

Szegedi Innovatív Informatika Verseny

2019

Együttműködő partnereink:



Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

SZIIV Program (2019. április 5. – péntek)

Időpont	Műszaki informatika szekció	Informatika szekció
9:00–10:00	Regisztráció	
10:00–10:30	Megnyitó	
10:40–12:20	Előadások I.	
10:40–11:00	LogB Project: LogB - Mérj bármit, bárhol, bármikor	Bingoo!: Guess What Will Happen!
11:00–11:20	Bláthy42: Dallamcsengő	Chuck Norris: Learning Machine
11:20–11:40	2_1337_4_U: Smart thermostat system	BeHealthy Team: BeHealthy étrend és edzésterv tervező
11:40–12:00	Bánkirobot: Underwater cyclops	Gucci Flip Flops: MilAxe
12:00–12:20	Berendi Bence: Kulcsautomata	Házizz velünk: Házizz
12:30–13:30	Szendvicsebéd	
13:40–15:20	Előadások II.	
13:40–14:00	Bánkirobot EMG: Biodaq 2.0	The Stylers: Innovatív képfilterek Machine Learning Technikákkal
14:00–14:20	SopRobotics: EsőCseppek	ANN Generator: ANN Generator
14:20–14:40	Bánkirobot_Smog: Every Breath You Take	Sentimentalists: SentiVision
14:40–15:00	Felső ToronyHáz: ToronyMentő Dőlésgátló	Graphite – Relációs adatanalízis gráfokkal a felhőben
15:00–15:20	Sziksi Robot TÉP: Intelligens Szortírozó Rendszer	KGHSZ: JegyKalk
15:20–16:00	Eszközök, szoftverek előkészítése Kávészünet	
16:00–17:30	Eszközök, szoftverek bemutatása	

KTPK Program

Időpont	2019. április 5. – péntek
9:00–18:30	A három verseny döntője, záró rendezvénye
9:00–17:30	Szegedi Innovatív Informatika Verseny SZTE Bolyai épület
13:00–18:30	Polygon Pályázat Matematikából Középiskolásoknak SZTE Bolyai Épület
15:00–18:00	Játsszunk fizikát! Telkes Mária Emlékverseny SZTE Dóm tér 9.
18:00–20:00	Vacsora
	2019. április 6. – szombat
8:30–13:15	Interaktív előadások és játékos versenyek – Együtt a Jövő Mérnökeiért Szövetség, Eredményhirdetés SZTE Dóm tér 9.
13:15–14:00	Szendvicsebéd



Tartalomjegyzék

<i>SZIIV program</i>	1
<i>KTPK program</i>	2
<i>Előszó</i>	5
<i>Programbizottság</i>	6
<i>Műszaki informatika szekció</i>	11
Kolláth István Tibor, Remény Olivér:	
<i>LogB - Mérj Bármit Bárhol Bármikor</i>	13
Kőrösi Ádám, Dudás Miklós:	
<i>Dallamcsengő</i>	17
Kiss Ádám, Tankó-Bartalis Bence:	
<i>Smart thermostat system</i>	21
Tóth Bence, Zsigó Miklós:	
<i>Underwater cyclops</i>	25
Berendi Bence:	
<i>Kulcsautomata</i>	29
Vas Bertalan:	
<i>Biodaq 2.0 - Biológiai mérőlabor középiskolásoknak</i>	33
Bejő Mátyás, Molnár Barnabás, Sztojka Áron:	
<i>EsőCseppek</i>	37
Halász Tony Wang:	
<i>Every Breath You Take</i>	41
Horváth Boldizsár, Kenessey Zsombor, Róth Gergely:	
<i>ToronyMentő Dőlésgátló</i>	45
Magyar Dávid, Felföldi János, Maródi Gergő:	
<i>Intelligens Szortírozó Rendszer</i>	49
<i>Informatika szekció</i>	53
Gordos Bence:	
<i>Guess What Will Happen!</i>	55
Antal Péter:	
<i>Learning Machine</i>	59
Skoda Barnabás:	
<i>BeHealthy étrend és edzésterv tervező</i>	63
Kovács Alex Márk, Szász Milán:	
<i>MilAxe</i>	67

Isakov Erik, Stefán Kornél, Vad Avar:	
<i>Házizz</i>	71
Horváth István, Hugyik Kornél:	
<i>Innovatív Képfilterek Machine Learning Technikákkal</i>	75
Laczkó Örs:	
<i>ANN_Generator</i>	77
Horváth István, Kónya Leon:	
<i>SentiVision</i>	81
Lévai Zsolt Bálint:	
<i>Graphite - Relációs adatanalízis gráfokkal a felhőben</i>	85
Kocsis Kamilla, Cseri Gábor:	
<i>JegyKalk</i>	89
<i>Együttműködő partnereink</i>	93

Előszó

A Szegedi Tudományegyetem Informatikai Intézete immáron ötödik 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 bemutatókon és interaktív programokon vehetnek részt.

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

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

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

A versenyre a két szekció zsűrije 13 belföldi és határon túli település 23 iskolájából beérkezett 34 pályamű alapján összesen 20 csapatot válogatott be a szegedi döntőbe, mely 2019. április 5-én került megrendezésre az SZTE Bolyai épületében.

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

Szeged, 2019. április

Dr. Vadai Gergely
a SZIIV verseny főszervezője

Programbizottság

Műszaki informatika szekció

A zsűri elnöke:



Dr. Gingl Zoltán
tanszékvezető egyetemi tanár
SZTE Műszaki Informatika Tanszék

A zsűri tagjai:



Dr. Mingesz Róbert
adjunktus
SZTE Műszaki Informatika Tanszék



Dr. Pletl Szilveszter
főiskolai tanár
SZTE Műszaki Informatika Tanszék



Kovács Tamás
ügyvezető
Optin Kft.



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

Informatika szekció

A zsűri elnöke:



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

A zsűri tagjai:



Dr. Vinkó Tamás
egyetemi docens
SZTE Számítógépes Optimalizálás Tanszék



Dr. Kertész Attila
egyetemi docens
SZTE Szoftverfejlesztés Tanszék



Fritz Zsombor
fejlesztési központ vezető, Szeged
NNG Kft.



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

Szervezők



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



Molnárné Rajki Rita
ügyintéző
SZTE Informatikai Intézet

Műszaki informatika szekció

LogB - Mérj Bármit Bárhol Bármikor

LogB Project

Kolláth István Tibor, Remény Olivér

Felkészítő tanár: Sikó Dezső, Pap-Szigeti Róbert

Kecskeméti Bolyai János Gimnázium,

6000 Kecskemét, Irinyi u. 49.

1. Bevezetés

Az Arduino nyílt forráskódú elektronikai platform, egyre nagyobb népszerűségnek örvend, mégis csak egy kis rétegét képes elérni az embereknek. A rendszer fő felhasználói és egyben fejlesztő gárdája a műszaki dolgok iránt nemcsak érdeklődő, hanem azt alkalmazni tudó diákok, kutatók, szakemberek és hobbisták. Nem ritka eset, hogy többen is megoldják ugyanazt a problémát, ugyanis ezek az emberek a világban szétszórva, egymástól fizikailag távol dolgoznak, alkotják meg saját projektjeiket. Ez körülményessé teszi a közös munkát, az együttműködések gyakran internetes fórumokra szorítkoznak. Ezeknek a társalgásoknak nagyobb része inkább a segítségkérésről, mintsem a saját kutató/kereső munka bemutatásáról szól. Amíg valaki nem tud Arduinot programozni, nem tudja, mit alkothat meg a segítségével, és emiatt nem is motivált abban, hogy elkezdje megismerni mélységeiben a rendszert. A LogB projekt egy keretrendszer segítségével teszi az Arduino alapú moduláris mérőeszközöket eddiginél sokkal több ember számára elérhetővé.

2. Probléma megoldásának menete

Sokan használják az Arduinot egyedi (saját igényekre szabott) adatgyűjtő eszköz megépítéséhez. Elterjedt módszer mások hasonló projektjeiből ihletet meríteni, de ennek feltétele az Arduino átfogó ismerete. Gyorsabb lenne a tervezés folyamata, ha mindenki egy közös keretrendszert fejlesztene. Így egy funkciót csak egyszer kell megírnia valakinek, és az is felhasználhatja, aki nem tudta volna azt a bizonyos dolgot leprogramozni. Ez a keretrendszer a LogB.

2.1. A standard

A LogB alapja a mérőeszközök működését egységesítő szabvány, LogB standard, ami meghatározza:

- a beérkező adat tulajdonságait,
- az adatok formázhatóságát,

a kimenet várható adatstruktúráját,
a mérés folyamatát.

Ez új lehetőségeket nyit meg. A standard révén egy LogB-s mérőeszköz pontosan leírható az előbb említett modulok és azok almoduljainak, beállításainak felsorolásával. Ezt a mérőeszközöket leíró adatstruktúrát **LogB Config**nak hívják. A moduláris konfiguráció struktúrát összekötve az LogB Arduinos kódjával kapható meg a **LogB KódGenerátor**, ami megoldást nyújt a programozásban nem jártas, de ötletekkel teli alkotóknak. A generátor képes előállítani a LogB Arduinos szoftverét. Nyilvánvaló jelentősége a funkciónak, hogy a LogB használatához nem feltétel az Arduinot programozni tudás. Így sokkal több ember számára elérhető, és azoknak sem kell foglalkozni a kreativitást gyakran nem igénylő munkafolyamattal, akik tudnának programozni. A felhasználók számára csak akkor jelenik meg temérdek új lehetőség, ha tudják, mi mindenre tudnak felhasználni egy ilyen mérőeszközt. A **ConfigGenerátor** segítségével ihletet szerezhetnek projektjeikhez.

Mivel a rendszer moduláris, az Arduinot ismerő fejlesztő lehet, hogy inkább a LogB segítségével készíti el a saját projektjét, mert így kevesebb idő befektetésével tud eljutni azonos szintre. Az elkészített funkciókat bárki hasznosíthatja a standard és a KódGenerátor segítségével.

2.2. Bővíthetőség

A rendszer moduláriságának bemutatására elkészítettük a LogB felhőalapú szolgáltatását. A LogB Cloud lehetőséget ad LogB-vel készült mérések internetre való feltöltésére, tárolására, illetve az adatok megtekintésére, feldolgozására. Ez egy ugyanolyan kimeneti opcióként jelenik meg a felhasználónak, mint a többi. A tárolt mérések alapvetően bárki számára elérhetőek, ezzel támogatva a közös kutatásokat, de lehetőség van priváttá tenni őket. A könnyű elérésük érdekében a felhasználó megoszthatja a mérés linkjét, vagy QR-kódot generálva beillesztheti például prezentációkba. Az adatok különféle módokon tekinthetők meg: Táblázatban, grafikonon, de akár json, vagy csv fájl formájában is exportálhatóak más programban való elemzéshez. Lehetőség van az adatok élő frissítésére is (Excelben jsonnal is) megtekintés közben. Ilyenkor az adatok a mérési időközöknek megfelelően frissülnek. A felhasználót segítjük az adatok feldolgozásában is:

Az adatok a fejléceknek megfelelően növekvő, illetve csökkenő sorrendbe rendezhetők.

Megjeleníthető a minimum, maximum érték, de lehetőség van átlag, szórás, medián számítására is.

A LogB projekt az itt felsorolt alapvető dolgokon kívül sok mindent tartalmaz:

A LogB nyílt forráskódú

A fiatalos lendületét a projektnek lendületes fiatalok hasonló attitűddel támogatják.

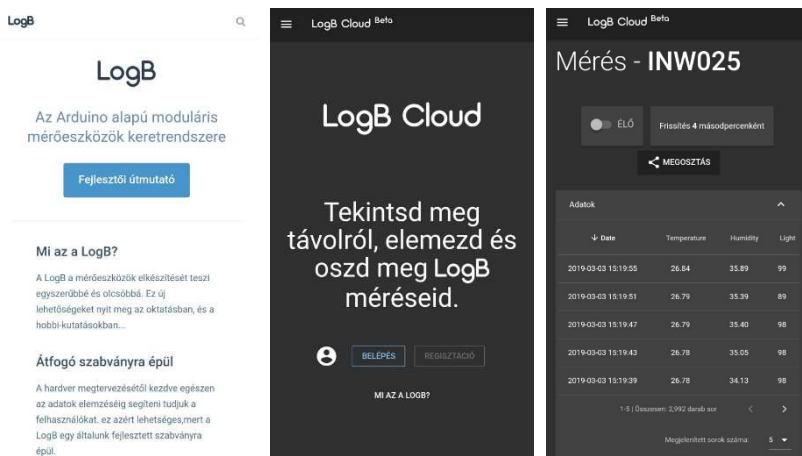
A LogB végső soron egyszerűvé és érdekessé szeretné tenni a természeti méréseket és a diákköri kutatómunkát.

Az oktatási szektornak szánjuk a LogB Maci-t, ami alatt a nagy példányszámban, professzionálisan előállított LogB alapokra épült, költséghatékony mérőeszközt értjük.

2.3. Ábrák



1. ábra: LogB.hu



2. ábra: A LogB, a LogB cloud weboldala, valamint az INW025 kódú mérés megtekintése

3. Elért eredmények

A LogB-vel elérhetjük, hogy az Arduino közösség a mérőeszközök terén kevesebb felesleges munkával többre jusson. Miközben saját problémáikat oldják meg, egy olyan közös értékét gyarapítanak, amivel fel lehet hívni az érdeklődő tömegek figyelmét. Könnyebben ki lehet bővíteni a természeti mérések jelenlegi körét. Akár egy felsős általános iskolás is használni tudja a LogB-t, miközben egy egyetemista is megtalálhatja számítását a projektben, ha természettudományos mérésekről van szó.

A jövőben - ahogy a neve is sugallja - azon fogunk dolgozni, hogy a LogB legyen az általunk fejlesztett standard.

Dallamcsengő

Bláthy42

Kőrösi Ádám, Dudás Miklós

Felkészítő tanár: Harangozó Zsolt

*BMSzC Bláthy Ottó Titusz Informatikai Szakgimnáziuma,
1032 Budapest, Bécsi út 134.*

1. Bevezetés

A BMSzC Bláthy Ottó Titusz Informatikai Szakgimnáziumának diákjai vagyunk és az iskola Arduino szakkörének (BOTDuino) tagjai. Adott volt egy probléma az épületben, a fizikai csengő nem volt mindenhol hallható a nagy alapterület miatt, azonban az iskolai bemondó az egész épületet lefedi. Korábban már próbálkoztak egy dallamcsengő kialakításával, azonban ez nem váltotta be a hozzáfűzött reményeket. Ehhez készítettek egy kapcsolást, aminek a lényege, hogy ha egy DIN-5-ös csatlakozón zárjuk a megfelelő pineket, akkor felkapcsolja az erősítőket. Ez azért kell, hogy ne legyenek az erősítők egész nap bekapcsolt állapotban. Kitaláltuk, hogy ezt a problémát egy saját vezérléssel fogjuk orvosolni.

2. Probléma megoldásának menete

A tervezés során Arduino, valamint Raspberry PI alapú megvalósítások jöttek szóba. Az Arduino alapú megoldásra tettük a voksot, mivel ez az eszköz hosszú távon, rendszergazdai beavatkozás nélkül stabilabbnak mondható.

A következő nagy kérdés a processzor volt. Először az Arduino Uno+Ethernet shield jutott az eszünkbe, azonban a nagy mérete és a nehézkes kábelezése miatt áttértünk az ESP 8266-ra. Ezen belül a Wemos D1 Mini mikrokontrollert alkalmaztuk. Az eszköz feladatai:

- Iskolai csengetési rend, rövidített tanítási csengetési rend kezelése
- jelző csengetés (3 perccel a csengő előtt)
- manuális tűzjelzés indítása
- automatikus idő szinkronizáció Network Time Protokollal

2.1. Az első konfiguráció

Az első működőképes konfiguráció 3 hét éles tesztüzemet teljesített a Bláthy épületében. Található rajta egy RTC modul, amit NTP-vel szinkronizálunk, az MP3 fájlok lejátszásáért pedig egy DF Player Mini felel, valamint az erősítők kapcsolásához természetesen egy relay. Az oldalára DIN-5-ös és jack csatlakozók kerültek. Azonban akadtak még hiányosságok. A

tűzjelző elindítása problémás lett volna, mivel az iskola stúdiójában volt bezárva. Az előbbieken kívül nem bírta a sok mozgatót a próbapaneles bekötés miatt. Az első működőképes konfigurációról készült fénykép az 1. ábrán látható.



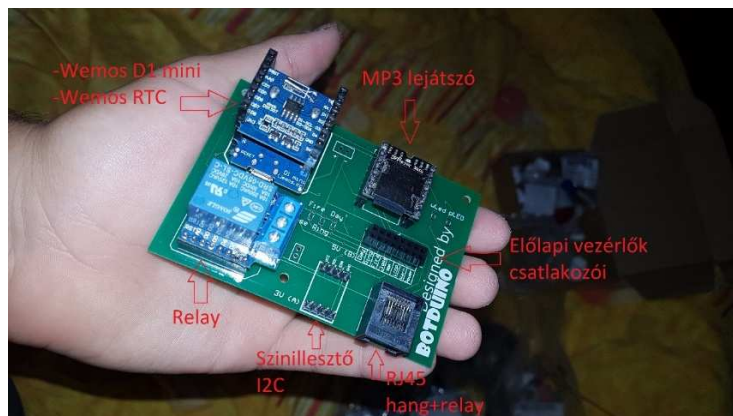
1. ábra: Az első működőképes konfiguráció

2.2. A mai konfiguráció

Az eszköz kapott egy saját tervezésű nyomtatott áramkört és egy új házat, amiről fényképet a 2. illetve a 3. ábrán láthatunk. A tűzjelző problémáját úgy oldottuk meg, hogy az erősítő relayt, stereo hangot ráraktuk RJ45-ös aljzatra. Az eszközt kihelyeztük a portára. A portán lévő eszköztől 40m UTP kábel vezet a stúdióig, ahol egy átalakító szedi szét az RCA-t és a DIN-5-öt. Az átalakítóról képet a 4. ábrán láthatunk. Az eszköz hálózati működését indításnál access point módban lehet konfigurálni böngészőből. A konstrukció jelenleg működőképes és a csengetésért felelős az iskolánkban, azonban, a hosszú távú üzemeltetésnél akadt egy probléma. Az az RTC modul amit használunk jelenleg, sajnos nem stabil, így napi 4 NTP szinkronizáció szükséges a pontos idő eléréséhez, emiatt a jelenlegi darab képes akár a napi 3-4 másodperc eltérésre is.



2. ábra: A jelenlegi ház



3. ábra: A jelenlegi nyomtatott áramkör



4. ábra: Az átalakító

2.3. Tervek a jövőre nézve

Szeretnénk ezt a konfigurációs módot kiterjeszteni, hogy belső hálózaton elérhető legyen az eszköz és a csengetési rendet be lehessen állítani (ami egyelőre sajnos kódszinten van tárolva), valamint telefonról is lehessen tűzjelzést indítani. Tervben van az RTC cseréje is a fentebb említett probléma miatt. A csere miatt a nyomtatott áramkört is újra kell tervezni.

3. Elért eredmények

Úgy gondoljuk, hogy nagyon büszkék lehetünk a csengőre. Adott volt egy probléma, amit sok tervezéssel, teszteléssel és hibakereséssel sikerült megoldanunk. Rengeteg szenzort, technológiát teszteltünk, amíg összeállt a jelen konfiguráció. Az RTC pontatlanságán kívül fél év hibamentes éles működés után kijelenthetjük, hogy bár még van hova fejlődni, de egy stabil konfigurációt raktunk össze. A projekt ugyan probléma specifikus projektnek indult, de a későbbi szoftverfrissítések során szeretnénk olyan irányba haladni, hogy az eszköz akár önálló termékként is megállja a helyét a piacon.

Az eszköz korábban egy régi szoftververzióval indult a 2018-as Működj! pályázaton. Azóta a csengő több szoftveres frissítést is kapott, megjelent a menü rendszer és a dinamikusan változtatható csengetési rend is.

Smart thermostat system

2_1337_4_U

Kiss Ádám, Tankó-Bartalis Bence

Felkészítő tanárok: Pákozdi Péter, Pirity Tamás Gábor

Eötvös Loránd Tudományegyetem Apáczai Csere János Gyakorló Gimnázium és

Kollégium,

1053 Budapest, Papnövelde utca 4-6.

1. Bevezetés

A mai fejlődő világban nagyon sok termék elérhető, amellyel könnyíthetünk az életünkön vagy „okosíthatjuk” a házunkat. A legtöbb ilyen berendezés viszont meglehetősen drága. Erre jó megoldásnak tűnhet, hogy nem egy teljesen új rendszert vásárolunk, hanem egy már meglévő rendszert bővítünk ki. Ilyenkor viszont azt tapasztalhatjuk, hogy vagy rosszul, vagy egyáltalán nem megoldható a régi és az új, „okos” rendszer összehangolt működtetése.

Otthonunkban pár éve beépítettünk egy fatüzelésű kályhát, a drágább üzemeltetésű, de addig egyszerűen szabályozható gázkazán mellé. Ezzel viszont az a probléma jelentkezett, hogy az egy (majd később 3) termosztát nem volt elég, hogy észleljék az egyenetlen fűtés miatt esetleg hűvösebb szobákat. Íme, egy leegyszerűsített példa: a nappaliban, ahol van termosztát, ég a kályha. Ekkor a gázfűtés kikapcsol, mert ott a szobahőmérséklet meghaladja a bekapcsolási küszöböt. Ilyenkor viszont a dolgozószobában, ami távolabb esik a nappaltól és ahol nincs termosztát, lehűl a levegő.

A radiátoraink önszabályozóak. Ez azt jelenti, hogy ha a szobahőmérséklet elért egy bizonyos szintet, akkor a radiátor nem fűt tovább. Ennek az az előnye, hogy nem kell minden radiátort külön szabályoznunk.

A házban már több éve voltak Raspberry Pi-k, amik egyéb főbb feladatokat mellett a csatlakoztatott hőmérőkkel folyamatosan mérték és rögzítették a hőmérsékletet. Ez adta az ötletet ahhoz, hogy saját fűtésvezérlő rendszert tervezzünk.

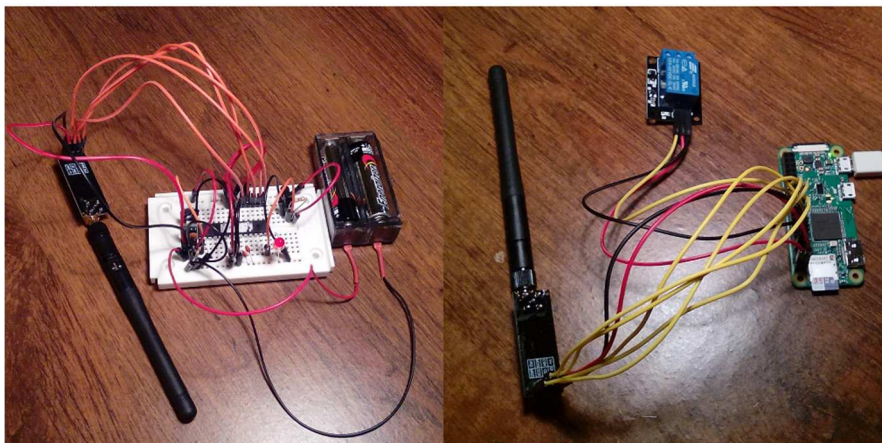
2. Probléma megoldásának menete

Egy olyan rendszert szerettünk volna létrehozni, amely dinamikusan bővíthető, és mivel ennyi egységet külön beállítani bonyolult lenne, kézenfekvő megoldás volt egy webes felületet készíteni, amin keresztül könnyen áttekinthető és beállítható minden. További fontos szempont volt,

hogy vezetékek nélkül kommunikáljanak egymással az egységek, ezzel kompaktabbá és rugalmasabbá téve a rendszert.

2.1. A hardware

A rendszer két fő részből áll. Egy központi vezérlőegységből és hőmérős modulokból. (1. ábra) Az előbbi rögzíti és feldolgozza a hőmérők által mért adatokat, kapcsolja a fűtést egy relay segítségével és szolgáltatja a webes felületet. Ezzel kommunikálnak a hőmérős modulok, amelyekhez egyenként akár 3 hőmérő csatlakoztatható.



1. ábra: balra a hőmérős egység prototípusa, jobbra a központi vezérlőegység

A központ egy Raspberry Pi Zero W, ami beépített Wi-Fi modullal rendelkezik, és a mérete is kicsi. Ezen fut az Apache webszerver, amin keresztül be lehet állítani a különböző hőmérsékletküszöböket, illetve ki- és bekapcsolhatjuk a fűtést.

Ehhez csatlakoznak a hőmérős modulok, amiket Atmega328P-Pu mikrokontrollerek vezérelnek. Azért ezeket választottuk, mert az áramfogyasztásuk rendkívül alacsonyra csökkenthető, tudják kezelni az összes csatlakoztatott hőmérőt és a vezetékek nélküli kommunikációs modult.

Az egyszerűség és a biztonság érdekében Enhanced ShockBurst™ protokollú, 2.4 GHz-es rádiómodult választottunk. Ez biztosítja, hogy a rendszernek ne legyen infrastruktúrai igénye, ami a Wi-Fi-vel nagyobb területeken probléma lenne.

2.2. A software

A rendszer megalkotásához több programozási nyelvet is használtunk. Az Atmega chipen az Arduino C alapú nyelven írt program percnként leolvassa a hőmérők által mért értékeket, és elküldi a központi egységnek. Itt egy Pythonban megírt program fogadja és fájlba elmenti a beérkező adatokat. A nyers adatokat egy Perl program alakítja át CSV formátumba, és közben kezeli az esetleges hibás, illetve hiányzó adatokat. Az első sor az időt, a többi pedig a különböző hőmérők által mért hőmérsékletek tartalmazza. Ezzel dolgozik a weboldalt kezelő Python program. Ez a program kezeli a dinamikus HTML oldalakat, amik megjelenítik az adatokat JavaScript és Jinja segítségével. (2. illetve 3. ábra)

A weboldal okostelefonról is kényelmesen használható. A beállítások megtekintéséhez, illetve módosításához jelszó szükséges, így illetéktelen személyek nem férhetnek hozzá.



2 ábra: Áttekintés bejelentkezés után

Smart thermostat home page Set temperature Log Out

#	Last data: 23:59	Min	Set Min	Set Name
Bedroom heating	Currently: off			New Name
Bedroom	22.2 °C	20.0 °C	16.5 °C	New Name

3. ábra: A fűtés beállítása

3. Elért eredmények

A célunk a hőmérők vezetékek nélkülivé tétele és a fűtés kapcsolásának megvalósítása volt. Ezeket sikeresen megvalósítottuk. A prototípus hibátlanul működött a tesztidőszak alatt. Íme, pár jelentősebb eredmény:

A fogyasztást annyira lecsökkentettük, hogy akár egy évet képes működni 2 AA elemről.

Olcsó építőelemeket használtunk

A hőmérő egységek végleges mérete 7x4x3.5 cm alatt lesz (Az opcionális antennát leszámítva)

A maximális hatótáv bőven meghaladja egy lakóház méreteit

A csatlakoztatott hőmérők száma korlátlan

A webes felület letisztult és könnyen kezelhető

A rendszer feldolgozza és tárolja az elmúlt két év alatt mért hőmérsékleteket.

4. További terveink

A rendszert folyamatosan fejlesztjük, célunk addig kibővíteni, amíg a legkorszerűbb okosotthonok funkcióit is el tudja látni. Először viszont a fűtéssel kezdtük, mert nálunk arra volt éppen igény. A fűtéshez még tervezzük a szabályozás kiterjesztését külön zónákra. Ez azt jelenti, hogy modernebb házakban, ahol a fűtés zónákra van bontva, a fűtést csak a fűtendő zónára kapcsolja fel. Tervezünk még padlófűtés szabályozást készíteni és különböző fűtési módszereket párhuzamosan kezelni, illetve a hőmérők üzemidejét növelni.

Hosszabb távon pedig még több funkciót szeretnénk hozzáadni, például a riasztó kezelését, redőnyök időzítését, ablakok nyitását-zárását, és egyéb fontos szenzorokat, például tűz-, füst- illetve szén-monoxid jelzőt. A további bővítéseknek csak a képzelet szabhat határt.

Underwater cyclops

Bánkirobot

Tóth Bence, Zsigó Miklós

Felkészítő tanár: Zsigó Zsolt

NySZC Bánki Donát Műszaki Középiskolája,

4400 Nyíregyháza Korányi F. 15.

Nyíregyházi Krúdy Gyula Gimnázium,

4400 Nyíregyháza Epreskert u. 15

1. Bevezetés

Az általunk fejlesztett víz alatti kutatórobot képes olyan tevékenységet is végezni, amelyekre egy bűvár nem lenne képes, és mindezt költséghatékonyan. A robot távirányítással végzi feladatát, előre tervezett útvonalon, előre tervezett feladatokat képes végrehajtani. Az elmúlt évben készített prototípus az Ifjúsági Innovációs Versenyen díjazásban részesült, az itt szerzett tapasztalatok alapján a robotot szenzorokkal egészítettük ki.

A kis ifjúsági fejlesztő csoport, amely Bánkirobot néven dolgozik évek óta, ezzel a kérdéssel kereste fel a városunkban tevékenykedő víziközmű szolgáltató cég szakembereit, keresve az együttműködés lehetőségeit. A szakemberek nagyon nyitottan hallgatták a víz alatti kutatórobottal kapcsolatos gondolatainkat, és közösen áttekintettük a mosatási és fertőtlenítési (MOSATFERTUT) munkautasítást, amely a tárolómedencékben végzett munkafolyamatokat szabályozza. A dokumentum áttanulmányozása utána készült egy lista, hogy mi mindenre használható a távirányítással mozgatható víz alatti kutatórobot, és hogyan lehet csökkenteni a vízveszteséget.

A víziközművek szerves részét képző víztároló medencék karbantartása a víziközmű szolgáltatók feladata. Ezt a feladatot a NYÍRSÉGVÍZ Zrt. évente két alkalommal éves karbantartási ütemterv keretén belül hajtja végre, a medencék leürítése utáni felület-tisztítási, szerelvény működőképesség vizsgálat, javítási feladatok elvégzésével.

A karbantartások hatékonyságának növelése érdekében fontos információk kaphatók, ha a karbantartást megelőzően fotó, illetve videó felvételek készülnek, amelyeket a szakemberek kiértékelnek. A Szolgáltató elvárása, hogy a víztároló medencék felesleges leürítése nélkül győződhessen meg a tároló felületének állapotáról. Ha mégis szükséges a javítás, akkor, az

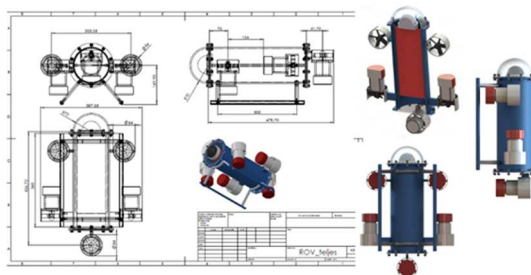
előzetes információk alapján, egy esetleges hosszabb időintervallumot jelentő javítás előkészítését megfelelő alapossággal tudja elvégezni a Szolgáltató.

A rendszeres vizsgálatok felvételeinek archiválásával lehetőség nyílik a korábbi időszakokban készült felvételekkel történő összehasonlításra, így az esetleges nagyobb javítások, beruházások tervezése költséghatékonyan oldható meg, a medencék várható élettartama is meghatározható.

Az előzetes becslések alapján ideális esetben így több ezer köbméter víz leürítése kerülhető el!

2. A fejlesztés menete

A fejlesztések fő részét ketten végeztük, de ha megakadtunk, akkor szakemberektől kértünk segítséget. A NYSZC Bánki Donát Műszaki Középiskolájának gépész és villamosmérnök tanárai hasznos tanácsokat adtak, és ha szükséges volt segítettek a robot alapelemeinek elkészítésében. Ennek balesetvédelmi okai is voltak, ugyanis mi önállóan nem üzemeltethetünk ilyen gépeket. A tervezés AutoDesk Inventorban történt, amelyben a gyártáshoz és a szereléshez szükséges rajzokat készítettük (1. ábra).



1. ábra: A rajz

A nyomtatott áramkörök tervezéséhez Eagle CAD szoftvert használtuk. A teszteléshez a víziközmű szolgáltató rendelkezésünkre bocsátott egy tartályt, amelyet csak kültéren tudtunk elhelyezni, és a nagyon rossz időjárási viszonyok miatt kevés tesztidőnk volt. Egy alkalommal tudtunk üzemi körülmények között tesztelni, erre a későbbiekben kapunk még lehetőséget.

Szerkezet felépítése

Géptest, hajtómű tartóval:

4mm vastag alumínium vagy POM műanyag, átmérője 140mm hossza 370mm. Hajtómű tartók 10mm-es alumínium vagy POM anyag, csavarkötéssel a géptesthez csatlakoztatva. A hajtómű tartók 3D

nyomtatottak. 10-10 db csavarkötés a hát és homlokfalon végzáróval. M6 csavar és önzáró anya az összeszorításhoz. Géptestben elhelyezett sínbe lehet helyezni a vezérlő panelt, ami könnyedén kivehető, megkönnyítve a szerelést. A sín, a vezérlő panel tálca, és a kamera ablak (kupola) szintén 3D nyomtatás segítségével készült.

Vízhatlanítás

- A kamera ablak és a végelzáró 3mm vastag O gyűrűvel van tömítve
- A motorok gyári kialakításuk miatt vízhatlanok
- A testen lévő furatok szaniter szilikonnal lettek kitömítve

Elektronika

Motorok és a motorvezérlő:

5 fenékszivattyúból átalakított motor végzi a test mozgatását. ezek közül 3 a fel-le és 2 az előre-hátra mozgásért, valamint a fordulásért felelős. A motorok áramfelvétele terheletlen állapotban 5-6A. Üzemi feszültségük 12V. Az átszívott vízmennyiség 1100GPH. Minden motorvezérlőhöz 2 motor csatlakozik, a motorvezérlőket megfelelő módon lehet PWM, valamint digitális I/O jellel vezérelni. A motorvezérlő FET-jei teljes H hídba vannak kapcsolva.

Világítás:

A világításért 6 db 1.7W-os 12V-os LED felelős. Fényárama 1000 lm.

Tápellátás:

Az eszköz tápellátásáért 12 db Li-Ion ipari akkumulátor cella felelős.

Az akkumulátorok négyesével vannak sorba kapcsolva. A 4 sorba kapcsolt pakk pedig párhuzamosan van kötve. Így az összkapacitás 7500mAh.

Kamera:

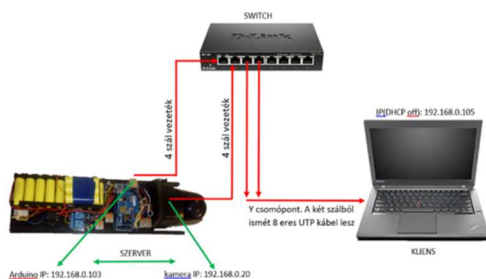
Az eszközön 2 db kamera kapott helyet. Az első egy real-time IP kamera mely a tájékozódásért felelős. Az IP kamera 360 fokban forgatható. Azonban a víztározókban szinte alig van fény, így szükséges volt külső megvilágítást is alkalmazni. A kamera felbontása 480p. A második egy akciókamera.

Vezérlés

Irányítás:

Az irányítás egy joystick segítségével történik, amelyet a laptop-hoz kell csatlakoztatni. A kommunikáció vezetékesen történik. Egy 8 eres UTP kábelon keresztül zajlik az irányítás, mely a vezérlő panelen 4-4 vezetékre van osztva így lehetővé téve a kamera és az Arduino egyidejű használatát. Az eszköz offline környezetben is irányítható, valamint a kamera kép is elérhető.

Így egy hálózati switchre van szükségünk, hogy „újraegyesítsük” a szétválasztott vezetékeket (2. ábra).



2. ábra: Az elvi felépítés

Az eszköz programozásához kétféle nyelvet használtunk. A vezérlő panelen található Arduino MEGA-t C++ nyelven, míg a PC-n található felület Python nyelven íródott. A Python kód felelős a joystick kezeléséért, valamint a vezetékcsatlakozás létrehozásáért, és a ROV-től érkező adatok (szenzor értékek) feldolgozásáért. Az adatokat egy .csv állományba mentjük, amely könnyen kiértékelhető egy grafikon készítésével a kapott adatok alapján.

Alkalmazott szenzorok

Az eszközön jelenleg megtalálható egy DS18B20 digitális hőmérsékletmérő szenzor, mely a víz hőmérsékletét méri, és egy fényerősség mérő cső is, mely egy konstans fényforrást és egy fényellenállást tartalmaz.

3. Elért eredmények

A NYÍRSÉGVÍZ Zrt. mindösszesen 37 850 m³ ivóvíztároló medencét üzemeltet. Üzemi körülmények közötti, 90 %-os töltöttség melletti vizsgálat esetén ~ 34 m³ ivóvíz szükségtelen kieresztését takaríthatjuk meg. Ez a mennyiség kb. egy 2 000 lelkes település egész éves vízfelhasználást jelenti. Országos viszonylatban a tárolómedencék tárolókapacitása 950 000 m³. A KSH szerint az egy főre jutó éves vízfogyasztás hazánkban 34 m³ (2015), így az eljárás alkalmazásával egy 30 000 fős kisváros (Szekszárd, Gödöllő, Orosháza) éves vízfogyasztása takarítható meg.

Az említett víziközmű szolgáltatási területén bevezetve a robot használatát, több mint 6 millió forint megtakarítást - tárolónként kb. 100 ezer Ft forint - eredményez. Természetesen a fel nem használt elektromos energia, és a dolgozók munkaideje is része a megtakarításnak. A robot fogókarral, mintavételező egységgel felszerelve a kutatók, barlangi bűvárok, vagy akár a katasztrófavédelemben dolgozók munkáját is segítheti.

Kulcsautomata

Berendi Bence

Felkészítő tanár: Hegedüs Tamás

Tatabányai SZC Széchenyi István Közgazdasági és Informatikai

Szágimnáziuma,

2900 Komárom, Táncsics Mihály utca 75.

1. Bevezetés

A kiindulási ötletet egy, az iskolánkban régóta fennálló probléma adta. A különböző kulcsokat a portás adta ki, de néha előfordult, hogy nem tudták kinél van a kulcs, illetve nem találták a portást. Célnak egy olyan eszköz megépítését tűztem ki, amely képes kezelni a kulcsok kiadását és visszavételét, valamint ezeket rögzíteni is tudja. Előzetes kutatásaim alapján, több területen is lenne igény ilyen eszközre, például építőiparban, illetve irodaházakban.

2. Probléma megoldásának menete

Fontosnak tartottam, hogy a szerkezet megépítése közben minél kevesebb problémával kerüljek szembe, ezért a megvalósítást hosszú tervezés előzte meg, melyet több részre bontottam.

2.1. A szerkezet megtervezése és kivitelezése

A kulcsok elhelyezését egy forgódobon oldottam meg. Mivel ez nevezhető a projekt lelkének, ezért fontos volt, hogy stabil legyen és a forgórész minél könnyebb legyen. A forgódobot PVC csőből és rétegelt lemezből készítettem. A könnyű forgatás és a stabilitás miatt, a forgódobot egy vas keretben helyeztem el, melyen kialakítottam egy felső függesztett golyóscsapágyat, amin lóg a dob, alul pedig egy siklócsapágyat, ami támasztja a tengelyt, és így csökkenti a megfeszülés esélyét. A váz és a forgódob az 1. ábrán látható.



1. ábra: A váz és a forgódob

2.2. A mozgatás

A forgatást egy bipoláris léptetőmotor végzi, melyet egy A4988-as motorvezérlő hajt meg. Mivel az indításhoz nagyobb nyomatékra van szükség, ezért hogy ne terheljem a motort, készítettem bele egy 3d nyomtatott áttételt. A kiadó ablakhoz készítettem egy ajtót, amelyet egy SG90-es mikro szervó nyit. Ezt későbbiekben tervezem egy fogasléces mechanikára cserélni a nagyobb biztonság érdekében. Az ajtónyitó mechanika és az áttétel a 2. ábrán láthatóak. Egy esetleges áramszünettel számolva elhelyeztem egy végállás kapcsolót, és ennek segítségével alappozícióba tudja állítani a dobot induláskor a mikrokontroller.



2. ábra: Az ajtónyitó mechanika és az áttétel

2.3. A kulcsok érzékelése

A kulcsok érzékelésének megoldására szükség volt, hiszen tudnia kell a mikrokontrollernek, hogy ha például az ötös számú kulcs nincs az automatában, akkor ne akarja kiadni, hanem ismerje fel, hogy már korábban kiadta. Ennek megoldására infra szenzorokat építettem be a forgódobba minden kulcs mögé, amely a kibocsájtott infrasugár visszaverődésének idejéből határozza meg, hogy bent van-e a kulcs.

2.4. Kommunikáció

Az egyszerű információközlés végett a mikrokontrollerhez egy LCD kijelzőt csatlakoztattam, ami I2C-n kommunikál a mikrokontrollerrel. A felhasználó részéről több beviteli módot alkalmaztam. A felhasználók bejelentkezéséhez az RFID technológiát használtam, mivel így könnyen meg lehet különböztetni a felhasználókat a kártyájuk egyedi azonosítója alapján. A kártyák leolvasásához egy egyszerű MFRC-522-es modult választottam. A további adatbevitelhez egy 4x4-es membránbillentyűzetet helyeztem el az automatában. Az RFID olvasó, a billentyűzet, és az infra szenzor a 3. ábrán látható.



3. ábra: Az RFID olvasó, a billentyűzet, és az infra szenzor

2.5. A logolás

Fontosnak tartottam, hogy a kulcsok visszamenően nyomon követhetőek legyenek. Ezt a problémát egy számítógépen futó SQL adatbázissal oldottam meg, és kulcskiadásakor elküldi a mikrokontroller PHP scriptek segítségével, hogy melyik kulcsot ki- és mikor vitte el.

2.6. A feladathoz megfelelő mikrokontroller kiválasztása

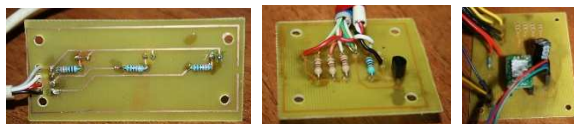
Elsődleges szempont volt, hogy megfelelő mennyiségű digitális be- és kimenet rendelkezésre álljon ahhoz, hogy minden eszközt megbízhatóan kezelni tudjon. Másodlagos szempont volt, hogy minél olcsóbb legyen a mikrokontroller, hogy az esetleges előállítási költség minél kedvezőbb legyen. Ezeket szem előtt tartva a választásom egy Arduino Mega-ra esett. Az Arduino Mega 54 darab digitális I/O-val rendelkezik, ami a későbbi fejlesztések kiszolgálásához is elegendő. További hasznos funkciója, hogy hardveres I2C buszrendszerrel rendelkezik. Mivel a logolást hálózaton végzi, ezért az Arduinohoz csatlakoztattam egy Ethernet shieldet. Az Arduino Mega és az Ethernet shield a 4. ábrán látható.



4. ábra: Az Arduino Mega és az Ethernet shield

2.7. A kiegészítő áramkörök

Az átlátható kábelezés, és a biztos kötések miatt több nyomtatott áramkört is készítettem az automatához, például egy tranzisztoros LED vezérlést. A nyomtatott áramkörök az 5. ábrán láthatóak.



5. ábra: A nyomtatott áramkörök

2.8. A program megírása

A programot az Arduino saját, C++-ra hasonlító nyelvén írtam. A program megírása közben több header fájlt is használtam, konkrétan: SPI.h; MFRC522.h; Ethernet.h; Keypad.h; Wire.h; LiquidCrystal_I2C.h; Servo.h.

3. Elért eredmények

A kivitelezés során több munkaterületen alkalmaztam tudásomat, illetve fejlesztettem is azt. Szükség volt CAD tervezésre a 3D nyomtatott alkatrészek megtervezésénél, fémmegmunkálásra a váz kivitelezésénél, faipari munkákra a szerkezet házának megépítésénél, valamint elektronikai ismeretekre a panelek és a kapcsolások tervezésénél valamint létrehozásánál. Sikerült elkészíteni egy olyan gépet, amely működik, és bejelentkezés után kezeli a kulcsok kiadását, visszavételét, valamint logolja is ezeket egy SQL adatbázisba. Az automata az 6. ábrán látható. A továbbiakban tervezek még pár apróbb módosítást a még nagyobb biztonság érdekében.



6. ábra: Az elkészült kulcsautomata

Biodaq 2.0 - Biológiai mérőlabor középiskolásoknak

Bánkirobot EMG

Vas Bertalan

Felkészítő tanár: Zsigó Zsolt

*NySZC Bánki Donát Műszaki Középiskolája,
4400 Nyíregyháza Korányi F. 15.*

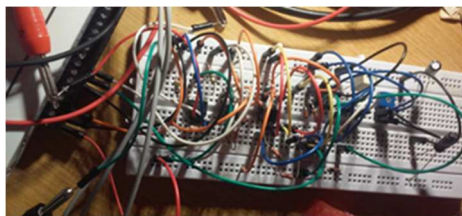
1. Bevezetés

Egy olyan mérés adatgyűjtő-elemző rendszert és a hozzá tartozó szoftvert valósítottam meg, - egy MyDAQ eszköz felhasználásával -, amivel ki lehet bővíteni a középiskolai biológiai kísérleteit, és lehetősége van a diákoknak sok emelt szintű biológia kísérlet elvégzésére is.

Az ELTE Neurológiai Tanszékén kipróbáltam egy Biopac nevű rendszert. A rendszer része egy hazugságvizsgáló, és a működése lenyűgöző. A Biopac meglehetősen drága, az ott látott rendszer értéke kb. 3 millió forint. A saját készítésű szenzorjaimat kibővítettem a National Instrumentstól kapott Iworx szenzorokkal, így tovább bővült a mérési lehetőségek száma. Az EMG és EKG mérést fogom részletesebben is bemutatni, mivel azokhoz saját készítésű szenzorokat használtam. A mérésekhez szükséges szoftvert Labviewban fejlesztettem. Az elmúlt évben az Ifjúsági Innovációs Versenyen egy korábbi verzió dicséretben részesült.

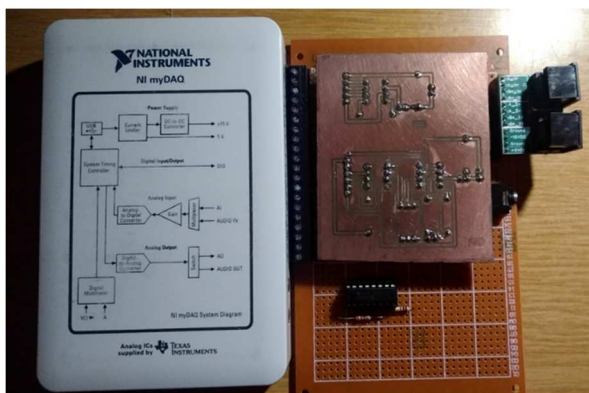
2. A fejlesztés menete

Az EMG méréshez egy ina106-os és egy TL072-es műveleti erősítőt használtam (1. ábra). Ez az EMG frekvenciáján körülbelül 1000-szeresére erősíti fel a jelet. Az így kapott felerősített jel nagyon hasonlít a Biopackal és már eszközökkel mért jelhez.



1. ábra: Az áramkör a próbapanelen

Az EKG szenzorhoz egy ina126p típusú precíziós műveleti erősítőt használtam. Ez a szenzor 250-szeresére erősíti fel a bemenő jeleket. (2. ábra)



2. ábra: Az első verzió a myDAQ-kal

Labviewban végeztem további szűréseket. Ezek után a szűrések és zajmentesítések után meglepően tiszta jelet kaptam a saját EKG jelemről.

2.1. EMG (electromyography)

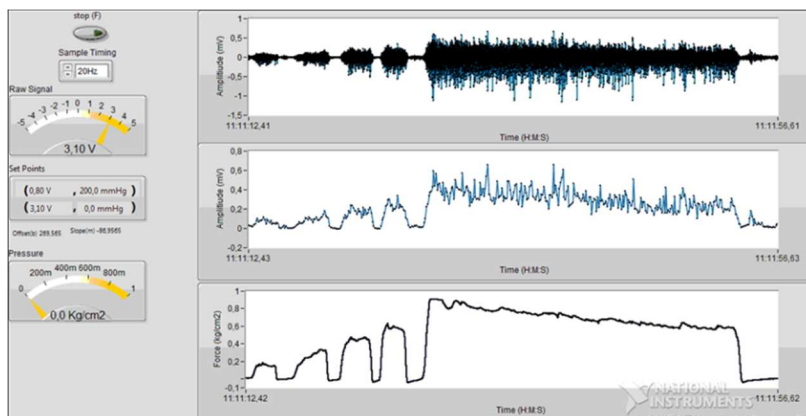
EMG-vel történő izomfáradás mérése:

„Az izmokban lezajló folyamatok nyomon követésének egyik módja az izmok elektromos aktivitásának mérése. Az elektromiográfia (EMG) az izmokban lezajló elektromos tevékenységet regisztrálja. A test térfogati vezetőképessége miatt az izomrostok plazmamembránján lejátszódó töltésmozgások az izomrostok közül és az izmok feletti testfelszínről is elvezethetőek. A bőrfelszínre ragasztott felszíni elektródák segítségével sok motoros egység együttes aktivitását vezethetjük el. A felvett EMG jelek egyenirányításával nyert integrált EMG jel az izom-összehúzódáskor kifejtett erővel arányos és a gyakorlatban is jól használható paraméter. Az integrált EMG jel amplitúdója végső soron arányos az aktivált motoros egységek számával. Az izomerő fokozásakor egyre több motoros egység aktiválódik (toborzás), így az EMG jel amplitúdója is nő. Az amplitúdó növekedésének mértéke az izom edzettségétől függ. Edzés hatására a motoros egységek száma nem, csak az izom térfogata változik. Ez azt eredményezi, hogy egy edzett személy esetén ugyanakkora erő kifejtéséhez kevesebb izomrost, kevesebb motoros egység aktiválása is elegendő, mint a kevésbe edzett személyben, ezért a mért EMG jel kisebb lesz.”¹

A szakirodalomban talált eljárás a következő: „A vizsgálatokat elsőként a domináns karon végezzük el. A Record gomb megnyomását követően fejszünk

¹ http://detari.web.elte.hu/gyak_jegyzet/emg.pdf

ki a dinamométerre egyre nagyobb erőt, úgy hogy az a képernyőn 5-10-15-20-25-30-40-50 Kg-nak feleljen meg. Az egyes erő kifejtéseknél 2-3 s-ig tartssuk meg a szorítást, majd 2-3 s szünet következzen”.



3. ábra: A mérés eredménye;a grafikonok

2.2. Iworx szenzorok alkalmazása

A BioDAQ-hoz felhasználtam a saját szenzorjaim mellett az Iworx biológiai szenzorcsomagot is. Ennek a szenzorcsomagnak köszönhetően lehetőségem volt EMG jel mellett szorítóerőt is mérnem. Ehhez a kézi dynamometert használtam. A szenzorcsomagban található egy vérnyomás mérő, pulzusmérő, spirometer, hőmérséklet mérő és az ezekhez a szenzorokhoz tartozó adapterek amik segítségével lehet a szenzorokat csatlakoztatni MyDAQ-hoz és NI ELVIS-hez is. Ezekhez a szenzorokhoz tartozik egy-egy bemutató VI is.

Az iworx szenzorok használatával kibővültek a mérési lehetőségek:

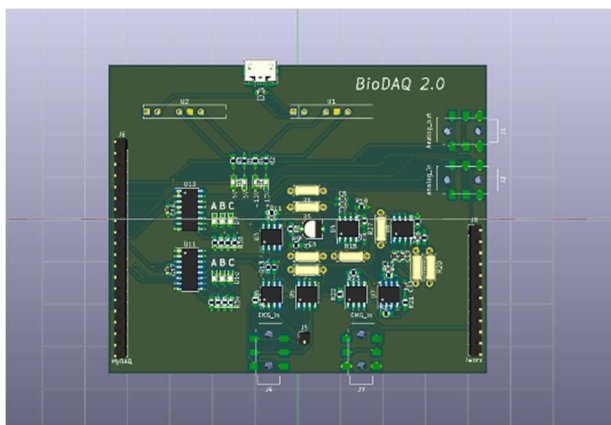
Vérnyomásmérés; Pulzus mérése; Spirometer - tüdőkapacitás mérése; Hőmérsékletmérés; **Szorítóerő mérés**



4. ábra: Az iworks szenzorok

3. Elért eredmények

Összefoglalva sikerült mindent megcsinálni, amit eddig terveztem. Jól működnek a saját fejlesztésű szenzorjaim, és az ezekhez tartozó VI is jól működik. Elkészült az új nyomtatott áramkör is, amelyben az előző változathoz képest a két szenzort egy áramkörbe sikerült integrálni.



5. ábra: Az új áramkör

Az eszköz lehetőségeit bemutató kisfilm:

<https://www.youtube.com/watch?v=L8KmKTs2nk4>

EsőCseppek

SopRobotics

Bejő Mátyás, Molnár Barnabás, Sztojka Áron

Felkészítő tanár: Lang Ágota

Soproni Széchenyi István Gimnázium,

9400 Sopron Templom u. 26

1. Bevezetés

A meteoritok rengeteg érdekes és hasznos információt tartalmaznak, akár a Föld, vagy az élet keletkezésére vonatkozóan, azonban a vizsgálathoz először meg kell őket találni. Ez nem egyszerű feladat, mivel ezek csak a légkör felső rétegeiben izzanak, és ezután nem áll rendelkezésre semmilyen információ a hollétükre vonatkozóan. Projektünkkel a meteorok esésének ezt a szakaszát, az úgynevezett sötétrepülést szimuláljuk, és erről szeretnénk minél több információt beszerezni, különös tekintettel a pillanatnyi helyzetére, sebességére, stb.

A meteorok egy része kis mérete, vagy porózus szerkezete miatt elég, mielőtt elérné a süllyedési végsebességét, míg mások egészen a földre csapódásig izzanak. A becsapódó meteorok többsége ezen kategóriák egyikébe sem tartozik. Miután elérték a légkört, felizzanak és folyamatosan lassulnak. Rövid idő elmúltával a sebességük annyira lecsökken, hogy a súrlódás nem izzítja őket tovább. Miután elérték a süllyedési végsebességüket, pontosan úgy viselkednek, mint az EsőCseppek, így a meteorkamerás felvételek, az időjárási viszonyok és a cseppeink mérései alapján lehet következtetni a becsapódásuk helyére. Az eszközeinket egy meteorológiai ballonnal juttatjuk fel közel 40 km magasságba, ahonnan leejtve őket nyomon követjük az esési pályájukat. Az erre alkalmas elektronikát különböző átmérőjű és anyagú (és ebből következően különböző sűrűségű) gömb alakú házba szereljük, ezzel modellezve a különböző összetételű meteorokat. Ők lesznek tehát a(z le)eső cseppek, innentől EsőCseppek. A méretnek fontos szerepe van, a minimalizálásra kell törekednünk.

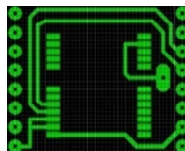
A projekt a Soprobotics és az SZTE TTIK Bajai Obszervatóriumának együttműködésének egyik eredménye, ugyanis az alapötlet Hegedűs Tibor Tanár Úrtól, az obszervatórium igazgatójától származik. A ballonos repüléseket szintén a Bajaobs ballonos csapata koordinálja.

2. Az eszköz felépítése

Az eszközt egy Wemos D1 mini Pro vezérli (1. ábra). Ez egy ESP8266 alapú WI-FI lehetőségekkel rendelkező mikrokontroller, amit többek között Arduino IDE-ben lehet programozni. A GPS kiválasztása nem volt egyszerű feladat. Az első chip, amit használtunk, egy GPS03 volt, amely 18 000 m magasságig tudott csak mérni. Mivel az EsőCseppeket ennél magasabbról szeretnék ledobni a ballonból, ezért be kellett ruháznunk egy teljesen új típusú GPS-re, amely nagyobb magasságokban is tud adatot szolgáltatni. Ezért esett a választásunk az ublox NEO-M8Q típusú GPS modulra (1. ábra). Ez egy fejlett modul a hihetetlenül kis mérete mellett sok más hasznos tulajdonsággal is rendelkezik. Nem csak az amerikai GPS műholdakat támogatja, hanem a BEIDOU-t, GLONASS-t, sőt még az európai GALILEO műholdakat is, így még pontosabb és gyorsabb helymeghatározást biztosít. Továbbá 4 különböző csatlakozási lehetőség áll a rendelkezésünkre, hiszen található rajta soros port, I²C port, SPI portok valamint USB port is. Üzem feszültsége 3,3V hideg indítással 1 percen belül képes bemérni a helyzetét.



1. ábra: ublox NEO-M8Q GPS



2. ábra: Számítógépes terv

Apró mérete miatt egy saját tervezésű nyomtatott áramkörre volt szükség, hogy a Wemos tűskéire közvetlenül lehessen ráhelyezni a kisebb helyfoglalás érdekében. Az áramkört Sprinter Layout 5 programban terveztük meg (2. ábra) majd kinyomtattuk. A vas-kloridos maratás után az első prototípuson ellenőriztük, hogy megfelelőek - e méretei. Később jobb minőségben is elkészítettük őket.



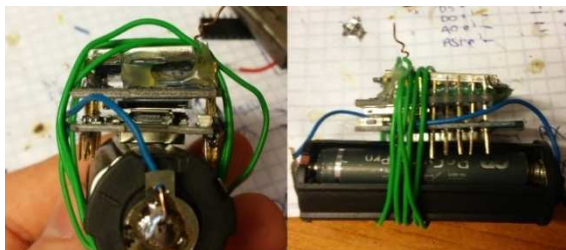
3. ábra: Kész GPS panel

A GPS modult végül egy hőlégfúvó segítségével forrasztottuk be az áramkörbe. Az eredmény a 3. ábrán látható. Az adatok tárolását SD kártyával oldottuk meg, ezért a panel szendvicsben helyet kapott még egy Wemosra illeszkedő SD kártya shield. Áramforrásként egy 3,6V-os zsák akkumulátort használunk, ennek kiválasztásának oka elsősorban a kis méret valamint a nagy kapacitás volt.

3. A program

A cseppeket az Arduino IDE 1.8.7-es verziójában programoztuk. A futó program a GPS-ből érkező adatokat (érzékelte műholdak száma, földrajzi szélesség és hosszúság, magasság, óra, perc, másodperc) eltárolja kettő txt fájlbn. (A két fájlra azért van szükség, mert leállításkor megvan az esélye, hogy az egyik éppen nyitva van és megsérül, de a másik fájl ekkor is olvasható marad.) Valószínű, hogy földre érés után külső tényezők miatt nem látjuk a cseppeket. Ezen úgy próbálunk segíteni, hogy a cseppek kommunikálnak egymással. Ehhez használjuk az ESP8266os wifi modult. Első tervünk egy MESH hálózat kialakítása volt, de ehhez a wifi modulok teljesítménye kevésnek bizonyult. Jelenlegi tervünk egy Master-Slave rendszer kialakítása, melyben lenne egy „server” csepp és több „client” csepp. A rendszer célja hogy a „server” cseppben az összes „client” csepp által mért adat megtalálható legyen, így ha az egyik „client” csepp eltűnne, akkor a „server” cseppben az általa mért adatok is megtalálhatóak lennének.

4. Az ideáig vezető út



4. ábra: Csepp 1.0

Csepp 1.0

Az 1.0 verzió (4. ábra) volt első működő eszközünk, amely csak az elv és az eszközök kompatibilitásának tesztelésére épült. Az energiaellátásról három AAA ceruzaelem gondoskodott, a mérést egy XN934C GPS modul végezte. Csak 18000 m magasságig működik, eddig két repülésen teszteltük.



5. ábra: Csepp 1.1 zsákakuval, antennával

Csepp 1.1

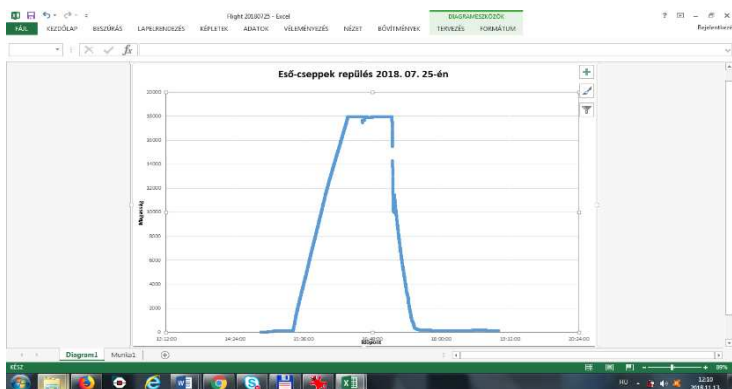
Ebben a verzióban (5. ábra) lecseréltük az elemeket egy megbízhatóbb és kompaktabb zsákakkura, és elláttuk a mikrokontrollert egy külső WIFI antennával.

Csepp 2.1

Miután bebizonyosodott, hogy a XN934C GPS nem működik nagy magasságokban, és ez a probléma nem oldható meg szoftveresen, áttértünk az új GPS modulra (ublox NEO-M8Q). Ezt egy saját tervezésű nyomtatott áramköri panellel szilárdan forrasztottuk a többi alkatrészhez, ahogy ezt a 2-es fejezetben taglaltuk.

5. Eredményeink

Ahogy fentebb említettük, a cseppjeink már kétszer repültek. A második indításában, majd a leejtőernyőzött gondola keresésében magunk is részt vettünk. A három cseppből az egyik működött csak kifogástalanul, és szépen rögzítette a magasságs adatokat pontosan 18000 méterig. Ennél megállt, és csak akkor kezdett ismét adatokat adni, amikor már lefelé zuhant a gondolával, ahogy ez a 6. ábrán látható. A grafikon a txt fájlban rögzített adatok alapján készült.



6. ábra: Grafikon

Every Breath You Take

Bánkirobot_Smog

Halász Tony Wang

Felkészítő tanár: Zsigó Zsolt

*NySZC Bánki Donát Műszaki Középiskolája,
4400 Nyíregyháza Korányi F. 15.*

1. Bevezetés

A légszennyezés hazánkban, és különösen városunkban, Nyíregyházán az utóbbi években fokozatosan vált égető problémává. Különös figyelmet érdemel ez a terület, hiszen ha hosszabb ideig lélegezzük magas koncentrációban a szennyezett levegőt, akkor ez nagy veszélyt hordoz. Köztudott, hogy a cigarettafüst után a légszennyezettség az oka a tüdődaganatok kialakulásának.

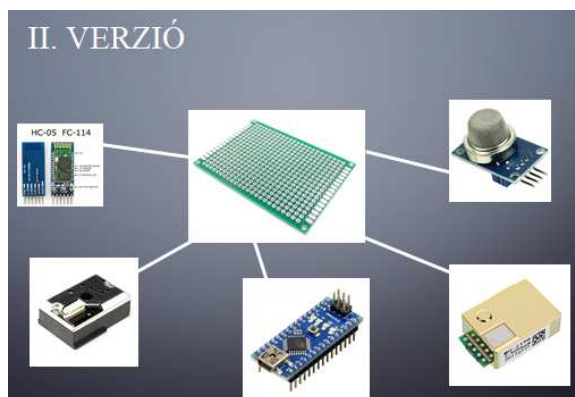
Nyíregyháza város levegőminőségét a Széna téren üzemelő automata mérőállomás a nap 24 órájában folyamatosan figyeli. A levegőminőséggel kapcsolatos hivatalos mérési adatok a www.levegominoseg.hu internetes oldalon tekinthetők meg.

A PM10 légköri aeroszol/ szálló por mennyisége bizonyítékot szolgáltat arra vonatkozóan, hogy felelős az emberek egészségügyi problémáiért. A PM, a radon és a CO₂ helyi értékeinek időbeli, és térbeli eloszlásának meghatározása kulcsfontosságú mind a betegségek megelőzése és megfigyelése, mind a városi közlekedés tervezésének segítése érdekében, így nagyobb figyelmet lehet fordítani a közegészségügyi kérdésekre és a környezettudatosság fejlesztésére is.

2. A fejlesztés menete

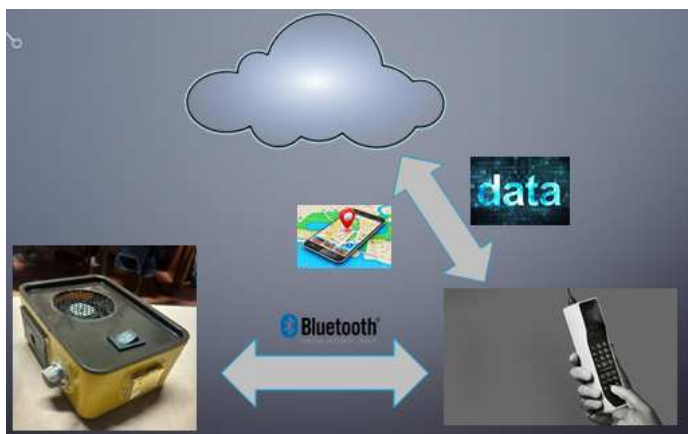
A projektem célja egy olcsó eszköz kifejlesztése, és megépítése a PM koncentrációjának mérésére úgy, hogy az adatok könnyen összehasonlíthatók legyenek máshol mért hasonló adatokkal, ezért a mérési folyamat pontos protokoll szerint történik. A mikrokontrollert tartalmazó dobozt a mobiltelefonokhoz kell csatlakoztatni, és bluetoothon keresztül az adatok a telefonon kerülnek rögzítésre. A szenzorok és a mikrokontrollerek kiválasztása több lépcsőben történt. Először az ESP32-t választottam, de sajnos olyan problémák léptek fel, hogy az Arduinot használnom helyette.

Az Arduino bluetoothon keresztül küldi a telefonra egyszerre több szenzor adatait.



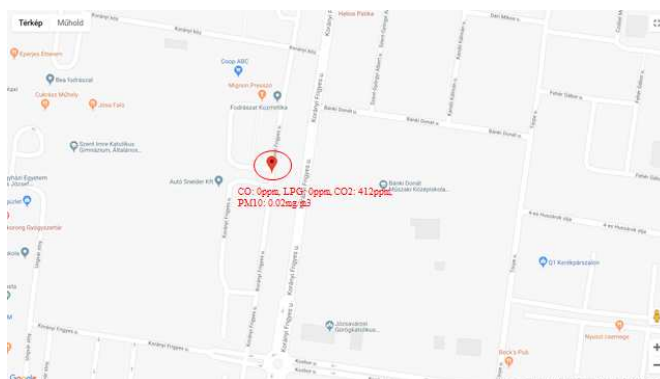
1. ábra: Az elvi felépítés

A telefon az adatokat szenzoronként szétválogatja és eltárolja, igény szerint megjeleníti.



2. ábra: Az adatok útja

Minden egyes telefon interneten feltölti az adatait egy felhőtárhelyre, ahol további feldolgozás lehetséges. Hozzáadja a GPS koordinátákat és az időbélyeget, hogy térkép-szerűen lehessen pl. a szálló por koncentrációját megjeleníteni.



3. ábra: Az adatok megjelenítése térképen

A méréshez a Sharp GP2Y1010AU0F (szálló por és füstérzékelő) optikai szenzort szeretném használni, de a doboz kialakítása lehetővé teszi, hogy a CO_2 és MQ-xM^3 szenzorok megfelelő légáramlást kapjanak, a hőmérő is el legyen különítve a belső melegedő alkatrészekről.



4. ábra: A mérésre kész eszköz

3. Elért eredmények

A szenzorok és a mikrokontroller integrálása sikerült. A mikrokontroller, és a telefon között működik a Bluetooth kommunikáció, és a telefonos applikáció, amit írtam kezeli az adatokat, egy adatbázisban tárolja őket, és a térképen is megjeleníti. Az eszköz működését teszteltem, és mérési adatokkal szeretném feltölteni az adatbázist.

² https://www.hestore.hu/prod_10038169.html

³ https://www.hestore.hu/prod_10035584.html

ToronyMentő Dőlésgátló

Felső ToronyHáz

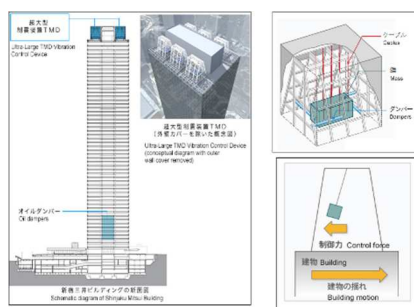
Horváth Boldizsár, Kenessei Zsombor, Róth Gergely

Felkészítő tanár: Lang Ágota

*Berzsenyi Dániel Evangélikus Gimnázium,
9401 Sopron, Széchenyi tér 11.*

1. Bevezetés

A múlt tanév végén akkori fizikatanárunk mutatott egy videót, amelyen a Taipei 101 – a világ 5. legmagasabb épülete a maga 509 méterével – visszaverte egy földrengés támadásait a benne található hatalmas inga segítségével. A film felkeltette az érdeklődésünket. Egyrészt utánanéztünk, hogy ez más magas épületekben is létezik-e; illetve, mivel mindannyian az iskola programozó szakkör tagjai vagyunk, úgy döntöttünk, hogy magasabb szintre lépünk és megpróbálkozunk egy mikrokontroller programozásával. Célunk egy modell megépítése volt, amelyben a TMD technikát alkalmazzuk, amelynek magyarul a ToronyMentő Dőlésgátló elnevezést adtuk.



1. ábra: Shinjuku Mitsui Building

A szerkezet, amelyet modellezni kívánunk, elsősorban magas épületeket óv meg az összeomlástól pl. erős szélben vagy földrengéskor, amikor a torony rezgésbe jön és kileng. Az eljárás angol neve Tuned Mass Damper, a magyar szakszó pedig tömeghangolt csillapítás.

Működésének elvét egy másik épületen, mégpedig a Tokióban emelkedő Shinjuku Mitsui Buildingen szemléltetjük. A második kép mutatja a torony tetején lengő tömeget, valamint a csillapítást, ami ebben az esetben a tömeg és a váz közötti összeköttetést jelenti. A harmadik képen a folyamat elvét láthatjuk leegyszerűsítve: a tömeg az épülettel ellentétes irányba mozogva

igyekszik visszatéríteni a tornyot a függőleges állásba. Úgy gondoltuk, hogy ezt a vezérlést lehetne szemléltetni egy egyszerű modellel.

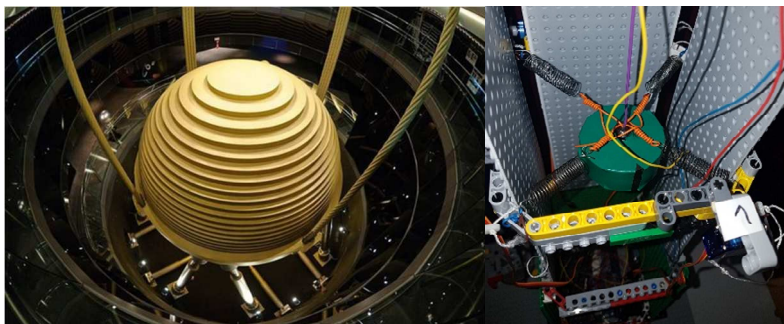
2. A mi kis tornyunk

2.1. Felépítése

Első lépésben a tornyot kellett megépítenünk. Első ötletünk ennek anyagára a Java építő volt, mert szerettük volna a belsejét is láthatóvá tenni. Azonban így nagyon instabil építményt kaptunk. Bár az is szempont volt, hogy majd lengésbe tudjon jönni, de ez túl hamar összedőlt volna. Így fordultunk a LEGO felé. A falakat LEGO-lemezek alkotják, így eleget tesz az előbbi feltételnek. Ezeket fogattuk össze Technics elemekkel, ahogy ez az első ábrán is látható.



2. ábra: Felső Toronyház



3. ábra: Taipei 101 és a mi ingánk

2.2. Az inga beszerelése

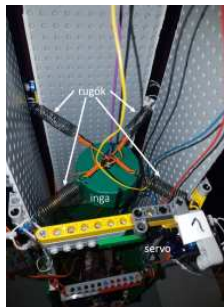
Természetesen az interneten utánanéztünk a Taipei 101-nek. Erről azt lehet tudni, hogy benne a csillapításról egy inga gondoskodik, amelynek a végén egy hatalmas, közel 6 méter átmérőjű gömb lóg. Az inga itt is magasan, a 92. és a 87. emelet közé felfüggesztve található. A maga 730 tonnájával ez a világ legnagyobb és legnehezebb ilyen csillapítószervezete. A tehetetlenség

elvét kihasználó ingát 16 acélkábelrel rögzítették a helyére, alulról pedig olajjal töltött lengéscsillapító karok tartják.

Ebből kiindulva mi az ingaként választott közel fél kilogramm tömegű korongunkat – azon kívül, hogy egy drótra függesztettük – 4 rugóval feszítettük ki a torony négy sarkához. A rugókban van elvezetve az a négy vékony madzag, amelyeknek a másik vége 1-1 szervomotorhoz van erősítve. Ezek segítségével lehet az ingát lengésbe hozni.

2.3. Gyorsulásmérő szenzor

Szükségünk volt még egy olyan szenzorra, amelynek adatai alapján lehetséges maga a vezérlés. Erre a célra a gyorsulásmérőt választottuk, amely a torony tetején kapott helyet. Elhelyezkedése olyan, hogy tengelyei párhuzamosak a rugókkal. Ennek köszönhetően közvetlenül az a_x és az a_y komponensek előjeléből tudunk következtetni arra, hogy az inga merre mozdult ki.



4. ábra: Az inga

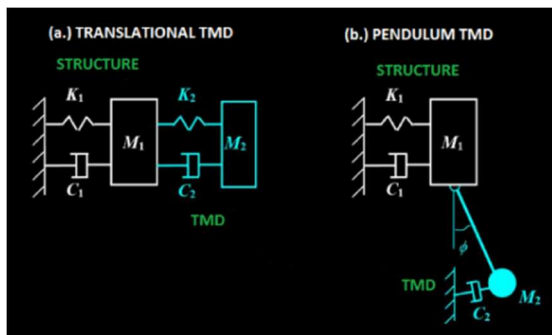
3. Az elektronika

Ahogy fentebb említettük, az ingát a szervomotorok segítségével mozgatjuk. Mikrokontrollerként Arduino Unot használunk, amelyet az Arduino IDE környezetben programozunk. A vezérléshez be kell vezetni a szenzor adatait az Arduinoba, illetve a megfelelő szervomotorokat kell működtetni. Ezek mindig párban mozoghatnak, a két szemközti egyszerre, az egyik húzza, a másik pedig engedi, vagyis nem tart ellent. Ez elég sok kábelt jelent, amelyeket igyekeztünk színkóddal ellátni.

4. Fizikai alapok

A TDM-technika feladata tehát az, hogy ha egy építmény valamilyen külső gerjesztő erő – ami származhat pl. földrengéstől vagy erős szélről – hatására rezgésbe jön, akkor annak amplitúdóját egy elfogadható értékre csökkentse.

A TMD maga is egy rezgő rendszer, így a megvalósításnak két fő formáját különböztetik meg: az úgynevezett transzlációs TMD, ami gyakorlatilag rugókkal rezget egy nagy tömeget, illetve a lengő TMD, vagyis egy inga, mint esetünkben is.



5. ábra: A kétféle TMD technika

Az ábra fehérrel jelöli az építményt, amit védeni akarunk, és ami maga is rezgésre képes; világoskékkel pedig a TMD rendszert. Ennek 3 fő része van: maga a tömeg, egy rezgést vagy lengést lehetővé tevő rugó illetve inga, illetve a C-vel jelölt csillapítás. Ez vezeti el a rezgési energiát. Megvalósítása valamilyen viszkozus folyadékot használó hidraulikával történik, mint a Taipei 101 esetében is.

5. Hol tartunk?

Az épület ingával és a szoftver elkészült. Azonban rá kellett döbbernünk, hogy a teszteléshez szükség van még valamire, ami a tornyunkat rezgésbe hozza. Erre egy, a torony alapterületénél nagyobb keretet használtunk, és rugókkal kikötve hozzá a tornyot, a keretet mi ráztuk. Azt tapasztaltuk, hogy a programunk nem bír el ezzel a „földrengéssel”. Ezért a közeljövőben feljebb költöztetjük a külső gerjesztő rezgést, reményeink szerint így már az inga lengései valóban csillapítják a toronyt.

Intelligens Szortírozó Rendszer

Sziksi Robot TÉP

Magyar Dávid, Felföldi János, Maródi Gergő

Felkészítő tanár: Bakti András Tamás

*Széchenyi István Közgazdasági, Informatikai Szakgimnáziuma és Kollégiuma,
Nyíregyháza, Városmajor utca 4.*

1. Bevezetés

Rendszerezés, egy olyan folyamat, ami különös odafigyelést igényel; hiszen a legkisebb hiba is felboríthatja a dolgok egyensúlyát. A rendszerezés feladata – pontosan a fent említett dolog miatt – eddig többnyire emberek által végzett volt, mivel egy ember szakmai tapasztalata és odafigyelése elengedhetetlen volt. Viszont az ember sem tévedhetetlen, a munka monotonitása miatt a hibamutató potenciálisan elég magas; ami elvesztegetett időhöz vezethet, amikor egy adott dolgot meg kell keresni – tegyük fel egy nagy raktárban.

Az Intelligens Szortírozó Rendszer ezt a problémát hivatott megoldani. Felhasználva modern technológiákat, mint pl.: képfelismerés, mikrovezérlő elektronikák és a rendszer különböző egységeiken összehangolt – egymástól független működése. Ennél fogva munkahelyeken, vagy akár más ipari környezetben is kiválóan használható lenne. Továbbá, a munkaerőhiány ellensúlyozására is alkalmas.

2. Probléma megoldásának menete

A tárgyak/dolgok rendszerezésénél a legfőbb probléma a tárgyak elhelyezése, majd azoknak effektíven és gyorsan történő megkeresése. Illetőleg a balesetveszély.

Ezen problémák kiküszöbölésének érdekében szükség van egy gyors és megbízható rendszerre, amely – a balesetek elkerülése érdekében – teljesen automata és akár embertől függetlenül, felügyelet nélkül is működőképes.

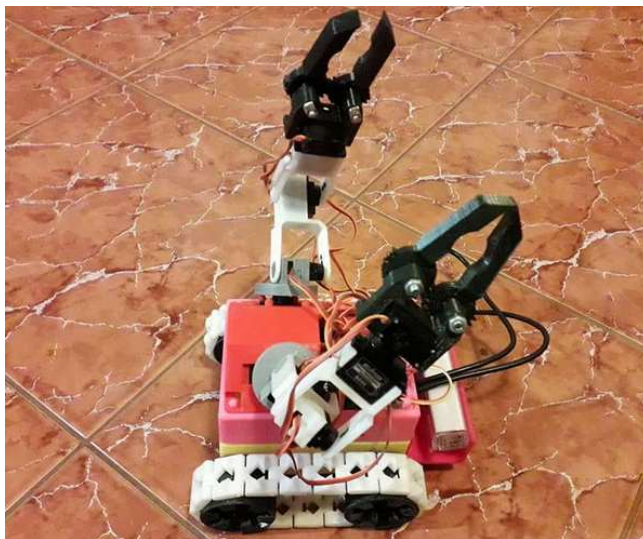
A programozható mikrovezérlők és az egy-lapos számítógépek a rendszer vezérlésére tökéletesen alkalmasak. Ezekre a programok MicroPython, Python ill. C/C++ nyelvekben kerülnek megírásra, illetőleg a használt Op. Rendszerként: a Linux lesz felhasználva.

A Python és MicroPython rendre a számítógépen illetőleg a mikrovezérlőkön lesznek felhasználva: pl.: ESP8266; míg a C/C++ programok – a teljesítmény igényesebb – mikrovezérlőkön: pl.: Arduino.

A már bejegyzett, rendszerezett tárgyakat egy WEBes felületen keresztül lehet megtekinteni a pozíciójukkal és a hozzáadás dátumával együtt. A felület lehetőséget biztosít egy kiválasztott tárgy elővételére is.

2.1. Tárolás, mozgatás

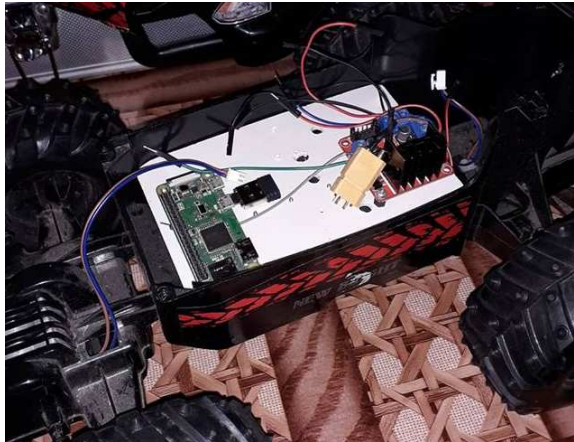
A tárgyak tárolása egy alumínium kerettel megerősített „raktárban” történik, ami lehetőséget ad számos dolog tárolására. A dolgok/tárgyak egy – a kategóriájának megfelelő – cellában kerülnek majd tárolásra. Ide, egy rakodó tálcán keresztül jutnak majd el, amit egy 3D Nyomtató vagy egy CNC gravírozó rendszerhez hasonlóan léptető motorok vezérelnek. Ez a tálca 3 tengelyen képes mozogni: x, y, z. A tárgyat/anyagokat egy szállító/előzetes tároló helyről választja ki; jelen esetben egy szállítójárműről.



2.2. Indexelés, rendszerezés

A tárgyak indexelése és rendszerbe sorolása egy képfelismerő rendszer segítségével történik egy Raspberry Pi 3-on egy Kamera felhasználásával. A képfelismerő rendszer lehet QR kód alapú, vagy akár a tárgy alakjától függően forma felismerés is. A beolvasott/felismert kép alapján a rendszer egy előre definiált adatbázis alapján megállapítja, hogy a tárgy milyen kategóriába tartozik, illetőleg a tárgyat magát. Ezt bejegyzí és tovább adja a rakodónak, ami aztán a fent említett módon elhelyezi azt a megadott helyen. Ha

felhasználói beavatkozás történne, tegyük fel, hirtelen kell megkeresni valamit, akár a rakodó kézi irányításával – a rendszer ennek befejeztével onnan folytatná, ahol a beavatkozás/megszakítás előtt abbahagyta; ugyanez vonatkozik pl.: áramkimaradás esetére is, vagy akár fizikai akadály esetén is.



2.3. Egymástól független működés

A rendszer egységei egymástól függetlenül képesek működni, hiszen az azonosító és továbbító egységnek nincs szüksége a rakodó egységre a beazonosításhoz; csak úgy, mint a rakodó egységnek nincs szüksége az azonosító egységre ahhoz, hogy fel tudja pakolni a tárgyat egy helyre. (Jól lehet, az egységek képesek egymás nélkül működni, viszont elég jól kiegészítik egymást)

3. Elért eredmények

Ezzel a projekttel elég sokat tanultunk egybe, de nem egymástól függő rendszerek tervezéséről, illetve annak megvalósításáról. A programozói „vénánk” (hogy így fogalmazzunk) is erősödött ennek a feladatnak az elvégzésének hála.

A projekt lehetőséget adott felhasználó-barát WEBes felületek tervezésére, illetőleg bizonyos biztonsági technikák megtanulására és alkalmazására, mind WEBes, mind Rádiófrekvenciás téren is. A feladat elvégzése során a csapat tagjai szabadon dobálhatták ötleteiket, szabadon engedve kreativitásukat.

Informatika szekció

Guess What Will Happen!

Bingoo!

Gordos Bence

Felkészítő tanár: Olajosné S. Erika

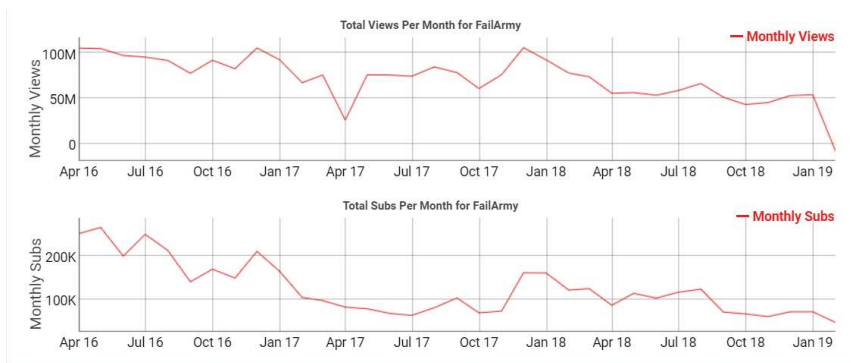
*Szegedi SzC Gábor Dénes Szakgimnáziuma és Szakközépiskolája,
6724 Szeged, Mars tér 14.*

1. Bevezetés

Gordos Bence 10. osztályos tanuló vagyok. A pályamunkám egy androidos alkalmazás, melynek a Guess What Will Happen! címet adtam. Az alkalmazást Java és XML nyelven írtam az Android Studio fejlesztői környezetben. Az alkalmazás az interneten hatalmas népszerűségnek örvendő „fail” videókkal interaktív módon történő szórakoztatás, illetve az ezekben a videókban végzett különféle veszélyes cselekményektől való elrettentésről szól.

2. Probléma megoldásának menete

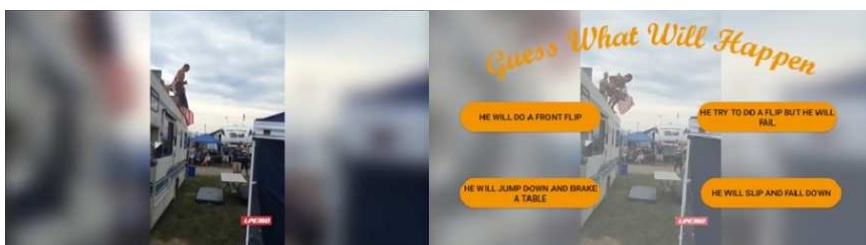
Ezek az említett fail videók általában YouTube-on terjednek. A legnépszerűbb ilyen csatorna a FailArmy, ami az egész platformon jelenleg 172. a sorban, több mint 14,5 millió feliratkozóval. Az 1. ábrán látható a FailArmy nézettségének és feliratkozóinak alakulása.



1. ábra: A socialblade.com adatai a FailArmy havi nézettségéről (felső diagram) és a havi új feliratkozók számáról (alsó diagram)

Ennek a rengeteg nézőnek nagy része azonban kiskorú, és miután barátaival ezeken a videókban nevet, lehet, hogy jó ötletnek tartja majd a videókban látottakat kipróbálni. Ezen a gondolatmeneten elindulva szerettem volna egy olyan alkalmazást fejleszteni, ami egyszerre szórakoztat, és valahogy megpróbálja elrettenteni a nézőt/játékost a látottak kipróbálásától.

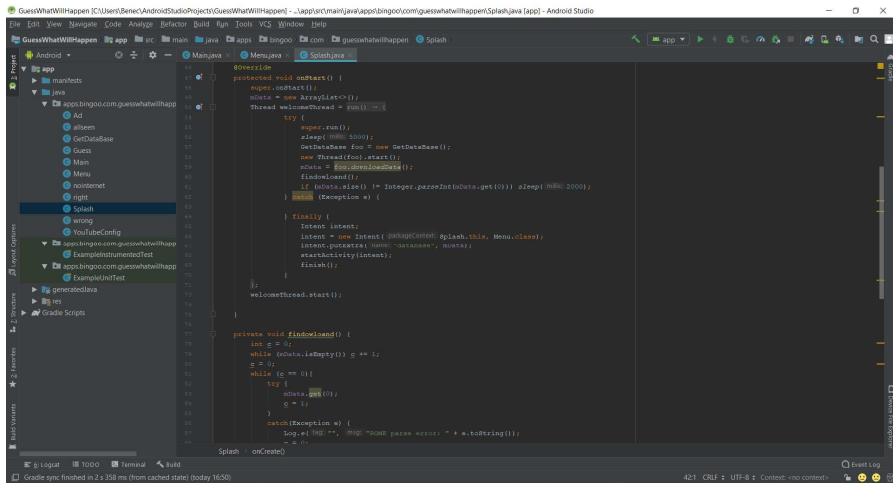
Ezt úgy gondoltam, hogy a legjobban egy interaktív videó lejátszó segítségével lehet elérni. Az ötletem az volt, hogy a videók a „tetőpontjuk” előtt, vagyis az előtt, hogy a „fail”, „vicces” esemény megtörténne, megáll, és itt a játékosnak négy opció közül ki kell választania, hogy szerinte hogyan fog végződni a videó. Ezzel arra sarkalva a nézőt, hogy végig gondolja, mit csináltak a videó szereplői, és vajon milyen rossz dolgok sühetnek ki belőlük. Ez a folyamat látható a 2. ábrán.



2. ábra: Kép a játékból

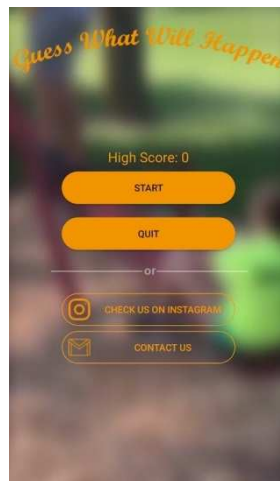
A programot három részre lehet bontani: videók, válaszok betöltése – töltőképernyő; főmenü; videó lejátszó.

A videókat előre megvágтам Sony Vegas program segítségével, és két részre bontottam: tetőpont előtt és után. Ezeket külön feltöltöttem YouTube-ra privát videóként. Ezeknek a videóknak a linkjét a hozzájuk tartozó válaszokkal és a videó megállításának pillanatában vett index kép linkjével együtt egy txt fájlban feltöltöttem Dropboxra. A töltőképernyő innen olvassa ki és menti el egy tömbbe a fájl tartalmát. Ennek a folyamatnak Java kódjából egy részletet ábrázol a 3. ábra. A három linkből és opciókból fog majd később összetevődni a videó lejátszó rész. Ezzel a módszerrel az alkalmazás kevesebb helyet foglal a telefonon, és ha videókat a későbbiekben frissítem, nem kell a programban változtatásokat eszközölnöm, csak a txt fájlban kell átírnom vagy hozzáadnom linkeket, opciókat.



3. ábra: Részlet a töltőképernyő kódjából

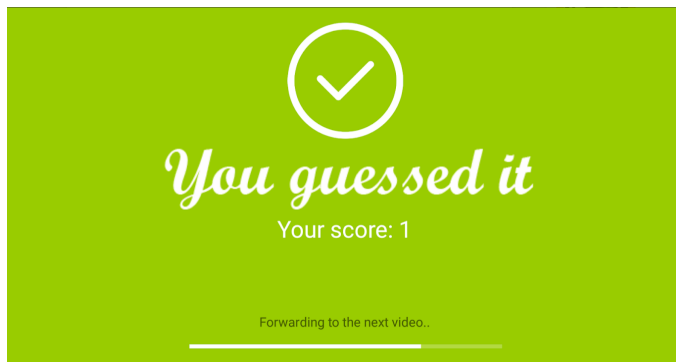
A főmenünek a játék alakulásának szempontjából nagyobb jelentősége nincs. Itt a megjelenéssel, formatervezéssel dolgoztam sokat. Egy videó háttérrel kapott és a gomboknak egyedi megjelenésük van. Sajnos ezt 4. ábrán nem tudom teljes egészében bemutatni.



4. ábra: Főmenü (a háttérben videó)

A videó lejátszó még a töltőképernyő segítségével betöltött tömbből egy olyan véletlenszerű videót indít a hozzá tartozó opciókkal, amit a játékos eddig még nem látott. Miután a videó véget ért annak függvényében, hogy a játékos jó opciót választott vagy rosszat, a High Score-hoz hozzáír egyet (ez a

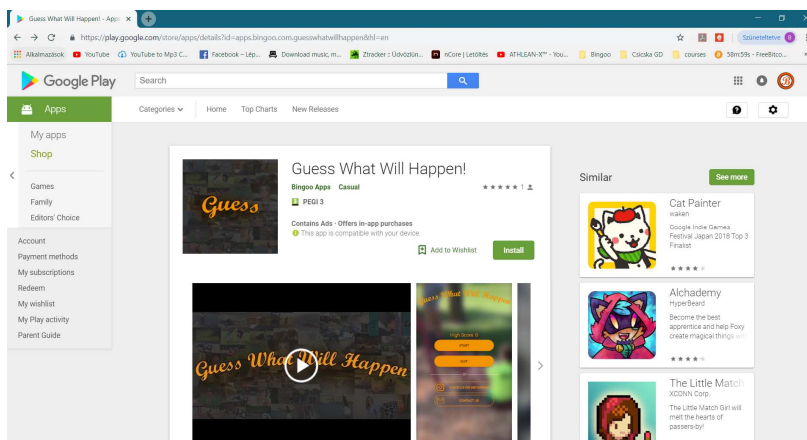
változó eltárolódik és az applikáció minden indulásakor az eltárolódott értéket fogja mutatni), és egy újabb véletlenszerű videóhoz továbbítja (5. ábra) a játékost, vagy eltörli az ebben a körben elért pontokat.



5. ábra: Ha a játékos kitalálta a videó befejezését ezt látja

3. Elért eredmények

Az eddig elért eredményekről sajnos nem sokat tudok mondani. 4 hónapos kemény tanulás és fejlesztői munka után közzétettem GooglePlay Áruházban (6.ábra), és létrehoztam egy Instagram oldalt az applikáció népszerűsítéséért (<https://www.instagram.com/bingoo.apps/>). Marketing kampányba kezdetem, de ezen a területen is sokat kell még fejlődnöm, hogy lényegesebb sikereket érjek el.



6. ábra: GooglePlay áruházi adatlap

Learning Machine

Chuck Norris

Antal Péter

Felkészítő tanár: Pap-Szigeti Róbert

Kecskeméti Bolyai János Gimnázium,

6000 Kecskemét, Irinyi u. 49.

1. Bevezetés

A tanulás életünk szerves részét képezi, és ez nem csak a formális, az iskolában történő tanulást jelenti. A környezetből, amely minket körülvesz, rengeteg összefüggést, szabályt megtanulunk, amit viszont az iskolában nem fognak számon kérni tőlünk. Amit ott kérnek, azt koncentrált, tudatos tanulás folyamán tudjuk „bemagolni”, ami viszont sok diáknak nagyon nehézkesen, vagy egyáltalán nem megy (például a figyelemzavaros diákoknak). Továbbá rengetegen nem képesek - esetleg egyéb elfoglaltságaik miatt - elegendő energiát fektetni ebbe a folyamatba. Viszont ezek a diákok ugyanúgy tanulnak kevésbé tudatosított módon, például a környezetükben bekövetkező eseményekből úgy, hogy mindezt észre sem veszik. Ilyen kevésbé tudatos tanulási forma a spontán tanulás, és ez – ahogyan azt a pszichológia tankönyvekből is megtudhatjuk - az egyik, ha nem a leghatékonyabb mód, mert ilyenkor a figyelem spontán módon összpontosul. Így jegyeznek meg a tanulók az okoseszközeiken megjelenő napi több ezer, a diákok tanulmányainak szempontjából felesleges adat közül jó néhányat.

De hogyan is hasznosíthatjuk ezeket a tényeket? Nincs könnyű dolgunk, hiszen egy spontán történéshez – melyből tanulhatunk – néhány ismétlődés után hozzászokik az ember, így az már nem is spontán többé. Itt jön képbe az okostelefon, amely mindig egyfajta spontaneitást tesz lehetővé értesítéseinek ingergazdagságából, és állandó változásaiból fakadóan. Az értesítéseit mindenki rendszeresen figyelemmel követi, így az kiváló alapja lehet ennek a tanulási formának. Alkalmazásomban ezt próbáltam kihasználni.

2. Probléma megoldásának menete

Megoldásom tehát egy, az okostelefonos értesítésekre épülő tanulást szolgáló alkalmazás, amely témakörökre bontva tartalmaz különböző évszámokat, idegennyelvű szavakat és kifejezéseket. Mindezek szerkeszthetőek, van keresési lehetőség, sőt hozzá is adhatunk tantárgyakat és elemeket a meglévőkhöz.

Az alkalmazás beállítás után véletlenszerű tananyag-értesítéseket dobál fel a felhasználónak, bizonyos tananyagtípusok esetén pedig vissza is kérdez szintén értesítések formájában. Ezekre még az értesítésen belül válaszolhat a felhasználó, helyes válasz esetén pedig a későbbiekben ugyanezt kisebb valószínűséggel kapja meg értesítésben. Így téve a tanulás ezen formáját összetettebbé, illetve így a jövőben súlyozottan olyan értesítések jelennek majd meg, amelyek relevánsak a felhasználó tudásszintjének szempontjából.

Az alkalmazást Java nyelven írtam, Android operációs rendszerre. Android 6.0-tól felfelé telepíthető, viszont kifejezetten Android 8.0-ra és 9.0-ra épülő rendszereken ajánlom a használatát.

A program elkészítése során egy nagyobb probléma, amibe belefutottam az volt, hogy egy olyan adatbázist kellett létrehoznom, amely az érettségi-követelményeknek megfelel például az évszámok terén, vagy idegenszavak tekintetében. Ezeket a tananyagokat végül különböző internetes oldalakról válogattam össze, ezzel is a felhasználót segítve, hiszen így nem kell majd mindent egyesével begépelnie.



1. ábra: Az alkalmazás különböző ikonjai

3. Elért eredmények

Habár az alkalmazásboltokban számos tanulást segítő app található, az értesítésekkel tanító, illetve a magyar tanterv anyagának mechanikusan tanulni való részét ilyen szinten összefoglaló alkalmazás eddig még nem létezett. A munkánk célja az volt, hogy a diákoknak kevésbé megerőltetővé tegyük a tanulás folyamatát, és az eddigi visszajelzések alapján erre hatalmas igény van, akár ilyen formában is. Ez a megoldás pedig – lévén, hogy az iskolákban szinte mindenkinek van okostelefonja - széleskörűen is használható a diákok körében.

A design kialakításánál a Google által publikált „Material” -elveket követtem, a kódot igyekeztem a lehető leghatékonyabbra megalkotni. Az app menürendszere intuitív és átlátható minden tanulni vágyó számára.

Ez az alkalmazás összességében könnyen válhat a tanulók mindennapjainak részévé. Mindennek „mellékhatásaként” pedig az okoseszközök társadalmi megítélése is javulhat, hiszen az alkalmazással valóban a tanulás folyamata zajlik le, a diák számára szinte észrevétlenül, ezáltal a sok „felesleges” információ mellett a tárgyi tudásuk is bővíülhet.

BeHealthy étrend és edzésterv tervező

BeHealthy

Skoda Barnabás

Felkészítő tanár: Halász Sándor

Budapesti Műszaki Szakképzési Centrum Petrik Lajos Két Tanítási Nyelvű

Vegyipari, Környezetvédelmi és Informatikai Szakgimnázium,;

1146 Budapest, Thököly út 48-54.

1. Bevezetés

A 21. Században, a felgyorsult világunkban nagyon nehéz mindenre odafigyelni és egészségesen élni. Magyarországon 3 felnőttből sajnos 2 elhízott vagy túlsúlyos, és a fiataloknál sincs ez másképp. Egy átlagos felnőtt naponta több mint 5 órát ül és csak 10 percet sportol. A diákoknál ez az arány rosszabb mivel az iskolában napi 7-8 órát is végig kell ülni és koncentrálni, igaz van mindennapos testnevelés, de ez sokszor nem megfelelő és (vagy) nincs kialakítva hozzá az alkalmas környezet. A fiatalok körében elterjedtek a számítógépes illetve telefonos játékok is, melyek még inkább növelik az üléssel töltött órák számát.

Egy felmérés szerint Magyarországon a 14 éven felüliek 15% sportol rendszeresen, 23% alkalmanként, 18% ritkán, és 44% pedig egyáltalán nem sportol. A rossz minőségű ételek fogyasztása és a mozgásszegény életmód az életminőség romlásához és elhízáshoz vezet, mely növeli a magas-vérnyomás, cukorbetegség, koszorúér-betegség és a daganatos betegségek kialakulásának esélyeit. Megfelelő minőségű és mennyiségű ételmiszerrel valamint rendszeres sportolással csökkenteni lehetne a daganatos betegség kialakulásának 20-40%-át és a halálozások felét el lehetne kerülni

A sportolni és változtatni vágyók gyakran ütköznek akadályokba és megfelelő háttértudás nélkül kudarcba fulladhatnak. Az akadályok átlépését nem csak a megfelelő háttértudás hiánya nehezíti, hanem mind az interneten mind a sportlétesítményekben terjedő tévhitek, melyek nem csak gátolnak célunk elérésében, hanem súlyos egészségügyi problémákhoz vezethetnek.

2. Probléma megoldásának menete

2.1. Célkitűzés

Célom egy telefonos applikáció megtervezése és felépítése, mely elérhető lenne minden olyan olyan telefonon, amin Android 7.0 Nougat vagy újabb

operációs rendszer fut. A telepítést könnyen elvégezhető Play Áruházból és regisztráció után azonnal használatra kész.

2.2. Az alkalmazás funkciói

Az étrend tervező

Az alkalmazás nagy segítséget nyújt egy megfelelő étrend megtervezéséhez, ez történhet az alkalmazás segítségével vagy akár egy meglévő étrendet is bevihetünk, tervezéskor több tippet és szempontot is kapunk, amely nagyban megkönnyíti a dolgunkat. Nap közben lehetőségünk van az étrend naplózására is, mely segítségével könnyedén nyomon tudjuk követni, hogy mit ettünk és mit nem. De ha véletlenül megszegnénk az étrendet, az elfogyasztott ételt egyszerűen hozzá tudjuk adni a napi bevitt mennyiséghez.

Az edzésterv tervező

Megfelelő fizikum, edzettség és jó közérzet eléréséhez nem elegendő csupán egy megfelelően megtervezett és felépített étrend. Szükséges a megfelelő mennyiségű és jól végrehajtott edzés is.

Az edzettség felmérése után mely pár kérdésből áll. Az alkalmazás megállapítja, hogy mennyire vagyunk edzettek és ez alapján több előre gondosan elkészített edzéstervből ki fogja választani, hogy melyik lenne számunkra a legideálisabb. Ezt az edzéstervet kedvünkre módosíthatjuk (adhatunk hozzá és vehetünk el gyakorlatokat) de itt is szintúgy, mint az étrend tervezőnél szempontokat és tippeket kapunk a módosításhoz.

Minden egyes edzésünket naplózhatjuk, ami nagyban megkönnyíti fejlődésünk nyomon követését melyet egy grafikonon is megtekinthetünk. Ha funkcionális edzés vagy testépítés helyett/mellett szeretnénk más sportokat is naplózni azt könnyedén kiválaszthatjuk egy listából és elindíthatjuk az edzést.

Egyéb funkciók

Kalóriatáblázat

Ha esetleg meg szeretnénk tudni, hogy egy étel vagy ital milyen arányban tartalmaz makrotápanyagokat, akkor ezt könnyedén megtehetjük a kalóriatáblázat segítségével. A kalóriatáblázat kezdetben az alap összetevőket tartalmazza, de bármelyik felhasználó adhat hozzá terméket vagy receptet így a táblázat folyamatosan bővül és akár kedvencünket is beírhatjuk és ezt fel is tudjuk használni az étrendtervezés során.

Bevásárlólista

A bevásárlólista nagyban megkönnyíti életünket, be tudjuk állítani, hogy heti vagy kétheti felosztásban szeretnénk vásárolni, ezt követően a meglévő aktuális listát könnyen meg tudjuk osztani e-mailen vagy ismertebb social médián keresztül ismerőseinkkel, családtagjainkkal.

Nyelvválasztás

Az alkalmazás egyelőre két nyelven elérhető (magyar és angol) a nagyobb célközönség elérése érdekében.

Tudásbázis

A tudásbázisban fontos információkat olvashatunk a gyakorlatok helyes végrehajtásáról (Animált képekkel illusztrálva), fontosságáról valamint táplálkozás alapjairól.

2.3. Az alkalmazás megvalósítása

Használt programnyelvek, könyvtárak és keretrendszerek

Az Alkalmazás megvalósításához többféle programnyelvet, stílus/jelölő nyelvet, keretrendszert és függvénykönyvtárat használtam.

A szerver elkészítése

A szerver elkészítése volt az első lépés, ez azért volt fontos, hogy az applikációt letöltve bármelyik eszközünkön elérhetőek legyenek az eddigi eredményeink. Az adatbázis elkészítéséhez a MySQL adatbázis kezelő szervert használtam. Az ellenőrzéseket és lekéréseket egy PHP szerver végzi, amit az applikáció egy Ajax kapcsolaton keresztül ér el.

Az alkalmazás elkészítése

Az alkalmazás elkészítése volt a második lépés, ehhez a Cordova keretrendszert hívtam segítségül, melyben HTML, CSS és JavaScript segítségével tudunk web alapú hybrid alkalmazásokat készíteni. A Cordova keretrendszerre egy Onsen UI keretrendszer épül, amelyet egy AngularJS Model View Controller keretrendszerrel egészítettem ki. A kinézetet a JQuery, GSAP JavaScript könyvtárával és Less CSS-sel építettem fel. Az adatokat egy telefonon lévő adatbázisban tároljuk (WIFI esetén azonnal szinkronizálódik a központi adatbázissal) és az alkalmazás minden különálló részéhez külön Controller tartozik, amely megkönnyíti az alkalmazás kódjának átláthatóságát és fejlesztését.

3. Elért eredmények

Az alkalmazás fejlesztését tanulási céllal kezdtem el. Valami olyan témát, projektet szerettem volna választani, ami közel áll hozzám, és ami nagyon fontos az, hogy hasznos a társadalom számára is mivel nagyon szeretek embereken segíteni. Körülbelül 3 éve kezdtem el a súlyzós edzést, hol kisebb, hol nagyobb félbeszakításokkal. Sajnos az elején fogalmam sem volt, hogy igazából mit is kellene csinálni, de ezalatt a három év alatt nagyon sokat tanultam, több könyvet is kiolvastam a témában. Szerettem volna, ha minden tudás egy helyen lenne, és nem kellene innen-onnan összeollóztatni, ezért kitaláltam, hogy írok egy alkalmazást, ahol minden szükséges alaptudás ott van egy helyen rendszerezve, elemezve megtalálható ezért még jobban beleástam magam a témába és még többet tanultam.

Programozás terén is rengeteget fejlődtem. Megtanultam rengeteg keretrendszernek a használatát. Megtanultam objektum-orientáltan gondolkodni és a kódok sokaságát átlátni, rendszerezni. Megtapasztaltam milyen egy több ezer kódsoros projekten dolgozni és legutolsó sorban megtapasztaltam milyen valami teljesen mást alkotni.

MilAxe

Gucci Flip Flops

Kovács Alex Márk, Szász Milán

Felkészítő tanár: Pap-Szigeti Róbert

Kecskeméti Bolyai János Gimnázium,

6000 Kecskemét Irinyi út 49

1. Bevezetés

Egy tanulást, illetve memóriát elősegítő/fejlesztő játék, melynek használata azért előnyös, mert a grafikai megjelenítések miatt agyunk könnyebben megjegyzi a megtanulandó anyagot, miközben a memóriánk is fejlődik, illetőleg jól is érezzük magunkat akár egy számunkra kedves személy környezetében.

A játék egy fogócska digitális verziója, mely egy 9 kérdésből és 9 válaszból (!) álló memóriajátékkal van megfűszerezve. A játékkal jelenleg csak 9 kérdéssel és 9 válasszal lehet optimálisan játszani {fejlesztés alatt...}. Lehetőség nyílik egyedül, vagy párosan játszani. Amennyiben a felhasználó az 'Egyjátékos' módot választja, úgy az ő célja a pálya végére történő eljutás lesz. Ekkor a játék egyetlen kör megtételéből áll. Amennyiben a felhasználó a 'Kétjátékos' módot választja, úgy az ő célja a másik játékos utolérése lesz (ez fordítva is igaz). Ekkor a játék addig tart, amíg az egyik játékos utol nem éri a másikat, és rá nem lép arra a pályaelemre, amin éppen játékos társa tartózkodik. A nyertes a korább említett játékos.

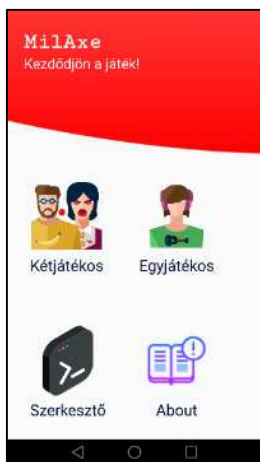
2. Az alkalmazás működésének menete

2.1. Alkalmazás indítása

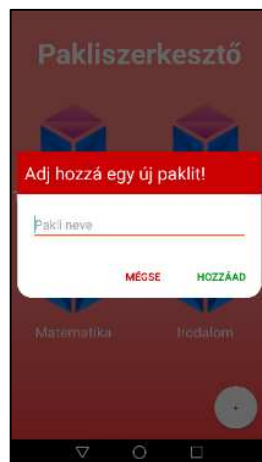
A bevezető animáció után a menü fogadja a felhasználót. (1. ábra)

2.2. Megfelelő menüpont kiválasztása

A felhasználónak el kell döntenie, hogy a már meglévő paklikkal vágyik/vágnak játszani, vagy újakat szeretne/szeretnének hozzáadni, esetleg szerkeszteni akarja/akarják a már meglévőket. Amennyiben az utóbbiak valamelyikét kívánjuk végrehajtani, akkor a 'Szerkesztő' menüpontot kell választani (2. ábra). Amennyiben játszani szeretnénk, akkor a megfelelő játékmódot kell választani.

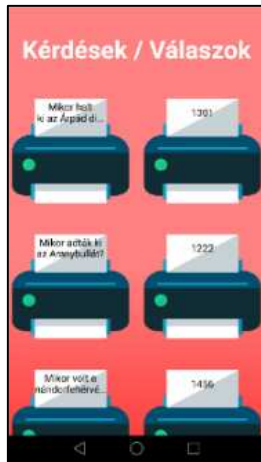


1. ábra: A menü

2. ábra:
Pakliszerkesztő3. ábra: Új
pakli
hozzáadása

2.3. Pakliszerkesztő funkciói

A 'Szerkesztőn' belül lehetőségünk van megtekinteni az eddig létező kártyapaklikat. Ha szeretnénk új paklit hozzáadni (3. ábra), akkor a jobb alsó sarokban elhelyezett '+' gombra kattintva tehetjük ezt meg, a számunkra megfelelő név beírása után (már létező nevű paklit nem adhatunk hozzá). Ha már egy meglévő pakli nevét szeretnénk megváltoztatni, vagy esetleg törölni akarjuk azt, akkor tartsuk ujjunkat az adott paklin hosszan, majd járjunk el a kívántak szerint. További lépésként egy kártyapaklira kattintva tekinthetjük meg a pakli által tartalmazott kérdéseket, illetve válaszokat. Amennyiben egy már létező kérdést/választ szeretnénk szerkeszteni, akkor csak kattintsunk az adott nyomtató ikonra, és írjuk be a megfelelő szöveget. Ha egy új paklit adunk hozzá, akkor gyárilag nem tartalmaz sem kérdéseket, sem válaszokat. Ilyenkor nekünk kell hozzáadni a kérdéseket és a válaszokat is! (9-9 db {fejlesztés alatt...})



4. ábra: Kérdések és válaszok



5. ábra: A pálya

2.4. A játék

A játék indításához először ki kell választanunk, hogy melyik paklival szeretnénk játszani. Miután ez megtörtént, a játék betöltődik (5. ábra), majd kezdődhet is. A 'Kétjátékos' módban a játékosoknak ki kell választaniuk, hogy melyik játékos legyen a kezdő.

Ez után arra a pályaelemre kell kattintani, amelyiken az éppen soron lévő játékos áll, majd a kérdés elolvasása után a középben lévő memóriakártyákat tartalmazó részre kattintva a kérdéshez tartozó megfelelő válasz memóriakártyákat (2db) kell kiválasztani. Ha a tippünk jó volt, akkor újra mi jövünk. Ekkor újra az aktív pályaelemre kell kattintani, majd a kérdés elolvasása után a megfelelő kártyákat kiválasztani. Egészen addig jövünk mi, amíg jó válaszokat adunk. Ha tippünk rossz volt, akkor a másik játékos következik, akinek ugyanezt a folyamatot kell végrehajtania. Az nyer, aki utoléri a másikat! Az 'Egyjátékos' módban hasonlóan kell eljárni, a különbség mindössze annyi, hogy ha rossz választ adunk meg, akkor is mi következünk... A játék végén a felület értesít minket a nyertesről!

3. Elért eredmények

Az alkalmazás által a tanulás egyfajta játékká alakul át, ezáltal a felhasználó sokkal nagyobb kedvvel használja az applikációt. A játék megnyeréséhez szükséges, hogy a felhasználó felismerje az adott kérdésre a választ és mivel a kérdések egy játék alatt többször is előfordulnak, így még

jobban rögzül a játékos memóriájában az adott információ. A kétjátékos mód azon kívül, hogy egy közös elfoglaltság, a játékosokat még jobb teljesítményre is motiválja. A pakliszerkesztő pedig lehetőséget nyújt adott tantárgyak tanult tananyagainak játékba fűzésére. Véleményünk szerint a MilAxe egy olyan játék, amely változatosságot hoz a tanulásba, ezáltal élvezetesebbé és hatékonyabbá téve azt. Ez az alkalmazás vagy ennek az alkalmazásnak egy majdani tovább gondolása alkalmas lehet az oktatás korszerűsítésére. Hogy mit is értünk ez alatt? Ha tudni akarja, akkor tudja mit kell tennie ;). A viccet félre téve a mai oktatásra ráférne egy modern arculat és egy ilyen program lehet a korszerűsítés egyik lépcsőfoka, hisz ma már mindenkinek ott van a zsebében az eszköz, amely végtelen lehetőséget hordoz magában, mégis a rendszer ezt visszafojtja, félve az újtól.

A közeljövőben tovább fejleszteni a MilAxe alkalmazásunkat. Bővíteni szeretnénk a játékban megjelenő kérdések számát, bluetootht szeretnénk építeni a kétjátékos módba, ezzel sokkal kényelmesebbé téve azt. Másodszor szeretnénk több, másféle tanuló programot is fejleszteni, amelyek ehhez hasonlóan újdonságot hozhatnak a hétköznapi tanulásba.

Házizz

Házizz velünk

Isakov Erik, Stefán Kornél, Vad Avar

Felkészítő tanár: Répásné Babucs Hajnalka

*BMSZC Neumann János Számítástechnikai Szakgimnáziuma,
1144 Budapest, Kerepesi út 124.*

1. Bevezetés

Egy diák élete minden, csak nem mese. Ha nem elég, hogy a tanáraik nap, mint nap véges -végtelen feladattal ostromolják, még olyan örökzöld problémákkal is küzdhetnek, mint például: "Mi volt a házi feladat angolból?", vagy hogy "Mikor írjuk töriből a 2. VH témazárót?" és még sorolhatnánk napestig. Hogy ezekre a kérdésekre mi a megoldás? Egy felület, ahol lehetőségünk van a nap történéseit feljegyezni, időzíteni és ezeket könnyedén megosztani a társainkkal, az általunk választott eszközről.

Kiemelt célunk még, hogy ez az eszköz könnyen magába olvasszon más platformokat (pl: E-Kréta), valamint, hogy mások egyszerűen integrálhassanak minket.

2. Elveink

A projektnél minden döntésünkkel olyan irányelveket próbálunk tükrözni, hogy a projekt mind technikailag, mind felhasználói élményben olyan minőségű legyen, amire akár mikor büszkéek lehetünk.

Felhasználói élmény centrikus megközelítés

Igazi multiplatform szabványok használata (Pl. OAuth2, REST)

Felhő barát platform tervezés

Rugalmas felépítés, tekintettel a jövőbeli fejlesztési tervekre

3. Probléma megoldásának menete

Ahhoz, hogy a fentebb vázolt problémát hatékonyan megoldjuk, egy termék és egy technológia nem elég, ezért a lehető legmodernebb felfogásban több részre osztottuk a projektet, az alábbiak mintájára:

3.1. Házizz Platform

Az ún. Házizz Platform a projekt szíve, ahol az adatkezelés nagy része történik. A kliensek ezt a kiszolgáló gyűjteményt használják a felhasználói szükségletek ellátásához. A Platform több microservice-ből áll, más szóval cél feladatokra kitalált szerverekből épül fel.

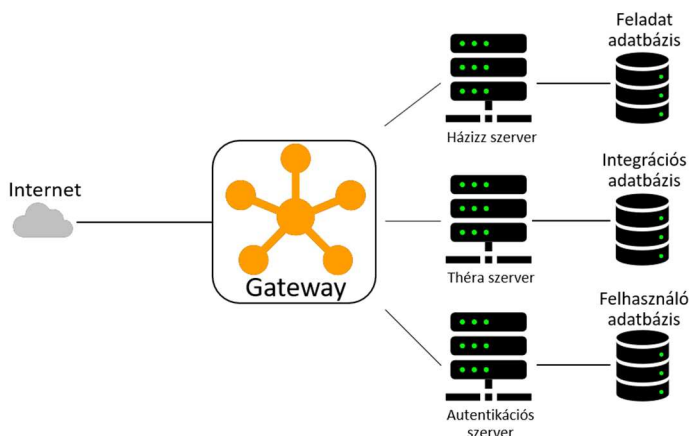
Az alábbiakban felsorolt modulok mind Javában megírt szerverek, a Spring Framework használatával készülnek, vagy teljesítmény kritikus szituációkban kiegészítő Rust szerverek.

A platformnak először is szüksége van egy regisztrációs, egy bejelentkezési, és egy fiókkezelési felületre. Erre a célra van az autentikációs és autorizációs szerverünk, ami OAuth2 standard-del és 0-s szintű REST API-val elérhetővé teszi a kliensek számára a felhasználó igazolását és adatainak elérését.

Az első fő modul egy "közösségi szolgáltatás", ahol a felhasználóknak lehetősége van csoportokat létrehozni, azon belül témákat, feladatokat, és bejegyzéseket készíteni. A csoportokban több felhasználó is lehet, így az eszköz alkalmas csapatmunkára. A modul támogatni fogja a csoportokhoz való tanár hozzáadását, aki a saját témáját tudja feladatokkal és bejegyzésekkel ellátni. Ez és a következő modul 2. szintű REST API-val érhető el a kliensek számára.

Egy integrációs modul is helyet kapott a platformon belül (kódnév: Théra). Ennek a modulnak a célja, hogy olyan szolgáltatásokat integráljunk, mint például az E-Kréta, vagy a Google Calendar, a felhasználói élmény növelésének érdekében.

Ezek a modulok egy gateway szerver mögött működnek, és csak HTTPS protokollal, a gateway szerveren keresztül érhetőek el.

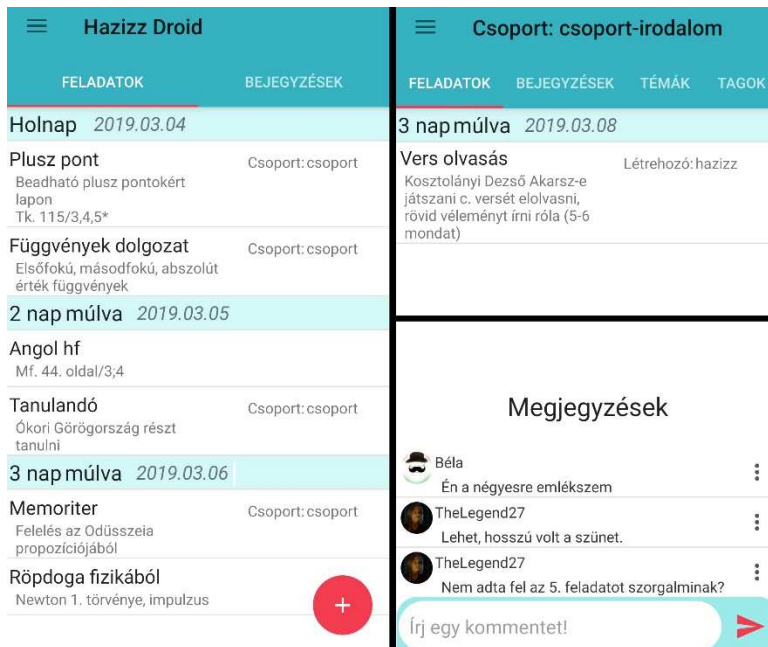


1. ábra: Házizz Platform diagramja

3.2. Házizz Droid

A Házizz Droid egy Android kliens, amelynek célja az, hogy a felhasználó gyorsan, egyszerűen és megbízhatóan érje el a Házizz funkciókat, pont akkor, amikor szüksége támad rá. Technikai szempontból a rendszer natív Javában van megírva, a lekérdezésekhez a Retrofit könyvtárat használjuk, és a legtöbb funkció fragment-ek használatával van implementálva.

Az applikáció egy widgetet is tartalmaz, mellyel gyorsan és könnyen megtekinthetőek a feladatok, illetve egy kényelmes plusz gombra kattintva feladatot lehet létrehozni. Ezeken túl beállítható egy rendszeres időzítő az applikációban, amely push-to-notification-t küld a felhasználónak, ami tartalmazza a teendőit.



2. ábra: Házizz Droid teszt verziója

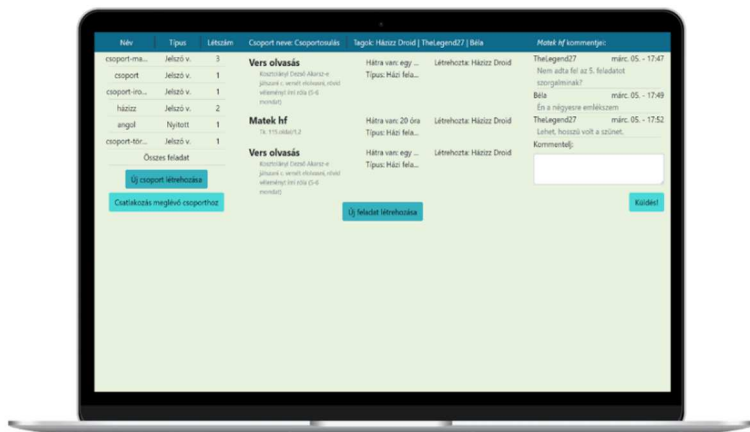
3.3. Házizz Mindenhol

A projekt irányító pultja a Házizz Mindenhol. A nevéhez híven ez egy webfelület, ahol a felhasználók egy helyen megtalálják mindent, amit használni szeretnének, és nem menük hada mögé elrejtve.

A web felület több JavaScript keretrendszert használ, melyek együttműködve adják meg a Házizz Mindenhol eszköztárát. Az adatok lekérdezése a jQuery könyvtár segítségével, AJAX-szal történik. Az adatok

megjelenítésében a Knockout keretrendszer segít, mely az MVVM modellt alkalmazza.

A felület kinézetéért egy közismert CSS/JavaScript keretrendszer, a Bootstrap felel. Ennek használatával könnyen lehet teljes mértékben reszponzív, mobile-first oldalakat készíteni.



3. ábra: Házizz Mindenhol első verziója

4. Elért eredmények

A projekten 2018 szeptemberében kezdtünk el dolgozni, kevés tapasztalattal, de annál több ambícióval.

A projekt első próbája a 2018. október végi határidő volt, amit mi tűztünk ki, hogy meglássuk, mire vagyunk képesek, ha valamit el szeretnénk érni. Addigra elkészült az első hivatalos alfa változata a Házizz Droid applikációnak, amihez rögtön kerestünk vállalkozó szellemű tesztelőket, akik azóta is lojálisan segítik a projektet visszajelzéseikkel. A projektnek mára már védjegyévé vált képét ekkor kaptuk meg az egyik tesztelőnkől, akinek emiatt nagyon hálásak vagyunk.

2019. január elején privát módon elérhetővé vált a Házizz Mindenhol első teszt verziója, ami azóta is rendszeres funkció frissítéseket kap.

Innovatív Képfilterek Machine Learning Technikákkal

The Stylers

Horváth István, Hugyik Kornél

Felkészítő tanár: Esztelecki Péter, Kőrösi Gábor

Bolyai Tehetséggondozó Gimnázium és Kollégium,

24400 Zenta, Posta utca 18

1. Bevezetés

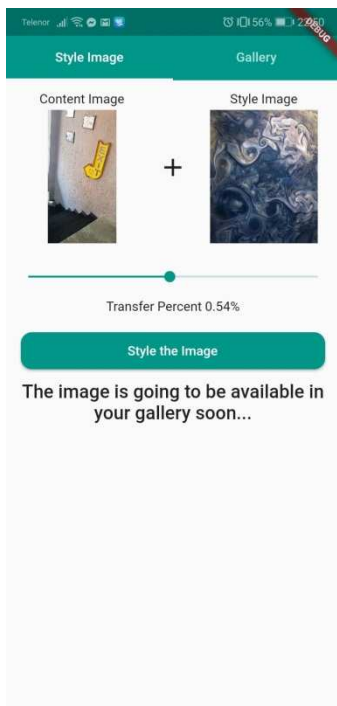
A közösségi médiákon megosztott képek átesnek valamilyen utófeldolgozási műveleten/műveleteken. „Filtereket” rakunk a képekre így tesszük őket érdekesebbé, látványosabbá. Egy egyszerű vagy unalmas képet is izgalmassá tehetünk akár egyszerű filterekkel is.

Manapság nagyon fontos egy közösségi médiában, hogy jó képeket oszthassunk meg. vegyünk csak a Snapchat példáját. Több innovatív ötlet mellett a teljesen újszerű filterek segítségével lett világhírű. Manapság több nagyobb közösségi médiának vannak a Snapchathez hasonló filtereik.

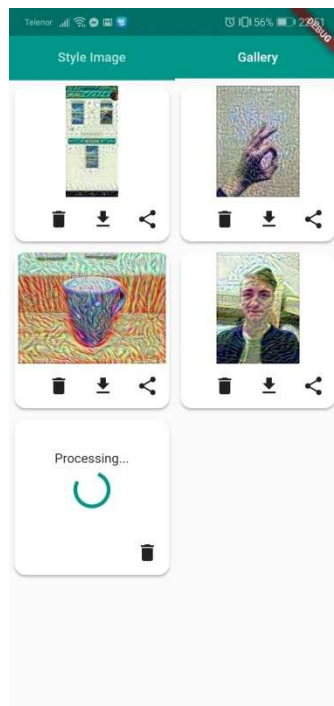
2. Probléma megoldásának menete

A kiválasztott tartalom és stílusképből egy tartalom reprezentációt készítünk egy előre betanított hálózat konvolúciós rétegeivel, majd később ezek a tartalom reprezentációk segítségével számoljuk ki a „hibát”. Azért fontos, hogy egy előre betanított Keras modellt használjunk (VGG19), mert „látja” azt, hogy mi van a képen, így számunkra jobban felhasználható hiba értéket kaphatunk. Ez azért fontos, mert nem használhatjuk fel az egyenkénti pixeleket a számításainkhoz. Amikor kiszámítottuk a hibát, Gradient Descent módszerrel Adam optimizert alkalmazva változtatunk egy kicsit a képünkön, azaz gradiens műveletet végzünk el rajta. Majd az említett lépéseket többször is elvégezzük és megkapjuk az új, egyéni filterekkel ellátott képünket.

3. Ábrák



1. ábra: Stílus és tartalom kép kiválasztása és feltöltése



2. ábra: Az előzőleg elkészült képek megjelenítése a galériában, majd innen a képek megosztása

4. Elért eredmények

Célunk egy olyan alkalmazást készíteni, ami bárkinek elérhetővé teszi a technológiát, könnyen használható és természetesen ingyenes, mivel úgy gondoljuk, hogy a megoldásunk megtalálná a helyét az iparban.

ANN_Generator

Laczkó Örs

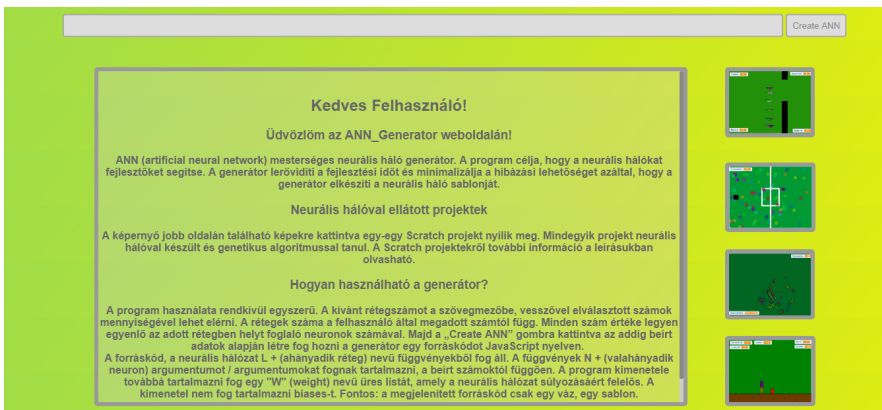
Felkészítő tanár: Kondorné Kovács Irén

Kerék Általános Iskola és Gimnázium,

1035 Budapest, Kerék utca 18-20.

1. Bevezetés

Számos mesterséges intelligenciával és/vagy neurális hálókval való foglalkozást megkönnyítő szoftver található az interneten, ilyenek például a TensorFlow és a PyTorch. A legtöbb ilyen alkalmazás használata a kezdő programozóknak rendkívül bonyolult, nehezen követhető. Ezen programokat leginkább a Python programnyelv támogatja. Ugyan magam is használom a Pythont, de jelenleg webfejlesztéssel foglalkozom. Számos projektemhez használok neurális hálókat, melyek magas neuron és/vagy réteg számánál a manuális programozás sok időt igényel, továbbá exponenciálisan nő a hibázás lehetősége. Ezen probléma megoldására gondoltam ki, majd fejlesztettem az ANN_Generator nevű webfelületet. Működése – ami az oldalon is olvasható – rendkívül egyszerű. Az oldal tetején lévő párbeszédablakba (vesszővel elválasztva, szóköz nélkül) be kell írni minden kívánt réteg neuron számát, például: [2,3,1]. Majd a „Create ANN” gombra kattintani, s megjelenik az oldalon elhelyezett szürkés alapú paragrafusban a program által generált forráskód (1 ábra).



1. ábra

A forráskód vágólapra helyezése, majd beillesztése után szabadon felhasználható számtalan projektben. Az ANN_Generátor web felületének jobb oldalán négy Scratch fejlesztőkörnyezetben megtekinthető projekt van elhelyezve. Mindegyik projekt neurális hálóval készült és genetikus algoritmussal tanul. A Scratch projektekről további információ a leírásukban olvasható (2. ábra).

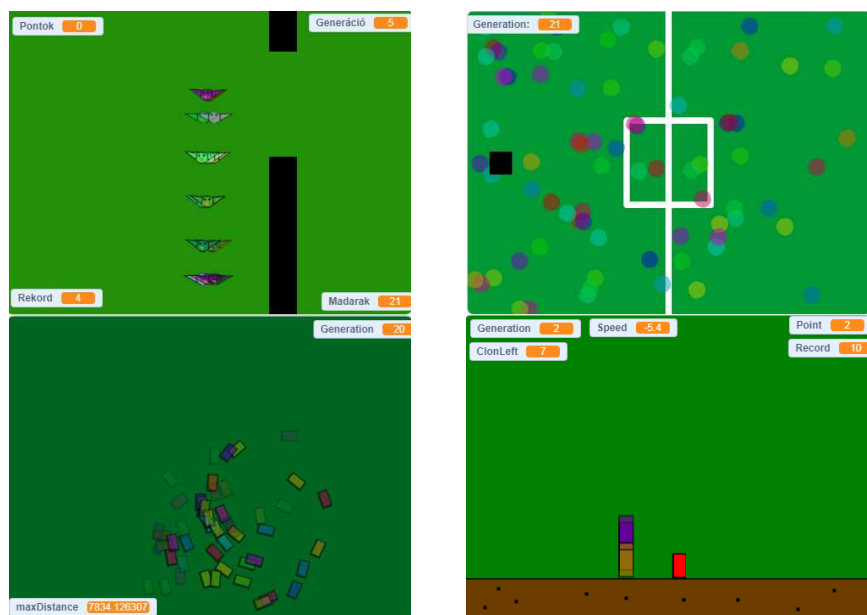
Elérhetőek az alábbi linkeken (AI – artificial intelligence):

AI bird <https://scratch.mit.edu/projects/238932969/>

AI football <https://scratch.mit.edu/projects/246768302/>

AI car <https://scratch.mit.edu/projects/249160965/>

AI rectangles <https://scratch.mit.edu/projects/239441108/>



2. ábra

Jelenleg is több projekten dolgozom, melyekhez neurális hálót használok, például egy adattömörítő program, valamint egy olyan alkalmazás, amely egy számsorba megkeresi annak rendezési szabályát, s az alapján fog következtetni a következő elemekre. Ezekben a projektekben óriási segítségemre van az ANN_Generátor.

2. A probléma megoldásának menete, az ANN_Generátor létrehozása

A probléma megoldásának az első lépése paradox módon a végeredmény volt. Először végig gondoltam, hogy milyen kimenetet szeretnék, hogy milyen

output lenne számomra az ideális. Ezt követően a program felhasználói interfészét terveztem meg, hogy milyen inputokra van szükség, illetve, hogy mi az ideális elosztásuk az adatfogadó és megjelenítő elemeknek. Miután minden terv elkészült, először a JavaScript kódot írtam meg, melynek az első része a felhasználói bemenet kezelése, feldolgozása. Működése: a feldolgozott adatlista egy összetett függvényrendszerbe kerül, ahol minden elem (a felhasználó által megadott szám) kap egy metódus fejléctet, argumentumot/argumentumokat, valamint egy metódus törzset. Az így kapott függvény stringek egy listába kerülnek. A függvények legenerálása után egy for ciklus végig lepkéd az elemeken és belesűríti egy változóba, majd megjeleníti az „output” id-vel ellátott paragrafusba a változó tartalmát (3. ábra).

```
var W = [];

function L0(N0, N1)
{
  L1(N0 * W[0], N1 * W[1], N0 * W[2], N1 * W[3], N0 * W[4], N1 * W[5]);
}

function L1(N0, N1, N2)
{
  L2(N0 * W[6], N1 * W[7], N2 * W[8]);
}

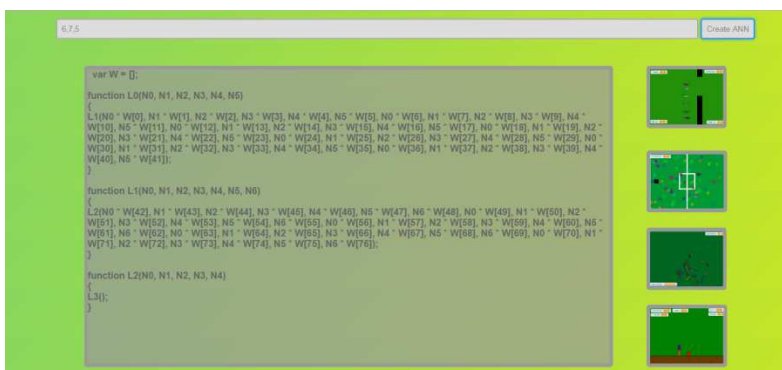
function L2(N0)
{
  L3();
}
```

3. ábra

Második lépés volt a weboldal formázásának megoldása, az oldal struktúrájának a kialakítása. Idetartoznak az elemek elhelyezése és méretezése, mértékük természetesen százalékban van megadva, így minden képernyőn ugyanolyan megjelenésű. Idetartoznak továbbá a stilisztikai elemek is, mint például a színek, betűtípusok, kiemelések, illetve a görgetősáv megformázása. A programablak tartalmaz még CSS animációkat is, ilyen például a színváltós háttér és az érintésre felvillanó elemek.

3. Elért eredmények

A program jelenlegi állapotában a felhasználótól kapott/beolvasott érvényes adatokat feldolgozza és eredményül hibátlan JavaScript kódot generál a kapott adatok alapján. A program hatékonysága a neurális háló méretével nő. Például egy 6 input 7 hidden-layer 5 output neuron méretű neurális háló legenerálásához a következőt kell beírni a szöveglapba: 6,7,5 (4. ábra). Eredményül az 1. táblázatban található forráskód keletkezik.



4. ábra

1. Generált forráskód
<pre> var W = []; function L0(N0, N1, N2, N3, N4, N5) { L1(N0 * W[0], N1 * W[1], N2 * W[2], N3 * W[3], N4 * W[4], N5 * W[5], N0 * W[6], N1 * W[7], N2 * W[8], N3 * W[9], N4 * W[10], N5 * W[11], N0 * W[12], N1 * W[13], N2 * W[14], N3 * W[15], N4 * W[16], N5 * W[17], N0 * W[18], N1 * W[19], N2 * W[20], N3 * W[21], N4 * W[22], N5 * W[23], N0 * W[24], N1 * W[25], N2 * W[26], N3 * W[27], N4 * W[28], N5 * W[29], N0 * W[30], N1 * W[31], N2 * W[32], N3 * W[33], N4 * W[34], N5 * W[35], N0 * W[36], N1 * W[37], N2 * W[38], N3 * W[39], N4 * W[40], N5 * W[41]); } function L1(N0, N1, N2, N3, N4, N5, N6) { L2(N0 * W[42], N1 * W[43], N2 * W[44], N3 * W[45], N4 * W[46], N5 * W[47], N6 * W[48], N0 * W[49], N1 * W[50], N2 * W[51], N3 * W[52], N4 * W[53], N5 * W[54], N6 * W[55], N0 * W[56], N1 * W[57], N2 * W[58], N3 * W[59], N4 * W[60], N5 * W[61], N6 * W[62], N0 * W[63], N1 * W[64], N2 * W[65], N3 * W[66], N4 * W[67], N5 * W[68], N6 * W[69], N0 * W[70], N1 * W[71], N2 * W[72], N3 * W[73], N4 * W[74], N5 * W[75], N6 * W[76]); } function L2(N0, N1, N2, N3, N4) { L3(); } </pre>

1. táblázat

A projektekben kedvenc témámmal foglalkoztam, a mesterséges intelligenciával. Az MI fejlesztésében látom mind magam, mind az emberiség jövőjét. Hiszek abban, hogy ennek a projektnek köszönhetően mások is könnyebben, kevesebb idő ráfordítással készítenek neurális hálókat tartalmazó programokat.

SentiVision

Sentimentalists

Horváth István, Kónya Leon

Felkészítő tanár: Kőrösi Gábor

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

1. Bevezetés

Célunk egy olyan eszköz kifejlesztése, elkészítése volt, amely képfeldolgozás segítségével könnyíti a vakok illetve látássérültek mindennapi funkcióit, konkrétan az érzelem felismerést. Egy olyan mobilapplikáció volt a vízióink, amely a felhasználó társalgási partnerének arckifejezéseit elemzi ki valós időben, és annak pillanatnyi érzelméről adott becslését közli a felhasználóval nem zavaró hanghatás formájában.

A vízióink ötletét olyan kísérletek adták, amelyek a vak vagy látássérült illető által viselt kamera felvételeit próbálták hang formájába önteni, hogy azok segítsék őt mindennapi tevékenységei hátránytalan végzésében. Viszont ezek nem arattak nagyobb sikert, és a képfeldolgozó algoritmusaik aránylag primitívek voltak.

Az utóbbi évtizedben azonban robbanásszerűen növekedni kezdett a GPU-ipar, így elterjedtebbek lettek a nyílt forráskódú, neurális háló alapú gépi látási algoritmusok. Ennek előnyét számos tech-óriás kihasználja, kommerciális szinten és ipari szinten is (pl. önvezető autók).

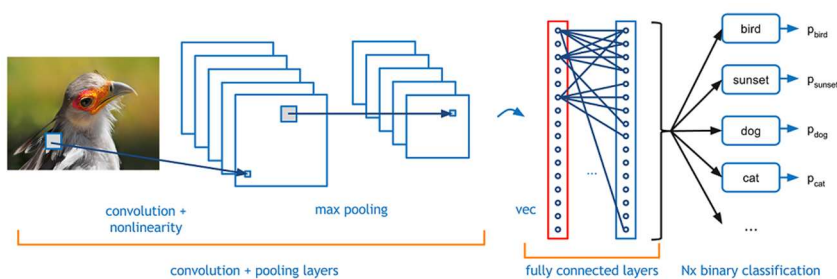
Ez a két gondolatmenet összekombinálása adta számunka az ötletet, ugyanis nem találkoztunk még vakokat segítő konvolúciós neurális hálókkal.

2. Probléma megoldásának menete

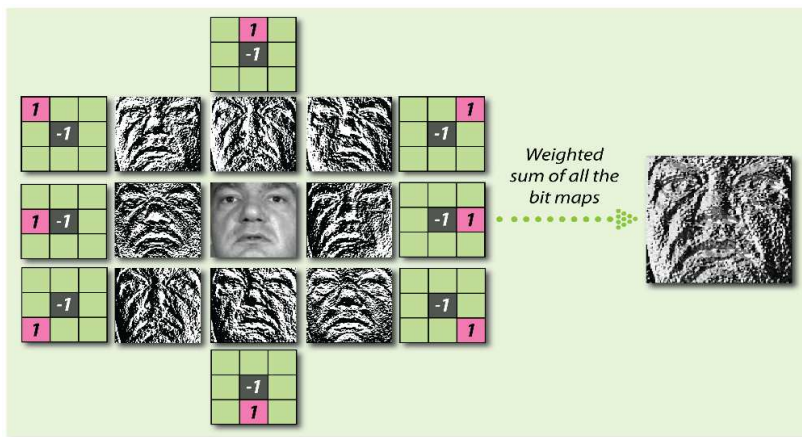
A munkánk kivitelezése a következő főbb problémák megoldását igényelte: az arcelemző algoritmusunk kidolgozása során értelmetlennek láttuk a 0-ról indulást, így egy meglévő, előre betanított CNN (Convolutional Neural Net) architektúrát kellett találnunk, amely elég kompakt ahhoz, hogy egy okostelefon futtatni tudja valós időben. A MobileNetre esett a választásunk. Ezek után egy hasonlóan kompakt arckifejezés-adatbázist kellett találnunk, amelyen az eredetileg ImageNetre tanított MobileNetet „újratanítjuk”. Ez a FER2013-as adatbázis lett. Python környezetben, Tensorflow könyvtár segítségével lett a feladat elvégezve. Az applikáció Firebase platform segítségével lett létrehozva.

2.1. Konvolúciós neurális hálók az arcfelismerésben

A konvolúciós neurális hálók működési elve nagyvonalakban az 1. ábrán látható. Tradicionális neurális hálókkal szemben előnyös képfeldolgozásnál, mivel úgynevezett „konvolúciós kerneleket” használ: $n \times n$ -es súlyozott mátrixokat, amelyeknek a bemeneti kép minden $n \times n$ -es szegmensének és a kernelnek skaláris szorzata által alkotott feldolgozott képet adják tovább a hálózat további rétegeinek. A 2. ábra szemlélteti, hogy mi is történik egy konvolúciós művelet során.



1. ábra: A konvolúciós hálózatok



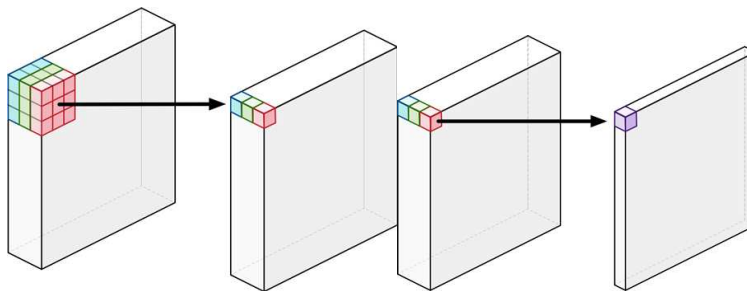
2. ábra: Különböző konvolúciós kernelek

Látható, hogy különbözőképpen súlyozott kernelek különböző tulajdonságokat emelnek ki egy képen. Azt pedig, hogy mely tulajdonságok lesznek kiemelve, maga a hálózat alakítja ki, az alapján, hogy mi a legoptimálisabb neki a tanulási folyamat során, példák alapján.

Mint ahogyan a sztandard neurális hálózatoknál, úgy a konvolúciósaknál is gradiens süllyedéssel illetve hátrapropagációval történik a tanulás. Ez gyakorlatilag kategóriába való osztályozás során egy hibafüggvény definiálásával kezdődik a hálózat összes paraméterének függvényében, amelynek kimenetelét minimalizálja Newton numerikus módszereivel, és ez alapján változtatja a hálózat összes paraméterét a hibafüggvény minimuma felé.

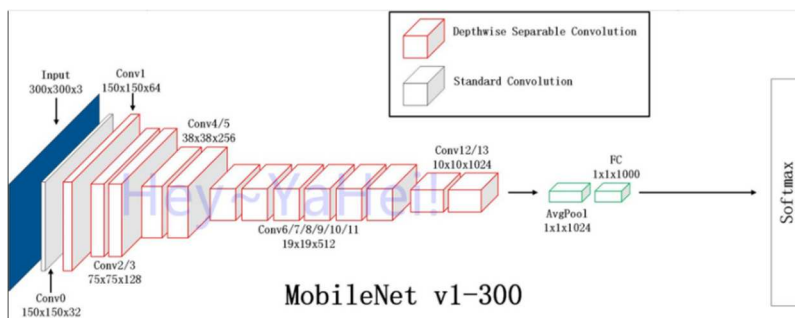
2.2. Mobilenet

A Mobilenet tökéletes gépi látási architektúra mobilos alkalmazásokhoz, ugyanis kisebb kapacitású számításoknál ugyanannyi paramétermennyiség esetén (Mobilenet-nél 4.24 millió) az eredeti teljesítménynek többszörösét is el tudja érni egy pár egyszerű trükkel, köztük a mélységi és a pontbeli konvolúciók szétválasztásával (a 3. dimenzió a színsatorna), amint az a 3. ábrán is látható.



3. ábra: A két konvolúció szétválasztása

Maga a Mobilenet architektúrája a 4. ábrán látható.



4. ábra: Mobilnet architektúrája

A hálózat eredetileg egy sokkategóriás képosztályozó kategóriára lett betanítva, amely ismét ideális nekünk, ugyanis feltételezhetjük, hogy sok általános tulajdonságot elsajátított a tanulás során (ImageNet konvolúciós súlyok), amelyeket újratanításkor csak nagyon minimálisan kell módosítani, hogy jól tudjon osztályozni.

2.3. Újratanítás után

Újratanítás után elkészítettük a confusion matrix-et, amely 0-tól 3-ig indexelt soraiba és oszlopaiba az undor, neutrális, mosolygó illetve meglepődött arckifejezéseket kell behelyettesíteni. Ebben a mátrixban (5. ábra) leolvasható, hogy mely arckifejezéseket mekkora számban tévesztett a hálózat össze teszteléskor melyekkel.

```
Confusion Matrix
[[ 1   2  45   7]
 [ 1  69 230 326]
 [ 22 463 288 106]
 [ 0  25 342  49]]
```

5. ábra: Confusion Matrix az újratanított Mobilenethez

Első látásra nem tűnik úgy, hogy nagy precizitása lenne a hálózatnak, viszont az applikáció működésben rezponzív legalább 3 arckifejezésre. Nagyobb, professzionálisabb adatbázisokkal ez orvosolható, a célunk egyelőre a rezponzív működés volt.

3. Elért eredmények

Eddigi eredményeink arra utalnak, hogy az általunk elképzelt eszköz kivitelezhető, életképes. Elegendő adattal számos hiányosság pótolható, ezért a jövőben szert szeretnénk tenni akadémiai adatbázisokra, több arckifejezés bevonásával.

Az eszközben főleg azért látunk potenciált, mert nem verseng a vakvezető kutyával, illetve a bottal sem. Ezeket évtizedek óta próbálják szakemberek lecserélni, korszerűsíteni, ám nem lehet őket legyőzni. Éppen ezért a SentiVision ezeket kiegészíteni kívánja, egy olyan funkciót ad az ember kezébe, amelyet a bot és kutya nem tudnak megadni.

Ennek az innovatív funkciónak köszönhetően feltételezni lehet, hogy termék formájában lenne piaca az erre igényt tartók körében. Számtalanszor könnyebb lehet egy konverzáció, hogyha a vak ember tisztában van a társalgási partnere nonverbális jeleivel – végülis a testbeszéd többet mond minden szónál.

Graphite – Relációs adatanalízis gráfokkal a felhőben

Lévai Zsolt Bálint

Felkészítő tanár: Bärnkopf Bence

Városmajori Gimnázium,

1122, Budapest, Városmajor u. 71

1. Bevezetés

A jelenleg elérhető adatanalizáló eszközök nem fordítanak kellő figyelmet az adatok közt fennálló kapcsolatok, azok egymásra vonatkozó relációjának megjelenítésére, így a sokszor legértékesebbnek mondható információk kerülnek látókörön kívül. Projektem többek között ezen problémát célozza megoldani; olyan felületet kínál adathalmazokhoz, melyben az adatelemek relációja elsődleges szerepet kap. Ez ideális számos területen előkerülő problémák esetén; legyen szó szociális hálókról, számítógépes hálózatokról, üzleti menedzsment megoldásokról vagy logisztikai rendszerekről.

Továbbá az eddigi megoldások körében megoldatlan problémának bizonyult az egyidejű munkavégzés. Míg a klasszikus dokumentumok csoportos szerkesztésére számos lehetőség létezik, addig ez nem megoldott az adatelemzés terén; ezzel jelentősen lassítva a munkavégzést.

2. Probléma megoldásának menete

2.1. A gráfmotor; *Graphite Core*

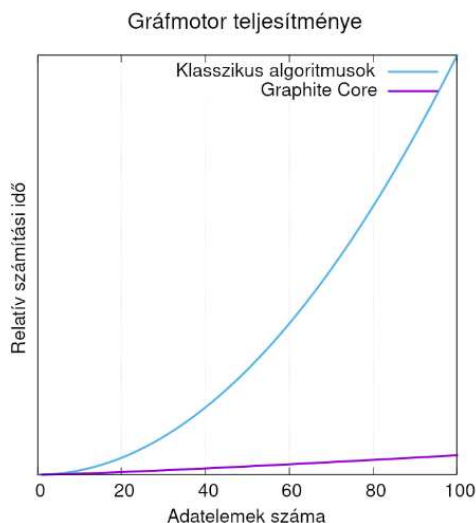
A projekt magját képező, fizikai együttműködésekkel alapuló, erő-vezérelt gráfok kezelésére alkalmas gráfmotor a *Graphite Core*. Tervezése során elsődleges szempont volt a nagy teljesítmény elérése, valamint az algoritmusok maximális párhuzamosíthatósága, és szálasíthatósága. Ennek érdekében a *Graphite Core* egyebek mellett egy saját, parallelizált, Barnes-Hut szimulációs algoritmus implementációt alkalmaz, amely az átlagos n -test fizikai szimulációkkal szemben messzemenőig gyorsabbnak és kisebb számítási komplexitásúnak bizonyul, amely az 1. ábrán is látható.

Míg a klasszikus fizikai motorok algoritmusai (1) komplexitásúak, addig a *Graphite Core* által használt ezeknél jelentősen kevesebb, (2) komplexitású.

$$O(n^2) \quad (1)$$

$$O(n \times \log n) \quad (2)$$

Ennek eredménye, hogy a gráfok valós idejű kezelése több ezer adatelem esetén is gyors és használható marad.



1. ábra: A *Graphite Core* relatív teljesítménye az átlagos számítási idő szerint

A teljesítmény maximalizálása mellett kiemelt szerepet kapott a gráfmotor fejlesztése során annak bővíthetősége, modularizálása. Ezt példázza többek között annak saját interfész nyelve, amely a megszokott szűrő query-ken kívül számos egyéb feladat ellátására képes; hozzáférést nyújt a felhasználók számára a *Graphite Core* teljes API felületéhez.

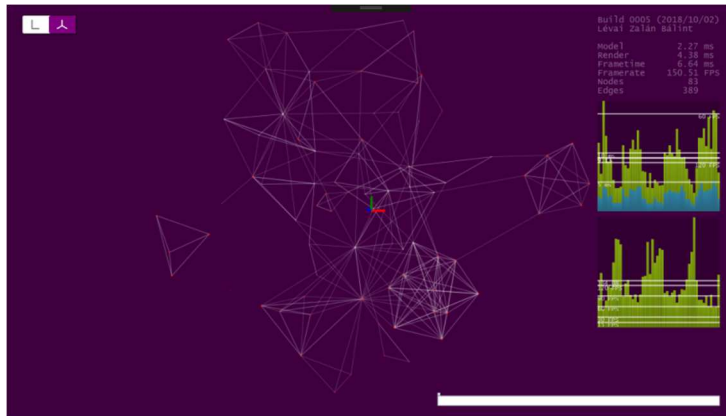
2.2. Az első prototípus

A projekt egy Windows-on futtatható kliens alkalmazásként öltött először formát, melynek fejlesztése 2018 októberében kezdődött. Ekkor ment végbe a gráfmotor fejlesztésének jelentős része is.

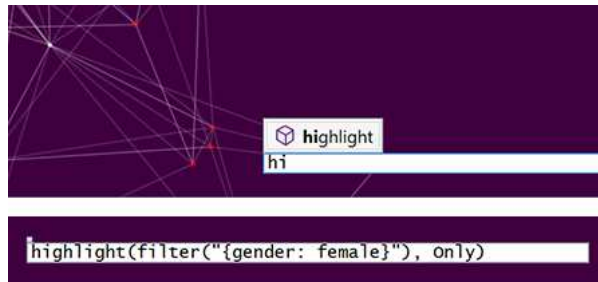
Ez, a .NET keretrendszer és a Windows Presentation Foundation (WPF) segítségével megírt prototípus számos funkcionalitásnak adott teret. Megjelennek benne a máig is használt kettő- és többdimenziós gráf nézetek (2. ábra), amelyek a böngészést és a szűrést segítik elő.

Továbbá lehetőséget nyújt a *Graphite Core* által nyújtott interfész nyelvhez tartozó parancsértelmező használatára. Mindezt kiegészíti a programfejlesztő környezetekből ismertekhez hasonló intelligens, kontextuális parancs javaslatokkal (3. ábra).

Bár később a kliens oldalon futtatott számításokra vonatkozó terveket felváltotta egy felhő alapú számítási modell - többek között a kliens gépek leterheltségének csökkentése érdekében -, az itt feltüntetett funkcionalitás meghatározó volt a további fejlesztésekre nézve is.



2. ábra: 3D-s gráfnézet



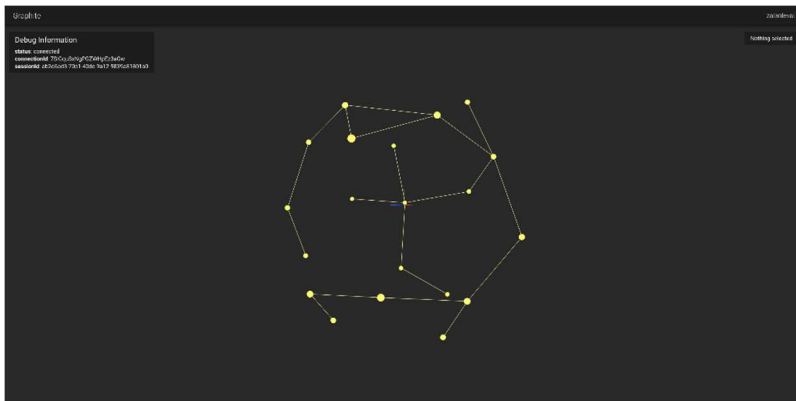
3. ábra: Parancsértelmező kontextuális javaslatokkal

2.3. Felhő és Web

A projekt fejlesztésének következő és jelenlegi állomását a szerver oldali számításokon alapuló webes alkalmazás képezi. Erre a változtatásra 2019 januárjában került sor. Ennek a változtatásnak részben a platform függetlenség hiánya, a helyi alkalmazások telepítésével járó komplikációk és költségek elkerülése, valamint a korábban említett együttműködési lehetőségek felmerülése adott okot. A felhő alapú modell ezekre mind megoldást nyújt és lehetőséget ad az együttműködési lehetőségek megvalósítására.

A szerver oldali megvalósításhoz az ASP.NET Core keretrendszer használatával került sor. Az architektúrális váltás megvalósításának legnehezebb lépése a szerver és kliens kommunikációja volt; erre megoldást nyújtott a WebSocket alapú SignalR protokoll, amely lehetővé teszi az alkalmazás által elvárt magas frissítési rátájú és közel valós idejű kommunikációt a felek között.

Ezzel lehetővé vált a korábban említett egyidejű munkamenetek megvalósítása.



4. ábra: Web szerkesztő felülete

A kliens oldalon egy modern alapokra épülő webalkalmazás működik. Megvalósítására a React komponens könyvtár és a TypeScript nyelv segítségével kerül sor; a szerkesztő pedig a WebGL-en alapszik.

3. Elért eredmények

A webes platform jelenlegi állapotában közel funkció kompatibilis az eredeti kliens alkalmazással, valamint bizonyos területeken messzemenőig meghaladja azt. Képes teljesen valós időben a munkamenetek kezelésére, a gráfokon végzett módosítások és egyéb műveletek szinkronban tartására.

Természetesen a projekt korai fázisa, valamint egyedüli fejlesztése miatt az analízis eszközök még számottevő része nem került megvalósításra, valamint a felhasználói felület is sok területen hiányokban szenved. Mindennek ellenére ezek jelentős részének technológiai háttere már biztosított a kódbázis jelenlegi formájában is.

A Graphite sok szempontból egyedülálló szoftver a kategóriájában; a korábban nehezen vizsgálható információk és összeköttetések kinyerésére ad lehetőséget, amely számos területen hasznos eszközzé teszi. A kódbázis is ezt reflektálja; könnyedén bővíthető és modularizált.

JegyKalk

KGHSZ

Kocsis Kamilla, Cseri Gábor

Felkészítő tanár: Brósch Éva

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

1. Bevezetés

Szorgalmas, mindenre felkészült; csendes vagy éppen hangos; sosem csinál házi feladatot, vagy majd óra előtt, éppenséggel a buszon ülve; aki egy ötös alát is ki szeretne javítani, de olyan is akad, aki még az egyest is alig, a diák az diák. És minden tanuló életében megtörténik az, mikor a jegyeire kell néznie.

Valaki úgy szereti, ha napra pontosan tudja átlagát, de akadnak olyanok is, akik félév, évvége utolsó hetén akar csak bosszankodni ezen. Ez mind nem számít, hisz mi mindenkinek segíteni próbálunk, akiknek tudását jegyekben méri.

Bevalljuk őszintén még a legelején: bármennyire is egyszerű, nem szeretünk átlagot számítani. Ami nem annak köszönhető, mert a jegyeink csak nagy ritkán akarnak az ötösre hasonlítani, hanem mert unalmas és hosszadalmas. Mennyire egyszerű lenne, ha esténként nem számológéppel a kezünkben kéne számolgatnunk az érdemjegyeink átlagát, kiderítve így, hogy mennyit kell még letenni az asztalra, hogy elérjük az elvárt szinteket. Helyette inkább egy forró teával a kezünkben beülnénk a kényelmes fotelba, egy izgalmas kalandregénnyel és nem hagynánk hátra a telefonunkat sem. Ugyanis ezt a borzalmasan unalmas és fárasztó munkát elvégzi a telefonunkon található alkalmazás is.

2. Probléma megoldásának menete

A kitűzött cél egy könnyen kezelhető, praktikus és főként számunkra soha nem látott program elkészítése volt. Kezdetben két fő feladat megvalósítását terveztük el, ezek a számolások megírása, tesztelése, valamint az adatbázis felépítése volt. Ezt a két feladatot felosztottuk egymás közt.

2.1. Számolások

A matematikai számolásokat először Netbeansben írtuk meg, ekkor még konzolon keresztül írtuk ki és kértük be a szükséges adatokat.

2.2. Adatbázis

A programunk adatbázisát SQLite-ban írtuk. Az adatbázis megírása során törekedtünk arra, hogy kevés tárhelyet foglaljon. Igyekeztünk elkerülni benne a redundanciát és a vele járó anomáliákat.

2.3. A program működése

A JegyKalk főmenüjében 4 opció közül választhatunk. Mindegyik lehetőség különböző feladatok ellátására szolgál.

2.3.1. Hozzáadás menüpont

A program működésének megkezdése előtt szükséges felvenni azokat a tantárgyakat és a hozzá fűződő jegyeket, dolgozatokat, amelyeket szeretnénk, ha tartalmazna a program adatbázisa. Azonban a JegyKalkban nem szükséges azonnal megadnunk ezeket az adatokat, ahhoz, hogy elmentésre kerüljön a tantárgy.



1. ábra: Hozzáadás kinézete

2.3.2. Gyors Számlálás menüpont

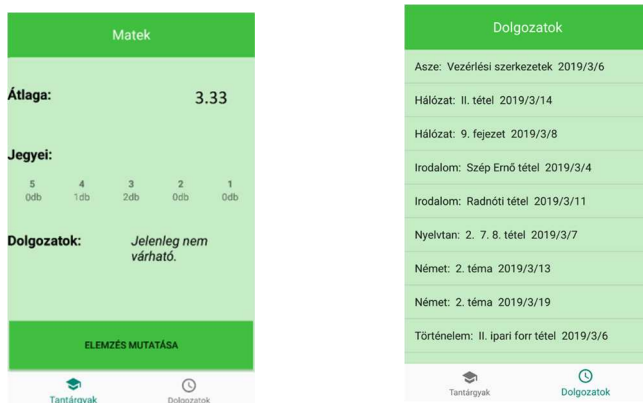
Mi történik akkor, ha valaki nem akarja napról napra végigvezetni tanulmányait, viszont szeretné tudni, hogy milyen közel vagy éppen távol áll ahhoz, hogy átlaga megváltozzon? Erre a problémára is megoldást találtunk ebben a menüpontban. Ha a felhasználó megadja a számolásokhoz szükséges adatokat, egy kattintással megtudhatja a választ égető kérdésére.



2. ábra: Gyors számlázás kinézete

2.3.3. Jegyek és Dolgozatok menüpont

Ebben a menüpontban tekinthetőek meg a már felvett tantárgyak jegyei és a hozzá kapcsolódó dolgozatok listája. Ha kiválasszuk a tantárgy nevét a listában, megjeleníthető az adott tantárgyhoz fűződő adatok, ha pedig az Elemzés gombra kattintunk, azokat az adatokat láthatjuk, amelyen az egész program fő célja és haszna nyugszik. Ha a Dolgozatok almenüt válasszuk, kilistázva tekinthető meg a ránk váró dolgozatok, valamint ahhoz a tantárgyhoz tartozó célkitűzés és átlag.



3. ábra: Jegyek és Dolgozatok kinézete

2.3.4. Törlés menüpont

A felhasználó a Törlés menüpontban ismét kilistázva a felvett adatokat a tantárgy fontosságának csökkenése vagy éppen a tantárgy megszűnése miatt könnyen és egyszerűen, egy kattintással kitörölheti.

2.4. Nehézségek

Számunkra az Android Studio teljesen ismeretlen volt, így nem tagadás, hogy számos nehézséggel kellett megküzdenünk, valamint ennyire összetett adatbázist se állítottunk még össze. Annak ellenére, hogy a program megírásához nélkülözhetetlen tudásnak csak az egyik részével: az objektum-orientált programozással voltunk tisztában, úgy gondoljuk sikerült az elvárásoknak megfelelően teljesíteni.

Az első nehézségeket az adatbázis és a felület összekötése jelentette, majd az, hogy a Fragmentek megfelelően kommunikáljanak egymással. Ehhez a Bundle-t használtuk.

A következő nehézséget a felület kialakításában találtuk, hisz fontos, hogy jól érthető és könnyen kezelhető legyen.

3. Elért eredmények

Az első hetekben autodidakta módon igyekeztük megtanítani magunkat az Android Studio használatára, ennek köszönhetően rengeteg egyéb programot megírtunk a könnyebb megérthetőség végett. Úgy gondoljuk megérte minden fáradságot, hisz fontos szempontjaink közé tartozott a könnyebb elérhetőség, így jó ötlet volt a mobiltelefonokat megcélzni. Mindazok mellett, hogy fontosnak tartjuk a kinézetét és az egyszerű kialakítását, a programot igyekeztünk a legegyszerűbben megoldani. Noha a program alapja az átlagszámítás, annál sokkal több mindent képes kiszámolni és erre is szerettük volna a figyelmet irányítani. A tantárgy jegyeinek elemzésével olyan fontos dolgokra figyelhetünk, mint az átlag rontásának veszélye, a javítás közelsége, valamint a dolgozatok időpontjának lementésével a felkészülés észben tartása is könnyebbé válik.

Mind a hibák észrevételét és a felület egyszerűbb felépítését diákoknak köszönhetjük, akik mai napig rendszeresen használják programunkat, és mindig igyekszünk az ők véleményük alapján módosítani azt.

Visszatekintve az elmúlt hónapokra, számos fontos dolgot tanultunk, mind az objektum-orientált programozásban, mind a megbeszélések, ötletcserék szervezésében és tovább erősíthettük a csapatmunkában való képességeinket. Úgy gondoljuk, programunk többet adott, mint lexikális tudás, hisz jó érzés olyat készíteni, amire környezetünkben nagy szükség van.

Együttműködő partnereink

Együttműködő partnereink



National Instruments
hungary.ni.com



Epam
www.epam.hu



3i
www.3i.hu



Ultinous
www.ultinous.com



NNG
www.nng.com



TEConcept Hungary Kft.
www.teconcept.hu



Optin Kft.
www.optin.hu



Digital Kft.
www.digital.co.hu



Creativ_IT
www.creativit.hu



Evosoft
evosoft.hu



Antavo
antavo.com



LogMeIn
www.logmein.hu



CAS Software
www.cas-software.hu



EGYÜTT
A JÖVŐ MÉRNÖKEIÉRT
SZÖVETSÉG



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

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

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

A program a "A 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: 150 példányban