

Szegedi Tudományegyetem
Informatikai Intézet

SZAKDOLGOZAT

Kovács-Bodó Csenge

2024

Szegedi Tudományegyetem

Informatikai Intézet

**Beszédfelismerő és képernyő felolvasó játék
fejlesztése vakok és látássérültek számára**

Szakdolgozat

Készítette:

Kovács-Bodó Csenge

Gazdaságinformatikus

BSc hallgató

Témavezető:

Dr. Jánki Zoltán Richárd

Egyetemi adjunktus

Szeged

2024

Feladatkíírás

Az esélyegyenlőség jegyében már számos program elérhető az interneten, melyeket kifejezetten vakok és látássérültek számára készítettek el, emellett nagy népszerűségnek örvend általános felhasználók körében is, ha egy program felhasználói felülete minél könnyebben használható, felhasználóbarát. Szakdolgozatom célja olyan frontend oldali technológiák, módszerek tárgyalása, melyekkel akadálymentesíthetünk egy programot látásukban korlátozott felhasználók számára. Példa programom egy egyszerű memóriajátékok mutat be, melyet hangvezérléssel irányíthatunk, és a program folyamatos szövegfelolvasással visszajelzést küld és segít a tájékozódásban.

Tartalmi összefoglaló

A téma megnevezése :

Beszéd felismerő és képernyő felolvasó játék fejlesztése vakok és látássérültek számára

A megadott feladat megfogalmazása :

Szakdolgozatom célja olyan módszerek és technológiák bemutatása, amelyekkel vakok és látássérültek számára kényelmesen használható, kiváló felhasználói élményt nyújtó játékot készítek.

A megoldási mód :

Az alkalmazás célja, hogy a felhasználók billentyűzet és egér nélkül, akusztikus módszerekkel irányíthassák a játékot, mivel nem támaszkodhatnak a látásukra. Ehhez a Web Speech API beszéd felismerő és beszéd szintetizáló funkcióit használtam, melyeket a program egyes részein felüldefiniáltam.

Alkalmazott eszközök, módszerek :

A demonstrációs projekt a legújabb Angular keretrendszerrel készült. Létrehoztam egy Firebase projektet, amelyben Firestore-t használtam a játék eredményeihez, Storage-t a kártyák képeinek tárolására, és Hostingot a program kitelepítéséhez. A beszéd felismerést és szintetizációt a Web Speech API két interfészével valósítottam meg.

Elért eredmények :

Létrehoztam egy webes memória játékot, amelyhez nem szükséges billentyűzet vagy egér, mert hangparancsokkal irányítható, és beszéd szintetizációval ad visszajelzést. Az egér mozgatásával az alkalmazás felolvassa a szövegeket és elemek nevét. A játék magyar vagy angol nyelven is játszható, testreszabható beállításokkal. Van egyszemélyes és kétszemélyes mód, az egyéni eredmények pedig ranglistára küldhetők.

Kulcsszavak :

beszéd felismerés, beszéd szintetizáció, szöveg felolvasás, Web Speech API

Tartalomjegyzék

Feladatkiírás	3
Tartalmi összefoglaló	4
Bevezetés	7
1. Terület áttekintése, jelenleg ismert technológiák	8
1.1. Apple VoiceOver	8
1.2. Intelligens személyi asszisztens	8
1.3. SciFY	9
1.4. Chrome-ban támogatott bővítmények és csomagok	9
1.5. Referencia alkalmazások	10
2. A program funkcionális specifikációi	11
2.1. Funkcionális elvárások	11
2.2. Használati eset diagram	12
2.3. Szekvenciális diagram	13
3. Használt technológia-A Web Speech API	14
3.1. Gépi beszédfelismerés háttere	14
3.1.1. Alapfogalmak: Az osztályozási feladat	14
3.1.2. A statisztikai módszerek kihívásai	14
3.1.3. A beszédfelismerő rendszer felépítése	15
3.2. A Web Speech API	16
3.2.1. SpeechRecognition	16
3.2.2. SpeechSynthesis	17
4. A program megvalósítása	18
4.1. A program architektúrája	18
4.2. Adatmodellek	20
4.3. Verziókövetés	22
4.4. A játék grafikus interfészének elkészítése, jellemzői	22

4.5. Beszédfelismerés implementálása	23
4.6. Beszédszintetizáció impementálása	23
4.7. Képernyőfelolvasás implementálása	23
4.8. Angol-és magyar nyelvre fordítás implementálása	24
4.9. A játék logikájának implementálása	25
4.10. Egyéb funkciók implementálása	25
5. Az elkészült alkalmazás ismertetése	26
5.1. A szoftver végleges specifikációi	26
5.2. A szoftver hiányosságai, továbbfejlesztési lehetőségek	26
5.3. Összefoglalás	26
Irodalomjegyzék	27
Nyilatkozat	29
Köszönetnyilvánítás	30

Bevezetés

Vakok és látássérült emberek számára elérhető eszközök, játékok, könyvek vagy alkalmazások tervezése és készítése napjaink legnagyobb kihívása, viszont az egyik legszebb feladata. Fontos missziója a társadalmunknak ezen emberek integrálása, befogadása közösségünk közé. A testileg vagy szellemileg sérült emberek gyakran szembesülhetnek azzal, hogy a piacon található alkalmazások nem megfelelő számukra, emiatt kirekesztettnek érezhetik magukat, önbecsülésük csökken, szociális készségeik visszamaradottak. Középiskolában volt egy osztálytársam, aki egy ritka szembetegséggel született, ennek eredményéül maximum 10 százalékot látott a külvilágból. Nehezen nyitott mások felé, nehezen tudott tanulni, lassan írt kézzel viszont gépelni egyáltalán nem tudott.

A technológia rohamszerű fejlődésének köszönhetően már számos olyan könyvtár elérhető, amit a legtöbb magas szintű programozási nyelv is keretrendszer támogat. Az elmúlt évtizedben nagy hangsúlyt fektettek a fejlesztő cégek olyan szoftverek értékesítésébe, melyben különféle akadálymentesítést szolgáló funkciók elérhetőek, és a felhasználói élményt maximalizálták.

Szakdolgozatom célja ezen technológiák, frontend fejlesztési alapelvek, módszerek ismertetése, összegyűjtése, melyekkel vakoknak, látássérültek és színvakoknak is egyaránt tudunk szoftvereket tervezni és fejleszteni. Példaprogramom egy olyan memóriajátékot mutat be, melyben nem szükséges a látásunkra hagyatkozni, illetve billentyűzetre és egérre sem lesz feltétlen szükség, hanem akusztikus módszerekkel tudjuk kezelni az alkalmazást, mégpedig úgy, hogy az a felhasználó szóban kiadott parancsokkal tudja irányítani az egész programot, tehát a játék beszédfelismerésre képes, eközben szövegfelolvasással és beszédszintetizációval segít az oldalon való könnyebb tájékozódásban.

1. fejezet

Terület áttekintése, jelenleg ismert technológiák

Már egyre több új szoftver, létező alkalmazásokban új funkciók elérhetőek a piacon, melyeket kimondottan vakoknak, látássérültek vagy színvakok számára fejlesztettek. Ezek közül bemutatom a legnépszerűbbeket.

1.1. Apple VoiceOver

A VoiceOver [1] egy fejlett képernyőfelolvasó funkció Mac Os X operációs rendszerekben, mely lehetővé teszi látássérült felhasználók számára, hogy könnyedén vezérelhessék telefonjukat, táblagépüket vagy számítógépüket billentyűzetparancsokkal [3] és gesztusokkal. [2] A képernyő tartalmában szakaszosan tud lépegetni a felhasználó, miközben a készülék automatikusan felolvassa a képernyő tartalmát. Mac gépeken a Command-F5 billentyűkombinációval kapcsolható be, Iphone-okon és Ipad-eken a beállításokon belül a kisegítő lehetőségek menüpontban található, vagy Siri-nek kiadott utasítással is bekapcsolható. A Braille kijelzők [4] és a Multi-Touch trackpad [5] megjelenése óta már egyszerű kézi gesztusokkal is vezérelhető a felolvasás. [2]

1.2. Intelligens személyi asszisztens

A VoiceOver-nél említett Siri [6], Samsung eszközökön a Bixby [7], Amazon Alexa és Alice mind-mind olyan intelligens személyi asszisztens program okoseszközökön, mely nagy mértékben egyszerűsíti és gyorsítja eszközeink használatát. Elég egyetlen szóban kiadott utasítás és a program a felhasználó helyett elvégzi az eszközön a feladatot, például hívást indít, rákeres bármire a böngészőben, megnyitja a letöltött applikációkat. Ez a

funkció rendkívül hasznos látássérülteknek, mivel egyáltalán nem kell a kezüket használni okostelefonjuk kezeléséhez.

1.3. SciFY

A SciFY (Science For You) [8] egy görög nemzetközi szervezet, mely ingyenes és nyílt forráskódú LEAP (Listen - LEArn - Play) játékokat fejleszt kifejezetten vakok és gyengénlátó gyermekek számára. A játékok célja, hogy segítsenek a gyerekeknek új készségek elsajátításában, miközben szórakoztatják őket. Hivatalos weboldalukon, a <https://gamesfortheblind.org> oldalon különböző játékokat található, mint például a Tic Tac Toe, Tennis és Curve, amelyek különböző nehézségi szinteken játszhatók. Ezek a játékok háromdimenziós binaurális hanggal rendelkeznek, amelyeket a játékosok teljes mértékben kihasználhatnak. A weboldalon kívül a Memor-i Online platformon bármelyik felhasználó könnyedén létrehozhat saját, inkluzív memória játékot, amelyet akár vak, semleges vagy látó játékosok is játszhatnak. A platform célja, hogy lehetővé tegye a vak és látó gyermekek közötti együtt játszást, és segítse a gyerekek fejlődését

1.4. Chrome-ban támogatott bővítmények és csomagok

Fellehető egy kiváló leírás a Google Codelabs jóvoltából a <https://codelabs.developers.google.com/angular-a11y#0> oldalon, ahol pontokba szedve adnak tippeket, hogyan lehet egy Angular-ban írt felhasználó felület minél jobban akadálymentesített, mire figyeljen egy fejlesztő, mikor színvakoknak tervez alkalmazást. Egy egyszerű képernyőolvasó módszert is bemutat, melynek lényege, hogy a HTML elemeinket felcímkézzünk ARIA label-ökkel, ahol megadjuk, hogy a program mit olvasson fel, majd egy Chrome Web Store-ból letöltjük a megfelelő bővítményt [11], amely felolvassa a label-ök tartalmát.

Példa egy ARIA label-el felcímkézett Angular Material elemre[10]:

```
<mat-slider
  aria-label="Dumpling order quantity slider"
  id="quantity"
  name="quantity"
  color="primary"
  class="quantity-slider"
  [max]="13"
  [min]="1"
  [step]="1"
  [tickInterval]="1"
  thumbLabel
  [(ngModel)]="quantity">
</mat-slider>
```

További akadálymentesítést elősegítő bővítmények: Chrome Web Store-ban [12].

1.5. Referencia alkalmazások

Példaprogramom fejlesztése előtt áttekintettem valamennyi nyílt forráskódú programokat, publikus GitHub repository-kat [13][14][15], melyek beszéd felismeréssel és beszéd szintézisével működnek. A kiválasztott technológia, amivel elkészítettem a játékot, a Web Speech API, mely egy következő fejezetben bemutatásra kerül.

2. fejezet

A program funkcionális specifikációi

2.1. Funkcionális elvárások

A szakdolgozatomhoz elkészített szoftver egy olyan játékot mutat be, amely a Web Speech API SpeechRecognition és SpeechSynthesis részAPI-jait használja, tehát ismerje fel a felhasználó által adott verbális utasításokat, majd dolgozza fel őket a teljes weboldalon, emellett adjon rendszeres visszajelzést szóban a felhasználónak beszédszintézissel.

A weboldal megnyitásakor a felhasználó kiválaszthatja, hogy magyarul, vagy angol nyelven szeretné használni az alkalmazást, és a program eszerint le lesz fordítva.

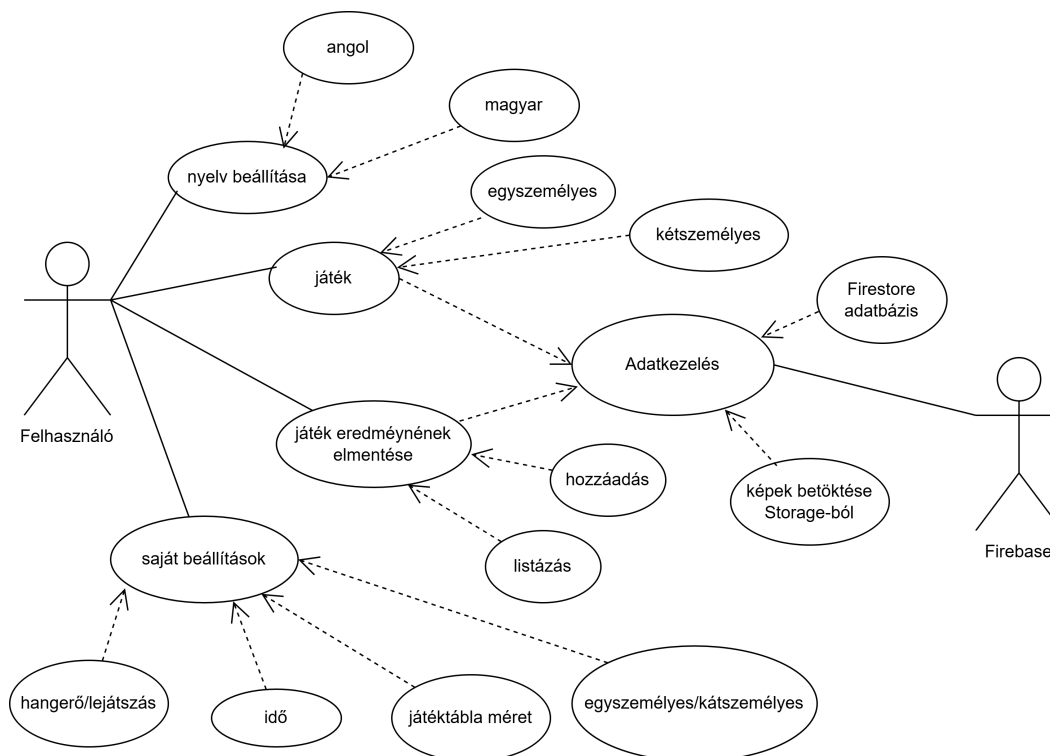
A program egy egyszerű memóriajátékot mutat be, melyet egyedül és párban is lehet játszani. Egyszemélyes játékban pontokat lehet gyűjteni időre. Sikeres játék végén egy becenévvel felkerülhet a játékos egy globális ranglistára.

Elérhető egy beállítások menü, ahol a játékos személyre szabhatja a játék méretét, időtartamát, beállíthatja, hogy egyszemélyes vagy kétszemélyes játékkal szeretne játszani, illetve a program hangerejét és lejátszási sebességét is megváltoztathatja.

A játék mögött áll egy Firebase alkalmazás, mellyel eltároltam a kártyák képeit és a ranglista adatait, illetve kitelepítettem publikusan a Webre a Hosting szolgáltatással.

2.2. Használati eset diagram

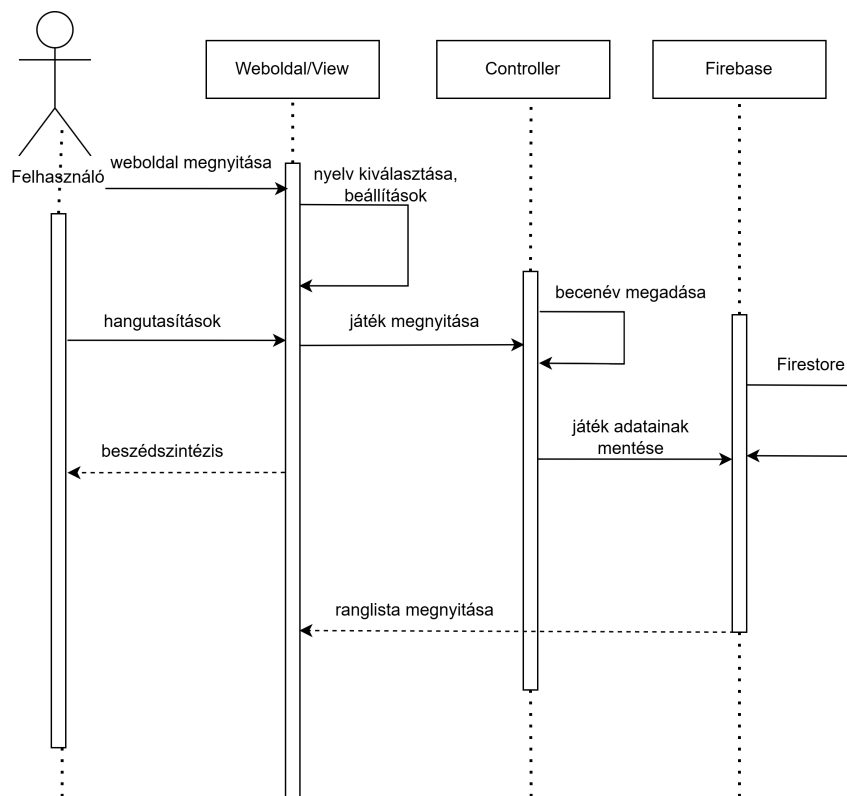
A program használati esetei a nyelv kiválasztása, a játék személyre szabhatósága és használata, illetve az eredmények tárolása Firebase adatbázisban. Ezeket az eseteket mutatja be az alábbi ábra.



2.1. ábra. Use case diagram

2.3. Szekvenciális diagram

Az alábbi ábra bemutatja a program legfontosabb folyamatait, hogy hogyan jut el a felhasználó program megnyitásától a játék eredményeinek tárolásáig. A eljárások során a játékos végig képes hangutasításokat kiadni és a program beszédszintézissel válaszol.



2.2. ábra. Szekvenciális diagram

A használati esete és szekvenciális diagramokat a draw.io programmal készítettem el.[16]

3. fejezet

Használt technológia-A Web Speech API

3.1. Gépi beszédfelismerés háttere

A számítógépes beszédfelismerés [17] egy összetett folyamat, amely az emberi beszéd digitális feldolgozásán alapul, hogy azt a számítógép írott formába alakítsa. Az alábbiakban a legfontosabb elemeket és alapelveket foglalom össze, érthető formában.

3.1.1. Alapfogalmak: Az osztályozási feladat

A beszédfelismerés olyan osztályozási probléma, ahol a cél az, hogy adott hangokról eldöntsük, melyik fonémához (beszédhanghoz) tartoznak. Ehhez a hangokat számítógéppel feldolgozva, jellemzőkkel (adatokkal) írjuk le. Az osztályozás Bayes-döntésmélet alapján történik, amely azt mondja ki, hogy az a döntés optimális, amelyik a legvalószínűbb kimenetet választja az adatok alapján.

3.1.2. A statisztikai módszerek kihívásai

A statisztikai alapú megközelítés során gyakran két fő probléma merül fel:

- Adatkorlátok: Kevés példa esetén a statisztikák pontatlanok lehetnek.
- Dimenziós problémák: Sok jellemző és lehetséges kombináció mellett az adatok feldolgozása exponenciálisan nehezebbé válik.

Lehetséges megoldások:

- Több adat gyűjtése.
- A jellemzők számának csökkentése, azaz kizárólag a releváns adatok használata.
- A valószínűségek egyszerűsítése (például az egyes jellemzők függetlenségét feltételezve).

3.1.3. A beszédfelismerő rendszer felépítése

A tipikus beszédfelismerő rendszer több modulból áll:

a) Előfeldolgozás:

A beszédet digitális jellé alakítják, amelyet spektrogram formájában elemeznek. Ezt követően csökkentik az adatmennyiséget, például a kevésbé fontos részek eltávolításával.

b) Akusztikai-fonetikai modell:

Ez a rész a hang és a beszédhangok (fonémák) közötti kapcsolatot térképezi fel. Ehhez gyakran rejtett Markov modelleket (HMM-eket) használnak, amelyek a beszédet kisebb egységekre bontva értékelik.

c) Nyelvi modell:

Ez a modul a lehetséges beszédkombinációk valószínűségét határozza meg. Például az angol nyelvben a szavak előfordulási valószínűségét, míg magyarban a ragozási szabályokat figyelembe véve működik.

d) Dekóder:

Ez az egység összevonja az előző modellek eredményeit, hogy az egész beszédet szöveggé alakítsa. A folyamat során számos hipotézist vizsgál meg, és a legvalószínűbbet választja ki.

Tehát megállapítható, hogy a számítógépes beszédfelismerés lényege, hogy az emberi beszédet digitális jelekké alakítja, és statisztikai, valamint gépi tanulási módszerek segítségével fonémákra, majd írott szövegre fordítja. A módszerek fejlesztésének kulcsa az adatok pontos feldolgozása és a modellek folyamatos finomítása.

3.2. A Web Speech API

A Web Speech API [18] egy böngésző alapú API, amely lehetővé teszi beszéd alapú alkalmazások létrehozását webfejlesztők számára. Ez az API különösen hasznos a látássérült felhasználók számára, vagy azoknak az alkalmazásoknak, amelyek interaktívabb élményt szeretnének nyújtani. Két fő eleme:

- SpeechRecognition:
 - lehetővé teszi a weboldalak számára, hogy felismerjék és feldolgozzák a felhasználó beszédét. Ez hasznos lehet például hangalapú keresésekhez vagy hangvezérelt feladatokhoz.
- SpeechSynthesis:
 - lehetővé teszi, hogy a program beszéljen egy adott nyelven. Például, ha egy weboldal szeretne szöveget felolvasni, a SpeechSynthesis segítségével ezt megteheti.

Használata kizárólag JavaScript-ben biztosított, ezen két funkció a SpeechSynthesisUtterance és a SpeechRecognition objektumok használatával érhető el.[19]

3.2.1. SpeechRecognition

A SpeechRecognition [20] API a Web Speech API része, amely segítségével webalkalmazások képesek felismerni és feldolgozni a felhasználók beszédét. Az API lehetővé teszi a beszédfelismerési folyamat indítását, amely során a mikrofonon keresztül érkező hangokat elemzi és szöveggé alakítja. Számos eseményt kínál, mint például onresult, onspeechstart, onspeechend, amelyekkel a fejlesztők különböző eseményeket kezelhetnek a felismerési folyamat során. Támogatja a folyamatos felismerést is, ahol a beszédfelismerés nem áll le az első mondat után, hanem folytatódik, amíg a felhasználó beszél.

Támogatottsága böngészőnként eltérő lehet, jelenleg Safari-ban még nem támogatott. Emellett biztonsági kockázatokkal is járhat a mikrofon engedélyezése. Bármely program inicializálásakor a felhasználó dolga mikrofonjának engedélyezése a böngészőnek, ennek hiányában nem fog működni a beszédfelismerés. Példa kód:


```
var recognition =  
new (window.SpeechRecognition || window.webkitSpeechRecognition)();  
recognition.lang = 'hu-HU'; recognition.continuous = true;  
recognition.interimResults = false;  
recognition.onresult = function(event) {  
    var transcript = event.results[0][0].transcript;  
    console.log(transcript);  
};  
recognition.start();
```

3.2.2. SpeechSynthesis

A SpeechSynthesis [21] API a Web Speech API része, amely lehetővé teszi, hogy egy weboldal beszédfelismerést és beszédszintetizációt valósítson meg, ezzel akár saját képernyőolvasó funkciót is implementálhatunk. Az API szöveget alakít hanggá és lejátsza felhasználóknak. Személyre szabhatjuk a szintetizált beszéd tulajdonságait, mint nyelvét, a beszéd sebességét, hangmagasságát és hangerejét. A SpeechSynthesisUtterance objektum tartalmazza a szintetizált beszéd szövegét és beállításait és a SpeechSynthesis objektum vezérli a beszédszintetizálási szolgáltatást, például elindítja vagy leállítja a beszédet. Legfontosabb függvénye a speak(), mely két paramétert vár: a kimondandó szöveg és a nyelv.

Példa kód:

```
var msg = new SpeechSynthesisUtterance();  
msg.text = "Heló Világ!";  
//msg.lang = "hu-HU" // Nyelv  
msg.volume = 1; // Hangerő  
msg.rate = 1; // Lejátszási sebesség  
msg.pitch = 1; // Hangmagasság  
window.speechSynthesis.speak(msg, "hu-HU");
```

4. fejezet

A program megvalósítása

Ebben a fejezetben bemutatom a szakdolgozat példaprogramjának megvalósítását, a tervezéstől a kivitelezésig.

4.1. A program architektúrája

Szakdolgozati szoftverként egy Angular CLI alkalmazást készítettem el. A jelenleg legfrissebb, 17.3.0-s verzióju Angularral fejlesztettem, 20.12.1-es Node.js-sel és 10.5.0-s verzióju Node Package Managerrel. A program teljes logikáját a komponensek TypeScript file-jaiban írtam meg, néhány service-szel kiegészítve, nem kötöttem hozzá másik backend programot.

Az alkalmazás számos komponensből épül fel, melyek a következők:

- AppComponent: a program magja, ebből indul ki a többi komponens, kivéve a SetlanguageComponent, és adjuk át a service-eknek az alkalmazás nyelvét.
- HomepageComponent: a főoldal komponense.
- HelperComponent: egy segítség dialógus, amely felolvassa az összes olyan utasítást, amelyeket a felhasználó ki tud adni, hogy irányíthassa a programot.
- GameComponent: egyszemélyes játék komponense. Egy egyszerű memóriajátékkal tudunk játszani, mely méri az időnket és gyűjti a pontokat.
- SuccessComponent: egy dialógus ablak, amely akkor jelenik meg, ha megtaláltuk az idő letelte előtt az összes párt. Itt megadhatjuk a nevet amivel szeretnék szerepelni a ranglistán és elküldjük az eredményeket a Firestore-ba.
- RetryComponent: egy dialógus ablak, amely akkor jelenik meg, ha nem találtuk meg az összes párt. Innen új játékot lehet kezdeni.

- RankingsComponent: a korábbi eredményeket listázza ki a Firestore-ból rangsorolva pont- és idő alapján.
- SetlanguageComponent: a weboldal megnyitásakor mindig ez a komponens jelenik meg. Itt állítjuk be a szoftver nyelvét.
- SettingsComponent: a beállítások menü komponense, mely az alábbi almenüket tartalmazza:
 - BoardsizeComponent: játékméret beállítása. Háromszor négyes, Négyes négyes és négyszer ötös méretek elérhetőek.
 - NumberofplayersComponent: játékosok számának beállítása. Egyszemélyes és kétszemélyes játék közül lehet választani.
 - TimeComponent: játékidő beállítása. 60, 120, 180 vagy 300 másodperc állítható be.
 - VolumeandspeedComponent: itt a hangerő és a lejátszási sebesség állítható be.
- MultiplayerComponent: kétszemélyes játék komponense. Nem méri az időt és a pontokat, kizárólag a két játékos talált párjainak a számát.
- MultiplayersuccessComponent: ez a dialógus akkor jelenik meg, ha megtalálták az összes párt. Kiírja, illetve felmodja a nyertes nevét, vagy ha döntetlen lett az állás.

Valamint az alábbi service-ekből:

- CardService: kártyák betöltése Storage-ból
- GameService: segít a játék és a beállítás menü között a játék tulajdonságainak a kezelésében.
- LanguageService: kommunikál a SetlanguageComponent és a többi komponens között, beállítja nyelvet.
- SpeechRecognizerService: a beszéd felismerésért felelős service.
- SpeechSynthesizerService: a beszéd szintézisért felelős service.
- VoiceoverService: a képernyőolvasást megvalósító service.

Mivel a böngésző az operációs nyelv beállított, illetve letöltött nyelvi csomagjait tudja használni a Web Speech API használatához, így nagyon fontos, hogy az alkalmazás használata előtt azokat a nyelveket, melyeket használni szeretnénk a programban, az operációs rendszer nyelvi beállításában töltsük le a megfelelő nyelvi csomagokat. Alapértelmezetten támogatott minden operációs rendszerben az angol nyelv, viszont a magyar nyelv nem, ezért különösen fontos a felhasználóknak erről gondoskodniuk.

4.2. Adatmodellek

Három fő adattal dolgoztam az alkalmazás fejlesztése során. A program elején beállítottam a nyelvet, melyet egy string-ként átadtam a beszédfelismerést és a beszédszintézésért felelős service-eknek. Amennyiben magyar nyelvet választ ki a játékos, a "hu-HU", angol nyelv esetén a "en-US" string kerül átadásra. A játékban használt kártyákat nem a program repository-jában tároltam el, hanem az alkalmazás mögött álló Firebase Storage-ban. A képeket a Storage-ból az elérési útvonaljukkal (string) és egyedi Access token-jük (string) segítségével értem el. A képek betöltéséhez készítettem egy service-t CardService néven.

Kártyák betöltése Storage-ból:

```
getImage(imageUrl: string) {  
    return this.storage.ref(imageUrl).getDownloadURL()  
}
```

Ez a metódus a játék során betölti azt annak kártyának a képét, melyet kiválasztott a játékos. Paraméterben megkapja a kép elérési útvonalat (esetünkben a cards/ mappából), a this.storage.ref(imageUrl) segítségével létrehoz egy hivatkozást a Firebase Storage-ban található fájlhoz, majd a getDownloadURL() metódust hívja meg a hivatkozáson, amely egy Promise-t ad vissza, mely tartalmazza a képeket. A kártyák elérési útvonalát egy

konstansban tároltam el.

```
export const cardpictures =[
  "cards/cat.png",
  "cards/ball.png",
  "cards/candy.png",
  "cards/car.png",
  "cards/cloud.png",
  "cards/dog.png",
  "cards/flow.png",
  "cards/rose.png",
  "cards/sun.png",
  "cards/umb.png"
]
```

Végezetül a játék során eltároljuk az eredményeket a ranglistába. Az adatokat a Firebase app Firestore-jába mentjük el, a rankings kollekcióba:

- `documentId(string)`: egyedi azonosítója a Firestore dokumentumnak, amely egy rangsori helyezés adatait tárolja
- `name(string)`: felhasználónév(felhasználói inputról)
- `points(number)`: a játék során elért pont
- `time(number)`: a játékban eltöltött másodpercek száma
- `date(timestamp)`: a játék adatainak mentési ideje(mm dd, yy at hh:mm:ss AM/PM timezon formátumban)

Az adatok kezeléséhez készítettem egy TypeScript interfészt is.

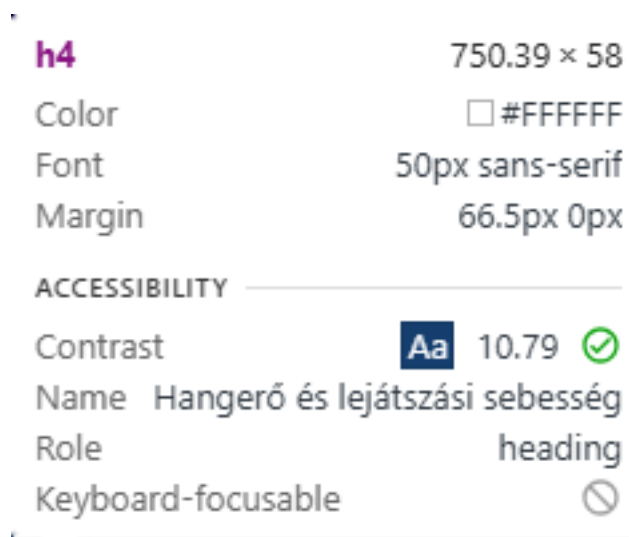
```
export interface Ranking{
  name:string;
  points: number;
  time: number;
  date: any;
}
```

4.3. Verziókövetés

A fejlesztési folyamat során a programot a kezdetektől a befejezésig verziókövetéssel kezeltem a GitHub platformon, biztosítva a kód folyamatos nyomon követhetőségét és dokumentáltságát. A végleges alkalmazást a Firebase Hosting szolgáltatás segítségével deploy-oltam, amely stabil és megbízható környezetet biztosított a projekt publikálásához. A `https://blindmemorygame-szakdolgozat.web.app/` címen publikusan elérhető bármely böngészőben az alkalmazás.

4.4. A játék grafikus interfészének elkészítése, jellemzői

A szoftver felhasználó felületének elkészítése során különös figyelmet szenteltem arra, hogy színvakok számára megfelelő legyen. Figyelmet fordítottam a színekre, nem használtam olyan kombinációk, melyeket színtévesztők könnyen összekeverhetnek, mint például a piros és zöld együttese. Egy sötétkék háttért választottam, és a szövegeket fehérre színeztem, hogy minél kontrasztosabb legyen. Átlagosan 60 pixel nagyságú betűméretet sans-serif betűtípust adtam meg a css file-okban, amely egy talpatlan betűtípus, ezáltal könnyebben olvasható. Minden gombot, toggle csúszkát, görgetős elemet és ikont a lehető legnagyobb méretűre állítottam. Az elért eredményem pedig, hogy a Developer Tools minden elememre pipát írt a Contrast értékre, amely azt mutatja, hogy könnyen olvasható, akadálymentesített a felület.



4.1. ábra. Vizsgálat Developer Tools-ban egy tetszőleges elemre.

4.5. Beszédfelismerés implementálása

4.6. Beszédszintetizáció implementálása

4.7. Képernyőolvasás implementálása

4.8. Angol-és magyar nyelvre fordítás implementálása

Annak érdekében, hogy elkerüljem a magyar és angol szövegek közvetlen beleégetését a HTML és TypeScript kódokba, kidolgoztam egy megoldást, amely támogatja a magyar-angol fordítást a programban. Ezáltal a szövegek dinamikusan kezelhetők és könnyen bővíthetők a jövőben. A szoftver elején megnyílik a SetlanguageComponent, itt tudjuk beállítani a program nyelvét, itt inicializáljuk a currentLanguage értékét. A reguláris kifejezéseknek, a megjelenített- és kimondott szövegeknek külön-külön létrehoztam egy JSON file-t, melyekben a magyar- és angol nyelvhez értékpárokat hoztam létre.

Részlet az egyik JSON file-ból:

```
{
  "hu-HU": {
    "notfoundallpairs": "Nem találtad meg az összes párt!",
    "retrygame": "Új játék",
    "backtomainpage": "Vissza a főoldalra",
  }
  "en-US": {
    "notfoundallpairs": "You didn't find all the pairs!",
    "retrygame": "Retry",
    "backtomainpage": "Back to the main page",
  }
}
```

Mindkét objektumhoz ugyanolyan nevű értékpárokat rendeltem. Ezeket az file-okat importáltam a komponensekben, és az alapján, hogy a currentLanguage változónak mi az értéke, annak az objektumnak az értékeit használtam fel.

Részlet a GameComponent-ből, ahol a sikeres szöveg tartalmát a spokenText.json-ből olvastuk ki:

```
this.speechSynthesizer.speak(
  this.spokenText.allpairsfound + ". " +
  this.spokenText.gotpoints +
  this.points.toString() + ". " +
  this.spokenText.addaname , this.currentLanguage
);
```


4.9. A játék logikájának implementálása

4.10. Egyéb funkciók implementálása

5. fejezet

Az elkészült alkalmazás ismertetése

5.1. A szoftver végleges specifikációi

5.2. A szoftver hiányosságai, továbbfejlesztési lehetőségek

5.3. Összefoglalás

Irodalomjegyzék

- [1] Introducing VoiceOver

https://www.apple.com/voiceover/info/guide/_1121.html

- [2] VoiceOver felhasználói útmutató

<https://support.apple.com/hu-hu/guide/iphone/iph3e2e415f/ios>

- [3] VoiceOver billentyűparancsok

<https://support.apple.com/hu-hu/guide/voiceover/cpvokys01/mac>

- [4] Braille Screen

<https://support.apple.com/en-us/101637>

- [5] Multi-Touch trackpad

<https://support.apple.com/hu-hu/102482>

- [6] Siri

<https://www.apple.com/siri/>

- [7] Bixby

<https://www.samsung.com/hu/support/mobile-devices/hogyan-hasznalhatom-a-bixby-alkalmazast/>

- [8] SciFY

<https://scify.org/>

- [9] gamesfortheblind

<https://gamesfortheblind.org>

- [10] Provide control labels with ARIA

<https://codelabs.developers.google.com/angular-a11y7>

- [11] ChromeVox

https://chromewebstore.google.com/search/chromevox?hl=en-USutm_source=ext_sidebar

- [12] További bővítmények

https://chromewebstore.google.com/category/extensions/make_chrome_yours/accessibility

[13] web-speech-angular

<https://github.com/luixaviles/web-speech-angular>

[14] Pacman

<https://github.com/chandradharrao/Voice-Controlled-Games-For-Persons-With-Disabilities>

[15] Audio Games For Blind/Low Vision Gamers

<https://veroniiica.com/audio-games-for-blind-low-vision-gamers/>

[16] draw.io

<https://app.diagrams.net/>

[17] A számítógépes beszédfelismerés alapjai

<https://www.inf.u-szeged.hu/tothl/speech/ASRhun.html>

[18] Web Speech API

<https://developer.mozilla.org/en-US/docs/Web/API/WebSpeechAPI>

[19] Using the Web Speech API

<https://developer.mozilla.org/en-US/docs/Web/API/WebSpeechAPI/UsingtheWebSpeechAPI>

[20] SpeechRecognition

<https://developer.mozilla.org/en-US/docs/Web/API/SpeechRecognition>

[21] SpeechSynthesis

<https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesis>

Nyilatkozat

Alulírott Kovács-Bodó Csenge gazdaságinformatikus BSc szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet Szoftverfejlesztés Tanszékén készítettem, gazdaságinformatikus BSc diploma megszerzése érdekében.

Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat a Szegedi Tudományegyetem Diplomamunka Repozitóriumban tárolja.

Szeged, 2024. november 26.

.....

aláírás

Köszönetnyilvánítás

Ezúton szeretném kifejezni hálámat Dr. Jánki Zoltán Richárd témavezetőmnek, aki végig támogatta a dolgozat elkészítését. Különösen értékelem a türelmét és a segítőkész hozzáállását, amelyek nélkül nem érhettem volna el a kívánt eredményeket.

Hálával tartozom azoknak az oktatóknak, tanároknak és szakembereknek, akik az egyetemi éveim során tanítottak és inspiráltak. Az általuk átadott tudás és tapasztalat nemcsak a szakdolgozatomban, hanem az életem minden területén értékes útmutatóként fog szolgálni.