

Szegedi Innovatív Informatika Verseny

2024



SZIIV Program (2024. április 19. – péntek)

8:30–9:00	Regisztráció		
9:00–9:15	Megnyitó		
Informatikai szekció előadások és projektbemutató		Műszaki informatikai szekció előadások	
9:20–9:50	A Backstage Bajnokai: SGxGacha	9:20–9:40	E.R.SZ.: Kételtű drón
9:50–10:20	Polyp: Avatar Projekt	9:40–10:00	Bits&Recyclers: QR-kód alapú hulladékszállító rendszer
10:20–10:50	VDF Dev: Virtual Front Desk, a virtuális repció szoftver	10:00–10:20	Három a magyar igazság! Trinár alapú számítógép
10:50–11:10	KÁVÉSZÜNET	10:20–10:40	Prototech Technologies: Kontaktmentes anyagvizsgáló fémekhez és mágnesekhez
11:10–11:40	NemCsoport: LovassyApp – Tanulást ösztönző rendszer	10:40–11:00	Infosok: Cérna Mutatvány
11:40–12:10	Team Algorithm: Algorithm	11:00–11:20	KÁVÉSZÜNET
12:10–12:40	NemCsoport: Hermes – Sportversenyek szervezése és lebonyolítása egyszerűen	11:20–11:40	Polyp: Farfler – Önvezető kerekeszkék fejlesztése
12:40–13:40	EBÉD	11:40–12:00	Team Komposzt: Komposztkezelés automatizálása és szabályozása
13:40–14:10	Byte brigád: BáttyaBoss	12:00–12:20	Team HardTFM: HardTFM – Egy modern Total Fluid Management rendszer az Ipar 4.0-hoz
14:10–14:40	eredet.app: Eredet: Szövegek eredetiségének ellenőrzése mesterséges intelligencia használatával	12:20–12:40	autodino: Autó diagnosztika
14:40–15:10	Ailurus: GraphBox	12:40–13:40	EBÉD
15:10–15:40	Otthonos okos: Okosotthon platform Node- RED-ben	13:40–14:10	Projektbemutató előkészületek
13:30–14:00	Projektbemutató		
16:00–17:00	Eredményhirdetés		

Tartalomjegyzék

<i>SZIIV</i>	1
<i>Tartalomjegyzék</i>	2
<i>Előszó</i>	4
<i>Programbizottság</i>	6
<i>Informatikai szekció</i>	9
Benkó Barnabás, Zsikai Eszter:	
SG×Gacha.....	10
Bacsur Dániel, Vajda Ádám:	
Avatar Projekt – Metaemberek az oktatásban.....	14
Horváth Zsombor:	
Virtual Front Desk, a virtuális recepció szoftver	18
Ocskó Nándor:	
LovassyApp – Tanulást ösztönző rendszer.....	22
Váradi Márk, Király Márk, Vodenicsár Rajmund:	
Algorithm	26
Ocskó Nándor:	
Hermes – Sportversenyek szervezése és lebonyolítása egyszerűen	30
Véber-Jurassa Márk:	
BáttyaBoss	34
Józsa Kristóf:	
Eredet: Szövegek eredetiségének ellenőrzése mesterséges intelligencia használatával	38
Kurucz Kristóf:	
GraphBox	
Patkós Máté:	
Okosotthon platform Node-RED-ben	
<i>Műszaki informatika szekció</i>	51

Kucsera István, Szántai Ádám:	
Kételtű drón	52
Bartha Jenő:	
QR-kód alapú hulladékkezelő rendszer	56
Bábel Edvárd István:	
Trináris alapú számítógép	60
Burger Alfréd:	
Kontaktmentes anyagvizsgáló fémekhez és mágnesekhez	64
Kun-Baumann Bálint:	
Cérna Mutatvány	68
Vajda Ádám, Bacsur Dániel:	
Farfler – Önvezető kerekesszék fejlesztése	72
Preisler András, Szigeti Szabolcs:	
Komposztkazán automatizálása és szabályozása	76
Paróczai Olivér:	
HardTFM – Egy modern Total Fluid Management rendszer az Ipar 4.0-hoz	82
Bence Béla:	
Autó diagnosztika	86
<i>Együttműködő partnereink</i>	<i>91</i>

Előszó

A Szegedi Innovatív Informatika Verseny 2024-ban tizedik alkalommal kerül megrendezésre, ahol a középiskolás diákok mutathatják be megvalósított innovatív ötleteiket. A versenyt a Szegedi Tudományegyetem Természettudományi és Informatikai Karán belül az Informatikai Intézet indította útjára.

A verseny keretein belül ezúttal sem kötöttük meg a diákok kezét és kreativitását, bármilyen, őket foglalkoztató informatikai témájú pályamunkával nevezhettek az alábbi két szekció valamelyikére:

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

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

Az idei versenyre összesen 31 pályamunka érkezett be. Versenyünk kétfordulós, az első fordulóra beérkezett pályamunkából a zsűri az informatikai szekcióban 10, a műszaki informatikai szekcióban 9 pályamunkát benyújtó csapatok ad lehetőséget, hogy a döntőben bemutassák elért eredményeiket. Remélhetőleg idén ismét élőben találkozhatunk a diákokkal és ismerhetjük meg alkotásaikat.

A rendezvényt idén is számos neves informatikai és műszaki cég támogatta, amelyért ezúton is hálásak vagyunk. A zsűri által legjobbnak ítélt pályamunkákat értékes és hasznos díjakkal jutalmazzuk. A verseny szervezői számára kiemelten fontos, hogy ne csupán a csapatok tagjait és felkészítő személyeit, de iskoláit is díjazzuk, hogy ily módon támogassuk tehetséggondozó tevékenységüket.

Szeged, 2024. április

Dr. Németh Gábor
a SZIIV verseny szervezője

Programbizottság

Műszaki informatika szekció

A zsűri elnöke:



Dr. Pletl Szilveszter
főiskolai tanár
SZTE Műszaki Informatika Tanszék

A zsűri tagjai:



Dr. Mingesz Róbert
adjunktus
SZTE Műszaki Informatika Tanszék



Dr. Kincses Zoltán
adjunktus
SZTE Műszaki Informatika Tanszék



Dr. Balla Zsolt
üzleti elemző
4iG



Dr. Kelemen András Félix
középiskolai tanár (informatika szak)
Radnóti Miklós Kísérleti Gimnázium, Szeged

Informatika szekció

A zsűri elnöke:



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

A zsűri tagjai:



Dr. Kőrösi Gábor
adjunktus
SZTE Számítógépes Algoritmusok és Mesterséges
Intelligencia Tanszék



Dr. Pflanzner Tamás
adjunktus
SZTE Szoftverfejlesztés Tanszék



Szabó Csaba
full stack developer
ExxonMobil Hungary Kft.



Fodor Zsolt
középiskolai tanár (matematika-fizika-informatika szak)
Baranya Megyei SZC Komlói Technikum, Komló

Szervezők



Dr. Nagy Antal
egyetemi docens, intézetvezető helyettes
SZTE Képfeldolgozás és Számítógépes Grafika Tanszék



Dr. Németh Gábor
adjunktus
SZTE Képfeldolgozás és Számítógépes Grafika Tanszék



Dr. Kincses Zoltán
adjunktus
SZTE Műszaki Informatika Tanszék



Kun Eszter
közkapcsolati szakértő
SZTE Informatikai Intézet

Informatikai szekció

SG×Gacha***A Backstage Bajnokai****Benkó Barnabás, Zsikai Eszter**Felkészítők: Bíró Zsolt, Dr. Németh Tamás**SZTE Gyakorló Gimnázium és Általános Iskola, Szeged, Szentháromság utca 2.***1. Bevezetés**

Az iskolánkba járó diákok közötti összetartó kapocs az évek során gyengülni látszik. Ez a probléma nagyrészt a közösségi tevékenységek háttérbe szorulásából és a fiatalok mindennapos médiahasználatában történő változásokból fakad.

A diákok a szünetekben kevesebb időt töltenek együtt, sokan magukba fordulnak, a közösségi médián ütik el az időt. A magány érzete az iskolákba is beszivárog. Erre a problémára fejlesztettük ki a tökéletes megoldást: egy online webes kártyajáték, ahol a kártyák az iskolai közösség fő mozgatórugói, a tanárok és a diákok. Mivel a karakterek csak az iskolában tanulóknak ismerősek, a játékot exkluzívan az iskola zárt közössége fogja játszani, ezzel erősítve az összetartó hatást. A meccsek könnyen elindíthatók bármely aktív játékosal, illetve olyan hosszúságúak, hogy egy szünetbe mindig beleférjenek.

2. Probléma megoldásának menete

A pályamunkát szeptember eleje óta fejlesztjük, természetesen számtalan problémába ütköztünk azóta. Ez az első projektmunkánk, amely a Svelte keretrendszerrel használja, szerverrel kommunikál és rengeteg egyedi illusztrációt és hanganyagot igényel. Az 1. ábrán a kártyák hátoldalán található illusztráció látható.



1. ábra: A kártyák hátlapjára készült illusztráció

2.1. Ötletelés

Kezdetben egy kooperációs játék készítésén gondolkodtunk, de még elég korán irányt váltottunk, mert úgy véltük a PvP formátum több lehetséges felhasználó ízlésének felelne meg. Az osztályunk jelentős része szereti a kártyajátékokat, így erre esett a választásunk. Tudtuk, hogy valami különlegeset kell alkotnunk, hogy lojális és aktív játékosközösséget tudjunk kiépíteni. Ezt úgy oldottuk meg, hogy a játékot személyessé tettük, a témája az autentikus ságváris élet.

2.2. A játék menete

A játékosok felhúzzák a kezdő kezüket, majd egymást váltva kijátszanak tetszőleges számú karakter- vagy varázsigekártyát. A kártyák erejüknek és hasznosságuknak megfelelő számú „manába” kerülnek, a következő kört $n+1$ manával kezdik a felek. A játék elején a második játékos megfelelő kompenzációt kap az egyenlő győzelmi esély érdekében. Miután mindkét játékos köre befejeződik, megkezdődik a második fázis. Ekkor bármilyen akció megengedett: lehet kártyát kijátszani, illetve támadni is. Ha van az ellenfél $2*5$ -ös tábláján támadható célpont, az ellenséges játékosba közvetlenül nem lehet sebezni. Ha egy karakterkártya megtámad egy másikat, mindkét kártya hp-ja (életerőpontja) az ellenséges kártya atk (támadóerő) értékével csökken. Ha ezáltal bármely kártya hp-ja 0-ra vagy az alá csökken, az a kártya lekerül a tábláról, a túlélő kártya atk-jának és az elkárhozott kártya hp-jának a különbsége átfolyó sebzésként vonódik le a mögötte lévő célpont hp-jából. Mindkét játékos limitált számú „kő” -vel kezdi a játékot, ezeket tetszőlegesen használhatják a játék során. A köveknek nincs atk-ja, pusztán védekezésre szolgálnak. Győzni az ellenfél hp-jának 0-ra csökkentésével lehet.

2.3. A kártyák

Összesen 170+ kártyát készítettünk, ezekből kb. 100 teljes mértékben eredeti, mindegyik különleges képességekkel rendelkezik. Vannak tulajdonságok, amelyek közül minden karakterkártya rendelkezik néhányval, ezek érthetővé és kategorizálhatóvá teszik őket, néhol humoros módon elkülöníthetőbbé válnak a kártyák a saját képességeiken felül is. A képességeket függvényekben programoztuk le, így könnyen meghívhatók a játék során. Minden kártya rendelkezik egy képességeírással, egy idézettel, egy hanganyaggal, egy képpel és egy ritkasági besorolással. A kártyák 4 fő csoportra vannak osztva: tanárok, diákok, varázsigék és extrák. Az extrák között mindenféle karakterkártya található, többségük popkulturális utalás vagy iskolai személyzet.

2.4. Kollekcio és Pakli felületek

A fent említett kreatív elemeket a játékosok a kollekcio felületen tudják megtekinteni. A pakli felület arra szolgál, hogy a korábban megszerzett kártyákból tetszőleges összetételű kártyapaklit tudjanak a felhasználók alkotni.

2.5. Gacha felület

A játék során szerezhető kristályokból lehet kártyacsomagokat vásárolni a gacha felületen. A kártyák tematikus csoportokba vannak sorolva, az ezek közti szinergia jóval magasabb, mint ha más-más csoportokból raknánk össze egy paklit, ezzel segítjük a játékosok dolgát, hogy ne érezzék túlterhelve magukat.

2.6. A metajáték

A játékon belül van egy szintrendszer, ami azt a célt szolgálja, hogy a kártyák és a szükséges tudás megszerzése mellett legyen egy külsőleges fejlődési íve is a játéknak, ami által nagyobb arányban tudjuk megtartani a játékosainkat. (Ezt az állítást sikeres játékfejlesztő cégek szóvivőinek megnyilvánulásaival tudjuk alátámasztani, pl.: SUPERCELL)

2.7. Design

A felhasználói élményt állítottuk középpontba, a UI és UX kifejezetten kedvez azoknak a játékosoknak, akik még sohasem játszottak online kártyajátékkal. A játék a böngészőben érhető el, működik számítógépen és mobilon is. Miután a játékos regisztrál és belép a fiókjába, a kezdőlappal találja magát szemben, innen bármi elérhető, bármikor vissza is lehet ide térni. Nem mindenhol egységes a design, de ez nem is feltétlenül volt célunk. Sok különböző helyről nyertünk inspirációt.

2.8. A közönség elérése és bevonása

Készítettünk egy hivatalos weboldalt a játékról a Wix szolgáltatással, ahol megosztjuk a különböző médiatartalmainkat, fejlesztési naplót írunk és gyakran ismételt kérdésekre válaszolunk. Az oldal az alábbi linken tekinthető meg: <https://sagvarigachatetms.wixsite.com/sggacha>. Írtunk továbbá egy cikket az iskolai újságba, ami nagy sikernek örvend. Természetesen több ezer sor kódban akad egy-egy hiba. Néhány barátunk jelentkezett tesztelőnek, így remélhetőleg a bugok nagy részét ki tudjuk javítani a közeljövőben. Hirdettünk ki Kahoot! délutánokat, és igyekszünk minden érdeklődőnek segíteni, ha elakad a játék bármely pontján. A kész játék az sggacha.hu oldalon lesz elérhető. Az iskolai hálózaton tudjuk majd hostolni az SZTE Gyakorló Gimnázium és Általános Iskola jóvoltából.

2.9. A legnehezebb kihívások

Nem tudtuk, hogy az óriási mennyiségű adatot, amivel dolgozunk, milyen módon érdemes tárolni:

- A felhasználó mely adatait szükséges rögzíteni?
- A kártyák mely adatait szükséges rögzíteni?
- Ezeket az adatokat hogyan és hol kell tárolni?
- A tábla aktuális állását a játékos szemszögéből hogyan kell úgy tárolni, hogy mindkét játékos kliense fel tudja dolgozni?

Ezekben a kérdésekben felkészítő tanáraink, a Stackoverflow és a ChatGPT segítettek, ha igazán elakadtunk.

A game balance (játék egyensúly) is komoly fejtörést okozott:

- A kártyák erősségét hogyan lehet megítélni tesztelés nélkül (vagy minimális teszteléssel)?
- Hogyan lehet egy játékosnak sikerélménye akkor is, ha a paklija nem követi az általunk indirekten javasolt összetételre vonatkozó elveket és kártyakombinációkat?
- Honnan tudjuk, hogy nem túl egyszerű vagy túl bonyolult a játék menete a közönségünk számára?

Számtalan forrás elérhető a YouTube felületén, számunkra a leghasznosabb ismeretforrásoknak a GDC-n bemutatott előadások bizonyultak ezen a téren. Utolsó problémánk az internetre való feltöltés, bonyolultabb, mint azt mi eredetileg gondoltuk, de ezt a feladatot is lassan megoldjuk.

3. Elért eredmények

Összesen kb. 350-400 órányi aktív munkánknak az eredménye ez a játék, meglehetősen túllőttünk az eredeti célunkon, de barátaink és a projekt által bevonzott ismerőseink folyamatos lelkesedése motivál már egészen a kezdetek óta. A játék még nem széles körben elérhető, de már így is gyarapodik a kis közösségünk körülötte. Legfőbb céljaink a tanulás, szórakoztatás és a közösségépítés voltak, a jelek biztatóak, hogy sikerült is ezeket elérni.

A játékfejlesztés ezekre a célokra egy egyedi megoldás, de a közösségre fordított figyelem kreatív és eredeti megoldásokat serkentett, így egy karakteres és sokak szívéhez közel álló terméket tudtunk létrehozni.

Avatar Projekt – Metaemberek az oktatásban

Polyp Csapat

Bacsur Dániel, Vajda Ádám

Felkészítő: Kováts Livia, Molnár Zsuzsanna

Közgazdasági Politechnikum (Budapest, Vendel u. 3, 1096)

ELTE Radnóti Miklós Gimnázium (Budapest, Cházár András u. 10, 1146)

1. Bevezetés

Az elmúlt hónapban a projekt prototípusán dolgoztunk, amit decemberben nagy érdeklődéssel mutattunk be szüleinknek és rokonainknak. A demo során mindannyiuknak alkalma volt beszélgetni Kolumbusz Kristóffal - vagyis az ő avatárjaival - aki mindeközben első szám első személyben mesélte felfedezéseit és kalandjait. Elképesztő élmény volt látni, ahogy újabb és újabb kérdésekkel álltak elő, ahogy próbálták minél több dolgot megtudni a kalandoroktól.



2. ábra: Az avatár alap állapotban

A kísérlet közben döbrentünk rá, hogy ez mekkora élményt okozott rokonainknak és hogy egy kis továbbfejlesztéssel mekkora segítséget jelenthetne tanulók ezreinek.

1.1. Az általunk megálmodott jövő

Az általunk megálmodott jövő, hogy újraalkossuk a digitális oktatás módját olyan ultrarealisztikus avatárak/digitális karakterek segítségével, melyek képesek magával ragadó és élethű kommunikációs élményt nyújtani. Többek között ilyen az 1. ábrán látható avatár, kivel honlapunkon szabadon beszélgethetünk a minket érdeklő témákról és elméleti anyagokról.

2. Probléma megoldásának menete

A projektünk során elsődleges célunk volt egy olyan avatár létrehozása, amely nem csak szépen és érthetően beszél, de képes 1,5 másodpercen belül válaszolni a felhasználók kérdéseire, így biztosítva a zökkenőmentes és természetes kommunikációt. Továbbá az avatárunknak képesnek kellett lennie széles körű, szakspecifikus tudás biztosítására, amihez egy egyedi, oktatási célokra szabott nyelvi modellt fejlesztettünk ki.

2.1. Egy egyedi LLM nyelvi modell

A projekt egy speciálisan kifejlesztett nyelvi modellen alapul, amelyet kifejezetten az oktatási célokra szabtuk. Ennek célja, hogy a tanulók által feltett kérdésekre releváns, érthető és informatív válaszokat adjon, illetve hogy a beszélgetés előzményei alapján képes legyen adaptálódni a tanulók egyedi igényeihez és előzetes tudásához.

2.2. Háttértudás betanítása

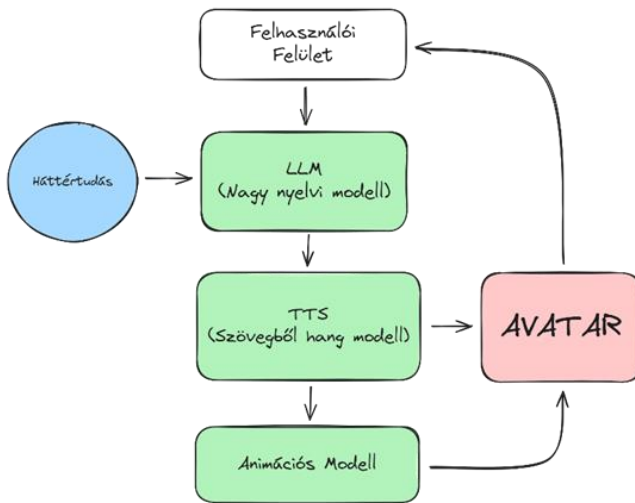
A projekt keretében különös hangsúlyt fektetünk a modell általános és szakspecifikus tudásának bővítésére. Ez magában foglalja a legújabb tudományos kutatások, történelmi események, irodalmi művek és egyéb releváns információk integrálását beleértve a nemzeti alaptantervben (NAT) megfogalmazott pontokat. Ehhez egy 1536 dimenziós vektor adatbázist használtunk, aminek beágyazásait az OpenAI Ada-2 modellével generáltattuk le.

2.3. Magyar beszédhangok szintetizálása

A projektünk egyik kulcsfontosságú eleme volt a magyar beszédhangok hiteles és természetes szintetizálása. Ennek érdekében a Microsoft Speech Services szolgáltatását választottuk, amely kiváló minőségű, természetes hangzású szöveg-hang konverziót tesz lehetővé. A szolgáltatás előnye, hogy többféle nyelvet és dialektust támogat, így képes voltunk egy olyan egyedi hangprofil létrehozására, ami illeszkedik a magyar nyelvi sajátosságokhoz és a projektünk követelményeihez.

2.4. Szájmozgás és az Avatár generálása

A szájmozgás és az avatár generálásának célja az, hogy az avatár szájmozgása szinkronban legyen a beszédhanggal, ezáltal élethűbbé téve a kommunikációt. Ebben a fázisban a beszédhang elemzése alapján generáljuk a szájmozgásokat, figyelembe véve a beszéd tempóját, hangsúlyát és az artikulációs pontokat.



2. ábra: Modelleink működése

3. Elért eredmények

Az Avatar Projekt nemcsak újszerű megközelítést kínál a digitális oktatásban, hanem kiemelkedő személyre szabottságot és interaktivitást is nyújt, ami elősegíti a tanulók jobb megértését és érdeklődésének fenntartását. A projekt eredményei között szerepel az egyéni tanulási igényekhez igazodó interaktív oktatási környezet kialakítása, melyet teljes mértékben az általunk feljesztett modellek tesznek lehetővé.

3.1. Amit kifejezetten hasznosnak gondolunk

Mivel a projekt egy szoftveres innováció, így lehetőségünk nyílik arra, hogy minden tanulóhoz egyéni segédet/mentort/korrepetitort csatoljunk, aki a későbbiekben vele és csak vele foglalkozik, ezzel kiemelt kapcsolatot létesítve az avatar és a tanuló között. Továbbá honlapunk fenntartási és működtetési költségei rendkívül alacsonyak, melyek hozzájárulnak ahhoz, hogy projektünket a lehető legtöbb diáktársunkhoz eljuttathassuk.

3.2. Meghívó az Avatar Projekt Kipróbálására

Összefoglalva, az Avatar Projekt úttörő lépést jelent az interaktív digitális oktatás területén, és büszkéek vagyunk arra, hogy ezt a forradalmi eszközt a nagyközönség elé tárhatjuk. Szeretnénk meghívni minden érdeklődőt, hogy próbálja ki alkalmazásunkat és tapasztalja meg az általunk elképzelt tanulási élményt. Látogasson el a sziiiv.danielbacsur.dev weboldalra.

Virtual Front Desk, a virtuális recepció szoftver

VFD Dev

Horváth Zsombor

Felkészítő: Horváth Zsombor

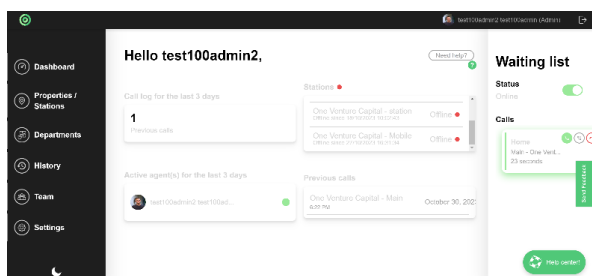
*Budapesti Fazekas Mihály Gyakorló Általános Iskola és Gimnázium,
Budapest, Horváth Mihály tér 8, 1082*

1. Bevezetés

Egy 2020-as felmérés szerint csak az USA-ban több mint 1.2 millió recepciós dolgozik hotelekben, szállodákban, orvosi rendelőkben és egyéb intézményekben. Ez cégenként több százezer dollár költséget is jelenthet, mivel egy recepciós nem mindig képes ellátni az összes feladatot, így több ember bérét kell megfizetniük. Ha például egy egyetemi kampuszról van szó, akkor ott minden épületbe kell fizetni egy recepciós, mivel egyszerre nem lehet ott élőben mindenhol.

2. A probléma megoldása

Erre nyújt megoldást a Virtual Front Desk, amely szoftver segítségével a vállalatok akár több százezer dollárt megspórolhatnak évente. Az alapgondolat az mögötte, hogy előre kihelyezett érintő képernyős kioszk állomásokon tudnak videóhívást kezdeményezni az épületbe betévedő emberek egy recepciós felé. A kérdező félnek nincs más dolga, mint kiválasztani a megfelelő embert, akitől kérdezni szeretne és rányomni a hívás gombra. Ez rögtön jelzi a recepciósnak, hogy van egy bejövő hívása egy adott épületből és tudja fogadni,



3. ábra: Bejövő hívás a recepciós felületén

vagy később akár vissza is tudja hívni az állomást (1. ábra). A recepciós helyileg bárhol tartózkodhat, ahol stabil internet elérése van, így nincsen fizikailag helyhez kötve és emiatt akár több épületért is felelhet egy ember. Ezzel hatalmas összegeket spórolhat egy cég, mivel 3 vagy 4 ember helyett elég 1-et felvenniük, aki felel az összes felmerülő kérdésért. De nemcsak arra képes a

szoftver, hogy összekösse a két személyt, hanem a recepciós akár csinálhat valós idejű képernyőképet az állomás kamerájáról vagy kitöltethet egy adat bekérő kérdőívet a hívást kezdeményező személlyel. Ennek köszönhetően használható a szoftver kórházakban is, ahol a betévedő páciensek igazolják magukat az okmányukkal, amiről a recepciós fényképet készít, hogy később vissza lehessen ellenőrizni. Mivel az épületbe betévedő emberek több nemzetiségűek is lehetnek, ezért a rendszer összesen 13 nyelven érhető el, amelyek közül az aktuális felhasználó választhatja ki a neki megfelelőt (2. ábra).

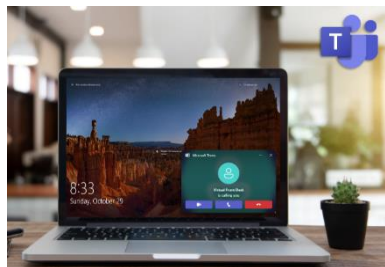


2. ábra: Az 13 nyelvet támogató állomás kinézete

Gyakran fordul elő azonban, hogy nincsen bejelentkezve a recepciós ebbe a web applikációba, ahol alap esetben fogadni tudná a hívásokat. Erre két megoldást is lefejlesztettem, annak érdekében, hogy minden kezdeményezett hívás fogadóra találjon. Az első az, hogy egy előre beállított telefonszámra egy SMS üzenetet küldd, ha valaki hívást kezdeményezett az adott személy felé. A második funkció pedig az, hogyha valaki több mint fél percre nem veszi fel a video hívást akkor a rendszer automatikusan telefonon kezdi el hívni a recepciót, hogy onnan legyen képes felvenni a hívást.

3. Microsoft Teams integráció

Egy 2023-as felmérés szerint csak az USA-ban több mint 15 ezer cég használ Microsoft Teams-et elsődleges kommunikációs felületként. Ezek közül nem csak az informatikai területén találhatunk szervezeteket, hanem vannak



3. ábra: Microsoft Teams felületén beérkező hívás

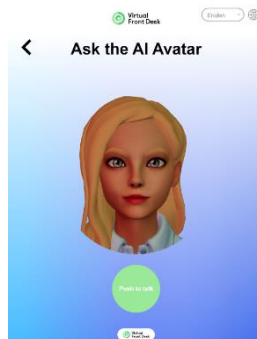
köztük kórházak, ügyvédi irodák, állami szervezetek és oktatással foglalkozó cégek is. A nagy cégek nem igazán szeretik, ha a beosztottak egy külsős cég szoftverét használják még a Teams mellett, ezért próbálnak mindent egy szoftver alatt tartani és intézni. Innen született meg az ötlet a Virtual Front Desk egy Microsoft Teams-el való integrációjára (3. ábra). Ezzel a funkciójával úttörő az IT iparában, mivel nem található még egy olyan külsős cég, amely a saját videóhívásait a Microsoft Teams platformjába intézné direkt (kicsöngéssel, mint egy rendes „hagyományos” Teams hívás).

4. Látogató nyilvántartó rendszer

Ha egy látogató betéved egy épületbe és egy bizonyos személyhez jött egy előre megbeszélt időpontra, akkor hasznos lenne értesíteni az adott személyt, hogy az ügyfele megérkezett és éppen rá vár. Erre született meg a “visitor management” funkció, amelynek keretein belül az admin képes feltölteni vagy összeállítani szerződéseket, amiket a látogatónak az értekezlet előtt digitálisan alá kell írnia. Ezen felül kérhet selfie fotót kérhet az ügyféltől, valamint beállíthatja, hogyha új látogató érkezett hozzá, akkor milyen formában legyen értesítve erről (SMS, Email vagy video hívás).

5. Mesterséges intelligenciával működő 3D recepciósök

Egy kétségtelenül új innováció volt a szoftverben, mikor lefejlesztettem, hogy 3D renderelt karakterek jelenjenek meg az állomáson, akik előre meghatározott információbázisból, valamint GPT modellek adataiból képesek válaszokat adni a feltett kérdésekre. Az adminnak nincs más dolga, csak létrehozni egy új 3D recepcióst a felületén, ahol kiválasztja a használni kívánt karakter kinézetét, valamint adatokat táplál be neki a cégéről, épületeiről és bármiről, amit kérdezhetnek az épületbe betévedő emberek egy valódi recepcióstól. Ezután az állomás, ha ezt az opciót választja a felhasználó, egy ilyen „avatar” fog megjelenni (4. ábra). A felhasználó egy gomb lenyomásával

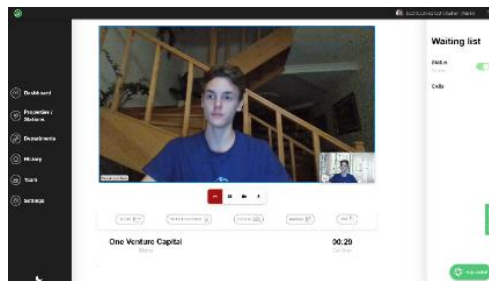


4. ábra: A 3D recepció megjelenése az állomáson

kérdezhet szóban a rendszertől, majd a hangja szöveggé alakul a háttérben, amire válaszol a mesterséges intelligencia szövegben, majd az AI válasza is hanggá alakul vissza. Közben mozog a 3D renderelt karakter is, mintha egy valódi ember beszélne hozzánk, válaszolna kérdéseinkre.

6. Technikai megvalósítás

Alapvetően az egész rendszer webes alapú és egy React JS nevezetű Javascript könyvtárban íródott. Ez a React frontend használ egy Tailwind CSS nevezetű CSS keretrendszert a könnyű és modern dizájnok megvalósítása érdekében. Ugyanígy frontend oldalon van használva még az Azure Communication Services (ACS) egy React projektekhez készített modula is, amely egyszerűbbé és gyorsabbá teszi a videó hívásokat (5. ábra). Annak érdekében, hogy egy hívás elindításánál instant megjelenjen a notifikáció a főoldalon, a Google Firebase Realtime adatbázisát használja a rendszer. Más fajta adatok (például nevek, címek) egy Express JS backend API-on keresztül kerülnek be egy PostgreSQL adatbázisba. A 3D renderelt AI recepcióshoz egy Three JS nevezetű Javascript könyvtárat használtam a 3D-s megjelenítéshez, valamint GPT API-t a megfelelő és pontos válaszok generálásához.



5. ábra: Példa ACS video hívás kinézete

7. Elért eredmények

A szoftver fejlesztésében nagyjából másfél évem munkája van benne, ennek köszönhetően lehetett ilyen összetett technikailag és üzletileg is emiatt kezdi elérni a sikereit. Annak érdekében, hogy valós ügyfelek is használják a mindennapi életük során a szoftvert, közel fél éve elkészült a marketing célú website-om is, ahol információt nyerhetnek az ügyfelek az egész szolgáltatásról. Az előbbieken felsoroltak következtében mára már több mint 70 külföldi és belföldi cég (pl. ügyvédi irodák, szállodák, kórházak) használja aktívan a szolgáltatást, akik ezért havonta pénzt is fizetnek. Ezen felül a 2023-as év elején a szoftver egy kifejezetten korai állapotában megnyerte a Neumann János Informatikai Verseny webfejlesztői díját is (ekkor még csak egy leegyszerűsített video hívás létezett, semmi más).

LovassyApp – Tanulást ösztönző rendszer

NemCsoport

Ocskó Nándor

Felkészítő: Báder Anikó

Lovassy László Gimnázium, 8200 Veszprém, Cserhát ltp. 11.

1. Bevezetés

A LovassyApp egy tanulást ösztönző webes szoftver, amely az iskolám egy korábban nem digitális rendszertét váltja le. Ennek a rendszernek a lényege az, hogy az iskola diákjai az órákon szerzett jegyeiket egy iskolai valutára válthatják be (ez esetünkben a LovassyLóvé, későbbiekben: LoLó) és a beváltott pénzüikkel a diákok különböző, tanárok által felajánlott, termékeket vásárolhatnak, majd használhatnak fel órán vagy órán kívül. Ilyen termék például a „kimentés”, amely egy órán felhasználható termék, a felajánló tanár egy óráján mentesíti a felhasználó diákot a be nem jelentett dolgozatok és felelések alól. Egy másik termék például a „zene az iskolarádióban”, amely egy órán kívül felhasználható termék és technikai feltételei közé tartozik, hogy beváltáskor a diák megadhatson különböző értékeket bemeneti mezőkbe (ebben az esetben egy ilyen mező a zeneszám címe).

Egy másik fontos funkciója az alkalmazásnak a szavazó rendszer, amely szintén egy korábbi iskolai rendszert vált le, és lehetővé teszi a diákoknak és a tanároknak, hogy iskolai eseményeken képszavazásokban tudjanak részt venni.

Bár maga az alkalmazás a Lovassy László Gimnázium diákjainak készült, de az is fontos volt, hogy konfiguráción keresztül az alkalmazás adaptálható legyen új követelményekhez, a LoLó rendszer változásaihoz és más iskolák sajátos követelményeihez is.

2. A probléma megoldásának menete

2.1. A kitűzött célok

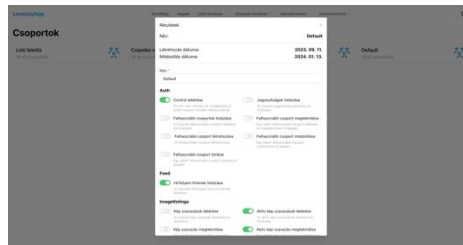
Fontos célkitűzésem volt, hogy a jegyek külön munka nélkül, vagy minimális külön munkával jussanak el az adatbázisába egy megbízható forrásból, valamint az alkalmazás legyen minden eszközön jól használható, és a rendszerrel visszaélő felhasználók legyenek kitilthatók, valamint csak az iskola diákjai tudjanak regisztrálni. Mindezek mellett lehetővé akartam tenni, hogy bárki saját alkalmazásokat tudjon fejleszteni, amelyek a LovassyApp funkcióit valósítják meg (a központi iskolai rendszert használva) vagy azzal

összhangban működnek. Végül pedig azt is fontosnak tartottam, hogy az alkalmazás minimális emberi beavatkozással és költségek nélkül működjön, karbantartást lehetőség szerint ne igényeljen, ezt el is értem.

2.2. A megvalósítás

A megvalósítás modulárisan történt. Minden funkció egy funkciós csoportba tartozik, és a funkciós csoportok egymástól teljesen független egységeket alkotnak egy monolitikus architektúra keretein belül. A funkciós csoportok mind egyenként ki- és bekapcsolhatók és a köztük történő kommunikáció egy Event-Listener megoldással történik.

Az első ilyen funkciós csoport az autentikációért és autorizációért felelős csoport. Ez foglalja magában a belépést, a jogosultságok kezelését, az e-mailes megerősítést és a jelszó visszaállítását is. A rendszer csak @lovassy.edu.hu-ra végződő email című diákokat enged regisztrálni és amikor valaki regisztrál, automatikusan bekerül a „Default” felhasználói csoportba és megkapja az ezzel járó jogosultságokat is. A rendszerrel visszaélőket is ezen felhasználói csoport megváltoztatásával lehet kitiltani.



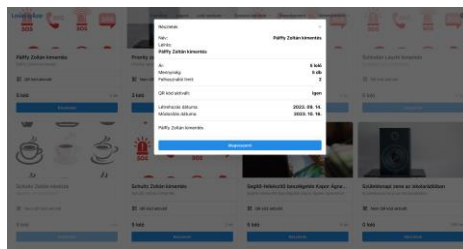
4. ábra: A jogosultságok kezelése

A második ilyen modul a szavazásokért felelős modul. Ez foglalja magában a képszavazások létrehozását, kezelését... Egy szavazás lehet inkrementáló (tetszik/nemetszik) típusú vagy választásos (az összes közül csak egy kép választható). Megszabható az is, hogy egy ember hány képet tölthet fel, mely felhasználói csoportok tölthetnek fel képeket, meg legyen-e jelenítve a diák neve és osztálya is a beküldött képek mellett és még sok más.

A harmadik ilyen modul a jegyek importálásáért felelős. Ez foglalja magában az „import kulcsok” kezelését (ilyen birtokában lehet csak jegyeket feltölteni), valamint ehhez tartozik egy asztali alkalmazás is, amely feldolgozza a jegyeket és titkosítva elküldi a szerveren futó alkalmazásnak. Minden héten egyszer az iskolánk igazgatóhelyettese tölti fel a krétából kimenthető összes jegyet tartalmazó Excel táblázatot. Korábban ez a rendszer a Kréta API-t használta és így nem is volt szükség emberi interakcióra a jegyek

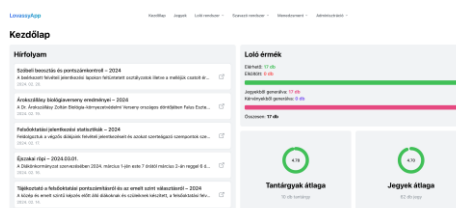
megszerzéséhez, de sajnos a Kréta egy kérésemre válaszul nem engedélyezte ezt, így ezt le kellett cserélnem a jelenlegi rendszerre.

Következő nagyobb funkciós csoport az a bolt rendszer. Ez foglalja magában a „bazárt” és a „kincstárt” (minden felhasználó megvásárolt termékeit) és ezen keresztül lehet termékeket létrehozni, kezelni és beváltani, valamint kérvényeket kezelni, ugyanis nem csak jegyekért, hanem versenyeredményekért is lehet LoLót kapni, és ehhez a diáknak egy kérvényt kell benyújtania. A termékek lehetnek órán aktiválhatók, ekkor a diáknak egy QR kódot kell beszkenneálnie az órán, amely egyedi és a tanárnál van kinyomtatva, és lehetnek nem órán aktiváltak, ekkor csak egy gombot kell megnyomniuk. A termékek tartalmazhatnak még bemeneti mezőket is megadható az is, hogy egy ember hányat vehet az adott termékből.



2. ábra: A bazár

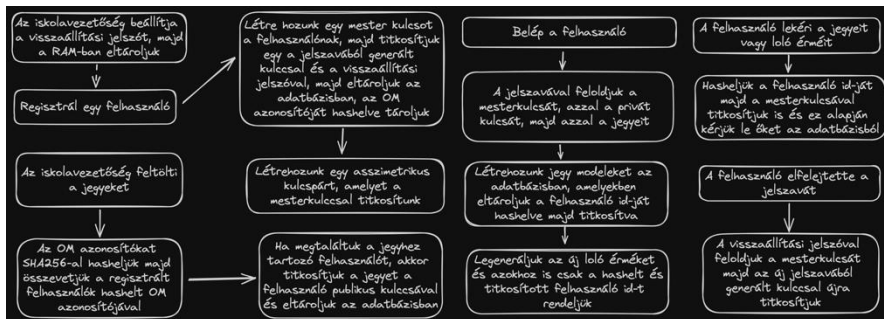
Végül pedig a kisebb modulok: a felhasználók modul, a hírfolyam, az „iskolai” modul és a státusz modul. Ezek felelősek a felhasználók kezeléséért, regisztrációért, az iskola híreinek jelzéséért, azért, hogy a diákok meg tudják nézni a jegyeiket az alkalmazásban és a rendszer állapotának kijelzéséért is.



3. ábra: A kezdőlap, ahol láthatók az iskola hírei

A LovassyAppnak sok biztonsági kikötésnek is meg kellett felelnie, így egy extenzív biztonsági rendszert is tartalmaz. Elvárás volt, hogy a diákok jegyeit senki se lássa, még az sem, akinek hozzáférése van az adatbázishoz, valamint, hogy a jegvek titkosíthatatlanul soha ne hagyják el az

igazgatóhelyettesünk számítógépét (és így is csak a regisztrált felhasználók jegyei). A rendszer működését az alábbi ábra mutatja:



4. ábra: A biztonságért felelős folyamatok rendszere

2.3. A felhasznált technológiák

Mivel költségmentesen akartam megvalósítani az alkalmazást, az iskola erőforrásaival kellett dolgoznom, és ez mindössze egy nagyon gyenge régi linux szervert jelent, emiatt sok teljesítmény optimalizáció is kellett.

Az alkalmazás központját, a szerveren futó részét ehhez igazítva ASP.NET Core keretrendszerben írtam meg, mivel elképesztően gyors nagyon kevés erőforrás mellett is, és websocketeken keresztül lehetővé tette, hogy minden, ami az oldalon látható élőben (újra töltés nélkül) frissüljön. A külső alkalmazásokat támogatva ez egy publikus API-t tesz elérhetővé.

A webes felületet React keretrendszerben írtam TypeScript segítségével és az asztali alkalmazást Tauri és React használatával írtam meg.

Az egész alkalmazás teljes mértékben nyílt forráskódú, forráskódja elérhető itt: <https://github.com/LovassyApp/LovassyApp>

3. Elért eredmények

Az alkalmazás nyújtotta előnyök már az első hetekben elérték, hogy a diákság több mint 90%-a regisztráljon, és a LoLó rendszert az alkalmazás előtti évek 5-6 fős felhasználói táboráról egy jelenleg 537 fősre bővítette, ez által is motiválva a diákokat a tanulásra. Az iskolánkon túlmutató potenciálját az is bizonyítja, hogy már a fejlesztés alatt is érkezett kérés (a Pannon Egyetemről), hogy csináljuk meg a Neptun-hoz is, hogy egyetemek is használhassák.

Néhány statisztika az alkalmazás kihasználtságáról: 105 terméket vásároltak az idei tanévben, ebből 87-et fel is használtak, és 3597 LoLót gyűjtöttek a diákok (3 ötös/5 négyes = 1 LoLó), és 56 tanár ajánlott fel termékeket.

Algorithm

Team Algorithm

Váradi Marcell, Király Márk, Vodenicsár Rajmund
 Felkészítő: Péterffyné Takács Katalin & Bene Ádám
 Siófoki SZC Mathiász János Technikum és Gimnázium,
 8630 Balatonboglár Szabadság utca 41.

1. Bevezetés

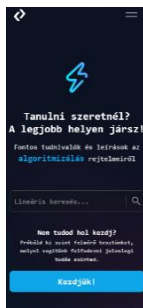
Az "https://algorithms.hu/ " egy olyan webalkalmazás, melynek célja az algoritmusok bemutatása és megértetése a középiskolások vagy bárki számára. Az oldal széles spektrumon keresztül mutat be különböző algoritmusokat, segít animációkkal és kódrészletekkel megértetni azokat. Az oldal a diákok, és érdeklődők számára egyaránt hasznos információkat nyújt az algoritmusok különböző típusairól és alkalmazásairól is.

Az oldal folyamatos fejlesztés alatt áll, így az algoritmusok száma bővülni fog.

2. Alkalmazott erőforrások

A weboldalon Express JS keretrendszert használtunk MySQL adatbázis támogatásával. Az oldalunk EJS motoron fut, illetve Tailwind CSS-t használ a grafikai elemek megjelenítéséhez, az animációkat pedig Canva segítségével hoztuk létre.

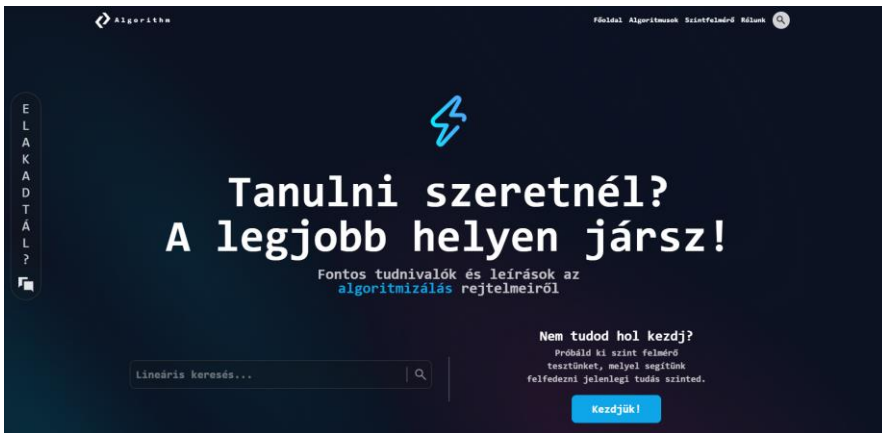
3. Képernyőképek és leírások a weboldal funkcióiról



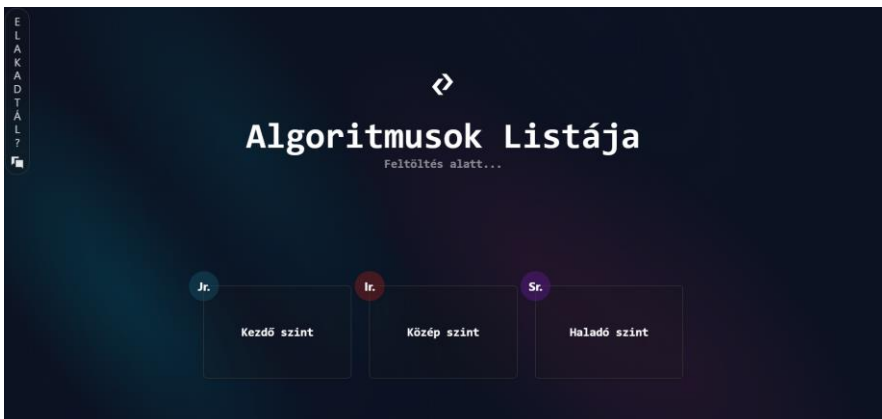
6. ábra: Telefon menü - Telefon nézetben a menü egy legördülő menüvé alakul át.



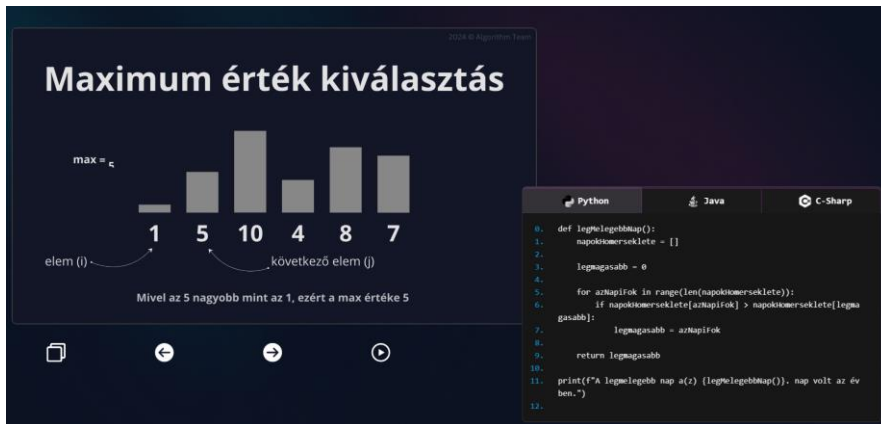
5. ábra: Menü – Weboldalunk navigációs rendszere, amely segítségével elérhetőek szolgáltatásaink.



7. ábra: Kezdőlap, keresés – A főoldalon található a kereső, segítségével kereshetünk rá algoritmusokra, amelyet a menüben lévő keresés ikonnal is elérhetünk. A „Kezdjük!” gombra kattintva megnyílik szintfelmérő oldalunk, amelyben a felhasználó visszajelzést kap jelenlegi tudásáról a szintfelmérő kitöltésével. Az „E L A K A D T Á L ?” oldalsáv alján található ikon használatával megnyílik a „Rólunk” nevezetű oldal amelyen belül egy űrlap kitöltésével visszajelzést, segítségkérést lehet kezdeményezni.



8. ábra: Algoritmusok listája – Itt van felsorolva az összes algoritmus amely megtalálható az adatbázisunkban. Kategóriákra van bontva (Kezdő, Közép, Haladó), a fent lévő kártyákra kattintva, oda fog görgetni az oldal, amelyiket kiválasztotta.



9. ábra: DEMO egy algoritmról – Ezen az oldalon látható az algoritmus vizualizálása a forráskóddal együtt. A forráskód háromfajta programozási nyelven tekinthető meg (Python, Java, C#). A bal oldalon lévő animáció alatti gombok segítségével tudja kezelni az animációt (Animáció kimásolása, Visszatekerés – 5s, Előre tekerés – 5s, Animáció szüneteltetése/elindítása).

Válassza ki a helyes algoritmust

```

max = t[0]
ciklus i = 1 .. n - 1
  ha t[i] > max akkor
    max = t[i]
  ha vége
ciklus vége
ki max

```

A: Buborekrendezés

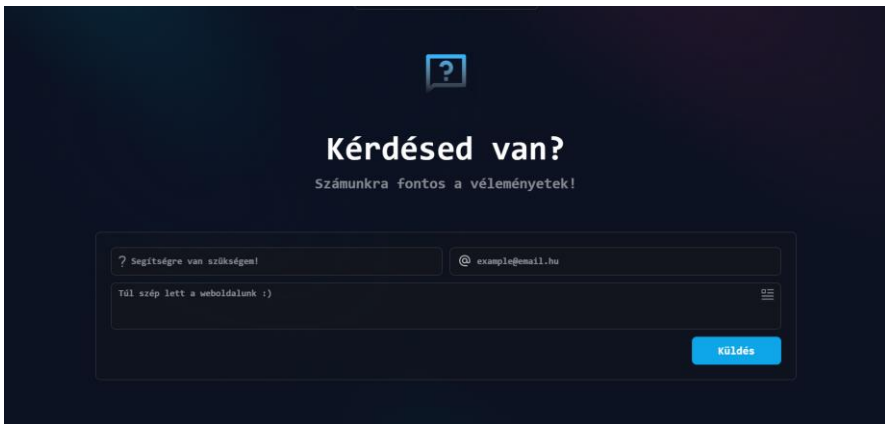
B: Egyszerű cserés rendezés

C: Maximum kiválasztás

D: Lineális keresés

Ellenőrzés

10. ábra: Szintfelmérő – A felhasználó a szintfelmérő oldalon tesztelheti tudását algoritmizálás terén. A szintfelmérőn meg kell oldani különböző algoritmizálással kapcsolatos problémákat és ezek megoldása után az elért pontszámok alapján fog javasolni egy szintet az oldal. (🔒 Jelenleg ez a funkció fejlesztés alatt áll! 🔒)



?

Kérdésed van?

Számunkra fontos a véleményetek!

? Segítségre van szükségem!

@ example@email.hu

Túl szép lett a weboldalunk :)

küldés

11. ábra: Kapcsolat – A felhasználó itt tud visszajelzést küldeni számunkra. A beküldött űrlapokat egy adatbázisba mentjük és feldolgozzuk.

4. Linkek

GitHub: <https://github.com/MystioDev/algorithm-tailwind-express-ejs>

Bemutató videó: <https://www.youtube.com/watch?v=aao5rZmfuNs>

Weboldalunk: <https://algorithms.hu/>

Hermes – Sportversenyek szervezése és lebonyolítása egyszerűen

NemCsapat

Ocskó Nándor

Felkészítő: Báder Anikó

Lovassy László Gimnázium, 8200 Veszprém, Cserhát ltp. 11.

1. Bevezetés

A Hermes egy web alkalmazás egy mobil alkalmazással társítva, amely jelentősen leegyszerűsíti a (csapat) sportversenyek szervezését és lebonyolítását. Az iskolában évente megszervezett röplabda verseny kapcsán merült fel ennek a programnak az elkészítése, és emiatt ennek a szükségleteit teljes mértékben le is fedti a Hermes funkcionalitása, de mindemellett még sok más egyéb igényt kielégítő funkciót is megvalósítottam benne.

2. A probléma megoldásának menete

2.1. A kitűzött célok

Az alkalmazásnak egy sportverseny szervezésének és lebonyolításának minden fázisát le kellett fednie. Ez azt jelenti, hogy meg kellett oldani, hogy a regisztrációs időszak alatt csapatok tudjanak regisztrálni (a megadott paraméterek között – létszám, csapatok maximális száma ...), a regisztrációs időszak után, de még a verseny kezdete előtt meccseket lehessen létrehozni, valamint automatikusan generálni konfiguráció alapján, és a verseny alatt valós időben nyomon lehessen követni bárki által az éppen zajló meccset, valamint a korábbi és elkövetkezendő meccseket.

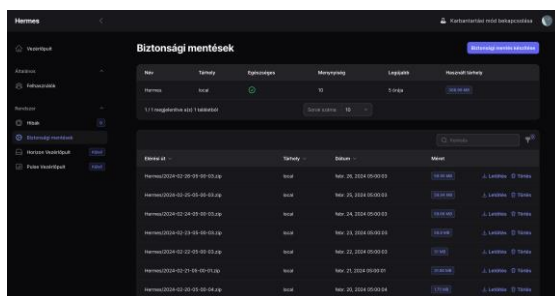
Ezek mellett, mivel sok eseményen maga a sportverseny csak egy mellék program (például egy DÖK napon egy foci bajnokság más programok mellett), azt is meg szerettem volna oldani, hogy a csapatok tagja nyugodtan foglalkozhassanak bármivel amíg nem ők játszanak, és amikor el kell kezdeniük készülni egy meccsre, akkor kapjanak egy értesítést erről.

Végül pedig a versenyekkel és csapatokkal kapcsolatos statisztikákat is szerettem volna elérhetővé és exportálhatóvá (CSV, XLSX formátumban) tenni. Statisztikákat viszont nem csak a versenyekkel kapcsolatban szerettem volna kiállítani, hanem az alkalmazás kihasználtságával, sebességével és felmerülő hibáival („exception”-ökkel) kapcsolatban is.

2.2. A megvalósítás

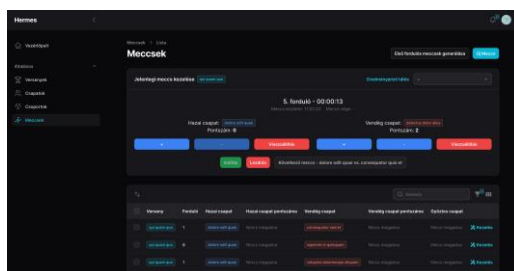
Az alkalmazásban két féle felhasználó létezhet: adminisztrátor és menedzser. A versenyeken résztvevőknek nincs szüksége felhasználói fiókra. Mindkét felhasználó típusnak egy saját felülete van.

Az adminisztrátor felületen lehet kezelni a többi felhasználót (megszemélyesíteni is, azaz úgy nézni az alkalmazást, mintha ekként a felhasználóként lenne bejelentkezve a megszemélyesítő adminisztrátor), megnézni a le nem kezelt hibákat, biztonsági mentéseket létrehozni az alkalmazásról és az adatbázisról, valamint a háttérben futó folyamatokat kezelni (Horizon vezérlőpult), és az alkalmazás kihasználtságával kapcsolatos statisztikákat megnézni (Pulse vezérlőpult).

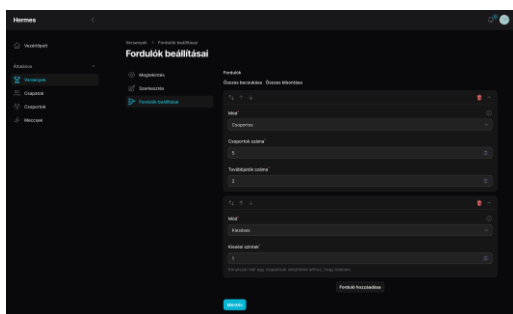


12. ábra: Az adminisztrátori felület

A menedzser felületen lehet új versenyeket létrehozni és meglévőket kezelni, csapatokat létrehozni, kezelni és jóváhagyni (ha nincs bekapcsolva az automatikus jóváhagyás), csoportokat létrehozni és kezelni (csoportos fordulókhöz), és meccseket létrehozni, generálni, sorba rendezni és kezelni, valamint az éppen zajló meccset külön kezelni (pontokat növelni/csökkenteni, lezárni ...).



2. ábra: Meccsek kezelése a menedzser felületen

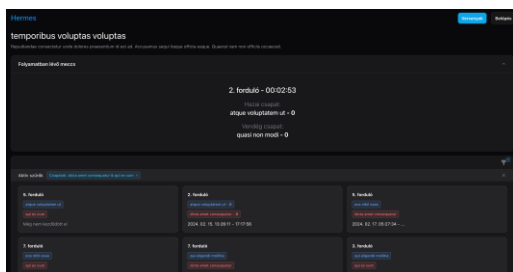


3. ábra: Egy verseny fordulójának kezelése a menedzser felületen (automatikus meccs generáláshoz)

Akik pedig nem rendelkeznek felhasználói fiókkal (azaz a versenyek résztvevői), egy 6 jegyű betűkből és számokból álló kódot kapnak, amelynek segítségével egy verseny állapotától függően tudnak különböző dolgokat tenni.

Ha a regisztrációs időszakon belül járunk, akkor tudnak csapatot regisztrálni (amíg a verseny be nem telt), és a regisztrációkor meg tudják adni a csapattagok e-mail címét. Ha ezt megteszik, akkor egy értesítést kapnak, amikor jóváhagyták a jelentkezésüket, amelyben egy letöltési linket is találnak a Hermes mobilalkalmazáshoz. Ha ezt letöltik, akkor az emailben kapott „hermes://...” linket meg tudják nyitni és össze tudják kapcsolni a telefonos alkalmazást a csapatukkal, így mindig 2 meccsel a sajátjaik előtt kapnak egy „push” értesítést a telefonjukon.

Ha már folyamatban van egy verseny, akkor egy oldalon az éppen zajló meccs adatait látják, valamint egy szűrhető táblázatban (ami telefonra is optimalizáltan jelenik meg) látják a korábbi és jövőbeli meccseket és azok eredményeit. Szintén a folyamatban levő versenyekhez elérhető egy eredményjelző tábla is, amelyet ki lehet vetíteni a helyszínen. Természetesen mindenhol, ahol eredményekről és pontokról van szó, minden élőben frissül websocketeken keresztül.



4. ábra: Egy folyamatban levő verseny kóddal elérhető oldala

2.3. A felhasznált technológiák

Az alkalmazás a Laravel PHP keretrendszer használatával készült. Ez a keretrendszer egészül még ki a Livewire-vel és a Filament-vel. A Livewire teszi lehetővé, hogy ne kelljen minden HTTP kérés után hagyományos PHP stílusban újra tölteni az oldalt, és a Filament egy komponens könyvtár, amely a Livewire-ön alapul. A stílusozás TailwindCSS használatával lett megvalósítva.

A mobil alkalmazás React Native segítségével készült, emiatt elérhető (lenne, de sajnos ennek a költségek nem engedik ezt meg IOS esetén) Android és IOS rendszerre is.

Ezek mellett, használ az alkalmazás konténerizációt Docker-en keresztül, ezt egy ismerősöm, Gyimesi Máté, valósította meg (eredetileg nem ehhez a projekthez) és tette publikusan elérhetővé bárki számára. A jó futási sebesség érdekében pedig nem FPM-en futtatom a PHP-t, hanem a Swoole nevű PHP runtimeon.

Az egész alkalmazás teljes mértékben nyílt forráskódú, és a forráskódja elérhető itt: <https://github.com/Xeretis/Hermes>

3. Elért eredmények

Az elkészült alkalmazás elérte az összes kitűzött célt, és jelenleg is aktívan használatban van az iskolában a bevezető elején említett röplabda verseny szervezésére.

Emellett azt is elérte az alkalmazás, hogy az iskolának más diákjai is beszálljanak a fejlesztésbe kisebb kontribúciókkal Githubon, ezáltal is azt mutatva, hogy van igény az ilyen irányú fejlesztésekre, különösen nyílt forráskóddal, a felhasználókat aktívan bevonva.

Az eddig összegyűjtött technikai statisztikák pedig azt mutatják, hogy eddig teljesen akadálymentesen, hiba nélkül futott az alkalmazás.

BáttyaBoss

Byte brigád

Véber-Jurassa Márk

Felkészítő: Tóásó Előd

TSZC Mikes Kelemen Technikum és Szakgimnázium, 2800 Tatabánya, Béke utca 8.

1. Bevezetés

Az általam kidolgozott pályaműben szeretném bemutatni a Discord szerveremre készített botom fejlesztését és működését. A bot létrehozásakor a fő célom az volt, hogy egy olyan szolgáltatást nyújtsak, amely egyedi és személyre szabott élményt biztosít a felhasználóknak, miközben elkerülöm a más fizetős botokra való költsékezést. A projektem során nagy hangsúlyt fektettem a funkcionalitás és a felhasználói élmény kombinálására. Célom az volt, hogy egy olyan botot hozzak létre, amely egyszerűen kezelhető, ugyanakkor sok hasznos funkciót kínál a felhasználóknak, ezáltal hozzájárulva a Discord szerverünk aktív és élvezetes légköréhez.

2. Probléma megoldásának menete

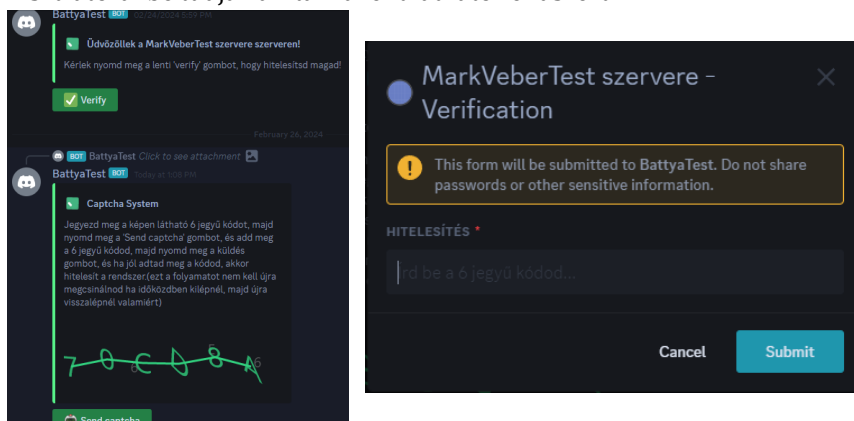
A Discord bot létrehozásához először regisztrálnom kellett a botot a Discord fejlesztői portálján ahonnan le tudtam kérni a bot tokenjét (bejelentkezési adatait), majd hozzá kellett adnom egy teszt szerverhez. A bot fejlesztéséhez a node.js nyílt forráskódú JavaScript alapú környezetet választottam. A telepítés után a discord.js könyvtárat választottam a Discord API kezeléséhez. A rendezett filekezeléshez készítettem egy úgy nevezett file loadert, ami betölti a botomnak az event és command loaderébe a .js fileokat egy megadott mappából a könnyebb kezeléshez, ehhez a glob könyvtárat használtam. A command és event loaderekben nem volt nehéz dolgom köszönhetően a discord.js jó dokumentációjának. Az alapvető munkálatok után következhetett a funkciók fejlesztése, de előtte adatbázishoz kapcsoltam a botot ahhoz, hogy több szerveren is teljesíthesse szolgálatát. Alapvető funkciónak tartottam például hogy CAPTCHA kódokkal akartam a Discord szerveremen a felhasználókat hitelesíteni, ehhez először egy parancsot fejlesztettem ami eltárolja a beállított rangot amit majd a hitelesített felhasználóknak ad illetve elküldi a megadott csatornába az üzenetet a gombbal aminek a megnyomásakor aktiválódik egy event ami legenerálja a 6 karakteres CAPTCHA kódot illetve bekéri azt a felhasználótól, ellenőrzi majd megadja a megfelelő rangot a felhasználónak amennyiben sikerült a hitelesítés.

3. Funkciók, parancsok

A botnak számos hasznos és szórakoztató funkciója van, itt nem is tudok ki térni mindenre, de például fej vagy írás parancsa is van, illetve régebben zenét is játszott YouTube-ról csak azt sajnos egy új YouTube-on bevezetett szabályozás miatt ki kellett vennem.

3.1. Hitelesítő rendszer

A Discord szerveremen Battyaboss kezeli az emberek hitelesítését captha kódokkal. Ehhez tartozik egy parancs is amivel az esetleges szerver adminisztrátorai be tudják állítani a lent látható rendszert.



13. ábra: A hitelesítés menete

3.2. Rang rendszer

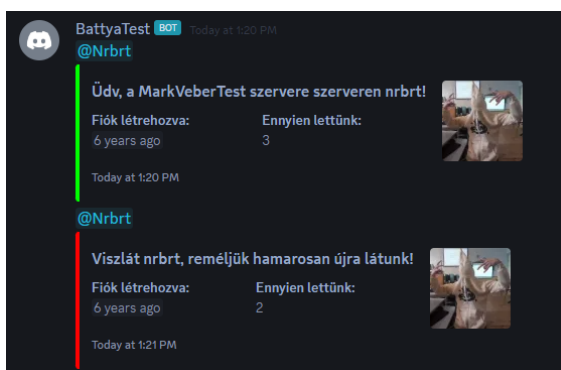
Minden szerveren az adminisztrátorok beállíthatják, ha szeretnék felhasználóikat gratulációval jutalmazni az aktivitásukért, ezt egy event kezeli és csak akkor veszi figyelembe az üzenetet amennyiben nem ugyan az a felhasználó küldte az előző üzenetet, minden üzenetért random mennyiségű pont jár, illetve ehhez tartozik egy parancs is, amivel bárki lekérheti az aktuális szerveren bármelyik felhasználó állapotát.



2. ábra: Rank parancs

3.3. Üdvözlő rendszer

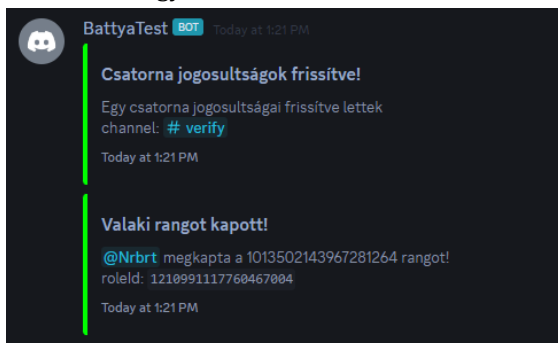
Minden szerveren az adminisztrátorok beállíthatják, ha szeretnék felhasználóikat üdvözölni a bot által, minden felhasználó belépést egy event figyel és amennyiben az adminisztrátor beállította, hogy szeretné a funkciót, illetve, hogy melyik szobába szeretné, az alábbi módon üdvözlő a bot:



3. ábra: Üdvözlő rendszer

3.4. Log rendszer

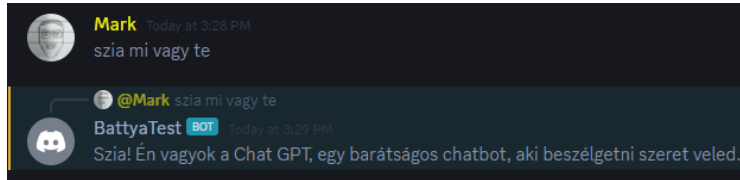
Az adminisztrátorok beállíthatják, ha szeretnék egy csatornát fenntartani a szerver naplójának, a bot ide beküldi a fontosabb módosításokat a szerveren, mint például, ha törölnek egy üzenetet.



4. ábra: Log rendszer

3.5. OpenAI bot

Amennyiben az adminisztrátorok beállítják a szerverükön a bot egy szobában a feltett kérdésekre automatikusan válaszol az OpenAI API segítségével az általuk fejlesztett gpt-3-turbo modell segítségével:



5. ábra: Üdvözlő rendszer

4. Elért eredmények

A Discord botom több szerveren is jelen van közel fél éve, miközben folyamatos újításokkal tehát barátainknak és mellette még egy pár embernek segítséget nyújt a mindennapokban, de legfőképp nekem nyújt segítséget mert nem kell fizetnem valamelyik másik bot testre szabásáért stb. és tudom házon belül futtatni a saját erőforrásaimon. A bot teszt verziója kipróbálható ezen a Discord szerveren: <https://discord.gg/xtpsnijsmTt> A fő bot egyelőre még nem publikus és csak korlátozott számú embernek adtam még oda az invite linkjét, mert minél több felhasználó annál nagyobb erőforrás igénye lenne a botnak. A bot hivatalos github oldala: <https://github.com/MarkVebeer/BattyaBoss>

Eredet: Szövegek eredetiségének ellenőrzése mesterséges intelligencia használatával

eredet.app

Józsa Kristóf

Felkészítő: Szigeti-Pekár Emese

Zafféry Károly Szalézi Középiskola, 2536 Nyergesújfalu, Kossuth Lajos utca 63.

1. Bevezetés

2024-ben a mesterséges intelligenciák által írt beadandók, dolgozatok, házi feladatok egyre népszerűbbek lettek. Ennek az oka a könnyebben rendelkezésre álló MI programok (például: ChatGPT, Google Gemini, Meta LLaMA). Bár már sok ezeket felismerő program van, szinte az összes csak az angol nyelvet támogatja. Projektem készítése során egy könnyen elérhető eszközt próbáltam fejleszteni a magyar szövegek ellenőrzésére.

2. Probléma megoldásának menete

2.1. Adatok gyűjtése

A projekthez szükséges MI modellek készítéséhez sok-sok szövegre volt szükség. A modellek jó teljesítményéhez minél változatosabb szövegeket kell használni, hogy felismerje a különböző emberi írás stílusokat. Az emberek által írt szövegek a cseh Károly Egyetem magyar nyelvi korpuszából (mely az internetet használja forrásként), és további más forrásokból (könyvek, filmfeliratok, érettségi tételek, stb.) lett összeépítve. A nem emberi szövegek a piacon lévő MI szolgáltatások által generált szövegekből lett felépítve. A végső tanulási adatbázis kb. 4000 szöveget tartalmaz, amelynek 35%-a MI által generált és 65%-a ember által írt. Ez az arány általánosságban nehezítené a modellek tanulását (a legoptimálisabb esetben egy 50-50 arány lenne), de ebben az esetben ez az arány nem okozott problémát a modellek eredményeiben.

2.2. Modellek tanítása

A projekthez 3 MI modell lett készítve, ezek a modellek a tanulási fázis különböző szakaszaiban készültek. A modellek alapját a magyar BME egyetem által kiadott huBERT modell képezi. A hárommodelles szerkezet azért előnyösebb, mert vannak szituációk melyben a 3 modell egyidejű használatát véve sokkal pontosabb eredmények születnek. A modellek a transzformer architektúrát használják. A modellek a Google Colab rendszerben lettek

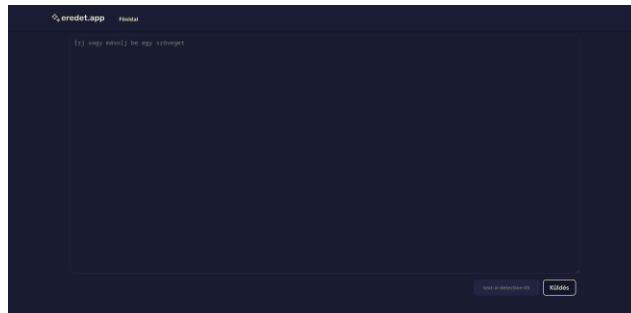
tanítva. A modellek tanítása 2 órát vett igénybe. A modellek 0 és 1 közötti értékeket adnak vissza (0 = emberi szöveg, 1 = MI szöveg), melyet egy 0-100 közötti tartományba átszámolva jelenít meg a felhasználónak.

2.3. Felhasználói felület létrehozása

A projekt egyik fő célja az, hogy a szövegek ellenőrzése minél elérhetőbb legyen. Ezzel a céllal készült el a projekt webes felülete, melyben egy felhasználó könnyedén igénybe veheti a szolgáltatást. A felület a SvelteKit keretrendszerben készült, TypeScript felhasználásával, mely egy gyors és olvasható kódbázist eredményezett.

A webes felületnek a másik fő célja az, hogy tanítsa a felhasználót a mesterséges intelligencia írásáról, és a felméréskor használt szempontok részletes leírásáról. Egy pozitív eredménynél a felhasználónak 4 gyors és egyszerű leírás található az eredmény alatt mely megpróbálja megértetni az elemzési szempontokat, és hogy miben ismerhetők fel az MI által írt szövegek. Ha a felhasználó további információt szeretne megtudni, akkor egy hosszabb formátumban meg tudja ismerni az MI eszközök használatáról, és a projekt modelljei által használt szempontokról. Ez a dokumentáció egy Markdown file-ba van, amelyet a webes kliensek automatikusan megformáznak egy könnyen olvasható honlapba.

A weboldal a felhasználó által begépett vagy másolt szöveget egy Flask-ban írt Python API-nak továbbítja, mely a felhasználó által kiválasztott modellen végigfuttatja az elemzést. Az API egy 1035 szavas mintaszöveget 564 milliszekundum alatt tud elemezni egy GTX 1050 Ti videokártyán (az MI modellek az NVIDIA CUDA technológia segítségével a videokártyán futnak). A projekt készítése során próbáltam követni a mikroszolgáltatásos szerkezetet, amelynek segítségével az alkalmazás hatékonyabban tudja fogadni a felhasználók gyors változását és a felhős szolgáltatások által biztosított előnyöket is fel tudja használni.



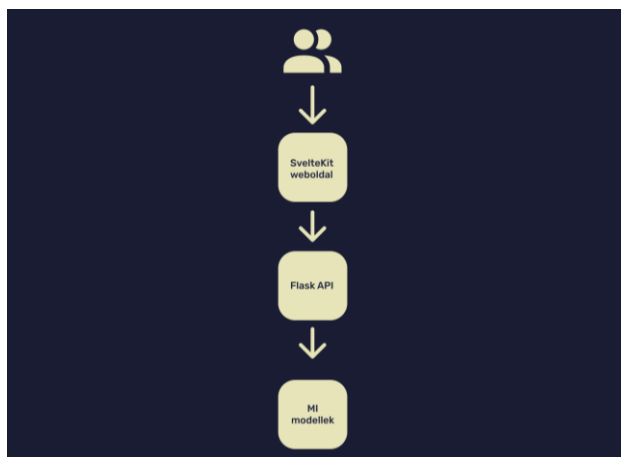
1. ábra: A felhasználói felület



2. ábra: A pozitív eredmény kijelzése



3. ábra: További információk a felismeréskor használt szempontokról



4. ábra: A projekt architektúrája

3. Elért eredmények

A modellek a vártnál sokkal jobb eredményeket hoztak. Ilyen kis modelleknél általában nem várható el a magas pontosság, ezek a modellek nem tanulták át minden ember stílusát. A legjobb modell a 650 szövegből álló validációs adatbázisból 98% pontossággal eltalálta a szövegek eredetiségét. A validációs adatbázis a magyar korpuszból tanuláshoz fel nem használt szövegeket használta, tehát lehet némi eltérés azon az alapon, hogy a szövegek ebben a korpuszban hasonló írásstílust alkalmaznak, de a valós használatban is remek eredményt adtak a modellek.

4. Összefoglalás

A projektem célja, hogy felhívjam a figyelmet a ChatGPT és további eszközök potenciális veszélyeire és előnyeire. A projekt nem azért készült, hogy a ellehetetlenítse az MI-k használatát szövegíráshoz, hanem azért hogy felhívja a figyelmet az MI-k túlzott használatára, és a felelősségteljes használatukra. A projekt a jövőben elérhető lesz a nyilvánosság számára az <https://eredet.app> linken, és bízom abban, hogy ezzel hozzájárulok a biztonságos MI fejlesztéshez.

GraphBox

Ailurus

Kurucz Kristóf

Felkészítő: Motesiczki Ottó

Esztergomi Dobó Katalin Gimnázium, 2500 Esztergom, Bánomi út. 8

1. Bevezetés

A középiskolai tanulmányi versenyeken informatikából elvárt a gráfok megfelelő kezelése. A gráfbejárás a nehezebben elsajátítható tananyagok közé tartozik, ennek a megkönnyítésére írtam a programomat. Célom egy interaktív program létrehozása gráfok vizualizációjára, és a gráfelméleti algoritmusok szemléletes futtatására.

2. A probléma megoldásának menete

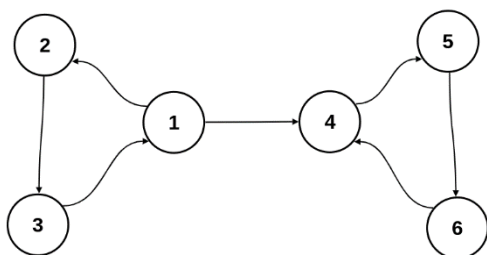
2.1. Mit, miben?

A legelső kérdés az volt, hogy milyen formában szeretném megírni a programot, illetve milyen nyelven. Egy webalkalmazás mellett döntöttem, mert így nagyon könnyen elérhetővé tudom tenni bárki számára. A háttérhez pedig a Blazor keretrendszert választottam, mert a C# terén viszonylag sok tapasztalattal rendelkezem.

2.2. Gráf vizualizáció

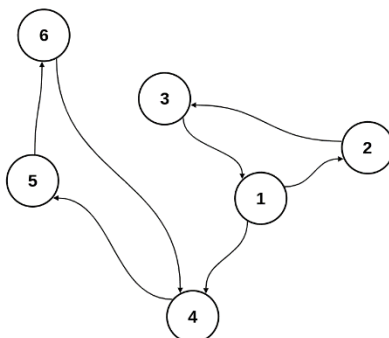
Szerettem volna, hogy maga a gráf jól értelmezhető, interaktív legyen. Ezt a [Blazor.Diagrams](#) könyvtár segítségével sikerült megvalósítanom: a felhasználó könnyedén, kurzorával tudja mozgatni a csúcsokat, eközben az élek automatikusan a megfelelő formába rendeződnek.

Gráf importálás | Új pont felvétele | Új él felvétele | Szélességi bejárás | Mélységi bejárás | Dijkstra | A* | Bellman Ford | Dominátor halmaz | Kosaraju | Kruskal | Elvágtó élek, pontok (Tarjan)



1. ábra: Minta gráf

Gráf importálása Új pont felvétele Új él felvétele Szélességi bejárás Mélységi bejárás Dijkstra A* Bellman-Ford Dominátor halmaz Kosaraju Kruskal Elvágó él, pontok (Tarjan)



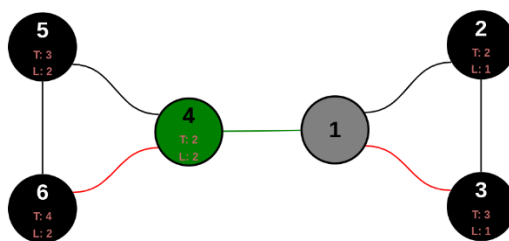
2. ábra: Az 1. ábrán szereplő gráf átalakítva

Új csúcsok és élek könnyen felvehetők az “Új pont felvétele” és “Új él felvétele” gombokkal. A program tud éllista alapján is készíteni gráfot (“Gráf importálása” gomb). Az élek lehetnek irányítottak, irányítatlanok, illetve súlyozottak.

2.3. Algoritmusok

Az algoritmusok implementálásához írtam egy keretrendszert, melynek segítségével nagyon egyszerűen lehet új algoritmusokat hozzáadni a programhoz. Generátorok segítségével lehet töréspontokkal tagolni az algoritmusokat, így a felhasználó tudja őket léptetni, ezzel szemléltetve pontos működésüket. Az aktuális állapotok szemléltetését a csúcsok és élek színezésével, illetve extra információ kiírásával értem el. A két alapvető algoritmussal kezdtem, a szélességi, és a mélységi gráfbejárással, majd ezekre épültek a bonyolultabbak. Az oldal tetején található menüsorban tudjuk kiválasztani, hogy melyik algoritmust szeretnék futtatni.

Lépcső Ugrás Visszaállítás



3. ábra: Elvágó él-kereső algoritmus (Tarjan) egy pontja futás közben.

3. ábra jelmagyarázat:

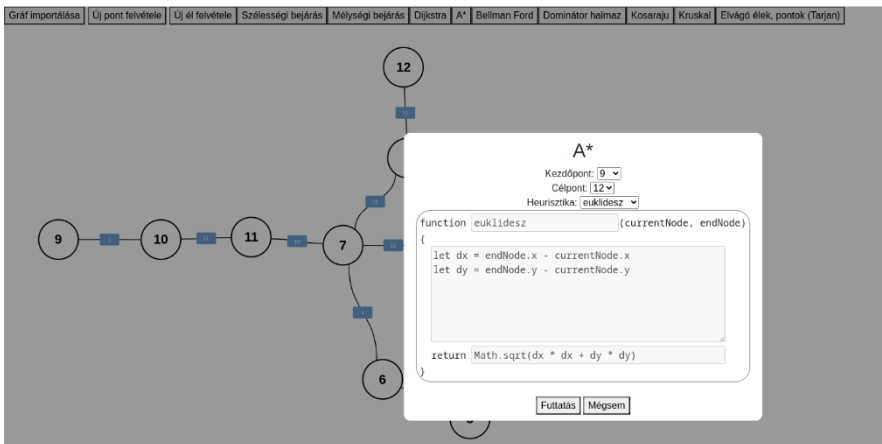
- Szürke csúcs: Bejárás közben elért csúcs
- Fekete csúcsok: Megvizsgált, véglegesített csúcsok
- Zöld csúcs: Elvágó csúcs
- T: Belépési idő
- L: Vissza-élekből elért legalacsonyabb idő
- Fekete élek: Faél
- Piros élek: Vissza-élek
- Zöld él: Elvágó él

Jelenleg elérhető algoritmusok:

- Alapvető bejárás: Szélességi, Mélységi
- Legolcsóbb utak: Dijkstra, A*, Bellman Ford
- Elvágó élek, pontok: Tarjan
- Minimális költségű feszítőfa: Kruskal
- Erősen összefüggő komponensek: Kosaraju
- Dominátor halmaz (minimum szükséges pontok a teljes gráf bejárására)

2.4. Probléma – A* algoritmus

Az A*-algoritmus Dijkstra algoritmusának egy fejlesztett változata, ami a legolcsóbb utat keresi meg A csúcsból B csúcsba. Ennek az algoritmusnak része egy heurisztikus függvény, ami becslést ad az éppen aktuális pont, illetve a célpont távolságára. Ez a függvény nem konstans, a felhasználó szükségletei szerint változhat. A probléma ezen a ponton az, hogy hogyan adjam meg a felhasználónak a lehetőséget tetszőleges heurisztikus függvény megadására. A megoldás végül az lett, hogy az indító paraméterek közé elhelyeztem egy mezőt, amibe a felhasználó megírhatja saját függvényét JavaScript nyelven, vagy kiválaszthatja a kettő alap függvény egyikét (euklideszi-, manhattan-távolság). A `currentNode`, illetve `endNode` paraméterek megadják a csúcs pozícióját, azonosítóját, illetve a kezdőponttól való távolságát (utóbbi csak a `currentNode` esetén).



4. ábra: A*-algoritmus indító paramétereinek megadása

2.5. Probléma – Kiindulási forma

Szükség volt arra, hogy éllista importálása esetén a gráf csúcsai ne egymáson legyenek kiinduláskor. E probléma megoldásában az [MSAGL](#) könyvtár volt segítségemre, ami meg tudja mondani a csúcsok ideális kezdő elhelyezkedését.

3. Elért eredmények

Véleményem szerint sikerült megvalósítanom a tervezett célokat a programmal kapcsolatban. A program tetszőleges nagyságú gráfokat képes vizualizálni, különböző algoritmusokat képes rajtuk végrehajtani interaktívan, lépésenként, ezzel szemléletessé és érthetővé téve azokat. Remélem ezzel a programmal tudok segíteni azoknak, akiknek problémát okoz a gráfbejárás elsajátítása. A projekt elkészítése számomra is hasznos volt, több algoritmussal megismerkedtem a fejlesztés során. Számomra is nagy segítséget jelentett ezek megértésében az, hogy egy vizuális környezetben implementáltam először ezeket, így sikerült megértenem minden egyes lépés lényegét.

Remélem, hogy ez a program másoknak is segít ennek a nehéz témakörnek az elsajátításában. A projekt körülbelül 3 hét alatt készült el.

Felhasznált irodalom:

[Zsakó László: Gráfok 1. előadás](#)

[Zsakó László: Gráfok 2. előadás](#)

[cp-algorithms: Finding bridges in a graph in \$O\(N+M\)\$](#)

Wikipédia: [Dijkstra's algorithm](#), [A* search algorithm](#), [Bellman-Ford algorithm](#), [Kruskal's algorithm](#), [Kosaraju's algorithm](#)

Okosotthon platform Node-RED-ben

Otthonos okos

Patkós Máté

Felkészítő: Bicskei Róbert

Szegedi SZC Déri Miksa Műszaki Technikum, 6724 Szeged, Kálvária tér 7

1. Bevezetés

A mai világban az otthonainkban már egyre több olyan elektronikus berendezés és eszköz található, amiben fejlett informatikai megoldások találhatók. Ezek az eszközök általában a gyártójuk által elkészített rendszerben vezérelhetők. Olyan eset is előfordul amikor ezen eszközök esetleg nem tudnak egymással kommunikálni, így több szoftvert, több gyártó alkalmazását kell használnunk, hogy a különböző okoseszközeinket vezérelni tudjuk.

Általában a nagy okosotthon platformok (Google Home, Home Kit) sem tudják az összes okoseszköz gyártó platformját összegyűjteni. Így egy alternatív platformot érdemes használni amely nyílt forráskódú így szinte minden gyártó eszköze tud csatlakozni hozzá. Ilyen platform a Node-RED ahova szinte minden okosotthon szoftvert kiadó cég felhőbeli szolgáltatását hozzá lehet kapcsolni. Illetve a magunk által készített eszközöket például egy ESP-vel vezérelt lámpa vagy elektronikus eszközt is könnyedén fel tudunk csatlakoztatni erre a platformra.

2. A probléma megoldásának menete

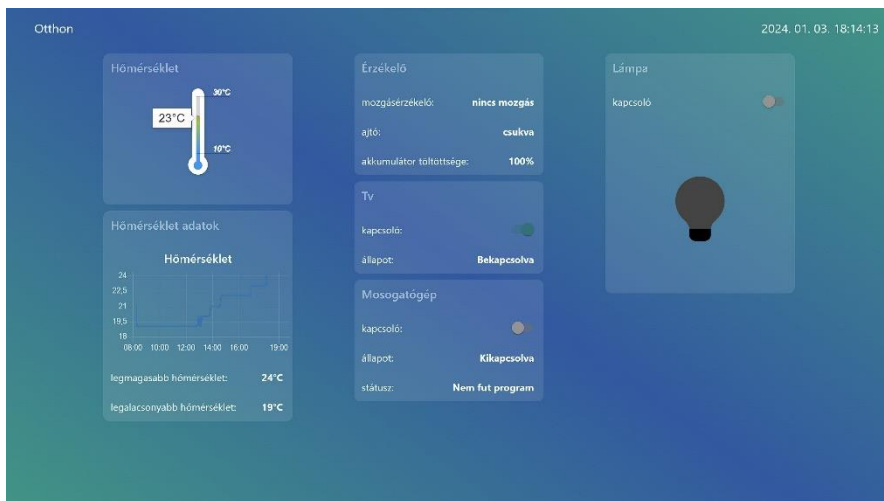
Ebben a projektben egy Raspberry Pi-n futó Node-RED szerver a projekt lelke a mire egy Z-Wave adóvevő van csatlakoztatva, aminek segítségével egy multifunkcionális érzékelő tud kommunikálni a szerverrel. Illetve a platformon megtalálható még az otthoni mosogatógép és tv vezérlése is. Mindemellett egy ESP által vezérelt led lámpa vezérlése is lehetséges.

A projektben felhasznált protokollok: Z-Wave, WiFi, illetve MQTT. Mint ahogy azt fentebb említettem Z-Wave protokollal a multifunkcionális érzékelő csatlakozik a Raspberry Pi-re. WiFi protokollt használ a tv és a mosogatógép amely a Samsung Smartthings felhőbeli szolgáltatásán keresztül kommunikál az MQTT protokollt pedig az ESP használja kommunikációra.

Ebben a szoftverben folyamatábrákkal tudjuk vezérelni azt, hogy az adott eszköznek milyen adata legyen kiírva a szoftver által biztosított internetes felületen. Ezen a felületen az eszközeinket is tudjuk vezérelni feltéve, ha az funkciójából adódóan lehetséges.

A felhasználói felületen lehetőségünk adódik arra hogy az érzékelőből kinyert hőmérséklet adatot lássuk illetve grafikonon elemezni tudjuk annak változását. De akár a tv és a mosogatógép be és kikapcsolását is el tudjuk végezni szintűgy, mint az ESP-vel vezérelt lámpa is fel- és lekapcsolását. A tv és a mosogatógép jelenlegi állapotát is figyelemmel tudjuk kísérni. A multifunkcionális érzékelő ajtónyitó és mozgásérzékelő szenzorának jelenlegi állapota is látható.

Az ESP Python kód segítségével tud kommunikálni a Raspberry Pi-n található MQTT szerverrel, illetve ez a kód is adja ki a jelet a rákötött relének, amely vezérli a saját készítésű lámpánkat.



14. ábra: A felhasználói felület

3. Elért eREDmények

Ez a program azért fontos, mert a jövőben minden háztartásban okosotthon megoldások lesznek megtalálhatóak, így ehhez a programhoz csak hozzá kell adnunk az általunk választott okoseszközöket, amelyek idővel bekerülnek az otthonunkba így a későbbiekben az egész otthonunkat vezérelni és felügyelni tudjuk ennek a platformnak köszönhetően.

Az előnye ennek a platformnak, hogy ellentétben a nagy cégek szoftverével, ebbe szinte az összes okoseszközt képesek vagyunk csatlakoztatni. Hiszen a nyílt forráskód miatt minden gyártó szoftverét integrálni tudjuk. Emiatt széleskörűen tudjuk használni.

Előnyei közé tartozik még, hogy a felhasználói felületet magunknak tudjuk személyre szabni, amelyet a hálózatban lévő bármilyen böngészővel rendelkező eszközzel el tudunk érni.

Az adatok megjelenítése, kiértékelése és feldolgozása is személy szerint változtatható a magunk általkészített folyamatábrás programozásnak köszönhetően.

A projekt innovativitását az is jól bemutatja, hogy az okosotthonok nem elterjedtek manapság, de mivel ehhez a szoftverhez könnyedén hozzá tudunk adni bármilyen okoseszközt. Így akár egy régi hazát is könnyedén tudunk felokosítani és vezérelni.

Műszaki informatika szekció

Kételtű drón

E.R.SZ.

Kucsera István, Szántai Ádám

Felkészítő tanár: Lang Ágota

Széchenyi István Gimnázium, Sopron

1. Bevezetés: miért?

A Robotika szakkörünkben egy kis csapat CanSat építésével töltötte a tavalyi évet, amit szerettek volna tesztelni: hogyan ereszkedik az ejtőernyő segítségével, küldi-e az adatokat, stb. Egy ismeretség révén a fertőszentmiklósi repülőtéren egy girokopterrel emelkedett magasba, hogy aztán egy repcetáblában landolva örökre eltűnjön. Talán ha már az ereszkedés közben nyomon követhettük volna egy drónról figyelve/rögzítve a mozgását, sikeresebb lett volna a keresés. Ez volt az indíttatás, hogy drónt építsünk nekik. Azonban, ha már belefogunk, csináljunk valami olyat is, ami érdekesebbé teszi a projektünket: innen jött a kételtű drón ötlete – mi lenne, ha a víz alá is láthatnánk? Egy videó ami az átlátszó kék vízben kezdődik, majd a part fölött az égbe szökik – valószínűleg élvezetes látványt és élményt adna.

2. Hogyan? – Kell egy terv

Az ötletgazda fejében a végső modell egy csepp alakú, 2 + 2 motoros drónként jelent meg. Matt fekete színével, gyorsan hasít az égben miközben az éjszaka leple öleli körbe.

2.1. Az elérni kívánt cél, vagyis a kételtű dreamdrón felépítése

2 légcsavaros motor lenne egy csepp alakú borítás kiszélesedő részén, a drón síkjával párhuzamosan. A cső másik végében lenne a kettő hajócsavar, egy eggyel kisebb, de erősebb, vagy ugyanakkora, de nagyobb áttételű motorral. A kormányzást egyrészt a testen található kinyitható szárnyakkal oldanánk meg, valamint a kereszt alakú farokszárnyakon lévő irányítható bevágásokkal, ezeket szervo motorok mozgatják.

Tekintettel a használat helyszíneire, vannak bizonyos, a természetből fakadó problémák és akadályok:

- a motorokat vízálló rekeszekbe kell tenni, hogy víz alatti működés során ne kapjanak zárlatot.
- a kétféle motor mindegyike egy-egy csőben kap helyet, a hatékonyság növelése érdekében

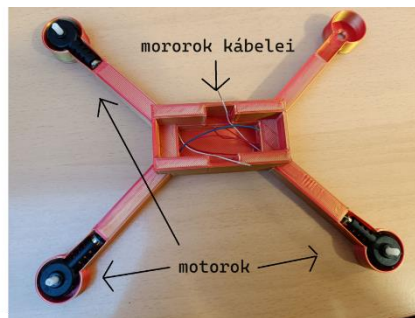
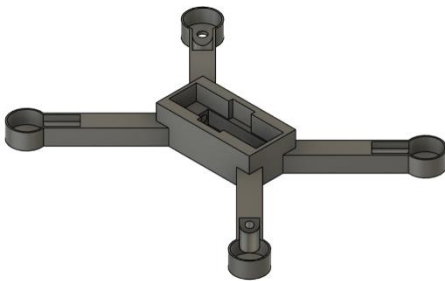
A tranzisztorok további szerepe még, hogy lehetővé teszik a nagyobb feszültség befolyásolását kisebb feszültséggel. Így az arduino digitális pinjeivel tudjuk irányítani akár az 5V-nál nagyobb feszültségigényű motorokat is.

Mivel a motorok a drón síkjával párhuzamosak, a dreamdrón nem tud egyhelyben lebegni. Megadnánk neki egy hely két földrajzi koordinátáját, valamint egy magasságot, amiben repül. A térben való tájékozódásban a beépített GPS illetve iránytű szenzora segíti. A megadott pont vagy egy keringési középpont lenne, vagy egy oda-vissza úthoz szolgálna.

Landoláskor először a földre merőlegesen felfelé száll egy kis ideig, majd a motor erejét lassan gyengítve, leszáll a hátsó lapátjaira. Az önálló repülés mellett lehetőséget adunk a beavatkozásra rádiós kapcsolat révén.

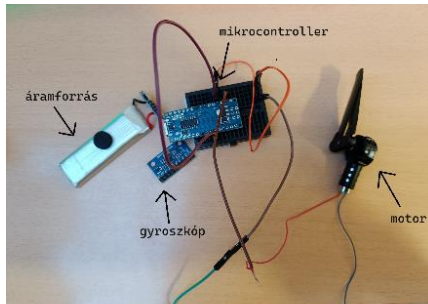
2.3. Előkészületek: a tesztdrón

Természetesen nem a végső modellel kezdtünk neki a munkának, hanem előtte egy kisebb, kalibráláshoz, és tanuláshoz használatos drónt építettünk. Ezt a kisebb drónt egy „kereső” bevetésen tudjuk tesztelni, az új CanSat röptetésekor. Ezen az úton lehet kalibrálni az Arduino nano-t, ahhoz, hogy a drónokat egyhelyben tudja lebegtetni.



3. ábra: A tesztdrón terve és megvalósítása

Ez a drón egy hagyományos 4 motoros, avagy quadcopter drón mintájára készült. Saját tervezésű 3D nyomtatott testtel rendelkezik, amiből négy irányba egy-egy motorállás ágazik ki. A központi rekeszbe kerül az Arduino nano és MPU 6500 szenzor, melléjük pedig az akkumulátort teszünk, míg a motorokat pedig a nekik szánt állásokba.



4. ábra: A rekesz tartalma még a szerelőlapon

A végső modellhez képest annyi változás van a működési elvében, hogy nincsenek szervo motorokkal irányított szárnyak. A kormányzást minden esetben a motorok végzik, azok fordulatszámának befolyásolásával tudjuk a drónt irányítani.

3. Elért eredmények, további lépések

Jelenleg elkészült a tesztdrón. A végső célhoz vezető következő lépés egy GPS szenzor beépítése a drónba és a programba, ami az irányítás fő eleme lenne. Másrészt a tesztdrón aljára ultrahangos szenzorok kerülnek még, amelyek segítségével érzékelni tudja, ha vészesen közel kerül a földhöz. Ekkor a program automatikusan felrepteti.

Az adatokat és tapasztalatokat, amit tesztdrónból szerzünk, használjuk majd fel a végső modell kialakítása és elkészítése, valamint működtetése során.

Az esetleges áprilisi bemutatóig a dreamdrón szerkezetének tervei és a 3D nyomtatás biztosan elkészül, a beüzemelése azonban nem valószínű.

QR-kód alapú hulladékszortírozó rendszer

Bits&Recyclers

Bartha Jenő

Felkészítő: Bicskei Róbert, Gál Csaba

Szegedi Szakképzési Centrum Déri Miksa Műszaki Technikum

6724 Szeged, Kálvária tér 7.

1. Bevezetés

Meggyőződésem, hogy napjainkban egyre fontosabb a környezetvédelem, a fenntartható jövő érdekében való cselekvés. A fejlett országokban a felnövekvő generációknál jelentős hangsúlyt kap a környezettudatosagra való nevelés. A folyamat már óvodákban elkezdődhet, de legkésőbb általános iskolában a diákok találkoznak ezzel a fontos témakörrel.

A hulladékok megfelelő gyűjtése is erősen szem előtt tartandó szempont a fenntarthatóságra való nevelés során. A Szegedi Szakképzési Centrum Déri Miksa Műszaki Technikumban a 2023/2024-es tanévben indult el a zöldDÉRIért kezdeményezés, melynek fő célja pontosan a környezettudatos gondolkodásra való nevelés. A kezdeményezés keretében az iskola pedagógusainak és tanulóinak lehetősége van szelektív módon elhelyezni, kezelni a mindennapok során keletkezett hulladékokat, csomagoló anyagokat, PET-palackokat stb.

Mivel a téma személyes indíttatásból is érdekel, másrésről pedig informatika szakos tanuló vagyok, mindig is kíváncsi voltam, hogy hogyan tudná egy-egy informatikai megoldás segíteni a szelektív hulladékgyűjtést és kezelést. Jelentősebb internetes kutatómunkát követően bukkantam rá a TOMRA Systems ASA cég számomra lenyűgöző, egyébként rendkívül komplex hulladékkezelést megvalósító berendezéseire.

Ennek kapcsán támadt az az ötletem, hogy mi lenne, ha ugyan jóval kisebb léptékben, de én is megpróbálnék megvalósítani egy olyan berendezést, amellyel a mindennapi iskolai életben keletkező hulladék legalább egy részét szelektíven lehessen gyűjteni. Ennek okán született meg a következő oldalakon röviden bemutatásra kerülő hulladékszortírozó berendezésem.

2. A probléma megoldásának menete

2.1. Az első lépések

Először is azt kellett meghatároznom, hogy milyen típusú hulladékokra helyezem a hangsúlyt. A választásom a fél literes PET-palackos, illetve

aludobozos italokra, üdítőkre esett, mivel a különböző büfékben és automatákból ilyeneket lehet venni legnagyobb számban.

Másrészről mivel a projekthez szükséges szenzorokat, aktuátorokat saját forrásból szereztem be, így nem volt lehetőségem arra, hogy nagyobb volumenben gondolkozzak. A tápellátáshoz szükséges készülékeket, szerelvényeket pedig iskolám bocsájtotta rendelkezésemre. A mechanikus szerkezeti elemeket pedig keresztapámmal közösen készítettük el. Értendő ezalatt a berendezés dobozolása, illetve a hulladék szortírozását végző lengőkar mechanikus részegységeinek a kialakítása, amelyeket később fogok bemutatni röviden. Ezután következett annak a kérdésnek a megválaszolása, hogy a szétválogatást mi alapján fogom végrehajtani.

2.2. Az egyedi QR-kód, és a kódolvasó szenzor

Hosszas ötletelés után végezetül arra a döntésre jutottam, hogy egyedi QR-kódot készítek, annak érdekében, hogy eltároljam benne a következő két információt: újrahasznosítható-e és hogy milyen anyagból készült. A QR-kód mellett szólnak további érvek is, mint például az alfanumerikus karakterek maximális száma, vagy a hibajavítási képesség. Az általam elgondolt kód 7 karakterből áll. Ezt mutatja be az 1. táblázat.

Egyedi QR-kód felépítése	
Újrahasznosíthatósági jelző	Jelentése
0REC	Újrahasznosítható
1REC	Nem újrahasznosítható
Anyag azonosítója	Jelentése
PLA	Plastic = műanyag
MET	Metal = fém

1. táblázat: Az egyedi QR-kód felépítését bemutató táblázat

Természetesen ahhoz, hogy a QR-kódokat letudjam olvasni, szükségem volt egy kódolvasásra képes szenzorra, melyre megoldásként a Barcode Scanner Module (B) típusút választottam, mely gyorsan és pontosan képes a különböző kódformátumokat dekódolni. A következő lépés nem volt más, mint a működési elv, és az ahhoz szükséges eszközök meghatározása.

2.3. Szerkezet működési elve, és a szükséges eszközök

A tervezési, kivitelezési fázis alatt az egyik fő szempont az volt, hogy amennyire lehet automatizáltan működjön a berendezés. A működési elve a következőkben olvasható. Két egymás mellé elhelyezett hengert dörzshajtás segítségével egy 30 RPM-es 12 V-os DC motor megforgat, mely hengerekre egy

nyíláson keresztül helyezük rá a vizsgálni kívánt flakont. A hengerektől balra lévő falon kap helyet egy HC-SR04 típusú ultrahangos távolságmérő szenzor, melynek feladata, hogy a hengerekre érkező flakonokat érzékelje, és elindítsa a folyamatot. A hengerek végénél lévő meghosszabbított karral rendelkező Tower Pro MG996R típusú mikroszervó megtámasztva tartja a flakont a leolvasási folyamat végéig. A kódolvasó szenzor egy rövidített szárú U alakú fémlemezre van rá rögzítve, mely a hengerek alatt található alumíniumlemezre van csavarozva. A kódolvasó szenzor rögzítése mellett található a motorvezérlő áramkör, mely a mikrovezérlő által előállított PWM jel függvényében állítja a DC motor fordulatszámát. A nyílás után közvetlen található keret feladata, hogy védelmet nyújtson az esetleges kiesések ellen. A felhasználóval történő kommunikációra egy I2C modullal ellátott LCD kijelzőt választottam, mivel a hátulján lévő modul nagymértékben megkönnyíti a későbbi bekábelezést. A szemetesekben lévő hulladékok szintjének a mérésére a folyamat beindításához is használt HC-SR04 típusú ultrahangos távolságérzékelő szenzorra esett a választásom. A szerkezet áramellátása, illetve a rendszer lelkét adó Arduino UNO R4 WiFi típusú mikrovezérlő a hengerek bal oldalán kaptak helyet. A hengerek bal oldalán található még a tartó szervóval megegyező típusú mikroszervó, melynek feladata, hogy a leolvasott QR-kód függvényében irányba állítsa a hengereket tartó alumíniumlemezt a megfelelő szemetes felé.

3. Elért eredmények

Végezetül szeretném összefoglalni az elért eredményeket, és felvázolnék néhány továbbfejlesztési lehetőséget. A szerkezet jelenlegi állapotában két fajta hulladékot (PET-palack, alumínium doboz) képes megkülönböztetni QR-kód alapján. Az alkatrészek számára a megfelelő tápellátás biztosított. A felhasználóval történő kommunikáció megoldott.

Továbbfejlesztési lehetőségeket is látok még a rendszerben, amelyek közül most párat szeretnék felsorolni:

- Gépi látás / intelligens objektum felismerő algoritmusok – a hulladékok még pontosabb és szélesebb körű válogatására
- Mobilalkalmazás / weboldal fejlesztése – a jobb minőségű felhasználói kommunikáció érdekében
- Pneumatikus présrendszer – amely képes lenne összenyomni a PET-palackokat, illetve az alumínium dobozokat

Meggyőződésem szerint a bevezetőben leírt célokat sikeresen megvalósítottam, és ennek a projektnek köszönhetően új szakmai ismereteket is szereztem, mint például:

- Szenzorok, aktuátorok használata
- Áramkörök kialakítása
- Mikrovezérlők programozása

Trináris alapú számítógép

Három a magyar igazság!

Bábel Edvárd István

Felkészítő: Kámán Tamás

Kiskunfélegyházi Móra Ferenc Gimnázium, Kossuth Lajos utca 9

1. Bevezetés

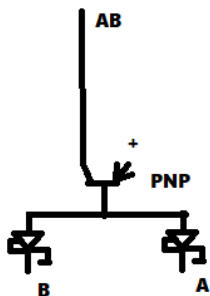
Régebben, az informatika hajnalán, próbálkoztak nagyobb bázisú rendszerekkel, mint a bináris. De a bonyolultsága, és mérési pontatlanságai miatt nem voltak annyira megbízhatóak, mint a bináris. Ez óta eltelt 100 év, és a bináris lett a megkérdőjelezhetetlen standard. Viszont mi lenne, ha nagyobb bázisú rendszert építenénk? Akkor több adatot tudna tárolni, gyorsabban futtatná a programokat (hisz több adatot tárol 1 memóriarészleg, amin számolunk) így hatalmas előnyben lenne az új rendszer. Emellett, másfajta számításokat, mint a mesterséges intelligenciához szükséges számítások, kvantumfizikai számítások, több dimenziós mátrix számítások igényelnek, gyorsabban tudná számítani. Egy nagyobb számrendszer, amit használhatunk az a trináris. Ennek a mérése is igen egyszerű, elég megkülönböztetni a 0, pozitív és negatív feszültségeket. Így ugyan annyi kábellel, alkatrészszel, sokszor több adatot tárolhatnánk, futtathatnánk. Máshogy nem lenne hatékony, hisz ha feszültség alapján mérnénk, akkor durván függene, hogy párhuzamosan vagy sorosan van kapcsolva az áramkör. Ami esélyesebb az a váltóáram, és a frekvenciával számolni, de azt mérni pontosan nagyon nehéz, és a váltóáramban megfigyelhető skin effektus miatt, stb. Sok problémát hoz fel.

2. A probléma megoldásának menete

Viszont kiépíteni egy trináris rendszert így is elég bonyolult, hiszen a logikai kapuknak az áramkörét teljesen át kell építeni. Ki kell építeni egy programozási nyelvet, memóriát, logikai rendszert, operációs rendszert teljesen nulláról. Elejében vákuumcsövekkel akartam megoldani, mert megtetszettek, és még soha senki sem miniatürizálta őket, de rájöttem, hogy a vákuumkompresszor, ami a laborban van túl gyenge hozzá, és egy vákuumcső megépítése nagyon bonyolult, még a „hideg” vákuumcső is hihetetlenül bonyolult. Így kénytelen vagyok a tranzisztorokra támaszkodva megépíteni, de ha tudok, megpróbálok belehinteni ide-oda vákuumcsöveket. Logikusan Változtatni lehet, hogy melyik milyen arányban van benne, vagy akár kivenni belőle teljesen a vákuumcsöveket, ha nem hatékonyak.

2.1. A logikája

A logikai kapukat át kellett írnom. Kerestem logikai kapukat internetről, de ránézve egyáltalán nem logikusak, mert csak plusz egy táblázatot hozzáadtak, mint egy mátrix egyenlethez és pipa, arra nem rájőve, hogy $1 + 1 = -1$ trinárisban, nem pedig 0, mint binárisba ahol $1 + 1 = 0 + 1$ a következő kábelben (10) A Trináris and ki/bemeneti logikáját az 1. táblázatban lehet látni. Persze, emiatt saját áramkört kellett terveznem. (mintha találtam volna normális trináris ármkört.) 1. ábrában látható az áramkör.



15. ábra: A trináris „or” áramköre

Trináris „and” (összeadásra optimalizálva)			
or	0	1 (+)	-1 (-)
0	0	0	0
1 (+)	0	0	1 (+)
-1 (-)	0	1 (+)	1 (+)

2. táblázat: A trináris „or”

3. Elért eredmények

A logikája és az alap áramkörei készen állnak, már csak meg kell építenem/ elküldeni valahova és megépíttetni az áramköreit. Azután csak rá kell programoznom programokat, és valamilyen adattárolási módot kell hozzáadnom, valószínűleg mechanikait, de lehet SSD alapút fogok használni,

mivel sokkal könnyebb azt megépíteni. Ez a rendszer végtelenül előnyösebb lenne a

- Kvantumszámítógépek,
- Mesterséges intelligencia,
- Memória,
- Analóg számítás,
- Szimulációk

terén. És unblock gyorsabban tudna számolni mindenhol, akár ha az egész internetet átalakítanánk trinárisra, logikusan gyorsabb lenne, hisz több „jelet” küldhetünk egy kábelen keresztül.

Kontaktmentes anyagvizsgáló fémekhez és mágnesekhez

Prototech Instruments

Burger Alfréd

Felkészítő: dr. Piláth Károly

ELTE Trefort Ágoston Gyakorló Gimnázium, 1088 Budapest, Trefort utca 8.

1. Bevezetés

Mindennapi életünk során kerülhetünk olyan helyzetbe, hogy egy előttünk levő fémdarab valamely tulajdonságát gyorsan és egyszerűen szeretnénk megmondani. Ehhez megvizsgálhatjuk az ötvözet mivoltát, ami azonban időt, eszközöket és tudást igényel, továbbá mintavétellel, azaz roncsolással jár. A sűrűség vizsgálata se mindig megoldás, ugyanis ennek a folyamata is körülményes. Van-e arra módszer, hogy mindezt egyszerűbben vizsgálhassuk, valamilyen műszer segítségével?

2. A probléma megoldásának menete

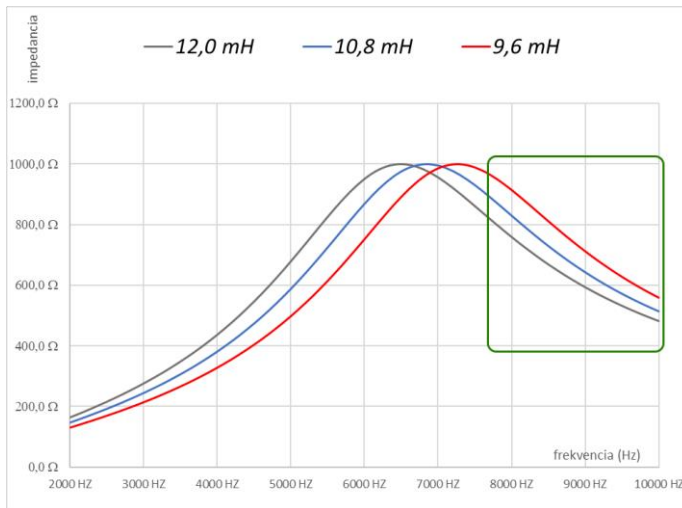
2.1. Elmélet

Egy tekercs induktivitását megváltoztatja egy, az indukált mágneses térbe helyezett vezető vagy mágnes. Ha ezt a tekercset nem önmagában, hanem egy RLC rezgőkörben alkalmazzuk, és ezt szinuszos váltóárammal hajtjuk meg, két tulajdonságot mérhetünk jól: a kör látszólagos ellenállását (impedancia Z) és a tekercs induktivitását (L). Minket itt a látszólagos ellenállás érdekel. A legegyszerűbben úgy tudjuk ezt mérni, ha sorba kapcsolunk vele egy fix ellenállást, és az arra eső feszültséget mérjük, ugyanis egy soros kapcsolásban az ellenállások aránya megegyezik a rájuk eső feszültségek arányával (1).

$$\frac{R_1}{R_2} = \frac{U_1}{U_2} \quad (1)$$

A másik tényező az impedancia és a tekercs induktivitása közötti összefüggés. Ha kirajzoljuk az impedanciát (2) a frekvencia függvényében különböző induktivitásokra, mint az 1. ábrán látható, észrevehetjük, hogy lesz olyan frekvenciatartomány, ahol L és Z között monoton korreláció van.

$$\frac{1}{Z(\omega)} = \sqrt{\frac{1}{R^2} + \left(\frac{1}{L\omega} - C\omega\right)^2} \quad (2)$$



16. ábra: A kirajzolt grafikonok

A „mag”, tehát a vizsgált fém vezetőképessége egyenesen arányos a benne indukált Örvény-áramokkal, tehát a tekercs mágneses mezőjét fékező hatással is. A tekercs eredő indukciója tehát csökken.

Megj.: A Lenz-tv. miatt ez mindenképp fékező hatásként fog jelentkezni.

Megj.: Ismert dimenziójú (kiterjedésű és vastagságú) mintánál ezt a hatást számszerűen is ki tudjuk számítani.

E két pont és az utóbbi, ismert elmélet alapján kimondható, hogy a vizsgált fém vezetőképessége fordítottan arányos a fix osztóellenálláson mért feszültséggel a megfelelő frekvenciatartományban. Ebben a gondolatmenetben megjelenik a tekercs induktivitása is.

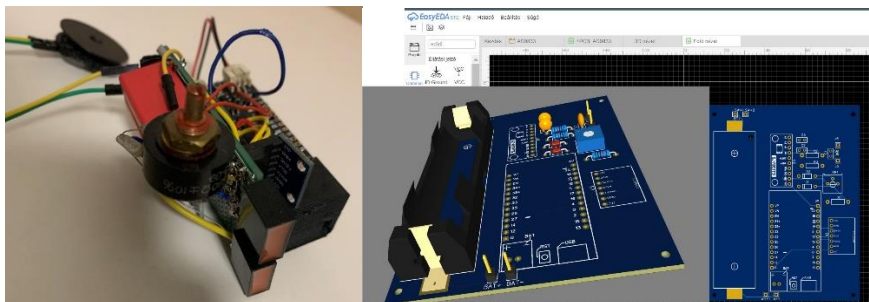
Ennek köszönhetően a mért értékhez kalibrálhatunk mintákat, természetesen a minták dimenzióit figyelembe véve.

2.2. Az eszköz és az applikáció

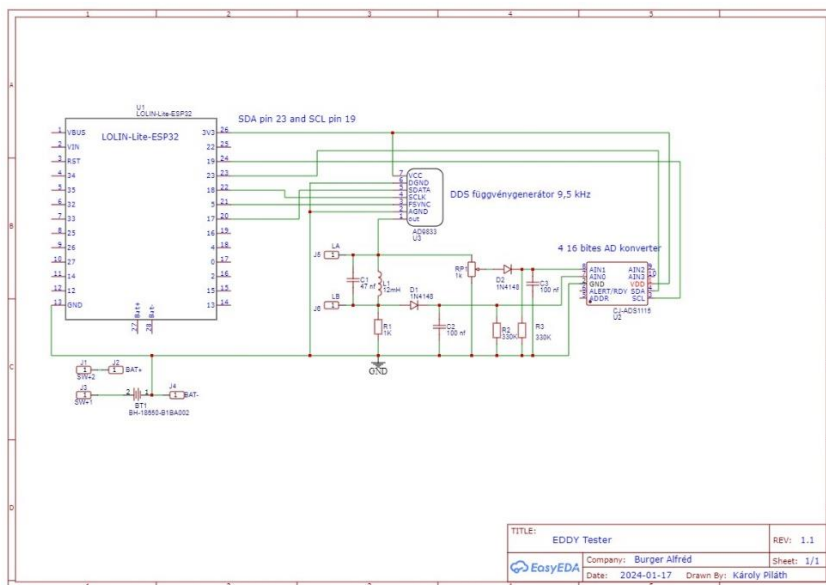
A műszerhez egy ESP32 (Wemos Lolin 32 Lite) mikrokontroller, egy függvénygenerátor (MD9833) és egy AD-konverter (Adafruit AD1115) IC szükséges. A mérőeszköz a 2. ábrán látható. A műszer méri az előző pontban említett ellenállásra eső feszültséget (ami helyett potenciométert építettünk be a tesztelhetőség miatt) és a generátorból kijövő önálló jelet is, aminek a jelentőségére később még visszatérek. A műszer kapcsolási rajza a 3. ábrán látható. A rendszert az ESP vezérli, és az eredményeket Bluetooth-on keresztül

küldi telefonra. A pontosság növeléséért 150 minta átlagát tekintjük egy mért értéknek.

A telefonos Android applikáció alapjaként szolgált egy „open source” GitHub projekt¹, mely kiváló minőségben kezelte a Bluetooth soros monitort. Ezt a mi igényeinknek megfelelően módosítottuk, egyelőre azonban csak minimálisan.



2. ábra: A műszer prototípusa (bal) és kész terve (jobb)



3. ábra: A műszer kapcsolási rajza

2.3. Továbbfejlesztési lehetőségek

A korábban említett két csatorna mérése lehetővé teszi a két jel között (RLC által okozott) fáziskésés mérését is. Így lehetséges lesz L számszerű mérése, amivel, és egy „üres” referenciaméréssel az anyag mágneses tulajdonságait tudjuk kiszámítani. Levezettük, hogy az induktivitás aránya megegyezik a relatív permeabilitás arányával.

Megj.: Vezető ferromágnesnél felhasználástól függően érdemes lehet figyelembe venni az örvényáramok hatását is. Ismert fémnél ez nem gond.

Megj.: E híján a rezonanciafrekvenciával lehetne meghatározni L-t, amihez azonban spektrumot kell mérni, ami jóval tovább tart.

Ezekkel az az adatokkal már csupán szoftver és kalibráció kérdése, hogy mire használható a műszer. Jelenleg a minta anyagát mondja meg, a későbbiekben azonban szeretnénk anyagvastagság, fedőréteg-vastagság / távolságméréshez használni. Az applikáció segítségével felhasználói oldalról is kalibrálható lesz a rendszer. Gondolkodunk még törésvizsgálaton is (ebben az esetben az Eddy-áramok máshogy folynak), de ez még további fejlesztéseket igényel.

Természetesen az applikációt is meg fogjuk tervezni a célnak megfelelően, de előbb a fent említett funkciókat szeretnénk hozzáadni.

3. Elért eredmények

A kifejlesztett és fejlődő rendszerünk alkalmas roncsolásmentes anyagvizsgálatra. Kalibrációtól és programozástól függően lehetséges többek között az anyag ötvözetének, mágneses és elektromos tulajdonságainak, vastagságának és rétegvastagságának gyakorlatilag kontaktmentes mérése telefonról is.

Felhasználható például fémelemek beazonosításához, mágnesek tulajdonságainak, vékony (néhány mm-es) fémlamezek vastagságának, fémlamezek festékvastagságának megméréséhez, vagy falban vezetékek kereséséhez.

Mindez természetesen függ a felhasznált tekercstől, illetve a rendszer zajosságától.

Folyamatban van a mérőeszköz „gyári” kalibrációjának kidolgozása, mely minden jövőbeli módhoz tartalmazni fogja a szükséges adatokat, azonban lehetőséget szeretnénk hagyni egyedi kalibráció elvégzésére is.

Cérna Mutatvány

Infosok

Kun-Baumann Bálint

Felkészítő tanár: Motesiczki Ottó

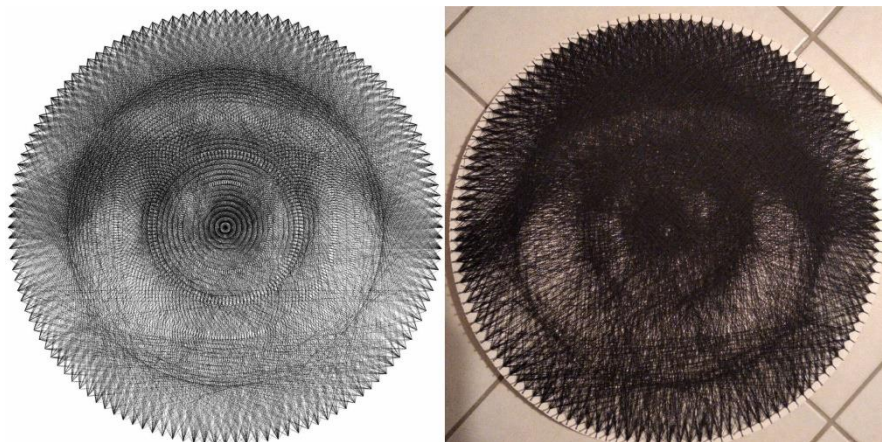
Esztergomi Dobó Katalin Gimnázium, 2500 Esztergom, Bánomi út 8.

1. Bevezetés

Egy évben sok olyan alkalom van, amikor szeretteinknek, barátainknak kötelesek vagyunk ajándékot adni. Nekem ez gyakran fejfájást okoz, ezért építettem egy gépet, hogy ezt megcsinálja helyettem. Ezt megtervezni, leprogramozni, összerakni, tesztelni, az alkatrészeit legyártani és megvenni sokkal drágább volt, mint ajándékot venni és sokkal időigényesebb, mint bármi ilyesmit kézzel elkészíteni.

Egy string art készítése egy megfelelő kép kiválasztásával kezdődik, amiből a vonalazó program kiszámolja a meghúzendó vonalakat, amit az irányító program átküld a gépnek.

Egy string art (1., 2. ábra) egy kör alakú falapból, ennek szélébe bevert szögekből, és **egy** szál cérnából áll. Az így alkotott képek általában csak messziről nézve láthatóak jól, ez (pl. az 1. ábránál) jelentheti a szoba másik végét is.



1. ábra: Szempár

2. A probléma megoldásának menete

2.1. A cél

Személyre szóló, kézzel készített ajándékok tömeggyártásán kívül a fő célom az volt ezzel a projekttel, hogy nyílt-forráskódú (a pályázat végen fel tervezem rakni GitHubra) és hozzáférhető legyen. Egy ilyen látványos, egyszerűen érthető, otthon is könnyen elkészíthető gép olyan emberekben is felkeltheti az érdeklődést a műszaki tudományok iránt, akiket eddig a hangos marógépek és ronda parancssor ablakok nem fogtak meg.

A szoftvert szerettem volna gyorsra és szépre megírni, úgy, hogy közben megtanuljak egy új alacsony szintű programozási nyelvet. Ez a 3 három cél a projekt felénél összeegyeztethetetlennek tűnt, de a végére nagyjából sikerült.

A gépet minél olcsóbbra szerettem volna tervezni és csak olyan elemekből összeállítani, amelyek vagy készen megvehetők (mikrovezérlők, motorok, táp stb.) vagy otthon legyárthatók.

2.2. Szoftver

A szoftver 3 nagy részből áll:

- A vonalazó program (Rust)
- Az irányító program (Python)
- Arduino program (C++)

2.3. A vonalazó program

Ez a program számolja ki, hogy mely szögeket kösse össze a gép. Elsőként feldolgozza a képet: átalakítja monokrómra, átméretezi, kontrasztot állít.

Ezután egy körben kijelöl N pontot, fizikailag ezek lesznek a szögek. Azt, hogy mely szögek összekötése a legoptimálisabb egy mohó algoritmus számolja ki. A program kiszámolja, hogy mely egy N szögből (kezdetben a 0. szög) mely másik szögbe húzott szakasz csökkenti legjobban a készülő kép hibáját az eredeti képhez képest (1-es egyenlet), a legnagyobb hibacsökkentési értékkel rendelkező szakasz kerül behúzásra a készülő képen, a szakasz másik végpontja lesz N a következő iterációban.

$$E = \sum_{i=0}^n (P_i - (F_i + L_i) \geq 0 \rightarrow L_i) \wedge (P_i - (F_i + L_i) < 0 \rightarrow P_i - F_i) \quad (1)$$

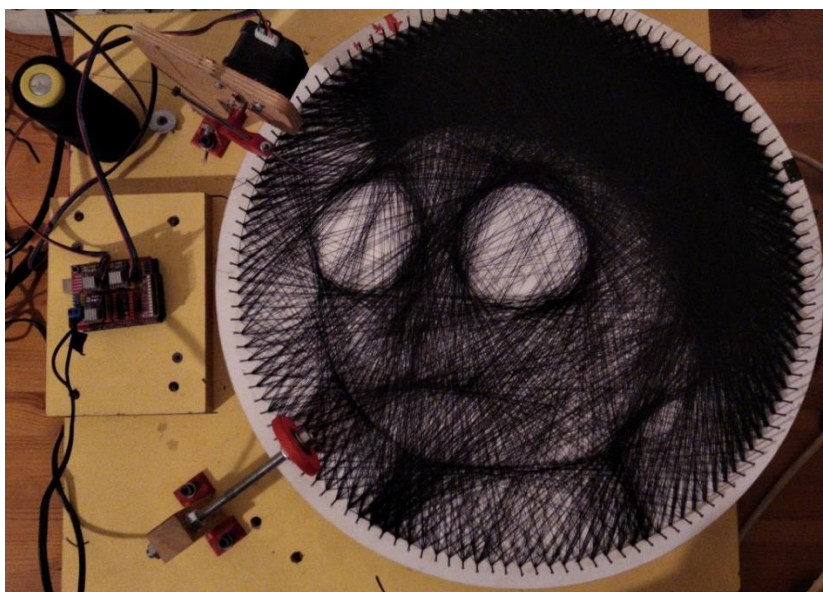
E a szakasz hibacsökkentési értéke, P_i a vizsgált szakasz i . pixelének sötétsége az eredeti képen, F_i a vizsgált szakasz i . pixelének sötétsége a készülő képen, L_i a vizsgált szakasz i . pixelének sötétsége a cénán (Xiaolin Wu vonal)

Ez az algoritmus addig fut, ameddig a legnagyobb hibacsökkentési érték negatív nem lesz, azaz több cérnaszál behúzásával már csak rontana a képen, vagy a behúzott cérnák száma el nem éri a megadott maximális értéket.

2.4. Az irányító program

Ez a program küldi a behúzendó vonalakat a gépeknek. Erre azért volt szükség, mert a vezérlésre használt Arduino UNO magában nem lenne elég erős a gép irányítására, a ki- és bement feldolgozására és egy egész kép parancsainak tárolására, ezen felül így egyszerűbb és olcsóbb a gép, és a kezelőfelület is felhasználóbarátabb.

Egy Python CLIn keresztül vezérli a felhasználó az Arduino-t. Az irányító program és az Arduino egy egyszerű G-code szerű, ember által könnyen olvasható nyelven kommunikál. A rendszer úgy van megtervezve, hogy a vonalzó és az irányító program is ugyanazt a kimenetet adja a gép hardverétől függetlenül.



2. ábra: A gép

2.5. A gép

A gépet 2 motor mozgatja. Az 1. motor forgatja a string artot (forgóasztalt), a 2. motor egy tűt mozgat a szögkörön belülre és kívülre.

A gép mechanikai elemei a költséghatékonyság jegyében olyan anyagokból és alkatrészekből készültek, amelyek éppen kéznél voltak. Az

alapja egy régi bútorlap. A forgó asztal és a túmozdító-motor tartója plexilapból/rétegelt lemezből készült. (a 2 anyag mechanika tulajdonságainak és megmunkálhatóságuknak különbségeinek ennél a projektnél nincs jelentősége) Ezeket én egy MPCNC-vel (nagyraoszt 3D nyomtatott CNC gép) vágtam ki, ami leegyszerűsítette a prototípusok legyártását, de egy dekopírfűrész is elegendő ehhez. Más kisebb elemek 3D nyomtatóval készültek.

A forgóasztalt a léptetőmotor egy bordás-szíjjal, 10:1 áttétel hajtja. A vezérlés egy Arduino UNO-ból, egy léptetőmotor-vezérlő shieldből, 2 DRV8825-ös léptetőmotorvezérlőből áll. Ezek 2 Nema 17-es léptető motort mozgatnak.

2.6. Problémák

Problémát jelentett az, hogy a tű beleütközött a szögekbe a be- és kifordításnál.

1. Miközben a forgóasztal lassult és megállt, a bordásszík megnyúlt, mert a string art falapjának viszonylag nagy a tehetetlenségi nyomatéka. Ezen nem tudtam változtatni, ezért a forgóasztal lassulását csökkentettem.
2. Először a szögeknek kézzel fúrtam elő a lyukakat, ez nem bizonyult elég pontosnak. Erre azt a megoldást találtam, hogy a forgóasztalt a gép forgatja el szögtávolságonként, és így fúrom elő a lyukakat egy, az alaplaphoz erősített fúróvezetővel.

A megoldást arra, hogy hogyan lehet átfűzni a hosszú vékony csövön (2. ábra) a cérnát, rábírom az olvasó fantáziájára. (Egy varrótú megakad benne)

3. Elért eredmények

A projekt fejlesztése alatt sokat tanultam. Megtanultam Arduinot programozni, elfogadható minőségű Rust kódot írni, CAD szoftvert használni, CNC gépet kezelni, és 3D nyomtatni.

A céljaimat nagyrészt elértem, a kód gyors és átlátható lett. A gép szép képeket csinál, és munka közben magára lehet hagyni. Egy képet nagyjából 3 órába telik elkészítenie.

A környezetemben olyanok is érdeklődtek a projekt iránt, akik a műszaki tudományok iránt egyébként nem, ennek nagyon örültem

Az eredeti motivációm ellenére még nem adtam ilyen képet senkinek ajándékba.

A 3 program összesen nagyjából ezer sorból áll, a gép költsége 20.000 Ft alatt van.

Farfler – Önvezető kerekesszék fejlesztése

Polyp Csapat

Vajda Ádám, Bacsur Dániel

Felkészítő: Kováts Livia, Molnár Zsuzsanna

Közgazdasági Politechnikum (Budapest, Vendel u. 3, 1096)

ELTE Radnóti Miklós Gimnázium (Budapest, Cházár András u. 10, 1146)

1. Bevezetés

A Farfler-projekt a mobilitást segítő technológiák átalakítását célozza, új mércét állítva az autonómia és a függetlenség terén a mozgássérült személyek számára. A projektünk Stephan Farfler, egy olyan újtó tiszteletére kapta a nevét, aki a 17. században tervezte meg a kerekesszék egyik korai prototípusát. A Farfler projekt célja, hogy a modern robotika technológiáját felhasználjuk a mozgáskorlátozottak mindennapjainak segítésére.

A megvalósításunk középpontjában a MEYRA Nemo elektromos kerekesszékhez tervezett összekötő modul kifejlesztése áll. Fontos számunkra egy olyan termék elkészítése, ami univerzálisan használható bármilyen másik elektromos kerekesszéken. A modulunk lehetővé teszi a kerekesszék számára, hogy Controller Area Network (CAN) üzeneteket küldjön és fogadjon az RNet rendszeren keresztül, hasonlóan egy intelligens, másodlagos joystick beépítéséhez. Ez a technológia biztosítja, hogy a kézi vezérlés továbbra is elsődleges maradjon, lehetővé téve, hogy a joystick bármilyen mozgása azonnal és biztonságosan kikapcsolja a rendszert, ezáltal a felhasználó szándékát és biztonságát előtérbe helyezve. Ezen kívül az elektromos kerekesszékek automatikus fékezőrendszerét is alkalmazzuk, ami biztosítja, hogy 10 km/h sebességet ne haladja meg a jármű.



1. ábra: MEYRA Nemo kerekesszék

A projekt a Robot Operating System (ROS) 2 Humble Hawksbillt használja, amely a robotikai szoftverfejlesztés legmodernebb keretrendszere. Ez a választás a robusztus, skálázható megoldások iránti elkötelezettségünket tükrözi. A szenzorok, többek között a kettős LIDAR és RGB-kamerák stratégiai elrendezésével 360 fokos képet készítenek a környezetről. Ezzel az információval tudjuk autonóm irányítani az eszközünket. A kézi vezérlés elérése egy Xbox-szerű kontrolleren keresztül jelentős mérföldkövet jelent a teljesen autonóm beltéri navigáció felé vezető útján.

A projekt innovatív megközelítésével nemcsak Stephan Farfler öröksége előtt tisztelgünk, hanem egy olyan jövő felé is utat szeretnénk nyitni, amelyben a mobilitási korlátok jelentősen csökkennek. A kerekesszékeken való alkalmazás mellett a technológia hasonlóan bevethető más segédeszközökön, mint például elektromos hordágyak vezérléséhez.

2. Probléma megoldásának menete

A kezdeti fázis az elektromos kerekesszékekkel való kommunikáció alapvető kihívásaira összpontosított. A legfontosabb kérdések közé tartozott a kerekesszék RNet rendszere és az újonnan kifejlesztett összekapcsoló modul közötti zökkenőmentes kommunikáció biztosítása, a valós idejű érzékelőfúzió megvalósítása, valamint egy olyan intuitív vezérlési mechanizmus megtervezése, amely a biztonság érdekében könnyen felülbírálnak.

2.1. Az összekötő modul tervezése

Műszaki törekvésünk sarokköve az összekapcsoló modul kifejlesztése. A kihívás egy olyan eszköz létrehozása, amely képes utánozni a kerekesszék joystickjének működését, miközben CAN-üzeneteket küld és fogad anélkül, hogy megszakítaná a kerekesszék natív vezérlését. Iteratív tervezéssel és teszteléssel olyan modult fejlesztettünk ki, amely zökkenőmentesen integrálódik az elektromos kerekesszékekbe, biztosítva, hogy a joystickon keresztül történő bármilyen kézi beavatkozás azonnal felülírja az automatizált rendszert, így a felhasználó irányítása és biztonsága élvez prioritást.

2.2. Szenzorfúzió és a környezeti feltérképezése

Az összetett beltéri környezetben történő önálló navigáláshoz a kerekesszéknek robusztus rendszerre volt szüksége a környezetének ismeretéhez. A LIDAR-ok és az RGB-kamerák integrálása jelentős kihívást jelentett, ami kifinomult algoritmusokat igényelt az érzékelők időbeli fúziójához. A ROS 2 képességeit kihasználva olyan csomagot fejlesztettünk ki, amely valós időben dolgozza fel és egyesíti az említett érzékelők adatait,

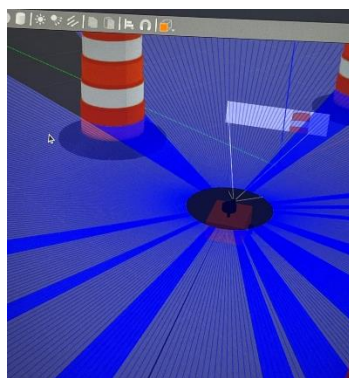
létrehozva egy részletes és dinamikus térképét a környezetről. Ez a térkép kulcsfontosságú az akadályok elkerülése, az útvonaltervezés és az autonóm navigációt megalapozó döntéshozatali folyamatok szempontjából.

2.3. Felhasználóközpontú vezérlési mechanizmusok

Felismerve a potenciális felhasználók sokféleségét, prioritásként kezeltük egy hozzáférhető és intuitív vezérlőrendszer kifejlesztését. Annak biztosítása, hogy ez az irányítási módszer megbízható és könnyen felülbíráható legyen, kiterjedt tesztelést és finomítást igényel. A megoldás egy kettős üzemmódú rendszer, amely lehetővé teszi a felhasználók számára, hogy könnyedén váltsanak a kézi vezérlés és az autonóm navigáció között, olyan biztonsági mechanizmusokkal, amelyek kézi beavatkozásra azonnal kikapcsolják az automatizálást.

2.4. Autonóm navigáció fejlesztése

A teljesen autonóm navigáció kifejlesztése számos technikai akadályt jelent, az összetett térbeli összefüggések megértésétől a szenzoradatokon alapuló valós idejű döntések meghozataláig. Projektünkkel először a beltéri, majd későbbiekben a kültéri navigáció megvalósítását célozzuk. Megközelítésünk a fejlett gépi tanulási technikákat alkalmaz a környezeti adatok pontos értelmezéséhez. Ez a folyamatos fejlesztési szakasz szigorú szimulációs és valós körülmények között végzett teszteléseket foglal magában.



2. ábra: Tesztelés szimulált környezetben

A Farfler-projekt problémamegoldási folyamata a műszaki innováció, a felhasználóközpontú tervezés és a biztonsági prioritások ötvözetét példázza. Az iteratív kihívások és megoldások sorozatán keresztül a projekt egy olyan jövő alapjait fekteti le, amelyben a mobilitási kihívások többé nem jelentenek akadályt a függetlenség és az életminőség szempontjából.

3. Elért eredmények

A Farfler-projekt jelentős előrelépést jelent a segédtechnológia területén, különösen a mozgássérült személyek mobilitásának javítása érdekében.

3.1. Hozzáférés az RNet rendszerhez

Projektünk kulcsfontosságú eredménye az összekapcsoló modul sikeres kifejlesztése és integrálása, esetünkben a MEYRA Nemo elektromos kerekesszékek RNet rendszerébe. Ez az áttörés lehetővé teszi a CAN-üzenetek zökkenőmentes küldését és fogadását, lehetővé téve a kerekesszékek fejlett robotikával történő vezérlését biztosítva, hogy a felhasználó a joystick segítségével bármikor manuálisan felülbírálhassa a rendszert. Ez a kettős vezérlési funkció a legmagasabb szintű biztonságot és felhasználói bizalmat eredményezi.

3.2. Teljesen autonóm beltéri navigáció

A teljesen autonóm beltéri navigáció folyamatos fejlesztése a Farfler-projekt talán legambiciózusabb és legátfogóbb aspektusa. A kifinomult útkereső algoritmusok és gépi tanulási technikák felhasználásával a teljes beltéri autonómia elérésének küszöbén állunk az elektromos kerekesszékek esetében. Ez az eredmény azt ígéri, hogy újradefiniálja a kerekesszékek használók mobilitását, és példátlan szintű függetlenséget és mozgásszabadságot biztosít a beltéri környezetben.

3.3. A mozgáskorlátozott személyek helyzetének javítása

A technológiai és tudományos eredményeken túl a Farfler-projekt igazi sikere a mozgáskorlátozottakra gyakorolt hatásában rejlik. A projekt hozzájárul a mozgáskorlátozottak társadalmi aktivitásának kiterjesztéséhez és teljes értékű életéhez.

Komposztkazán automatizálása és szabályozása

Team Komposzt

Preisler András, Szigeti Szabolcs

Felkészítők: Gőgh Zsolt, Merényi Miklós

BMSZC Petrik Lajos Két Tanítási Nyelvű Technikum

Budapest, Thököly út 48-54, 1146

1. Bevezetés

A mostanában divatot élő komposztálás egy a természetben is végbemenő folyamat, ami nem csak éltető humuszt ad, de akár más, jól felhasználható lehetőség alapját is nyújthatja. Az érlelődő komposzt a hozzávaló nyersanyagok biológiai lebomlása során hatalmas hőt bocsájt ki, amely hő szabályozott körülmények között fűtési célokra is felhasználható.

Immár második éve fejlesztjük elképzelésünket. Célunk egy mobil, telepíthető, érzékelőkkel és szabályozással, közüzemi csatlakozási lehetőséggel rendelkező, konténer típusú egység létrehozása. A tervezett üzleti stratégia velejárója lenne az a szolgáltatás is, amely biztosítja a fent leírt „blokk” stabil összetételű komposztkeverékekkel történő ellátását.

2. A probléma megoldásának menete

2.1. A megépített kazán prototípus



1. ábra: Komposztkazán



2. ábra: A kazán irányító egysége

A konténert mintázó vázszerkezetben henger alakú kazánteret, ettől elkülönítve pedig egy vizes elosztókat és szabályozó elemeket tartalmazó egységet alakítottunk ki. A henger belsejében megépült a két, párhuzamosan, spirálisan feltekert hőcserélő egység. Egyharmad-kétharmad arányban osztja meg a közel 100 l-es, teljes vízmennyiséget. A melegvíz kivétele külön-külön, illetve egyesítve is megtörténhet. A hőcserélő 32mm átmérőjű KPE víznyomó csőből készült. Anyagi kereteink függvényében egy következő típusba már egy speciális formájú, korrózióálló fém hőcserélőt kívánunk beépíteni. Az előállított melegvizet IBO OHI 25-40/180 háromfokozatú keringető szivattyú továbbítja, amelynek maximális teljesítménye 4,5 m³/h.

A fűtési „vízkör” kialakítása is megtörtént, hőleadó egységként két nagyobb radiátor fűti a szomszédos, cc.200 légköbméteres tantermet.

A komposztot befogadó hengertest méretét megnöveltük, a korábbihoz képest 8-ról 13m³ fűtőanyag betöltésére tettük alkalmassá. Egy teljes fűtési szezont (5 hónap) kiszolgáló alaphelyzet kialakítása volt a célunk. A homogenizált komposztkeverék kétharmad részben vegyes faaprítékból, egyharmad részben friss szalmás lótrágyából áll, amelybe kis mértékben nehezebb lebomlású marhatrágyát is kevertünk.

A megfelelő biológiai lebomlás egyik fontos tényezője az érlelődő komposztanyag nedvesen tartása (40-60 % között) A permetező locsolásához klórtól kiszellőztetett csapvizet, és a konténer tetőfelületéről összegyűjtött esővizet használunk, ennek tárolására egy 300 l-es tartály szolgál. A szivárgató rendszert egy bűvárszivattyú működteti. Az optimális lebomlás a megfelelő oxigén ellátottságon is múlik. Ennek elérése érdekében három szintben, csillag alakban, 10 cm átmérőjű dréncső elemeket építettünk a komposztanyagba.

A kazántest külső határrétege természetes szigetelőként is értelmezhető. A spirális hőcserélőket tekintve alul, felül és oldalt 50cm-es szigetelő rétegben gondolkodtunk, ezt alakítottuk ki. A tetőfólia a nagymennyiségű eső kiszámíthatatlan lehűtő hatását akadályozza meg, ugyanakkor magassági beállítása sajátos, párás mikroklímát alakít ki a komposzttest fölött.

2.2. A szenzorhálózat és a szabályozás kialakítása

Az optimális biológiai lebomlási folyamat követéséhez a „boglyába” páratartalom érzékelőket, továbbá termoszenzorokat volt szükséges telepítenünk. Utóbbiakat a kísérletben nagyszámban építettünk be, ezáltal térbeli hőeloszlás képet tudtunk előállítani. A spirális hőcserélő ki- és belépő ágába telepített hőérzékelőkkel pedig a keringtetett melegvíz állapotáról kapunk megfelelő tájékoztatást. A folyamatokat egy külső levegőhőmérő segítségével is követni tudjuk.

A folyamat során a beépített szenzorok segítségével állandóan mérjük és optimalizáljuk a paramétereket (pl.: maghőmérsékletet a komposzt belsejében, vízhőmérsékletet, hőleadási teljesítményt).). A keringető- és öntöző szivattyúkat a szenzoroktól begyűjtött adataink alapján egy Arduino (ESP) mikrovezérlő irányítja.

Amikor az ESP kapja a jelet a szervertől, akkor ő visszaküld egy JSON objektumot, amit utána a mobil alkalmazás lekezel és megjeleníti.

A folyamat során nemcsak hőt kapunk, hanem felhasználható, érett humuszt is. A létrehozott kazán újra tölthető, így teljes mértékben hulladéktmentes eszköznek tekinthető.

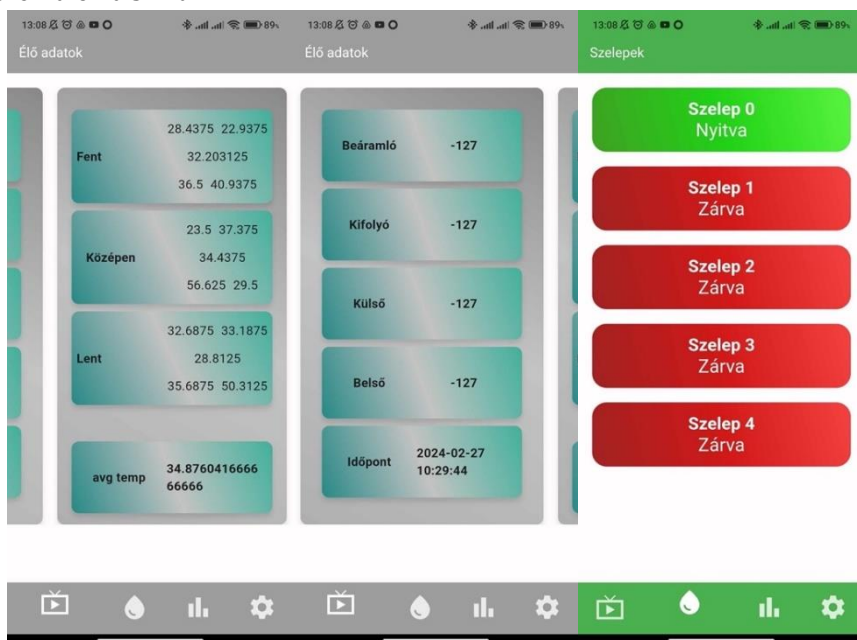
A növekvő energia igények mellett nyitottnak kell lennünk bármilyen irányú energiatermelési formára. A jelenleg hulladékhőként ismert komposzthő hasznosításával szeretnék ehhez a kutatáshoz mi is hozzájárulni.

A projektünk során egy innovatív automatizált komposztkazánt fejlesztettünk ki, amely az Arduino mikrovezérlő egység és egy dedikált szerver összehangolt működésével gyűjti és elemez a komposztálási folyamat során keletkező adatokat. Az eredményeket egy általunk fejlesztett mobilalkalmazáson keresztül teszi elérhetővé a felhasználók számára. A projektünkkel elért eredményeket az alábbiakban soroljuk fel, kiemelve az előnyöket és a hasznosságot.

3. Eredmények

3.1. Újszerűség és Eredetiség:

A projektünk egyedülálló ötvözete a komposztálási folyamat automatizálásának és a modern IoT (Internet of Things) technológiáknak. Az Arduino mikrovezérlő használatával először készítettünk olyan komposztkazánt, amely képes önállóan monitorozni és szabályozni a komposztálás kritikus paramétereit, mint a hőmérséklet, a páratartalom és a szén-dioxid szint.



3. ábra: Az applikáció V10 verziója

3.2. Megoldott Probléma Fontossága:

A komposztálás folyamata kulcsfontosságú a környezetbarát hulladékkezelésben és a fenntartható mezőgazdaságban. Projektünkkel automatizált megoldást nyújtunk a komposztálás optimalizálására, lehetővé téve a felhasználók számára, hogy hatékonyabban és környezettudatosabban kezeljék a szerves hulladékot.

3.3. Szakmai Minőség és Igényesség:

A projekt megvalósításához magas szintű mérnöki és programozási ismereteket alkalmaztunk. Az Arduino és a szerver közötti kommunikáció biztonságos és megbízható, a gyűjtött adatokat pedig valós időben dolgozzuk fel és jelenítjük meg a mobilalkalmazásban. A komposztkazán tervezése során figyelembe vettük a legjobb gyakorlatokat és a fenntarthatósági szempontokat, biztosítva a készülék hosszú távú használhatóságát.

3.4. Bemutató Logikája és Világossága:

A projekt eredményeit egy felhasználóbarát mobilalkalmazás formájában prezentáljuk, amely lehetővé teszi a felhasználók számára, hogy egyszerűen nyomon követhessék és optimalizálhassák komposztálási folyamataikat. Az alkalmazás intuitív felületet biztosít, amely részletes információkat nyújt a komposzt állapotáról, valamint javaslatokat tesz annak javítására.

Összefoglalva, projektünkkel egy olyan rendszert hoztunk létre, amely egyszerű megoldást nyújt a komposztálás automatizálására, elősegítve ezzel a környezettudatos hulladékkezelést és hozzájárulva a fenntartható mezőgazdasági gyakorlatok elterjedéséhez. A fejlesztés során különös figyelmet fordítottunk a rendszer szakmai minőségére és igényességére, valamint arra, hogy az eredményeket világos és logikus módon mutassuk be.

HardTFM - Egy modern Total Fluid Management rendszer az Ipar 4.0-hoz

Team HardTFM

Paróczai Olivér

Felkészítő: Gajda Judit, Tusjak Brigitta

*Békéscsabai SZC Nemes Tihamér Technikum és Kollégium, 5600 Békéscsaba,
Kazinczy u. 7.*

1. Bevezetés

A TFM, azaz Total Fluid Management, egy módszer, amely az iparban használt kenőanyagok kezelésének optimalizálására használandó, ezáltal javítva a gépek teljesítményét, miközben csökkenti az operációs költségeket.

A TFM módszeren belül sok folyamat található:

- A fémmegmunkáló gépekben található emulzió megfelelő koncentrációjának fenntartása időnkénti mérések és ütemezett utántöltések, cserék segítségével
- Az egyéb kenőanyagok szinten tartása, ellenőrzése, időszakos cseréje
- Vállalattól függően a kenőanyag-kezelési folyamat részleges vagy teljes automatizálása, és a nem automatizált folyamatok automatikus leosztása a megfelelő dolgozókra

Piackutatást végeztem az európai iparban használt TFM rendszerek terén, és jelenleg a TFM rendszerek jelentős része a 2000-es évek elején készült. Ezért általában nehezen átlátható felületekkel rendelkeznek. Sok rendszernél pedig teljesen logikátlanul van kialakítva a felhasználói UX, köszönhetően az időnkénti funkciófrissítéseknek, amikkel a régi, legacy rendszereket próbálták modernizálni.

Az Ipar 4.0 megjelenésével egyre nagyobb lett a szükség az eddig használt TFM rendszerek újragondolásra. A meglévő rendszerek nem voltak képesek real-time analitikai készülékekkel való integrációra, és az adataik megfelelő reprezentálására, ami szükséges az új ipari forradalomnál.

A HardTFM egy megoldás ezekre a problémákra, egy központi rendszer, ami szinte az összes iparban fellelhető hálózatképes eszköztípus felületéről elérhető a többretegű kompatibilitási rendszerének köszönhetően, és sok testreszabási és módosítási lehetőséget nyújt a vállalatok számára, hogy a felületet valóban a sajátjukká tehessék.

Megmaradnak a manuális adatbeviteli lehetőségek is, csak egy jóval felhasználócentrikusabb módon, mint a jelenlegi rendszereknél. Így a vállalatoknak szabad döntése, hogy integrálva szeretnék használni a meglévő IoT rendszerükbe hibrid módon, vagy a megszokott módszerrel szeretnék ezeket a folyamatokat manuálisan végezni.

2. Probléma megoldásának menete

2.1. A fejlesztés kezdete

Elkezdtem felkutatni a jelenlegi rendszereket és azok hátrányait vállalati látogatások révén, és elkezdtem megtervezni az új rendszer felépítését. Minden vállalati beszélgetés egy lépéssel közelebb vitt a végeredményhez. Miután megtudtam hogy milyen igényeknek kell eleget tennem az új TFM rendszerrel, elkezdtem a fejlesztési folyamatot, amit az AdminBSB keretrendszerben végeztem. A keretrendszert a vele való eddigi jó tapasztalataim miatt választottam.

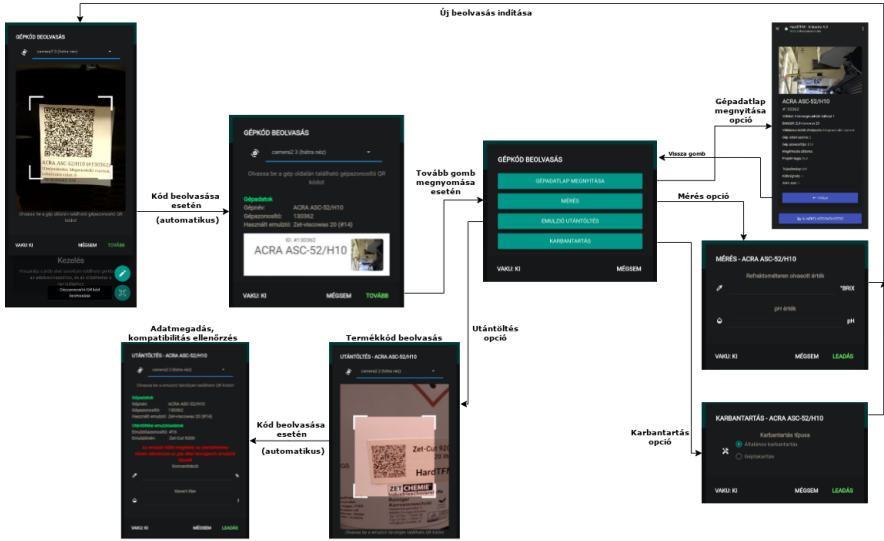
A rendszer PHP, MySQL, JavaScript használatával lett kifejlesztve, a megbízhatóságuk és az hosszú távú támogatásuk miatt. A kliensoldali felülethez egyéni HTML és CSS kódok lettek készítve, és ez a felület progresszív webalkalmazás (PWA) módon is képes futni a kliens eszközén.

A cél a folyamatok egyszerűsítése és felgyorsítása volt, ezért a gépbeolvasási folyamat a gépek oldalára helyezett QR kódok segítségével történik. A dolgozó a főoldalon indíthat egy olvasási folyamatot, és ahogy a QR kód beolvasásra kerül megjelennek a lehetséges opciók, mint például az gépadatlap megnyitása, kenőanyag utántöltés és a karbantartás. Ez a folyamat az 1. ábrán látható.

Ahogy az első prototípus elkészült, felvettem a kapcsolatot néhány helyi vállalkozással azzal kapcsolatban, hogy tudnák-e tesztelni és véleményezni a rendszert.

Az első tesztelési fázis jó eredményekkel zárult, a dolgozók jobban szerették használni a rendszert az előző rendszerekhez képest, és drasztikusan csökkent a felhasználói hibák száma, amivel a dolgozók időt spóroltak meg. A kenőanyag-fogyasztás is csökkent a megfelelő koncentráció fenntartása révén. A dolgozók pozitívan fogadták az új felületet és annak felépítését, segítség nélkül voltak képesek használni az összes funkcióját.

HardTFM Felhasználói Munkafolyamat - Mobileszköz



17. ábra: HardTFM Felhasználói Munkafolyamat

2.2. A vállalatok visszajelzése alapján történő fejlesztések, funkcióbővítések

Az első tesztfázison belüli vállalatok közül többen is visszajeleztek, hogy jó volna egy raktárkezelési modul is a rendszerben, mivel a többi TFM szoftverből ez hiányzik és nagyban megkönnyítené a dolgozók munkáját. Ezek alapján létrehoztam egy modern raktárkezelési felületet a rendszerhez, amin belül egyszerűen lehet hozzáadni új termékeket az rajtuk található QR kódok és/vagy vonalkódok alapján. A rendszer könnyen bővíthetőre lett kifejlesztve, így bármely gyártó termékeivel képes működni miután azok azonosítói hozzá lettek adva a rendszerhez. Miután a termék hozzá lett adva a vállalat raktárkészletéhez és a dolgozó el szeretné kezdeni használni, egyszerűen „megbonthatja” a terméket a raktárkezelési menün belülről, ami után minden utántöltés automatikusan levonásra kerül a megbontott termékből. Ezáltal egyszerűen követhető a vállalat jelenlegi raktárkészlete.

Elkezdődött az automatizálási modul fejlesztése is, amivel különböző PLC-eket, mikrovezérlőket és egyéb IoT eszközöket lehet a rendszerbe integrálni, automatikus döntéshozatalhoz és adatbevitelhez. Az első hardveres prototípus egy Raspberry Pi alapú egység lett ami képes volt valós időben figyelni az emulzió koncentrációját egy in-line Brix szenzor segítségével, és az emulziótartályra épített két szintérzékelővel az emulzió minimum és

maximum szintjét. Ezek segítségével egy proporcionális keverőhöz kötve automatizálható a teljes utántöltési folyamat.

2.3. A jelen

Jelenleg a szoftver folyamatos fejlesztés alatt áll, és a vállalatok visszajelzése alapján állandóan új funkciók kerülnek hozzáadásra.

3. Elért eredmények

A szoftver első tesztfázisú prototípusa 1. helyezett lett a Békés Megyei Műszaki Haladásért Alapítvány „Innovációs díj 2022” versenyén, és bejutott a Magyar Innovációs Szövetség „32. Országos Tudományos És Innovációs Olimpia” versenyen a 2. fordulóba, de ezek óta a szoftver drasztikus változásokon esett át. Ezen versenyek idejében még csak az alap mérési funkció volt működőképes, míg azóta számos funkcióval bővült a szoftver.

A versenyeken kívül a helyi vállalatok körében nagyrészt a napi munkafolyamatok részévé vált a szoftver, segítve a dolgozókat a feladatok elvégzésében egyszerűen és megbízhatóan.

4. Végző és belépési adatok

A belépési adatok a következők:

Vállalati adminisztrátor fiók, felhasználónév: szte_cadmin , jelszó: 82sgNoVZc

Felhasználói fiók, felhasználónév: szte_user, jelszó: m2Ws22PTX

Hozzáférés: <https://oliverparoccai.dev/tfm>

Ha bármi egyéb információra lenne szükség, nyugodtan lépjenek velem kapcsolatba a verseny@oliverparoccai.dev e-mail címen.

Nagyon köszönöm a figyelmüket!

Autó diagnosztika

autodino

Tóth Bence Béla

Felkészítő: Bicskei Róbert

Szegedi SZC Déri Miksa Műszaki Technikum, Szeged, Kálvária tér 7, 6724

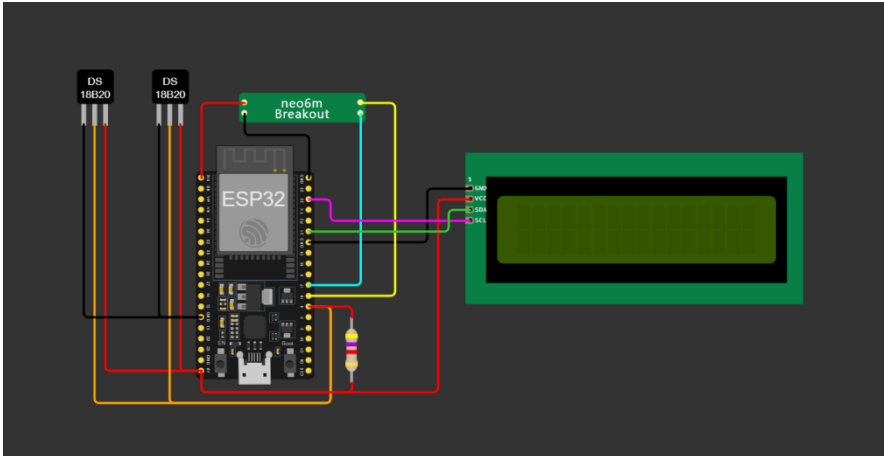
1. Bevezetés

Az autóversenyzés világában nagyon fontos a motorok megfelelő kenése és az üzemi hőfok elérése és megtartása. Ezeket az adatokat folyamatosan figyelik, hogy a motort ne érje károsodás. Az adatok kinyeréséhez drága motorvezérlők kellenek és nagy szaktudás, hogy minden adat pontos legyen. Általában az értékek egy képernyőn vagy egy mérőórán jelennek meg. A képernyő a jármű óracsoportját is felválthatja, ahol a sebességet, fordulatszámot, víz hőfokot, olaj hőfokot és egyéb értékeket jelenít meg. Így a pilóta folyamatosan tud tájékozódni az adatokról. Az én kivitelezésem egy olcsóbb alternatíva a gyári motorvezérlő megkerülésével. Sajnos így nem minden funkció érhető el, de a monitorozásra tökéletes.

2. A probléma megoldásának menete

A célom egy olcsóbb alternatíva az adatok kiírására egy LCD kijelzőre, amit nem csak versenyautókba, hanem egy átlagos autóba is beszerelhetünk. A hőfokok figyelése nem csak a versenyautóknál, hanem az utcai autóknál is fontos. Ha figyeljük a hőmérsékletet (víz, olaj) akkor az autónkat megmenthetjük az esetleges meghibásodásoktól. Az első lépésben ötleteltem a kivitelezésen. Az iskolában tanulunk műszaki programozást, ahol ESP8266 mikrovezérlővel használtunk hőmérséklet szenzorokat a szoba hőmérsékletének figyelésére. Innen jött az ötlet, hogy ezt tökéletesen lehet használni folyadék hőmérséklet figyelésére is, ezért a DS18B20 típusút választottam. Az ESP mikrovezérlő is megtetszett, ezért az ESP32 mikrovezérlőre esett a választásom. Az ESP32 jobb, mint amit az órán használtunk, mert van benne Bluetooth, több port található rajta és még 5 voltos kimenete is van, ami a szenzoroknak kell. A következő megoldandó probléma a sebesség kiírása volt. Az interneten utánajártam, hogy GPS segítségével ezt a problémát meg lehet oldani, ehhez NEO6M GPS modul kellett. Az értékek kiírására egy LCD kijelzőt választottam, amelyre 32 karakter fér rá 2 soron. Ezek az alkatrészek Magyarországon drágák vagy nehezen beszerezhetők, ezért Kínából rendeltem meg őket. A következő lépés az összeállítás volt. Az interneten sok segítő anyagot lehet találni az egyes

alkatrészek bekötéséről, én is ezekből tájékozódtam. A bekötési áramkör az 1. ábrán látható.

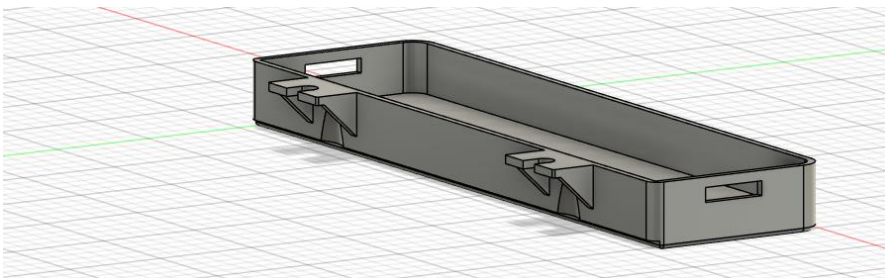


18. ábra: Áramkör

Az alkatrészek bekötése után a programozás következett. A Python volt a megfelelő programozási nyelv a sok beépített kód miatt, továbbá ezt a nyelvet tanuljuk az iskolában is. Az interneten néztem különböző kódokat és a sajátomat ezek segítségével írtam meg.

2.1. Autóba építés

Az autóba úgy szerettem volna beépíteni, hogy egy olyan embernek, aki nem ismeri ezt a típust ne legyen neki feltűnő, hogy utólagosan vannak beépítve az alkatrészek. Ezt úgy tudtam elérni, hogy a műszerfal közepén található elemet újra terveztem a Fusion 360 tervezőprogramban. A 2. ábrán az alkatrész látható tervezés közben.



2. ábra: Műszerfal elem tervezés közben

Nehézséget okozott még a beszerelésben a feszültség különbség. Az autóban a rendszer 12 volton működik az ESP32 pedig 5 volton. Ezt egy L7805CV

feszültség stabilizátorral oldottam meg. A 12 voltot a szivargyújtó kábeleiről szedtem le, mert a szivargyújtó el van látva biztosítókkal ezáltal védve van az esetleges elektromos tüzek kialakulásától. A másik indok pedig az volt, hogy a szivargyújtó kábelein csak akkor van feszültség, ha gyújtás van az autón, ezért csak gyújtáson működik az ESP. Az utolsó nehézségforrást a szenzorok bekötése okozta. Az olajteknőre egy lyukat kellett fúrni és ráhegeszteni egy anyacsavart, amibe beletudtam tekerni az egyik hőmérséklet érzékelőt. A másik hőmérséklet érzékelőnél a tágulási tartályt kellett átalakítani úgy, hogy bele tudjam helyezni.

3. Elért eredmények

A második fejezetben említett dolgok nagy részét én csináltam. Az beépítendő alkatrészek megkeresését vagy tervezését. Az autóba beépítés során hova kerüljenek az alkatrészek és hogy miként legyenek beépítve. Az ESP32 programja, amihez a Thonny alkalmazást használtam. Nem utolsó sorban az autóba beépítést is én csináltam.

Együttműködő partnereink

Együttműködő partnereink



ExxonMobil Hungary Kft.
www.exxonmobil.hu



4iG Nyrt.
www.4ig.hu



EPAM Systems Kft.
www.epam.com



**Morgan Stanley Magyarország
Elemző Kft.**
www.morganstanley.com



3i Fejlesztő és Szolgáltató Kft.
www.3i.hu



Digital Kft.
www.digital.co.hu



TEConcept Hungary Kft.
www.teconcept.hu



Creativ_IT
www.creativit.hu



Accenture

www.accenture.com



CAS Software Kft.

www.cas-software.hu



**SZTE TTIK Informatikai
Intézet**

www.inf.u-szeged.hu



**SZTE Természettudományi és
Informatikai Kar**

www.sci.u-szeged.hu



Kiadta: SZTE TTIK Informatikai Intézet

Készítette: Dr. Németh Gábor

Design: Dr. Németh Gábor

Együttműködő partnereink:



SZTE TTIK
INFORMATIKAI INTÉZET



ExxonMobil



Morgan Stanley

<creativ_it



accenture