker-t használtam, így az applikáció PWA (progressive web app) lett. (4. ábra)



4. ábra: Lighthouse PWA Analysis Tool-lal készített report az applikációról.

3. Elért eredmények

A felhasználó az applikáció jelenlegi állapotában képes cset üzeneteket küldeni és fogadni, valamint saját sztorit posztolni és másokét megnézni. Jelenleg a grafikonok mentését és vizualizálását implementálom react-chartist segítségével. Mivel egy egy rész implementálása rendkívül időigényes, az applikáció még közel sincs kész. Bár már több hónapnyi kemény munkám van benne, még mindig rengeteg ötletem van arra vonatkozóan, hogy mivel tudnám bővíteni. Amellett, hogy új funkciókkal szeretném bővíteni, a meglévőket is szeretném továbbfejleszteni: átláthatóvá és könnyen fenntarthatóvá tenni. Szeretném emellett a jelenlegi biztonsági szabályokat megerősíteni az adatbázisokon és a legtöbb kliensoldali adatbázissal kapcsolatos függvényt áthelyezni Firebase Cloud Functions-ba, szintén biztonsági okokból.

BuildUp

Janus3D_BuildUp

Rendes Botond, Kis-Bogdán Kolos

Felkészítő tanár: Dlusztus Péter

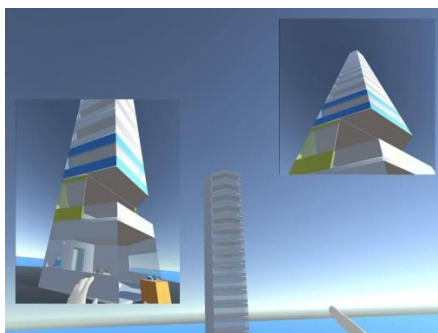
Pécsi Janus Pannonius Gimnázium, Pécs

1. Bevezetés

A BuildUp egy térgeometria oktatóprogram virtuális valóságban. A szoftver lényege a különböző térgeometriai problémák szemléltetése és kidolgozása egy VR környezetben, ahol a felhasználó bekerülhet ebbe térbe és figyelemmel követheti a megoldásokat, valamint interakcióba léphet az elemek többségével.

2. Probléma megoldásának menete

Az alkalmazás elsődleges célja az új ismeretek megtanítása. Ennek megfelelően a hangsúly a magyarázatokon és szemléltetéseken lesz. Először a térgeometria alapjaival ismerkedik meg a felhasználó, majd a fokozatosan nehezedő anyagok biztosítják majd az újdonságokat és kihívásokat. Ezeknek az egymásra építése ihlette a projekt nevét, és ez a lineáris haladás, tanulás lesz a szoftver gerincét alkotó történetre felfűzve. A hagyományos oktatóprogramokkal ellentétben (például: Geogebra), ahol a felhasználó egy 2D-s kijelzőn látja a tartalmakat, a mi alkalmazásunk lehetővé teszi, hogy a felhasználó ténylegesen 3D-ben élhesse át a térgeometriai szemléltetéseket egy VR szemüveg segítségével. A programot elsődlegesen VR szemüvegre fejlesztjük, de tervezzük egy részét okostelefonos alkalmazás formájában mindenki számára elérhetővé tenni.

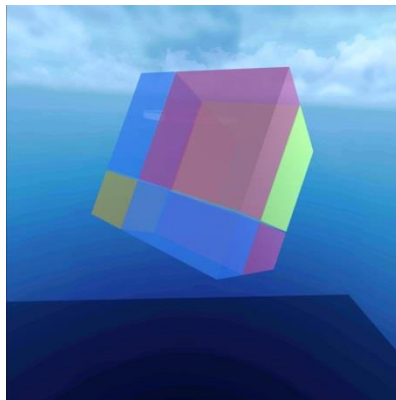


1. kép: Az épülő torony

2.1 A fejlesztés

A BuildUP Unityben készült. A Unity olyan objektumorientált fejlesztői keretrendszer, amelyben egy-egy alkalmazás programozható objektumokból (GameObjectek) épül fel. Ezekhez csatolhatók különböző komponensek (scriptek), amelyekkel az alkalmazás programozható. A scripteket C# programnyelven írtuk. A fejlesztés során kihasználtuk a VR nyújtotta térbeliséget, hogy minél immerzívebb élményt hozhassunk létre. Az alkalmazás két fő részre osztható:

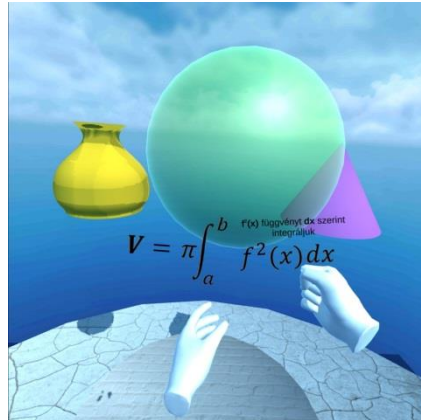
Az épülő torony (1. kép), ahol a felhasználó egyre magasabb szintekre jutva egyre több és fokozatosan nehezedő feladattal és feladvánnyal szembesül. Minden szint egy (vagy több) térgeometriát bemutató animációból, illetve egy ahhoz kapcsolható feladványból áll. Ezen kívül a szintek tartalmaznak más interaktív elemeket is, melyeknek célja a felhasználó motiválása. A torony megalkotásánál a legnehezebb feladat egy olyan rendszer elkészítése volt, amely lehetővé teszi, hogy gyorsan és hatékonyan adhassunk a meglévőkhöz új, egymástól teljesen eltérő szinteket. Ennek segítségével a torony, tehát az oktatóanyag könnyen bővíthető, akár más tárgykörökkel, tantárgyakkal is.



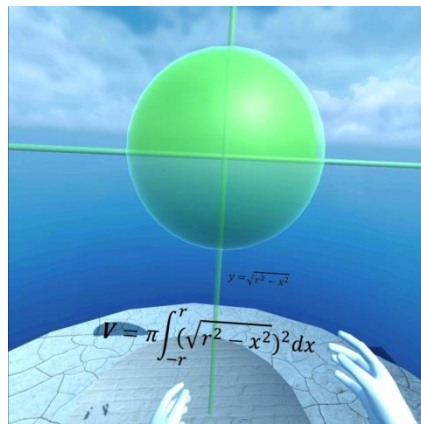
2. kép: Animáció az $(a + b)^3$ nevezetes azonosság geometriai szemléltetéséhez

Oktató animációk (2-4. kép), melyek egy-egy térgeometriai problémát-tételt szemléltetnek, bizonyítanak az alaptestek származtatásától kezdve a forgástestek térfogatának integrálással való kiszámításáig. Itt a legfőbb cél az anyag világos, érthető magyarázata volt. Így az animáció során egyszerre jelennek meg képletek és a hozzájuk tartozó testek, interaktív ábrák. Mindezt egy előre felmondott magyarázó szöveg kíséri. A felhasználónak akár arra is

lehetősége van, hogy az animációt megállítsa, vagy, hogy szó szerint kezébe vegyen egy képletelemet. Ekkor megjelenik az adott képletdarabhoz tartozó magyarázó szöveg vagy ábra.



3. kép: Animáció az integrálás megtanulásának elősegítéséhez



4. kép: Integrálás (gömb térfogata)

3. Elért eredmények

Az rendszer teszteléséhez az Oculus Quest VR headsetet használjuk, ami a gyártó legújabb standalone headsetje. A VR fejlesztéshez az Oculus Integration package-et használtuk, amely lehetővé teszi, hogy az alkalmazás használható legyen a kiválasztott VR headset-tel. Az interaktív objektumok teljesen saját fejlesztésűek (gombok, karok, stb.) Saját megoldást dolgoztunk ki többek között a VR-kezek ütközésére is. Az alkalmazás kisebb változtatásokkal kompatibilis a SteamVR-ral, ezáltal bármely VR headsettel is.

Jelenleg a Build^{UP} áll egy tutorial szintből és ezen kívül még négy oktató-fejlesztő szintből, melyek között találunk alaptestekkel, geometriai összefüggésekkel, algebra és geometria kapcsolatával, integrálással és magasabb dimenziójú testekkel foglalkozó animációkat is. Továbbá minden szinten található egy-két érdekes feladat az adott témához kapcsolódóan.

Miért gondoljuk, hogy innovatív?

- 3D-ben oktat 3D-s tartalmakat: A felhasználó nem egy 2D-s kijelzőn látja a „3D-s” programot, hanem bekerül ugyanabba a 3D-s világba, ahol az események, szemléltetések zajlanak.
- Ez a környezet új szintre emeli a bevonódást, az „immersiót” is: A felhasználó a kezébe veheti a tárgyakat, saját maga fedezheti fel a szinteket
- A történet és az állandó feladatok motiválják a felhasználót
- Még nem létezik térgeometria oktató program VR-ban, egyedülálló
- Az eszközökből adódóan a headseten kívül nem kell hozzá semmi (kábelek, számítógép)
- Az oktatóanyagok hozzáférhetők bármilyen eszközről (okostelefon, stb.) 360-as videó formájában
- Könnyen eladható, terjeszthető, mivel szoftver.
- Unityben fejlesztettük, így a program lefordítható szinte bármilyen eszközre
- Személyre igazodik: tempójában, elmélyülésben

Mivel a Unity egy multiplatformos fejlesztőrendszer, így a későbbiekben az alkalmazást egyszerűen elérhetővé tudjuk tenni más eszközökre, vagy platformokra is.

MyStudiez – Órarend alkalmazás diákoknak

MilanCodes

Herke Milán

Felkészítő tanár: Vargáné Magusics Erika

EJSz Széchenyi István Gimnázium, Pécs.

1. Bevezetés

Régóta a fejemben volt egy alkalmazás ötlete, melynek segítségével könnyebbé tehetem saját magam és mások iskolai életét is.

Miután középiskolába kerültem, rögtön szembesültem egy személyes problémával – nagyon szétszórt voltam. Elegem volt a papír órarendekből, amiket folyamatosan elhagytam, a házfeladatokból, amiket sajnos sokszor elfelejtettem és a dolgozatokból, amikre mindig az előző éjszaka kezdtem el készülni. Úgy gondoltam, hogy nem én vagyok az egyetlen diák, aki ilyen problémákkal küzd. Gimnáziumi vagy akár egyetemi szinten sem.

Az alkalmazás tulajdonképpen e problémákra ad megoldást. Segítségével teljes mértékben digitalizálható egy diák, vagy akár egy tanár órarendje is. Amellett, hogy a felhasználó mindennap láthat egy részletes áttekintést a napjáról, felírhatja a megoldásra váró házfeladatokat, teendőket, illetve a közelgő dolgozatokat, eseményeket is. Továbbá a felhasználó, amennyiben kívánja, minden hozzáadott feladat/dolgozat előtt egy általa választott időpontban értesítve lesz.

2. Probléma megoldásának menete

2019 decemberében kezdtem el dolgozni az alkalmazáson. Megfelelő körülmények hiánya miatt, mivel az Apple csak MacOS-ról teszi lehetővé a rendszereikre való fejlesztést, úgy döntöttem, hogy Android operációs rendszerre fogom megírni az applikációt.

2.1 Megfelelő nyelv és Android verzió kiválasztása

Habár a fejemben volt, hogy Flutter-t fogok használni a fejlesztéshez, ami megoldást jelentett volna még az iOS/Android problémára is, hiszen a Google által kibocsátott SDK lehetővé teszi a cross-platform programok írását, de végül teljesen elvettem az ötletet, mert nem akartam, hogy egy több platformon futó alkalmazásnak egy közös kódbázisa legyen. Többek között teljesítménybeli gondokhoz is vezethetett volna. A natív Android fejlesztés egyszerűen szimpatikusabb volt és úgy gondoltam, hogy ahogy egyre nagyobb és nagyobb lesz a projekt, egyszerűbb is lesz átlátni.

Androidon belül is egy kisebb problémába ütköztem, hiszen el kellett döntennem, hogy Java-t vagy Kotlin-t fogok használni. Végül a Kotlin mellett döntöttem, hiszen a Google 2019 nyarán bejelentette, hogy az lesz az „első-nyelvük” és az Androidra történő fejlesztés „Kotlin-first” lesz. A következő feladat a minimum API-szint kiválasztása volt, amiben javarészt az alábbi táblázat segített:

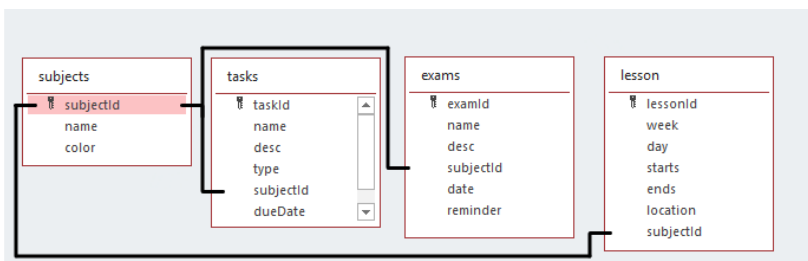
Verzió	Kódnév	API	Felhasználók
<6.0	-	<23	25.2
6.0	Marshmallow	23	16.9
7.0	Nougat	24	11.4
7.1		25	7.8
8.0	Oreo	26	12.9
8.1		27	15.4
9	Pie	28	10.4

1. táblázat: OS verziók megoszlása az összes Androidos készüléken 2019 májusában

Mivel az egyik legtöbb újdonságot bevezető API az Android 6.0-ás verziójával jött be, és az Android felhasználók több, mint 74%-a Marshmallow-t vagy újabb Android verziót használ, ezért a döntésem az Android 6.0 (API-level 23) mellé esett.

2.2 Az adatbázis tervezése

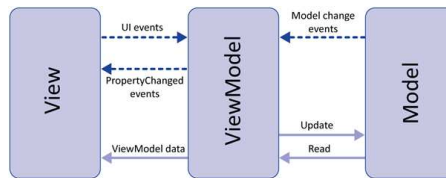
Az alkalmazás egy SQLite3 adatbázist használ, ami egy relációs, gyors, illetve nyílt forráskódú adatbázis, továbbiakban támogatja az SQL szintaxist. Alapvetően 4 táblát használ az applikáció. Egyet a tantárgyak tárolására (subjects), egyet a tanórák tárolására (lesson), egyet a feladatok tárolására (tasks) és egyet a dolgozatok tárolására (exams). Az 1. ábrán megtekinthető az applikáció adatbázissémája:



1. ábra: Adatbázisséma az alkalmazás 0.3.3-as verziója után

2.3 Az alkalmazás architektúrája

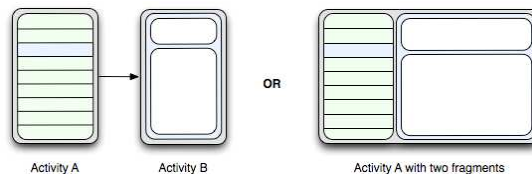
Még a fejlesztés megkezdése előtt el kellett döntenem, hogy milyen architektúráis mintát fog követni az alkalmazásom. A választásom végül a MVVM-re (Model-View-ViewModel, vagy magyarul Modell-nézet-nézetmodell) esett. Tulajdonképpen kettéválasztja a felhasználói felületre és az üzleti logikára vonatkozó kódot. Így minden UI-ra vonatkozó osztály mellé (Pl.: UiExampleFragment.kt) készült egy ViewModel osztály is (Pl.: UiExampleFragmentViewModel.kt), ahol például meghatározásra kerül, hogy mit csináljon egy gomb, vagy ahol lekérdezzük, frissítjük az adatokat az adatbázisból. Az architektúra használata miatt sokkal könnyebben átlátható lett a projekt. A 2. ábra demonstrálja a működésének lényegét:



2. Ábra: MVVM architektúra működése

2.4 Az alkalmazás UI részei

Az alkalmazás egy Activity-vel (MainActivity.kt) dolgozik, amibe számos fragment-et tölt be. Egyrészt azért döntöttem a fragment-ek használata mellett, mert a fejlesztések későbbi szakaszában az egyes fragmentek könnyedén újra felhasználhatóak, másrészt, mert így sokkal egyszerűbb lesz táblagépekre optimalizálni a layout-ot (ahogy azt a 3. ábra is mutatja). Továbbá minden fragment-nek van egy ViewModel párja, a MVVM architektúráis mintának megfelelően.



3. ábra: Fragment-ek és Activity-k használata különböző méretű eszközökön

2.5 Tesztelés

Úgy vélem, hogy még az ilyen, kisebb méretű projektek fejlesztése közben is fontos a tesztek írása, hiszen segítségükkel sokkal egyszerűbben

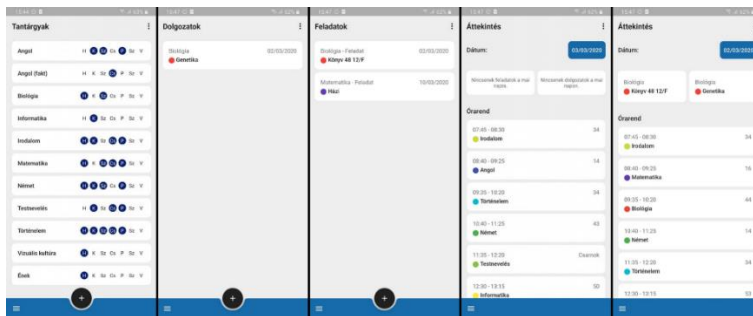
megtalálunk egy bugot a kódban, sokkal egyszerűbben észrevesszük, ha valami nem úgy működik az alkalmazásunkban, ahogy azt mi szeretnénk. A tesztek megírásához a JUnit4 keretrendszert használtam. Idő hiánya miatt jelenleg csak az adatbázisra vonatkozó egységteszteket tartalmazza a projekt.

2.6 Versioning

A szoftver lefejllesztése előtt egy SCRUM-táblát készítettem, ahol 4 részre osztottam a teendőket: „Közelgő verziók”, „Jelenleg fejlesztés alatt”, „Teszt írásra vár” és végül az „Implementált” kategória. Így nem csak sokkal átláthatóbbak lettek a teendőim, de ha esetleg a későbbiekben valaki be akar szállni a fejlesztésbe, sokkal könnyebben fel tudja venni a fonalat és egyszerűbben el lehet osztani a feladatokat majd a fejlesztők között.

3. Elért eredmények

Jómagam sokat tanultam a projekt elkészítése során, hiszen előtte még nem programoztam Kotlin nyelven és az XML Layoutok elkészítésében sem voltam túlzottan jártas, de végül sikerült minden tervezett funkciót implementálni a programba. Az alkalmazás jelenlegi állapotáról több képernyőkép is látható a 4. ábrán.



4. Ábra: „Áttekintés”, „Tantárgyak”, „Feladatok”, „Dolgozatok” Képernyőképek

Közel 3 havi munka után, használható lett az alkalmazás. A projektet teljesen nyíltforráskódúvá tettem, hiszen, ha már diákoknak készült, akkor minden diáknak joga van hozzá, hogy teljesen ingyen használhassa, illetve, ha valaki úgy érzi, hogy segíteni szeretne a fejlesztésbe, arra is lehetősége van. A projekt megtekinthető a [GitHub \(https://github.com/Studician/android-client\)](https://github.com/Studician/android-client) oldalán, ahol részletesebb információk találhatóak a telepítésre, code style-ra vonatkozóan. Így, ha bárki bármilyen problémát észlel, nyithat egy új issue-t és akár el is kezdhet dolgozni a megoldásán.

Communit.me

MakulátLANok

Szijártó Bence, Cselényi Dániel, Csorba Ákos Bence

Felkészítő tanár: Pásztorné Petréttei Noémi

Gödöllői Török Ignác Gimnázium, Gödöllő

1. Bevezetés

A mai oktatási rendszerben kötelező az 50 óra közösségi szolgálat teljesítése, hogy érettségizni tudjanak a diákok. Ez sokaknak nehézséget okoz, illetve tanácstalanok azt illetően, hogy hol végezhetnék el a feladatot. Az iskolában általában csak egy-két munkát ajánlanak, mint például a könyvpakolást a könyvtárban, vagy az iskola udvarán való segítségnyújtást (kertészet, ásás, gyomtalanítás).

Adott egy diák, Tájékozódó Terka, aki nem szeretne fizikai munkát végezni, ezért nem tudja, kihez forduljon. A tanuló személyes adatlapja az 1. ábrán látható. A feladat az, hogy segítsünk rajta.

Tájékozódó Terka
17 éves
Tanuló

“Miért pakoljak a könyvtárban könyveket, ha valami számomra érdekes munkát is tudnék vállalni?”

Célok

- 50 óra közösségi szolgálat teljesítése
- Hasznos, érdekes munkavállalás

Akadályok

- A lehetőségek korlátozottsága
- Nem tudja, hol dolgozhatna

Nyitottság

Törődés

Tájékozottság

Facebook, Messenger, Instagram, Snapchat, WhatsApp

1. ábra: Tájékozódó Terka adatlapja

Ebből leszűrhető, hogy a célközönség a mai fiatalság, a diákság;

- akik közösségi szolgálatukat úgy szeretnék teljesíteni, hogy az élvezhető és hasznos legyen, viszont a lehetőségek ismerete nélkül nem tudják, hogy kihez forduljanak.
- akik diákmunkát szeretnének végezni, hogy egy kis zsebpénzt keressenek, de ugyancsak nem tudják, hova mehetnének el.

2. Probléma megoldásának menete

Létrehoztunk egy mobil alkalmazást Android típusú készülékekre, amelyben rendszereztük azokat a diákmunkákat, ahol a kötelező közösségi szolgálatot lehet teljesíteni vagy akár némi zsebpénzt keresni.

Az applikáció neve: „Communit.me”, amely két fontos dolgot egyesít, a közösséget (community), illetve engem (me). A logója a 3. ábrán látható. Ezzel a programmal tudunk segíteni az olyan diákokon, akik tájékozási lehetőségek híján nem ismerik a munkalehetőségeket diák korukban.

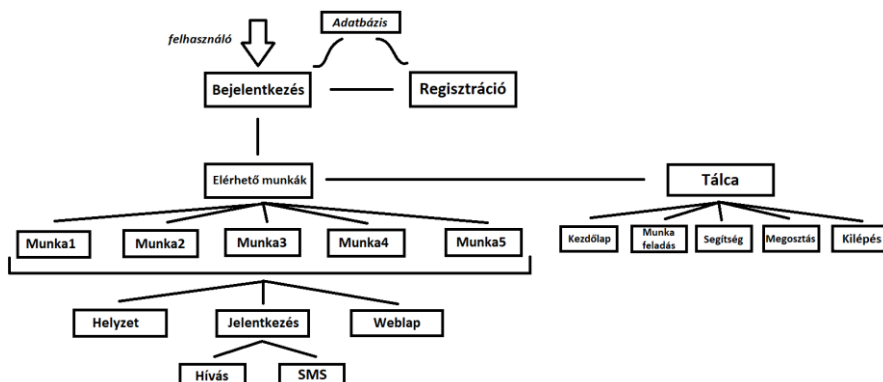
2.1 Szoftverfejlesztői dokumentáció

Az alkalmazást a MIT App Inventor online felületén fejlesztettük. Az applikáció .apk kiterjesztéssel tölthető le egyelőre csak a szoftverfejlesztők számára. Amennyiben a program sikeressé és ismertté válik, akkor feltöltjük a Google Play Áruház-ba, ahol már mindenki számára elérhető lesz.

Programozás szempontjából a szoftver logikai felépítése a 2. ábrán látható, ami a következőket tartalmazza:

- i.) Felhasználói adatbázist (User Registration System), amelynek az adatait egy weboldalon tároljuk (felhasználónév, jelszó), így lehetővé tettük azt, hogy minden egyes felhasználónak külön profilja legyen. A regisztráció után csupán egy felhasználónév és jelszó megadásával beléphetünk saját profilunkba. Mivel személyes adatok kerülnek megadásra, így a törvények betartása érdekében létrehoztunk egy Felhasználási Feltételek (Terms and Conditions) felületet, amely elfogadásával lehet csak regisztrálni. A felhasználói adatbázis rendszerét a 4. ábrán tekinthetjük meg.
- ii.) Kezdőképernyőt, ahol az elérhető munkákat láthatjuk, amelyek közül szabadon választhatunk. Ezt és ennek alfelületeit az 5. ábra mutatja.
- iii.) Tálcát, melyet a 6. ábrán láthatunk. Ezt csak a lényeges felületeken lehet megtalálni, ahonnan elnavigálhatunk a kezdőképernyő, a munkafeladás, a segítségkérés, az applikáció megosztása felületekre, sőt még a profilunkból is kijelentkezhetünk.
- iv.) Munka adatlapot, melyet a kiválasztás után érhetünk el, ahol láthatjuk a munka képét, nevét, értékelését, helyzetét, illetve weboldalát. Itt tudunk jelentkezni is, ahol két lehetőség közül választhatunk, a telefonos tárcsázás, vagy az üzenetküldés.
- v.) Munkafeladás felületet, ahol pár információ megadása után feltölthetjük a központi adatbázisba az adatokat, amelyhez a

- fejlesztők, az adminisztrátorok férhetnek hozzá és válogathatják ki a megfelelő munkákat, amelyek később feltöltésre kerülhetnek.
- vi.) Segítségkérő felületet, ahol kérdést tudunk feltenni, hibát tudunk bejelenteni, illetve fejlesztési tanácsot tudunk adni a fejlesztők számára. Az előzőt, illetve az adott oldalt a 7. ábrán találhatjuk.
- vii.) Megosztás felületet, ahol bármely közösségi médián meg tudjuk osztani az adott alkalmazást, hogy növeljük a népszerűségét.

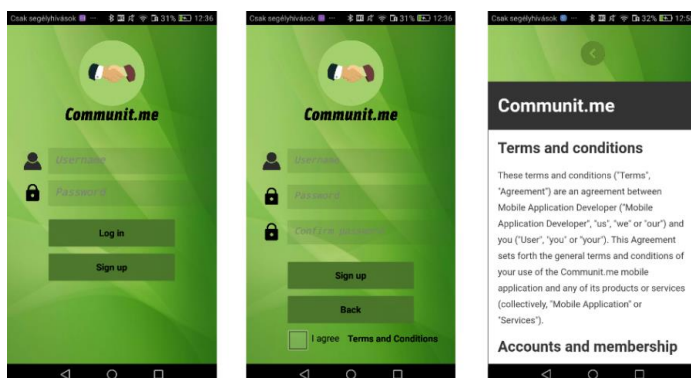


2. ábra: Szoftver logikus felépítése

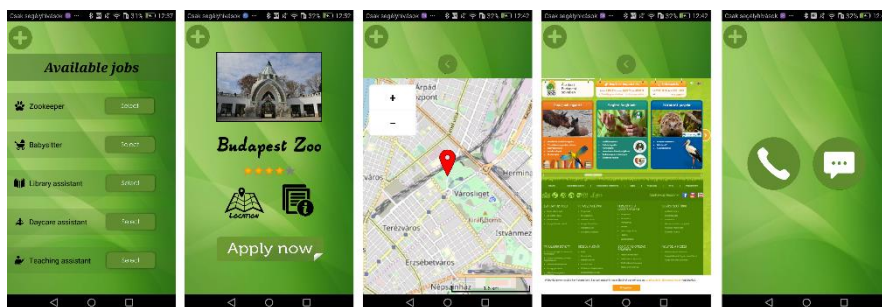
2.2 Felhasználói felület



3. ábra: Communit.me logója



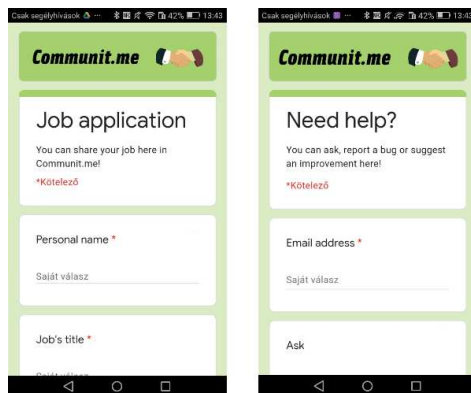
4. ábra: Felhasználói adatbázis rendszer



5. ábra: Kezdőlap és annak alfelületei



6. ábra: Tálca



7. ábra: Munka regisztráció és segítségkérés

3. Elért eredmények

Tájékozódó Terka a közösségi médián keresztül tudomást szerzett a mobil alkalmazás létéről, letöltötte, majd használatba vette. Megnézte a munkalehetőségeket, majd sikeresen jelentkezett az állatgondozói munkára a Budapesti Állatkertben. Manapság Tájékozódó Terka egy rendkívül befolyásos állatorvos, aki rengeteg család háziállatát kezelte, gyógyította meg, ezzel boldogságot, örömet szerezve a kis kedvenceknek, illetve a családnak egyaránt.

A Communit.me alkalmazás használata nélkül Tájékozódó Terka lehet, hogy nem fedezte volna fel tehetségét, érdeklődését az állatokkal kapcsolatos munkakör iránt, enélkül pedig sok boldogság és öröm elveszhetett volna, vagy akár egy élet is...

Face Trace

Team Face

Horváth István

Felkészítő tanárok: Esztelecki Péter, Kőrösi Gábor

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

1. Bevezetés

A [Face Trace](https://github.com/Isti01/Face-Trace) (<https://github.com/Isti01/Face-Trace>) egy társkereső alkalmazás. Célja az, hogy hatékonyabbá tegye a társkeresést az által, hogy figyelembe veszi az algoritmus, hogy milyen kinézetű emberek tetszenek a felhasználónak. A jó algoritmushoz egy kitűnő felhasználói felület társul, azért, hogy az alkalmazás használata minél könnyebb és élvezhetőbb legyen.

A backendként a Google Firebase-t használja. A Firebase segítségével az alkalmazás akár több eszköz között is valós időben szinkronizálja az adatokat. A Firebase több-százezer felhasználót is elbír.

Az alkalmazás értesítést küld a felhasználónak, amikor üzenete érkezik, vagy amikor egy pár egymásra talál az alkalmazásban, így senki sem marad le a legújabb eseményekről.

A pároknak lehetőségük van chatelni, így megbeszélhetik az első találkát, hiszen a párkapcsolatnak a való életben kell, hogy történjen, az alkalmazásnak csak az a célja, hogy rátaláljunk a párunkra.

2. Az alkalmazás fejlesztésének a menete

2.1 Backend

2.1.1. Az adatbázis

Adatbázisként a [Cloud Firestore](#) került felhasználásra. A Firestore egy dokumentum-orientált NoSQL adatbázis. Az adatbázis kollektiókból (táblázat) és dokumentumokból áll (táblázati sor). A fő különbsége egy SQL adatbázistól az, hogy a dokumentumokban tárolt adatnak nincs előre meghatározott alakja vagy típusa, és nincs kapcsolat az adatok között. Az adatokhoz való hozzáférés kezeléséhez és adat validáláshoz biztonsági szabályokat kell írni.

2.1.2. Tárhely

A felhasználói fileok (képek) tárolására a [Cloud Storage](#)-t használja az alkalmazás. A tárhelyre feltöltött fileokhoz tartozó url-ek kerülnek tárolásra az adatbázisban, majd azok kiolvasásával lehet hozzáférni a tárolt tartalomhoz.

2.1.3. Hitelesítés, beléptetés, regisztráció

A felhasználói fiókokat a [Authentication](#) kezeli. Minden fiókhoz tartozik egy egyedi azonosító (uid), név, profilkép és email cím. Az adatbázisban az uid segítségével utalunk az adott felhasználóra.

2.1.4. Szerveren futó kód, szolgáltatások közötti kommunikáció

A szerveren az adatok aggregálásához az adatbázisban, az értesítések elküldéséhez, az arcok osztályzásához és a potenciális párok ajánlásához a [Cloud Functions](#) segítségével kerül a kód lefuttatásra. A függvényeket meg lehet hívni HTTP kérésekkel és adatbázis eseményekkel (pl dok. íráskor) is.

2.1.5. Értesítések kiküldése

Az értesítéseket a [Cloud Messaging](#) segítségével kapják meg a felhasználók. Az értesítéseket a készülékeknek lehet küldeni, nem a kliensnek. Minden fizikai eszköznek van egy egyedi azonosítója (token). A felhasználókhoz tartozó azonosítók az adatbázisban kerülnek tárolásra, majd, amikor egy felhasználónak értesítést küld a rendszer, akkor az adatbázisból kiolvasott tokenekhez tartozó eszközökre küldi el.

2.1.6. Arcok osztályzása, felhasználók ajánlása

Minden felhasználóhoz tartozik egy vektor, ami az arcáról generálódik regisztrációkor egy Cloud Function segítségével. A vektorokat a [Google FaceNet](#) egy implementációja ([facenet-pytorch](#)) generálja a feltöltött képből. Erre azért van szükség, hogy a felhasználók arcképét olyan formába hozzuk, hogy könnyen össze lehessen hasonlítani másokéval.

Ajánláskor a rendszer figyelembe veszi az érdeklődési kört, távolságot, életkort és figyelembe vesz azt, hogy mennyire hasonlít a személy azokhoz, akiket a felhasználó előzőleg kedvelt. A vektorokat úgy hasonlítja össze, hogy kiszámolja a két vektor közötti szög koszinuszát, tehát skaláris szorzat / két vektor hosszának szorzata, minél nagyobb a kapott érték, annál hasonlóbb a kettő. Az algoritmus ezek a paraméterek alapján készít egy rangsort, amit megkap a felhasználó.

2.2 A mobil alkalmazás

2.2.1. Felhasználói felület

Az alkalmazás a Flutter frameworkkel készült. A framework jó választás, különleges felhasználói felületek készítése esetén, ugyanis a Flutter segítségével minden egyes pixel az irányítása alatt van a programozónak. A

Flutter deklaratív, komponens alapú keretrendszer, könnyű egy megírt elemet újrahasonítani, több helyen is felhasználni az alkalmazásban.

A cél az volt, hogy az alkalmazást érdekes és könnyű legyen használni. Az alkalmazásnak a színe változtatható, egy új színnel egy új hangulatú alkalmazást kaphatunk. Navigációkor minimalisztikus, de ugyanakkor játékos animációk növelik az élményt.

Az alkalmazás nem csak több színben, hanem több nyelven érhető el, így több emberhez juthat el. Screenshotok 1. ábra.

2.2.2. Háttérlogika

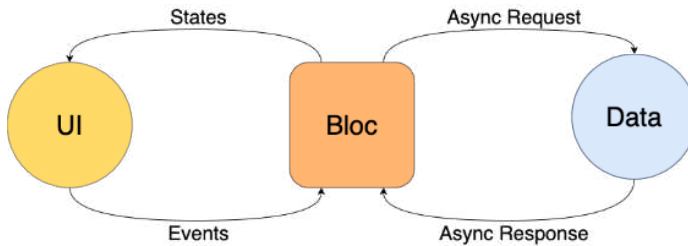
A logika, ami összeállítja, hogy mi jelenjen meg a UI-on, a [BLoC pattern](#) segítségével az alkalmazás valós időben reagál a bejövő eseményekre, legyen szó adatról, ami a backend felől érkezik, vagy akár egy gombnyomásról, ami a felhasználó felől érkezik.



1. ábra: Regisztráció, Kezdőlap, Felhasználó adatainak megtekintése, Saját adatok szerkesztése, Chat, A felhasználó párra talált

A BLoC egy absztrakt osztály. A legfontosabb elemei: A State Stream, az Event Stream, és az az eljárás (mapEventToState), amivel egy eseményből (Event) egy alkalmazás állapotot (State) hoz létre 2. ábra.

A Face Traceban több implementációja működik a BLoC-nak. Egy felel azért, backend által ajánlott emberek a képernyőre kerüljenek. Egy leellenőrzi a megadott adatokat és elvégzi a regisztrációt. Egy másik figyel, hogy éppen be van-e jelentkezve a felhasználó, és ha igen akkor szinkronizálja az adatait az adatbázissal. Egy nyilván tartja a chatszobákat és egy felel a chatelésért.



2. ábra: Bloc felépítése

3. Elért eredmények

Az alkalmazás fejlesztése egy jó élmény volt, sok tapasztalatot szereztem, és rengeteg dolgot megtanultam. Ez az első olyan munkám, ami a felhasználók körében megállna a helyét, az app egy használatra kész termék.

- Tanultam az arcfelismerés működéséről
- További tapasztalatot szereztem a machine learningről
- A cloud szolgáltatásról, a serverless infrastruktúráról
- Szerveroldali programázásról
- A NoSQL adatbázisokról
- Fejlődtem alkalmazáskészítés terén, megtanultam komplex kezelőfelületeket és animációkat írni.

A Face Trace a legnagyobb volumenű projektem, és minden eddig szerzett tudásomat felhasználtam benne.

Együttműködő partnereink

Együttműködő partnereink



National Instruments

www.ni.com/hu-hu.html



4iG Nyrt.

www.4ig.hu



Optin Kft.

www.optin.hu



EPAM Systems Kft.

www.epam.com



3i Fejlesztő és Szolgáltató Kft.

www.3i.hu



Digital Kft.

www.digital.co.hu



Creativ_IT

www.creativit.hu



Morgan Stanley

www.morganstanley.com



Lufthansa Systems

Lufthansa Systems Hungária Kft.

www.lhsystems.com



TEConcept Hungary Kft.

www.teconcept.hu



CAS Software

www.cas-software.hu



LogMeIn
www.logmein.hu



MVMI Zrt.
www.mvmi.hu



EGYÜTT
 A JÖVŐ MÉRNÖKEIÉRT
 SZÖVETSÉG

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



**SZTE Természettudományi
 és Informatikai Kar**
www.sci.u-szeged.hu



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

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

SZÉCHENYI 2020



MAGYARORSZÁG
 KORMÁNYA

Európai Unió
 Európai Szociális
 Alap



BEFEKTETÉS A JÖVŐBE



Kiadta: SZTE TTIK Informatikai Intézet

Készítette: Dr. Jász Judit

Design: Dr. Németh Gábor

Nyomda: Innovariant Nyomdaipari Kft.

Készült: 100 példányban

Együttműködő partnereink:



EGYÜTT
A JÖVŐ MÉRNÖKEIÉRT
SZÖVETSÉG



SZTE TTK
INFORMATIKAI INTÉZET



4G

<realtiv_it



Lufthansa Systems

OPTIN

Digital

Morgan Stanley

<epam>

TEConcept
Hungary Kft.

LogMeIn®

