Ignácz Eszter

Vakokat segítő szoftver fejlesztése Összefoglaló

A feladat során egy már elkezdett programot kell folytatni, kiegészíteni. A szoftver azért készült, hogy megkönnyítse a látásukban korlátozott emberek tájékozódását a Windows XP alatt futó meghatározott programok használata közben. Bizonyos objektumok felismerésekor rövid hangfájlokat játszik le, melyet egy meglévő adatbázisból keres ki. Felismeri az ikonokat, menüelemeket, gombokat a kurzor helyzetének lokalizálásával. A szoftvert a célcsoport igényeihez kell igazítani. Elsőként szükséges a meglévő programot megismerni. A hibákat fel kell ismerni, lehetőség szerint megoldást keresni. A célcsoporttal célszerű felvenni a kapcsolatot, és személyesen megkérdezni őket az igényeikről, szokásaikról. Látókkal is érdemes lehet konzultálni viszonyítási alapként. A felhasználói felületet mind a vakok, mind a látók számára könnyen használhatóvá, barátságossá kell alakítani. Mindezt a Visual Studio keretrendszer segítségével igyekeztem megvalósítani. A program C++ nyelven íródott, és a Microsoft Active Accessibility segédprogram által kommunikál az operációs rendszer elemeivel. A kapcsolatot telefonon, személyesen és közösségi oldalakon, a győri Vakok és Gyengénlátók Egyesületével valamint magánszemélyekkel létesítettem. Előre átgondolt és spontán kérdéseket tettem fel, melyek alapján kigyűjtöttem a felhasználható információkat. Látássérült emberek véleményére alapozva néhány változtatást eszközöltem. A grafikus felület mindenki számára egyszerűen használható lett. A hiányzó hanghatásokat implementáltam, elhagytam néhány feleslegesnek ítélt hangot. A hibák közül megmaradt a program időnkénti leállása. Tervben van a program jövőbeni fejlesztése: még jobban a felhasználók igényeihez igazítani, az időnkénti leállást kiküszöbölni, így megbízhatóvá tenni a működését, implementálni a smiley-kat, megoldást keresni a billentyűparancsok meghangosítására.

A műben található kulcsszavak: beszédszintézis, C++, gyengénlátók, Microsoft Active Accessibility, vakság.

Ignácz Eszter

Development of software for blind people Summary

This project is a continue of a started software. I have to supplement this program, what was made for the blind and partially sighted people primarily. It makes easier using the computer programs of Microsoft Windows XP for the visually impaired people. When it localizes a specified object, plays a short voice, what is searched from a database. It recognizes icons, menu items, buttons. I had to be formed it to the needs of the target group. At first it is necessary to study the software. I have to recognize the errors and search solutions if it is possible. It is practical to communicate with the target audience and people with normal vision, to get acquainted with the habits and needs concernig the use of computer to make the software useful. I have to form the user interface to be easier, clearer and friendlier. The framework is Microsoft Visual Studio, the programming language is C++. The MSAA implements the connection with the objects of the operating system. I talked to individuals and the contact person of the Vakok és Gyengénlátók Egysülete in Győr. I asked some preplanned and spontaneaus question. Based on these I collected the useful information. Based the opinion of blind and partially sighted people I made some modifications. The grafical user interface is easy to use for everyone. The error, that the program somtimes stop with error message, still exists. I implemented the missing sound effects and I left some, that the interviewed persons concidered unnecessary. The plan is the further development. That means, I will adjust the software to the needs of users more, make it more trustworthy and implement smileys and shortcuts.

Keywords in the document: blindness, C++, low vision, Microsoft Active Accessibility, text-to-speech.

Tartalomjegyzék

1. Bevezetés	3
2. Célcsoport megismerése, igényeinek felmérése	6
2.1. Néhány alapfogalom	7
2.1.1. Braille-írás	7
2.1.2. "Text-to-speech", azaz a beszédszintézis	9
2.2. A vakok által használt szoftverek	9
2.2.1. JAWS	10
2.2.2. NVDA	11
2.3. A megkerdezettek velemenye	12
2.3.1. Hasznalt programok	13
2.3.2. Hasznart turkciók	15
2.3.4. A vélemények összehasonlítása	15
3 Hibák felderítése	18
	10
3.1. Programieallas	18
	20
4. Egyszerű grafikus felület letrenozása	. 21
4.1. Kezdőoldal	21
4.2. Menü	22
5. Tervezési dokumentáció	. 26
5.1. Hangok	26
5.2. Keretrendszer	27
5.3. Változók és konstansok	29
5.4. Főbb metódusok	30
5.5. Felhasználói felület elemei	30
5.6. Az eredeti es az uj szoitver kulonbsegei	
5.6.2 Változók	
5.6.3 Felhasználói felület	
6 Felhasználói dokumentáció	34
	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
<ul> <li>6.2 Talapítási útmutatá</li> </ul>	
6.3 A szoftver használata	
6.4. Szolgáltatások	
7. Irodalomiegyzék	. 36
······································	

8. Melléklet	
8.1. Forráskód	
8.2. A párbeszédablakok elemei, tulajdonságai	
8.3. További fájlok listája	
8.4. Lemezmelléklet tartalma	47

# 1. Bevezetés

Ebben a dolgozatban egy olyan szoftvert szeretnék bemutatni, mely elsősorban a vakoknak és gyengénlátóknak segít tájékozódni a számítógépes felületen. A program már félkész állapotban került hozzám, egy beadandó dolgozat eredménye. Az eredeti fejlesztők Blindsoft névre keresztelték, és ezen nem kívántam változtatni. Fő funkciója, hogy hanghatást ad ki, ha az egér egy meghatározott objektum fölött halad el, vagy megáll rajta. Objektumnak nevezzük az operációs rendszer alkalmazásainak elemeit, ilyenek például az ikonok, ilyen a mentés gomb, a megnyitás, a kilépés, stb. A lejátszott hanghatás a meghangosított funkcióra jellemző, így megnyitásnál zipzárlehúzás, kilépésnél ajtóbecsapódás hallatszik, néhány esetben pedig ez a hanghatás az objektum neve. A hangeffektusok megerősítést adnak a felhasználónak, hogy tudja, valóban jó helyre szeretne kattintani. Ennek a szoftvernek valamilyen módon kommunikálnia kell az operációs rendszerrel, hiszen információkat kér az egyes objektumokról. Ebben segít a Microsoft Active Accessibility (MSAA), mely a Windows 95-től kezdve az összes Windows operációs rendszer része. Ő biztosítja a kommunikációt az alkalmazások és az általam fejlesztett szoftver között. Elérhetjük vele az objektum tulajdonságait, köztük például típusát, nevét, helvét, állapotát, információt kapunk az eseményekről, például arról, hogy az egérrel az objektum fölött haladunk el, vagy az objektumra kattintunk. Azokat az alkalmazásokat, amelyeknek elemeiről információt kérünk, szervernek nevezzük. Ő szolgáltatja a kért adatot. Az az alkalmazás, amely kéri az adatokat, a kliens. Ebben az esetben az operációs rendszer alkalmazásait tekinthetjük szervernek, a felhasználó által látható elemeiről szeretnénk megtudni dolgokat. A kisegítő szoftver pedig a kliens, ő intézi a kérést a szerver felé. A kliens az MSAA-t használja ennek érdekében. Nemcsak vakoknak és gyengénlátóknak ajánlott használni a programot, hiszen bárkinek segítséget nyújthat munkája során. Így például a számítógéppel való munka közben megerősítést nyerhetünk, hogy valóban a mentésre fogunk-e kattintani, vagy elvétettük a mozdulatot.

A vakoknak a hétköznapi élethez, a járáshoz, a munkához, mindennapi tevékenységeikhez sok segédeszközt kell használniuk, mivel valamilyen módszerrel kompenzálni kell a hiányzó

érzékszervet. A járáshoz a fehér botot használják, amivel a körülöttük lévő területet biztosítják, sokaknak pedig egy barátságos segítőtársuk akad, a vakvezető kutya. Tiszteletre méltó, hogy mennyire odaadó szolgálói ezek az állatok a rászoruló embereknek. Az úttesten helyenként az útjelző lámpa hangjelzése segíti át őket. Egy főzési folyamathoz például hagyatkoznak a tapintásra az étel előkészítésénél, főzés közben hallhatják, hogy mennyire forr a víz, érezhetik a megfelelő illatokat. A lakáson belüli közlekedéshez a tapintásukra és az emlékezetükre tudnak hagyatkozni. A figyelemnek azt a részét, amit a látó emberek a látásra fordítanak, ők a többi érzékszervükre fordítják, így azok valamivel fejlettebbek is a többi emberénél. Ezért vannak rendkívül hátrányos helyzetben a *siketvak* emberek. Ők azok a vak emberek, akiknek a hallásuk is nagy mértékben károsodott. Ők még kevesebb érzékszervükre hagyatkozhatnak. Ezek az emberek rá vannak kényszerülve, hogy a tapintásukkal kommunikáljanak. Egy ilyen lehetőséget biztosít a Braille-írás. Az írás lényege, hogy a karaktereket az ujjbeggyel kell kitapogatni. Karakterei megfeleltethetőek az abc betűinek. Így a vak emberek is tudnak olvasni anélkül, hogy szükségük lenne a látásra. A vakok közül sokan előszeretettel alkalmazzák ezt a módszert, sokan pedig idegenkednek tőle. Ez persze nem olyan egyszerű, mint gondolnánk, mivel meg kell tanulni a Braille-abc betűit, és újra meg kell tanulni írni, olvasni. Ennek ellenére érdemes időt szánni rá, hiszen ha nem áll rendelkezésre más eszköz, akkor így is lehet kommunikálni. Ezen felül megadja a vak embereknek is az olvasás élményét. A számítógép iránti érdeklődés inkább a fiatal és középkorú emberekre jellemző; nyitottabbak az új dolgokra, a szórakozásra, valaki pedig munkaeszközként használja nap mint nap. A számítógép használatához nagyon sok kisegítő szoftver létezik. Ezek között van a gyengénlátók számára, a vakok számára és a siketek számára is megfelelő. Van, amelyik ingyenes, olyan is van, aminek magasan van az ára. A vakok számára a felolvasóprogramok a leghasznosabbak. Ezek közül a dolgozatban röviden be fogok mutatni kettőt. Mindegyik segítő szoftver értékes, hiszen bármilyen segítség hozzájárul ahhoz, hogy a sérült emberek jobban kivehessék a részüket a világ szépségeiből.

Segítségemre volt néhány ember a célcsoportból, illetve néhány látó személy. Készségesen válaszoltak a feltett kérdésekre, melyek közül néhány helyben alakult ki. Tudni kell, hogy a megkérdezettek nagy része használ valamilyen kisegítő szoftvert, és bármilyen apró ötletnek rendkívül örülnek. A látó és nem látó csoportnak más-más igényei vannak, viszont a célcsoport igényeihez igyekeztem igazítani a szoftvert. Mindkét féltől kaptam építő jellegű

kritikákat, beépíthető funkciókat, valamint kevésbé fontos információkat. Sokat tanultam a beszélgetések során, és sikerült a célnak megfelelőbbé tenni a szoftvert. Rengeteg olyan információt kaptam, amit a későbbiekben még szeretnék beépíteni a programba. A fejlesztés során természetesen nem egy problémába ütköztem. Ezek közül nem sikerült mindent kiküszöbölni. Valamire rengeteg elfecsérelt idő után sem sikerült megoldást találni, néhány dolgot pedig sikerült megoldanom.

A témaválasztásom azért erre esett, mert mindig foglalkoztattak a testileg és/vagy szellemileg sérült emberek. Egy egyetemi tantárgy keretében eltöltöttem egy félévet egy győri szociális foglalkoztatóban, itt kerültem közelebb sérült emberekkel. Ezalatt az idő alatt azt tapasztaltam, hogy még segítséggel is nehezen állják meg a helyüket a mindennapi életben, legyen szó akár testileg, akár lelkileg sérült emberekről. Az optimális számítógép-felhasználók azok az emberek, akiknek mindegyik érzékük megvan és alkalmasak elvégezni a használatához szükséges alapvető mozdulatokat. A sérült emberek ilyen szempontból kiszolgáltatottak, hiszen nem tudják a megfelelő módon használni a számítógépet. A mai világban már olyan hosszú a kisegítő eszközök listája, hogy jobbnál jobbak közük válogathatunk. Így van ez a számítógépes kisegítő szoftverekkel is. Olyan színes skálát kínálnak a piacon, hogy a legtöbben megtalálják azt, ami nagyban hozzájárul a számítógép használatának megkönnyítéséhez. Ha én is tudok ezzel a programmal akár kicsit is segíteni a látássérült emberek kiszolgáltatottságán, akkor máris hasznosnak érzem tevékenységemet.

# 2. Célcsoport megismerése, igényeinek felmérése

"Az a látássérült ember, akiknek látásteljesítménye az ép látáshoz (100%) viszonyítva két szemmel és korrigáltan is 0-33% közötti, vagy a látótér jelentősen (10-20 fokban) beszűkült." [1] Nemcsak egyféle vakság létezik. Lehet, hogy a látássérült egyáltalán nem érzékeli a fényt, ilyenkor teljes vakságról beszélünk. Részleges vakság esetén megmaradhat a fényérzékelés, ekkor az egyén csak a világos és a sötét megkülönböztetésére képes. Van, aki még azt is meg tudja mondani, merre van a fény forrása, ők a fényérzékelők. Beszélhetünk aliglátókról, ők az ép látás maximum 10%-ával rendelkeznek. A gyengénlátó emberek nem tartoznak a vakok közé, viszont az ő látásuk is súlyosan károsodott. A hivatalos meghatározás szerint a jobban látó szemen a szeműveggel javított látás a teljes látásnak 10-33%-a. A siketvak emberek pedig olyan vakok, akiknek annyira sérült a hallásuk, hogy a látás semennyire nem kompenzálható ezzel az érzékükkel. Az ő esetükben különböző speciális eszközökkel, módszerekkel tudnak segíteni a tájékozódásban, gyengénlátás esetén nagyítóval, nagyothallás esetén hallókészülékkel. Magyarországon kissé elérhetetlenek az egyéb segédeszközök, mivel nem megfizethetőek a hétköznapi ember számára. Vannak olyan országok, ahol létrehozták a jogi vakság fogalmát, hogy meg tudják határozni, ki szorul gondozásra. Ide nem csak a teljes vakokat sorolják, hanem például olyan embereket is, akiknek a látószögük maximum 20 fok. A látássérült emberek különböző segédeszközöket használnak, hogy kompenzálni tudják a látás hiányát. A biztonságos közlekedésben segít nekik a fehér bot, aminek több változatát is használják, többféle anyagból és többféle elemből készülhet. Segítségükre van még például a vakvezető kutya, a hangjelzést adó lámpa. Olvasni a Braille-írás segítségével tudnak, a számítógépen való eligazodást a beszédszintézis segíti. [2] A két utóbbi fogalommal a 2.1. pontban ismerkedünk meg.

A számítógépes megjelenítés legjobb eszköze a grafikus felhasználói felület. Ez azt jelenti, hogy a monitoron keresztül láthatjuk az operációs rendszer elemeit, az alkalmazásokat, az ikonokat, stb. Ezeket az elemeket az évek során egyre színesebb és sokatmondóbb külsővel látták el, legyen szó bármilyen szoftverről. A vakok esetében minden ilyen jellegű fejlesztés

értelmét veszíti. Ezért az ő számukra a szoftverek egy teljesen más csoportját fejlesztették ki. A számítógépen való tájékozódáshoz ezeket a szoftvereket veszik igénybe. Persze tartalmaznak grafikai megjelenítést is, hiszen nem egy esetben egy látó végzi el a vak embernek a megfelelő beállításokat a szoftverrel kapcsolatban. Általában a grafikai felületre nem fektetnek nagy hangsúlyt. Ezek leginkább olyan programok, amelyek valamilyen más módon próbálják közvetíteni az információt, mint a monitor. Ez történhet például hanggal vagy speciális kijelzővel. A legtöbb ilyen program felolvassa a megjelenítendő információt. A szöveg beszéddé való átalakítását a beszédszintézis végzi, ezért szokás ezeket a programokat beszédszintetizátoroknak nevezni. A mai világban már semmiből sincs egyféle, a beszédszintetizátorokból sem, ahogy ezt már a Bevezetésben is említettem. Az általam fejlesztet szoftver is ilyen, hanghatásokat ad ki az operációs rendszer egyes objektumainak hatására. Szükséges volt tehát megismerni a célcsoport számítógépes szokásait, mely programokat használják a legtöbbször, milyen műveleteket hajtanak végre a leggyakrabban. Különböző felmérésekből kiderül, hogy a vakok néhány programot használnak csak rendszeresen, nagyon sokat pedig ritkán vagy egyáltalán nem. Ez adódhat abból is, hogy sok vizuális megjelenítést alkalmazó program van, amit nem tudnak kihasználni, vagy abból, hogy bonyolult a program használata, esetleg abból is, hogy a szórakozáshoz, munkához elég az a néhány program, aminek igénybe veszik a szolgáltatásait.

# 2.1. Néhány alapfogalom

#### 2.1.1. Braille-írás

Az írás lényege, hogy a papíron kidomborodó betűket az ujjbegyünkkel kell letapogatni. Az abc-ben minden betűnek megfelel egy minta. Az írás jelei 6 pont kombinációjából állnak, amelyek 2 pont szélességben és 3 pont magasságban helyezkednek el kemény papírba nyomva. Az új információt a pontok közti szünet jelzi. Az írás *Braille-tábláv*al vagy pontírógéppel papírlapra történik. A számítógépre speciális *Braille-billentyűzet*tel lehet bevinni a szöveget, *Braille-nyomtató*val pedig ki lehet nyomtatni. Ezenkívül a számítógépre lévő szöveget *Braille-kijelző*n is megjeleníthetjük – ezt az eszközt a képernyőolvasó programok és a normál billentyűzet kiegészítőjeként szokták használni. A *Braille-írás*sal

írhatunk kottát, számokat, törteket, képleteket, egyenleteket, és sok egyéb jelet, valamint különböző formázási lehetőségeink is vannak. [2] [3]



1. ábra. Louis Braille portréja

Az írás megalkotója *Louis Braille*. 1824-ben fejlesztette ki ezt a speciális abc-t a francia karakterekre alkalmazva, mégpedig kifejezetten a vak emberek számára. Ezzel az abc-vel lehetővé tette a vakok számára, hogy olvasni és írni tudjanak. *Charles Barbier* alkotta meg a rendszer alapját, hiszen ő fejlesztette ki az *éjszakai írást Napóleon* kérésére. A célja az volt, hogy a katonák éjszaka, fény nélkül is tudjanak kommunikálni. A katonák túl nehéznek tartották ezt az érintésen alapuló olvasást, így a hadsereg elvetette az ötletet. *Barbier* 1821-ben ellátogatott a párizsi *Vakok Intézeté*be, ahol ismertette rendszerét. *Braille* ez alapján alkotta meg a saját rendszerét. A nehezen kezelhető *éjszakai írást* leegyszerűsítette 6 pontosra, mivel észrevette, hogy a hat-pont rendszer egyetlen ujjhegy tapintásával felismerhető. [4]

Sokan nem szívesen tanulják meg ezt az abc-t. Például, aki idősebb korára szenved valamilyen látáskárosodást, annak nehezebb dolga van, hiszen gyakorlatilag az elejéről kell kezdeni az írás- és olvasástanulást, és ebben a korban már a tapintás érzékenysége is csökken. A fiatalabbak pedig általában jobban kedvelik a képernyőolvasó programokat, az ő világukban is már fontos szerepet játszik a számítógép.

Elmondható az a tény, hogy akik használják a *Braille-írást*, jobban tudnak helyesen írni, könnyebben tudnak tanulni. Olvasás közben a saját tempónkban haladhatunk, a saját

hangsúlyainkkal érezhetjük át a szöveget. A siketvak emberek pedig rá vannak kényszerülve a használatára, mivel ők a hallásukra sem tudnak biztosan támaszkodni. Én azon a véleményen vagyok, hogy bátran kéne népszerűsíteni a látássérültek körében, hiszen nagyon sok előnyét láthattuk.

#### 2.1.2. "Text-to-speech", azaz a beszédszintézis

A *beszédszintézis* lényege, hogy mesterségesen hozunk létre emberi hangot. Az eljárás az írott szöveget átalakítja hangokká különböző hangfeldolgozó eljárások segítségével. Ez nehéz feladat, mivel ugyanazt a mondatot többféle hangsúllyal is mondhatjuk, így már mindjárt mást jelent. Mindig figyelni kell a szövegkörnyezetet is. Az is probléma, hogy más és más betűkészletek léteznek a világon. Néhány nyelvnél egy-egy karakter új, valahol pedig teljesen másképp írjuk az összes betűt, mint például az orosz vagy a görög nyelvben. Valahol pedig nem is egy betűt jelöl egy karakter, hanem egy szótagot, szót vagy szóösszetételt. Ilyen a kínai nyelv is. Általában a beszédszintézisen alapuló programok a legtöbb nyelvre jellemző módon működnek: szavanként alakítanak át beszéddé. A kész beszédelemek egy adatbázisban vannak eltárolva. A felismert szöveget először át kell alakítani a gép által ismert nyelvezetté, majd az adatbázisból a program a szükséges elemeket kikeresi, és a megfelelő sorrendben lejátssza. Az átalakítást előre meghatározott szabályok szerint végzi.

Sokan próbálkoztak több-kevesebb sikerrel olyan gépezeteket készíteni, amelyek kimondott szöveget alkotnak valamilyen forrásból. Ez egészen az 1700-as évek végére nyúlik vissza, amikor egy dán tudós, *Christian Kratzenstein* és a magyar származású *Wolfgang von Kempelen* is sikereket ért el az ilyenfajta gépek megalkotásában. Sok operációs rendszerben régóta van beépített beszédszintetizátor, egészen az 1990-es évektől kezdve. Már létezik Mozilla Firefox kiegészítő, melynek a beszédszintézis a feladata. [5]

### 2.2. A vakok által használt szoftverek

A látás hiányának mértékétől függően népszerűek a képernyőnagyító szoftverek, amelyeknél lehetőség van felnagyítani a képeket, szövegeket. Valamelyikük támogatja a hanglejátszást, felolvasást is. A számítógépes kisegítő eszközök közül a vakok leggyakrabban a felolvasó programokat használják, hogy kommunikálni tudjanak a géppel. Fő funkciójuk, hogy a vak emberek számára hozzáférhetővé tegyék a képernyőn lévő üzeneteket. Ezt

többféleképpen képesek megtenni, például a hangkártyán keresztül vagy *Braille-kijelző*n át. Ez attól függ, hogy melyik eszközt támogatja a program. A felhasználó akár ki is tudja választani, hogy melyik módszert részesíti előnyben. Működésüknek az alapja általában a 2.1.2. pontban tárgyalt beszédszintézis. Az ilyen jellegű szoftverből sokféle létezik. A célcsoport legszívesebben azokat alkalmazza, amely hanggal adja tudtukra a számítógép által megjelenítendő üzeneteket. Ezek közül a szoftverek közül én most kettőt emelnék ki. Az egyik a JAWS (Job Access With Speech) nevű képernyőolvasó program, a másik az NVDA (NonVisual Desktop Access). Mindkét program ismertebb a vakok és gyengénlátók körében. Inkább az előbbit használják, mivel az megbízhatóbb. Az utóbbi nyílt forráskódú, így nincs szigorú feltételekhez kötve a fejlesztése, és lehetnek benne akár olyan funkciók, amelyeket sokan nem támogatnak.

#### 2.2.1. JAWS

A JAWS egy Windows alatt futó kisegítő szoftver, mely beszédszintézist használva felolvassa a képernyőn látható szöveget. Támogatja a *Braille-kijelző*t is. Sok lehetőséget nyújt, mellyel a látássérült emberek könnyebben tudják használni a programokat, könnyebben böngésznek a világhálón és szerkesztenek dokumentumokat. A világon ez a legnépszerűbb képernyőolvasó, ezt használják a legtöbben. Az általam megkérdezett emberek is kivétel nélkül a JAWS-szal dolgoznak.

Eredetileg *Ted Henter* és *Rex Skipper* alkották meg a JAWS első változatát az 1980-as évek közepén. *Skipper* helyett *Charles Opperman* folytatta a fejlesztést. Őt később a *Microsoft* felbérelte, majd 1995-ben meg is jelent a JAWS for Windows 1.0. A legutóbbi változat a 14.0, ez 2012. október 22-én jelent meg. A fejlesztések során belekerült számos segítség a használathoz, egyre több program támogatására került sor, mint például böngészők, olvasók, valamint mindig az éppen aktuális Windows-hoz igazították.

A 11.0 verzió támogatja a Windows 7 operációs rendszer 32 bites változatát, a 2007-es és 2010-es Microsoft Office csomagot is, valamint a Mozilla Firefox és Internet Explorer újabb változatait is. A *Braille-kijelző*k használatát is lehetővé teszi. Lehetőség van benne a számítógép távvezérlésére, automatikus frissítésre és áttekinthetőbb a böngészés. A beszéd a Ctrl billentyűvel ki- és bekapcsolható. A sorok között a fel/le nyilakkal tudunk váltogatni, a linkek között a tabulátor billentyűvel tudunk váltani. A

magyar változat a Profivox beszédszintetizátor négy hangkarakterét tartalmazza, két férfit, két nőit. [6]

#### 2.2.2. NVDA

Ingyenes felolvasóprogramok is rendelkezésünkre állnak, ilyen az NVDA, amelyet szintén Windows számára terveztek. Python programozási nyelven készült, és mivel nyílt forráskódú, szabadon lehet fejleszteni. Hasonlóan a JAWS-hoz, támogatja a beszédszintézist, tartalmazza az eSpeak nevű beszédszintetizátor programot, mely szintén ingyenes, nyílt forráskódú. Támogat néhány típusú *Braille-kijelzőt*. A telepítője könnyen használható. Futtatható telepítés nélkül, külső adathordozó segítségével is. Ennek is egyszerű a módja, csak meg kell adni a célmappát, majd megtörténik a szükséges adatok másolása. A program támogatja a Mozilla Firefox 3-at, az Internet Explorert, A Microsoft Outlook, Word és Excel programokat, az Adobe Readert, a vakbarát Java alkalmazásokat, és még sok más programot.

Üdvözli az NVDA	· · sylmatri	X
Üdvözi az NVDA! A legtöbb NVDA parancs használatához tartsa lenyomva Alap esetben a numerikus, valamint a normál insert bille Az NVDA gomb a CapsLock billentyűre is beállítható. Nyomja meg az NVDA +n billentyűparancsot az NVDA me Itt megváltoztathatja a beállításokat, megtekintheti a su	a az NVDA gombot, üsse le hozzá a billentyűparancsot, majd engedje fe ntyűk használhatóak NVDA gombként. enüjének aktiválásához. igót és további lehetőségeket érhet el.	l mind a kettőt.
☑ A CapsLock billentyű használata NVDA gombként	☑ Jelenjen meg ez a párbeszédpanel az Nvda minden indításakor	
	Igen	

2. ábra. Az NVDA indulóképernyője

A <u>http://www.nvda.hu/</u> honlapról tölthető le, a *Letöltések* menüpont alatt találjuk. A telepítés során is hallhatjuk már az útmutatót. A felolvasás nyelve automatikusan magyar, viszont később ezt meg tudjuk változtatni. Ha a programot elindítjuk, a *2. ábrán* látható képernyőállás fogad minket. Nem túl látványos, viszont minden fontos információ megtalálható rajta. Tekintve, hogy ki a célcsoport, ez teljesen kielégítő. A hang felolvassa a tájékoztatót, ami alapján lehetőségünk van testre szabni a program billentyűparancsait. A menü aktiválásához le kell nyomnunk egyidejűleg az NVDA billentyűt (ami alapértelmezett esetben a Caps Lock billentyű) és az n karaktert. A menüben rengeteg beállításra nyílik

lehetőségünk, még akár adományozhatunk is segítségképpen. Kiválaszthatjuk az alapértelmezett NVDA billentyűt, megtekinthetjük az egyes billentyűparancsokhoz tartozó eseményeket, majd beállításainkat el is menthetjük, hogy ne kelljen a következő indításnál újra megtenni az összes lépést.

Az NVDA egy jó példa arra, hogy non-profit módon is segíthetünk az embereken. Ez a program nem tökéletes, de funkcióinak nagyon színes skálájáról válogathatunk. Ezenkívül bárki elvégezhet rajta olyan fejlesztéseket, amitől jobb lesz, és ez nagyon jó kiindulópont egy még megbízhatóbb kisegítő szoftver elkészítése felé. Erre alapozva megfontolandó, hogy felvegyem a kapcsolatot az NVDA főbb fejlesztőivel. Az NVDA hangjait érdemes lehet lecserélni esetleg az általam fejlesztett szoftver hangjaira, azokat a felhasználó által választhatóvá tenni. Így eldöntheti, hogy a szöveget szeretné hallani, vagy az új hanghatást. Mivel más nyelven íródott, nehezebb lehet a két programot közelebb hozni egymáshoz. Akár lehetne olyan megoldást is találni, hogy az NVDA elemeiből ötleteket felhasználni a szoftverbe, vagy ezt a szoftvert átírni az NVDA nyelvére. A lehetőség sok, érdemes minél több irányban elindulni és próbálkozni, hátha elérhető fejlődés.[7]

### 2.3. A megkérdezettek véleménye

Több helyről kaptam segítséget, de egy fiatal gyengénlátóval részletesen is tudtam beszélni a számítógépes szokásairól, szükségleteiről. Bár *gyengénlátó*, de számítógép használat szempontjából a *vak* felhasználók közé tartozik. Fontos leírnom, hogy rendkívül segítőkész volt, minden tapasztalatát elmondta, és a jövőben is rendelkezésemre áll, hogy információt szolgáltasson a még felmerülő kérdésekben. Olyan hasznos dolgokat tudtam meg tőle, amik nagyban segítették munkámat, és a program további fejlesztése során is fel tudom használni a véleményét. Néhány kérdést előre megfogalmaztam, ezeket a 2.3.1. és 2.3.2 pontban fogom ismertetni. A beszélgetések során más kérdések is megfogalmazódtak bennem, és volt, amit meg sem kellett kérdeznem. Ezt pedig a 2.3.3. pontban fogom részletezni. Rendelkezésemre állt továbbá az eredeti szoftver dokumentációja, melyben ismertetve van egy 150 főből álló felmérés egy része. A felmérés 100 látó és 50 vak ember mindennapi számítógépes szokásait mutatja. Az 2.3.4. pontban össze fogom hasonlítani a felmérés eredményét az általam megtudott véleményekkel.

#### 2.3.1. Használt programok

"Milyen gyakran használja a számítógépet?"

"Milyen operációs rendszert használ?"

"Milyen rendszerességgel használja a következő programokat:

- Microsoft Word
- Microsoft Excel
- Microsoft Access
- Microsoft PowerPoint
- Microsoft Outlook
- Internet Explorer
- Mozilla Firefox
- Google Chrome
- Adobe Reader
- Intéző
- Lomtár?"

"Az előző kérdésben felsorolt programokon kívül milyen alkalmazásokat használ még?"

A válaszadó a Windows nagyon sok szolgáltatását igénybe veszi. Napi 8-10 órát ül a számítógép előtt, és a felsorolt programok közül szinte mindet használja. Gyakorlatilag a gép a munkaeszközének tekinthető. Napi rendszerességgel a Microsoft Word, Microsoft Excel, Internet Explorer, Mozilla Firefox és Google Chrome programokat használja. A Microsoft Outlook fiókját heti rendszerességgel nézi, ugyanilyen gyakorisággal néz PowerPoint diavetítést. A diaszerkesztéssel már gondjai akadnak. Microsoft Accesst is használ havi rendszerességgel. A Lomtárat pedig 3-4 havonta üríti.

Ezekből az információkból azt szűrtem le, hogy a Microsoft Office csomag és a böngészők funkcióit érdemes meghangosítani, illetve a programok ikonjait. Különösebb hangsúlyt nem kell fektetni a PowerPoint szolgáltatásaira; mivel ez a program egy bemutató-készítő alkalmazás, így kevés előnyét élvezheti a célcsoport. Az Intézőre, az Adobe Reader-re időnként szükség lehet, viszont csak a fő funkciókat érdemes itt is kiemelni (például megnyitás, mentés, kilépés). Az Access-re kevesebb szükség van, itt is hasonlóképpen érdemes eljárni, mint az Intéző vagy az Adobe Reader esetében.

#### 2.3.2. Használt funkciók

"Milyen rendszerességgel használja a következő funkciókat:

- mentés
- mentés másként
- megnyitás
- bezárás
- kilépés
- másolás
- kivágás
- beillesztés
- új (például mappa, txt fájl)
- lomtár ürítése?"

"Ezeken kívül milyen funkciókat használ még gyakran?"

A megkérdezett fél napi szinten használja ezeket a funkciókat. Legtöbbször a mentés, mentés másként, kivágás, másolás gombokat. Az előző pontban már utaltam rá, hogy a lomtár ürítését csak nagyon ritkán teszi meg. Amit szívesebben használ, mint a gombokat, azok a billentyűparancsok, mint például: mentés – Ctrl+S, másolás – Ctrl+C, beillesztés – Ctrl+V.

A véleménye alapján leszűkíthetjük a szükséges funkciók körét a következőkre:

- mentés
- mentés másként
- megnyitás
- kilépés
- másolás
- kivágás
- beillesztés
- új elem létrehozása.

Ezek tulajdonképpen azok a funkciók, amik valamilyen dokumentum szerkesztéséhez

szükségesek, és még néhány egyéb programban is megtalálhatók.

#### 2.3.3. Észrevételek, vélemények

Elég sok javaslatot, észrevételt kaptam segítségként mind gyengénlátó, mind vak szempontból. Megtudtam, hogy a vakok szívesebben használják a billentyűparancsokat. Ez érthető, hiszen könnyebben csúszik félre a kéz az egér használata közben, míg billentyűzettel több támpontunk is van a helyes billentyű megtalálásához. Valamint nem minden felolvasó program olyan részletes, hogy az összes funkcióra kiterjedjen, így lehetséges, hogy nem találjuk meg elsőre vagy akár többedszerre se a keresett funkciót. A gyengénlátó felhasználók közül is sokan ezt a módszert választják a nagyítóprogramok helyett, hiszen akkor nagyon kell koncentrálniuk, sokkal hamarabb elfáradnak. Személyes véleményként megtudtam azt is, hogy az általam fejlesztett program inkább a gyengénlátók számára hasznos. Később ezt beláttam én is, mert ha ők használják inkább az egeret, akkor valóban náluk van értelme hangot rendelni az egyes funkciókhoz, ha végighalad rajta az egér. Amit ebből tanultam, hogy érdemes lehet a vakok számára akár a billentyűparancsokhoz hozzárendelni a hangokat. Ez a gondolat már korábban is felötlött bennem, de így már biztos, hogy később is érdemes lesz foglalkozni vele. A következő javaslat az volt, hogy ne legyen olyan sok hang a programban, mert az nagyon zavaró lehet. A felhasználó pedig igény szerint hadd válassza ki, hogy mihez szeretne hangot rendelni. Ezt nagyon jó ötletnek tartottam. Amit meg is valósítottam belőle, hogy leszűkítettem a legfontosabb hangokra a listát. A későbbiekben tervezem, hogy a hallani kívánt hangok tetszőleges összeválogatását is lehetővé teszem. A vélemények szerint hasznos lenne még, ha a Sajátgép vagy a Dokumentumok és egyéb ikonok is adnának hangot, ha a felirat valamilyen vizuális jelzést adna, ha odaér az egér, ha a helyi menü megjelenésekor is megszólalna egy speciális hang. Így a javaslat alapján néhány ikont elláttam hangjelzéssel. A vizuális jelzésen nagyon sokat gondolkoztam, és még nem jutottam döntésre. Érdemes lesz a jövőben még sok véleményt meghallgatni, hogy minél biztosabban tudjam, hogy mire kell a legnagyobb hangsúlyt fektetni.

#### 2.3.4. A vélemények összehasonlítása

Az információkat tehát a dokumentációban szereplő felmérés alapján és az általam megkérdezett emberek véleménye alapján összesítettem. Hatalmas eltérések vannak a vakok

és látók között, kisebb eltérés van a most és a régebben megkérdezett célközönség véleménye között.

A vakok szempontjából:

- az asztalon lévő ikonokat használják,
- a böngészőt sokat használják,
- a böngészőben inkább új ablakot, mint új lapot használnak,
- legtöbbször az előtérben lévő, általános funkciókat használják,
- nem szívesen dolgoznak leginkább vizuális megjelenítést alkalmazó programokkal.

A látók véleménye:

- szívesen használják a könyvtárstruktúrában való kereséshez használatos programokat,
- a böngészőt sokat használják,
- a böngészőben szívesebben váltanak a lapok között, mint az ablakok között,
- előszeretettel alkalmazzák a vizualitásra épülő alkalmazásokat.

Az én véleményfelmérésem alapján:

- az asztalon lévő ikonok kerülnek előtérbe,
- a böngésző fontos szerepet játszik,
- ritkán jön elő a színes vizuális megjelenítést alkalmazó programok használata,
- a fiatalok nyitottabbak a nehezebb programok használatára is.

Nem nehéz megérteni az eltéréseket a látók és vakok között, illetve a fiatalok és idősebbek között. Nyilvánvalóan a látók szeretik a látványos dolgokat, a képes illusztrációkat. Könnyen eligazodnak a könyvtárszerkezetben, mivel a látással azonnali és egyértelmű visszajelzést kapnak tevékenységükről. A vakok az egyszerű megoldásokat kedvelik, könnyen elkeveredhetnek a bonyolult mapparendszer mélyén. Egyszerűbb az asztalon lévő programokat használni, nem kell máshonnan kikeresni. Az alkalmazásoknak a látók nagyon sok funkcióit használják, kipróbálják, a vakok számára viszont elég a főbb funkciók hanggal való ellátása. Amit mindenki fontosnak tartott, hogy ne legyen túl sok hanghatás, mivel az zavart okozhat. Ezért úgy gondoltam, hogy a fontosabbnak ítélt programok ikonjait, és ezek sokszor használt funkcióját hangosítom meg. A véleményekre alapozva további fejlesztések során megvalósítom a hangok kiválasztását, azaz a felhasználó egy beállítás ablakban maga válogathassa össze, hogy mely ikonhoz és funkcióhoz szeretne hangot rendelni.

#### A meghangosított ikonok:

- Sajátgép
- Dokumentumok
- Microsoft Word
- Internet Explorer.

#### A meghangosított funkciók:

- megnyitás
- mentés
- mentés másként
- nyomtatás
- kivágás
- másolás
- beillesztés
- törlés
- új
- kilépés
- start

Összevetve a véleményeket, úgy gondolom, hogy ezeket az ikonokat és funkciókat elég hanggal ellátni. Figyelembe kellett vennem azt is, hogy ne legyen túl sok hang a programban, azt is, hogy a fontosabb, sokszor használt programok ikonjaira és funkcióira kell koncentrálni. Ez a lista magában foglalja a legtöbbet használt ikonokat és funkciókat. Ezek nemcsak egy programban elérhetők, hanem az összes olyanban, amely tartalmazza a funkció nevét. Így például a megnyitás nem csak a Microsoft Wordben lesz elérhető, hanem mindegyik szoftverben, amelyben szerepel a megnyitáshoz hozzárendelt nevek egyike. Ha valaki az összes hangot szeretné hallani, még akkor sem túlzottan zavaró. Ha pedig csak 1-2 hang van, ami segítséget nyújtana, akkor lehetősége lesz a későbbiekben kiválasztani a megjelenő listából ezeket.

A vélemények felmérését a továbbiakban is folytatni fogom, hiszen fontos, hogy minél több embert kérdezzek meg a számítógépes szokásairól, esetleges jövőbeni igényeiről. Erre alapozva hitelesebben tudom majd tovább fejleszteni a programot.

# 3. Hibák felderítése

A program tanulmányozása során néhány problémába ütköztem, melyek közül valamit sikerült megoldanom, valamire pedig nem találtam megfelelő megoldást. Nehéz volt, hogy már egy meglévő programban kellett hibákat keresnem.

## 3.1. Programleállás

Időközönként leáll a program hibaüzenet kíséretében. Különféle hibaüzeneteket kaptam, amelyeknek természetesen utána is jártam, viszont jó megoldást nem találtam rá. A leggyakrabban a *3. ábrán* látható hibaüzenetet kaptam.



3. ábra. Hibaüzenet

Ami a legfeltűnőbb volt, hogy általában akkor történt meg, amikor a Microsoft Excel programot indítottam el. Ezt mutatja a *4. ábra*. Ez az üzenet pont egy olyan esetben készült, amikor az említett szoftvert elindítottam. Különösnek találtam, hogy egy konkrét program indulásakor rendszeresen megtörtént ez a probléma.

Blindsoft: example1.exe - Alkalmazáshiba	×
Kivétel (ismeretlen szoftverkivétel (0xc0000409)) történt az alkalmazásban a következő l	helyen: 0x77e8b0d7.
OK Mégse	

4. ábra. Hibaüzenet az Excel megnyitásakor

Volt olyan eset, amikor még a segítő szoftver megnyitása előtt elindítottam az Excelt, ekkor az volt a jellemző, hogy rövid használat után szintén leállt. Például egy ilyen hiba esetén készült az 5. *ábra*.



5. ábra. Hibaüzenet Excelben való munka közben

Egy-két alkalommal néhány program elindulásakor is megtörtént, ez véletlenszerűnek látszott. Amelyik pedig sok aktív használat után keletkezett, azt a *6. ábra* mutatja.

A(z) example1.exe hil kellemetlenségekért (	bát észlelt, ezért leáll. A elnézését kérjük.
Az adatok, melyekkel é	open dolgozott, valószínűleg elvesztek.
Tájékoztassa a Mic A hibajelentés elküldése adatokat bizalmasan és	rosoftot a problémáról. ivel segítheti munkánkat. A jelentésben közölt név nélkül kezeljük.
A hibajelentésben szere	plő adatok megtekintéséhez <u>kattintson ide</u>
100 1 1	

6. ábra. Hibaüzenet hosszú működés után

A hibák különböző időközönként jelentkeztek. Néha azonnal indulás után, néha egy-két perc munka után, valamikor pedig jóval hosszabb idő után. Mindegyik leállásra az volt a jellemző, hogy néhány pillanatra jelezte az operációs rendszer, hogy várakozzak (Windows XP esetében homokóra), majd megjelent egy hibaüzenet.

Felmerült a kérdés, hogy vírus dolgozik-e, vagy talán van néhány frissítetlen szoftver a számítógépen, viszont a víruskereső lefuttatása után, és a szoftverfrissítés után bebizonyosodott, hogy egyik sem lehet a probléma. Miután tanulmányoztam kicsit a programot, felfedeztem néhány sort, ami tulajdonképpen nélkülözhető volt. A sorok nélkül újra lefordítottam a programot, és úgy tűnt, hogy több ideig működött, mielőtt újra leállt. Az Excel indításakor fellépő kilépés még megmaradt. Arra nem derült még fény, hogy a változtatás miatt történt mindez, vagy pedig csak a véletlen műve. A jövőben tervezem, hogy kutatok tovább a fennálló hiba oka után.

### 3.2. Hiányzó hanghatások

Néhány elem hatására a program nem adott ki hangot, holott kellett volna. Vegyük például a Wordöt, itt az új dokumentum létrehozását menüből is el tudjuk érni, illetve általában a gyorsindító ikonok között is szerepel. Ha a menüben szereplő funkció fölé vittem az egeret, akkor kiadta a rá jellemző hangot, ha pedig a gyorsindító gombnál akartam ugyanezt elérni, akkor nem történt semmi.

Miután végignéztem a programot, észrevettem, hogy nem egy adott funkció lokalizálásakor játssza le a hangot a szoftver, hanem a funkció felirata alapján. Az objektumokhoz tartoznak különböző tulajdonságok is. Ilyen lehet a név. Ezt a nevet láthatjuk megjelenni a menüben, vagy a gyorsindító gombok esetében kis várakozás után egy sárga színű téglalapban a gomb közelében. Az objektum nevét tartalmazó, megjelenő kis téglalapot a 7. *ábrán* láthatjuk. Így egy adott funkcióhoz több felirat is tartozhat.

Például az új dokumentum létrehozásához tartozó feliratok lehetnek:

- Új
- Új dokumentum...
- Új üres dokumentum.



7. ábra. A megjelenő téglalap

Nemcsak a példában említett funkciónál, hanem többi esetén is hozzárendeltem a hangot még néhány gyakran használt felirathoz. Így a különböző programoknál, ha más az azonos gombokhoz vagy menüelemekhez tartozó felirat, abban az esetben is halljuk a hanghatást. Ennek megfelelően, ha bármely programban megtalálható a funkció, és szerepel a segítő programban a hozzá tartozó felirat, ekkor szintén hallataszik a hang. Hogy mely ikonokhoz és funkciókhoz rendeltem hanghatásokat, azt feljebb, a 2.3.4. pontban ismertettem.

# 4. Egyszerű grafikus felület létrehozása

A kinézet megtervezésénél azt vettem fő szempontnak, hogy a célcsoport számára nem a szépség a fontos, hanem a rendezettség. Ezért gondoltam, hogy nem fektetek nagy hangsúlyt a látványos kinézetre, hanem inkább az egyszerű, világos felületre, amit a látók első ránézésre értelemszerűen tudnak használni, a látássérültek pedig a felolvasóprogramjuk révén egyértelmű tájékoztatást kapnak a program használatáról.

# 4.1. Kezdőoldal

A kezdőoldal eredetileg nem sok információt tartalmazott. Ahhoz, hogy eljussunk a menübe, át kellett méretezni az ablakot. Ez tulajdonképpen még a látó embereknek is okozhat nehézséget, sőt bosszantónak is tarthatják. A vakok többsége pedig nem tudja elvégezni ezt a műveletet, hiszen ebben nincs segítségükre a felolvasóprogram. Ezt a képernyőállást a 8. *ábra* tartalmazza.



8. ábra. Régi kezdőoldal

Az átméretezés után látszódott ugyan a menü, viszont még így sem éreztem elég informatívnak, mert a két menüponton kívül nem tartalmazott mást az indulóképernyő. Az átméretezett kezdőoldalt a 9. *ábra* mutatja.



9. ábra. Az átméretezett régi kezdőoldal

A következő két változtatást vittem véghez a kezdőképernyőn:

- tájékoztató szöveggel láttam el és
- már induláskor olyan méretet használtam, hogy látszódjon a menü és a tájékoztató szöveg.

A tájékoztató szöveg a következő: 'Az információhoz lépjen be a 'Help' menü 'About' almenüpontjába!' A két menüpontot meghagytam az almenüpontokkal együtt. Az ablak méretét pedig úgy adtam meg, hogy már induláskor olyan nagyságú legyen, hogy kiférjen a tájékoztató szöveg. A szoftver eredeti ikonját meghagytam. Az új kezdőképernyőt a 10. ábrán tekinthetjük meg.



10. ábra. Új kezdőképernyő

A régi kezdőképernyő hátrányai:

- nem látszódott a menü az átméretezés gondot okoz,
- nem tartalmazott elég információt a felhasználónak magának kell kitalálnia, hogy melyik menübe lépjen be.

Az új kezdőképernyő előnyei:

- látszanak a menüpontok,
- tartalmaz információt a használatra vonatkozóan.

## 4.2. Menü

A menüpontokon nem változtattam, nem adtam hozzá és nem vettem el. Két főmenüpont van: *Options* és *Help*. Ezek a *10. ábrán* láthatók. Az *Options* menüben két almenüpont található: *Settings* és *Exit*, a *Help* menüben pedig egy: *About*... Ezeket a *11.* és *12. ábrán* tekinthetjük meg.

Eredetileg a *Settings* menüpontra való kattintáskor előjött egy ablak, ami két beállítást tartalmazott. Az egyik az volt, hogy induljon-e automatikusan a program az operációs rendszer betöltésekor, a másik pedig, hogy az összes hangot (egyszerre) kikapcsoljuk-e. Mivel egyik beállítás sem volt implementálva a programban, ezért úgy gondoltam, hogy megváltoztatom ennek az ablaknak a tartalmát. A jelenlegi állapot, ha a *Settings* azaz beállítások menüpontra kattintunk, a felbukkanó ablakban láthatjuk a listát, ami a meghangosított ikonokat és funkciókat tartalmazza. Előtte kis jelölőnégyzetek találhatók, melyekkel majd a későbbi fejlesztések után kiválaszthatjuk, hogy mihez szeretnénk hanghatást rendelni. Ezt az ablakot a *13. ábrán* láthatjuk.

Blindsoft	
Options Help	
Settings Exit	lépjen be a 'Help' menü
'About' alme	nüpontjába!

11. ábra. Options menüpont

)ptions	Help	
Az info	About	a 'Help' menü

12. ábra. *Help* menüpont

/álassza ki, hogy mely ikonokhoz szeretne nanghatást rendelni!	Válassza ki, hogy mely funkcióhoz szeretne hanghatást rendelni!
🗖 Sajátgép	🔽 Megnyitás
🔽 Dokumentumok	🔽 Mentés
T Microsoft Word	🔲 Mentés másként
☑ Internet Explorer	🖵 Nyomtatás
	🗔 Kivágás
	🔽 Másolás
	🔲 Beillesztés
	🔲 Törlés
	Γ Új
	🔲 Kilépés
	🗖 Start

13. ábra. Settings ablak

Az *Options* menü másik almenüpontja az *Exit*. A számítógép-felhasználók körében ez a funkció általában egyértelmű: ezzel tudjuk bezárni a programot. A menüpontra kattintva előjön egy nyugtázó ablak (*14. ábra.*), itt még meggondolhatjuk magunkat, hogy valóban bezárjuk-e a programot.

Do you really want to exit?	
	OK
	Cancel

#### 14. ábra. Exit ablak

A *Help* menüpont *About*... almenüpontjában szintén megjelenik egy ablak, akárcsak a *Settings* almenüpont esetén. Itt egy rövid információt találunk a programról, annak használatáról. Szerepel rajta a program neve és verziószáma, illetve az eddigi fejlesztők listája. Az ablak eddig az előző verziószámot és a két fejlesztő nevét tartalmazta. Ezt az ablakot a *15. ábrán* nézhetjük meg.



15. ábra. About... ablak

Összevetve a régi és az új menüpontokat, a következőket állapíthatjuk meg:

- a Settings menü előjövő ablakánál eltüntettem az eddig nem működő részeket és kiegészítettem egy tervvel a jövőbeni fejlesztésre nézve
- az About... menü előjövő ablakán frissítettem a verziószámot, a fejlesztők listáját, és kiegészítettem egy rövid tájékoztatóval, ami a program használatára vonatkozik.

# 5. Tervezési dokumentáció

A feladathoz tartozó fájlokat a konzulensem bocsátotta rendelkezésemre. A hangfájlokat, a lefordított programot, a programdokumentációt, és a forráskódot. Mindent szabadon felhasználhattam munkámhoz. A szoftver futtatásához szükséges Windows XP vagy későbbi operációs rendszer.

### 5.1. Hangok

A hangminták *wav* formátumban vannak. A *wav (Waveform Audio File Format)* az audio fájlok egy formátuma, amelyet a *Microsoft* és az *IBM* közösen fejlesztett ki digitális hangminták tárolására. Az audio adatokat általában nem tömöríti, veszteségmentesen tárolja. Viszonylag nagy a mérete a többi audio formátumú fájlokhoz képest. A feladatban szereplő hangok hossza 0,6 és 1,7 másodperc között van. A hangfájlokat fölvették vagy az Internetről töltötték le, és Adobe Audition programmal szerkesztették 16 bit, 44100 Hz mono wave formátumban.

Három csoportra oszthatjuk őket: earconok, spearconok és hangzó emoticonok. Az első csoport tagjai nonverbális hangok, melyek az adott objektumra jellemzőek. Könnyen megfejthető, hogy mihez tartozik és könnyen megtanulható. Ilyen a kivágás funkció, amihez ollócsattogtatás tartozik. A spearconok olyan hangok, amelyek speciális technikával felgyorsított beszédminták. Egy szót vagy rövid mondatot hallunk felgyorsítva, angol nyelven. Például a Microsoft Word-höz is ilyen hangminta tartozik, a Word szót halljuk felgyorsítva. A hangzó emoticonok nonverbálisak, akárcsak az earconok. Érzelmi töltetű emberi hangminták, valamilyen hangulatot akarnak kifejezni. Amikor írásban egy mosolyt fejezünk ki, akkor erre a karaktersorozatra (kettőspont és zárójel bezárva ©) főkuszálva egy szolid nevetést hallunk. Ezek a minták nem kerültek bele a programba. A három csoporton kívül megkülönböztetjük még a jelentéssel nem bíró hangok csoportját. Ilyenek például az operációs rendszer figyelmeztetései, a be- és kikapcsolás hangja, stb. A meghangosított ikonokat és funkciókat, a hozzá tartozó hangokat és a hangfájlok nevét az *1. táblázat* tartalmazza.

1. táblázat. Funk	cciók és	hangok
-------------------	----------	--------

Alkalmazás/Funkció	Hang	Hangfájl
Sajátgép	Számítógép bekapcsolása	My Computer.wav
Dokumentumok	Spearcon	S_MyDocs.WAV
Microsoft Word	Spearcon	S_Word.WAV
Internet Explorer	Kulcscsörgés, ajtónyitás	Browser1.wav
Megnyitás	Zipzár húzás	Zip_up.wav
Mentés	Csörgés	Save.wav
Mentés másként	Csörgés és egy hümmögés	Save_as.wav
Nyomtatás	Nyomtató hangja	Print.wav
Kivágás	Vágás ollóval	Cut.wav
Másolás	Fénymásoló hangja	Copy_(loop).wav
Beillesztés	Festés, fütyülés	Paste.wav
Törlés	WC lehúzás	Delete.wav
Új	spearcon	S_New.WAV
Kilépés	Ablakbecsukás	Window_close.wav
Start	spearcon	S_StartMenu.WAV

A hangok egyszer játszódnak le egy esemény bekövetkeztekor. Feladatuk, hogy közöljék a felhasználóval, hogy milyen objektumot lokalizáltak az egérrel. [8] [9]

# 5.2. Keretrendszer

A programot a Microsoft Visual Studioban fejlesztettem C++ nyelven. Az operációs rendszer objektumai és a kliens program közötti kommunikációt az MSAA (Microsoft Active Accessibility) biztosította, mely egy *komponens alapú fejlesztő technológia (COM)*. Az MSAA Windowsba beépített rendszer, amely információt kér a szervertől a kliens számára. Szerver az az alkalmazás, amely az információt szolgáltatja, kliens pedig, aki a kérést intézi.

Az MSAA a következő komponenseket tartalmazza:

- **IAccessible** interfész: információkat szolgáltat a felhasználói felület (UI) elemeiről. Ezek az információk:

- accChild: az objektum leszármazottjának interfésze; minden objektum rendelkezik ezzel a tulajdonsággal.
- accChildCount: az objektum leszármazottjainak száma; minden objektum rendelkezik ezzel a tulajdonsággal.
- **accDefaultAction**: sztring, mely tartalmazza az objektumhoz tartozó alapvető eseményt; nem minden objektumnak van alapértelmezett eseménye.
- accDescription: sztring, mely leírja az objektum vizuális megjelenését; nem mindegyik objektumhoz tartozik leírás; az MSAA-nak nem szokta használni ezt a tulajdonságot.
- accFocus: megadja, hogy objektumot kijelöltük-e a billentyűzet segítségével; azoknak az objektumoknak van ilyen tulajdonsága, amelyek fogadják a billentyűzettől a kijelölést.
- o **accHelp**: súgó sztring; nem minden objektum tartalmazza.
- accHelpTopic: az adott objektum súgófájljának teljes elérési útja és a témaazonosító; nem minden objektum tartalmazza.
- accKeyboardShortcut: az objektumhoz tartozó gyorsbillentyű; amelyik objektumhoz tartozik gyorsbillentyű, annak van ilyen tulajdonsága.
- o **accName**: az objektum neve; az összes objektumnak van ilyen tulajdonsága.
- accParent: az objektum szülőjének interfésze; minden objektum rendelkezik a tulajdonsággal.
- accRole: olyan információ, mely leírja az adott objektum szerepét; minden objektum rendelkezik ezzel a tulajdonsággal.
- accSelection: megmutatja, hogy mely leszármazottai vannak kijelölve az objektumnak; amelyik objektum engedélyezi a kijelölést, annak van ilyen tulajdonsága.
- accState: az objektum aktuális állapotát jelenti; mindegyik objektum rendelkezik ezzel a tulajdonsággal.
- accValue: az objektum értékét mutatja meg (például egy szövegdoboz szövege); nem minden objektumnak van ez a tulajdonsága.

- WinEvents: lehetővé teszi a szervereknek, hogy jelezzék a kliensek felé, ha megváltozott valamelyik objektum. A változások a következők lehetnek:
  - Ha létrejön vagy megszűnik egy objektum.
  - Ha az objektumon állunk vagy elhaladunk fölötte.
  - Ha a státusza vagy helye megváltozik egy objektumnak.
  - Ha egy objektum tulajdonsága megváltozik.

A kliens alkalmazás nem kapja meg automatikusan azt az információt, hogy milyen változás történt, hanem meg kell hívnia a **SetWinEventHook** függvényt, és regisztrálnia kell, hogy mely változásokról szeretne tudni. A klienseket úgy értesíti a szerver, hogy meghívja a **NotifyWinEvent** függvényt, ami megnézi, hogy az adott esemény regisztrálva van-e valamelyik kliens program számára. Ha igen, akkor elindítja az esemény számára létrehozott műveletet.

- Oleacc.dll: függvényeket tartalmaz az MSAA számára. Ezek a függvények lehetnek a következők:
  - o Függvények, melyek lehetővé teszik a kliensek számára a kérést.
  - o Függvények, melyek lehetővé teszik a szerverek számára a választ.
  - Olyan függvények, melyek szöveget lokalizálnak az objektum szerepére és státuszára vonatkozóan.
  - Segédfüggvények.

```
Például: AccessibleObjectFromWindow, LresultFromObject,
GetRoleText, AccessibleChildren.[10]
```

# 5.3. Változók és konstansok

- BSTR a1001-a1025: sztringek, melyek definiálják a különböző objektumok neveit.
   Például ilyen az "Új dokumentum".
- const char g_szClassName[] = "myWindowClass": a program indulásakor létrejövő ablak neve.
- HWINEVENTHOOK g_hook: olyan változó, melynek megadjuk a figyelt eseményeket, például hogy az objektumon állunk-e, vagy kijelölték-e az adott objektumot.

- BSTR name = NULL: ebben a sztring típusú változóban tárolódik az aktuális objektum neve.
- EVENT_OBJECT_FOCUS: konstans annak kifejezésére, hogy az objektumot kijelöltük-e a billentyűzet segítségével.
- EVENT_OBJECT_SHOW: konstans annak kifejezésére, hogy az objektumon állunk-e éppen.

A két konstanst azért emeltem ki, mert ezt a két eseményt figyeli a program, és ezeknek hatására játssza le a megfelelő hanghatást.

# 5.4. Főbb metódusok

- void InitializeMSAA(): az MSAA elindítása, itt lehet beállítani azokat, hogy mik történjenek az induláskor, például itt állíthatjuk be a setWinEventHook függvény segítségével, hogy milyen eseményeket szeretnénk figyelni. Kötelező minden MSAA-t használó programban megadni.
- void ShutdownMSAA(): az MSAA leállítása, ennek is szerepelnie kell a programban.
   Itt lehet leállítani az események figyelését.
- void CALLBACK HandleWinEvent (HWINEVENTHOOK hook, DWORD event, HWND hwnd, LONG idObject, LONG idChild, DWORD dwEventThread, DWORD dwmsEventTime): ebben a metódusban meg tudjuk adni, hogy ha a figyelt esemény lejátszódik, mi történjen. Ebben a programban két eseményt kell figyelni: EVENT_OBJECT_FOCUS és EVENT_OBJECT_SHOW. Ha ilyen esemény történik, és az aktuális objektum éppen közte van a figyelt objektumoknak, akkor a megfelelő hang lejátszása történik.

# 5.5. Felhasználói felület elemei

A felhasználói felület egy kezdőablakból (10. ábra.) és három párbeszédablakból áll (13. ábra.) (14. ábra.) (15. ábra.). A kezdőablak a program indulásakor jön létre, a párbeszédablakok pedig már készen vannak. A kezdőablak tulajdonságainak megadását és létrehozását az int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow); metódusban tehetjük meg.

Az ablak tulajdonságai:

- wc.cbSize = sizeof(WNDCLASSEX);
- wc.style = 0;
- wc.lpfnWndProc = WndProc;
- wc.cbClsExtra = 0;
- wc.cbWndExtra = 0;
- wc.hInstance = hInstance;
- wc.hlcon = LoadIcon(NULL, MAKEINTRESOURCE(IDI ICON1));
- wc.hCursor = LoadCursor(NULL, IDC ARROW);
- wc.hbrBackground = (HBRUSH) (COLOR WINDOW+1);
- wc.lpszMenuName = MAKEINTRESOURCE(IDR MYMENU);
- wc.lpszClassName = g_szClassName;
- wc.hlconSm = (HICON)LoadImage(GetModuleHandle(NULL), MAKEINTRESOURCE(IDI ICON1), IMAGE ICON, 16, 16, 0);

Ezek az ablak alapvető tulajdonságai, például az alapértelmezett mérete, kinézete, ikonja, hozzá tartozó menü azonosítója, stb.

A LRESULT CALLBACK WndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM 1Param) metódus meghatározza, hogy mi legyen az ablakon és hogy mi történjen az ablak elemeinek aktiválódására. Az útmutató szöveget az utmutato1[] és utmutato2[] karaktertömb tartalmazza. A menü azonosítója: IDR_MYMENU; két menüpontból áll, az Options menüpontnak két almenüpontja van: Settings (ID_SETTING) és Exit (ID_FILE_EXIT). A Help menüpontnak egy almenüpontja van: About... (ID_HELP ABOUT).

Mindegyik almenüponthoz tartozik egy-egy párbeszédablak:

- IDD_DIALOG1: a Help menü About... menüpontjának párbeszédablaka. Ezen az ablakon négy szövegdoboz (IDC_STATIC1, IDC_STATIC2, IDC_STATIC3, IDC STATIC4) és egy nyomógomb (IDOK) helyezkedik el.
- IDD_DIALOG2: az Options menü Settings menüpontjának párbeszédablaka. Ezen az ablakon két szövegdoboz (IDC STATIC1, IDC STATIC2), 15

jelölőnégyzet (IDC_CHECK1-IDC_CHECK15) és egy nyomógomb (IDOK) helyezkedik el.

 ID_EXIT: az Options menü Exit almenüpontjának párbeszédablaka. Egy groupbox (IDC_STATIC1) és két nyomógomb (IDOK, IDCANCEL) helyezkedik el rajta.

# 5.6. Az eredeti és az új szoftver különbségei

#### 5.6.1. Hangok

Abban a változatban, amit megkaptam, sokkal több hang implementálva volt a programba, viszont ezek közül nagyon sok nem volt kihasználva, vagy csak nagyon ritkán, vagy pedig a célcsoport nem tartotta szükségesnek. Ezek a funkciók az *Internet Böngésző* elemeihez kapcsolódó funkciók, például *Előre, Vissza, Kezdőlap*; a *Word* egy-egy funkciója, például *Nyelvhelyesség-ellenőrzés, Keresés, Az összes kijelölése, E-mail*; a *Lomtár ikonja*, és a *Lomtár ürítése* funkció és még sok egyéb. A jelenlegi programban azok a hangok szerepelnek, amik a vélemények alapján a legfontosabbak. Néhány hangot több felirathoz is hozzárendeltem.

#### 5.6.2. Változók

A változók közül az előző verzióban szerepelt egy integer típusú Objectname azonosítójú változó, valamint négy long típusú változó:

pxLeft: az elem bal-felső sarkának x koordinátája

pyTop: az elem bal-felső sarkának y koordinátája

pcxWidth: az elem saját szélessége

pcyHeight: az elem saját magassága.

Az utóbbi négy változó azt a célt szolgálta, hogy visszaadja az épp vizsgált objektum helyzetét. Az Objectname változó pedig megkapta az aktuális objektum nevét. Ez a név akkor került bele a változóba, amikor bekövetkezett az EVENT_OBJECT_FOCUS esemény, és meg is szólalt a felismert eseményhez tartozó hang. Így egyelőre nem láttam értelmét, hogy a változóval végzett további műveleteken hatására újra megszólaljon az objektumhoz tartozó hanghatás. Az új változatban nem szerepelnek ezek a változók.

#### 5.6.3. Felhasználói felület

Az eredeti szoftverben a kezdőoldal kisebb volt, nem volt rajta a tájékoztató szöveg. Az Settings ablak (IDD_DIALOG2) két jelölőnégyzetet (checkbox) tartalmazott 'Sounds off' és 'Run at Windows startup' felirattal. A jelenlegi ablak a 13. ábrán látható jelölőnégyzeteket tartalmazza két oszlopban, felettük egy-egy szövegdoboz. Az About... ablak tartalmaz négy szövegdobozt, melyeken szerepel a program verziószáma, az eddigi fejlesztők és a tájékoztató szöveg a program használatáról. Az eredeti párbeszédablakon a régi verziószám és a fejlesztők szerepeltek. Összességében a felhasználói felület átláthatóbb, egyszerűen kezelhetőbb lett.

# 6. Felhasználói dokumentáció

### 6.1. Rendszerkövetelmények

A szoftver működéséhez feltétlenül szükséges Windows XP vagy későbbi Windows operációs rendszer. Az optimális működés érdekében javasolt a Windows XP használata, valamint a Microsoft Office csomag 2003-as változatát.

### 6.2. Telepítési útmutató

- lépés: Másolja fel egyazon tetszőleges mappába a Blindsoft.exe fájlt és a Sounds mappát (ez utóbbi tartalmazza a hangfájlokat).
- lépés: Kattintson duplán a Blindsoft.exe fájlra, és máris működésbe lép a program.

A szoftver adathordozóról is működtethető (Például pendrive, külső merevlemez.).

### 6.3. A szoftver használata

Először hajtsa végre a *Telepítési útmutató*ban leírtakat! Amint működésbe lép a program, a *10. ábrán* látható kezdőképernyő fogadja. A képernyőn elolvashatja a rövid tájékoztató szöveget a program használatához.

Menüpontok:

- Options: két almenüpontot tartalmaz (11. ábra.):
  - Settings: erre a menüpontra kattintva előjön egy ablak, melyet a 13. ábrán tekinthet meg. A fejléc tartalmazza az ablak nevét (Settings). Az ablakban láthatja a meghangosított elemek listáját.
  - *Exit*: ezzel a menüponttal tudja bezárni a programot. Kattintás után előjön egy ablak, ahol angol nyelven rákérdez a program, hogy valóban szeretnénk-e kilépni. Ha tényleg szeretne kilépni, kattintson az *OK* gombra, ha nem, kattintson a *Cancel* gombra! (*14. ábra*)

- *Help*: egy almenüpontot tartalmaz (*12. ábra*.):
  - About...: erre a menüpontra kattintva szintén egy ablak jön elő, melyet a 15. ábrán láthat. A fejléc neve: About... Ebben az ablakban megtekintheti a program verziószámát, az eddigi fejlesztők nevét, és a rövid tájékoztatót a szoftver működéséről.

Mozgassa az egeret az *1. táblázat*ban felsorolt funkciók fölé, és hallani fogja a hozzájuk rendelt hanghatást. A legtöbb esetben az adott funkcióra jellemző hang hallatszik. Például kivágásnál vágás ollóval, nyomtatásnál a nyomtató hangja. A működési idő nagyon változó, átlagban néhány perc, ettől lehet eltérő. Ha a program működése közben hibaüzenet jelenik meg, zárja be a programot, majd indítsa újra!

# 6.4. Szolgáltatások

A program alapvető szolgáltatása, hogy segítséget nyújt a látássérült embereknek a számítógép használatához. Ezt úgy viszi véghez, hogy ha az egérrel megállunk egy-egy menüelem, gyorsindítógomb, ikon felett, akkor valamilyen rövid hanghatást hallat. Ez lehet az elemre jellemző hanghatás, vagy az elem neve gyorsan kimondva angol nyelven. Egy menüelemet, egy gyorsindító gombot és egy ikont láthatunk a *16. ábrá*n.



16. ábra. Menüelem, gyorsindító gomb, ikon

A szoftvert bátran használhatják látók is, mindenki, akinek szüksége van apró visszajelzésekre munkája nyugtázásául.

# 7. Irodalomjegyzék

- [1] Bankóné Virág, I.: A magyar könyvtárpolitika, a könyvtári rendszer, és a könyvtárak online szolgáltatásainak szerepe a látássérültek esélyegyenlőségének megvalósulásában. http://mek.oszk.hu/06100/06100/index.phtml/, 2008. Szakdolgozat
- [2] http://www.ki-latas.hu/fogalomtar/
- [3] Braille: http://www.omniglot.com/writing/braille.htm/
- [4] The Life and Legacy of Louis Braille: http://www.afb.org/louisbraillemuseum/
- [5] Rouse, M.: speech synthesis http://whatis.techtarget.com/definition/speech-synthesis/, 2005.
- [6] Fábri, T. és Szatmári, P: A látássérült emberek számítástechnikai lehetőségei és eszközei
   Budapest, 2007.
- [7] NVDA: http://www.nvda.hu/
- [8] Wersényi, Gy.: Auditory Representations of a Graphical User Interface for a Better Human-Computer Interaction.
   in S. Ystad et al. (Eds.): Auditory Display. CMMR/ICAD 2009 post proceedings edition, Lecture Notes in Computer Science (LNCS) 5954, Springer Verlag, Berlin, 2010. /80-102/
- [9] Körtélyesi, K. és Szőts, V.: *Eseményhangok vakok számára* 2009.
- [10] Microsoft Active Accessibility: http://msdn.microsoft.com/en-us/library/dd373592/

# 8. Melléklet

### 8.1. Forráskód

//example1.cpp

//Szukseges fejfajlok //#pragma comment(lib, "winmm.lib") #include "windows.h" #include "resource.h" #include "stdio.h" #include <cstdio> #include "WINUSER.H" //#include "WINABLE.H" #include "OLEACC.H" #include <comdef.h> #include <mmsystem.h> #include <tchar.h> #include <iostream> #include <sstream> //Ezek a valtozok tartalmazzak a figyelt neveket. BSTR a1001 = SysAllocString(L"Sajátgép"); bstr t a1001 bstr(a1001); BSTR a1002 = SysAllocString(L"Dokumentumok"); _bstr_t a1002_bstr(a1002); BSTR a1003 = SysAllocString(L"Microsoft Word"); _bstr_t a1003_bstr(a1003); BSTR a1004 = SysAllocString(L"Microsoft Office Word 2003"); _bstr_t a1004_bstr(a1004); BSTR a1005 = SysAllocString(L"Internet Explorer"); bstr t a1005 bstr(a1005); BSTR a1006 = SysAllocString(L"Megnyitás"); bstr t a1006 bstr(a1006); BSTR a1007 = SysAllocString(L"Megnyitás..."); _bstr_t a1007_bstr(a1007); BSTR a1008 = SysAllocString(L"Megnyitás...\tCtrl+0"); _bstr_t a1008_bstr(a1008); BSTR a1009 = SysAllocString(L"Mentés"); _bstr_t a1009_bstr(a1009); BSTR a1010 = SysAllocString(L"Mentés Ctrl+S"); _bstr_t a1010_bstr(a1010); BSTR a1011 = SysAllocString(L"Mentés másként..."); bstr t a1011 bstr(a1011); BSTR a1012 = SysAllocString(L"Nyomtatás..."); _bstr_t a1012_bstr(a1012); BSTR a1013 = SysAllocString(L"Nyomtatás...\tCtrl+P "); bstr_t a1013_bstr(a1013); BSTR a1014 = SysAllocString(L"Kivágás"); bstr t a1014 bstr(a1014); BSTR a1015 = SysAllocString(L"Kivágás\tCtrl+X"); bstr t a1015 bstr(a1015); BSTR a1016 = SysAllocString(L"Másolás"); bstr t a1016 bstr(a1016); BSTR a1017 = SysAllocString(L"Másolás\tCtrl+C"); bstr_t a1017_bstr(a1017); BSTR a1018 = SysAllocString(L"Beillesztés"); _bstr_t a1018_bstr(a1018); BSTR a1019 = SysAllocString(L"Beillesztés\tCtrl+V"); bstr t a1019 bstr(a1019); BSTR a1020 = SysAllocString(L"Törlés"); _bstr_t a1020_bstr(a1020); BSTR a1021 = SysAllocString(L"Új dokumentum..."); bstr t a1021 bstr(a1021); BSTR a1022 = SysAllocString(L"Új üres dokumentum"); _bstr_t a1022_bstr(a1022); BSTR a1023 = SysAllocString(L"Kilépés"); _bstr_t a1023_bstr(a1023); BSTR a1024 = SysAllocString(L"Start"); _bstr_t a1024_bstr(a1024); BSTR a1025 = SysAllocString(L"Az indításhoz kattintson ide."); bstr t a1025 bstr(a1025);

const char g_szClassName[] = "myWindowClass"; //Ez az ablak osztaly neve. HWINEVENTHOOK g_hook; //Ebben a valtozoban taroljuk a figyelt esemenyeket. BSTR name = NULL; //Az aktualis objektum neve.

```
//Itt kapcsoljuk ki az MSAA szolgaltatasait.
void ShutdownMSAA()
{
   UnhookWinEvent(g hook);
    CoUninitialize();
}
//Itt allithatjuk be, hogy mi tortenjen a fogadott esemenyek hatasara.
void CALLBACK HandleWinEvent(HWINEVENTHOOK hook, DWORD event, HWND hwnd,
                             LONG idObject, LONG idChild,
                             DWORD dwEventThread, DWORD dwmsEventTime)
{
    IAccessible* pAcc = NULL;
    VARIANT varChild;
       VariantInit(&varChild);
    HRESULT hr = AccessibleObjectFromEvent(hwnd, idObject, idChild, &pAcc, &varChild);
    if ((hr == S OK) && (pAcc != NULL))
    {
                if (pAcc->get accName(varChild, &name) != S FALSE) //Kinyerjük az objektum nevet.
        {
                       bstr t name bstr(name); //Ebbe a valtozoba rakjuk az objektum nevet.
                switch (event) //Esemeny tipusa.
                      {
                       case EVENT OBJECT FOCUS: //Elem kijelolese (billentyuzet).
                       {
if (a1001 bstr==name bstr)PlaySound("Sounds\\My Computer.wav", NULL, SND ASYNC |
SND FILENAME); //sajatgep
if (a1002 bstr==name bstr)PlaySound("Sounds\\S MyDocs.WAV", NULL, SND ASYNC | SND FILENAME);
// dokumentumok
if (a1003 bstr==name bstr)PlaySound ("Sounds\\S Word.WAV", NULL, SND ASYNC | SND FILENAME);
//word
if (a1004 bstr==name bstr)PlaySound ("Sounds\\S Word.WAV", NULL, SND ASYNC | SND FILENAME);
//word
if (a1005 bstr==name bstr)PlaySound ("Sounds\\Browser1.wav", NULL, SND ASYNC | SND FILENAME);
//internet explorer
if (a1006 bstr==name bstr)PlaySound ("Sounds\\Zip up.wav", NULL, SND ASYNC | SND FILENAME);
//megnvitas
if (a1007 bstr==name bstr)PlaySound ("Sounds\\Zip up.wav", NULL, SND ASYNC | SND FILENAME);
//megnyitas
if(a1008_bstr==name_bstr)PlaySound("Sounds\\Zip_up.wav", NULL, SND_ASYNC | SND_FILENAME);
//megnvitas
if (a1009 bstr==name bstr) PlaySound ("Sounds\\Save.wav", NULL, SND ASYNC | SND FILENAME);
//mentes
if (a1010 bstr==name bstr)PlaySound ("Sounds\\Save.wav", NULL, SND ASYNC | SND FILENAME);
//mentes
if(a1011_bstr==name_bstr)PlaySound("Sounds\\Save_as.wav", NULL, SND_ASYNC | SND_FILENAME);
//mentes maskent
if (a1012 bstr==name bstr)PlaySound("Sounds\\Print.wav", NULL, SND ASYNC | SND FILENAME);
//nyomtatas
if (a1013 bstr==name bstr)PlaySound("Sounds\\Print.wav", NULL, SND ASYNC | SND FILENAME);
//nyomtatas
if (a1014 bstr==name bstr)PlaySound ("Sounds\\Cut.wav", NULL, SND ASYNC | SND FILENAME);
//kivagas
```

if (a1015 bstr==name bstr) PlaySound ("Sounds\\Cut.wav", NULL, SND ASYNC | SND FILENAME); //kivagas if (a1016 bstr==name bstr)PlaySound ("Sounds\\Copy (loop).wav", NULL, SND ASYNC | SND FILENAME); //masolas if (a1017 bstr==name bstr)PlaySound ("Sounds\\Copy (loop).wav", NULL, SND ASYNC | SND_FILENAME); //masolas if (a1018 bstr==name bstr)PlaySound("Sounds\\Paste.wav", NULL, SND ASYNC | SND FILENAME); //beillesztes if (a1019_bstr==name_bstr)PlaySound ("Sounds\\Paste.wav", NULL, SND_ASYNC | SND_FILENAME); //beillesztes if (a1020 bstr==name bstr)PlaySound("Sounds\\Delete.wav", NULL, SND ASYNC | SND FILENAME); //torles if(a1021 bstr==name bstr)PlaySound("Sounds\\S New.WAV", NULL, SND ASYNC | SND FILENAME); //uj if (a1022 bstr==name bstr)PlaySound ("Sounds\\S New.WAV", NULL, SND ASYNC | SND FILENAME); //ui if (a1023 bstr==name bstr)PlaySound ("Sounds\\Window close.wav", NULL, SND ASYNC | SND FILENAME); //kilepes if (a1024 bstr==name bstr)PlaySound("Sounds\\S StartMenu.WAV", NULL, SND ASYNC | SND FILENAME); //start menu if (a1025 bstr==name bstr)PlaySound("Sounds\\S StartMenu.WAV", NULL, SND ASYNC | SND FILENAME); //start menu

#### } break:

case EVENT_OBJECT_SHOW: //Elem kozeleben megjelenik a nev.
{

if (a1001 bstr==name bstr)PlaySound("Sounds\\My Computer.wav", NULL, SND ASYNC | SND FILENAME); //sajatgep if(a1002_bstr==name_bstr)PlaySound("Sounds\\S_MyDocs.WAV", NULL, SND_ASYNC | SND_FILENAME); // dokumentumok if (a1003 bstr==name bstr)PlaySound("Sounds\\S Word.WAV", NULL, SND ASYNC | SND FILENAME); //word if (a1004 bstr==name bstr)PlaySound("Sounds\\S Word.WAV", NULL, SND ASYNC | SND FILENAME); //word if (a1005 bstr==name bstr)PlaySound("Sounds\\Browser1.wav", NULL, SND ASYNC | SND FILENAME); //internet explorer if (a1006 bstr==name bstr)PlaySound ("Sounds\\Zip up.wav", NULL, SND ASYNC | SND FILENAME); //megnvitas if (a1007 bstr==name bstr)PlaySound("Sounds\\Zip up.wav", NULL, SND ASYNC | SND FILENAME); //megnyitas if (a1008 bstr==name bstr)PlaySound("Sounds\\Zip up.wav", NULL, SND ASYNC | SND FILENAME); //megnvitas if (a1009 bstr==name bstr)PlaySound("Sounds\\Save.wav", NULL, SND ASYNC | SND FILENAME); //mentes if (a1010 bstr==name bstr)PlaySound("Sounds\\Save.wav", NULL, SND ASYNC | SND FILENAME); //mentes if (a1011 bstr==name bstr)PlaySound("Sounds\\Save as.wav", NULL, SND ASYNC | SND FILENAME); //mentes maskent if (a1012 bstr==name bstr)PlaySound("Sounds\\Print.wav", NULL, SND ASYNC | SND FILENAME); //nvomtatas if (a1013 bstr==name bstr)PlaySound("Sounds\\Print.wav", NULL, SND ASYNC | SND FILENAME); //nvomtatas if (a1014 bstr==name bstr) PlaySound ("Sounds\\Cut.wav", NULL, SND ASYNC | SND FILENAME); //kivagas if (a1015 bstr==name bstr)PlaySound("Sounds\\Cut.wav", NULL, SND ASYNC | SND FILENAME); //kivagas if (a1016_bstr==name_bstr)PlaySound("Sounds\\Copy_(loop).wav", NULL, SND_ASYNC | SND FILENAME); //masolas if (a1017 bstr==name bstr)PlaySound("Sounds\\Copy (loop).wav", NULL, SND ASYNC | SND FILENAME); //masolas

```
if (a1018 bstr==name bstr)PlaySound("Sounds\\Paste.wav", NULL, SND ASYNC | SND FILENAME);
//beillesztes
if (a1019 bstr==name bstr)PlaySound("Sounds\\Paste.wav", NULL, SND ASYNC | SND FILENAME);
//beillesztes
if (a1020 bstr==name bstr)PlaySound ("Sounds\\Delete.wav", NULL, SND ASYNC | SND FILENAME);
//torles
if(a1021_bstr==name_bstr)PlaySound("Sounds\\S_New.WAV", NULL, SND_ASYNC | SND_FILENAME); //uj
if (a1022 bstr==name bstr)PlaySound ("Sounds\\S_New.WAV", NULL, SND ASYNC | SND FILENAME);
//uj
if (a1023 bstr==name bstr)PlaySound("Sounds\\Window close.wav", NULL, SND ASYNC |
SND FILENAME); //kilepes
if (a1024 bstr==name bstr)PlaySound("Sounds\\S StartMenu.WAV", NULL, SND ASYNC |
SND_FILENAME); //start menu
if (a1025 bstr==name bstr)PlaySound("Sounds\\S StartMenu.WAV", NULL, SND ASYNC |
SND FILENAME); //start menu
                              }
                       break;
                       }
               SysFreeString(name); //Felszabaditjuk a nev valtozot.
                }
    pAcc->Release();
       }
       VariantClear(&varChild);
}
//MSAA inicializalasa, itt regisztralhatjuk be esemenyeket.
       void InitializeMSAA()
{
       CoInitialize(NULL);
       g hook = SetWinEventHook(
        EVENT_SYSTEM_MENUSTART , EVENT_OBJECT_VALUECHANGE ,
        NULL
        HandleWinEvent,
        0, 0,
        WINEVENT OUTOFCONTEXT | WINEVENT SKIPOWNPROCESS);
}
//WINDOW
//A parbeszedablakok kozos tulajdonsagai.
BOOL CALLBACK DialogusPROC(HWND hwnd, UINT Message, WPARAM wParam, LPARAM lPararm)
{
```

```
switch(Message)
{
    case WM_INITDIALOG:
        return TRUE;
    case WM_COMMAND:
        switch(LOWORD(wParam))
    {
        case IDOK:
        EndDialog(hwnd, IDOK);
            break;
        case IDCANCEL:
        EndDialog(hwnd, IDCANCEL);
    }
}
```

```
break;
            }
                       break;
     break;
                 case WM CLOSE:
                       DestroyWindow(hwnd);
                       break;
     default:
     return FALSE;
       }
       return TRUE;
}
//Az ablakon levo elemek letrehozasa.
LRESULT CALLBACK WndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam)
{
    PAINTSTRUCT ps;
       HDC hdc;
       //Az ablak szovege.
       TCHAR utmutato1[] = _T("Az információhoz lépjen be a 'Help' menü");
TCHAR utmutato2[] = _T("'About...' almenüpontjába!");
       switch(msg)
    {
               //Kirajzoljuk az ablak szoveget.
               case WM PAINT:
                       {
                               hdc = BeginPaint(hwnd, &ps);
                               TextOut(hdc, 15, 15, utmutatol, _tcslen(utmutatol));
                               TextOut(hdc, 15, 45, utmutato2, _tcslen(utmutato2));
                               EndPaint(hwnd, &ps);
                              break;
                       }
               //A menuelemekre valo kattintas.
               case WM COMMAND:
                       {
                               switch(LOWORD(wParam))
                               {
                               case ID FILE EXIT: //Kilepes megjelenik a kilepes ablak.
                                       {
                                       int ret = DialogBox(GetModuleHandle(NULL),
                MAKEINTRESOURCE(ID_EXIT), hwnd, DialogusPROC);
            if(ret == IDOK) {
                               PostMessage(hwnd, WM CLOSE, 0,0);
                               }
                                      }break;
                               case ID STUFF GO:
                                       {
                                       char szHandle[20]="Gone!";
                                       MessageBox(hwnd, szHandle, "GO", MB_OK |
MB ICONINFORMATION);
                                       }break;
                               case ID_SETTING: //Beallitasok_megjelenik a beallitasok ablak.
                                       {
                                       DialogBox(GetModuleHandle(NULL),
MAKEINTRESOURCE(IDD DIALOG2), hwnd, DialogusPROC);
                                       }break;
                               case ID_HELP_ABOUT: //About..._About ablak megnyitasa.
```

```
DialogBox(GetModuleHandle(NULL),
MAKEINTRESOURCE (IDD DIALOG1), hwnd, DialogusPROC);
                               break;
                       } }
                       break;
        case WM CLOSE: //Bezaras.
                       {DestroyWindow(hwnd);
                        ShutdownMSAA();
                        }
                       break;
        case WM DESTROY: //Bezaras.
            PostQuitMessage(0);
        break:
        default:
            return DefWindowProc(hwnd, msg, wParam, lParam);
    }
    return 0;
}
//Az ablak tulajdonsagainak megadasa, es az ablak letrehozasa.
int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPSTR lpCmdLine, int nCmdShow)
{
       WNDCLASSEX wc;
    HWND hwnd;
    MSG Msg;
    //Regisztraljuk az ablak tulajdonsagait.
    wc.cbSize
                     = sizeof(WNDCLASSEX);
    wc.style
                   = 0;
    wc.lpfnWndProc = WndProc;
    wc.cbClsExtra = 0;
wc.cbWndExtra = 0;
                     = 0;
    wc.hInstance = hInstance;
    wc.hIcon = LoadIcon(NULL, MAKEINTRESOURCE(IDI_ICON1));
wc.hCursor = LoadCursor(NULL, IDC_ARROW);
    wc.hbrBackground = (HBRUSH) (COLOR WINDOW+1);
    wc.lpszMenuName = MAKEINTRESOURCE(IDR MYMENU);
    wc.lpszClassName = g_szClassName;
    wc.hlconSm = (HICON)LoadImage(GetModuleHandle(NULL), MAKEINTRESOURCE(IDI_ICON1),
IMAGE_ICON, 16, 16, 0);
    //Ha nem sikerült regisztralni a tulajdonsagokat, akkor hibauzenetet generalunk.
       if(!RegisterClassEx(&wc))
    {
        MessageBox(NULL, "Window Registration Failed!", "Error!",
            MB ICONEXCLAMATION | MB OK);
        return 0;
    }
    //Ablak letrehozasa.
    hwnd = CreateWindowEx(
        WS_EX_CLIENTEDGE,
        g_szClassName,
        "Blindsoft",
        WS OVERLAPPEDWINDOW,
        CW_USEDEFAULT, CW_USEDEFAULT, 340, 140,
        NULL, NULL, hInstance, NULL);
```

```
//Ha nem sikerult az ablakot letrehozni, hibauzenetet kell generalni.
if(hwnd == NULL)
{
   MessageBox(NULL, "Window Creation Failed!", "Error!",
       MB ICONEXCLAMATION | MB OK);
    return 0;
}
ShowWindow (hwnd, nCmdShow); //Az ablakot megjelenitjuk.
UpdateWindow(hwnd);
   InitializeMSAA(); //Ablak keszitese utan inicializaljuk az MSAA-t.
while(GetMessage(&Msg, NULL, 0, 0) > 0)
{
    TranslateMessage(&Msg);
   DispatchMessage(&Msg);
}
return Msg.wParam;
```

# 8.2. A párbeszédablakok elemei, tulajdonságai

}

```
//resource.rc
//Ez a forrásfájl tartalmazza a párbeszédablakok elemeit
//és tulajdonságait.
// Microsoft Visual C++ generated resource script.
11
#include "resource.h"
#define APSTUDIO_READONLY_SYMBOLS
11
// Generated from the TEXTINCLUDE 2 resource.
11
#include "afxres.h"
#undef APSTUDIO READONLY SYMBOLS
// Hungarian (Hungary) resources
#if !defined(AFX RESOURCE DLL) || defined(AFX TARG HUN)
LANGUAGE LANG HUNGARIAN, SUBLANG DEFAULT
#pragma code page(1250)
#ifdef APSTUDIO INVOKED
11
// TEXTINCLUDE
11
1 TEXTINCLUDE
BEGIN
```

```
"resource.h\0"
   END
   2 TEXTINCLUDE
   BEGIN
      "#include ""afxres.h""\r\n"
      "\0"
   END
   3 TEXTINCLUDE
   BEGIN
      "\r\n"
      "\0"
   END
          // APSTUDIO INVOKED
   #endif
   11
   // Icon
   11
   // Icon with lowest ID value placed first to ensure application icon
   // remains consistent on all systems.
   IDI ICON1
                                              "icon2.ico"
                       TCON
   11
   // Dialog
   11
   ID EXIT DIALOG 0, 0, 239, 66
   STYLE DS SETFONT | DS MODALFRAME | WS POPUP | WS CAPTION | WS SYSMENU
   CAPTION "Exit?"
   FONT 8, "MS Sans Serif"
   BEGIN
      DEFPUSHBUTTON "&OK", IDOK, 174, 18, 50, 14
      PUSHBUTTON "&Cancel", IDCANCEL, 174, 35, 50, 14
      GROUPBOX
                    "Do you really want to exit?", IDC_STATIC, 7, 7, 225, 52
   END
   IDD DIALOG1 DIALOGEX 0, 0, 266, 124
   STYLE DS_SETFONT | DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
   CAPTION "About..."
   FONT 8, "MS Sans Serif", 0, 0, 0x0
   BEGIN
      DEFPUSHBUTTON "OK", IDOK, 199, 15, 50, 14
             "Blindsoft 1.1", IDC_STATIC, 17, 15, 40, 8
      LTEXT
      LTEXT
                    "Szőts Viktor Körtélyesi Károly Ignácz Eszter", IDC_STATIC, 60, 29, 58, 27
      CONTROL
                                     "Authors:", IDC STATIC, "Static", SS LEFTNOWORDWRAP
                                                                                   - I
WS GROUP, 18, 27, 40, 10
                     "A program hanghatásokat ad ki a számítógép néhány eleme hatására. Az
     LTEXT
'Options' feliratra kattintva a 'Settings' menüpontban beállíthatja, hogy mely ikonhoz és
funkcióhoz szeretne hanghatást rendelni. Az 'Options' menü 'Exit' opciójával
kiléphet.",IDC STATIC,16,58,185,51
  END
   IDD DIALOG2 DIALOGEX 0, 0, 338, 211
   STYLE DS_SETFONT | DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
   CAPTION "Settings"
   FONT 8, "MS Sans Serif", 0, 0, 0x0
   BEGIN
```

```
44
```

DEFPUSHBUTTON "OK", IDOK, 140, 181, 50, 14 LTEXT "Válassza ki, hogy mely ikonokhoz szeretne hanghatást rendelni!", IDC STATIC, 15, 14, 142, 16 CONTROL "Sajátgép", IDC_CHECK1, "Button", BS_AUTOCHECKBOX Т WS TABSTOP, 19, 35, 44, 10 "Microsoft Word", IDC CHECK3, "Button", BS AUTOCHECKBOX CONTROL WS TABSTOP, 19, 61, 64, 10 CONTROL "Internet Explorer", IDC_CHECK4, "Button", BS_AUTOCHECKBOX T WS_TABSTOP,19,74,67,10 "Megnyitás", IDC CHECK5, "Button", BS AUTOCHECKBOX CONTROL Т WS TABSTOP, 176, 35, 48, 10 "Dokumentumok", IDC CHECK2, "Button", BS AUTOCHECKBOX CONTROL Т WS TABSTOP, 19, 48, 66, 10 "Mentés",IDC_CHECK6,"Button",BS_AUTOCHECKBOX | WS_TABSTOP,176,48,39,10 CONTROL "Mentés másként", IDC CHECK7, "Button", BS AUTOCHECKBOX | CONTROL WS TABSTOP, 176, 61, 68, 10 LTEXT "Válassza ki, hogy mely funkcióhoz szeretne hanghatást rendelni!",IDC_STATIC,171,14,149,16 "Nyomtatás", IDC CHECK8, "Button", BS AUTOCHECKBOX CONTROL Т WS TABSTOP, 176, 74, 49, 10 CONTROL "Kivágás", IDC_CHECK9, "Button", BS_AUTOCHECKBOX T WS_TABSTOP, 176, 87, 41, 10 "Másolás", IDC CHECK10, "Button", BS AUTOCHECKBOX CONTROL T WS_TABSTOP, 176, 99, 42, 10 "Beillesztés", IDC CHECK11, "Button", BS AUTOCHECKBOX CONTROL WS_TABSTOP, 176, 111, 49, 10 CONTROL "Törlés", IDC CHECK12, "Button", BS AUTOCHECKBOX 1 WS TABSTOP, 176, 123, 35, 10 CONTROL "Új", IDC CHECK13, "Button", BS AUTOCHECKBOX | WS TABSTOP, 176, 135, 23, 10 "Kilépés", IDC CHECK14, "Button", BS AUTOCHECKBOX CONTROL 1 WS_TABSTOP, 176, 147, 39, 10 "Start", IDC CHECK15, "Button", BS AUTOCHECKBOX CONTROL T WS_TABSTOP, 176, 160, 31, 10 END

```
11
// Menu
11
IDR MYMENU MENU
BEGIN
  POPUP "&Options"
  BEGIN
    MENUITEM "&Settings",
                             ID SETTING
    MENUITEM "&Exit",
                              ID FILE EXIT
  END
  POPUP "Help"
  BEGIN
    MENUITEM "About...",
                             ID HELP ABOUT
  END
END
11
// DESIGNINFO
11
#ifdef APSTUDIO INVOKED
GUIDELINES DESIGNINFO
BEGIN
```

```
ID_EXIT, DIALOG
  BEGIN
  END
  IDD DIALOG1, DIALOG
  BEGIN
    LEFTMARGIN, 7
    RIGHTMARGIN, 259
    TOPMARGIN, 8
    BOTTOMMARGIN, 117
  END
  IDD DIALOG2, DIALOG
  BEGIN
    LEFTMARGIN, 7
    RIGHTMARGIN, 331
    TOPMARGIN, 7
    BOTTOMMARGIN, 204
  END
END
#endif // APSTUDIO_INVOKED
11
// Cursor
11
IDC CURSOR1
             CURSOR
                            "cursor1.cur"
11
// String Table
11
STRINGTABLE
BEGIN
               "Kilépés"
  ID_FILE_EXIT
END
#endif // Hungarian (Hungary) resources
#ifndef APSTUDIO_INVOKED
11
// Generated from the TEXTINCLUDE 3 resource.
11
```

# 8.3. További fájlok listája

//A program ikonja:

• icon2.ico

//A projekt fájlok:

- example1.dsp
- example1.dsw
- //Az MSAA kapcsolat létrehozásához és működéséhez szükséges:
  - oleaccproxy.cpp
  - oleaccproxy.h
- //A resource.rc fájl által használt fájl:
  - resource.h
  - cursor1.cur

# 8.4. Lemezmelléklet tartalma

A lemezmelléklet (DVD) tartalmazza:

- kiemelve a fent említett fájllistát,
- az összefoglalókat kiemelve, HTML formátumban, két külön fájlban (magyar és angol)
- a teljes szakdolgozatot, számítógépes dokumentumok formájában,
- a forrásprogramokat és egyéb, a telepítéshez szükséges szoftver anyagokat,
- a rövid bemutató anyagát HTML formátumban.