

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
alapszakos hallgató

Témavezető:

Dr. Jánki Zoltán Richárd

Egyetemi adjunktus

Szeged

2024

Feladatkiírás

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. Például 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. Képernyőtervek	12
2.3. Use case diagram	12
2.4. Szekvenciális diagram	12
3. Használt technológia-A Web Speech API	13
3.1. Gépi beszédfelismerés háttere	13
3.2. A Web Speech API	13
3.2.1. SpeechRecognition	14
3.2.2. SpeechSynthesis	14
4. A program megvalósítása	16
4.1. Adatmodellek	16
4.2. Verziókövetés	16
4.3. A játék grafikus interfészének elkészítése, jellemzői	16
4.4. Beszédfelismerés implementálása	16
4.5. Beszédszintetizáció impementálása	16
4.6. Képernyőfelolvasás implementálása	16

4.7. KAngol-és magyar nyelvre fordítás implementálása	16
4.8. A játék logikájának implementálása	16
4.9. Egyéb funkciók implementálása	16
5. Az elkészült alkalmazás ismertetése	17
5.1. A szoftver végleges specifikációi	17
5.2. A szoftver hiányosságai, továbbfejlesztési lehetőségek	17
5.3. Összefoglalás	17
Irodalomjegyzék	18
Nyilatkozat	20
Köszönetnyilvánítás	21

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 Sirinek 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, az [9] oldalon különböző játékokat találhatók, 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 együttjátszást, és segítse a gyerekek fejlődését.

1.4. Chrome-ban támogatott bővítmények és csomagok

Felelehető egy kiváló leírás a Google Codelabs jóvoltából az [10] oldalon, ahol pontokba szedve adnak tippet, 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 elemeket felcímkézzük ARIA label-ekkel, 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:

```
<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 akadálymenetsítést elősegítő bővítmények: Chrome Web Store-ban [12].

1.5. Referencia alkalmazások

Pédaprogramom fejlesztése előtt áttekintettem valamennyi nyílt forráskódú programokat, publikus github repositorykat [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. Képernyőtervek

2.3. Use case diagram

2.4. Szekvenciális diagram

3. fejezet

Használt technológia-A Web Speech API

3.1. Gépi beszédfelismerés háttere

3.2. A Web Speech API

A Web Speech API [16] 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.[17]

3.2.1. SpeechRecognition

A SpeechRecognition [18] 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 mikrofojának 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 [19] API a Web Speech API része, amely lehetővé teszi, hogy egy weboldal beszédfelismerést és beszédszintetizációt valósítson meg, ezzel akár saját képernyőfelolvasó funkciót is implementálhatunk. Az API szöveget alakít hanggá és lejátssza 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

4.1. Adatmodellek

4.2. Verziókövetés

4.3. A játék grafikus interfészének elkészítése, jellemzői

4.4. Beszédfelismerés implementálása

4.5. Beszédszintetizáció impementálása

4.6. Képernyőfelolvasás implementálása

4.7. KAngol-és magyar nyelvre fordítás implementálása

4.8. A játék logikájának implementálása

4.9. 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] A11y

<https://codelabs.developers.google.com/angular-a11y0>

- [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] Web Speech API

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

[17] Using the Web Speech API

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

[18] SpeechRecognition

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

[19] 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 5.

.....

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.