

Tudományos Diákköri Konferencia

Kognitív jellemzők és a szemmozgási paraméterek közötti összefüggések vizsgálata saját fejlesztésű és validált szemmozgáskövető rendszer alkalmazásával

Készítették: Andóczy-Balogh Gábor
másodéves mérnökinformatikus hallgató
Czeilinger Dominik József
másodéves mérnökinformatikus hallgató
Vámos Balázs
negyedéves mérnökinformatikus hallgató

Témavezetők: Dr. habil. Katona József, PhD
egyetemi docens, tanszékvezető
Prof. Dr. Kővári Attila, PhD
egyetemi tanár



DUNAÚJVÁROSI EGYETEM
UNIVERSITY OF DUNAÚJVÁROS

Dunaújváros, 2024.

Tartalomjegyzék

1. Bevezetés	5
1.1. Kutatás feltételei	6
1.2. Célkitűzések	6
1.3. Felhasznált módszerek	8
1.4. A dolgozat felépítése	8
2. Szemmozgáskövetési technológiák és módszerek	10
2.1. Pupilla központú szaruhártya tükröződés (PCCR)	10
2.2. PCCR elvet alkalmazó szemmozgáskövető eszközök	10
2.2.1. Képernyő alapú szemmozgáskövetők	11
2.2.2. Fejre csatolt szemmozgáskövetők	12
2.3. Kamera és szenzor technológia	12
2.4. Kognitív tulajdonságok és szemmozgások közötti általános összefüggések	13
2.4.1. Figyelem és szemmozgás kapcsolata	13
2.4.2. Szemmozgáskövető rendszerek alkalmazási lehetőségeinek elemzése	13
2.5. Statisztikai módszerek kiválasztása a szemmozgás adatainak elemzésére	14
3. Innovatív szemmozgáskövető szoftver fejlesztése	15
3.1. Követelményspecifikáció	15
3.2. Fejlesztői környezet kiválasztása	16
3.3. Verziókövető rendszer alkalmazása	17
3.4. A felhasznált programozásnyelv bemutatása	17
3.5. Model-View-Viewmodel (MVVM) minta hasznossága	17
3.5.1. MVVM minta alapelvei	18
3.5.2. MVVM előnyei	18
3.6. A saját szoftverünk nagyvonalú logikai terve	18
3.7. A saját szoftverünk statikus modelljének megalkotása	19
3.8. Alkalmazásunk implementálása	19
3.8.1. Projektkezelő (ProjectManager)	19
3.8.2. Tekintetadat-kezelő (GazeDataManager)	21
3.8.3. Adatkiírás (GazeDataSaver)	21
3.8.4. Eszközkommunikáció biztosítása	22
3.9. Elérhető szemmozgáskövető szoftverek összegyűjtése és elemzése	24
4. Az OGAMA bemutatása és összehasonlítása az általunk készített szoftverrel	26

4.1.	Felhasználói felület és felhasználói élmény	26
4.2.	Alkalmazásunk funkciói ismertetése	27
4.2.1.	Alkalmazás használata	27
4.2.2.	Valós idejű nézet	28
4.2.3.	Fixáció és statisztikai modul	28
4.2.4.	Konzol modul	29
4.3.	Kompatibilitás	29
4.4.	Validáció	30
4.5.	Fejlesztési irány és jövőbeli tervek	31
5.	Kognitív figyelem vizsgálata az elkészített alkalmazásunkkal	33
5.1.	A vizsgálatok körülményei	33
5.1.1.	A vizsgálatban résztvevő személyek	33
5.1.2.	A vizsgálatához használt tesztek	33
5.1.3.	A vizsgálat előkészületei	35
5.1.4.	A vizsgálat lefolytatása	35
5.2.	Teljes mintán végzett kísérlet eredményeinek kiértékelése	36
5.2.1.	A Stroop-Teszt eredményeinek normalitás vizsgálata	36
5.2.2.	A Visual Search Teszt eredményeinek normalitás vizsgálata	39
5.2.3.	A Stroop-Teszt eredményeinek összevetése a szemmozgásparaméterekkel .	41
5.2.4.	A Visual Search eredményeinek összevetése a szemmozgás paraméterekkel	44
5.2.5.	Az eredmények összefoglalása	47
5.3.	Az eredményekből levonható következtetések	47
6.	A kutatás korlátai	49
7.	Összefoglalás	50
8.	Mellékletek	60

Ábrák jegyzéke

1.	Folyamatkép a Stroop-tesztről.	34
2.	Folyamatkép a Vizuális Keresési Feladatról.	35
3.	Véletlenszerű tesztalanyok a szemmozgásadatok rögzítése közben.	36
4.	A fixációk számának normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).	37
5.	A pupillaátmérőjének normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).	37
6.	A fixációk időtartamának normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).	38
7.	A reakcióidők normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).	38
8.	A fixációk számának normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).	39
9.	A fixációk időtartamának normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).	40
10.	A pupillaátmérőjének normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).	40
11.	A reakcióidők normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).	40
12.	Az átlagos reakcióidő és az átlagos pupillaátmérő pontfelhődiagramja.	42
13.	A helyes válaszok száma és a fixációk számának pontfelhődiagramja.	43
14.	A helyes válaszok száma és a fixációk átlagos időtartamának pontfelhődiagramja.	44
15.	A pupillaátmérők és a reakcióidők pontfelhődiagramja.	45
16.	A fixációk számának és a helyes válaszok számának pontfelhődiagramja.	46
17.	A fixációk időtartamának és a helyes válaszok számának pontfelhődiagramja.	47

Táblázatok jegyzéke

1.	A ProjectManager (projektkezelő) szolgáltatás táblázata.	20
2.	A Projektkezelő által kiváltható események	20
3.	A GazeData absztrakt osztály táblázata.	21
4.	A GazeDataManager (adatkezelő) szolgáltatás táblázata.	21
5.	Az adatkiírást végző szolgáltatás megvalósításának táblázata.	22
6.	Az EyeTribeSDK fő függvényeinek táblázata.	23
7.	Az EyeTribe SDK által szolgáltatott kalibrációpontosság minősítések [30]	24
8.	A számított paraméterek táblázata.	31
9.	Az OGAMA illetve a SightScan statisztikai eredményeit összehasonlító táblázat.	31
10.	Az eloszlás alakjának felmérése a Stroop-Teszt eredmények esetében.	38
11.	Az eloszlás alakjának felmérése a Visual Search Teszt eredmények esetében.	41
12.	Spearman-féle rangkorreláció eredményei a pupillaátmérőre és a reakcióidőre a Stroop-Tesztben.	41
13.	Spearman-féle rangkorreláció eredményei a fixációk számára és a helyes válaszok számára a Stroop-Tesztben.	42
14.	Spearman-féle rangkorreláció eredményei a fixációk időtartamára és a helyes válaszok számára a Stroop-Tesztben.	43
15.	Spearman-féle rangkorreláció eredményei a pupillaátmérőre és a reakcióidőre a Visual Search Tesztben.	44
16.	Spearman-féle rangkorreláció eredményei a fixációk számára és a helyes válaszok számára a Visual Search Tesztben.	45
17.	Spearman-féle rangkorreláció eredményei a fixációk időtartamára és a helyes válaszok számára a Visual Search Tesztben.	46
18.	A tesztek eredményeinek és a szemmozgásparaméterek korrelációs kiértékelését összesítő táblázat.	47

1. Bevezetés

A szemmozgáskövetés és az aköré épült technológia rengeteg alkalmazási területet teremtett meg, például az egészségügyben a szemmozgás elemzése segíthet olyan betegségek korai diagnosztizálásában, mint az Alzheimer [1]-, vagy a Parkinson-kór. [2] [3] [4] A marketing területén a szemmozgások elemzése és kiértékelése segíthet megérteni a fogyasztók reakcióját például különféle reklámokra [5] és termékelhelyezésekre, betekintést nyújt az egyéni preferenciáiba [5] [6]. A fentebb említett alkalmazási területek kiemelik a szemmozgáskövető technológia fontosságát és alkalmazhatóságát. A programozás egy összetett kognitív tevékenység, amely magas szintű problémamegoldó képességet és mély koncentrációt igényel [7] [8] [9]. Ez a folyamat elengedhetetlen a jól strukturált, hatékony és hibamentes kód megalkotásához [8]. Ahhoz, hogy ezt a folyamatot mélyebben megértsük és elemezzük, olyan kutatási módszerekre van szükség, amelyek valós időben, a programozás tényleges végrehajtása során is alkalmazhatóak. Az ilyen módszerek lehetőséget teremtenek arra, hogy közvetlen kapcsolatokat tárjunk fel a programozási hatékonyság és a figyelmi állapotok, például a koncentráció vagy az éberség között. Korábbi vizsgálatok eredményei alapján a programozási hatékonyság szoros összefüggést mutat ezekkel a figyelmi jellemzőkkel [7] [8]. Az éberség és a koncentráció szintje hatással lehet arra, hogy a programozó mennyire képes összetett logikai feladatokat megoldani, valamint mennyire gyorsan és pontosan tud reagálni a kódolás során felmerülő problémákra. Az ilyen kutatások nemcsak a programozói teljesítmény optimalizálásában segíthetnek, hanem hozzájárulhatnak a tanulási folyamatok jobb megértéséhez, például a programozás oktatásában vagy a kezdők számára kialakított támogatási rendszerek fejlesztésében is [10] [11] [12]. Az ehhez hasonló vizsgálatokban az olyan technológiák, mint a szemmozgás követés, kiemelt szerepet játszhatnak, mivel ezek képesek valós időben rögzíteni a figyelmi fókusz mozgását, így betekintést nyújtanak a programozási folyamat során fellépő mentális terhelés dinamikájába [13] [14] [15]. Ezek a technológiák új távlatokat nyithatnak a programozás kognitív aspektusainak kutatásában, lehetővé téve a figyelmi mintázatok és a teljesítmény közötti kapcsolatok részletes feltárását.

Ugyanakkor ezek kutatások elvégzéséhez sajnálatos módon hiányoznak a jelenlegi piacról az olyan szemmozgás adatgyűjtő és elemző szoftverek, amelyek hosszú távon támogatottak, modern technológiával készültek és széleskörű funkcionalitást kínálnak, mindezt úgy, hogy közben megfizethetőek maradnak, vagy akár ingyenesen elérhetőek. Célunk, hogy ezt a hiányosságot pótoljuk. Dolgozatunk témája egy olyan szemmozgáskövető szoftver létrehozása, mely modern programozási technológiákra épül, hosszú távú támogatást és széleskörű funkcionalitást kínál ingyenesen bárkinek. Mielőtt megkezdtük volna a fejlesztést alaposan áttekintettük a hazai és nemzetközi szakirodalmat, hogy az elkészítendő szoftver maximálisan megfeleljen az ilyen típusú kutatások igényeinek. A dolgozat részeiben érinteni fogjuk a szemmozgás követés alapjait, a szemmozgások

fajtaikat, elemezzük a szemmozgás detektálás algoritmusait. A dolgozat másik fő célja egy olyan vizsgálat elvégzése, amely során a tekintetkövetésre alapozott elemzésekben összehasonlítjuk a szemmozgást jellemző paramétereket a pszichológiai kognitív koncentrációt és éberséget mérő tesztek eredményeivel. Az így feltárt összefüggések alapot adhatnak egy olyan módszer kidolgozásához, amely lehetővé teszi a koncentráció és éberség megfigyelését a programozási folyamat során, akár valós időben, akár utólagos elemzéssel, ezzel felmérni a programozó teljesítményét.

1.1. Kutatás feltételei

A kutatás során saját fejlesztésű szoftvert használtunk, adatrögzítéshez pedig Tobii Pro Nano szemmozgáskövető kamera, egy laptop, egy megfelelő méretű (24") másodlagos monitor, valamint különféle kognitív tesztek egészítettek ki. Az algoritmusok futtatásához a Microsoft Visual Studio Community fejlesztői környezetet, a kísérlethez szükséges tesztek implementálásához pedig átdolgozott, saját fejlesztésű figyelem tesztet alkalmaztunk, amelyeket PEBL (Psychology Experiment Building Language) platformról merítettünk. PEBL egy olyan ingyenes és nyílt forráskódú platform, ami lehetőséget ad a kutatóknak, hogy saját pszichológiai tesztet hozzanak létre, vagy mások által előre elkészített tesztet alkalmazzanak.

A kognitív tesztek lebonyolításához szükséges eszközöket és környezetet a Dunaújvárosi Egyetem, valamint a Dunaújvárosi Szakképzési Centrum Rudas Közgazdasági Technikuma és Kollégiuma biztosította. A kutatásban 18 év feletti résztvevők vettek részt, akik önkéntes alapon jelentkeztek, és nem álltak semmilyen gyógyszeres kezelés alatt. A résztvevőket a Rudas Közgazdasági Technikum és Kollégium diákjai és a Dunaújvárosi Egyetemen tanuló hallgatótársaink alkották.

1.2. Célkitűzések

A dolgozatunk elsődleges célja, hogy egy új szemmozgáskövető szoftver fejlesztésével egy olyan alkalmazást hozzunk létre, ami pótolja a jelenleg fennálló hasonló szoftverek hiányosságait, vagy a fizetős szoftverekkel szemben alternatívát nyújtson. A dolgozat másodlagos célja a programozás kognitív folyamatait és azok hatékonyságát meghatározó tényezők vizsgálata, különös tekintettel a koncentráció és az éberség szerepére. Ennek érdekében hat területet fogunk érinteni a kutatómunka során:

1. **Innovatív szemmozgáskövető szoftver fejlesztése:** A kutatás központi célkitűzése egy olyan modern szemmozgáskövető szoftver megalkotása, amely a legújabb programozási technológiákra épül. A szoftver tervezésekor fontos szempont, hogy hosszú távon fenntartható, folyamatosan frissíthető és széleskörű funkcionalitást kínáló platform legyen, amely képes pontosan és hatékonyan rögzíteni a szemmozgási adatokat. A rendszer felhasználóbarát ke-

zélőfelületet biztosít, és rugalmasan alkalmazkodik különböző kutatási és tesztelési igényekhez.

2. **A vizsgálatokhoz szükséges fejlesztési környezet megteremtése:** A kutatás során különböző fejlesztési eszközök, például a Microsoft Visual Studio és egy általunk készített tesztelőszoftver (CATC, Cognitive Attention Tests Collection) kerülnek felhasználásra. Ezek az eszközök lehetőséget adnak a kognitív tesztek integrálására, a vizsgálati adatok összegyűjtésére és azok valós idejű elemzésére. A cél, hogy olyan robusztus fejlesztési keretrendszert hozzunk létre, amely egyszerűsíti a szemmozgáskövetési kísérletek lebonyolítását és hatékony adatgyűjtést tesz lehetővé.
3. **Valós idejű adatfeldolgozás és vizualizáció megvalósítása:** A kutatás célja egy olyan szoftver kifejlesztése, amely valós időben képes nyomon követni a szemmozgásokat, és az eredményeket azonnal megjeleníti a vizsgálat során. Ez a funkció különösen fontos a viselkedés kutatásában, mivel lehetővé teszi a kutatók számára, hogy a teszt végeztével azonnal elemezzék a résztvevők reakcióit egyidőben. Az élő adatvizualizáció révén a kutatók gyors visszajelzést kaphatnak, és szükség esetén módosíthatják a vizsgálat paramétereit a pontosabb eredmények érdekében.
4. **Összehasonlítás és validáció egy piacon lévő szoftverrel:** Az újonnan kifejlesztett szemmozgáskövető szoftver megbízhatóságának és gyakorlati alkalmazhatóságának alátámasztása érdekében egy széles körben használt piaci szoftverrel való összehasonlítást tervezünk. Ez az összehasonlítás lehetőséget nyújt arra, hogy megmutassuk a saját fejlesztésű szoftver erősségeit és hiányosságait, miközben igazolja, hogy az új megoldás versenyképes alternatíva lehet a meglévő rendszerekkel szemben.
5. **Elemzési módszerek alkalmazása:** A szemmozgási adatok elemzéséhez különféle statisztikai és vizualizációs módszerek kidolgozása szükséges, hogy feltárhassuk a kognitív folyamatok és a szemmozgások közötti összefüggéseket. A vizsgálatok során kiemelten alkalmazzuk a Pearson- és Spearman-korrelációkat, amelyek lehetőséget adnak a szemmozgási paraméterek és a figyelem vagy memória kapcsolatának elemzésére. Továbbá, olyan vizualizációs technikákat, mint például hő térképeket is használunk a figyelmi mintázatok ábrázolására.
6. **A szemmozgás és kognitív folyamatok közötti összefüggések feltárása:** A kutatás alapvető célja, hogy részletesen megvizsgáljuk a szemmozgási paraméterek és a kognitív folyamatok közötti összefüggéseket. A szemmozgási adatok elemzése segíthet feltárni, hogyan befolyásolják a különböző kognitív terhelések és feladatok az egyén figyelmét és problémamegoldási folyamatait, ezzel pedig megfigyelni a programozó teljesítményét.

1.3. Felhasznált módszerek

A célkitűzéseink sikeres teljesítése érdekében, mielőtt belefogtunk volna a fejlesztési munkálatokba, alaposan áttekintettük számos belföldi és nemzetközi irodalmakat. Feltártuk a piacon elérhető szemmozgáskövető szoftverek főbb tulajdonságait, hiányosságait, hogy biztosítva legyen az elkészítendő szoftver megfeleljen teljes mértékben az elvárásoknak.

1.4. A dolgozat felépítése

A dolgozat 6 főrésze bontható, melyek az alábbiak:

1. A dolgozat első része bemutatja a kutatás célkitűzéseit és feltételeit, részletezi a felhasznált módszereket és eszközöket. Továbbá kitér a kutatási terület jelentőségére és a dolgozat tárgyalt témaköreinek fontosságára.
2. A második fejezet ismerteti a szemmozgáskövetés alapvető működési elveit, beleértve a technológiai alapokat és a gyakorlati felhasználási lehetőségeket. Bemutatásra kerülnek a különböző szemmozgáskövető eszközök, mint például a képernyőalapú és fejre csatolt rendszerek, valamint a hozzájuk kapcsolódó hardveres követelmények.
3. A dolgozat harmadik részében kerül bemutatásra a saját fejlesztésű szoftver elkészítésének folyamata, ideértve a fejlesztői környezet választását, a felhasznált programozási nyelvet (C#), valamint a Model-View-Viewmodel (MVVM) tervezési minta alkalmazását. A fejezet részletezi a szoftver felépítését, funkcionalitását, és az alkalmazott algoritmusokat.
4. Az negyedik fejezet célja a fejlesztett szoftver összehasonlítása egy már piacon elérhető, széles körben használt szemmozgáskövető rendszerrel, az OGAMA-val (Open Gaze and Mouse Analyzer). Az összehasonlítás során a megbízhatóság, funkcionalitás, felhasználói élmény és adatfeldolgozási képességek kerülnek kiértékelésre, hogy bemutassák a saját fejlesztésű rendszer előnyeit és korlátait.
5. A dolgozat következő részeiben kognitív figyelemtesztek végrehajtásával és eredmények elemzésével foglalkozunk. A kutatásban szereplő kognitív tesztek és azok eredményeinek értékelése ezekben a fejezetben található. A különböző szemmozgási paraméterek elemzésével a dolgozat betekintést nyújt a kognitív folyamatok és a szemmozgás közötti kapcsolatokba, és statisztikai módszerekkel tárja fel a vizsgálati eredményeket.
6. A dolgozat végén bemutatásra kerülnek a kutatás nehézségei és korlátai, valamint javaslatokat fogalmaz meg a jövőbeli kutatások és fejlesztések finomítására.

A dolgozatban az irodalmi források hivatkozása az IEEE szabványának megfelelően történt. Az olvashatóság és a terjedelem csökkentése érdekében néhány, a magyarázatokat kiegészítő ábra a mellékletben található.

2. Szemmozgáskövetési technológiák és módszerek

A szemmozgások vizsgálatának történelme már az 1800-as évektől kezdve követhető. Ezek a kutatások még az olvasás pontosabb megértése érdekében folytak. Mivel itt még nem beszélhetünk technológiákról, hanem közvetlen megfigyelésekről, így ezt az időszakot átugorva áttérünk a modernebb technológiákra. A szemmozgások vizsgálatához ezekben az időkben különböző elektródákat helyeztek a páciens szemei köré, hogy ezzel mérjék a potenciál különbségeket, amely folyamatos kapcsolatban állt a tekintettel [16]. Ezek után előtérbe kerültek a kontaktlencsék, melyek fém szélekkel rendelkeztek, ez lehetővé tette, hogy az elektromágneses változás alapján kövessék az emberi tekintetet [17]. Bár ezek a módszerek hatékonyak voltak a maguk idejében, ugyanakkor rendkívül költségesnek számítottak és gyakran orvosi beavatkozást is igényeltek [18] [19].

A technológia fejlődésével külső, infravörös alapon követő szemmozgáskövető kamerák megjelenése óta beszélhetünk a modern szemmozgás követésről. Ezek a kamerák pupilla és a szaruhártya tükröződésének mérésére szolgáló eljárást használják, amely nem más, mint a PCCR (*Pupil Center-Corneal Reflection*) [19].

2.1. Pupilla központú szaruhártya tükröződés (PCCR)

A PCCR eljárás során egy fényforrás használatával megvilágítják a szemet, amely a szaruhártyán és a pupillán tükröződik vissza, létrehozva így egy jól látható reflexiót (14. számú melléklet). Ezt a reflexiót az infravöröskamera rögzíti, a kamera és a reflexió között létrejött vektorból pedig megállapítható a tekintet pontos iránya [20] [21] .

Amikor a szem mozog, a pupilla pozíciója változik a kamera látóterében, így a kamera más szögben rögzíti a pupilla tükröződését. Ez a változás teszi lehetővé, hogy a PCCR alapú rendszerek nagyon pontosan kövessék a tekintet irányát. A kamera két egymást követő képkockáról megállapítja a két visszatükröződési pont távolságát és irányát egymástól [22] [23] [24].

2.2. PCCR elvet alkalmazó szemmozgáskövető eszközök

A videóalapú szemmozgáskövető rendszereknek két típusa van: képernyő alapú és fejhez csatoltak [25] [15]. A képernyőalapú kamerák esetében a kamera egy megjelenítő eszköz, mint például egy számítógépes monitor vagy televízió alatt helyezkedik el, ahol a kamera fix távolságban van, és felvétel során nem mozgatható. Ebben az esetben a szemmozgás-analizáló szoftver végzi a feladatokat, és kezeli a kijelzőn megjelenő alkalmazásokat, míg a személy a kijelzőn látható tartalmakat vizsgálja [26] [25] [3] . Ezzel szemben a dinamikus típusnál a kamera, mint szemüveg működik, amit a vizsgálati alany visel a tesztelés ideje alatt. Ez lehetővé teszi, hogy a résztvevő ne legyen

közvetlen fizikai kapcsolatban a vizsgált termékkel, így a vizsgálati környezet nem korlátozott, ami szabadságot biztosít a mozgásban [27] [28].

2.2.1. Képernyő alapú szemmozgáskövetők

Alábbiakban felsorolunk néhány szemmozgáskövető eszközt, amely népszerűek ezen a területen. Részletezzük továbbá a kamerák specifikus funkcióit és az alkalmazási területeit.

- **Tobii Pro Nano:** A Tobii Technology által fejlesztett termékcsalád része. Ezt az eszközt elsősorban olyan kutatási és felhasználói tesztekhez tervezték, ahol a portabilitás és a könnyű integrálhatóság kulcsfontosságú. A kamera 60 Hz-es mintavételi frekvenciával működik, amely megfelelő a legtöbb viselkedési kutatáshoz, ahol a felhasználók szemmozgását kell elemezni valós időben. Az eszközt széles körben használják pszichológiai kutatásokban, felhasználói élmény (UX) vizsgálatokban, és piackutatásban. Az eszköz kiváló pontossággal képes követni a pupilla mozgását és a tekintet irányát, amely fontos a részletes felhasználói interakciók elemzéséhez [29].
- **The Eye Tribe:** A The Eye Tribe egy korábban dán székhelyű vállalat volt, amely azonos nevű, alacsony költségű szemkövető technológiát fejlesztett. A cég, melyet 2011-ben alapítottak, azért vált ismertté, mert piacra dobta az egyik legkisebb és legolcsóbb szemkövető eszközt. 2016 végén a cég technológiáját felvásárolta a Facebook Oculus VR részlege, ami az eszközök jövőbeli fejlesztésének megszűnését eredményezte, azonban mai napig használják kutatásokhoz ezt az eszközt. Az Eye Tribe technológiát széles körben alkalmazzák a felhasználói interakciók, a játékok, a webdesign tesztelése és az oktatási alkalmazások területén. Ezenkívül a készülék különösen alkalmas volt a piackutatás és a viselkedési kutatások számára, mivel pontosan rögzítette a felhasználók szemmozgását [30].
- **EyeLink 1000 Plus:** Ez az eszköz egy nagy teljesítményű mérőeszköz, amely kiterjedt szemmozgáskövetési feladatok elemzésére alkalmas. Alacsony háttérzaj mellett, magas mintavételi frekvenciával rögzíti az adatokat, és azon kevés eszközök egyike, amely szoftveresen és technikailag is kompatibilis az MRI¹, MEG² és EEG³ készülékekkel, valamint a viselkedési laboratóriumokkal. A kamera rendelkezik szemkamerán kívül kartartóval, fejtartóval és asztalra rögzíthető szerkezettel, ami hozzájárul a szemkövetési feladatok hatékony elsajá-

¹**MRI (Magnetic Resonance Imaging):** Egy orvosi képalkotó technológia, amely erős mágneses mező és rádióhullámok segítségével részletes képeket készít a belső szervekről és szövetekről. Különösen alkalmas agyi struktúrák és funkciók vizsgálatára.

²**MEG (Magnetoencephalography):** Egy olyan idegtudományi eljárás, amely az agyban keletkező mágneses mezőket méri. Az MEG időbeli felbontása rendkívül magas, így pontosan követhető az agyi aktivitás dinamikája.

³**EEG (Electroencephalography):** Az agyi aktivitás elektromos jeleinek mérésére szolgáló módszer. Az EEG-vel az agy különböző régióinak aktivitási mintázatai vizsgálhatók.

tításához. Azonban ez az eszköz rendkívül költséges, illetve nehezen beszerezhető, ezt az eszközt nem tudtuk alkalmazni [31] .

- **Gazepoint 3:** Ez egy nagy teljesítményű eszköz, amely szoftver-hardver integrációval rendelkezik. Ezek a szoftverek lehetőséget adnak Gazepoint kamera kalibrálására, adatgyűjtésre és annak elemzésére gyári szoftverén belül. Azonban ez szintén rendkívül költséges számunkra így nem tudtuk alkalmazni a szoftverünkben [32] .

2.2.2. Fejre csatolt szemmozgáskövetők

- **EyeLink II:** Szinte az egyik leggyorsabb adatátviteli sebességgel és legnagyobb felbontással rendelkezik a dinamikus kamerák piacán. Az eszköz továbbá 500 Hz frekvencián biztosít binokuláris szemmozgáskövetést [31] .
- **ETVision:** Az ETVision kamerát az Argus fejlesztette ki, amely szintén magas frekvencián képes adatot továbbítani WIFI-n vagy Etherneten keresztül Laptopra, vagy akár SD kártyára is képes adatot menteni közvetlenül. A szemüveget korrigált vagy nem korrigált látású résztvevők egyaránt viselhetik [33] .
- **Neon:** A Pupil Labs által készített Neon szemüvegek a piac egyik legjobbjai a természetes fényben történő felvételt készítéshez (kültéren, belső térben napfény mellett stb.), ami ideálissá teszi őket olyan természetes környezetben végzett tanulmányokhoz, ahol nem lehet a fényforrást szabályozni [34] .

Mivel fejre csatolt, illetve laboratóriumokban használt szemmozgáskövető kamerák nagyon költségesek, így ezt kutatási korlát szempontjából annak kezelését nem tudtuk megvalósítani. A szoftver stabil kiadása után igyekszünk felvenni a kapcsolatot a szemmozgáskövető gyártókkal, hogy együttműködve implementálni tudjunk minél több eszközt az alkalmazásunkba.

2.3. Kamera és szenzor technológia

A szemmozgáskövető rendszerek technológiai alapkövei a magas felbontású kamerák és infravörös szenzorok. Az ilyen típusú kamerák a szem apró mozgásait és a pupilla változásait nagy pontossággal képesek érzékelni. Az infravörös technológia kiválóan alkalmas a pupilla pozíciójának meghatározására, mivel az infravörös fény minimálisan, de sok esetben egyáltalán nem zavarja a felhasználó vizuális érzékelését és stabil mérési eredményeket biztosít, miközben kiküszöböli a háttérvilágításból adódó torzításokat. [35] A kamerák gyors válaszideje, mely gyakran meghaladja a 60 Hz-et, nélkülözhetetlen a szakkádok (gyors szemmozgás), pontos rögzítéséhez, amelyek alapvető információkkal szolgálnak a felhasználó viselkedési szokásai megértéséhez. Ilyen fel-

használói szokások között szerepelhet az, hogy melyik színre/területre néz rá egy adott hirdetésen vagy weboldalon [5] [6] .

2.4. Kognitív tulajdonságok és szemmozgások közötti általános összefüggések

A szemmozgás és a kognitív tulajdonságok közötti összefüggések kutatása számos tudományág számára nyújt hasznos betekintést. A vizuális információfeldolgozás alapvetően kapcsolódik a kognitív folyamatokhoz, mint a figyelem és a memória [36] [37]. A szemmozgás paraméterek mind különböző mértékben árulkodnak arról, hogyan dolgozza fel az agy a látott információkat [38] [22] . A fixációk, amelyek során a tekintet hosszabb ideig egy helyen marad, a figyelmi fókusz stabilizálódását jelzik, míg a gyors szakkádok az információkeresés dinamikus folyamatait tükrözik [39] . Ezek a mozgások lehetőséget adnak arra, hogy következtetéseket vonjunk le az egyén mentális állapotáról és kognitív teljesítményéről [40] .

2.4.1. Figyelem és szemmozgás kapcsolata

A figyelmi folyamatok szorosan összefüggenek a szemmozgással, hiszen a vizuális figyelem irányítása és fenntartása döntő szerepet játszik az információfeldolgozásban. A szemmozgások minitázatai sokszor tükrözik, hogy az egyén mire koncentrál, és milyen ingerek vonják el a figyelmét. Az elemzések során a kutatók különböző figyelmi állapotokat tudnak azonosítani a szemmozgások monitorozásával, mint például a koncentrált figyelem vagy a figyelmi elterelődés. Az ilyen típusú elemzések különösen hasznosak lehetnek a tanulási folyamatok során. [41] [42] [43]

2.4.2. Szemmozgáskövető rendszerek alkalmazási lehetőségeinek elemzése

A modern szemmozgáskövetési technológiák lehetőséget biztosítanak arra, hogy pontos adatokat gyűjtsünk az egyén tekintetének mozgásáról és annak változásairól [14]. A jelenleg elérhető eszközök lehetővé teszik a fixációk és szakkádok pontos rögzítését, és számos területen alkalmazhatóak, beleértve a klinikai kutatásokat [1] [2] [3] [4] [44] [45] [46], marketinget [47] [48] [49] [50] és a felhasználói élmény méréseit [51] [52]. A szemmozgáskövetés során rögzített paraméterek, mint például a fixációk száma, időtartama és a szakkádok sebessége, fontos információkat nyújtanak a kognitív folyamatokról [53] [54]. Katona J. például egyik kutatásában [55] a .NET C# programozási nyelv adta LINQ lekérdezőnyelven alapuló lekérdezéses metódus szintaktika átláthatóságát vizsgálták meg egyetemista résztvevőkön szemmozgás követő technológia használatával. A vizsgálat célja az volt, hogy igazolja a Microsoft azon állítását [49], [50], mely szerint a LINQ deklaratív lekérdezésekhez használható szintaxis alapú lekérdezések valóban könnyebben olvashatók és hatékonyabban értelmezhetők, a metódus szintaxissal ellentétben.

2.5. Statisztikai módszerek kiválasztása a szemmozgás adatainak elemzésére

A szemmozgási adatok elemzése során különböző statisztikai és vizualizációs módszereket lehet alkalmazni, amelyek segítenek feltárni a kognitív tulajdonságok és a szemmozgás paraméterek közötti összefüggéseket [56] [57]. Az alapvető statisztikai technikák mellett – mint a korrelációs elemzések – vizualizációs eszközök, például hő térképek, lehetővé teszik a figyelmi mintázatok és az érdeklődés középpontjainak vizsgálatát.

Az elemzési módszerek közül az egyik leggyakrabban alkalmazott technika a korrelációs elemzés, különösen a Pearson- és Spearman-korrelációk [58] [59]. Ezek segítségével megállapítható, hogy az egyes szemmozgás paraméterek, mint a fixációk száma vagy időtartama, milyen mértékben állnak kapcsolatban a kognitív folyamatokkal, például a figyelemmel és a memóriával. Az ilyen elemzések rávilágítanak arra, hogyan hatnak az egyéni különbségek az információfeldolgozásra, és segítenek azonosítani azokat a mintázatok, amelyek egy adott feladat elvégzése során előfordulhatnak.

3. Innovatív szemmozgáskövető szoftver fejlesztése

A saját szoftverünk elkészítése előtt számos szakirodalmat olvastunk el a szemmozgáskövetés technológia működéséről, megismertük a különböző szemmozgási követők teljesítményét, illetve egyes, próbaverzióban elérhető alkalmazások funkcionalitását. Az általunk kiválasztott szemmozgáskövető eszközök pontos működését megjobban megismertük, hozzá tartozó dokumentációkat, példaprogramokat megnéztük, áttanulmányoztuk. Utána jártunk rendelkezik-e saját fejlesztői csomaggal, ha igen hogyan tudjuk alkalmazni a saját szoftverünkhöz. A következő alfejezetek összefoglalja a Tobii Pro Nano és a The Eye Tribe működtetéséhez szükséges algoritmusokat, leírja a követelményspecifikációt és összegyűjti az elkészítendő szoftverünk céljait.

3.1. Követelményspecifikáció

A következő követelményeket határoztuk meg a tervezett adatfeldolgozó szoftver-hardver környezet kialakításához:

- Az alkalmazás forráskódja kövesse a Robert Ceil Martin által meghatározott [60] [61] [62] **SOLID**⁴ (Single responsibility principle, Open/closed principle, Liskov substitution principle, Interface segregation principle, Dependency inversion principle) alapelveket.
- Az alkalmazás minél kevesebb erőforrást használjon a számítógépből.
- Az alkalmazás alkalmazza a modern szoftverfejlesztési technikákat, technológiákat.
- Az alkalmazás nézete és forráskódja legyen szét választja, alkalmazza az MVMM (Modell – View – View Modell) – amely hasznossága később részletezve – architekturális mintát.
- Az alkalmazás legyen moduláris; nagyon kevés módosítással hozzá lehessen adni új szemmozgáskövető eszközöket, új adatelemző modulokat.
- Az alkalmazás tudja fogadni elsősorban a Tobii Pro Nano és a The Eye Tribe szemmozgáskövető kamerák vezetékes (USB 3.0 porton keresztül) küldött adatokat és továbbítsa azokat egy TCP⁵ hálózati protokollon keresztül.
- Az alkalmazásban lehessen külön projekteket létrehozni, amelyt különböző célokra lehessen használni, többi projektől függetlenül.
- Az alkalmazás egy egységes adatbázisban – projektől elválasztva – tároljon el minden felhasználói beállítást, és a rögzített paramétereket.
- Az alkalmazás személyenként külön rögzítse az adatokat, végezze el számításokat és kapott eredményeket külön tárolja el az adatbázisban.

⁴**SOLID**: A SOLID alapelvek öt szoftvertervezési irányelvet foglalnak magukba, amelyek segítenek a kód olvashatóságának, karbantarthatóságának és bővíthetőségének javításában.

⁵**Transmission Control Protocol**: Megbízható, kapcsolatorientált protokoll, amely az internetes adatforgalom szabályozására szolgál.

- Vizsgálat során valós időben lehessen követni egy valós idejű nézetet a tesztfeladatok és a szem jelenlegi állapotát.
- Lehessen futtatni a kutató által meghatározott különböző tesztfeladatokat és tudja vezérelni azokat.

3.2. Fejlesztői környezet kiválasztása

A követelményspecifikációban összefoglalt feltételek mellett figyelembe kell vennünk a szemmozgási eszközök kompatibilitási és fejlesztői követelményeit is. Az általunk kiválasztott két eszköz az alábbi programozási nyelveket támogatja:

- **The EyeTribe:**
 - C# / Unity
 - C++
 - Java
 - Objective C
- **Tobii Pro Nano:**
 - C# / Unity
 - Python
 - Matlab / Octave
 - C

Megnéztük a fent említett szemmozgás követő eszközök fejlesztői követelményét, az EyeLink II és a GazePoint 3 mind támogatja az említett nyelveket, így a programozási nyelv tapasztalatunk függvényében és a meghatározott célokkal együttvéve a szoftverfejlesztést is elősegítő Microsoft Visual Studio 2022 Community nevű professzionális fejlesztői környezet került kiválasztásra a hozzá járó az egyik legelterjedtebb magasszintű, objektumorientált technológiát támogató programozási nyelv, a .NET C# került kiválasztásra.

A .NET környezetben két féle változat érhető el: A .NET Framework és a .NET (régebben .NET Core). A .NET a .NET Framework leszármazottja, amely számos olyan előnyökkel rendelkezik, amely alkalmazásunk jövőben igénybe tud venni. Ilyen például a .NET rendszeres frissítése Microsoft által, amely rendszeresen hoz új funkciókat és hibákat, réseket javítanak ki. Jobb teljesítményt és optimalizációt kínál az újraírt fejlesztői könyvtáraknak hála, illetve platformfüggetlen dinamikus könyvtárakat csatolhatunk az alkalmazásunkhoz. Az Entity Framework újabb verziói és az általunk kijelölt adatbázismotort is csak az újabb .NET támogatja, így ezeket figyelembe véve a .NET 8 került kiválasztásra a .NET Framework 4.8-cal szemben. [63]

3.3. Verziókövető rendszer alkalmazása

Mivel a saját szoftverünket közösen fejlesztjük egyidőben, ezáltal elengedhetetlen valamilyen verziókövető rendszer használata. Egy verziókövető segíti a programkódunk és a különböző kiadásunk nyomon követését, összegyűjti a korábbi pontos kódváltozásokat sorról-sorra, amelyekhez címeket és leírásokat adhatunk mit miért módosítottunk. Emellett bármi hiba folytán egy régebbi verzióra vissza kell térnünk a verziókövető segítségével ezt könnyedén megtehetjük. Emellett olyan automatizáló eszközöket adhatunk a kódunkhoz (mint például összevonási konfliktus keresés, kódfordítás) ami fejlesztés során segít elkerülni jövőben felmerülő hibákat, illetve jelentős energiát spórol meg számunkra. Ezekre a feladatokra GitHub verziókövetőt jelöltük ki, ami szinte az egyik legnépszerűbb portál fejlesztők körében, több kurzusokon is szó esett már róla az egyetemen.

3.4. A felhasznált programozási nyelv bemutatása

A C# (C-Sharp) egy, a kétezres évek elején kifejlesztett objektumorientált, általános célú programozási nyelv, fejlesztője a Microsoft. Azzal a céllal hozták létre, hogy legyen egy könnyen kezelhető, sokoldalú és rugalmas nyelv, amely alkalmas különböző célú és típusú szoftverfejlesztési feladatokra [64]. A C# nyelv szintaxisa rendkívül hasonlít más népszerű programozási nyelvek szintaxisára, mint például a Java vagy a C++. A C# nyelv támogatja az deklaratív, imperatív, funkcionális, objektumorientált és generikus programozási pragmatikákat, ezzel lehetővé téve a fejlesztők számára, hogy a számukra legmegfelelőbb megközelítést válasszák ki az általuk fejlesztett programkódban. Ezen kívül a C# kiemelkedő jellemzői közé tartozik az automatikus memóriakezelés, amit az úgynevezett Garbage Collector (GC, az angol Garbage Collector kifejezésből, magyarul automatikus szemétyűjtő) biztosít, illetve a Type Safety (az angol Type Safety kifejezésből, magyarul típusbiztonság), ami csökkenti a futásidőben előforduló típusokkal kapcsolatos hibákat. Lehetőségünk van az aszinkron programozásra, ami lehetővé teszi robusztusabb programkódok elkészítését.

3.5. Model-View-Viewmodel (MVVM) minta hasznossága

Az MVVM (Model-View-Viewmodel) minta a szoftvertervezésben egy elterjedt architektuális minta. Az elnevezés három fő komponensre utal: a modellre, a nézetre, és a nézetmodellre. Ez a tervezésminta lehetővé teszi, hogy a felhasználói felületet leválasszuk a mögöttes kódról, az üzleti logikától, ezzel elősegítve a kód újra felhasználását, a tesztelhetőséget és a fejlesztési folyamatok egyszerűsítését. [65]

3.5.1. MVVM minta alapelvei

Az MVVM (Model-View-ViewModel) minta egy olyan tervezési minta, amely három fő komponenst tartalmaz: a Modellt, a Nézetet, és a Nézetmodellt. A Modell az alkalmazás adatokat tartalmazza, amelyek az üzleti logikát és az adatelérési műveleteket kezelik. A Nézet feladata a felhasználói felület megjelenítése, míg a Nézetmodell a Modell és a Nézet közötti közvetítőként funkcionál, és a Modellből származó adatokat átalakítja úgy, hogy azok megfelelően megjeleníthetők legyenek a Nézetben. [66]

3.5.2. MVVM előnyei

Az MVVM alkalmazásának sok előnye van, részben a felület és az üzleti logika elkülönítéséből adódik. Az MVVM minta különösen előnyös abban, hogy szigorúan elválasztja a felhasználói felületet (Nézet) a logikai és adatkezelési rétegtől (Modell), egy köztes réteg (Nézetmodell) révén, amely az adatok és a felhasználói felület közötti kommunikációt hídálja át. Ezáltal a Nézetmodell biztosítja, hogy a Nézet csak a megjelenítéssel foglalkozzon, míg a Modell az adatok kezelésével és logikai műveletekkel. Az MVVM struktúra lehetővé teszi, hogy a fejlesztők függetlenül dolgozhassanak a felhasználói felületen és a háttérlogikán anélkül, hogy egymást zavarnák. Ez növeli a fejlesztés hatékonyságát, mert a felület vagy a logika módosítása nem igényli a másik komponens átalakítását. Továbbá, az MVVM elősegíti a kód újra felhasználását és tesztelését, mivel a moduláris felépítés miatt könnyebb izoláltan tesztelni az egyes részeket.

Egy tanulmány kimutatta, hogy az MVVM minta jobb teljesítményt biztosít, 95,1%-os szinttel szemben az MVC 41,6%-os teljesítményével [67]. Az MVVM kifejezetten előnyös a kód újrahasználhatósága és a komponensek függetlensége szempontjából, mivel biztosítja a felhasználói felület módosítását az üzleti logika megváltoztatása nélkül. Fejlesztésünk során az MVVM-et alkalmaztuk, ami lehetővé tette az üzleti logika és a felhasználói felület egyidejű fejlesztését, így hatékonyabbá téve a munkafolyamatot.

3.6. A saját szoftverünk nagyvonalú logikai terve

A SightScan szoftver működésének logikai tervét a 6. és 7. számú melléklet mutatja be, amely átláthatóan szemlélteti a szoftver és a szemmozgáskövető eszköz közötti kommunikáció lépéseit. Az ábra bemutatja, hogyan zajlik az adatfeldolgozás és az adattárolás. Az eszközkommunikáció, a rögzítés és az adattárolás külön szálon futnak, amelyeket számokkal jelöltük meg.

3.7. A saját szoftverünk statikus modelljének megalkotása

A megvalósítandó szemmozgáskövető szoftver-hardver környezet tervezésében nagy segítséget jelentett a vizuális ábrázolást elősegítő UML (Unified Modelling Language) modellező nyelvben. Az alkalmazás osztály-szintű struktúrájának tervezésénél az UML egy statikus diagramját, az osztálydiagramot alkalmaztuk. A diagramok elkészítéséhez az Enterprise Architect[68] nevű vizuális modellező és tervező eszközt használtuk. (5.számú melléklet)

3.8. Alkalmazásunk implementálása

A szoftverünk megvalósítása során nagy segítségül vettük a Programozás I., II., és a Szoftverfejlesztési technológiák kurzusokon megszerzett ismereteket, valamint az e területtel foglalkozó nemzetközi szakirodalmakat. A fejlesztés során kulcsszempontként kezeltük a modularitást, az egyszerű integrálhatóságot és a tiszta kód megvalósítását. Ezek érdekében a SOLID alapelveket alkalmaztuk, amelyek hozzájárultak a program moduláris és jól strukturált felépítéséhez. Ezen kívül a függőségek kezeléséhez az IoC (Inversion of Control)⁶, azaz függőség megfordítás elvét, valamint a DI (Dependency Injection)⁷, azaz függőség befecskendezést használtuk. Ezek a tervezési minták lehetővé tették, hogy a rendszer könnyen bővíthető és karbantartható legyen, miközben biztosították a kód tisztaságát és átláthatóságát [69]. A program fejlesztése során kiemelten ügyeltünk arra, hogy a rendszer egyes moduljai jól elkülönüljenek egymástól, ezzel elősegítve a funkciók egymástól független módosíthatóságát és az új komponensek gyors integrálását, kompatibilitását.

Bár az alkalmazás minden osztályát terjedelmi korlátok miatt nem részletezzük, azokat a kulcsfontosságú elemeket mutatjuk be, amelyek a rendszer alapjait képezik, és amelyek a tiszta kódra, valamint a skálázhatóságra épülő fejlesztési megközelítésünket szemléltetik.

3.8.1. Projektkezelő (ProjectManager)

A követelmény specifikációban meghatároztuk, hogy az alkalmazásnak képesnek kell lennie különböző kutatási állapotok elkülönítésére, ezért ezeket külön projektként definiáltuk. Az egyes projektek önálló kezelését a ProjectManager szolgáltatás végzi, amely felelős a projektek létrehozásáért, tárolásáért, valamint az azokhoz kapcsolódó adatok és beállítások kezeléséért. Ez a modul biztosítja, hogy minden projekt elkülönített környezetben fusson, és a különböző kutatási fázisok adatai ne keveredjenek egymással.

⁶**Inversion of Control (IOC):** Az IoC egy szoftverfejlesztési elv, amelyben az irányítás a keretrendszer vagy külső komponens kezébe kerül, ahelyett, hogy az egyes objektumok maguk kezelnék az erőforrások kezelését vagy a folyamatirányítást.

⁷**Dependency Injection (DI):** A Dependency Injection egy szoftvertervezési minta, amely az osztályok közötti függőségek kezelésére szolgál. Ennek során az osztályok nem saját maguk hozzák létre a szükséges objektumokat, hanem külső komponensek injektálják ezeket az osztályokba.

A projektkezelő szolgáltatás részletes megvalósítását az 1. táblázat tartalmazza, amely bemutatja a fő funkciókat és a modul működéséhez szükséges paramétereket.

1. táblázat. A ProjectManager (projektkezelő) szolgáltatás táblázata.

Visszatérési típus	Függvény azonosítója	Leírás
bool	LoadLocalProject	Betölti a kiválasztott projektdokumentumot.
bool	SaveLocalProject	Elmenti a projekt aktuális állapotát.
bool	ModifyExistingProject	Módosítja az aktuális projekt tulajdonságait, mint például projektnév, projektleírás, elérési útvonal.
SightScanProject	GetCurrentProject	Visszaadja az aktuális projekt objektumát.
void	SetCurrentSubject	Elmenti az aktuális tesztalanyt a projekt állapotában.
SightScanSubject	GetCurrentSubject	Visszaadja az aktuális tesztalany példányát a projektből.
ObservableCollection	GetCurrentProjectSubjects	Visszaadja a tárolt tesztalanyokat a projektfájlból.
bool	AddSubject	Hozzáad egy új felhasználót a projekthez.
bool	RemoveSubject	Eltávolítja a kijelölt felhasználót a projektből.

A projektkezelő események célja, hogy lehetővé tegyék az alkalmazás különböző részei közötti hatékony kommunikációt és együttműködést. Ez a megközelítés egyszerűsíti a kód karbantarthatóságát és bővíthetőségét, hiszen az új funkciók hozzáadása vagy a meglévők módosítása kevesebb változtatást igényel az alkalmazás többi részén. Az események segítségével az osztályok automatikusan reagálhatnak a projekthez kapcsolódó változásokra.

A 2. táblázat részletezi azokat az eseményeket, melyeket a Projektkezelő szolgáltatás tud kiváltani, hogy értesítse az azokra feliratkozott osztályokat.

2. táblázat. A Projektkezelő által kiváltható események

A projektkezelő osztály által kiváltható események	
Esemény neve	Esemény leírása
ProjectChanged	Lefut, ha egy új/másik projekt betöltődött az alkalmazásban.
SubjectAdded	Lefut, ha egy új felhasználó hozzá lett adva a projekthez.
CurrentSubjectChanged	Lefut, ha egy másik felhasználó lett kiválasztva az alkalmazásban.

3.8.2. Tekintetadat-kezelő (GazeDataManager)

A szemmozgás paraméterek kezeléséért egy generikus osztály felel, mely paraméterül TGazeData osztályt várja. A TGazeData az adott eszköznek a szem adatainak modellje, amely jelen esetben TobiiGazeData és EyeTribeGazeData lehet. Ezek az eszközspezifikus modellek a GazeData absztrakt osztályból származik le, közös elemei a képernyő koordináták és az osztály létrehozásakor automatikusan deklarált Timestamp tulajdonság. Az eszközök közös absztrakt modelljét a 4. táblázat definiálja.

3. táblázat. A GazeData absztrakt osztály táblázata.

Típus	Változó azonosító	Változó részletezése
double	X	A fókuszpont X koordinátája kijelzőn (pixelben).
double	Y	A fókuszpont Y koordinátája kijelzőn (pixelben).
double	PupilDiaL	A baloldali szem pupillájának átmérője.
double	PupilDiaR	A jobboldali szem pupillájának átmérője.
SightScanProject	Project	A kiválasztott SightScan projekt objektuma.
SightScanSubject	Subject	A kiválasztott tesztalany objektuma.
long	Timestamp	A rögzítéskor lévő dátum és idő Unix Timestampben.

A GazeDataManager szolgáltatás megvalósítását a 4. táblázat részletezi.

4. táblázat. A GazeDataManager (adatkezelő) szolgáltatás táblázata.

Visszatérési típus	Függvény azonosító	Leírás
void	HandleGazeData	Paraméterül vár TGazeData típust a fent részletezett működés alapján. Ezt a függvényt az adott eszköz beérkezett adat eseményekkor futtatja le, amire a nézetmodell fel is van iratkoztatva.
IEnumerable<TGazeData>	GetGazeData	Visszaadja a memóriában tárolt szemmozgási adatokat megadott eszköztípus alapján.

3.8.3. Adatkiírás (GazeDataSaver)

Egy szemmozgáskövető alkalmazás alapvető funkciója az adatmentés, amely lehetővé teszi az adatok utólagos elemzését és visszakeresését. Ez nemcsak a kutatási folyamatok dokumentálásában

játszik szerepet, hanem az eredmények validálásában és a vizsgálatok pontos reprodukálásában is. Az adatmentést az IGazeDataSaver interfészt megvalósító osztályok végzik, amelyek felelősek a szemmozgási paraméterek megfelelő formátumban történő tárolásáért. Ezek az osztályok biztosítják az adatok rendszerezett és strukturált mentését, hogy a későbbi feldolgozás során az adatok könnyen elérhetők és elemezhetők legyenek.

Az adatkiírást végző szolgáltatást az 5. táblázat írja le.

5. táblázat. Az adatkiírást végző szolgáltatás megvalósításának táblázata.

Típus	Függvény azonosító	Leírás
void	SaveData	Publikus függvény, paraméterül várja az eszköznek az adattípusát. Függvény meghívásakor a memóriában lévő adatokból elvégzi a fixáció- és a statisztika számolást majd a megvalósítás alapján tárolja az összegyűjtött adatokat.
void	SaveRawData	Privát függvény, a nyers adat kiírását végzi el implementálás alapján.
void	SaveFixations	Privát függvény, kiszámolt fixációkat tárolja el implementálás alapján.
void	SaveStatistics	Privát függvény, a kiszámolt statisztikát tárolja el implementálás alapján.

3.8.4. Eszközkommunikáció biztosítása

A fenti fejezetekben leírtuk az szemmozgáskövető rendszer működéséhez elengedhetetlen részeit, most már csak az eszköz kommunikációjára van szükség. Mivel program egyik kiemelt célja és a követelményspecifikációban is meghatározásra történt, egyszerre több szemmozgáskövető eszközt kell támogatnia az alkalmazásnak. Ennek függvényében szintén interfészben írtuk meg a szemkövető eszközhöz szükséges függvényeket, amiből leszámaznak a különböző modellű eszközök. Jelen esetben két eszköz van implementálva, a The Eye Tribe az EyeTribeTracker osztályban, míg a Tobii Nano a TobiiNanoTracker megfelelőjében. A két eszköz kommunikáció teremtése hasonló, így a terjedelmi okokra hivatkozva csak a The Eye Tribeot részletezzük. A modul felépítését leíró UML statikus osztálydiagramot az 13. számú melléklet ábrázolja.

Kommunikáció teremtés

A szemmozgás követéséhez szükséges eszköz adatainak elérése érdekében először is el kell indítani az Eye Tribe szervert a tesztek során használt számítógépen, ami a kamera működéséért és a TCP

hálózati protokollon keresztüli adatok átviteléért felelős. Az adatáramláshoz a következő három lépést kell megtenni:

1. Csatlakozás az Eye Tribe TCP szerveréhez.
2. Egy külön szál létrehozása majd indítása, amely folyamatosan fogadja eszköztől az adatokat és adja tovább a nézetmodellnek.
3. Egy időzítő beállítása, amely meghatározott időközönként úgynevezett „heart beat” vagy másnéven „keep-alive” biteket küld a szervernek, hogy fenntartsa az aktív kapcsolatot az eszközzel.

Az Eye Tribe egy fejlesztői csomaggal is rendelkezik, amely támogatja ezeket a funkciókat. A szükséges funkciókat az EyeTribeSDK névtér tárolja. A 6. táblázat összefoglalja azokat a kulcsfontosságú funkciókat, amelyek lehetővé teszik az eszközzel való folyamatos kommunikációt az EyeTribe SDK felhasználásával.

6. táblázat. Az EyeTribeSDK fő függvényeinek táblázata.

Visszatérési típus	Függvény azonosító	Leírás
bool	Activate	Ez a tagfüggvény regisztrálja az alkalmazásunkat, mint kliens az Eye Tribe szerver felé. Amennyiben a szerver nem elérhető (például nem fut), a program felugró ablakban tájékoztatja a felhasználót, hogy nem sikerült a szerverre való csatlakozás.
void	AddGazeListener	A függvény az Eye Tribe fejlesztői csomag eseményeihez köti az aktuális osztály metódusait (például OnGazeUpdate).
bool	Deactivate	Feladata, hogy program befejezésekor kapcsolatot bontson az Eye Tribe szerverrel.

Eszközkalibráció

Ahhoz, hogy szemmozgási paramétereket kapjunk a szervertől, szükséges az eszköz kalibrálása. Ez az eljárás a résztvevő szeméit a kijelzőhöz rendeli. A kalibrálás folyamán a képernyő különböző részein (leginkább sarkok és középpont) pontokat jelenít meg, amik idővel eltérő helyekre mozognak, miközben a szemmozgás követő mintákat rögzít. Miután az összes pontot rögzítette, a szerver kiszámítja és aktiválja a kalibrációt, aminek segítségével az X és Y koordináták jelennek meg a kijelzőn. A kalibrációs pontok számát általában kilencre állítják, de egy funkcióval ez

személyre szabható, beleértve a pontok számát, koordinátáit, időtartamát és átmérőjét. Több pont hozzáadása növeli a tekintet koordinációjának pontosságát. A programban a kalibrációs pontosság az Eye Tribe szabványai szerint a 7. táblázat alapján kerül kiértékelésre⁸.

7. táblázat. Az EyeTribe SDK által szolgáltatott kalibrációpontosság minősítések [30] .

Kalibrációs pontosság	Minősítés	Leírás
<0,5	Kiváló	Ez az optimális kalibrációs eredmény, nincs szükség újra kalibrálásra.
0,7	Elfogadható	Ez a kalibrációs eredmény alkalmas a szemek követésére újra kalibrálás nélkül.
1	Közepes	Ez a kalibrációs eredmény elfogadható, viszont javasolt a kalibrálás ismétlése.
>1,5	Kedvezőtlen	Nem optimális a szem követésére, kalibrálás szükséges.
-	Nincs kalibrálva	Az eszköz nincs még kalibrálva.

3.9. Elérhető szemmozgáskövető szoftverek összegyűjtése és elemzése

Az első PPCR elvű szemmozgáskövető kamerák megjelenése óta több adatgyűjtő és elemző szoftver készült ezek eszközök kezelésére, van, ami eszközgyártók készítettek a saját hardverének kezelésére, van, amik eszköztől függetlenül működnek, azonban azok inkább kereskedelmi szoftverek, de találhatunk néhány ingyenesen elérhető nyílt forráskódú megoldásokat. Az alkalmazásunk elkészítése előtt fontosnak tartottuk már piacon lévő szoftvereket megismerni, illetve hiányosságait megkeresni. Néhány népszerűbb szoftver kerül részletezésre:

- **GazePointer:** A GazePointer a felhasználó webkameráját használja a paraméterek fogadására különböző algoritmusok alapján. Célja, hogy ne keljen költséges szemmozgáskövető eszközt beszerezni tekintet elemzésére. A GazePointer a GazeFlow algoritmuson alapszik, és egy fejlesztői könyvtárat biztosít más alkalmazásokba. Azonban ez a szoftver rendkívül korlátolt, kutatásokra nagyon pontatlan. A GazePointer Github oldalán utolsó frissítés 6 éve történt, illetve elavult .NET környezetet használja.
- **Tobii Nexus:** A Tobii Nexus egy most készülődő API fejlesztőcsomag, ami szintén webkamerát használja paraméterek fogadására, továbbá a 20 éves múltbeli tapasztalataival a cég AI bevonásával különböző adatok elemzésére próbál kitörni, mint például az oktatásban a tanu-

⁸A táblázat alapján azt feltételezve, hogy a felhasználó körülbelül 60 cm-re ül a képernyőtől vagy az eszköztől, a "kiváló" pontosság 0,5-1 cm-es átlagos eltérésnek felel meg.

ló teljesítményének figyelésére, vagy egészségügyben az ADHD, depresszió és más tünetek elemzésére. Azonban ez a fejlesztőkörnyezet régóta nem rendelkezik még semmilyen alkalmazással, semmilyen elérhető fejlesztői-csomaggal így nem tudunk együtt dolgozni vele.

- **Tobii Pro Lab:** A Tobii Pro Lab szintén Tobii Technology egyik szemmozgás elemző szoftvere, amely különböző beépített kiértékelési modulokat tartalmaz, azonban ez fizetős alkalmazás és csakis Tobii eszközökkel működik együtt. Mivel mi több eszközt próbálunk implementálni különböző gyártóktól, továbbá zárt forráskódú, így ezt az alkalmazást nem tudjuk használni továbbfejlesztésre.
- **iMotions:** Az iMotions nem feltétlen csak szemmozgáskövető eszközzel foglalkozik, hanem több más biometrikus eszközöket is integrál (mint például EEG, EKG stb). Az iMotions támogatja a mostani szemmozgáskövető eszközök zömét. Szinte ez az egyik legjobb szemmozgás elemző szoftver kutatási célra a nagy tudása és integrálhatósága miatt, azonban ennek a költsége sincs lentebb. Egy éves előfizetés 3400 euróba kerül, és ez a licenz, illetve további modulok hozzáadása alapján még jobban emelkedik. Szegényebb országokban, illetve kisebb kutatói körökben ez nagy hátrányt tud okozni.
- **PyGaze:** A PyGaze egy integráló hidat biztosít a PyGame, a PsychoPy, PyLink, SensoMotoric Instruments iView API és Tobii SDK között, így egy egyéges kezelőt ad a kamera és szoftver között, azonban ez csak egy fejlesztői modul, nem rendelkezik alapvetően grafikus felülettel és más használó környezettel, fejlesztőn múlik, hogy implementálja ezt a modult az alkalmazása között. Azonban ez a fejlesztőkörnyezet nem nagyon frissítik (van olyan modulja, ami 10 éve kapott utoljára frissítést, van, ami csak néhány éve).

Azonban elérhető egy olyan szemmozgás elemző alkalmazás, ami ingyenes, nyílt forráskódú, integrálja a szemmozgáskövető eszközöket különböző gyártóktól, rendelkezik grafikus felülettel, illetve olyan beépített modulokkal rendelkezik, ami kutató számára elengedhetetlen. Az alábbi fejezetben ezt a programot mutatjuk be részletesen és hasonlítjuk össze a mi programunkkal, továbbá választ adunk hiányosságaira és milyen limitációi, hibái vannak.

4. Az OGAMA bemutatása és összehasonlítása az általunk készített szoftverrel

A saját fejlesztésű alkalmazásunk leginkább OGAMA-val van vetélytársban. Az OGAMA egy nyílt forráskódú, ingyenes szemmozgás elemző szoftver, amely integrálja a különböző márkájú eszközöket és egységesen tud adatokat gyűjteni tőlük, elemezni adatokat, illetve lementeni adatbázisokba.

Az alkalmazásunk fejlesztése során az OGAMA funkcióit szeretnénk összehasonlítani a saját fejlesztésű SightScan alkalmazásunkéval, különös tekintettel a fixációk és szakkádok azonosítására és vizualizációjára. Az OGAMA által biztosított eszközök és analitikai lehetőségek alapján célunk, hogy a SightScan funkcionalitását értékeljük és fejlesszük, különösen a modern adatfeldolgozási és szemmozgás-elemzési technológiákra alapozva. Az összehasonlítás során figyelembe vesszük a szoftverek által nyújtott vizualizációs opciókat, az exportálható adatok formátumát, valamint az alkalmazások felhasználói élményét is, hogy átfogó képet kapjunk a két szoftver közötti különbségekről és lehetőségekről.

Az Open Gaze and Mouse Analyzer szoftver a hivatalos weboldalon, a www.ogama.net weboldalon tölthető le, a forráskódja pedig a GitHubon is elérhető. A szoftver telepítése és elindítása után egy párbeszédablak jelenik meg, ahol új projektet hozhatunk létre vagy meglévőt választhatunk ki. Egy új projekt létrehozásakor az alkalmazás különböző beállításokat kér, mint például hova mentse le a projektet, a monitor felbontását, a kamera frissítési rátáját, egyéb fixációs beállításokat. Ezek adatok megadása és mentése után a projekt elkészül, és a párbeszédablak bezárásával a program kezdőoldalára jutunk, amelyet alábbi ábra mutat.

4.1. Felhasználói felület és felhasználói élmény

A felhasználói felület tervezése kritikus fontosságú [70], mivel befolyásolja a termék sikerét, a felhasználói elégedettséget, illetve a szoftver használhatóságát. Hatékony UI tervezésnek integrálnia kell a fejlesztési folyamatba, hogy biztosítsa a termék megfelelő a felhasználói igényeit és elvárásait. Mivel a felhasználói felület megtervezése közvetlen hatással van a felhasználói élményre [71], alapvető tényezővé vált a szoftverek piaci elfogadottságában. A szoftverek és webalkalmazások fejlesztői manapság egyre inkább arra törekednek, hogy egyre letisztultabbá, és jobban kezelhetővé tegyék a szoftverüket. A UI/UX ezért kulcsfontosságú tényezővé vált abban, hogy a felhasználóból pozitív élményt váltsanak ki.

A Windows Presentation Foundation (WPF) egy olyan fejlesztői platform, amely lehetővé teszi a szoftverfejlesztők számára, hogy könnyen kezelhető és vizuálisan letisztult felhasználói felüle-

teket hozzanak létre. A WPF, mint korszerű felületfejlesztő eszköz, kulcsfontosságú az azonnali válaszadású és felhasználóbarát interfészek kialakításában. Ez az eszköz alapvető a szoftvermérnökök számára, hogy hatékonyan megvalósítsák a tervezési koncepciókat, és így jelentősen javítsák a felhasználói élményt. [72]

A korábbi sztenderdek számító ingyenesen elérhető, nyílt forráskódú OGAMA Windows Forms technológiát használ, amely nem rendelkezik semmilyen grafikai képességgel, nem tud összekötni különböző felületi modulokat egynél több szállal, nem támogatja a hardveres gyorsítást. Továbbá, ha gyengénlátó felhasználó beállította operációs rendszerében a száz százaléknál nagyobb nagyítást (például százhusz százalékot), akkor a programban megjelenő betűi homályosak lesznek, az ikonok össze mosódnak és több tartalom nem fér ki az ablakokon. Az alábbi ábrán látható, hogy feliratok teljesen homályosak, eszközeleíró sáv pedig teljesen kitakarja az OK és a Bezárás gombot. Ennek oka az OGAMA pixelben adja meg a megjelenési adatokat, nem vektorban.

Ezzel szemben, a mi felületünk a WPF előnyeit kihasználva kínál egy modern, letisztult felhasználói felületet, a fent említett hibákra pedig orvosságot talált. A WPF előnye, hogy XAML (eXtensible Application Markup Language) alapú leíró nyelvet használ, ami lehetővé teszi a felület és a funkcionalitás szétválasztását, ezáltal növelve a kód átláthatóságát és karbantarthatóságát. WPF segítségével részletes stílusokat és témát tudtunk alkalmazni, ami egy egységet megjelenést biztosítanak, ezzel javítva a kezelhetőséget. Alkalmazás fejlesztése során újra tudtuk használni a felületi elemeket, mivel WPF egy komponens-alapú architektúra, ezzel csökkentve a hibajelenségeket, erőforrást. A WPF rendelkezik hardveres gyorsítással, ezáltal kész a frissítések során elengedhetetlen a grafikai elemek és animációk zökkenőmentes megjelenítésére, amely jövőbeni frissítések során elengedhetetlen követelmény lehet. A 8. számú melléklet összehasonlítja a két alkalmazás megjelenését.

4.2. Alkalmazásunk funkciói ismertetése

Az alábbi alfejezetekben részletesen ismertetjük a saját fejlesztésű alkalmazásunk funkcióit és képességeit.

4.2.1. Alkalmazás használata

Az alkalmazásunk fejlesztése során figyelembe vettük az egyszerű kezelhetőséget, amely OGAMA-hoz hasonlítva nagymértékben gyorsítja a kutatási folyamatot. Míg OGAMA-ban a felvétel elkezdése előtt több lépésben, különböző ablakokban kell állítani az eszköz megfelelő működését, ami nagyon lelassítja a kutatást. Ezt figyelembe vettük, és ezalpján valósítottuk meg a projektkezelő nézetet, ahol egy oldalon lehet csatlakozni szemmozgás-követő eszközökre, továbbá tesztalanyokat

lehet hozzáadni, módosítani, törölni és itt érhetjük el a valós idejű nézetet is. Be lehet állítani melyik monitort szeretnénk megfigyelni, milyen frissítési frekvenciával rögzítsen a kamera, és kettőnél több monitor esetén melyik monitoron jelenjen meg a kezelő felület. Minden alapvető funkció egy helyen található meg. A 9. számú melléklet ezt a felületet mutatja meg képernyőkép formájában.

4.2.2. Valós idejű nézet

A „Begin Test” gombra kattintva megjelenik a kutatói nézet a kiválasztott monitoron, ahol kezelni lehet az eszközt és az alkalmazást; el lehet indítani a felvételt, szüneteltetni vagy leállítani ezzel pedig megkezdve az adatkiírást. Itt találhatjuk meg a valós idejű visszacsatolás ablakát, ahol rögzítés során figyelemmel tudjuk kísérni a szemmozgási paraméterek változását (mint például a fókuszpont aktuális koordinátája, fixációs adatok, statisztikák stb.) és más eszközök adatait (például okosóra pulzusmérője). Segítségével felvétel során észre tudjuk venni az eltéréseket, továbbá olyan további kutatásokat tesz lehetővé, ahol a valós idejű adatelemzés létszükségletű (mint például több tesztből álló feladatsor esetén további elágazást tehetünk tesztek közé). A 10. számú melléklet képernyőképe mutatja meg ennek a nézetét.

Az ábra bal felső sarkában található a szívritmus-ingadozások grafikon, amely a tesztalany szívritmusának alakulását mutatja az idő függvényében. A grafikon alatt megjelennek a statisztikai mutatók, beleértve az aktuális, átlagos, maximális és minimális szívritmust, valamint a szívritmus-ingadozás mértékét. A szívritmustól jobbra jelenik meg annak a monitornak az előnézete, ahol a teszt folyik, azon pedig lehet követni a fókuszpontot (azaz felhasználó hova néz pontosan), alatta pedig a koordinátát is frissíti. Bal alsó részen találhatóak a tesztelési vezérlők, ahol kiválasztható a tesztalany, elindítható, leállítható és befejezhető a tesztelési folyamat. A felvételi állapot (recording) azaz felvétel alatt van, és a monitor kijelzőjének adatai is itt jelennek meg. A jobb alsó sarokban egy pupillatágulás az idő függvényében grafikon található, amely a pupillatágulás változását ábrázolja egy rövid időintervallumon belül.

4.2.3. Fixáció és statisztikai modul

A SightScan szoftverünk funkciói közül kiemelnénk a különböző adatvizualizációs képek elkészítését például a statisztikai modulunk által: hőtérképek, szórási térképek, vektor térképek, korrelációs mátrixok, pupillaátmérő változás időfüggvényében, és különböző tekintet és fixációs hisztogramok. A képek elkészítését előre lefordított Python nyelven megírt .exe futtatható fájlok végzik el, a már elmentett tekintet és fixációs adatokból. Az előre lefordított, Windowson is egyből futtatható programokat Seaborn, [73] Pandas [74] és Matplotlib [75] Python könyvtárak felhasználásával készítettük el. A felhasználók kiválaszthatják, hogy melyik adatvizualizációs képekre tartanak igényt, és a program meghívja automatikusan a megfelelő fájlt a megfelelő bemeneti paraméterekkel és le-

generálja a felhasználó által kért képet vagy képeket majd megjeleníti azt alkalmazásban. A fixáció és statisztikai modul ábrázolását a 11. számú melléklet mutatja meg, azonban ez csak illusztráció, és nem feltétlen tükrözi a program végleges kinézetét, mindazonáltal fontos kiemelni, hogy működőképes és a rendszer alapelveit hitelesen szemléltetik.

A SightScan szoftverünkbe implementálásra került egy képernyőfelvevő szolgáltatást, amely rögzíti a projektben beállított „Presentation Screen” -t, vagyis a prezentációs képernyőt, melyet a végfelhasználó szabadon állíthat be, ez az a képernyő, amelyet a tesztalany néz, ezen történik a szemmozgásadatok követése és felvétele. Az így felvett videófájlokra egy átfedő réteggel rá lehet helyezni a tekintet adatokat az időfüggvényében. Így akár videófájlokra lehet hő térképeket megjeleníteni. Ehhez a képernyőfelvevő szolgáltatáshoz a ScreenRecorderLib [76] C# könyvtárat használtuk a Nuget csomagkezelőn keresztül, mely az előzetes tesztek alapján egy könnyen konfigurálható, megbízható, és legfőképp „light-weight” vagyis kevés erőforrást használó könyvtárnak bizonyult.

4.2.4. Konzol modul

Alkalmazásban elérhető egy konzol, amely lehetőséget biztosít a felhasználóknak bizonyos funkciók parancssorból történő elérésére és vezérlésére. A konzol segítségével az alkalmazás különböző moduljai közvetlenül elérhetők parancssorban, amely lehetővé teszi, hogy a felhasználók igényeik szerint konfigurálják az egyes műveleteket. A konzolon keresztül például elindítható a kalibrálás folyamata a „calibrate” parancs kiadásával, amivel a rendszer biztosítja az eszköz pontos beállítását a további adatgyűjtési lépésekhez. A felhasználók ezen túlmenően a „record” parancs segítségével azonnal elindíthatják az adatgyűjtési folyamatot, míg a „getfixations” parancs segítségével lekérhetik a rögzített fixációs adatokat közvetlenül a konzolon keresztül. A konzol scriptelhetősége egyedülálló lehetőséget nyújt az automatizálásra. Azáltal, hogy a kutatók képesek szkripteket írni, egyszerűen elvégezhetnek bonyolult vagy összetett konfigurációs folyamatokat, anélkül, hogy manuálisan kellene minden beállítást egyesével végrehajtaniuk. Például egy felhasználó megírhat egy olyan scriptet, amely automatikusan elvégzi az eszköz kalibrációját, elindítja az adatgyűjtést, majd az összegyűjtött adatokat azonnal elmenti. Ez különösen hasznos, ha az alkalmazást rendszeresen hasonló körülmények között használják, mivel a script elmenthető, újrafuttatható és testreszabható különböző paraméterekkel, igény szerint.

4.3. Kompatibilitás

A korábbi kutatásunkban részleteztük, hogy OGAMA hatalmas kompatibilitási problémákkal küszködik a modern eszközökkel és implementálható technológiákkal, mivel az x64-es környezetben nem tud megfelelően futni. Ha ebben használjuk, akkor a rögzítéskor számos kivételt kapunk, adatkiírás sem minden esetben sikeres. Gyakori kivételt szokott dobni a DirectShow modul, amit

OGAMA használ a rögzítésre, erre semmilyen megoldást nem találtunk. Ha x86-ra fordítjuk a szoftvert, akkor pedig számos szemmozgáskövető eszközt nem lehet implementálni manuálisan, mivel az csak x64-es környezetben tud dolgozni.

Az alkalmazás C# programozási nyelvben íródott, melynek számos előnye van a szemmozgáskövetés szempontjából, például kompatibilitás a meglévő könyvtárakkal és eszközökkel, valamint hatékony támogatás különböző operációs rendszerekhez. Az alkalmazás ezen tulajdonságai biztosítják, hogy a legkülönbözőbb szemmozgáskövető szoftverek integrálhatóak legyenek minimális fejlesztési erőfeszítéssel. A C# választása lehetővé teszi a különböző szemmozgáskövető eszközök és technológiák egyszerű összekapcsolását, mivel a nyelv rendelkezik az objektumorientált fejlesztés eszközeivel, valamint gazdag, széles körben elérhető könyvtárak állnak rendelkezésre, amelyek közvetlen kapcsolatot biztosítanak a hardvereszközökkel.

A program felépítése moduláris, így rugalmasan és hatékonyan bővíthető. Az alkalmazás modulközi kommunikációját az Inversion of Control támogatja, amely lehetővé teszi újabb modulok és eszközök hozzáadását a rendszerbe a kód alapvető struktúrájának megváltoztatása nélkül.

A moduláris struktúra kialakítása során az IoC konténer használata kiemelten fontos szerepet kapott. Az IoC konténer lehetővé teszi, hogy az egyes komponensek, például különféle szemmozgáskövető modulok vagy adatfeldolgozó elemek, ne legyenek szorosan kötve egymáshoz, így az új funkciók és eszközök hozzáadása során minimális kódmódosítás szükséges. Az IoC konténer elvén alapuló megközelítés az injektált függőségeket automatikusan kezeli, és lehetővé teszi, hogy az újonnan bevezetett modulok, mint például különböző típusú szemmozgáskövető eszközök, könnyedén csatlakozzanak a rendszerhez anélkül, hogy az alapalkalmazást módosítani kellene.

Az OGAMA alkalmazásunkkal szemben nem követ semmilyen függőségkezelést, nem támogatja az x64 modulok használatát, így egy új eszköz (például Tobii Pro eszközök) implementálása nagyon időigényes folyamat és rengeteg kivétellel jár amit kezelni kell. Az OGAMA már nagyon elavult, nem támogatott .NET Framework környezetet használ, így legtöbb modern könyvtárat már nem tud kezelni.

4.4. Validáció

A SightScan alkalmazásunk megbízhatóságát az OGAMA (Open Gaze and Mouse Analyzer) alkalmazás segítségével validáltuk, amely széles körben ismert és elismert a szemmozgáskövetési kutatások területén. Az OGAMA funkcióinak és eredményeinek elemzése lehetőséget nyújtott arra, hogy az általunk fejlesztett szoftver megbízhatóságát összehasonlítsuk egy már meglévő és validált rendszerével, ezzel megalapozva az alkalmazásunk hitelességét. Az összehasonlítás so-

rán több szempontot is figyelembe vettünk. Először is, a fixációk és szakkádok azonosításának pontosságát vizsgáltuk mindkét szoftverben. Az OGAMA által biztosított adatokat referenciaként használtuk, és összevetettük azokat a SightScan által rögzített paraméterekkel. A paramétereket a 8. táblázat részletezi.

8. táblázat. A számított paraméterek táblázata.

Paraméter elnevezése	Paraméter részletezése
Fixation Count	Az összes fixáció darabszáma.
Fixation Frequency	Másodpercenkénti fixációk száma.
Fixation Duration Median	A fixációk időtartamának mediánja.
Fixation Mean	A fixációk átlaga.
Total Fixation Time	A fixációk összesített időtartama.
Avarage Saccade Distance	Átlagos szakkád távolság két fixációs pont között pixelben.
Avarage Saccade Velocity	Átlagos szakkád gyorsaság két fixációs pont között pixel/másodpercben.

Az eredmény bizonyítja (1. sz. melléklet és 2. sz. melléklet, illetve a 9. táblázat), hogy a fejlesztett szoftver megbízhatóan reprodukálja az OGAMA szoftver által számított adatokat. Ezen felül a szoftverünk felhasználóbarát módon jeleníti meg a végfelhasználóknak a számított paramétereket.

9. táblázat. Az OGAMA illetve a SightScan statisztikai eredményeit összehasonlító táblázat.

Statisztikai modul eredményeinek összehasonlítása		
Paraméter azonosítja	OGAMA eredmény	SightScan eredmény
Fixation Count	12	12
Fixation Frequency	3,19	3,19
Fixation Duration Mean	142,92	142,92
Fixation Duration Median	149,5	149,5
Avarage Saccade Distance	264,36	264,36
Avarage Saccade Velocity	773,17	773,17

4.5. Fejlesztési irány és jövőbeli tervek

A szemmozgáskövető szoftverünket igyekszünk folyamatos és érdemi frissítésekkel ellátni. Természetesen általános és úgynevezett „quality-of-life” frissítések és fejlesztések is várhatóak a felhasználóiélmény javítása érdekében, azonban az egyik nagy mérföldkő, amit szeretnénk a jövőben megugrani, az a mesterséges intelligencia integrálása a szoftverkörnyezetünkbe. A szemmozgásadatok

vizsgálata mesterséges intelligencia segítségével egy rendkívül hatékony módja lehet a felhasználói interakciók és viselkedés mélyebb megértése érdekében. A mesterséges intelligencia képes tanulni a felhasználói interakciókból, így a használat során a szoftver algoritmus folyamatosan képes lesz a fejlődésre. Ezen alkalmazkodó és adaptálódó mesterséges intelligencia által biztosított algoritmusok révén a szoftver egyre pontosabban képes lesz előre jelezni és akár értelmezni a felhasználó szándékait [77].

Másik mérföldkő, melyet teljesíteni szeretnénk, hogy a szoftverünk interakcióba tudjon lépni különféle digitális eszközökkel - első sorban mobiltelefon (4. számú melléklet) -, hogy növeljük a felhasználói élményt. Ennek érdekében célunk, hogy a Windows programunk kommunikálni tudjon egy telefonos alkalmazással a lokális hálózaton belül, így könnyítve a tesztelést és a kutatást.

Az SightScan szoftverünkben már integrálásra került egy lokális szerver (12. számú melléklet), melyet bekapcsolva, a program képes a távoli parancsok feldolgozására és válasz küldésére. Az egyszerűsége és a skálázhatóságra törekedtünk a szerver létrehozásakor, így egy HTTP protokollt alkalmazó szerver került implementálásra, amely alapértelmezetten a 7070-es porton „hallgatózik”, ha a felhasználó bekapcsolja. A válaszokat JSON formátumban küldi el, ami lehetővé teszi, a könnyű integrálást és feldolgozást különböző platformokon.

Nem szeretnénk korlátozni a kutatókat, és a felhasználókat, így a szoftverbe integrált szerver API (*Application Programming Interface* vagy *Alkalmazásprogramozási felület*) nyilvános és dokumentált lesz, így bárki készíthet hozzá saját alkalmazást, ha úgy kívánja. A szoftver fejlődésével a nyilvános API is bővülni fog új parancsokkal.

5. Kognitív figyelem vizsgálata az elkészített alkalmazásunkkal

A saját fejlesztésű szoftverünk használatával két különböző kognitív koncentrációs és éberségi tesztet végeztünk el. Ezen figyelemtesztek során rögzítettük a tesztalanyok szemmozgás paramétereit (például: tekintet koordináták, pupillaátmérők), melyekből később a SightScan kiszámolta a következtetések levonásához szükséges adatokat, mint például a fixációk száma, vagy az egyes fixációk időtartama. A megfelelő tesztek kiválasztásához a PEBL nyílt forráskódú tesztgyűjteményéből válogattunk, majd pedig .NET 8 keretrendszeren egy saját fejlesztésű alkalmazásban futtatuk le azokat, hogy kutatásunkhoz leghatékonyabban gyűjtsük ki a szükséges paramétereket.

5.1. A vizsgálatok körülményei

Ebben az alfejezetben részleteésre kerülnek a tesztalanyok alapvető adatai, a vizsgálatra használt eszközöket és a figyelemteszteket.

5.1.1. A vizsgálatban résztvevő személyek

A tesztek elvégzéséhez segítséget nyújtott a Dunaújvárosi Szakképzési Centrum Rudas Közgazdasági Technikum és Kollégium több osztálya és a Dunaújvárosi Egyetemen tanuló hallgatótársaink, akiknek a résztvételi szándékukat innen is nagyon köszönjük. A diákok mind betöltötték a tizen-nyolcadik életévüket, önkéntes alapon jelentkeztek a kutatás segítéséhez, személyes adataik névtelenül kerültek tárolásra. A tárolásra került adatok tartalmazták a résztvevők életkorát, hogy bal, vagy jobbkezesek-e, illetve, hogy hordanak-e szemüveget. A tesztek elvégzésében összesen 100 fő vett részt, 31 nő és 69 férfi, 18-22 ($M=20,03$ $SD=1,16$) életév között. A tesztalanyok kipihenten érkeztek, alkohol, gyógyszer, vagy tudatmódosítószer hatása alatt nem álltak.

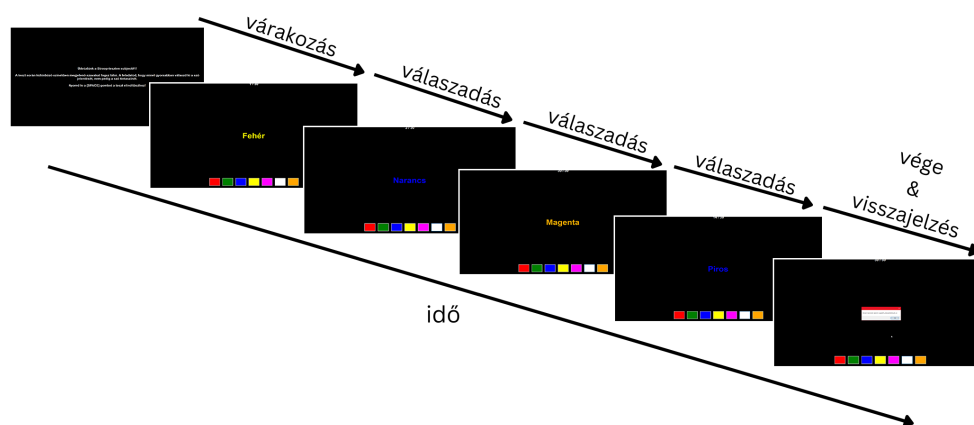
5.1.2. A vizsgálatához használt tesztek

A már fentebb említett PEBL tesztkörnyezetből kettő figyelemtesztet választottunk ki, valósítottunk meg és használtunk fel a kutatás során. Az egyik a Stroop-teszt, a másik pedig egy vizuális keresési feladat (Visual Search Test) volt. A tesztek alábbi alfejezetekben kerülnek részletezésre.

Stroop-Teszt

A Stroop-teszt egy klasszikus pszichológiai teszt, amely a szelektív figyelem, a kognitív kontroll és a gátlási folyamatok vizsgálatára szolgál [78]. A feladat során a résztvevők olyan szavakat látnak, amelyek színek nevét tartalmazzák, azonban a szavak tintaszíne eltérhet a szöveg jelentésétől. Például, ha a „piros” szó zöld színnel van írva, a résztvevők feladata a tintaszín azonosítása, nem

pedig a szó jelentésének felolvasása. Ez a konfliktushelyzet kognitív interferenciát generál, mivel a résztvevőknek gátolniuk kell az automatikus olvasási folyamatot, és a nem domináns választ kell preferálniuk. A Stroop-tesztet a kutatásunk keretében egy módosított formában alkalmaztuk. Ebben a verzióban a résztvevőknek a szó jelentését kellett kiválasztaniuk a képernyő alján elhelyezkedő színes gombok közül, miközben a szó tintaszíne eltért a megfelelő gomb színétől. Például, ha a „zöld” szó kék színnel volt írva, a résztvevőknek a zöld színű gombot kellett megnyomniuk, nem pedig a kékét. A Stroop-Tesztről a folyamatképet az alábbi, 1. ábra mutatja.

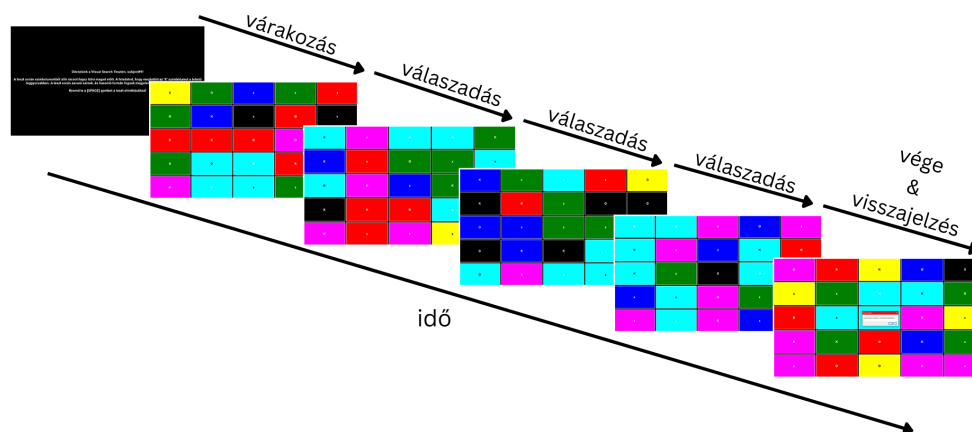


1. ábra. Folyamatkép a Stroop-tesztről.

Forrás: Saját készítésű ábra.

Vizuális keresési feladat

A vizuális keresési feladat során a résztvevők feladata egy adott célelem, jelen esetben az "X" szimbólum, azonosítása volt egy zavaró ingerekkel teli rácsban. A feladat komplexitását azáltal növeltük, hogy olyan elemeket helyeztünk el a rácsban, amelyek hasonló formájúak, de eltérő színűek voltak, így fokozva a vizuális figyelem igénybevételét. A zavaró ingerek jelenléte jelentős kihívást jelentett, mivel a célelem azonosítása mellett a résztvevők figyelmét a szín- és formabeli hasonlóságok is próbára tették. A Visual Search tesztről a folyamatképet az alábbi, 2. ábra mutatja.



2. ábra. Folyamatkép a Vizuális Keresési Feladatról.

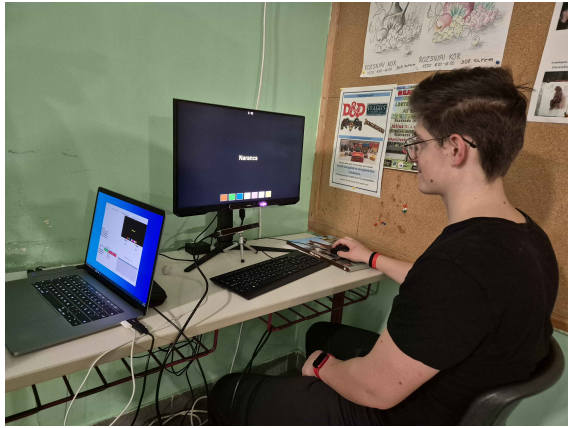
Forrás: Saját készítésű ábra.

5.1.3. A vizsgálat előkészületei

A vizsgálathoz szükségünk volt egy külső monitorra, melyet a teszteléshez beállított laptop-hoz csatlakoztattunk, mint másodlagos monitor. Erre a célra egy 24” átmérőjű, 16:9 képarányú és 1920x1080 felbontású monitort választottunk. A résztvevő tesztalany elé a vizsgálat során egy billentyűzetet helyeztünk, amivel be tudta vinni a számítógépbe a válaszait. A résztvevők szemmozgásainak figyelését az EyeTribe és Tobii Pro Nano eszközök végezték, amely a SightScan szoftve-rünkkel kommunikált, amely pedig rögzítette az adatokat. Az eszközt a résztvevőktől körülbelül 60 centiméter távolságban helyeztük el, és azt egyedileg, mindenkinek külön kalibráltuk. Kalibrálás során sosem alkalmaztunk kiválónál ($<0,5$) rosszabb minőségbesorolású kalibrációs eredményt.

5.1.4. A vizsgálat lefolytatása

A résztvevők egyesével ültek a tesztre kijelölt területre, ahol elsőnek megtörtént az eszköz kalibrációja személyenként külön-külön, sikeres kalibrációt követően pedig tájékoztattuk őket a teszt-feladatokkal kapcsolatban, és instrukciókat adtunk nekik a feladatok elvégzéséhez. A teljesítés személyenként körülbelül 5-6 percet vett igénybe. A tesztek lefolytatása közben a SightScan rögzítette a szemmozgásparamétereket, illetve képernyőfelvételt készített a monitoron. Ez látható a 3. ábrán.



3. ábra. Véletlenszerű tesztalanyok a szemmozgásadatok rögzítése közben.

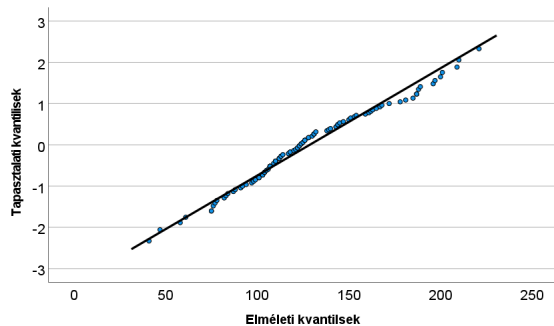
Forrás: Saját készítésű ábra.

5.2. Teljes mintán végzett kísérlet eredményeinek kiértékelése

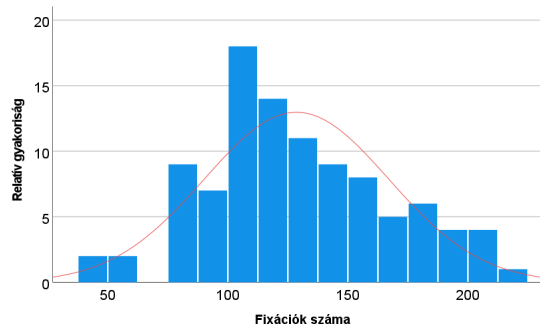
A saját tesztelőszoftver és a saját fejlesztésű szemmozgás követő alkalmazásunk által nyújtott mérési adatok közötti kapcsolatok az összes résztvevőn végzett kísérletek alapján értékelhetők és azonosíthatók a teljes minta tekintetében. A kapcsolatok intenzitása a saját fejlesztésű környezetben elvégzett egyes tesztek eredményeinek elemzésével, egy kiemelt, a teszt végeredményét a legjellemzőbb paraméter alapján lett megállapítva. Az változók értékeinek eloszlás vizsgálatra a Shapiro-Wilk (W) teszt alkalmazásával határoztuk meg. Az adatok intervallum skálán kerültek mérésre. A két változó közötti kapcsolat erősségének mérésére a korreláció módszerével kerültek meghatározásra. A két teszt helyes válaszok számával kerültek összevetésre a rögzített szemmozgás adataival. A kiértékelés során a saját környezetünkben futó Visual Search Teszt, valamint a Stroop Teszt végeredményeivel (helyes válaszok száma, átlag reakcióidő) kerültek összevetésre a szemmozgás paraméterek.

5.2.1. A Stroop-Teszt eredményeinek normalitás vizsgálata

A Stroop-Teszt eredményei esetében két esetben áll fenn a normalitás: a fixációk számánál ($W = 0,980, p = 0,133$) illetve a pupillaátmérőnél ($W = 0,985, p = 0,317$). Ezt az eredményt tovább erősítik az alábbi ábrák (4a, 4b, 5a, 5b) grafikonjai, mivel az eredmények nagy része illeszkedik a regressziós egyenesre, illetve a hisztogramon látható, hogy az eredmények jól követik a haranggörbe alakját.



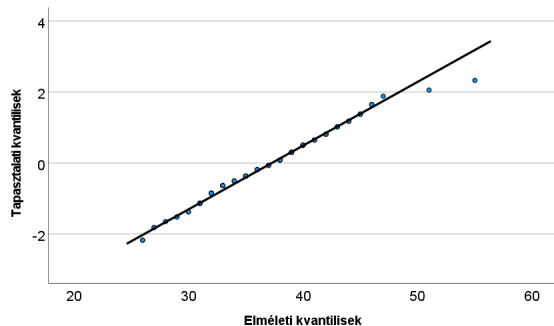
(a)



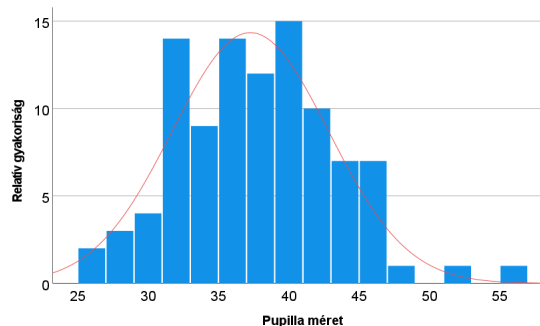
(b)

4. ábra. A fixációk számának normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).

Forrás: Saját készítésű ábra.



(a)

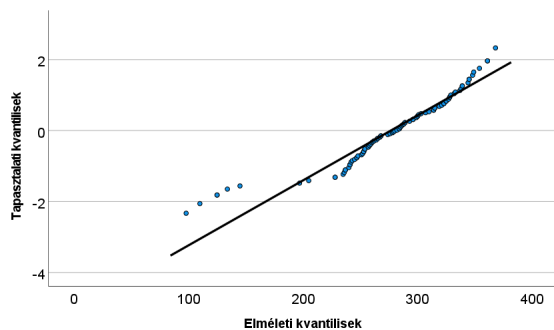


(b)

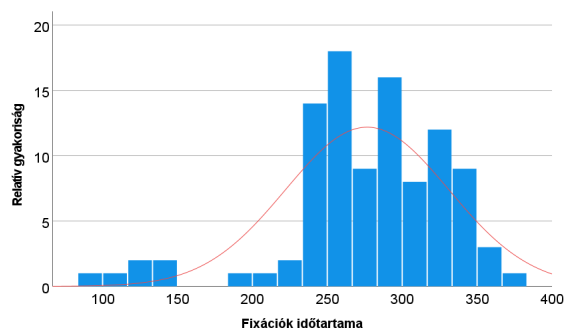
5. ábra. A pupillaátmérőjének normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).

Forrás: Saját készítésű ábra.

A további esetekben sem a fixációk időtartamánál ($W = 0,913, p < 0,001$), sem a reakcióidőnél ($W = 0,955, p < 0,002$), sem pedig a helyes válaszok számánál ($W = 0,814, p < 0,001$) nem áll fenn a normalitás. Ezt a megállapítást a következő ábrák (6a, 6b, 7a, 7b) grafikonjai is megerősítik, ugyanis az eredmények nem követik az ideális normál eloszlás regressziós egyenesét, illetve a hisztogramon látható, hogy az adatok jelentősen kívül esnek a haranggörbéből.



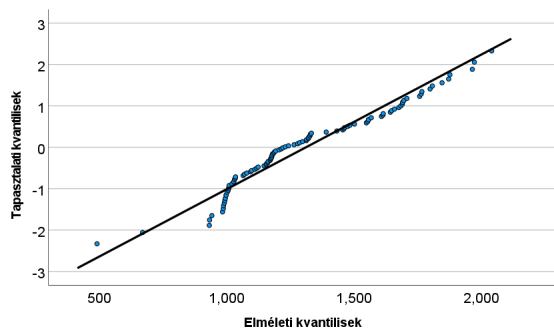
(a)



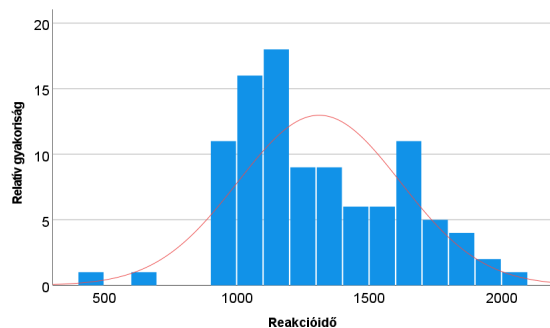
(b)

6. ábra. A fixációk időtartamának normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).

Forrás: Saját készítésű ábra.



(a)



(b)

7. ábra. A reakcióidők normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).

Forrás: Saját készítésű ábra.

A Stroop-Teszt eredményei esetében a 10. táblázat részletezi és összefoglalja a fent ábrázolt eredményeket.

10. táblázat. Az eloszlás alakjának felmérése a Stroop-Teszt eredmények esetében.

Stroop-Teszt						
Paraméter	Shapiro-Wilk			Átlag $\pm$ SD	Medián	Skewness (ferdeség)
	Próbastatisztika	p-érték	N			
Fixációk száma	0,980	0,133*	100	128,61 $\pm$ 38,449	123,5	0,306
Fixációk Időtartama	0,913	<0,001*	100	276,61 $\pm$ 54,562	281	-1,106
Pupillaátmérő	0,985	0,317*	100	37,27 $\pm$ 5,563	38	0,271
Reakcióidő	0,955	0,002*	100	1311,37 $\pm$ 307,463	1226	0,344

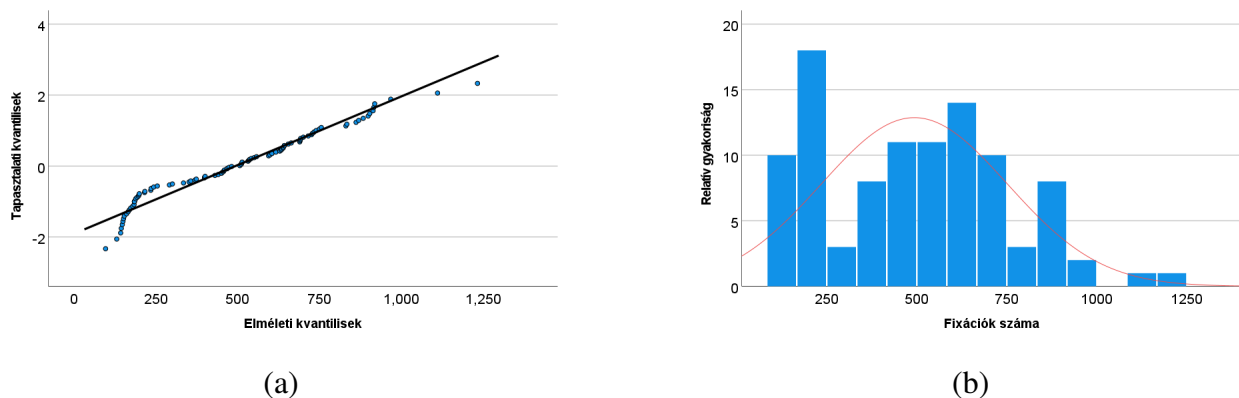
* 2-oldali

A szemmozgáskövető rendszer és a tesztelőszoftver által szolgáltatott paraméterek esetében enyhe aszimmetria figyelhető meg. Ez az eltérés főként a fixációk időtartamánál, a reakcióidőknél és a helyes válaszok számánál mutatkozik meg. A hisztogramokon ezek az aszimmetriák láthatók, azonban ahol a normalitás feltétele teljesül, az aszimmetria nem indokolja annak elvetését, mivel a különbségek nem számottevőek.

5.2.2. A Visual Search Teszt eredményeinek normalitás vizsgálata

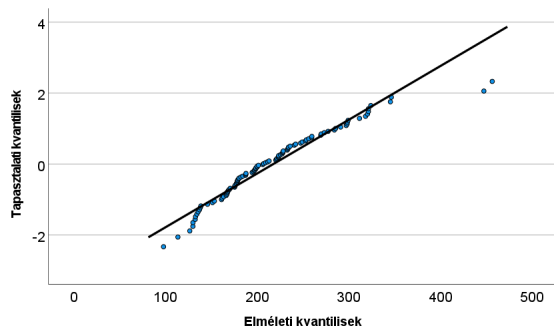
A Visual Search tesztfeladat eredményei esetén semmilyen rögzített paraméter esetén nem áll fenn a normalitás, ide értve a fixációk számát ($W = 0,953, p = 0,001$), a fixációk időtartamát ($W = 0,941, p < 0,001$), a pupillaátmérőt ($W = 0,904, p < 0,001$), a reakcióidőt ($W = 0,952, p = 0,001$), illetve a helyes válaszok számát ($W = 0,642, p < 0,001$). A kiértékelés eredményét az következő ábrák grafikonjai és alátámasztják, mivel a hisztogramok és a normál eloszlások formája jelentősen eltér egymástól, ami arra utal, hogy a mögöttes populációk nem közelíthetők normális eloszlással. Ezek az eredmények indokolták, hogy elsősorban nem parametrikus statisztikai tesztek alkalmazunk az elemzés során.

A Visual Search Teszt eredményeit a 11. táblázat foglalja össze és részletezi a alábbi ábrákon (8a, 8b, 9a, 9b, 10a, 10b, 11a, 11b) látható eredményeket.

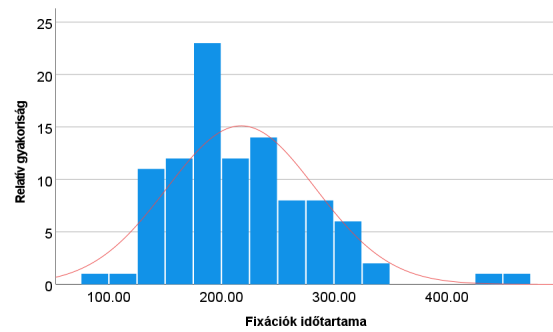


8. ábra. A fixációk számának normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).

Forrás: Saját készítésű ábra.



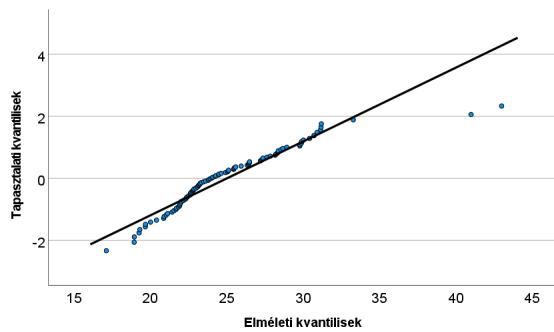
(a)



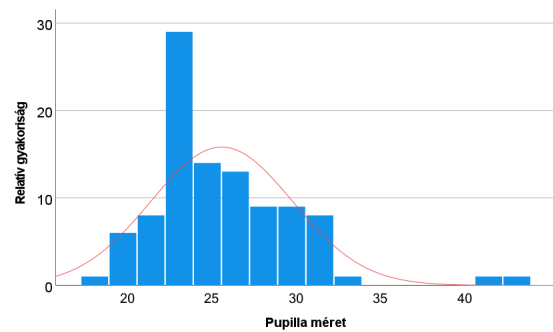
(b)

9. ábra. A fixációk időtartamának normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).

Forrás: Saját készítésű ábra.



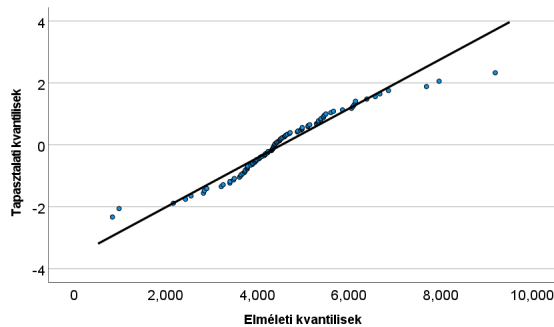
(a)



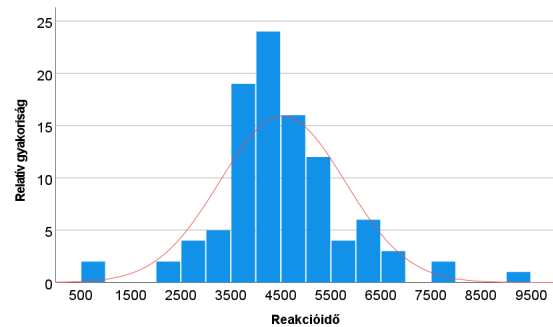
(b)

10. ábra. A pupillaátmérőjének normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).

Forrás: Saját készítésű ábra.



(a)



(b)

11. ábra. A reakcióidők normál eloszlás-ellenőrzése (a) és gyakorisági hisztogramja (b).

Forrás: Saját készítésű ábra.

11. táblázat. Az eloszlás alakjának felmérése a Visual Search Teszt eredmények esetében.

Visual Search Teszt						
Paraméter	Shapiro-Wilk			Átlag $\pm$ SD	Medián	Skewness (ferdeség)
	Próbastatisztika	p-érték	N			
Fixációk száma	0,953	0,001*	100	493,23 $\pm$ 258,365	495,5	0,361
Fixációk Időtartama	0,941	<0,001*	100	217,533 $\pm$ 65,987	206,6	1,026
Pupillaátmérő	0,904	<0,001*	100	25,01 $\pm$ 4,20	23,98	1,418
Reakcióidő	0,952	0,001*	100	4531,61 $\pm$ 1253,948	4383	0,455

* 2-oldali

5.2.3. A Stroop-Teszt eredményeinek összevetése a szemmozgásparaméterekkel

A következő alfejezetben összevetésre kerülnek a Stroop-Teszt eredményei (helyes válaszok száma, átlagos reakcióidő) a 3 kiemelt szemmozgás paraméterrel (fixációk száma, átlagos fixációs idő és a pupillaátmérő).

Pupillaátmérő összevetése az átlagos reakcióidővel

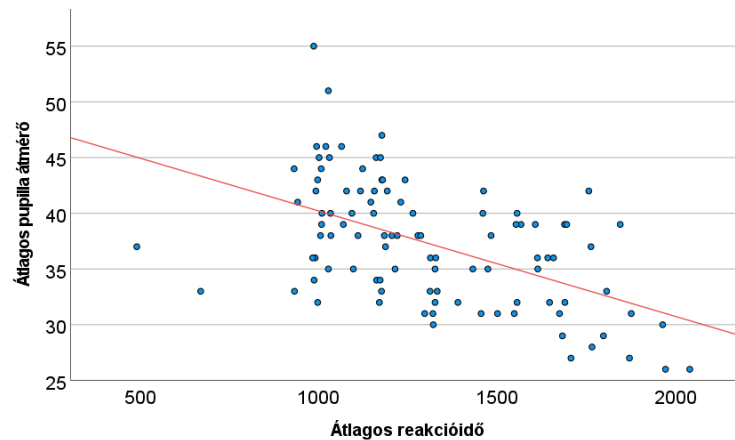
A normalitás vizsgálat során megállapítottuk, hogy míg a pupillaátmérő ($W = 0,985, p = 0,317$) normál eloszlású, addig a reakcióidő ($W = 0,955, p = 0,002$) nem normál eloszlású. Ezért a pupillaátmérő és a reakcióidő korreláció vizsgálata során a nem parametrikus, Spearman-féle rangkorreláció vizsgálatot végeztük el (12. táblázat). A vizsgálat elvégzése után megállapítottuk, hogy a két adat között jelentős, közepes, negatív korreláció áll fenn ($r_s = -0,514, p < 0,001$). Ez arra utal, hogy a pupillaátmérőjének növekedése a reakcióidő csökkenésével jár együtt.

12. táblázat. Spearman-féle rangkorreláció eredményei a pupillaátmérőre és a reakcióidőre a Stroop-Tesztben.

	Pupillaátmérő	Reakcióidő
Pupillaátmérő	1,000	-0,514**
Reakcióidő	-0,514**	1,000

** A korreláció szignifikáns a 0,01-es szinten (kétoldali).

A 12. ábrán az átlagos pupillaátmérő és az átlagos reakcióidő közötti kapcsolatot vizsgáljuk pont-felhődiagram formájában. Az ábra egyértelműen negatív irányú kapcsolatot mutat a két változó között, amit a piros lineáris regressziós egyenes is alátámaszt. Ez a vizuális ábrázolás is összhangban áll a Spearman-féle rangkorrelációs elemzés eredményével.



12. ábra. Az átlagos reakcióidő és az átlagos pupillaátmérő pontfelhődiagramja.

Forrás: Saját készítésű ábra.

A fixációk számának összevetése a helyes válaszok számával

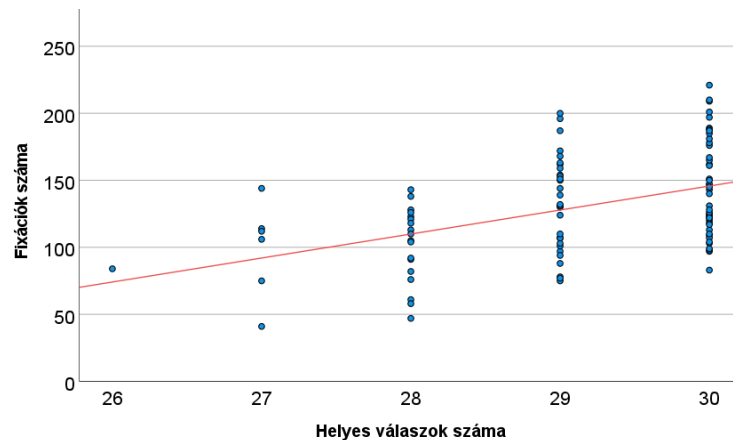
A normalitás vizsgálat megállapította, hogy a fixációk száma ($W = 0.980, p = 0.133$) normál eloszlású, addig a helyes válaszok száma ($W = 0.814, p = < 0.001$) nem a normál eloszlást követi. számú ábra. Ezért a Spearman-féle rangkorrelációt végeztük el az adatokon (13. táblázat). Az elemzés eredményei alapján megállapítható, hogy az adatok között szignifikáns, közepes pozitív korreláció áll fenn ($r_s = 0.420, p < 0.001$). Ez azt jelenti, hogy a fixációk növekedése együtt jár a helyes válaszok számának növekedésével.

13. táblázat. Spearman-féle rangkorreláció eredményei a fixációk számára és a helyes válaszok számára a Stroop-Tesztben.

	Fixációk száma	Helyes válaszok száma
Fixációk száma	1,000	0,420**
Helyes válaszok száma	0,420**	1,000

** A korreláció szignifikáns a 0,01-es szinten (kétoldali).

A 13. ábra a fixációk száma, illetve a helyes válaszok száma közötti kapcsolatot szemlélteti pontfelhődiagram formájában. A diagramon egy pozitív irányú trend figyelhető meg, amelyet a piros lineáris regressziós egyenes is jól mutat.



13. ábra. A helyes válaszok száma és a fixációk számának pontfelhődiagramja.

Forrás: Saját készítésű ábra.

A fixációs időtartam összevetése a helyes válaszok számával

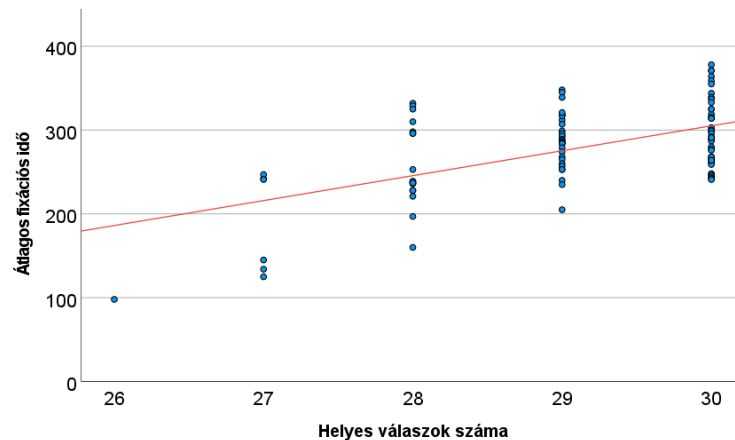
A normalitásvizsgálat eredményei azt mutatták, hogy sem a fixációk időtartama ($W = 0,913, p < 0,001$), sem pedig a helyes válaszok száma ($W = 0,814, p < 0,001$) nem követi a normális eloszlást. Ezért ennek megfelelően itt is a Spearman-féle rangkorrelációt alkalmaztuk az adatok elemzésére (14. táblázat). Az elemzés eredményei alapján jelentős, közepes, pozitív kapcsolat áll fenn a helyes válaszok száma és a fixációk időtartama között ($r_s = 0,410, p < 0,001$). Ez arra utal, hogy a fixációk időtartamának növekedése együtt jár a helyes válaszok számának növekedésével.

14. táblázat. Spearman-féle rangkorreláció eredményei a fixációk időtartamára és a helyes válaszok számára a Stroop-Tesztben.

	Fixációk Időtartama	Helyes válaszok száma
Fixációk időtartama	1,000	0,410**
Helyes válaszok száma	0,410**	1,000

** A korreláció szignifikáns a 0.01-es szinten (kétoldali).

A 14. ábra mutatja az átlagos fixációs idő illetve a helyes válaszok között fennálló kapcsolatot pontfelhődiagram formájában. Az ábra egy pozitív irányú trendet mutat, amit a piros regressziós vonal is szemléltet.



14. ábra. A helyes válaszok száma és a fixációk átlagos időtartamának pontfelhődiagramja.

Forrás: Saját készítésű ábra.

5.2.4. A Visual Search eredményeinek összevetése a szemmozgás paraméterekkel

A következő alfejezetben összevetésre kerülnek a Visual Search Teszt eredményei (helyes válaszok száma, átlagos reakcióidő) a 3 kiemelt szemmozgás paraméterrel (fixációk száma, átlagos fixációs idő és a pupillaátmérő).

Átlagos pupillaátmérő összevetése a reakcióidővel

Az átlagos pupillaátmérő vizsgálata során meghatározásra került, hogy adatok normalitása nem áll fenn ($W = 0,904, p < 0,001$), ugyanúgy mint a teszt átlag reakció idejénél ($W = 0,952, p = 0,001$). Ezért az átlagos pupillaátmérő és az átlag reakcióidő között nem parametrikus, Spearman-féle rangkorreláció számítás került elvégzésre (15. táblázat). A két adat között jelentős közepes kapcsolat illetve negatív korreláció mutatkozott, amely statisztikailag szignifikáns volt ($r_s = -0,470, p < 0,001$). Ez az eredmény arra utal, hogy a nagyobb pupillaátmérő rövidebb reakcióidővel párosul.

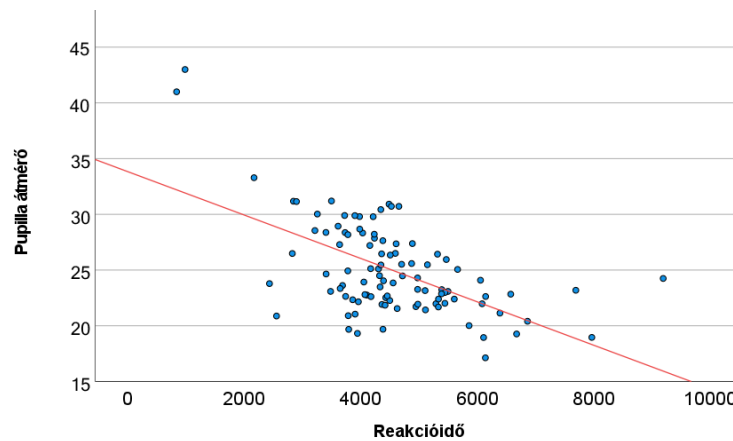
15. táblázat. Spearman-féle rangkorreláció eredményei a pupillaátmérőre és a reakcióidőre a Visual Search Teszten.

	Pupillaátmérő	Reakcióidő
Pupillaátmérő	1,000	-0,470**
Reakcióidő	-0,470**	1,000

** A korreláció szignifikáns a 0,01-es szinten (kétoldali).

A 15. ábra a pupillaátmérő és a reakcióidő közötti kapcsolatot reprezentálja pontfelhődiagram

formájában. A diagramon egy jól látható negatív irányú trend vehető ki, amelyet a piros lineáris regressziós egyenes is kitűnően szemléltet.



15. ábra. A pupillaátmérők és a reakcióidők pontfelhődiagramja.

Forrás: Saját készítésű ábra.

A fixációk számának összevetése a helyes válaszok számával

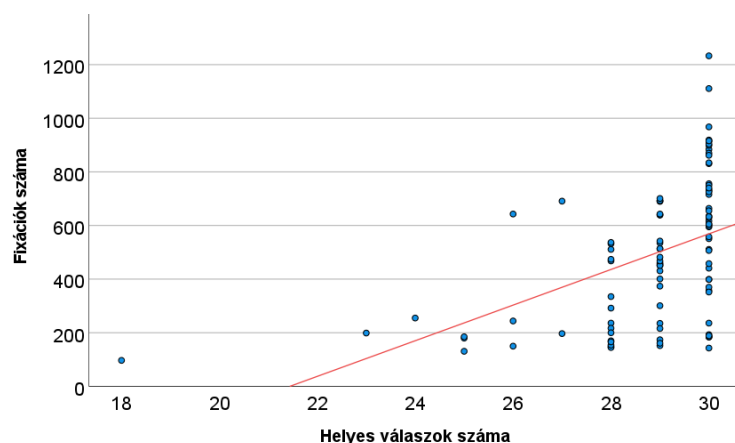
A fixációk száma és a helyes válaszok vizsgálata során a normalitásvizsgálat megerősítette, hogy az eredmények alapján a fixációk száma ($W = 0,953, p = 0,001$) illetve a helyes válaszok száma ($W = 0,642, p < 0,001$) nem követte a normál eloszlást. Emiatt a két változó közötti kapcsolat elemzésére nem paraméteres Spearman-féle rangkorreláció került alkalmazásra (16. táblázat). A korrelációs elemzés eredményei alapján a fixációk száma és a helyes válaszok száma között szignifikáns, közepes erősségű pozitív kapcsolat mutatkozott ($r_s = 0,521, p < 0,001$). Ez az eredmény azt jelenti, hogy a fixációk számának növekedésével együtt nő a helyes válaszok száma.

16. táblázat. Spearman-féle rangkorreláció eredményei a fixációk számára és a helyes válaszok számára a Visual Search Teszten.

	Fixációk száma	Helyes válaszok száma
Fixációk száma	1,000	0,521**
Helyes válaszok száma	0,521**	1,000

** A korreláció szignifikáns a 0,01-es szinten (kétoldali).

A fixációk száma és a helyes válaszok száma közötti kapcsolatot a 16. ábra szemlélteti. A pontfelhődiagram egyértelmű pozitív irányú kapcsolatot mutat, amelyet a piros lineáris regressziós egyenes is alátámaszt.



16. ábra. A fixációk számának és a helyes válaszok számának pontfelhődiagramja.

Forrás: Saját készítésű ábra.

A fixációk időtartamának összevetése a helyes válaszok számával

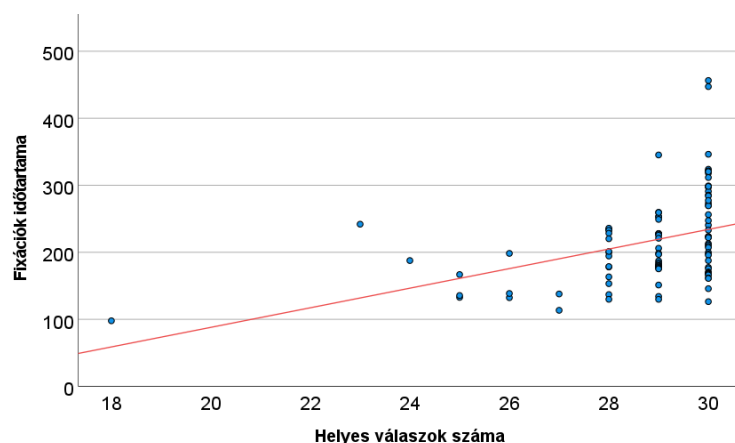
A fixációk időtartamának elemzése alapján kiderült, hogy adatok normalitása nem áll fenn ($W = 0,941, p < 0,001$), ugyanúgy mint a teszt helyes válaszainak számánál ($W = 0,642, p < 0,001$). Ezért a nem parametrikus, Spearman-féle rangkorrelációt alkalmaztuk (17. táblázat). A két adat között szignifikáns, közepes, pozitív korreláció fedezhető fel ($r_s = 0,414, p < 0,001$). Ez az eredmény azt tükrözi, hogy a fixációk időtartamának növekedése együtt jár a helyes válaszok számának növekedésével.

17. táblázat. Spearman-féle rangkorreláció eredményei a fixációk időtartamára és a helyes válaszok számára a Visual Search Teszten.

	Fixációk Időtartama	Helyes válaszok száma
Fixációk időtartama	1,000	0,414**
Helyes válaszok száma	0,414**	1,000

** A korreláció szignifikáns a 0,01-es szinten (kétoldali).

A 17. ábrán látható a fixációk időtartamának és a helyes válaszok számának kapcsolata pontfelhődiagramon ábrázolva. Az ábra egyértelmű pozitív irányú kapcsolatot mutat, amelyet a piros lineáris regressziós egyenes is megerősít.



17. ábra. A fixációk időtartamának és a helyes válaszok számának pontfelhődiagramja.

Forrás: Saját készítésű ábra.

5.2.5. Az eredmények összefoglalása

Az összes tesztek és a szemmozgásparaméterek korrelációs kiértékelését a 18. táblázat foglalja össze.

18. táblázat. A tesztek eredményeinek és a szemmozgásparaméterek korrelációs kiértékelését összesítő táblázat.

Paraméterek	Visual Search Teszt		Stroop Teszt	
	Korreláció	p-érték	Korreláció	p-érték
Fixációs mennyiség & Helyes válaszok száma	0,521	<0,001*	0,420	<0,001*
Fixációs időtartam & Helyes válaszok száma	0,414	<0,001*	0,410	<0,001*
Pupillaátmérő & Reakcióidő	-0,470	<0,001*	-0,514	<0,001*

* 2-oldali

5.3. Az eredményekből levonható következtetések

Az eredmények részletes vizsgálata alapján megállapítható, hogy a Stroop és Visual Search tesztek során gyűjtött adatok között jelentős és statisztikailag szignifikáns összefüggések mutatkoznak, amelyek lehetőséget teremtenek a kognitív folyamatok és a szemmozgás-paraméterek kapcsolatának mélyebb megértésére. A Stroop-teszt esetében a fixációk száma és a helyes válaszok száma közötti közepes pozitív korreláció arra utal, hogy a szemmozgások intenzitása összefügg a feladat teljesítésével. Hasonlóan, a fixációk időtartama is kapcsolatot mutatott a helyes válaszokkal, ami a figyelem fókuszálásának fontosságát emeli ki az információfeldolgozás során. A pupillaátmérő és a reakcióidő között mért negatív korreláció pedig arra utal, hogy a kognitív terhelés fokozódása

gyorsabb válaszidőkkel társulhat, amit a pupilla méretének változása is tükröz. A Visual Search tesztek eredményei szintén hasonló tendenciákat mutattak, azonban itt a normalitás hiánya miatt nem parametrikus módszerek alkalmazására volt szükség. Az eredmények alapján a szemmozgási paraméterek, például a fixációk intenzitása és időtartama, erős összefüggést mutattak a kognitív teljesítménymutatókkal. A nagyobb pupillaátmérő a gyorsabb reakcióidővel társult, ami a fokozott koncentráció jeleként értelmezhető. Ezek az eredmények nemcsak az ilyen jellegű tesztekben alkalmazhatóak, hanem rámutatnak arra is, hogy hasonló szemmozgási és kognitív mutatók nyomon követésével olyan komplex feladatok, mint például a programozás folyamata is vizsgálható. A programozás során hasonló kognitív mechanizmusok játszanak szerepet [11] [79], mint a Stroop- vagy Visual Search tesztek esetében: a figyelem fenntartása, az információk hatékony feldolgozása és a gyors döntéshozatal. Ezeket a jellegzetességeket és összefüggéseket több kutatásban is elemezték, vizsgálták és az itt kapott eredményekkel is összefüggésbe hozhatók [80] [81] [82] [83] [84] [85] [40] [86] [87] [88] [89] [90] [91] [92]. Ezek alapján a szemmozgás-paraméterek és a pupillaátmérő változásai segítségével betekintést nyerhetünk a programozási folyamatok figyelmi és kognitív aspektusaiba, ami új lehetőségeket nyithat a kognitív terhelés mérésében és optimalizálásában.

Az élő adat részletező modul lehetővé teszi különböző valós idejű megfigyelések megvalósítását, amelyek során a vizuális figyelem által közvetített különböző ingerek hatásai elemezhetőek. A modulban egy integrált képernyörögzítő funkció is helyet kapott, amely lehetővé teszi az aktuális tekintési pontok pontos követését a vizsgálatok során. Ez a funkció különösen hasznos olyan kutatások esetében, amelyek a résztvevők figyelmi mintázatainak részletes elemzésére irányulnak. A rendszer alkalmazása során alkalmunk nyílt megfigyelni, hogyan hajtják végre a résztvevők a PEBL-alapú kognitív feladatokat. Az valós idejű adatgyűjtés és a vizuális figyelem elemzése segítségével azonosíthatók voltak azok a feladatok és helyzetek, amelyek nagyobb kihívást jelentettek a résztvevők számára. Ez lehetőséget adott annak feltárására, hogy milyen típusú változások vagy kognitív terhelések okozták a teljesítmény csökkenését vagy lassulását. A modul további előnye, hogy a valós idejű adatfeldolgozásnak köszönhetően dinamikus elemzéseket lehet végezni, így a vizsgálatok során nemcsak a végső eredmények, hanem a teljesítmény közben fellépő változások is értékelhetőek. Ezáltal részletesebb képet kaptunk arról, hogyan reagálnak a résztvevők különböző ingerekre, és hogyan módosul figyelmük a feladat nehézségének változásával. Ezek az információk különösen hasznosak lehetnek a kognitív folyamatok és a vizuális figyelem kapcsolatának mélyebb megértésében.

6. A kutatás korlátai

A kutatás során alkalmazott szemmozgáskövető eszközök implementálásában jelentős akadályt jelentett a különböző eszközök magas költsége. A Tobii Pro Nano és a The Eye Tribe technológiák integrációja mellett számos modern eszköz, például az EyeLink 1000 Plus vagy a Gazepoint 3, az anyagi forrásaink szűkössége miatt nem kerülhetett felhasználásra. Ahhoz, hogy az alkalmazásunk képes legyen ezen kívül több eszközt is támogatni, elengedhetetlen lenne a gyártók lelkes és aktív támogatása, például maga az eszköz, API-k és dokumentációk biztosítása formájában. Alternatív megoldásként a közösségi fejlesztés bevonása is megoldást jelenthetne, ahol nyílt forráskódú platformokon keresztül dolgozhatnánk együtt a szükséges funkciók kidolgozásán. Saját forrásokból ezeknek az eszközöknek a beszerzése és integrációja sajnos nem kivitelezhető, ami korlátozza a kutatásban elérhető lehetőségeinket és az alkalmazásunk fejlesztésének irányát.

A kutatásban résztvevő alanyok száma és demográfiai jellemzői befolyásolhatják az eredmények általánosíthatóságát. A vizsgálatokat önkéntes alapon, főként egy szűk korcsoportot érintve végeztük, ami nem reprezentálja teljes mértékben a különböző populációkat. Továbbá az egyéni különbségek, például a látásélesség, figyelmi állapot és a stressz szintje, szintén torzíthatják a szemmozgási paraméterekre vonatkozó adatokat. Annak érdekében, hogy ezen korlátok hatását mérsékeljük, igyekeztünk a vizsgálatok során egy kérdőív alkalmazásával ellenőrizni a résztvevők állapotát, amelyek csökkenthetik a mintán belüli eltérésekből adódó torzításokat.

7. Összefoglalás

A fenti fejezetekben megismerkedtünk a szemmozgáskövetéssel, kitértünk külön a PCCR (Pupilla Központú Szaruhártya Tükröződés) működési elvre, ismertetve lett az alkalmazási területek (marketing, szórakozás és kutatás), megtekintettük a piacon lévő számos szemmozgáskövető eszközöket és azoknak a hardveres követelményeit, egyedi funkcióit. Ahhoz, hogy ezeket az eszközöket százszázalékosan ki tudjuk használni elengedhetetlen valamilyen erre készült alkalmazás, azonban csak nagyon drágák vagy korlátozott támogatottságú szoftverek érhetőek el az eszközök kezelésére. Korábbi kutatásban használtuk az OGAMA (Open Gaze And Mouse Analyzer) programot, ami egy ingyenes szemmozgás paraméter analízátor, viszont ahogy ismertetve lett rengeteg hibája van, korlátai, az újabb eszközünket (például a Tobii Pro Nano eszközünket) nem támogatja, azok implementálása nagyon időigényes folyamat. Ennek okán megvalósítottunk egy saját fejlesztésű szoftvert és összehasonlítottuk az OGAMA-val, majd a funkcióit (adatrögzítés, statisztika készítés, adatmentés) validáltuk, ennek eredményül bizonyítottuk, hogy alkalmas-e kutatási célokra a saját fejlesztésű alkalmazásunk. Úgy gondoljuk, hogy a fejlesztés, tesztelés, és kutatás során megtanultunk csapatban dolgozni, és együtt olyan dolgokat létrehozni, melyekre külön-külön nem lettünk volna képesek. Javult a problémamegoldó képességünk, hiszen folyamatos kihívásokkal találkoztunk, melyek megoldásához kreatív ötletekkel kellett előállnunk.

Kutatásunk során összefoglalva, az előző fejezetekben az alábbiakat sikerült végrehajtanunk:

1. Megismertük számos hazai és nemzetközi forrásokból a szemmozgáskövetők lehetséges működési elveit és bemutattuk azokat.
2. Felkutattunk számos szemmozgáskövető eszközt, abból kettőt (Eye Tribe, Tobii Pro Nano) implementáltunk egy saját fejlesztésű alkalmazásunkba, ahol minden létszükségletű funkciókat (eszköz kommunikáció, kalibráció, adatrögzítés, statisztika készítés, adatmentés) elkészítettünk a követelmény-specifikációban felsoroltak alapján.
3. Az alkalmazásfejlesztés során figyelembe vettük a modern szoftverfejlesztési technikákat, továbbá korszerű harmadék féltől származó modulokat csatoltunk az alkalmazáshoz. Figyeltünk a modularitásra SOLID elvek alapján (például új eszköz hozzáadása esetén nem kell már meglévő osztályokat módosítani, elég csak létrehozni újat és gyűjtő osztályba hozzáadni az új eszköz osztályait).
4. Az alkalmazásunk funkcióit validáltuk egy korábbi kutatásunkban is használt ingyenesen elérhető szemmozgás elemző szoftver alapján, moduljait összehasonlítottuk a mi alkalmazásunkkal.

5. A SightScan szoftverünk segítségével több résztvevővel végeztünk el különböző PEBL-alapú kognitív tesztek, amelyek lehetővé tették a szemmozgási paraméterek összehasonlítását a figyelmet és éberséget mérő tesztek eredményeivel.
6. A PEBL tesztek futtatásához egy saját fejlesztésű szoftvert készítettünk el, hogy a kutatáshoz szükséges eredményeket adjon vissza és igazodjon a kutatási feltételekhez.
7. Az elkészült vizsgálatokat kiértékeljük szemmozgás paraméterek függvényében, melyekből következtetéseket vontunk le.

A szoftver képes valós időben rögzíteni a szemmozgásokat, és az adatokat statisztikai módszerekkel elemezni. A Stroop-teszt és a Visual Search feladatok segítségével végzett vizsgálatok során a szoftver megbízhatónak és pontosnak bizonyult, miközben lehetőséget biztosított a kognitív folyamatok mélyebb megértésére. Az eredmények elemzése alapján megállapítható volt, hogy közepes erősségű korreláció mutatható ki az éberség és koncentráció tesztek, valamint a szemmozgást jellemző paraméterek között. Ezek az összefüggések lehetőséget adnak az éberségi szint és a koncentráció mértékének értelmezésére.

A validálás szintén sikeres volt, így felmerül a kérdés, hogy a saját fejlesztésű alkalmazásunk alkalmas-e kutatási célokra. Az SightScanre - mint minden további más alkalmazásra - további optimalizációk, javítások szükségesek, továbbá új funkciók hozzáadása is egy meglévő igény. Korábbi kutatásunkban leírt cél („A jövőbeni cél egy olyan saját alkalmazás elkészítése, amely OGAMA szoftvert teljes mértékben kiváltja, továbbá a mai, korszerű technológiákat alkalmazza.”) úgy gondoljuk teljesült. Így új jövőbeli célt tűztünk ki: a szoftverünknek képesnek kell lennie interakcióra különböző hordozható digitális eszközökkel, mint a mobiltelefonok. Ezzel szeretnénk lehetővé tenni olyan kutatásokat, amelyeket okoseszközökkel le lehet bonyolítani számítógépek helyett, a saját alkalmazásunk segítségével.

Hivatkozások

- [1] R. A. Armstrong, „Alzheimer’s Disease and the Eye”, en, *Journal of Optometry*, 2. évf., 3. sz., 103–111. old., 2009, ISSN: 18884296. DOI: 10.3921/joptom.2009.103.
- [2] D. Jansson, A. Medvedev, H. Axelson és D. Nyholm, „Stochastic Anomaly Detection in Eye-Tracking Data for Quantification of Motor Symptoms in Parkinson’s Disease”, en, *Signal and Image Analysis for Biomedical and Life Sciences*, C. Sun, T. Bednarz, T. D. Pham, P. Vallotton és D. Wang, szerk., 823. köt., Series Title: Advances in Experimental Medicine and Biology, Cham: Springer International Publishing, 2015, 63–82. old., ISBN: 978-3-319-10983-1 978-3-319-10984-8. DOI: 10.1007/978-3-319-10984-8_4.
- [3] L. Tao, Q. Wang, D. Liu, J. Wang, Z. Zhu és L. Feng, „Eye tracking metrics to screen and assess cognitive impairment in patients with neurological disorders”, en, *Neurological Sciences*, 41. évf., 7. sz., 1697–1704. old., 2020. júl., ISSN: 1590-1874, 1590-3478. DOI: 10.1007/s10072-020-04310-y.
- [4] S. K. Mannan, T. L. Hodgson, M. Husain és C. Kennard, „Eye movements in visual search indicate impaired saliency processing in Parkinson’s disease”, en, *Progress in Brain Research*, 171. köt., Elsevier, 2008, 559–562. old., ISBN: 978-0-444-53163-6. DOI: 10.1016/S0079-6123(08)00679-1.
- [5] M. Okano és M. Asakawa, „Eye tracking analysis of consumer’s attention to the product message of web advertisements and TV commercials”, *2017 5th International Conference on Cyber and IT Service Management (CITSM)*, Denpasar, Bali, Indonesia: IEEE, 2017. aug., 1–5. old., ISBN: 978-1-5386-2739-6. DOI: 10.1109/CITSM.2017.8089270.
- [6] T.-A. Nguyen, C. K. Coursaris, P.-M. Léger, S. Sénécal és M. Fredette, „Effectiveness of Banner Ads: An Eye Tracking and Facial Expression Analysis”, en, *HCI in Business, Government and Organizations*, F. F.-H. Nah és K. Siau, szerk., 12204. köt., Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, 445–455. old., ISBN: 978-3-030-50340-6 978-3-030-50341-3. DOI: 10.1007/978-3-030-50341-3_34.
- [7] R. Harimurti, E. Ekohariadi, M. Munoto, I. G. P. Asto B és E. T. Winanti, „Analysis of Programming Skills Concept in Developing Problem Solving Skills”, *Jurnal Pendidikan Teknologi dan Kejuruan*, 25. évf., 1. sz., 43–51. old., 2019. ápr., ISSN: 0854-4735. DOI: 10.21831/jptk.v25i1.22638.
- [8] S. Y. Lye és J. H. L. Koh, „Review on teaching and learning of computational thinking through programming: What is next for K-12?”, *Computers in Human Behavior*, 41. évf., 51–61. old., 2014. dec., ISSN: 0747-5632. DOI: 10.1016/j.chb.2014.09.012.
- [9] K. M. Yusoff, N. Sahari, T. Siti és N. Mohd, „Analysis on the Requirements of Computational Thinking Skills to Overcome the Difficulties in Learning Programming”, *International*

- Journal of Advanced Computer Science and Applications*, 11. évf., 3. sz., 2020, ISSN: 2158-107X. DOI: 10.14569/ijacsa.2020.0110329.
- [10] R. Amirova, G. Dlamini, A. Repryntseva, G. Succi és H. Tarasau, „Attention and Concentration for Software Developers”, *IEEE Access*, 11. évf., 96196–96205. old., 2023, ISSN: 2169-3536. DOI: 10.1109/access.2023.3309414.
 - [11] I. Fronza, A. Hellas, P. Ihanola és T. Mikkonen, „An Exploration of Cognitive Shifting in Writing Code”, *Proceedings of the ACM Conference on Global Computing Education, CompEd '19* sor., ACM, 2019. máj., 65–71. old. DOI: 10.1145/3300115.3309522.
 - [12] A. Thapaliya, „Evaluation of brain activity while pair programming”, *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings*, ICSE '20 sor., ACM, 2020. jún., 104–106. old. DOI: 10.1145/3377812.3382161.
 - [13] T. T. Brunyé és A. L. Gardony, „Eye tracking measures of uncertainty during perceptual decision making”, en, *International Journal of Psychophysiology*, 120. évf., 60–68. old., 2017. okt., ISSN: 01678760. DOI: 10.1016/j.ijpsycho.2017.07.008.
 - [14] K. Holmqvist, M. Nyström és F. Mulvey, „Eye tracker data quality: what it is and how to measure it”, *Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA '12* sor., ACM, 2012. márc. DOI: 10.1145/2168556.2168563.
 - [15] S. Hutton, „Eye Tracking Methodology”, *Eye Movement Research*, 2019. DOI: 10.1007/978-3-030-20085-5_8.
 - [16] J. Borah, *Eye Movement, Measurement Techniques for*, 2006. ápr. DOI: 10.1002/0471732877.emd112.
 - [17] L. Matin, „Measurement of Eye Movements by Contact Lens Techniques: Analysis of Measuring Systems and Some New Methodology For Three-Dimensional Recording*”, *Journal of the Optical Society of America*, 54. évf., 8. sz., 1008. old., 1964. aug., ISSN: 0030-3941. DOI: 10.1364/josa.54.001008.
 - [18] Uniwersytet Warszawski és M. Płużyczka, „The First Hundred Years: a History of Eye Tracking as a Research Method”, *Applied Linguistics Papers*, 4/2018. évf., 25. sz., 101–116. old., 2018. dec., ISSN: 25449354. DOI: 10.32612/uw.25449354.2018.4.pp.101-116.
 - [19] A. Fletcher, S. Dunne és J. Butler, *A Brief History of Eye Movement Research*, 2021. ápr. 14. DOI: 10.31234/osf.io/yb4zq. (elérés dátuma 2024. 11. 17.).
 - [20] E. Guestrin és M. Eizenman, „General Theory of Remote Gaze Estimation Using the Pupil Center and Corneal Reflections”, en, *IEEE Transactions on Biomedical Engineering*, 53. évf., 6. sz., 1124–1133. old., 2006. jún., ISSN: 0018-9294. DOI: 10.1109/TBME.2005.863952.

- [21] C. H. Morimoto és M. R. Mimica, „Eye gaze tracking techniques for interactive applications”, *Computer Vision and Image Understanding*, 98. évf., 1. sz., 4–24. old., 2005. ápr., ISSN: 1077-3142. DOI: 10.1016/j.cviu.2004.07.010.
- [22] T. Eggert, „Eye movement recordings: methods.”, *Developments in ophthalmology*, 40. évf., 15–34. old., 2007. DOI: 10.1159/000100347.
- [23] J. Zou, H. Zhang és T. Weng, „New 2D pupil and spot center positioning technology under real — Time eye tracking”, *2017 12th International Conference on Computer Science and Education (ICCSE)*, 110–115. old., 2017. DOI: 10.1109/ICCSE.2017.8085473.
- [24] J. Park és J. B. Park, „A novel approach to the low cost real time eye mouse”, *Comput. Stand. Interfaces*, 44. évf., 169–176. old., 2016. DOI: 10.1016/j.csi.2015.04.005.
- [25] F. Li, S. Munn és J. Pelz, „A model-based approach to video-based eye tracking”, *Journal of Modern Optics*, 55. évf., 503–531. old., 2008. DOI: 10.1080/09500340701467827.
- [26] D.-C. Cho, W.-S. Yap, H. Lee, I. Lee és W. Kim, „Long range eye gaze tracking system for a large screen”, *IEEE Transactions on Consumer Electronics*, 58. évf., 2012. DOI: 10.1109/TCE.2012.6414976.
- [27] S. Eivazi, T. C. Kübler, T. Santini és E. Kasneci, „An inconspicuous and modular head-mounted eye tracker”, *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, 2018. DOI: 10.1145/3204493.3208345.
- [28] N. Schneider, P. Bex, E. Barth és M. Dorr, „An open-source low-cost eye-tracking system for portable real-time and offline tracking”, 8. old., 2011. DOI: 10.1145/1983302.1983310.
- [29] *Tobii*,” *Tobii AB*, [Online], 2024. cím: <https://www.tobii.com/products/discontinued/tobii-pro-nano>.
- [30] *THEEYETRIBE*,” *The Eye Tribe, 2011-2016*. [Online], 2016. cím: <https://theeyetribe.com/theeyetribe.com/about/index.html>.
- [31] *SR Research EyeLink*,” *SR Research Ltd.*, [Online]. cím: <https://www.sr-research.com/eyelink-1000-plus/>.
- [32] *Gazepoint*,” *Gazepoint*, [Online]. cím: <https://www.gazept.com/product/gazepoint-gp3-eye-tracker/>.
- [33] *Argus Science*,” [Online]. cím: <https://www.argusscience.com/ETVision.html>.
- [34] *Pupil Labs*,” *Pupil Labs GmbH*, [Online]. cím: <https://pupil-labs.com/products/neon>.
- [35] D. Price, D. Kaputa, D. A. Sierra és J. Enderle, „Infrared-based eye-tracker system for saccades”, *2009 IEEE 35th Annual Northeast Bioengineering Conference*, Cambridge, MA, USA:

- IEEE, 2009. ápr., 1–2. old., ISBN: 978-1-4244-4362-8. DOI: 10.1109/NEBC.2009.4967836.
- [36] D. E. I. R. D. Gordon, „Eye Movements, Attention and Trans-saccadic Memory”, *Visual Cognition*, 5. évf., 127–155. old., 1998. DOI: 10.1080/713756783.
- [37] A. Herwig, M. Beisert és W. Schneider, „On the spatial interaction of visual working memory and attention: evidence for a global effect from memory-guided saccades.”, *Journal of vision*, 10 5. évf., 8. old., 2010. DOI: 10.1167/10.5.8.
- [38] S. Negi és R. Mitra, „Fixation duration and the learning process: an eye tracking study with subtitled videos”, *Journal of Eye Movement Research*, 13. évf., 2020. DOI: 10.16910/jemr.13.6.1.
- [39] H. B. Barlow, „Eye movements during fixation”, *The Journal of Physiology*, 116. évf., 3. sz., 290–306. old., 1952. márc., ISSN: 1469-7793. DOI: 10.1113/jphysiol.1952.sp004706.
- [40] M. A. Just és P. A. Carpenter, „Eye fixations and cognitive processes”, *Cognitive Psychology*, 8. évf., 4. sz., 441–480. old., 1976. okt., ISSN: 0010-0285. DOI: 10.1016/0010-0285(76)90015-3.
- [41] E. Kowler, „Attention and Eye Movements”, *Encyclopedia of Neuroscience*. Elsevier, 2009, 605–616. old., ISBN: 9780080450469. DOI: 10.1016/b978-008045046-9.01087-1.
- [42] K. Rayner, „Eye movements in reading and information processing: 20 years of research.”, *Psychological Bulletin*, 124. évf., 3. sz., 372–422. old., 1998, ISSN: 0033-2909. DOI: 10.1037/0033-2909.124.3.372.
- [43] D. D. Salvucci, „An integrated model of eye movements and visual encoding”, *Cognitive Systems Research*, 1. évf., 4. sz., 201–220. old., 2001. febr., ISSN: 1389-0417. DOI: 10.1016/s1389-0417(00)00015-2.
- [44] M. O. Al-Moteri, M. Symmons, V. Plummer és S. Cooper, „Eye tracking to investigate cue processing in medical decision-making: A scoping review”, *Computers in Human Behavior*, 66. évf., 52–66. old., 2017. jan., ISSN: 0747-5632. DOI: 10.1016/j.chb.2016.09.022.
- [45] M. D. Grima-Murcia, F. Sanchez-Ferrer, J. M. Ramos-Rincón és E. Fernández, „Use of Eye-Tracking Technology by Medical Students Taking the Objective Structured Clinical Examination: Descriptive Study”, *Journal of Medical Internet Research*, 22. évf., 8. sz., e17719., 2020. aug., ISSN: 1438-8871. DOI: 10.2196/17719.
- [46] W. Jones, C. Klaiman, S. Richardson és tsai., „Eye-Tracking–Based Measurement of Social Visual Engagement Compared With Expert Clinical Diagnosis of Autism”, *JAMA*, 330. évf., 9. sz., 854. old., 2023. szept., ISSN: 0098-7484. DOI: 10.1001/jama.2023.13295.

- [47] H. Khachatryan és A. L. Rihn, „Eye-Tracking Methodology and Applications in Consumer Research: FE947/FE947, 7/2014”, *EDIS*, 2014. évf., 7. sz., 2014. szept., ISSN: 2576-0009. DOI: 10.32473/edis-fe947-2014.
- [48] H. Zamani, A. M. Abas és M. Amin, „Eye Tracking Application on Emotion Analysis for Marketing Strategy”, *Journal of Telecommunication, Electronic and Computer Engineering*, 8. évf., 87–91. old., 2016.
- [49] K. Duerschmid és L. Danner, „Eye Tracking in Consumer Research”, *Methods in Consumer Research, Volume 2*. Elsevier, 2018, 279–318. old., ISBN: 9780081017432. DOI: 10.1016/b978-0-08-101743-2.00012-1.
- [50] P. T. Huddleston, B. K. Behe, C. Driesener és S. Minahan, „Inside-outside: Using eye-tracking to investigate search-choice processes in the retail environment”, *Journal of Retailing and Consumer Services*, 43. évf., 85–93. old., 2018. júl., ISSN: 0969-6989. DOI: 10.1016/j.jretconser.2018.03.006.
- [51] Z. Yang, Y. Zhang, M. Li és T. Chen, „The Comparison Study of Usability Test Methodology Based on Eye-Tracking Technology”, *Man–Machine–Environment System Engineering*. Springer Singapore, 2017. aug., 763–772. old., ISBN: 9789811062322. DOI: 10.1007/978-981-10-6232-2_91.
- [52] J. Fu, „Usability evaluation of software store based on eye-tracking technology”, *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, IEEE, 2016. máj., 1116–1119. old. DOI: 10.1109/itnec.2016.7560538.
- [53] M. Keskin, K. Ooms, A. O. Dogru és P. De Maeyer, „Exploring the Cognitive Load of Expert and Novice Map Users Using EEG and Eye Tracking”, *ISPRS International Journal of Geo-Information*, 9. évf., 7. sz., 429. old., 2020. júl., ISSN: 2220-9964. DOI: 10.3390/ijgi9070429.
- [54] A. R. Nikolaev, S. Pannasch, J. Ito és A. V. Belopolsky, „Eye movement-related brain activity during perceptual and cognitive processing”, *Frontiers in Systems Neuroscience*, 8. évf., 2014. ápr., ISSN: 1662-5137. DOI: 10.3389/fnsys.2014.00062.
- [55] J. Katona, „Analyse the Readability of LINQ Code using an Eye-Tracking-based Evaluation”, *Acta Polytechnica Hungarica*, 18. évf., 1. sz., 193–215. old., 2021, ISSN: 2064-2687. DOI: 10.12700/aph.18.1.2021.1.12.
- [56] M. Raschke, T. Blascheck és M. Burch, „Visual Analysis of Eye Tracking Data”, *Handbook of Human Centric Visualization*. Springer New York, 2013. jún., 391–409. old., ISBN: 9781461474852. DOI: 10.1007/978-1-4614-7485-2_15.
- [57] M. Abualsaud, M. D. Smucker és C. L. A. Clarke, „Visualizing Searcher Gaze Patterns”, *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, CHI-IR '21 sor., ACM, 2021. márc., 295–299. old. DOI: 10.1145/3406522.3446041.

- [58] S. Arndt, C. Turvey és N. C. Andreasen, „Correlating and predicting psychiatric symptom ratings: Spearmans r versus Kendalls tau correlation”, *Journal of Psychiatric Research*, 33. évf., 2. sz., 97–104. old., 1999. márc., ISSN: 0022-3956. DOI: 10.1016/S0022-3956(98)90046-2.
- [59] R. Haining, „Bivariate Correlation with Spatial Data”, *Geographical Analysis*, 23. évf., 3. sz., 210–227. old., 1991. júl., ISSN: 1538-4632. DOI: 10.1111/j.1538-4632.1991.tb00235.x.
- [60] R. C. Martin, *Clean code: a handbook of agile software craftsmanship* (Robert C. Martin series sor.), Repr. Upper Saddle River, NJ Munich: Prentice Hall, 2012, 431 old., ISBN: 978-0-13-235088-4.
- [61] R. C. Martin és R. C. Martin, *Agile software development: principles, patterns, and practices* (Alan Apt series sor.). Upper Saddle River, NJ: Prentice Hall/Pearson Education, 2003, 529 old., ISBN: 978-0-13-597444-5.
- [62] R. C. Martin és R. C. Martin, *Clean architecture: a craftsman's guide to software structure and design* (Robert C. Martin series sor.). London, England: Prentice Hall, 2018, 404 old., OCLC: on1004983973, ISBN: 978-0-13-449416-6.
- [63] Microsoft, „.NET vs. .NET Framework for server apps, 2022.
- [64] M. Learn, „MVVM Demonstration Using C#. cím: <https://learn.microsoft.com/en-us/dotnet/csharp/>.
- [65] S. Chakraborty és P. S. Aithal, „MVVM Demonstration Using C# WPF”, en, *International Journal of Applied Engineering and Management Letters*, 1–14. old., 2023. jan., ISSN: 2581-7000. DOI: 10.47992/IJAEML.2581.7000.0163.
- [66] W. Wongtanuwat és T. Senivongse, „Detection of Violation of MVVM Design Pattern in Objective-C Programs”, en, *Proceedings of the 8th International Conference on Computer and Communications Management*, Singapore Singapore: ACM, 2020. júl., 54–58. old., ISBN: 978-1-4503-8766-8. DOI: 10.1145/3411174.3411193.
- [67] G. Arcos-Medina, J. Menéndez és J. Vallejo, „Comparative Study of Performance and Productivity of MVC and MVVM design patterns”, *KnE Engineering*, 1. évf., 2. sz., 241. old., 2018. jan., ISSN: 2518-6841. DOI: 10.18502/keg.v1i2.1498.
- [68] Sparx Systems: *Enterprise Architect*, 2024. cím: <https://sparxsystems.com/>.
- [69] C. Nagel, *Professional C# 7 and .NET Core 2.0*, en, 1. kiad. Wiley, 2018. ápr., ISBN: 978-1-119-44927-0 978-1-119-54914-7. DOI: 10.1002/9781119549147.
- [70] R. M. Mulligan, M. W. Altom és D. K. Simkin, „User interface design in the trenches: some tips on shooting from the hip”, en, *Proceedings of the SIGCHI conference on Human factors in computing systems Reaching through technology - CHI '91*, New Orleans, Lo-

- uisiana, United States: ACM Press, 1991, 232–236. old., ISBN: 978-0-89791-383-6. DOI: 10.1145/108844.108898.
- [71] C. Garcia, „The Importance of User Interface and User Experience in the Adoption of Whole Slide Images for Clinical Use”, en, *Analytical Cellular Pathology*, 2014. évf., 1–1. old., 2014, ISSN: 2210-7177, 2210-7185. DOI: 10.1155/2014/836176.
- [72] N. Hirasawa, S. Ogata, K. Yamada-Kawai és H. Kasai, „Integration of user interface design process”, en, *Proceedings of the 11th International Conference on Product Focused Software*, Limerick Ireland: ACM, 2010. jún., 39–42. old., ISBN: 978-1-4503-0281-4. DOI: 10.1145/1961258.1961268.
- [73] *Seaborn: Statistical Data Visualization*, 2024. cím: <https://seaborn.pydata.org/>.
- [74] *Pandas: Open source data analysis and manipulation tool*, 2024. cím: <https://pandas.pydata.org/>.
- [75] *Matplotlib: Visualization with Python*, 2024. cím: <https://matplotlib.org/>.
- [76] *ScreenRecorderLib: A .NET library for screen recording in Windows, using Microsoft Media Foundation for realtime encoding to video or image files*. 2024.
- [77] T. Y. R. Destyanto és R. F. Lin, „Detecting computer activities using eye-movement features”, en, *Journal of Ambient Intelligence and Humanized Computing*, 14. évf., 11. sz., 14441–14451. old., 2023. nov., ISSN: 1868-5137, 1868-5145. DOI: 10.1007/s12652-020-02683-8.
- [78] E. C. d. S. Assef, A. G. S. Capovilla és F. C. Capovilla, „Computerized Stroop Test to Assess Selective Attention in Children with Attention Deficit Hyperactivity Disorder”, *The Spanish Journal of Psychology*, 10. évf., 1. sz., 33–40. old., 2007. máj., ISSN: 1988-2904. DOI: 10.1017/s1138741600006296.
- [79] T. Ormerod, „Human Cognition and Programming”, *Psychology of Programming*. Elsevier, 1990, 63–82. old., ISBN: 9780123507723. DOI: 10.1016/b978-0-12-350772-3.50009-4.
- [80] T. T. Brunyé és A. L. Gardony, „Eye tracking measures of uncertainty during perceptual decision making”, *International Journal of Psychophysiology*, 120. évf., 60–68. old., 2017. okt., ISSN: 0167-8760. DOI: 10.1016/j.ijpsycho.2017.07.008.
- [81] M. A. Just, P. A. Carpenter és A. Miyake, „Neuroindices of cognitive workload: Neuroimaging, pupillometric and event-related potential studies of brain work”, *Theoretical Issues in Ergonomics Science*, 4. évf., 1–2. sz., 56–88. old., 2003. jan., ISSN: 1464-536X. DOI: 10.1080/14639220210159735.
- [82] S. Mathôt, „Pupillometry: Psychology, Physiology, and Function”, *Journal of Cognition*, 1. évf., 1. sz., 2018, ISSN: 2514-4820. DOI: 10.5334/joc.18.

- [83] S. Ahern és J. Beatty, „Pupillary Responses During Information Processing Vary with Scholastic Aptitude Test Scores”, *Science*, 205. évf., 4412. sz., 1289–1292. old., 1979. szept., ISSN: 1095-9203. DOI: 10.1126/science.472746.
- [84] S. Białowas és A. Szyszka, „Eye-tracking in Marketing Research”, *Managing Economic Innovations – Methods and Instruments*. Bogucki Wydawnictwo Naukowe, 2019, 91–104. old. DOI: 10.12657/9788379862771-6.
- [85] M. A. Just és P. A. Carpenter, „A theory of reading: From eye fixations to comprehension.”, *Psychological Review*, 87. évf., 4. sz., 329–354. old., 1980, ISSN: 0033-295X. DOI: 10.1037/0033-295x.87.4.329.
- [86] D. Micic, H. Ehrlichman és R. Chen, „Why do we move our eyes while trying to remember? The relationship between non-visual gaze patterns and memory”, *Brain and Cognition*, 74. évf., 3. sz., 210–224. old., 2010. dec., ISSN: 0278-2626. DOI: 10.1016/j.bandc.2010.07.014.
- [87] A. Rizzo, S. Ermini, D. Zanca, D. Bernabini és A. Rossi, „A Machine Learning Approach for Detecting Cognitive Interference Based on Eye-Tracking Data”, *Frontiers in Human Neuroscience*, 16. évf., 2022. ápr., ISSN: 1662-5161. DOI: 10.3389/fnhum.2022.806330.
- [88] F. Daniel és Z. Kapoula, „Induced vergence-accommodation conflict reduces cognitive performance in the Stroop test”, *Scientific Reports*, 9. évf., 1. sz., 2019. febr., ISSN: 2045-2322. DOI: 10.1038/s41598-018-37778-y.
- [89] A. Szulewski, S. M. Fernando, J. Baylis és D. Howes, „Increasing Pupil Size Is Associated with Increasing Cognitive Processing Demands: A Pilot Study Using a Mobile Eye-Tracking Device”, *Open Journal of Emergency Medicine*, 2. évf., 1. sz., 8–11. old., 2014, ISSN: 2332-1814. DOI: 10.4236/ojem.2014.21002.
- [90] R. L. van den Brink, P. R. Murphy és S. Nieuwenhuis, „Pupil Diameter Tracks Lapses of Attention”, *PLOS ONE*, 11. évf., 10. sz., R. K. Mishra, szerk., e0165274., 2016. okt., ISSN: 1932-6203. DOI: 10.1371/journal.pone.0165274.
- [91] S. Negi és R. Mitra, „Fixation duration and the learning process: an eye tracking study with subtitled videos”, *Journal of Eye Movement Research*, 13. évf., 6. sz., 2020. aug., ISSN: 1995-8692. DOI: 10.16910/jemr.13.6.1.
- [92] K. Liu és C. Ka-lun Or, „Effects of Image Segmentation and Target Number on Visual Search: An Eye-Tracking Study”, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 62. évf., 1. sz., 1987–1991. old., 2018. szept., ISSN: 2169-5067. DOI: 10.1177/1541931218621450.