



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

**NEUMANN JÁNOS
INFORMATIKAI KAR**



SZAKDOLGOZAT

OE-NIK

Hallgató neve:

Kmettné Győri Szilvia

2022

Hallgató törzskönyvi száma:

T/008052/F112904/N

Óbudai Egyetem
Neumann János Informatikai Kar
Biomatika és Alkalmazott Mesterséges Intelligencia Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: **Kmettné Győri Szilvia**
Törzskönyvi száma: **T/008052/FI12904/N**
Neptun kódja: 

A dolgozat címe:

Gesztusvezérlés fertőzésveszélyes környezetben
Gesture control in environment with risk of infection

Intézményi konzulens: **Dr. Klespitz József**
Külső konzulens:

Beadási határidő: **2022. május 15.**

A záróvizsga tárgyai: **Informatikai rendszerek üzemeltetésének biztonsága**
IT hálózatbiztonság

A feladat

A mai modern orvostechnikai eszközök elengedhetetlen kelléke a grafikus felhasználói felület (GUI) többnyire érintőkijelzővel párosítva. Az egyszerűbb betanítás és az intuitív használat mellett ezek komoly hátrányt jelentenek fertőzésveszélyes környezetben (pl. infúziós pumpák, hemodialízis készülékek): Az érintéssel a felhasználó/nővér beszennyezheti a készüléket, vagy fordítva arról szed össze valamilyen csírárt. Ezen kívül a meglévő funkciók kibővítésére is lehetőséget ad az, ha a gesztusvezérlés kiterjesztett valósággal párosul.

A diplomaterv célja egy olyan gesztusvezérlés megalkotása, amely képes vezérelni egy grafikus felhasználói felülettel ellátott alkalmazást. Az ily módon elkészített szoftver lehetőséget ad arra, hogy a meglévő készülékeket kibővítve az érintésvezérlést részben vagy egészben feleslegessé tegye, a megelőző fertőzésekkel pedig emberéletek váljanak megmenthetővé.

A dolgozatnak tartalmaznia kell:

- kapcsolódó szakirodalom feldolgozása,
- irányításhoz szükséges gesztusok és környezet meghatározása,
- egy választott gesztusvezérlés megvalósítása,
- a megoldás használhatóságának értékelése.



Dr. Eigner György
mb. intézetigazgató

A szakdolgozat elvégzésének határideje: **2024. május 15.**
(OE TVSz 55.§ szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

.....
intézményi konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun Kód: Tagozat:
Kmettyné Győri Szilvia [REDACTED] Kiberbiztonsági szakmérnök (levelező)
Telefon: Levelezési cím (pl.: lakcím):
[REDACTED]

Projektmunka címe magyarul:
Gesztusvezérlés fertőzésveszélyes környezetben

Projektmunka címe angolul:
Gesture control in environment with risk of infection

Intézményi konzulens: Külső konzulens:
Dr. Klespitz József

Kérjük, hogy az adatokat nyomtatott nagy betűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2022.03.06.	Féléves teendők megbeszélése	[Signature]
2.	2022.03.07.	Elért eredmények megtekintése	[Signature]
3.	2022.05.01.	Diplomamunka első áttekintése	[Signature]
4.	2022.05.06.	Diplomaterv újabb átnézése, véglegesítése	[Signature]

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Projektlabor 1. / 2. / 3. / Záródolgozati projekt¹ tantárgy követelményét teljesítette, beszámolóra/védésre² bocsátható. A javasolt érdemjegy: 5

.....

Intézményi konzulens

Budapest, 2022. 05. 10

¹ A megfelelő bekarikázandó/aláhúzendő!

² A megfelelő aláhúzendő!

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2022. május 8.



hallgató aláírása

KÖSZÖNETNYILVÁNÍTÁS

Ezúton szeretnék köszönetet mondani az Óbudai Egyetem oktatóinak az érdekes és inspiráló előadásokért, melyből nagyon sokat tanultam és köszönöm a segítséget, melyet az ügyintézőktől is kaptam.

Szeretném megköszönni konzulensemnek, Dr. Klespitz Józsefnek a nyújtott támogatást és ösztönzést, amellyel nagy segítséget nyújtott szakdolgozatom elkészítésében.

Végül köszönöm a családomnak a tanuláshoz nyújtott segítséget és támogatásukat, nélkülük nem sikerült volna.

1. TARTALOM

1	ÖSSZEFOGLALÓ.....	8
2	PROBLÉMAFELVETÉS	9
2.1	A gesztusvezérlés technológia.....	9
3	AZ IPARÁGI SZEREPLŐK BEMUTATÁSA.....	16
3.1	Gépjárművekben alkalmazott megoldások	16
3.2	Mobiltelefonok gesztusvezérlése	17
3.3	Szórakoztató eszközök.....	18
3.4	Smart eszközök.....	21
3.5	Egészségügy	22
4	A MEGVALÓSÍTÁSHOZ SZÜKSÉGES KÖRNYEZET.....	26
4.1	A fejlesztőkörnyezet	26
4.2	Kiterjesztett valóság – Augmented Reality.....	26
4.3	Okoseszközök a kiterjesztett valóság megjelenítésér.....	28
5	AZ ALKALMAZÁS MEGVALÓSÍTÁSA.....	30
5.1	Az alkalmazás menüpontjai	33
6	KONKLÚZIÓ - A MEGOLDÁS HASZNÁLHATÓSÁGÁNAK ÉRTÉKELÉSE	42
7	CONCLUSION – THE EVALUATION OF THE USEFULNESS OF SOLUTION	43
8	IRODALOMJEGYZÉK	44
9	ÁBRAJEGYZÉK.....	46
10	KÉPJEGYZÉK	47
11	FORRÁSKÓDOK	48

1 ÖSSZEFOGLALÓ

Jelenleg az egészségügyi rendszerben használt és kivitelezett technológiákkal az eszközök képernyőjét a program vezérléséhez, a megnyitáshoz meg kell érinteni, amely képernyő az érintés után nem megfelelően fertőtleníti további fertőzésekhez vezethet. A vezérléssel történő programvezérlés a COVID-19 okozta járvány helyzetben még inkább hangsúlyosabbá és fontosabbá vált.

Szakdolgozatomban arra keresem a választ, hogy gesztusvezérlés megoldással az egészségügyben használt eszközök vezérelhetőek-e. Ha vezérelhetőek, akkor milyen megoldás a legmegfelelőbb.

2 PROBLÉMAFELVETÉS

Az egészségügyi rendszerben a jelenleg megvalósított és kivitelezett technológiákkal az eszközök képernyőjét a programvezérléséhez, megnyitásához meg kell érinteni, amely képernyő az érintés után nem megfelelően fertőtleníti további fertőzésekhez vezethet. A kézzel történő eszközérintés helyettesíthetőségének fontossága a programvezérlésben a COVID-19 okozta járvány helyzetben még inkább hangsúlyosabbá és fontosabbá vált.

Az eszköz érintésmentes vezérlése továbbá olyan környezetben is nagyon jól használható, ahol védőöltözetben szükséges az adott tevékenység elvégzése. Védőöltözetben, kesztyűben kevésbé érzékelhető a képernyőn az érintés. Védőfelszerelés használata az egészségügyben, iparban, vagy akár kutatások során és fertőzésveszélyes környezetben is kötelező.

Az eszközök érintésmentes vezérlése többféle módon valósítható meg. Megoldható hanggal vagy mozdulattal. A hanggal történő vezérlés tévedésre hajlamos lehet, az előre beprogramozott hang fel nem ismerése a program működés hiányát eredményezheti. A hangfelismerést betegség okozta rekedtség, avagy már egy fogorvos által történő kezelés, érzéstelenítés, de még egy gyorsabban vagy más hangsúllyal kiejtett szó is ellenőrizhetetlenné, felismerhetetlenné teszi.

A mozdulattal, gesztussal történő vezérlés megbízhatóbb megoldás. Nem lehet "megtéveszteni" a kívánt mozdulatot, a gesztushoz rendelt parancsot. A működésbeli megbízhatóság mellett az érintésnélküli gesztusvezérléssel idő takarítható meg, az érintés mozdulatának megtörténtéig, a védőöltözet levételéig, a kéz megtisztításáig.

A gesztusvezérlés egy non-verbális kommunikációs mód, melynél kézzel, arccal vagy a test bármely részével történik a kommunikációs tevékenység. Az érintésmentes gesztusvezérléssel ember és gép tud kommunikálni mechanikai eszköz vagy fizikai kapcsolat nélkül.

A technológia alkalmazásához és lehetőségeinek minél szélesebb körű felhasználásához kreativitás és innovativitás szükséges. A technológia működéséhez szükséges a felhasználó, egy kamera, egy kijelző panel és egy szoftver, amely a kamerából érkező képeket képfelismerés segítségével gesztusvezérléssé alakítja.

2.1 A gesztusvezérlés technológia

Az informatikai technológia fejlődésével a számítógép beviteli eszközök is fejlődtek. Az egér, billentyűzet, mint beviteli eszköz kiegészült az arc, a mozdulat, a gesztus felismerésének lehetőségével. Elérhetővé vált a gesztusvezérlés, mint az új adatbeviteli

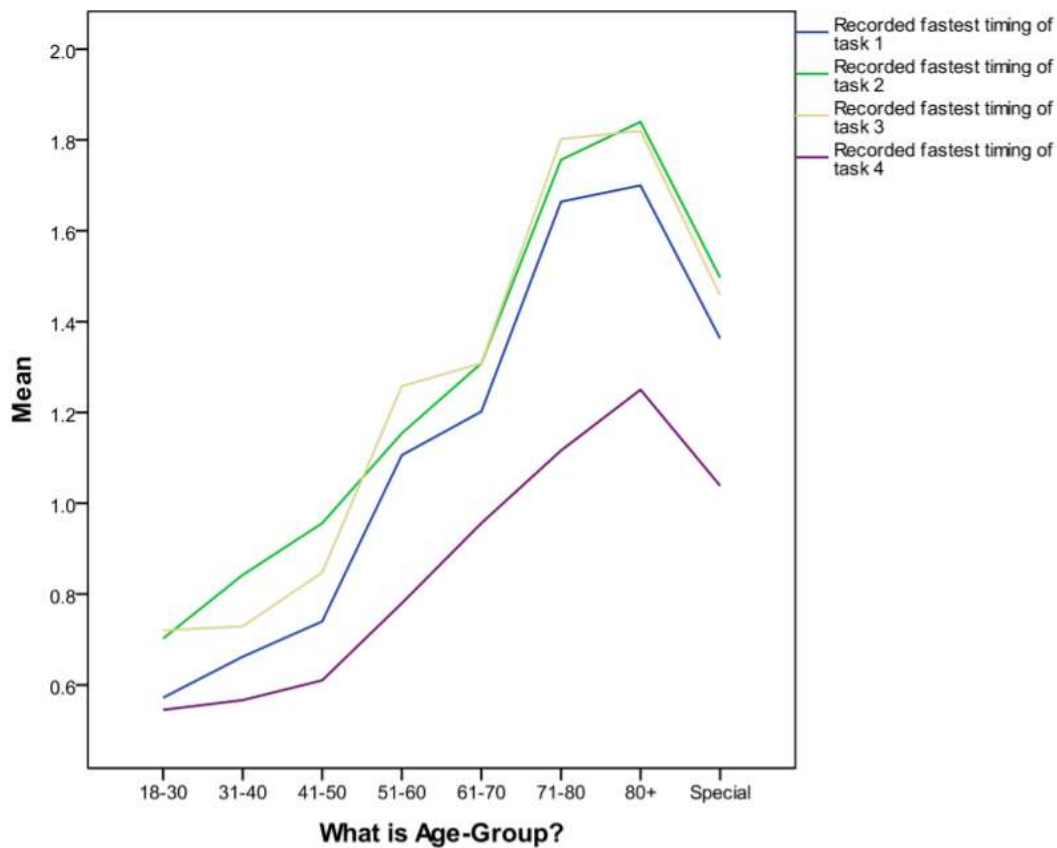
technológia. A felhasználóbarát, könnyen kezelhető megoldással interaktív adatbevitel valósult meg a felhasználó és a számítógép közötti kommunikációban.

A gesztusfelismerés érzékenyre hangolt technológiájának fejlesztéséhez évtizedek voltak szükségesek. A hangfelismerés technológiában az 1980-as években a rejtett Markov-módszer, a valószínűségi modell felhasználásának segítségével indította el az áttörést. A rejtett Markov-módszer (Hidden Markov Model) lényege, hogy amikor a rendszer regisztrálja a beszéd legkisebb elemét, a fonémiát, akkor az bizonyos valószínűséggel a következő lesz. A rejtett Markov-módszer ezen valószínűségeket használja a következő fonémia meghatározásához, alkotva ezzel a legvalószínűbb szavakat [1]. A rejtett Markov-módszert használják a mesterséges intelligenciában, a gépi tanulásban és a függvénytanulási problémában is. A gépi tanulásban a mély tanulás olyan neurális hálózatok segítségével történik, amelyben egynél több rejtett réteg van [2].

A hangfelismerés technológiájának népszerűsége erőteljesen a 2010-es évek elején nőtt meg, amikor a Siri-nek elnevezett digitális asszisztenst IOS alkalmazásként bevezették. A Siri Apple digitális asszisztense mellett megjelent a Google Segéd, az Amazon Alexa és a Samsung fejlesztése a Bixby digitális segéd is.

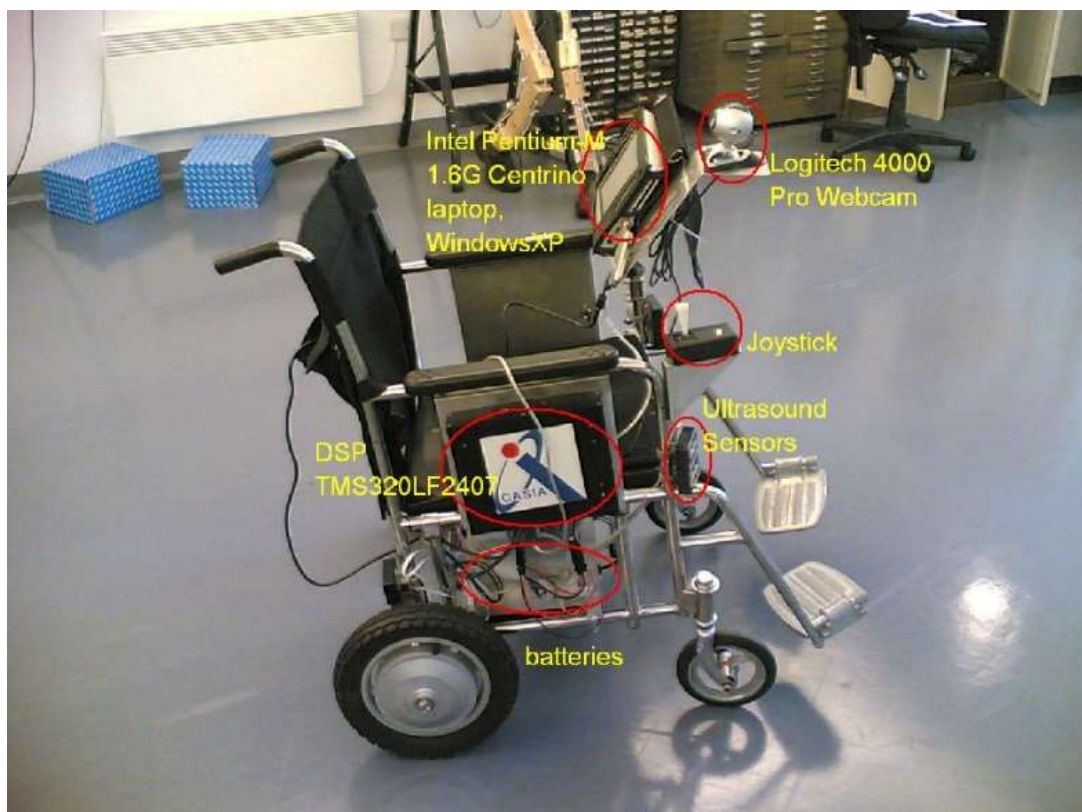
A hangfelismerés mellett megjelent a digitális kesztyű és nagy képernyőn történő egy kézzel történő gesztusvezérlés használata [3]. Az hang- és gesztusvezérlés technológiát az IBM 1990-ben elsősorban még az idősek és a sérültek számára fejlesztette, akik hordozható eszközként, nyakba akasztva hordták a bevitelt segítő eszközöket. Nagy áttörést jelentett a vizuális érintőpad megjelenése 2004-ben, ahol már mindkettő kéz gesztusvezérlésével oldották meg az adatbevitelt a számítógépbe. Egy sík felület fölé kettő kamerát rögzítettek, amely szoftveres megoldással figyelte az ujjak térbeli elhelyezkedését és mozgását, a 3D pozícióit. Ez volt az első igazi nagy lépés a gesztusvezérlés felé [4].

A 3 tengelyű gyorsulásmérő beépítésének használatával a gravitáció irányának mérése megoldottá vált. Nagyon sok tanulmány született a gesztus vezérelt beviteli eszközökről, azok használhatóságáról. Tesztelték a mozdulatokat életkor szerint, a mozgásukban korlátozottaknál, férfiak és nők csoportjára bontva. A tesztelés eredményeként megállapították, hogy a gesztus vezérelt beviteli módszert bármely korosztály vagy korlátozott képességű felhasználók is alkalmazni tudják. A különbség a beviteli eszköz működésének megértésének ideje-különbségéből fakadt [5]. (1. ábra)



1. ábra: Gesztusérzékelésen alapuló beviteli eszköz tesztelése

Elsőként 2007-ben a mozgásában korlátozottak számára vezették be gesztusvezérléssel vezérelhető kerekesszékes rendszert, amelyet egy hordozható számítógép és kamera segítségével használtak helyváltoztatásra. A fejmozdulatokat arcfelismerő és objektumkövető algoritmus segítségével érzékelte a program és vezérelte az alapján a kerekesszéket [6].



1. kép: Fejmozdulatot érzékelő gesztussal vezérelt kerekesszék

A gesztusvezérlést és gesztusok elemzését tovább kutatva a Microsoft kutatólaborjában 20 fő részvételével 1080 gesztust elemeztek. A 27 parancsot egy és kétkézes gesztusvezérléssel kellett végrehajtani. A résztvevők mindegyike jobb kezes volt és nem használtak számítógépet a munkavégzésükhöz és a mindennapjaikban. Az elemzés eredményeként megállapították, hogy a felhasználók az egy kézzel végrehajtott gesztusvezérlést a felhasználók jobban kedvelték. A felhasználóbarát megoldás sokoldalúan, könnyebben használható. Hogy a gesztusoknál mindenki ugyanarra gondoljon és ugyanazt használja, kialakították a felhasználói gesztusvezérlésen alapuló taxonómiát [7].

A gesztusvezérlésnek, mint minden technológia újdonságnak, fejlesztésnek, van egy életciklusa. A technológiai életciklus jobban szemléltethető, ha a technológiának az érettsége alapján van csoportosítva.

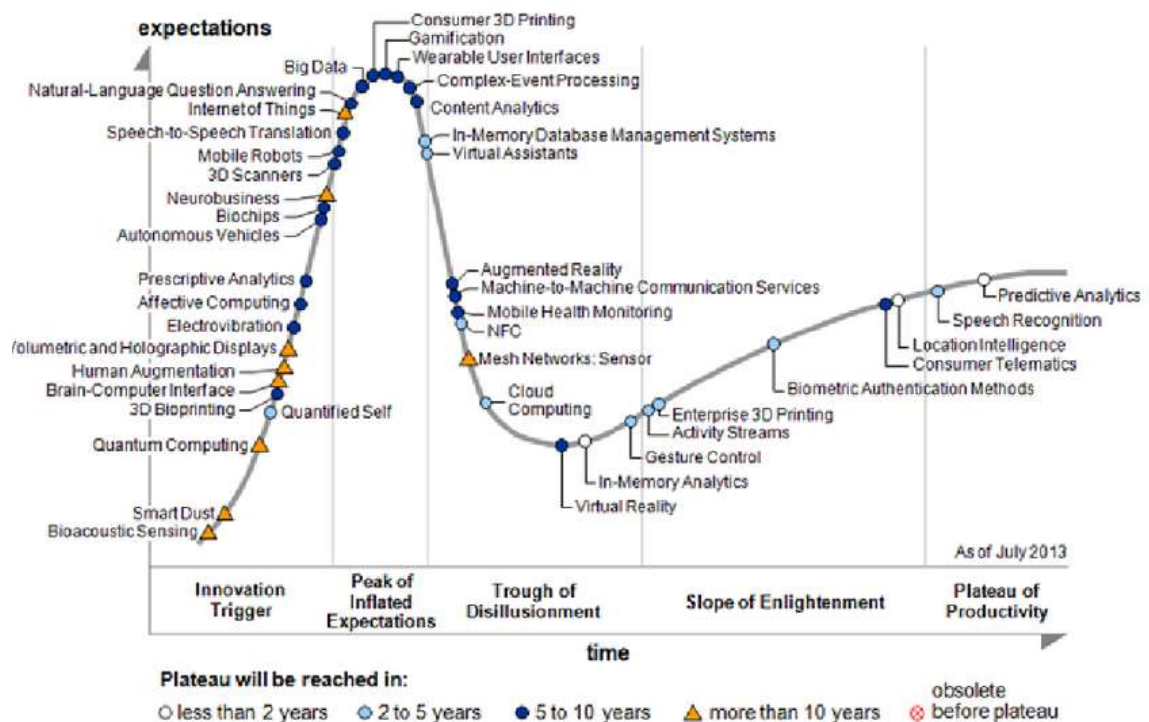
A Gartner-görbe a technológiai érettség mellett az ember a technológiához való viszonyát is mutatja. Figyelemre méltó, hogy a technológiai jövőjét kevésbé lehet megjósolni,

kevésbé van előtérbe helyezve, pedig a technológia felhasználásának és felhasználó nézőpontjából fontos szempont, hogy mennyire látható a technológia jövője.

A Gartner- görbe szakaszai: innováció „elsütése” (Innovation Trigger), a felfokozott várakozás csúcsa (Peak of inflated Expectations), a kiábrándulás völgye (Trough of Disillusionment), a felvilágosodás emelkedője (Slope of Enlightenment), a termelékenység fennsíkja (Plateau of productivity). Ahhoz, hogy a gesztusvezérlés technológiájának érettségéről dönteni lehessen, figyelni kell, megjelent-e a technológia a görbén és mikorra várható, hogy a termelékenységi fennsíkra érkezzon.

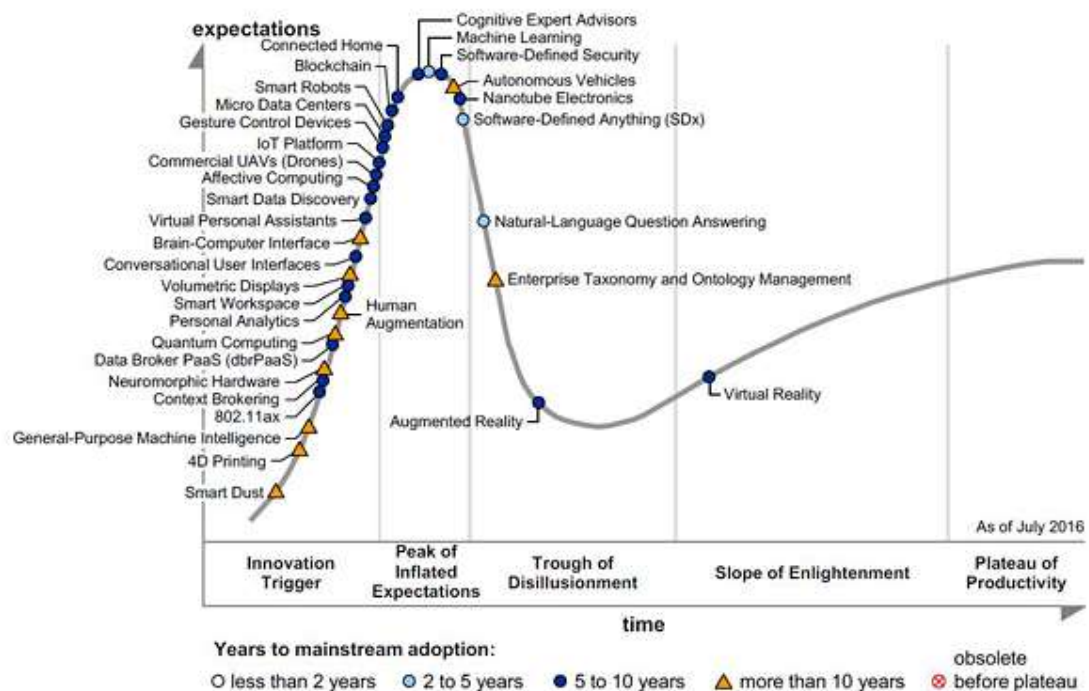
Az innovatív technológia indításával kezd a görbe, ami esetünkben a gesztusvezérlés, mint beviteli eszköz innovációja. A következő lépés az elvárési csúcs, ahol a média hatására is mindenre az új technológiával szeretnének megoldást találni. Megjelennek a piacon az első termékek, mint például a fejmozdulattal vezérelhető kerekesszék. Majd a „Kiábrándulás völgye”-ként definiált termék technológiai lejtőnél a problémák megjelenése kerül előtérbe, de közben tovább folytatódnak a fejlesztések a befektetők megjelenésével. Majd a megvilágosodási emelkedés során a technológia ismét ismertté válik a fejlesztések hatására, amely technológia immár húzóerőként marad jelen a mobiltechnológiában, a szórakoztatóiparban, az egészségügyben és a gépjárművek gyártásában [8].

A gesztusvezérlés a 2013. évi technológiai Gartner-görbéjén (2. kép) megfigyelhető, hogy a technológiai előrejelzés alapján a termelékenység fennsíkja 2-5 éven belül kerül, várhatóan 2015-2018 között.



2. kép: A gesztusvezérlés technológiai életciklusa 2013 (Kép forrása: Gartner)

Míg a gesztusvezérlést segítő eszközök a 2016-os technológiai görbén a felfokozott várakozás csúcsán van.



Source: Gartner (July 2016)

3. kép: A gesztusvezérlést segítő eszközök technológiai életciklusa 2016 (Kép forrása: Gartner)

A kézmozdulattal, a gesztusokkal történő vezérlés technológia láthatóan az életciklusban maradt három év elteltével is. A fejlesztések iránya eltolódott az érzékelők fejlesztése felé, hogy könnyebbek, gyorsabbak és hordhatóbbak legyenek.

A gesztusvezérlés technológiája jelenleg többféle iparágban van jelen.

3 AZ IPARÁGI SZEREPLŐK BEMUTATÁSA

A gesztusvezérlés technológiának a munkaerőre gyakorolt hatása képessége kiegészítő, azaz bővíti a dolgozók képességeit, lehetővé téve számukra új funkciók végrehajtását, növelve a termelékenységét.

A gesztusvezérléssel, az érintésmentes vezérléssel történő programvezérlés, a jövő egyik legjobban fejlődő és legjobban eladható terméke tud lenni, amelyet jelen pandémiás helyzet is igazolt. Az érintés nélküli fizetés, a kesztyűben történő közösségi érintkezés, az ajtók egy ujjal történő nyitása, a kézzel minél kisebb felületen érintkező kilincs, az emberek egymás közötti érintkezésére, a fizikai érintkezések elhagyását eredményezte a járványhelyzet.

A járványügyi helyzetben is egyre fontosabbá és felhasználóbaráttá tudott válni az érintésmentes technológia az előnyeivel az esetleges hátrányok mellett is. A következőkben a legfontosabb iparágakban kerül bemutatásra.

3.1 Gépjárművekben alkalmazott megoldások

A gépjárművekben alkalmazott érintésmentes programvezérlés forradalmi újítást hozott nemcsak a belső terek kialakításában, hanem a biztonság növekedésében is, a vezetéstámogató eszközök szempontjából is. A balesetek nagy része a gépjármű vezetőjének figyelmének elvonásából adódik a GPS, a rádió, a mobiltelefonok, a kijelző kezelése által. Az elmúlt években jelentős fejlesztések történtek a szórakoztató eszközök vezérlésével kapcsolatosan is, kormányra elhelyezett gombok, nagy érintő kijelzők, hangvezérléssel működő funkciók. A gesztusvezérléssel történő eszközvezérlés ezen eszközök „egybegyűrése”, jól beállítva és megfelelően használva nagyon hatékony tud lenni. Javíthat a figyelem elvonásán, kevésbé lesz figyelem elterelő, nem kell a gépjármű vezetőjének odanéznie a hívás fogadásához, elég, ha legyint egyet a kezével. Az eddig a csak csúcsmodellekben megtalálható technológiát a következő években a gépjárművekben alapszolgáltatássá teszik.

A 2015-ben[9] a BMW-ben elsőként alkalmazott gesztussal vezérelt technológiai megoldást. A gépjármű rendszer kijelzői feletti részen egy infravörös kamera van elhelyezve, amely figyeli a sofőr mozdulatait a kijelzők előtti adott területen.

A gépjármű, vezetőjének érzékelve a kézmozdulatát, a kijelzőt gesztusokkal vezérelheti. Hangerő változtatás, hívások kezelése, rádióvezérlés, csak néhány példa felsorolva a kijelző kézmozdulattal, gesztussal történő vezetéstámogató eszközként való funkcióról.



4. kép: BMW gépjárműtípusok gesztusvezérlés eszközrendszere (Kép forrása: BimmerTech)

A technológia előnye a kézmozdulattal történő információbevitel, amely egyúttal a hátránya is tud lenni. Motoron nem biztonságos a technológia használata.

3.2 Mobiltelefonok gesztusvezérlése

A Samsung típusú mobiltelefonok nagy újdíása a gesztusvezérléssel történő érzékelők használata, amely nagy sikert aratott a felhasználók körében. A gesztusvezérlés segítségével nem kell megérinteni a képernyőt a hívás felvételéhez vagy fénykép készítéséhez. Elég, ha a tenyeret a képernyő elé helyezve a kamera felismeri és készíti a képet. A fénykép készítése arcfelismeréssel mosollyal vagy hangvezérlésre is készülhet.

A Huawei Mate 30 típusú telefonoknál a fénykép készítésén kívül érintésmentes gesztusvezérléssel vezérelhető a telefon galéria menüpontja és a böngésző alkalmazás használata is.



5. kép: Gesztusvezérlés használata mobiltelefonon ¹

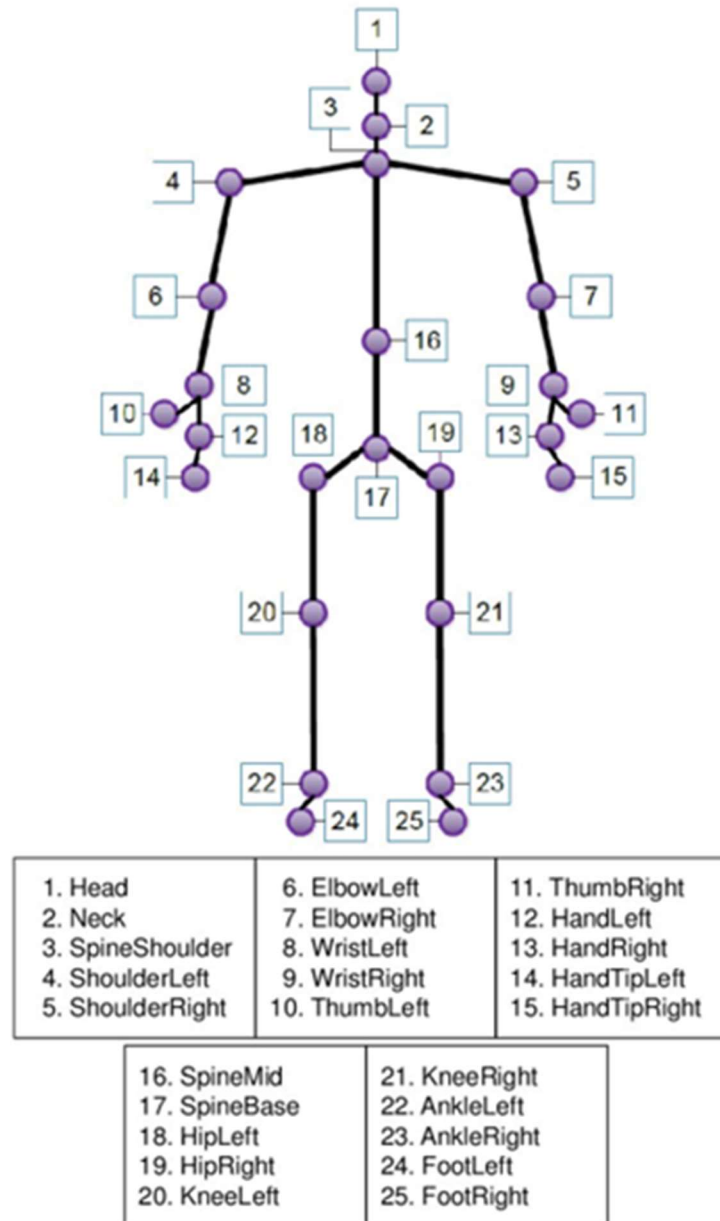
3.3 Szórakoztató eszközök

Elsőként 2006-ban a Nintendo Wii játékánál volt kézbevehető a szó szerinti kézmozdulatokkal történő játékvezérlés. A technológia gyorsulásmérő, optikai érzékelő segítségével érzékelte a gesztusokat.

Microsoft Kinect

A Microsoft fejlesztése a Kinect mozgásérzékelő, mely felismeri a testet, az arcot, a hangot, a gesztusokat és követni is képes azokat. A Kinect egy videokamerával és kettő infravörös mélységérzékelő kamerával ellátott eszköz, amely 3D térben képes érzékelni az emberi testet. Az érzékelés 25 ponton történik 30 másodperces mintavételekkel [10].

¹ Kép forrása: <https://consumer.huawei.com/en/phones/mate30-5g/experience/> (2021.12.17)



6. kép: Microsoft Kinect érzékelőknek megfigyelési pontjai [10]

A Kinect a működése során infravörös fénypontokat rácsoz ki a kamera által látott képre, majd megméri, hogy a pontokból a fény mennyi idő alatt ér vissza, ebből érzékelhető a távolság. A Kinect a távolságmérését a fény terjedési sebességére alapozta, a Time of Flight (ToF) eljárásra.

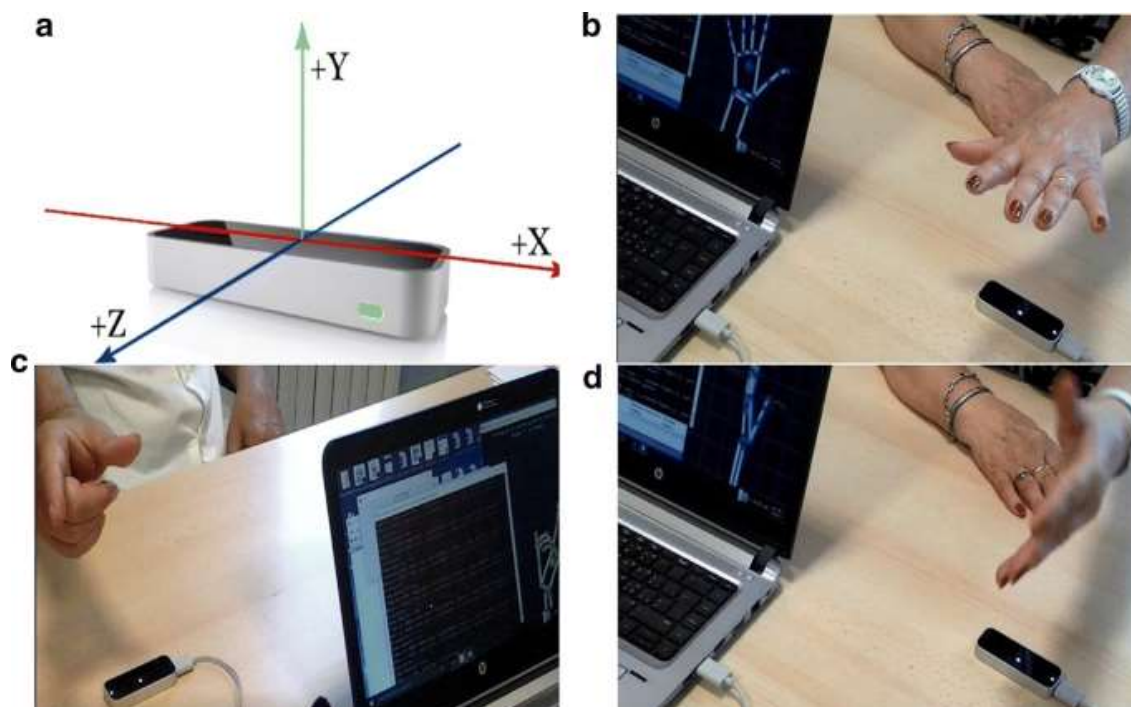
Az eszköz gyártását a Microsoft 2015 óta beszüntette, de a termék nagyban hozzájárult a következő Microsoft fejlesztésekhez, amely a kiterjesztett és a virtuális valóságon alapultak.

A Microsoft Kinect eszköz sokat lendített előre az egészségügyi fejlesztések gesztusvezérlésen alapuló felhasználásának körében.

Leap Motion

A Leap Motion egy számítógépes hardver érzékelő eszköz, az ujjak és a kéz mozgását érzékeli a térben. Az eszköz két monokromatikus infravörös kamerával és három infravörös LED-el rendelkezik, amelyek algoritmus segítségével számítják ki az ujjak és a kezek helyzetét. Az eszköz szoftvere 27 rész különbözőségét [11] képes érzékelni, beleértve az ízületeket és a csontokat. Ezen részeket akkor is érzékeli, ha a kéz részei eltakarják egymást. Az eszköz számítógéphez vagy közvetlenül az AR/VR fejhallgatókhoz is csatlakoztatható. Az eszköz infravörös mélységérzékelővel rendelkezik, a kéz és az ujjak legkisebb rezgését is érzékeli.

Az eszközt tesztelték a Parkinson kórban szenvedő betegek motoros diszfunkciójának felmérésére. A Parkinson kórt diagnosztikája a motoros funkciókon alapulnak. A funkciókat a neurológusoknak vizuálisan és kérdéssorozatokra adott válaszokkal vizsgálják, elemzik és sorolják be a betegség adott szakaszát a betegség vizsgálati skáláira. A Leap Motion eszköz nagy segítséget nyújt az állapot felmérésében, meglepően jó eredményt mutatott az eszköz az ujjkoppintás és a kéz nyitás és zárás funkcióinál a betegség felismerésében.



7. kép: Leap Motion eszköz a Parkinson betegség diagnosztizálása közben

Az eszköz az egészségügyben jól használható a Parkinson kór betegség fokának érzékelésében és értékelésében.

A Leap Motion eszközt a Kinect kiegészítheti a Parkinson kór tüneteinek a test egészén való remegésnek, az instabilitásának a felismerésében. A Kinect a láb és a test vizsgálatában nyújt az infravörös mélységérzékelők általi elemzést, a Leap Motion pedig a kéz és az ujjak motorikus funkcióinak megállapításában nyújt kiegészítő segítséget az orvoslásban [12].

3.4 Smart eszközök

A smart eszközök egyik tulajdonsága, hogy mesterséges intelligenciát használnak működés közben. Az okostévé felismeri a kéz mozgását a csatornaváltások vagy a menü kiválasztásánál. Arcfelismerésen alapuló, mosollyal vagy nyitott tenyér kézmozdulattal történő gesztusvezérlést alkalmaznak egyes okostelefonok a fényképek automata készítéséhez, a telefon feloldásához. Jelkommunikációt felismerő szoftver segítségével lehet kommunikálni sérült emberekkel, jelbeszéd előzetes ismerete nélkül. Okos otthonokban a fűtést lehet vezérelni, a lámpát, a klímát. Egyre inkább előtérbe kerülnek a gesztus vezérléssel működtetett eszközök, segítve az emberek mindennapi életvitelét.

3.5 Egészségügy

A szakdolgozat elején, a problémafelvetésben a gesztusvezérlés egészségügyben történő használatának, megoldásának a lehetősége a kérdés.

A gesztusvezérlés technológia egy évtizedes múltra tekint vissza, a 2. képen levő 2013. évi Gartner-görbén is jól látható a technológia termelékenységi fennsíkon levő elhelyezkedése.

Az egészségügyben a képalkotó eljárások használata már mindennapossá vált, míg a képalkotó technikának a műtét során vagy sterilizált környezetben az interaktív felhasználása nehezen tud megvalósulni az érintés szennyeződése nélkül. A képalkotó eljárás segít az orvosnak a diagnózis és a műtéti eljárás meghatározásában. A képalkotó eljárás során úgynevezett „látóvonal” segítségével a páciens 3 dimenziós képen szemlélve lehetőség van az egér és a billentyűzet segítségével a műtétet, a műtét során használt eszközt a testben pontosan nyomon követni, amely nyomkövetést műtét közben esélytelen asszisztens nélkül végezni.

A műtő berendezései, a műtéthez szükséges eszközök, számítógépek, képmegjelenítők többnyire a falak mellett helyezkednek el, amelyeket nem steril környezetben kezelnek az asszisztensek, nővérek és az aneszteziológusok. Az eszközök erősen szennyezettek, így az orvosok műtét közben nem érhetnek hozzájuk, miközben a képalkotó berendezések információi alapján műtenek. Ha szükséges egy információ az orvosnak vagy másik területet szeretne látni a képernyőn, jelez az asszisztensnek, aki igyekszik az orvos utasításait gyorsan és pontosan végrehajtani.



8. kép: Nem steril környezetben információ keresés és bevitel [12]

A műtéti asszisztenssel történő információ nehezé, lassítja a műtéti eljárást. A kért eredmények közlése is hibalehetőségre ad okot. A műtétet végző orvosnak gyakran a kesztyűjét kell levenni vagy a köpenyével megpöckölni az egér beviteli eszközt, hogy a kívánt kép látható legyen. Ezzel azonban megsérti a steril környezetet, amely a betegnél vérmérgezéshez vezethet.

A kézmozdulat felismerésével történő információbevitel segíti az orvost műtét közben. Gyorsabbá és pontosabbá teszi a műtéti eljárásokat.



9. kép: Gesztusvezérlés az érsebészetben [12]

A gesztusvezérlés támogatásával a műtéti eljárás közben a képet nagyítani, kicsinyíteni, forgatni, dönteni. Az egészségügyben a gesztusvezérlés bevezetését nagyban segítette a Kinect mozgáskövető szenzor megjelenése.

A Kinect rendszer a test csontváz szerkezetét követi vagy a felsőtestet figyeli.

A műtéti eljárásoknál a műtőasztal magassága, a műtét közben a páciens körül az orvosok, nővérek létszáma és elhelyezkedése behatárolja a gesztusvezérlés lehetőségét, a mozdulatok felismerhetőségét vagy félreérthetőségét.

A gesztusvezérléssel és az azonnali képalkotó eljárás segítségével a műtétet végző orvos saját maga által kezelt virtuális módon a páciens testében tud „látni és mozogni”, ellentétben az asszisztensnek adott instrukciókkal szemben, hogy a beviteli eszközt melyik irányba, hogyan mozgassa, hogy megfelelően látni lehessen az operálni kívánt részt. Az orvosnak a műtőasztaltól akár el is kell távolodnia, ha saját maga szeretné beállítani a látni kívánt részt [13].

Összehasonlítást végeztek a műtéti eljárások során gesztussal vezérelt környezetben, asszisztens hangvezérlés irányításával és az orvos saját maga általi billentyűzet, egér beállításával. A tesztkörnyezetben az orvos által kezelt billentyűzet és egér használata volt a leggyorsabb, majd a gesztussal vezérelt eszközök volt a következő leggyorsabb, a szóbeli utasítások alapján volt a lelassabb az elvárt eredmény. Az elégedettség mérés ugyanezt a sorrendet adta. Az eredmény által kimutatható volt, hogy a gesztussal vezérelt környezetben gyorsabban, pontosabban és hatékonyabban lehet dolgozni, mint a szóbeli utasítások alapján [14].

A szakdolgozat problémafelvetésére olyan mobil alkalmazás fejlesztésével látom a megoldást, amelynek menüpontjai gesztussal vezérelhetőek.

4 A MEGVALÓSÍTÁSHOZ SZÜKSÉGES KÖRNYEZET

4.1 A fejlesztőkörnyezet

Miután eldöntöttem, hogy fejlesztésemben gesztussal vezérelhető mobilapplikációt szeretnék megvalósítani, kiválasztottam a szükséges fejlesztőkörnyezetet. A gesztussal vezérelhető applikáció fejlesztéséhez a Unity Technologies által fejlesztett Unity szoftvert találtam a legalkalmasabbnak. Sokoldalú szoftver, mely erős játékmotorral rendelkezik.

Vizuális szerkesztőprogramjával egyaránt lehet fejleszteni vele számítógépre, Android és iOS operációs rendszer használó mobilra egyaránt. Komponensei és játék objektumai (GameObject) nagyon jól testre szabhatóak. Kiegészítők, „assetek” sokaságából, - animációból, felületből, modellekből - lehet választani, mely az Asset Store-ból való letöltés után azonnal használható. A környezetet C#, JavaScript és Boo nyelven lehet programozni. A Unity környezetet a Unity Hub-ról lehet letölteni ingyenesen, 5.6 és a 2017 verziótól.

A mobileszközön a gesztusvezérlést, a kézmozdulat felismerését, a mobil első kamerájával és hátsó kamerája használata közben lehet érzékelni. Sajnos a mobiltelefonok kamerája nem rendelkezik mélységérzékelőkkel, ami a kezet, a kézmozdulatot 3 dimenzióban érzékelné, ezért a kiterjesztett valóságot hívtam segítségül.

4.2 Kiterjesztett valóság – Augmented Reality

A kiterjesztett valóság meghatározása Ronald Azuma a világhírű valóság és virtuális kutató szerint „az AR olyan rendszereket jelent, amelyek az alábbi három fő jellegzetességgel rendelkeznek:

- valós és virtuális tárgyak keverednek valós környezetben
- valós időben interaktív
- 3D kiterjedésű” [15].

A gyakorlatban ezt azt jelenti, hogy a valós időben, térben látható elemek, dolgok mellé lehet interaktív, digitális elemeket vetíteni. A kiterjesztett valóság információi szabad szemmel nem láthatók, de mobiltelefon kameráját felhasználva, a telefonon levő alkalmazással, bárhol, bárkinek elérhetőek. A kiterjesztett valósággal készült alkalmazás használata új tudás megszerzését nem igényli. A kiterjesztett valóság elterjedését segíti,

hogy az okoseszközök egyre inkább elérhetőbbé válnak, egyre többen használják és mind jobban igénylik az interaktív tartalmakat.

A kiterjesztett valóság segítségével valós időben informatikai segédeszköz használatával digitális információk hozzáadásával lehet kiegészíteni egy meglevő információt. A kiterjesztett valóságban a valós világra lehet kivetíteni digitális információkat. A kiterjesztett valóság elméleti része a virtuális valósághoz képest kisebb mozgásterű rendszert ír le, mivel a kiterjesztett valóságban a virtuális tér megfigyelőpontjai valós fizikai távolságra vannak. A megjelenített digitális tartalom lehet állandó vagy időben változó.

A kiterjesztett valóság megjelenése óta a fejlesztése folyamatos. A kiterjesztett valóság a 2013. évi technológiai Gartner-görbén (2. kép) a „Kiábrándulás völgye”-ként definiált technológiai lejtőn van, ahol az új technológiával járó problémák megjelenése kerül előtérbe, de közben folytatódnak a fejlesztések, keresik és kutatják a lehetőséget a technológia előnyös és megfelelő használatára. A 2016. évi Gartner-görbén (3. kép) a technológia a felvilágosodás emelkedőjén van már, amikor ugyanebben évben a gesztussal vezérelhető eszközök a fokozott várakozás csúcsán járnak. Az új technológiák követik egymást, melyek lehetőségei szinte korlátlanok.

A kiterjesztett valóság népszerű használatát alátámasztja, hogy mindkét nagy mobil operációs rendszer gyártó rendelkezik (iOS és Android) kiterjesztett valóság fejlesztőkörnyezettel [16].

Tartalomfejlesztési és felhasználási lehetősége rendkívül sokrétű. Nagyon jól használható marketing- és reklám célra, szórakoztató eszközként, de megjelent az oktatásban, az egészségügyben, a turizmusban, az értékesítésben, a gyártásban és a karbantartásban is.

Az egészségügyben erőteljes törekvés látható a technológia lehetőségeinek felhasználására, a 3D adatok MRI és CT felvételekből a műtéti eljárásokat segítve. 2018-ban már elkészítették és tesztelték azon projektet, amely a kiterjesztett valóság ProjectDR elnevezésű szoftver segítségével az MRI és CT felvételek alapján azonnali, valós idejű képet illeszt (10. kép) a páciens testére [17].



10. kép: Kiterjesztett valóság alkalmazása az egészségügyben [17]

A kiterjesztett valóság megjelenítéséhez szükséges mobiltelefon, tablet vagy egy olyan okos szemüveg, amelyen keresztül látható a valós világ mellett megjelenített virtuális tartalom is.

A kiterjesztett valósággal támogatott gesztusvezérlésnél minden egyes eszközre külön szükséges alkalmazást fejleszteni a kívánt menüpontokkal, a virtuális kezelőgombok láthatóságához mobiltelefon, tablet vagy okos szemüveg használata szükséges.

4.3 Okoseszközök a kiterjesztett valóság megjelenítésér

A kiterjesztett valóságot többfajta eszközzel lehet megjeleníteni. A megjelenítés végezhető kézen tartandó eszközzel, okostelefonnal vagy fejen hordható okos szemüvegekkel.

A Microsoft által a világon elsőként elkészített, számítógéphez nem csatlakoztatott holografikus miniszámítógép. A HoloLens okos szemüveg a használatában nagyon sok mindenben hasonlít az okostelefon használatához. A kiterjesztett valóság megjelenítése és érzékelése a szemüvegkeretbe épített kamerák és érzékelők segítségével történik.

Jelentős energiákat fektetett a Microsoft a Hololens fejlesztésébe mind katonai, oktatási és üzleti felhasználási célból, de nem váltotta be a hozzá fűzött reményeket a Hololens 2 verziójú szemüvege sem.



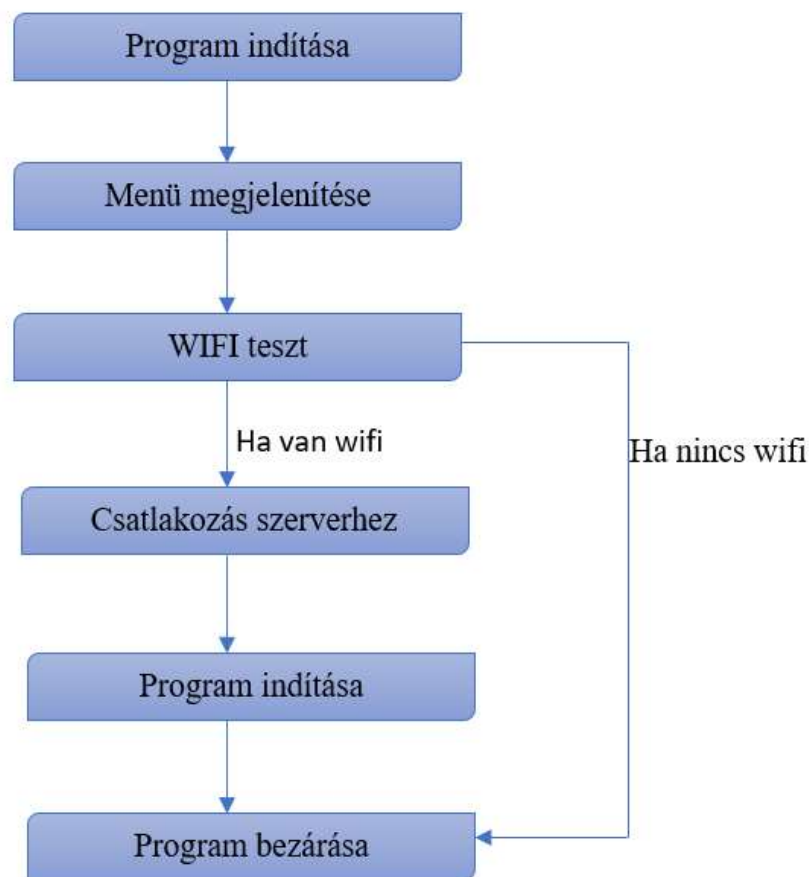
11. kép: 1. generációs Hololens okoszemüveg (kép forrása: www.microsoft.com)

Amíg a Hololens jól használható volt telefon nélkül is, addig a Google Smart Glass telefon nélkül szinte alig használható. Használatával nem érzékelhető a kiterjesztett valóság élménye, melyet a Hololens nyújt. A kézben tartható okostelefonok, legyenek azok IOS vagy Android operációs rendszerrel rendelkezők, képesek megjeleníteni a kiterjesztett valóságot, mely mindenki számára hozzáférhető.

A jövő a felhőben tárolható kiterjesztett valóság egyik irányába mutat, a Mark Zuckerberg által elnevezett „Multiverzum” új hívószóval és a Portal elnevezésű eszközzel, amely „okossá” tud tenni egy felületet. Ezen új eszköz a kiterjesztett valóság fejlesztések körében képes elérni azt a célközönséget, amelyet a Hololens és a Google Smart Glass okoszemüveg a költségei miatt nem ért el és amely már sokkal többet tud nyújtani, mint egy okostelefon képernyője.

5 AZ ALKALMAZÁS MEGVALÓSÍTÁSA

A szakdolgozatom célja olyan mobilalkalmazás készítése, mellyel egy helyi hálózatban levő, egy megadott számítógéphez csatlakozva, érintés nélkül lehet a számítógépen levő alkalmazást elindítani. A fejlesztést websocket alapú összeköttetéssel és a kiterjesztett valóság segítségével valósítottam meg, három virtuális, gesztussal vezérelhető menüpontot kialakítva az alkalmazás felületén. Az alkalmazást Unity fejlesztőkörnyezetben készítettem. (2. ábra)



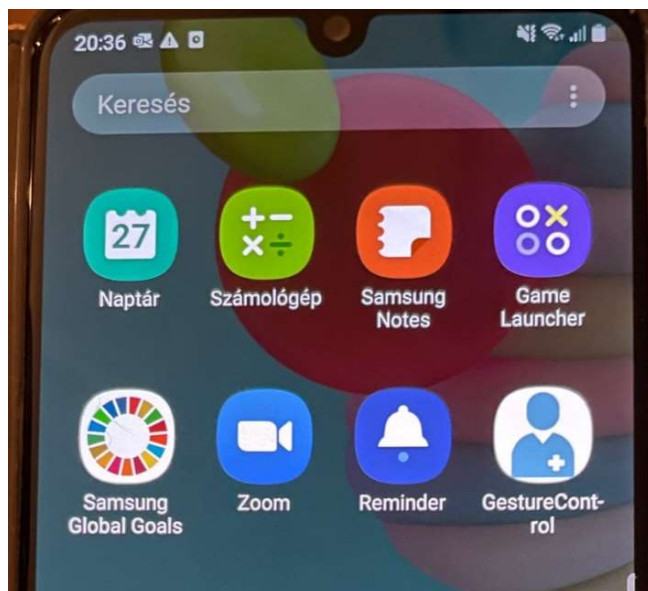
2. ábra: Rendszerterv

A Unity fejlesztőprogram nagyon erős játékmotort tartalmaz, két-, és háromdimenziós játékokat, animációkat, szimulációkat lehet vele fejleszteni. Platformfüggetlen fejlesztőkörnyezetet kínál, Microsoft Windows, Mac OS, Xbox, Playstation, Wii, iPhone, Android fejlesztői szoftverkörnyezet kiegészítőkkal. Az elsődleges nyelvként a C# programnyelvet használja, az alkalmazáshoz a scriptek is C#-ban készültek. A Unity egyik legfontosabb erőssége a támogatottsága, minden nap újabb segédprogramokat fejlesztenek hozzá, ezenkívül a fejlesztőkörnyezet könnyen átlátható és ingyenes.

A Unity fejlesztőkörnyezetben a 2020.3.33f1 LTS verzióval fejlesztettem a programot. A program készítésénél elsődleges szempont volt a használhatósága és a felhasználóbarát környezet.

Az alkalmazásban a gesztust, a kézmozdulatot a telefon vagy a tablet kameráján keresztül érzékeli, melyet az alkalmazás feldolgoz és az előre elkészített menürendszert aktivizálja. A megvalósítást a piacon jelenleg megtalálható, elérhető áron beszerezhető eszközökkel oldottam meg, így az alkalmazáshoz Android alapú telefon vagy tablet szükséges. A kéz mélység-távolság érzékeléséhez infravörös kamera szükséges, amellyel rendszerint nem rendelkeznek a telefonok, ezért az érzékelés 3 dimenziós problémáját a kiterjesztett valóság segítségével valósítottam meg.

Az alkalmazás Android operációs rendszerre történő fordítását a szoftverkörnyezet beállítása után végeztem. A telefon vagy tablet képernyőjén megjelenő alkalmazás indító ikonokat egyedire készítettem. (12. kép)



12. kép: Képernyőn megjelenő GestureControl alkalmazás ikonja

A fejlesztőkörnyezetben Vuforia pluginnal alakítottam ki a kiterjesztett valósághoz szükséges környezetet. A kiterjesztett valóság felületet indító QR kódjának telefonnal olvasható szövege: „Gesture control in environment with risk of infection”. Ezen QR kódot állítottam be cél képként a Vuforia SDK felületén és töltöttem be a Unity programba, amelyre az ikonokkal díszített virtuális gombokat helyeztem. A gombokat a kívánt tulajdonságoknak megfelelően C# scriptekkel kódoltam.

A kiterjesztett valóság alapú alkalmazás indításához és az alkalmazás használata közben nem szükséges a felhasználónak arrafele néznie, ahol a központi műveletet végző gép található, amelyen az alkalmazás által indított program fut. A virtuális felületet indító QR kód vagy marker elhelyezhető a beteg mellett is, az Android felületet bármelyik irányba lehet fordítani. Ha a QR kód a beteg mellett van elhelyezve közvetlenül, akkor egyszerre a beteg is megfigyelhető az alkalmazás használata közben.

A virtuális felületet indító QR kód (13. kép):

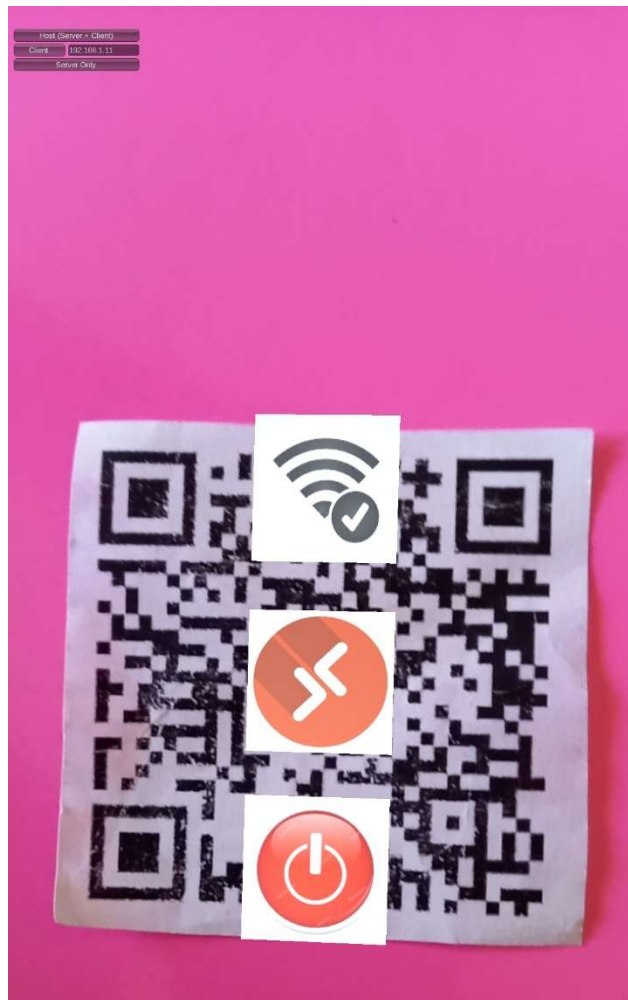


13. kép: A kiterjesztett valóság felületet indító QR kód

Az alkalmazás három virtuális menüpontot,

- wifi teszt gombot/ikont;
- szerveren való böngészőprogram alkalmazás indításának ikonját;
- kilépés ikont

és a NetworkManager HUD felület szerverhez való csatlakozás indítását tartalmazza (14. kép)



14. kép: GestureControl alkalmazás mobiltelefon képernyője QR kódról indítva

5.1 Az alkalmazás menüpontjai

Az alkalmazás menüpontjai ikonokkal indíthatóak, melyek funkcióit a virtuális gombokon elhelyezett képek mutatják. A gesztus vezérlést az alkalmazásban a QR kód és a virtuálisan megjelenő gombok közti területet kitakarását jelenti ujjal vagy kézzel.

Wifi tesztelés

A wifi gomb aktiválásával indul a script. Az eszköz helyi hálózatra való csatlakozását ellenőrzi egy scripttel, amelyet a „*StartCoroutine(CheckInternetConnection());*” paranccsal hívok meg. Ha van internet kapcsolat, a virtuális gomb gesztusvezérlése alatt „*CONN: OK*” felirat jelenik meg, ha nincs internet kapcsolat, a „*CONN: NOT OK*” felirat jelenik meg. A felirat a gomb területének elhagyása után eltűnik.

A virtuális ikonnál az ikon „lenyomását”, azaz az ujj vagy kéz érzékelésénél a „*public void OnButtonPressed_wifi(VirtualButtonBehaviour Vb_wifi)*” parancsot, az ujj vagy kéz elvonását az ikon alóli eseményt a „*public void OnButtonReleased_none(VirtualButtonBehaviour VB_wifi)*” paranccsal hívom meg.

Szerver-Kliens kapcsolódás

A szakdolgozati alkalmazás programozásban megvalósítani számomra a legnehezebb a Server-Kliens kapcsolat megfelelő kiépítése volt.

A Unity fejlesztőkörnyezetnek nagyon jól megtervezett parancskészletei vannak. Az szerver-kliens kapcsolódást a Multiplayer High Level API (HLAPI) segítségével fejlesztettem, amelyet a fejlesztőkörnyezet Package Manager menüpontján keresztül adtam hozzá az alkalmazáshoz.

A HLAPI lehetővé teszi, hogy az parancsokat használó alkalmazás felhasználók egyszerre legyenek kliens és szerver módban, így nem szükséges dedikált szerver használata.

A magas szintű API scriptek hozzáférést kapnak a parancsokhoz, amelyek lefedik a többfelhasználós fejlesztés követelményeit. A hálózati alkalmazásfejlesztéseket lehet építeni többfelhasználós módon vagy hálózati infrastruktúra építésével együtt többfelhasználós módon.

A HLAPI (3. ábra) az internet szolgáltatásokkal képes együttműködni.

Unity Multiplayer High Level API

Transport / Configuration	Low Level API
Connection / Reader / Writer	Messaging & Serialization
NetworkClient / NetworkServer	Connection Management
NetworkIdentity / NetworkBehaviour	Object state & Actions
NetworkScene / ClientScene	Object Life-Cycle
NetworkManager	Game Control
NetworkLobbyManager	Player Control
NetworkTransform NetworkAnimator NetworkProximityChecker	Engine Integration

3. ábra: Unity Multiplayer High Level API²

A HLAPI az alacsonyabb szintű Transport, kommunikációs rétegekre épül. *NetworkReader*: ez az osztály olvassa az objektumokat az adatfolyamból.

NetworkServer: High Level API menedzseli a csatlakozásokat a többfelhasználós üzemmódban.

NetworkClient: A szerverhez való hálózati csatlakozásokat, hálózati objektumokat kezeli. Az RPC üzenetküldés és fogadás és hálózati osztály kezelése is ezzel az osztállyal történik.

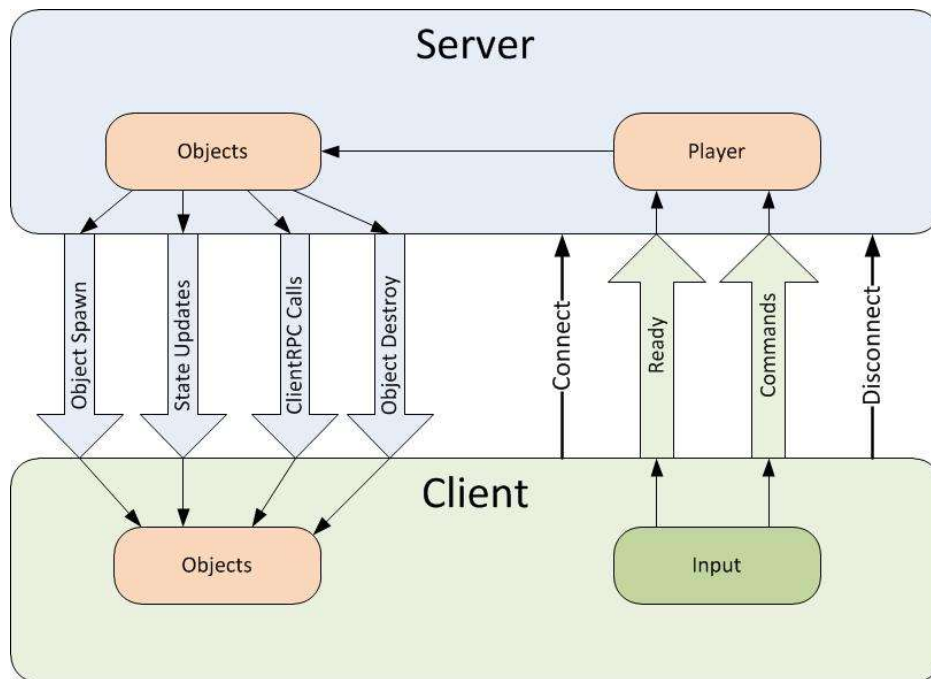
NetworkIdentity: A hálózati azonosításokat kezeli és felügyeli.

NetworkManager: Ez az osztály felügyeli az állapotot a hálózati alkalmazásnak.

NetworkLobbyManager: A felhasználók hálózatba léptetését ellenőrzi, mielőtt csatlakoztatja őket.

² Saját készítésű ábra

NetworkTransform: Jogosultságellenőrzést végez, szinkronizálja a pozíciót kliens-szerverre, majd a szerverről a kliensre.



15. kép: Kliens-Szerver kapcsolat Unity Network kapcsolattal³

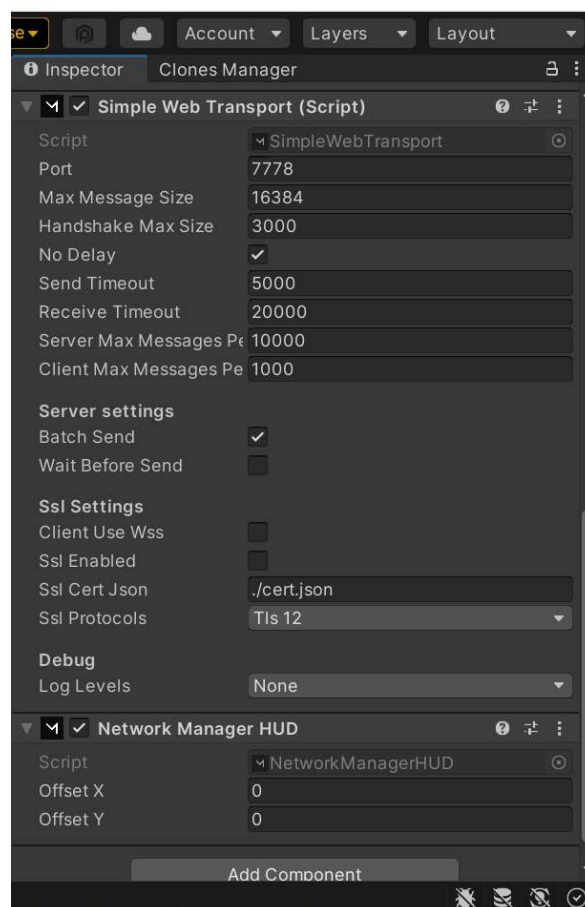
A HLAPI-t használja a Mirror plugin is, mellyel a szerver-kliens (15. kép) kapcsolat lett felépítve. A kapcsolat websocket típusú, amely WebGL réteget biztosít a szállítási réteghez.

A websocket kétirányú, full-duplex kommunikációs protokoll, egyszerre lehet vele információt küldeni és fogadni. A kétirányú kapcsolat TCP/IP protokollon keresztül valósul meg. Jól együtt tud működni a tűzfalakkal, HTTP kompatibilis, nagyon gyors információcserét biztosít. A kapcsolat felépítéséhez a kliens egy kézfogási (handshake request) kérelmet indít a szerver felé, amelyre a szerver kézfogási választ (handshake response) ad.

³ <https://docs.unity3d.com/2020.3/Documentation/Manual/class-NetworkBehaviour.html> (2022.05.01.)

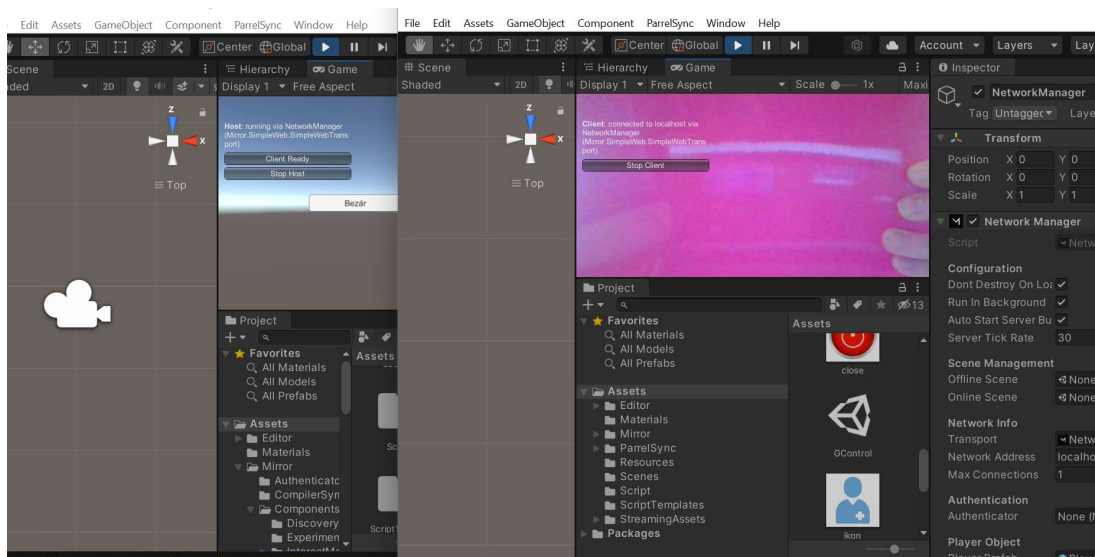
Hátránya a proxy szerverrel való nehezebb együttműködés és a skálázhatósága. Ha titkosítatlan a websocket forgalom, a websocket kapcsolatot a proxy szerver nem engedi felépíteni. A TLS-el (Transport Layer Security) felépített websocket kapcsolat HTTP CONNECT paranccsal kiépít egy csatornát, amely segíti a websocket kapcsolat kiépülését proxy szerver esetén. A protokoll URI (Uniform Resource Identifier) két sémát határoz meg, a nyílt kommunikációhoz a ws-t (WebSocket), a titkosított kommunikációhoz pedig a wss-t (WebSocket Secure).

Az alkalmazásban kiépített websocket kommunikáció Transport Layer Security (TLS) 1.2 biztonságú és a 7778-as számú porton kommunikál. (16. kép)



16. kép: Fejlesztőkörnyezet websocket beállítása

A fejlesztés alatt a program szerver-kliens közötti kapcsolat felépítését, működését és ellenőrzését a ParrelSync Unity klón kiegészítővel végeztem. (17. kép)



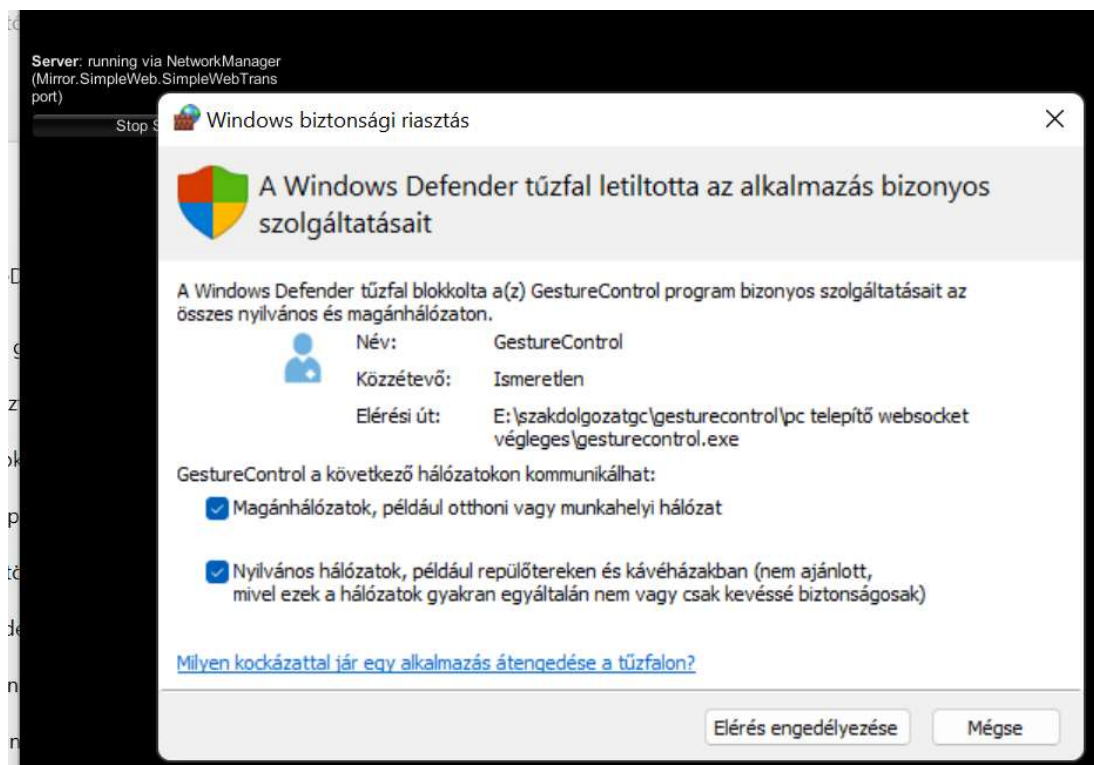
17. kép: Szerver-kliens közötti kapcsolódás ellenőrzése a fejlesztőkörnyezetben

Alkalmazás indítása

Az alkalmazások indításához a telepítőprogramokkal létrehozott szerver és kliens ikonra kell kattintani. A szerver oldali websocket indítását, a NetworkManager HUD panelján a számítógép IP címének beírása után a „Server” gombbal kell indítani.

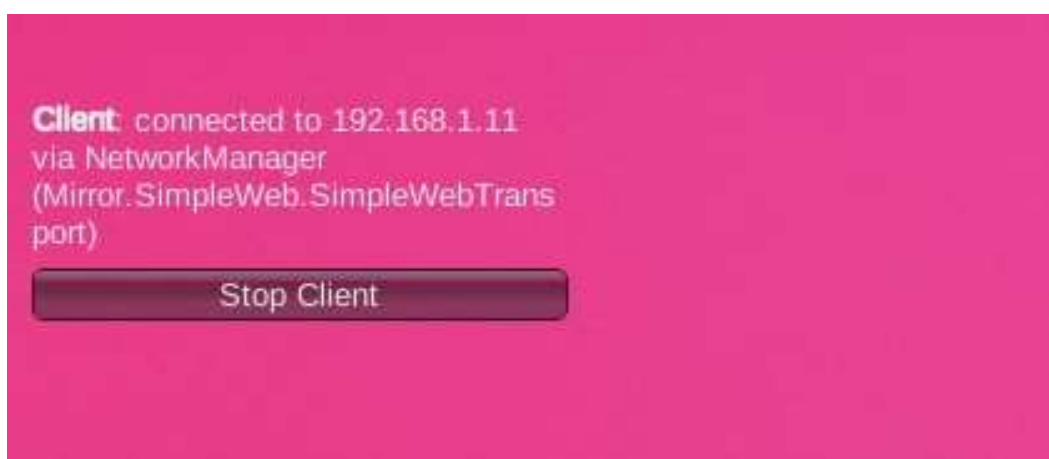
A mobiltelefonon a „Gesturecontrol” programot kell megnyitni. A mobiltelefon képernyőjén megjelenik a Networkmanager HUD (Heads-Up Display), a „Client” gombot kell megnyomni. A szerver-kliens csatlakozás dedikált szerver módban történik, amely azt jelenti, hogy a szerver számítógép mindig ugyanaz, amelyhez szeretnénk csatlakozni, amelyen fut az egészségügyi program. A szerver számítógép IP címét kell beírni a kliensen és a szerveren Networkmanager HUD kijelzőjén a localhost szó helyére.

Elsőként a szervert kell elindítani, utána a kliens programot. A szerver oldalon a tűzfalon az első indításnál engedélyezni kell az alkalmazást. (18. kép)

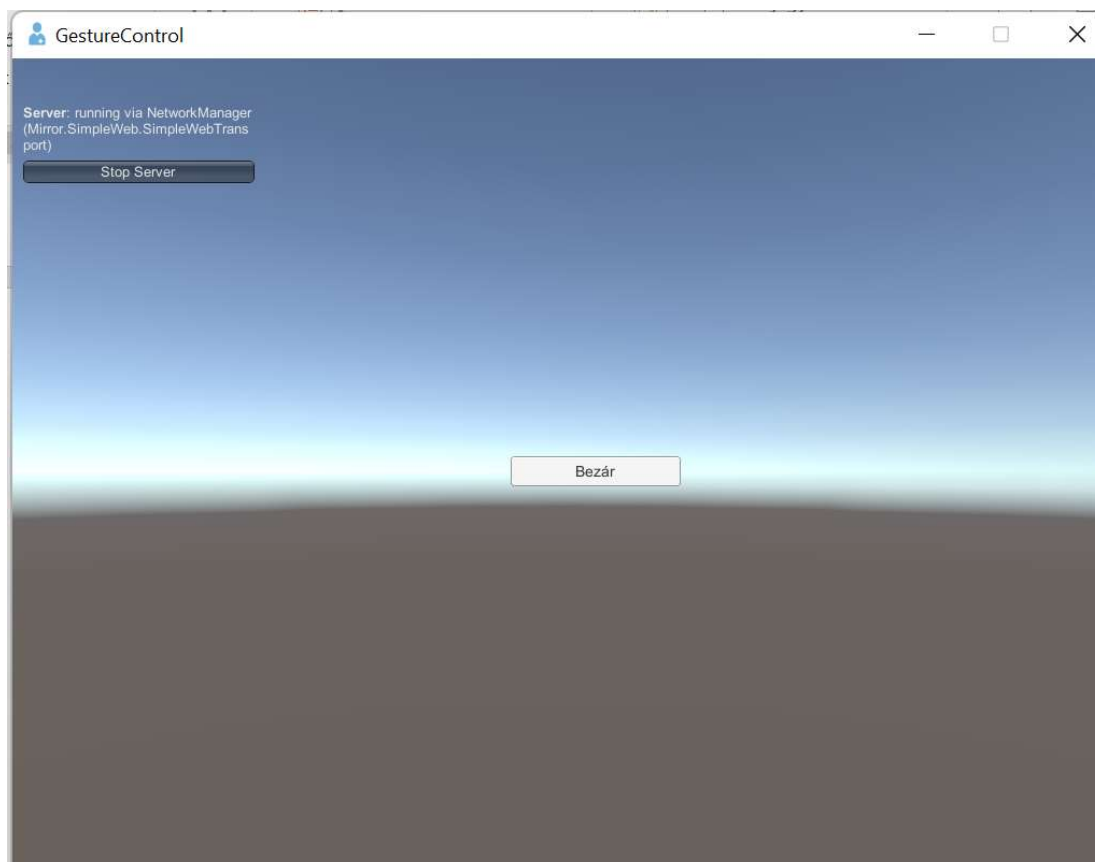


18. kép: A szerver oldali alkalmazás tűzfalon való átengedése

Miután a csatlakozás létrejött, a mobilalkalmazás NetworkManager HUD kezelőpaneljén (19. kép) és a számítógépen futó alkalmazás NetworkManager HUD kezelőpaneljén (20. kép) a következők láthatók:



19. kép: A mobiltelefon képernyőjén a NetworkManager HUD panelje csatlakozás után



20. kép: A számítógépen a szerver oldali program NetworkManager HUD panelje csatlakozás után

A létrejött websocket kapcsolatnál, a mobilalkalmazás virtuális menü gombja gesztusvezérlés segítségével elindítja a szerver oldalon a böngészőt, az Óbudai Egyetem kezdőoldalával.

Alkalmazás bezárása

A virtuális bezár gombot jelölő ikonra gesztussal való vezérlése után, 5 másodperc elteltével aktiválódik az alkalmazás bezárása.

A fejlesztés eredményeként egyszerű eszközökkel, mobiltelefonon keresztül megvalósulhat websocket kommunikáción keresztül a mobileszköz számítógéphez való csatlakozása, egy program indítása és vezérlése. A szakdolgozat célja, hogy egészségügyben használt számítógép összekapcsolható legyen mobilalkalmazással, amelyen virtuális, érintés nélküli vezérlő gombok vannak kiépítve és egy program indítása megtörténjen, megvalósult.

Amire ügyelni szükséges a virtuális gombokkal/ikonokkal ellátandó felhasználói felületen, hogy a felhasználói felület, amelyen a virtuális gombok meg fognak jelenni, nem túl nagy felület és a gesztusokkal történő vezérlés az ikonoknál helyet kíván.

Ez azt jelenti a gyakorlatban, hogy a vezérelni kívánt alkalmazást célszerű olyan menüpontokkal felépíteni, amelynél a kijelzőn maximum 4-5 db menüpont látható.

6 KONKLÚZIÓ - A MEGOLDÁS HASZNÁLHATÓSÁGÁNAK ÉRTÉKELÉSE

Alapérvényű igazság az, hogy minden megoldás annyira jó, amennyire jól használható.

A gesztussal való vezérlésnek az egészségügyben a fejlődés alapját a szórakoztatóiparban fejlesztett infravörös mélységkamerával rendelkező Kinect eszköz nyújtotta. A műtési eljárásokat folyamatosan fejlesztve, elemezve, orvosok által összehasonlító elemzések készültek, mely alapján a műtét közbeni leggyorsabb és legpontosabb megoldásnak, a képpalkotó program az orvos saját maga által történő egerrel és billentyűzettel történő kezelése, értéke adta. Ezen művelettel ugyanakkor az orvos megsértette a steril környezetet és nagyon sok időt vett igénybe az ismételt bemosakodás. A következő legjobb megoldás, a gesztusvezérléssel történő műtési eljárások használata volt, amelyhez még elég kevés lehetőség adott, de minden egyes fejlesztéssel javul a helyzet.

Szakdolgozatomban arra kerestem a választ, hogy vezérelhetőek-e gesztusvezérlés megoldással az egészségügyben jelenleg használt eszközök. Ha vezérelhetőek, akkor milyen megoldás a legmegfelelőbb az érintés nélküli vezérléshez.

Jól használható, működő megoldást találtam a kiterjesztett valósággal készített mobilalkalmazás fejlesztéssel.

A mobilapplikáció használható tableteken, telefonnal, amely költséghatékony megoldást nyújt a kivitelezéshez. A gesztussal történő vezérlést az alkalmazás kiterjesztett valósággal történő megoldása nyújtja. Az applikáció segítséget ad, megoldást nyújt a fertőzésveszélyes környezetben, műtét közben a számítógép kezelésére vagy olyan helyen, ahol nagy lenne a megteendő távolság a kezelni kívánt beteg és a kezelést végző gép között.

Az egészségügyi rendszernek szüksége van technológiai újításokra, amelyek növelik a betegek kezelésének hatékonyságát, javítják műtési környezetet és támogatja az orvosok munkáját fertőzésveszélyes környezetben. A gesztussal vezérelt mobilapplikációt ilyen jelentős technológiai újításnak gondolom.

7 CONCLUSION – THE EVALUATION OF THE USEFULNESS OF SOLUTION

It is a fundamental truth that every solution is as good as well it can be used.

The Kinect device with an infrared depth camera developed in the entertainment industry provided the basis for gesture control development in healthcare. By continuously developing, analyzing the surgical procedures, comparative analyzes were developed by physicians based on which the fastest and most accurate solution for operations, the imaging program which can be operated by the doctor themselves with mouse and keyboard, provided the value. However, with this operation, the doctor violated the sterile environment and rewashing took a lot of time. The next best solution was to use gesture-controlled surgical procedures, for which there are a few options, but with each development the situation improves.

In my thesis I was looking for the answer to whether the devices currently used in healthcare can be controlled with a gesture control solution. If they can be controlled, which is the best solution for no-contact control.

I found a workable solution that is a mobile application which is developed with augmented reality.

The mobile application can be used on tablets, phones, providing cost-effective solution for implementation. Gesture control is provided by the augmented reality solution of the application. The application provides help and solution in an infection-prone environment, for handling a computer during surgery, or in a place where there is a large distance to be covered between the patient to be treated and the machine performing the treatment.

The healthcare system needs technological innovations that increase the effectiveness of patient care, improve the surgical environment, and support the physicians' work in infectious environments. I reckon the gesture-driven mobile application is such a significant technological innovation.

8 IRODALOMJEGYZÉK

- [1] „About Technology - denizatm.com,” [Online]. Available: <https://hu.denizatm.com/pages/50607-voice-recognition-is-awesome-but-how-did-it-get-so-good>.
- [2] P. Nadkarni, „Core Technologies: Machine Learning and Natural Language Processing,” [Online]. Available: <https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/hidden-markov-model>.
- [3] R. A. Bolt, „Put-That-There: Voice and Gesture,” Architecture Machine Group, pp. 262-270.
- [4] J. L. Shahzad Malik, „Visual Touchpad: A Two-handed Gestural Input Device,” Department of Computer Science, <http://www.dgp.toronto.edu/>.
- [5] R. P. Moniruzzaman Bhuiyan, „A Gesture Controlled User Interface for Inclusive Design and Evaluative Study of Its Usability,” *Journal of Software Engineering and Applications*, pp. 513-521., 2011. 4..
- [6] H. H. H. T. L. K. Y. Pei Jia, „Industrial Robot: Head gesture recognition for hands-free control of an intelligent wheelchair,” 2007. [Online]. Available: <https://www.emerald.com/insight/content/doi/10.1108/01439910710718469/full/html> (2021.12.14).
- [7] A. D. W. a. J. O. W. Meredith Ringel Morris, „Microsoft Research One Microsoft Way: User-Defined Gestures for Surface Computing,” 2009. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2009/04/SurfaceGestures_CHI2009.pdf (2021.12.12).
- [8] D. K. –. L. D. I. –. S. L. –. R. BÉLA-GERGELY, „MAGYARORSZÁGI GYÁRTÓEGYSÉGEK IPAR 4.0 GYAKORLATÁNAK ELEMZÉSE – TECHNOLÓGIA, STRATÉGIA, SZERVEZET,” 2020. [Online]. Available: http://unipub.lib.uni-corvinus.hu/5428/1/VT_2020n4p2.pdf (2021.12.15).
- [9] B. Rommel, „BimmerTech - BMW gesture control - the next level of iDrive interaction,” [Online]. Available: <https://www.bimmer-tech.net/blog/item/124-bmw-gesture-control> (2021.12.16).

- [10] S. E. T. Archivum, „3.6 Mozgásérzékelők,” [Online]. Available: http://eta.bibl.u-szeged.hu/2193/3/36_mozgsrzkelk.html (2021.12.17).
- [11] „Ultraleap - Leap Motion ControllerTM,” [Online]. Available: https://www.ultraleap.com/datasheets/Leap_Motion_Controller_Datasheet.pdf (2021.12.17).
- [12] A. S. G. G. T. C. H. M. G. Kenton O’Hara, „Communications of the ACM - Touchless Interaction in Surgery,” 2014. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/ID20CACM-12-06-1404_final.pdf (2021.12.16).
- [13] V. D.-F. B. H. L. f Wipfli, „Gesture-Controlled Image Management for Operating Room: A Randomized Crossover Study to Compare Interaction Using Gestures, Mouse, and Third Person Relaying,” 2016. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4833285/> (2021.12. 06).
- [14] E. R. C. D. G. D. P. P. B. M. C. C. & F. C. A. H. Butt, „BioMedical Engineering OnLine - Objective and automatic classification of Parkinson disease with Leap Motion controller,” 12. November 2018. [Online]. Available: <https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/s12938-018-0600-7> (2021.12.17) .
- [15] R. T. Azuma, „A Survey of Augmented Reality, Presence: Teleoperators and Virtual Environments,” pp. pp. 355-385., 1997. augusztus.
- [16] I. R.-R. J. D. é. I. A.-S. J. Mota, „Augmented reality mobile app development for all,” *Comput. Electr. Eng.* 2017, 65, 250–260., pp. 250-260. o., 2017, 65, .
- [17] K. Willis, „University of Alberta,” 24. január 2018. [Online]. Available: <https://www.ualberta.ca/science/news/2018/january/augmented-reality-tech-see-under-skin-without-scalpel.html> (2021.12.17).

9 ÁBRAJEGYZÉK

<i>1. ábra: Gesztusérzékelésen alapuló beviteli eszköz tesztelése</i>	<i>11</i>
<i>2. ábra: Rendszerterv</i>	<i>30</i>
<i>3. ábra: Unity Multiplayer High Level API</i>	<i>35</i>

10 KÉPJEGYZÉK

1. kép: Fejmozdulatot érzékelő gesztussal vezérelt kerekesszék.....	12
2. kép: A gesztusvezérlés technológiai életciklusa 2013 (Kép forrása: Gartner)	14
3. kép: A gesztusvezérlést segítő eszközök technológiai életciklusa 2016 (Kép forrása: Gartner)	15
4. kép: BMW gépjárműtípusok gesztusvezérlés eszközrendszere (Kép forrása: BimmerTech).....	17
5. kép: Gesztusvezérlés használata mobiltelefonon	18
6. kép: Microsoft Kinect érzékelőinek megfigyelési pontjai [10]	19
7. kép: Leap Motion eszköz a Parkinson betegség diagnosztizálása közben.....	21
8. kép: Nem steril környezetben információ keresés és bevétel [12]	23
9. kép: Gesztusvezérlés az érsebészetben [12]	24
10. kép: Kiterjesztett valóság alkalmazása az egészségügyben [17].....	28
11. kép: 1. generációs Hololens okos szemüveg (kép forrása: www.microsoft.com).....	29
12. kép: Képernyőn megjelenő GestureControl alkalmazás ikonja.....	31
13. kép: A kiterjesztett valóság felületet indító QR kód.....	32
14. kép: GestureControl alkalmazás mobiltelefon képernyője QR kódról indítva	33
15. kép: Kliens-Szerver kapcsolat Unity Network kapcsolattal.....	36
16. kép: Fejlesztőkörnyezet websocket beállítása	37
17. kép: Szerver-kliens közötti kapcsolódás ellenőrzése a fejlesztőkörnyezetben.....	38
18. kép: A szerver oldali alkalmazás tűzfalon való átengedése.....	39
19. kép: A mobiltelefon képernyőjén a NetworkManager HUD panelje csatlakozás után	39
20. kép: A számítógépen a szerver oldali program NetworkManager HUD panelje csatlakozás után	40

11 FORRÁSKÓDOK

SimpleWebTransport forráskódja:

```
using System;
using System.Net;
using System.Security.Authentication;
using UnityEngine;
using UnityEngine.Serialization;

namespace Mirror.SimpleWeb
{
    [DisallowMultipleComponent]
    public class SimpleWebTransport : Transport
    {
        public const string NormalScheme = "ws";
        public const string SecureScheme = "wss";

        public ushort port = 7778;

        public int maxMessageSize = 16 * 1024;

        public int handshakeMaxSize = 3000;

        public bool noDelay = true;

        public int sendTimeout = 5000;

        public int receiveTimeout = 20000;

        public int serverMaxMessagesPerTick = 10000;
    }
}
```

```

public int clientMaxMessagesPerTick = 1000;

[Header("Server settings")]

public bool batchSend = true;

public bool waitBeforeSend = false;

[Header("Ssl Settings")]
public bool clientUseWss;

public bool sslEnabled;
public string sslCertJson = "./cert.json";
public SslProtocols sslProtocols = SslProtocols.Tls12;

[Header("Debug")]
[FormerlySerializedAs("logLevels")]
[SerializeField] Log.Levels _logLevels = Log.Levels.none;

/// <summary>
/// <para>Gets _logLevels field</para>
/// <para>Sets _logLevels and Log.level fields</para>
/// </summary>
public Log.Levels LogLevels
{
    get => _logLevels;
    set
    {
        _logLevels = value;
        Log.level = _logLevels;
    }
}

```

```

    }

    void OnValidate()
    {
        Log.level = _logLevels;
    }

    SimpleWebClient client;
    SimpleWebServer server;

    TcpConfig TcpConfig => new TcpConfig(noDelay, sendTimeout,
receiveTimeout);

    public override bool Available()
    {
        return true;
    }

    public override int GetMaxPacketSize(int channelId = 0)
    {
        return maxMessageSize;
    }

    void Awake()
    {
        Log.level = _logLevels;
    }

    public override void Shutdown()
    {
        client?.Disconnect();
        client = null;
        server?.Stop();
        server = null;
    }

```

```

    }

    #region Client

    string GetClientScheme() => (sslEnabled || clientUseWss) ? SecureScheme :
NormalScheme;

    string GetServerScheme() => sslEnabled ? SecureScheme : NormalScheme;

    public override bool ClientConnected()
    {
        return client != null && client.ConnectionState !=
ClientState.NotConnected;
    }

    public override void ClientConnect(string hostname)
    {
        // csatlakozás vagy csatlakozva már
        if (ClientConnected())
        {
            Debug.LogError("Már csatlakozva");
            return;
        }

        UriBuilder builder = new UriBuilder
        {
            Scheme = GetClientScheme(),
            Host = hostname,
            Port = port
        };

        client = SimpleWebClient.Create(maxMessageSize,
clientMaxMessagesPerTick, TcpConfig);
        if (client == null) { return; }
    }

```

```

        client.onConnect += OnClientConnected.Invoke;
        client.onDisconnect += () =>
        {
            OnClientDisconnected.Invoke();
            // kliens törlése
            // kapcsolat megszüntetése után nincs több üzenet
            client = null;
        };

        client.onData += (ArraySegment<byte> data) =>
        OnClientDataReceived.Invoke(data, Channels.Reliable);

        client.onError += (Exception e) =>
        {
            OnClientError.Invoke(e);
            ClientDisconnect();
        };

        client.Connect(builder.Uri);
    }

    public override void ClientDisconnect()
    {
        client?.Disconnect();
    }

    public override void ClientSend(ArraySegment<byte> segment, int channelId)
    {
        if (!ClientConnected())
        {
            Debug.LogError("Nincs csatlakozás");
            return;
        }
    }

```

```

        if (segment.Count > maxMessageSize)
        {
            Log.Error("Uzenet nagyobb, mint a maximum meret");
            return;
        }

        if (segment.Count == 0)
        {
            Log.Error("Uzenetek szama 0");
            return;
        }

        client.Send(segment);

        // esemény hívása
        OnClientDataSent?.Invoke(segment, Channels.Reliable);
    }

    // üzenetek feldolgozása
    public override void ClientEarlyUpdate()
    {
        client?.ProcessMessageQueue(this);
    }
    #endregion

    #region Server
    public override bool ServerActive()
    {
        return server != null && server.Active;
    }

    public override void ServerStart()

```

```

{
    if (ServerActive())
    {
        Debug.LogError("SimpleWebServer már elindítva");
    }

    SslConfig config = SslConfigLoader.Load(sslEnabled, sslCertJson,
sslProtocols);

    server = new SimpleWebServer(serverMaxMessagesPerTick, TcpConfig,
maxMessageSize, handshakeMaxSize, config);

    server.onConnect += OnServerConnected.Invoke;
    server.onDisconnect += OnServerDisconnected.Invoke;
    server.onData += (int connId, ArraySegment<byte> data) =>
OnServerDataReceived.Invoke(connId, data, Channels.Reliable);
    server.onError += OnServerError.Invoke;

    SendLoopConfig.batchSend = batchSend || waitBeforeSend;
    SendLoopConfig.sleepBeforeSend = waitBeforeSend;

    server.Start(port);
}

public override void ServerStop()
{
    if (!ServerActive())
    {
        Debug.LogError("SimpleWebServer nem aktiv");
    }

    server.Stop();
    server = null;
}

```

```

public override void ServerDisconnect(int connectionId)
{
    if (!ServerActive())
    {
        Debug.LogError("SimpleWebServer nem aktiv");
    }

    server.KickClient(connectionId);
}

public override void ServerSend(int connectionId, ArraySegment<byte>
segment, int channelId)
{
    if (!ServerActive())
    {
        Debug.LogError("SimpleWebServer nem aktiv");
        return;
    }

    if (segment.Count > maxMessageSize)
    {
        Log.Error("Az uzenet nagyobb, mint a maximalis meret");
        return;
    }

    if (segment.Count == 0)
    {
        Log.Error("Uzenetek szama 0");
        return;
    }
}

```

```

server.SendOne(connectionId, segment);

// esemény hívása
OnServerDataSent?.Invoke(connectionId, segment, Channels.Reliable);
}

public override string ServerGetClientAddress(int connectionId)
{
    return server.GetClientAddress(connectionId);
}

public override Uri ServerUri()
{
    UriBuilder builder = new UriBuilder
    {
        Scheme = GetServerScheme(),
        Host = Dns.GetHostName(),
        Port = port
    };
    return builder.Uri;
}

// üzenetek feldolgozása
public override void ServerEarlyUpdate()
{
    server?.ProcessMessageQueue(this);
}
}
}

```