Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Távközlési és Médiainformatikai Tanszék

Illés Tamás

DIGITÁLIS IKER VIZUALIZÁCIÓJA KITERJESZTETT VALÓSÁGBAN IPARI KÖRNYEZETBEN

TANSZÉKI KONZULENS

Dr. Varga Pál

BUDAPEST, 2023

Tartalomjegyzék

Kivonat

Az emberi érzékelést vizsgálva megállapítható, hogy a bennünket érő ingerek nagy részét vizuális formában észleljük, így az életünk legtöbb területén erősen támaszkodunk a látó szervrendszerünkre, hiszen látásunk segítségével tájékozódunk, azonosítunk objektumokat, figyelünk biztonságunkra, végezzük munkánkat, illetve gyakran a szórakozásunknak is elengedhetetlen része. Az érzékelésünk erre a formájára fektetett nagy hangsúly, valamint a vizuális élmények gazdagságának lehetőségei a technológia fejlődésével elkerülhetetlenné tették, hogy informatikai megoldásokkal átlépjünk a hagyományos látás korlátain.

Ebben szolgál segítségül a kiterjesztett valóság (Augmented Reality, AR) mely digitális információkkal és virtuális objektumokkal nyitja meg érzékelésünk új dimenzióit. Általa nyerhetünk széleskörű támogatást olyan területeken, mint például az oktatás, szórakozás, művészetek, sport, egészségügy, közlekedés, turizmus, építészet, tervezés, gyártás, karbantartás és a logisztika. Ahogy a hétköznapokban, a kiterjesztett valóság az ipari gyakorlatban is ígéretes eszköznek bizonyul, hiszen egy ilyen jellegű dinamikus, interaktív, intuitív és felhasználóbarát adatmegjelenítés használata csökkentett hibaszázalékot, gyorsabb és pontosabb döntéshozatalt, hatékonyabb elemzést, megnövelt biztonságot, ezáltal javuló termelékenységet eredményez.

Az ipari gyártórendszerek egyre népszerűbb tervezési módszere a virtuális beüzemelés digitalizációs eljárása, mely során a gyártósorunk vezérlő-logikájának fejlesztését el tudjuk végezni annak digitális ikrén. A digitalizáció számos előnye ellenére azonban gyakran bizonyulhat nem elég kézzelfoghatónak, hiszen a kiberfizikai rendszerek strukturált architektúrájának és szerkezetének elemi szintű átlátása sok esetben igényel magasabb szintű absztrakciós gondolkodást, mint egy fizikai infrastruktúra esetében. A kiterjesztett valóság (AR) az ilyen virtuális infrastruktúrák vizualizálására egy alkalmas immerzív technológia.

Dolgozatomban egy ipari folyamatnak a kiterjesztett valóság technológiájával való vizuális támogatásának megtervezését és implementálását tárgyalom. Ebben egy PLC (Programable Logical Controller) által vezérelt, virtuálisan beüzemelt gyártósor fizikai infrastruktúráját egészítem ki digitális ikrének a valóságban nem létező modellelemeinek megjelenítésével és illesztésével. Az általam készített megoldás

amellett, hogy a szó szoros értelmében látványosan segít a tervezés alatt álló eszközök és a meglévő fizikai eszközök illeszthetőségének vizsgálatában, a későbbiekben is reprezentatív és interaktív módon nagyban hozzá tud járulni az üzem operátori munkáihoz.

Abstract

In general, humans perceive the majority of stimuli in a visual form, making our visual sensory system a critical foundation in many aspects of our lives. We heavily rely on our visual system to navigate, identify objects, ensure our safety, perform work tasks, and often for entertainment purposes. The emphasis placed on this form of perception, along with the possibilities of rich visual experiences, has made it inevitable to transcend the traditional limitations of vision with technological solutions.

Augmented Reality (AR) serves as a tool in this context, opening up new dimensions of perception with digital information and virtual objects. It provides extensive support in areas such as education, entertainment, arts, sports, healthcare, transportation, tourism, architecture, design, manufacturing, maintenance, and logistics. Just as it has become a part of our everyday lives, augmented reality proves to be a promising tool in industrial practices. Such dynamic, interactive, intuitive, and user-friendly data visualization reduces error rates, facilitates faster and more accurate decision-making, enhances efficiency in analysis, improves safety, and ultimately leads to increased productivity.

An increasingly popular design approach in industrial manufacturing systems is the digitalization process of virtual commissioning, where the development of control logic for manufacturing lines can be carried out in their digital twin. Despite the numerous advantages of digitalization, it may not always seem tangible enough, as understanding the structured architecture and elements of cyber-physical systems often requires higher-level abstract thinking compared to physical infrastructure. Augmented reality is an immersive technology suitable for visualizing such virtual infrastructures.

In this paper I discuss the design and implementation of visual support for an industrial process using augmented reality technology. In this context, I augment the physical infrastructure of a manufacturing line, controlled by a PLC (Programmable Logical Controller), with the visualization and integration of digital twins representing non-existent elements in the physical reality. The solution I have developed not only spectacularly aids in assessing the compatibility between the upcoming equipments and existing physical assets but also contributes significantly to the operational tasks of the plant operators in a representative and interactive manner.

1 Bevezetés

Legtöbbünkről elmondható, hogy nélkülözhetetlenül tekintünk látásunkra, mely érzékelésünk számos formái között joggal örvend kiemelt fontosságú szerepnek, hiszen környezetünk vizuális érzékelésére az életünk legtöbb területén támaszkodunk. Nem véletlen, hogy az evolúció által ilyen precíz és összetett szervrendszer valósítja meg, mint a szemeink. Percepciónknak ennek a formájára fektetett nagy hangsúly indokolja a kiterjesztett valóság által nyújtható szolgáltatásokban rejlő magas potenciált, hiszen előnyeit majdnem minden tevékenységünk során élvezhetjük, amihez szükséges a látásunk.

A kiterjesztett valóság hétköznapokat támogató előnyein túl az ipari alkalmazásában is egyre nagyobb használati potenciál és igény mutatkozik, hiszen az AR jelentheti az ember-gép interfészek (Human Machine Interface, HMI) következő technológiai lépcsőjét, mely által az operátori munka soha nem látott ergonómiára, hatékonyságra és biztonsági szintre emelhető, ami egy jelentős szempont a jövő ipara felé vezető úton. Az Ipar 4.0 (Industry 4.0) által bevezetett digitalizációs eljárások kiterjesztett valóság eszközökkel és szolgáltatásokkal való támogatása, valamint az AR-támogatott digitális iker (AR-assisted Digital Twin, AR-enabled Digital Twin) koncepciójával kiegészítve a gyártásban résztvevő dolgozókat okos operátorokká alakíthatja, mely egy megkerülhetetlen lépcső az Ipar 5.0 (Industry 5.0) által egy központi szemléletként [1] tárgyalt kollaboratív robotokkal való emberi munkamegosztás, vagyis az ember-robot kollaboráció (Human-Robot Collaboration, HRC) eléréséhez. Az emberi tevékenységek ergonómiáját, kényelmét és hatékonyságát az ipari szempontokon túl, azokat általánosítva és kiterjesztve a Társadalom 5.0 (Society 5.0) [2] emberközpontú, az életszínvonalat modern technológiai megoldásokkal javító koncepciója is szem előtt tartja, mely tovább növeli a kiterjesztett valóság globális, felhasználási területeken átívelő relevanciáját.

2 Technológiai háttér

Ebben a fejezetben kerül tárgyalásra a feladat megvalósítása során felhasznált technológiák, eszközök bemutatása és műszaki hátterük részletezése, mely információk ismerete nagyban hozzájárul a tervezési szempontok megválasztásához.

2.1 Kiterjesztett valóság

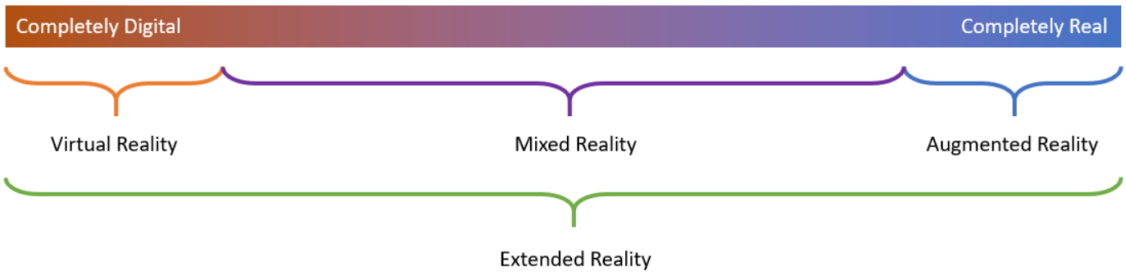
A kiterjesztett valóság egy interaktív vizuális technológia, melynek lényege a digitális információk valós idejű integrálása a felhasználó környezetébe. Ellentétben a virtuális valósággal (Virtual Reality, VR) [3], amely teljesen mesterséges környezetet hoz létre, az AR-felhasználók valós környezetet tapasztalnak meg, amely valamilyen észlelt, vagy generált információkkal van kiegészítve digitálisan. A kiterjesztett valóság célja, hogy valamilyen módon vizuálisan megváltoztassák a természetes környezetet, vagy, hogy további információkat, illetve gazdagabb, interaktívabb élményt nyújtsanak a felhasználóknak. Ennek egy marketing célzatú ábrázolása figyelhető meg a 1. ábrán.

1. ábra: A kiterjesztett valóság egy marketing célzatú ábrázolása. [4]

A kiterjesztett valóság koncepciójának következő lépcsője a virtuális objektumok integrálása után a velük való interakció megteremtése, melyet definíció szerint inkább a

kevert, vagy vegyes valóság (Mixed Reality, MR) tartalmazza. A vegyes valóság tehát azáltal fejleszti a kibővített valóságot, hogy túllép a virtuális tartalom egyszerű rávetítésén az igazi világra. Virtuális objektumokat hoz a felhasználó fizikai környezetébe, lehetővé téve a zökkenőmentes interakciót és integrációt a két valóság között. Ez a technológia nagy lehetőséget kínál különböző területeken, beleértve a játékot, az oktatást, a tervezést és sok más területet, mivel lehetővé teszi az elmerülést és interaktív élményeket, amelyek elmosódnak a fizikai és virtuális világok határai között.

Az Extended Reality (XR) egy olyan fogalom, amely magában foglalja az Augmented Reality (AR), Virtual Reality (VR) és Mixed Reality (MR) technológiákat. Az XR-t gyakran az immerzív vizuális technológiák spektrumaként írják le, amelynek két végpontja az AR és a VR. Az AR azon a végponton helyezkedik el, ahol a virtuális tartalmakat valós környezetbe helyezzük, például a mobilalkalmazások vagy okos szemüvegek segítségével. A VR az XR spektrum másik végpontját jelenti, ahol a felhasználók teljesen elmerülnek a virtuális világban. Az XR középső tartománya a Mixed Reality (MR), amely az AR és a VR elemeket egyesíti. Ezt a spektrumot szemlélteti a 2. ábra.



2. ábra: Az Extended Reality spektruma. [5]

2.2 A kiterjesztett valóságot megvalósító hardverek

A kiterjesztett valóságot megvalósító hardvereszközök kulcsfontosságú szerepet játszanak a technológia életre keltésében. Ezeknek a készülékeknek több formája is létezik a mobiltelefonoktól és táblagépektől a dedikált szemüvegekig (HMD, Head-Mounted Display), mint például a Microsoft HoloLens, a Google Glass Enterprise vagy a Magic Leap One, mely eszközök kamerákkal, szenzorokkal, erős számítási kapacitással és valamilyen kijelzőtechnológiával vannak felszerelve, hogy a virtuális elemeket szervesen össze tudják kapcsolni a valósággal. Ezeknek a hardverkomponenseknek a

minősége kulcsfontosságú szerepet játszanak a kiterjesztett valóság által nyújtható interaktív felhasználói élmény megteremtésében. Egy ilyen felhasználásra alkalmas eszköz figyelhető meg a 3. ábrán.

3. ábra: A Microsoft HoloLens 2 kiterjesztett valóság szemüveg. [6]

Az AR eszközöknek elengedhetetlen része egy nagy felbontású kijelző különösen akkor, ha az a látóterünk közvetlen közelében helyezkedik el, hiszen ellenkező esetben a valóságba integrált virtuális objektumok, vagy akár az egészében megtapasztalt vizuális élmény alacsony minősége, részletessége miatt jelentősen romolhat a felhasználói élmény. Egy kiterjesztett valóság szemüveg esetében továbbá a megjelenítőnek transzparensnek is kell lennie, így téve lehetővé, hogy a fogyasztó egyszerre tapasztalhassa meg a valós, illetve vetített tartalmakat, ezzel nyújtva vegyített forrású vizuális percepció lehetőségét.

Ezek az eszközök különböző érzékelőket használnak, például kamerákat, giroszkópokat, gyorsulásmérőket és mélységérzékelőket, hogy pontosan követhessék a felhasználónak és környezetének pozícióját, mozgásait, orientációját. Azt, hogy a virtuális tartalmakat a lehető legnagyobb pontossággal illeszthessük a valós világhoz, ezek a szenzorok teszik lehetővé, hatékonyan összehangolt működésük kulcsfontosságú

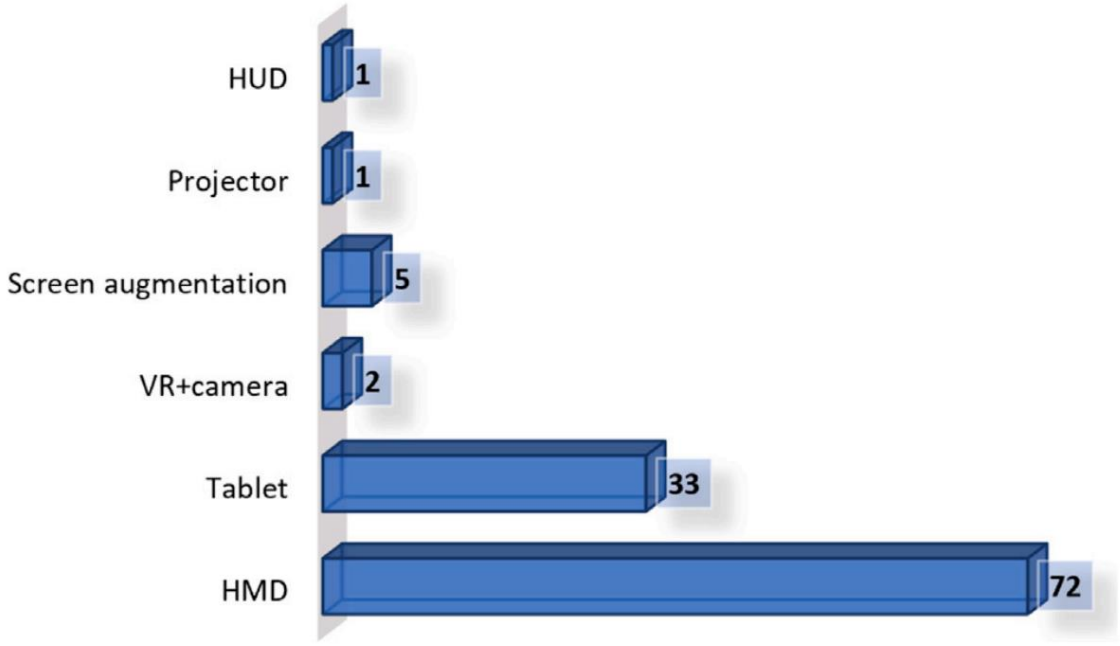
ahhoz, hogy a felhasználó nagy precizitású pozícionálást tapasztalhasson az AR alkalmazásokban.

Az érzékelők által precízen elhelyezett információk, objektumok sok esetben bonyolult, nagy felbontású, sok poligonból álló modellek, melyek valós idejű renderelése és megjelenítése nagy számítási kapacitást igényel, vagyis erős processzorra, illetve grafikus vezérlésre van szükség. A nagyteljesítményű hardveres erőforrások használata azonban sok energiát igényel, továbbá nagy hőkeletkezéssel jár, a nagy akkumulátor, illetve hűtőrendszer jelentősen megnövelheti az eszköz kiterjedését, illetve súlyát, melyek rontják az eszköz hatékonyságát, használatának kényelmét, fontos tehát, hogy az eszközök tervezésekor ezeknek a tényezőknek egy kompromisszumos határértékét vegyék figyelembe. A véges megengedhető számítási erőforrásokra való tekintettel ennek okán kiemelten fontos szerepet kap, hogy optimalizált futásidejű alkalmazások készüljenek a stabil képfrissítésű és reagálóképes AR élmény fenntartására.

A korlátozott hardveres kapacitásra megoldást jelenthet az eszköz nagy sávszélességű vezetéknélküli kommunikációval való ellátása, például Wi-Fi-vel, vagy mobilkommunikációval, hiszen sok esetben a valós idejű renderelésnél kevesebb számítást és energiát igényel egy kidelegált erőforrás által előállított adatstream hálózati úton való fogadása és megjelenítése. A vezetéknélküli interfész továbbá lehetőséget nyújt külső szolgáltatások igénybevételére, más eszközökkel való szinkronizációra, illetve adatok megosztására is.

A kiterjesztett valóság élménye akkor teljes, ha az interakció lehetősége is meg van teremtve, így az AR-t megvalósító eszközök esetében fontos, hogy az ezt lehetővé tevő módszerek rendelkezésre álljanak. Ennek okán szerelik fel őket például érintőképernyővel, hangalapú vezérelhetőséggel, gesztusfelismeréssel és kézkövetéssel, így biztosítva a lehető leginteraktívabb felhasználói élményt.

Ezek voltak azok a szempontok, melyeket érdemes figyelembe venni egy fejlesztés előtt álló AR applikáció célhardverének megválasztásához a tervezett felhasználási esetek igénye szerint. Sajnos a hardver kiválasztásának napjainkban még jelentős anyagi vonzata is van, azonban a kiterjesztett valóság eszközök piacának meglehetősen diverz kínálata megadja a lehetőséget a nagy választék közül való választásra. A 4. ábráról leolvasható, hogy 2022 júliusáig bezárólag 111 kiterjesztett valósággal foglakozó tudományos publikáció közül a táblagépek és a HMD-k voltak a legnépszerűbb AR eszközök.



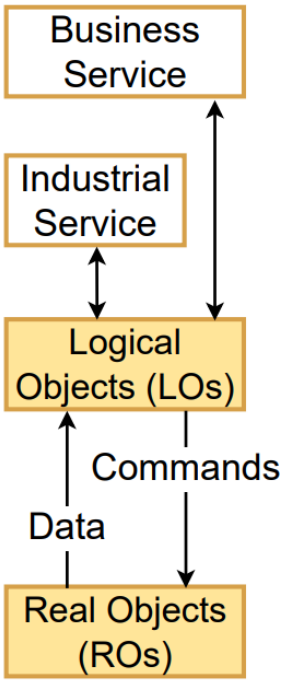
4. ábra: A kiterjesztett valóság eszközök használatának előfordulása az áttekintett tudományos publikációkban 2022. júliusáig. [7]

2.3 Digitális Iker (Digital Twin, DT)

A digitális iker fogalmát elsőként Dr. Michael Grieves vezette be a termék életciklus menedzsment (Product Lifecycle Management, PLM) koncepciós modelljeként 2002-ben, kezdeti meghatározásában a digitális ikerhez három összetevő kell: a fizikai objektum, a virtuális modell, valamint köztük megteremtett kapcsolat [8].

A kifejezés használata ennek következtében először a gyártáshoz kapcsolódó területeken terjedt el, idővel azonban bekerült a dolgok internete (Internet of Things, IoT), illetve a kiberfizikai rendszerek (Cyber-Physical Systems, CPSs) területeinek tudományos közösségeinek szóhasználatába is, így a fizikai termékek tervezésének és gyártásának eredeti elképzelésén túl a fogalom általános keretrendszerré bővülhetett, mely bármilyen fizikai vagy immateriális objektum digitalizációjára megfeleltethető. A digitális iker definíciója használatának kezdeti szakaszában annak folyamatos fogalmi bővülése miatt tehát nehezen meghatározható volt, szerencsére azóta már több jól körülhatárolható megfogalmazások is született többek között a negyedik ipari forradalom, vagyis az Ipar 4.0 (Industry 4.0) koncepciója által, azonban a növekvő számban felszínre törő innovatív kutatási eredmények és új felhasználási területek mai napig bővítik.

A digitális iker az eddig legfrissebb definíciója szerint egy fizikai, vagy akár nem kézzelfogható objektum virtuális reprezentációja. Ez a koncepció egy valós tárgy klónozását jelenti egy szoftverbeli megfelelőjébe, egy logikai objektumba, mely karakterisztikáját és tulajdonságait működési és életciklusbeli szakaszától függően a valós objektummal kölcsönösen tükrözi. A logikai objektumnak szolgáltatásorientált interfészt kell biztosítania a más logikai objektumokkal és szoftveres szolgáltatásokkal való zökkenőmentes interakció és kompozíció létesítésére annak érdekében, hogy elősegítse a valós és logikai elemek együttműködését, egyidejű fejlesztését.



5. ábra: A digitális ikrek szoftverarchitekúrájának logikai ábrázolása. [9]

A digitális ikreket bővülő felhasználási eseteinek köszönhetően egyre több ipari területen alkalmazzák, ilyenek például az anyagáram szimuláción alapuló vezérlőlogika tervezés, a virtuális üzembehelyezés alapú szoftverfejlesztés, infrastruktúratervezés és bővítés, valamint egyéb integrációhoz, monitorozáshoz, teszteléshez és üzemeltetéshez kapcsolódó ipari folyamatok. Egy tipikus digitális iker alkalmazási terület szoftverarchitektúrája figyelhető meg az 5. ábrán.

2.4 Virtuális üzembe helyezés (Virtual Commissioning)

A modern ipari gyártórendszerek, illetve azok integrátorai egyre nagyobb kihívásokkal kell, hogy szembenézzenek a versenyképesség fenntartása érdekében. A

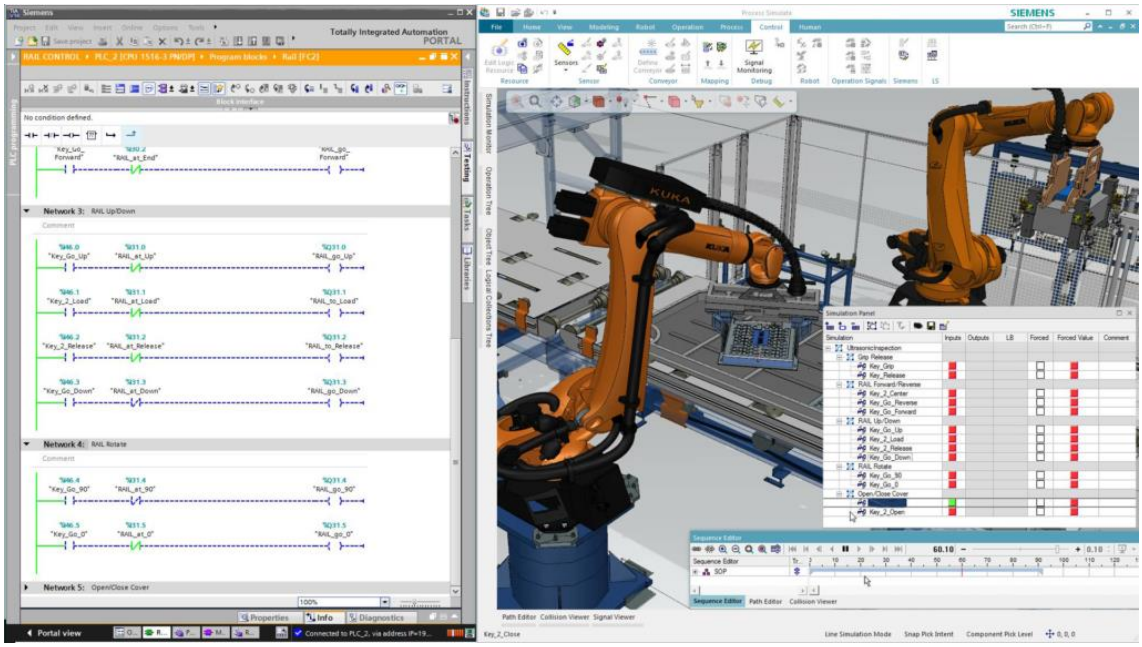
rohamosan fejlődő technológiai megoldások alkalmazása, mint a fejlett robotika, autonóm járművek, számítógépes látórendszerek és mesterséges intelligencia nagy szerepet játszanak egy rugalmas termelési létesítmény üzemeltetésében, azonban integrálásuk a gyártásba komoly komplexitású rendszert eredményeznek. A kihívás mértékét tovább fokozzák a folyamatosan növekvő, és gyakran változó ügyfél igények, valamint a 6. ábrán is megfigyelhető, gyártási technológiákat övező növekvő piaci trendek, mint a növekvő termék komplexitás, az emelkedő mértékű költségnyomás, a csökkenő piacra kerülési idő, a rugalmas gyárakhoz bevezetett új paradigmák és a növekvő szerelési automatizálás.

6. ábra: A gyártási technológiákat övező piaci trendek. [10]

Egy ilyen magas technológiai szintre helyezett termelő létesítmény irányításáért felelős komponenseinek vezérlőlogikával való felprogramozása hagyományos esetben az infrastruktúra beszerzését követően a gépészeti, elektrotechnikai és vezérléstechnikai elemek fizikai telepítésével egyidőben, valamint nagyobb részt azt követően zajlik. Ez a rendszer- és automatizálási integrátorok számára már önmagában is kihívásokkal teli, illetve nagy kockázattal járó feladat, az esetleges hibák vétése pedig sok mérnökórányi javításhoz, vagy még rosszabb esetben a gépek károsodásához vezethet.

Erre a problémahalmazra szolgál megoldásul a virtuális üzembe helyezés digitalizációs eljárás technológiája, melynek lényege, hogy a gyártórendszerünket vezérlő szoftvereinket képesek lehetünk kifejleszteni a termelő infrastruktúra virtuális megfelelőjén, vagyis digitális ikerpárján, annak fizikai telepítése, vagy akár komponenseinek beszerzése előtt, így, mikor sor kerül a valódi beüzemelésre, addigra már egy magas készültségi fokú, minden bizonnyal hibamentes szoftver állhat rendelkezésünkre. Ezt 3D szimulációval ellátott CAD/CAM/CAE (Computer-Aided Design, Computer-Aided Manufacturing, Computer-Aided Engineering) szoftverek teszik lehetővé, melyek közös tervezőfelületet biztosítanak mind a gépész-, villamos- és

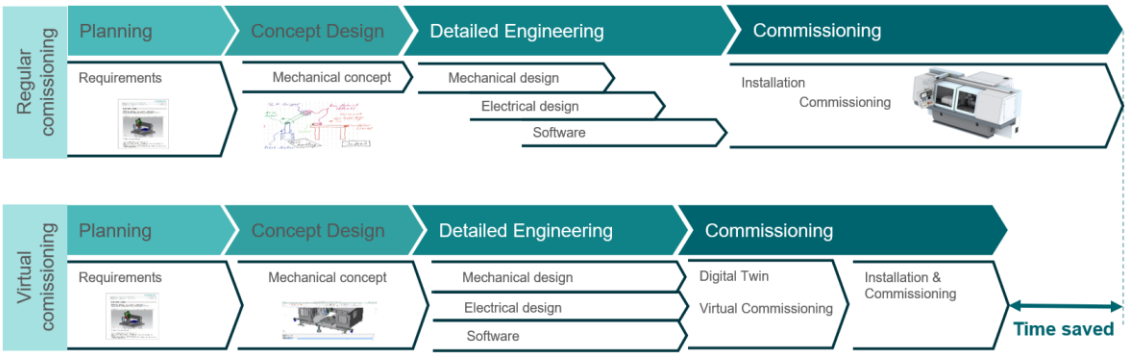
automatizálási mérnökök számára egyaránt, valamint párhuzamosan. A 7. ábrán egy ilyen szoftver használat közbe figyelhető meg.



7. **ábra:** Virtuális üzembe helyezés a TIA Portal és a Process Simulate VC Lite szoftverek segítségével. [10]

Ezek széles körű hardverkatalógussal, valamint valósághű kinematikán és fizikán alapuló modellezéssel járulnak hozzá tetszőleges méretű és komplexitású gyártórendszerek digitális ikrének megalkotásához. Ennek elkészültét követően a komponenseinek valós megfelelőjének működését szimulálva képesek lehetünk lefejleszteni azok vezérlőszoftvereit, mely vezérlési logikákat, mechanikai viselkedéseket akár korai HIL (hardware-in-the-loop) és SIL (software-in-the-loop) tesztelésnek is alávethetünk, így virtuálisan megvizsgálhatjuk a termelés minden aspektusát, mielőtt a gyártás elkezdődne. Kifejlesztett szoftvereinket virtuális infrastruktúránkon rendszerszintűen is végrehajthatunk realisztikus validációt, ami a valós idejű állapotmonitorozás mellett a mechanikai mozgások vizualizálásával lehetővé teszi az olyan jellegű hibák detektálását is, mint a helytelen anyagáramlás, események bekövetkezésének hibás sorrendje, valamint optimalizálatlan és hibás működés. Az így előállított, a valós beüzemelés idejére már magas készültségi fokkal rendelkező és hibamentes szoftverek rendelkezésre állása a telepítés helyszíni munkálatait több hétről akár néhány napra csökkentik, mely mindamellett, hogy előbb élesíthető rendszert és termelést eredményez, jelentős megtakarítást biztosít a beruházási költségek terén is. A

virtuális üzembe helyezés hagyományossal szemben nyújtott előnyeit a 8. ábra hivatott szemléltetni.



8. ábra: A virtuális és hagyományos üzembe helyezés összehasonlítása. [10]

A virtuális üzembe helyezés tehát számos előnnyel járul hozzá a termelői létesítmények megtervezéséhez, és előállításához, ez azonban nem azt jelenti, hogy a beüzemelést követően a digitális ikerre ne lenne szükségünk, hiszen a virtuális infrastruktúra valós idejű szimulációja és szinkronizációja a fizikaival rendkívül jó lehetőséget ad a különböző adatgyűjtő és diagnosztikai szolgáltatások számára, hogy központosított helyről férjenek hozzá az egyes statisztikák forrásául szolgáló rendszeradatokhoz. További előnye egy virtuálisan beüzemelt gyártórendszernek, hogy az esetleges átalakításokat, bővítéseket képesek lehetünk a tervezési folyamat elejéhez hasonló módon a virtuális térben előkészíteni, kipróbálni az egyes fejlesztések implementációjának optimalizált módját. Mindezt anélkül tehetjük meg, hogy a fizikai infrastruktúrán bármi módosítást kelljen végrehajtanunk, mely garantálja, hogy a valós beavatkozások sorra kerülésekor egy rövid termelésleállás ideje alatt egy jól tesztelt, hibamentes átalakítás vagy bővítés kerüljön telepítésre és illesztésre.

A virtuális üzembe helyezést megvalósító szoftverek önmagukban számos területet lefedve segítik a gyártórendszerek tervezését, ezáltal teljes keretrendszert nyújtanak a digitális ikrek előállításához és használatához. Fontos azonban még megemlíteni, hogy a 3D CAD/CAM/CAE jellegnek köszönhetően sok tervezőprogram natívan támogatja a virtuális infrastruktúra VR általi vizsgálatát is a megfelelő eszközök segítségével, mely további kapukat nyithat akár az ügyféllel közös tervezési fázisban, melynek részeként akár szó szerint körbe is vezethetjük megrendelőnket a még meg nem épült gyárában a világon bárhonnan. A szoftverek továbbá a digitális iker realisztikus és méretarányos 3D modelljei által lehetőséget adnak azok külön mozgó elemenként való

exportálására is, melyet felhasználva akár saját VR, AR applikációkat is fejleszthetünk a gyártórendszerünk operátorainak támogatására.

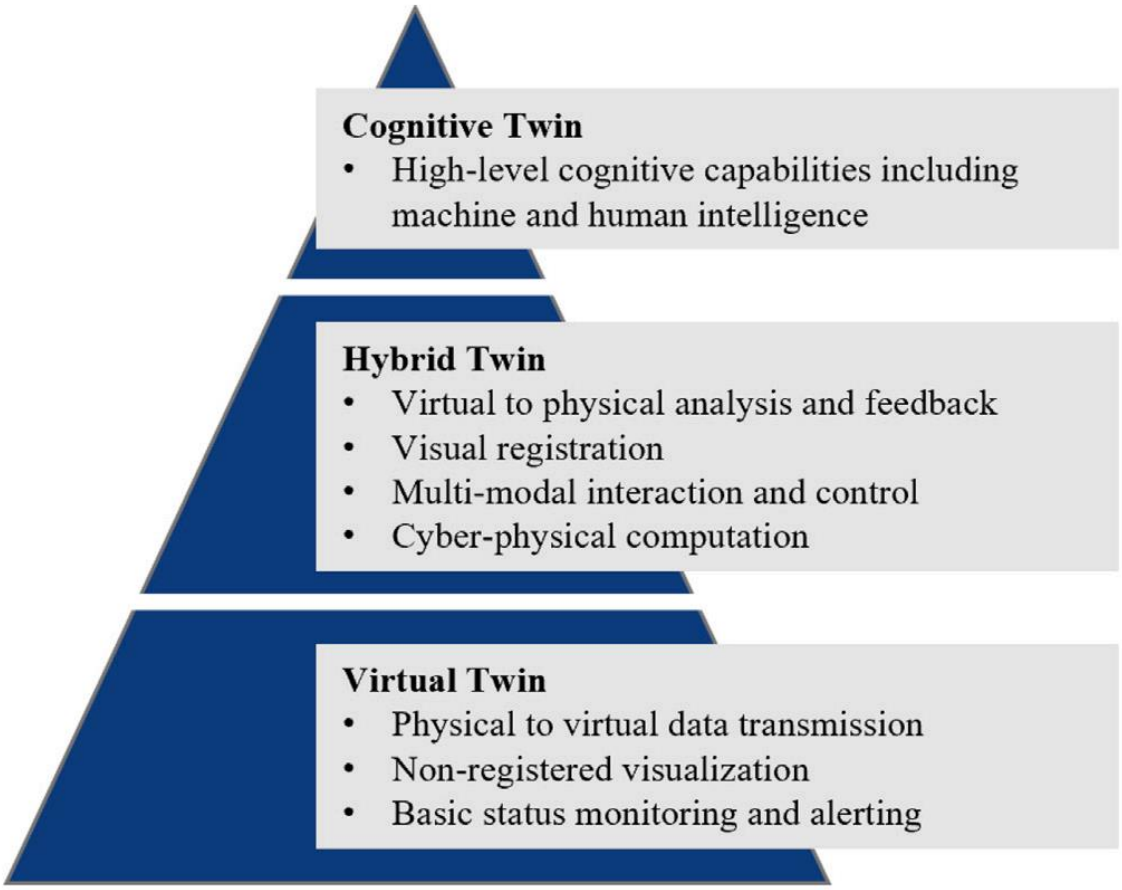
2.5 AR-támogatott digitális iker (AR-assisted/enabled Digital Twin)

Az Ipar 4.0 által bevezetett digitalizációs eljárások jelentős előrelépéssel segítették a kiberfizikai rendszerek elterjedését, mely egy mai napig fejlődő tudományterület [11]. Következő lépcsőjének egy ígéretes koncepciója az AR-támogatott digitális iker koncepciója, mely kiterjesztett valóság eszközökkel és szolgáltatásokkal emelik a virtuális reprezentációk funkcióbázisát. Az AR ugyanis egy természetes, rugalmas és gazdag lehetőségeket rejtő adatmegjelenítő technológia, melyre az ipar az ember-gép interfészek (HMI) egyfajta továbbfejlesztéseként tekint, emiatt pedig rendkívül népszerű kutatási terület mind az akadémiai, mind az ipari körökben.

A kiterjesztett valóság és a digitális ikrek technológiák kombinációja egy kiemelten fontos trend lehet a jövő iparában, hiszen amellett, hogy a gyártórendszerek és termékelőállítás minden életciklusbeli szakaszában hasznosnak bizonyulhat, eszközei és szolgáltatásai által nagy szerepet játszhat a gyári alkalmazottak okos operátorrá formálásában, mely az első technológiai lépcsőt jelentené a humán kiberfizikai rendszerek irányába. A kollaboratív robotokkal való emberi munkamegosztás, az intuitív robotirányítási módszerek, a vizuális robotprogramozás, a gesztusalapú, vagy akár kéz nélküli robotinterakció, mint hangfelismerés, vagy fejmozgás szemkontaktus követés mind olyan interakciós módszerek, melyeket mind a kiterjesztett valóság tesz lehetővé. Ezek olyan intuitív, illetve számos alternatívát magában foglaló vezérlési technikák, melyek lehetővé teszik, hogy nem szakértő operátorokat, vagy akár munkában akadályozottakat is alkalmazzunk. Az AR-támogatott digitális ikrek használata az imént felsorolt lehetőségekkel, illetve az általuk megteremthető személyre szabott, rugalmas és ergonomikus munkakörnyezettel támogatja az ember-robot kollaborációt, valamint a humánközpontú intelligens gyártást, mely az Ipar 5.0-nak egy kiemelt fontosságú elve.

Az AR-támogatott digitális iker, habár egy forró kutatási terület [7], már létezik jól körülhatárolható funkcióbázisa, mely elemeinek fejlettségi szintjétől függően kategorizálható be három különböző dimenzióba. Ezek a virtuális iker, a hibrid iker és a

kognitív iker, melyekről fontos megjegyezni, hogy nem három különböző felhasználási típust fednek le, hanem inkább egymásra épülve alkotnak egy hierarchikusan összefüggő modellt, mely a 9. ábrán figyelhető meg.



9. ábra: Az AR-támogatott digitális ikrek három dimenziójának hierarchikus modellje. [7]

2.5.1 Virtuális iker (Virtual Twin)

A virtuális iker az AR-támogatott digitális ikrek legalsó hierarchiai szintjét elfoglaló dimenziója, mely a hagyományos digitális iker alapkoncepciójának feljavítását jelenti a kiterjesztett valóság eszközök segítségével. A fizikai infrastruktúráról a virtuális modell felé irányuló egyirányú információáramlás jellemzi, mellyel egy fejlett ember-gép interfész (HMI) felhasználási esetét valósítja meg például a folyamatok szenzoralapú állapotellenőrzésére, figyelmeztetések, riasztások, rendellenes állapotok detektálásának vagy prediktálásának vizualizálására egy rugalmas, személyre szabott, intuitív adatvizualizációs felületen az operatív személyzet számára. Ez a dimenzió továbbá magában foglalja még az egyes helyszíni információkon alapuló kiterjesztett valóság

funkciókat, mint a jelölőalapú (marker-based) adatmegjelenítés például vonalkódok, QR (Quick Response) kódok beolvasására aktválódó adatmegjelenítés.

A virtuális iker funkcióit tekintve egy komoly, operátori munkát segítő szintlépése a digitális ikreknek, azonban korlátozott technológiai mozgásteréből adódóan a kiterjesztett valóság által nyújtott előnyöket nem használja ki teljes mértékben.

2.5.2 Hibrid iker (Hybrid Twin)

A hibrid iker AR-támogatott digitális iker dimenzió koncepciója a virtuális ikerrel ellentétben, illetve azon túl a vizuális analízis és visszacsatolás felé összpontosít, melyet kétirányú információáramlás, valamint multimodális interakció és irányítás jellemez. A digitális iker a fizikai térből összegyűjtött, illetve az AR által gazdagított adatokkal, mint a jelölők (marker-ek), detektált objektumok, felismert folyamatok, kiszámított kiberfizikai interakció vagy az ember-robot interakcióhoz végzett ütközésérzékelés információit felhasználva végez elemzést, mely alapján szimulációt, diagnózist, predikciót és optimalizálást, biztonsági beavatkozást képes végrehajtani, vagy akár az operátor döntéshozatalát képes támogatni.

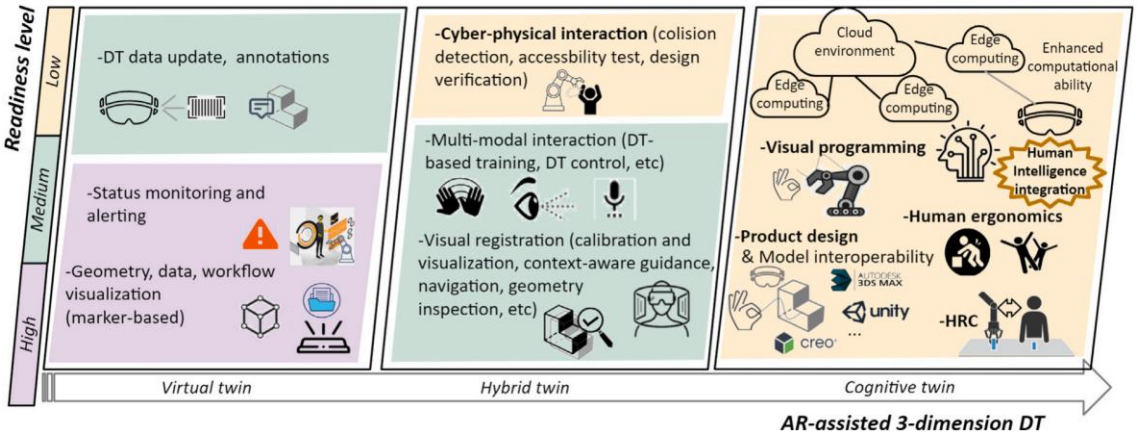
A hibrid iker mindamellett, hogy rugalmasan teszi hozzáférhetővé a digitális iker állapotait, széles spektrumú beavatkozási lehetőséget biztosít azokba, ezáltal a fizikai működésbe. Az operátorok fizikai gombokkal, aktuátorokkal, beavatkozó szervekkel való manuális interakciója a közvetlen fizikai jelenlét miatt adott esetben az egészségre káros, nem ergonómikus, vagy akár veszélyes munkakörülményeket idézhet elő. A hibrid iker koncepciójában képesek lehetünk felruházni őket egy AR által támogatott operátorbarát ember-gép interfésszel (HMI), mely által a fizikai berendezések működésébe digitális ikrükön keresztül lehet beavatkozni. Ezt az interakciós lehetőségek széles választéka biztosítja, mint virtuális vezérlőfelület, gesztusalapú irányítás, kéz nélküli interakciós lehetőségek, mint a fejmozgások, szemkontaktus és hangalapú vezérlés.

A kiberfizikai interakció számításához tartozik a virtuális és fizikai környezet hozzáférhetőségének, tervezésverifikációjának és ütközésdetektálásának vizsgálata, mely kiemelt hangsúlyt követel meg a kollaboratív humán-robot hatékony munkamegosztás és együttműködés mellett biztonságkritikus kockázati szempontokból.

2.5.3 Kognitív iker (Cognitive Twin)

A kognitív iker egy olyan AR-támogatott digitális iker koncepció, mely nevéből is adódóan magas szintű kognitív képességekkel rendelkezik ideértve mind a gépi, mind az emberi intelligenciát, melyek által összetett, nehezen prediktálható problémákat hivatott megoldani. Ezeket magas szenzorikájú és hardveres támogatású, hordható AR eszközökkel valósítanák meg, melyek a virtuális és hibrid iker tulajdonságain túlmutatva egy mozgó végpontú, kollektív adatgyűjtő és -feldolgozó Edge-Cloud architektúrát megvalósítva, elosztott, vagy delegált számítási kapacitással képezhetik a lehető legpontosabb digitális iker realizációját a fizikai térnek. A kognitív iker megvalósítása joggal tölti be az AR-támogatott digitális ikrek dimenzióinak csúcsát, hiszen az ötvözött emberi és mesterséges intelligencia jelentheti egy modern, magas szintű okos gyártás (Smart Manufacturing) kapuját, mely lerombolja az Ipar 5.0 humánközpontú koncepciója előtt húzódó falakat. A különböző modern informatikai megoldások feladata az emberek számára a robotokkal való együttműködés és munkamegosztás biztonságos módjának létesítésén kívül egy olyan ember-gép kommunikáció megteremtése, mely lehetővé teszi az egymás kölcsönös megértését, ezáltal elmosva a humán és gépi kompetencia között húzódó határokat. A gépi tanulás által adaptívan és dinamikusan cselekvő robotok, valamint a betanított modellekkel ellátott AR eszközök segítette okos operátorok soha nem látott humán-robot kollaborációt valósíthatnak meg az általuk betöltött kiberfizikai térben, melynek mai szemmel elméleti akadálya már nincsen, megvalósításában viszont kis túlzással még tudományos fantasztikumnak tűnhet.

Az AR-támogatott digitális ikrek három dimenziójának felhasználási eseteit a 10. ábra szemlélteti.

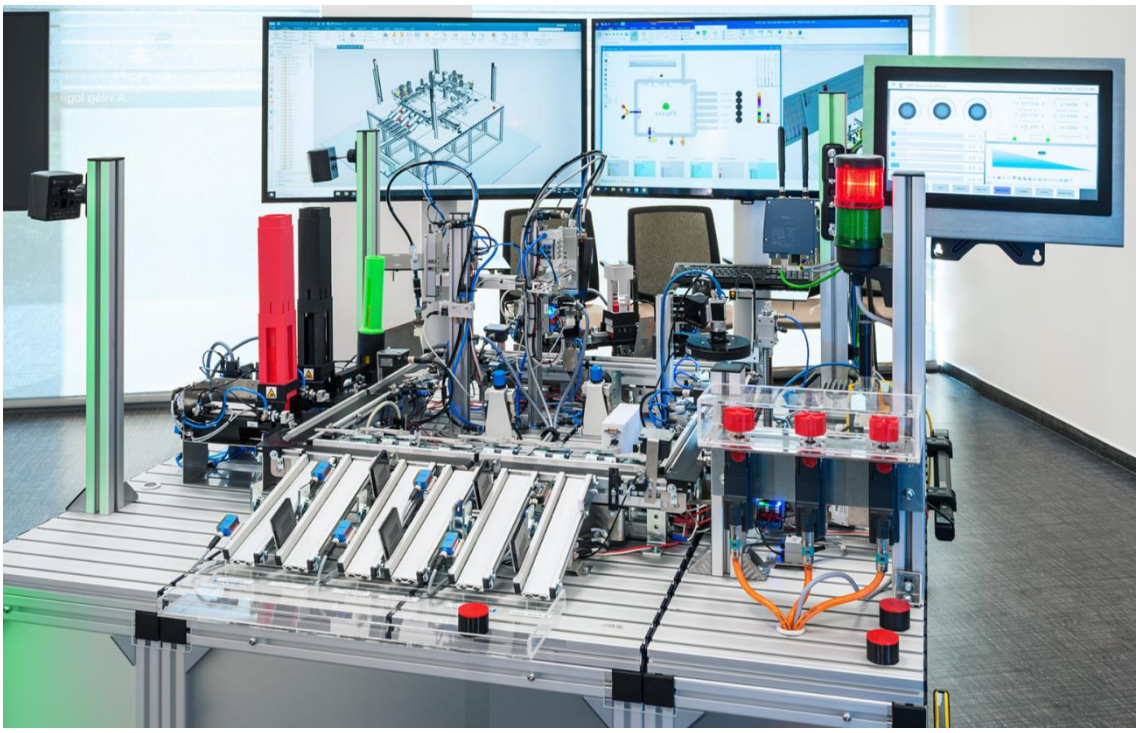


10. ábra: Az AR-támogatott digitális ikrek három dimenziójának felhasználási esetei és azok elérhetősége. [7]

3 Tervezési feladat és a fejlesztés környezete

3.1 Tervezési feladat

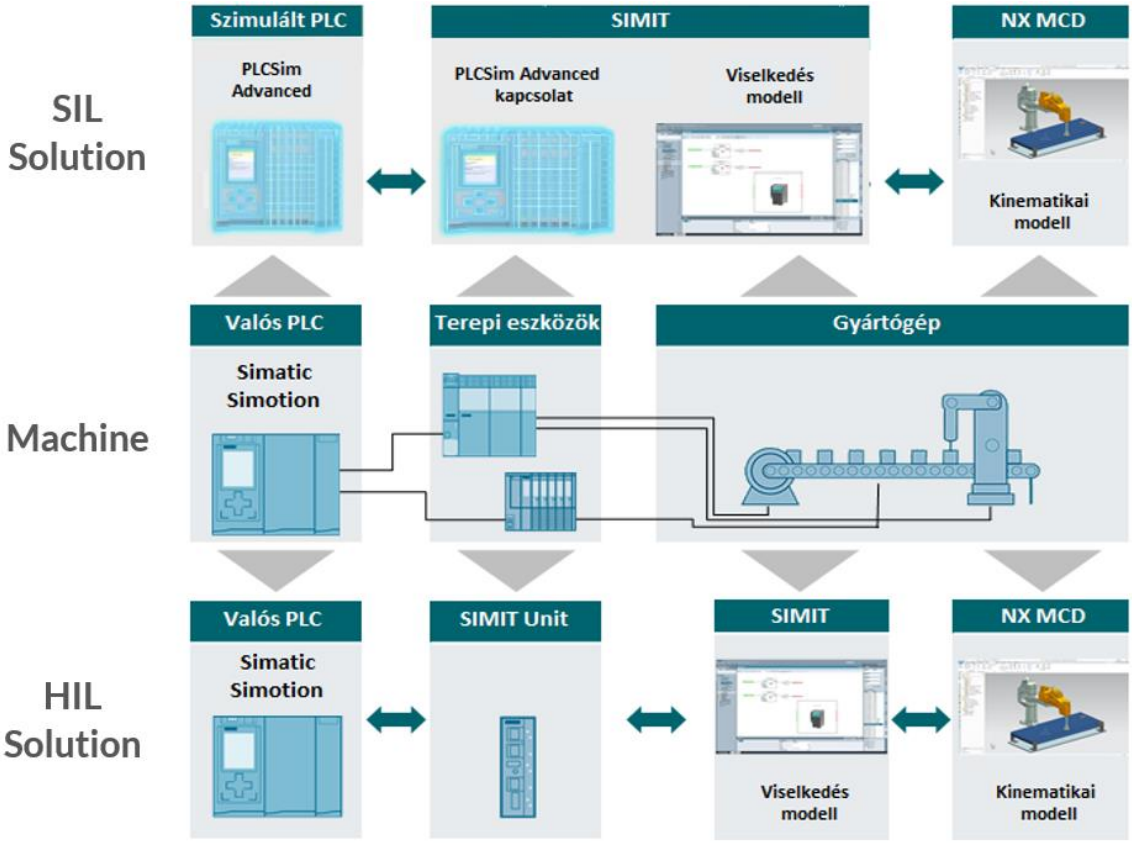
A kitűzött feladat egy SIMATIC S7-1500 PLC által vezérelt gyártóberendezésnek, valamint digitális ikerpárjának AR technológiájával való támogatása az AR-támogatott digitális iker koncepciójának lehető legmagasabb szintjén.



11. ábra: Az evosoft Hungary Kft. budapesti telephelyén bemutatott gyártósor modell.

Ez a gyártóberendezés, mely a 11. ábrán tekinthető meg, az evosoft Hungary Kft. budapesti telephelyén található, célja demonstrációs jellegű, mivel termékelőállítás helyett a Siemens által nyújtott legmodernebb automatizációs és digitalizációs technológiákat hivatott együttműködve bemutatni. A gyártósor modell egy három alkotóelemből álló termék elemeinek adagolókkal, futószalagokkal és aktuátorokkal automatizált összeillesztésén keresztül mutatja be a digitalizációs eljárások előnyeit a gyártási folyamatokra, és a termék életciklus menedzsmentre (PLM), hiszen a gyártósor különlegessége, hogy virtuális üzembe helyezés technológiájával készült. Ennek lépései a gyártósor realisztikus 3D modelljének megtervezése, benne a vezérlésért felelős PLC, HMI panel, hajtásvezérlésekkel a Siemens NX [33] CAD/CAM/CAE szoftverben. Ezt

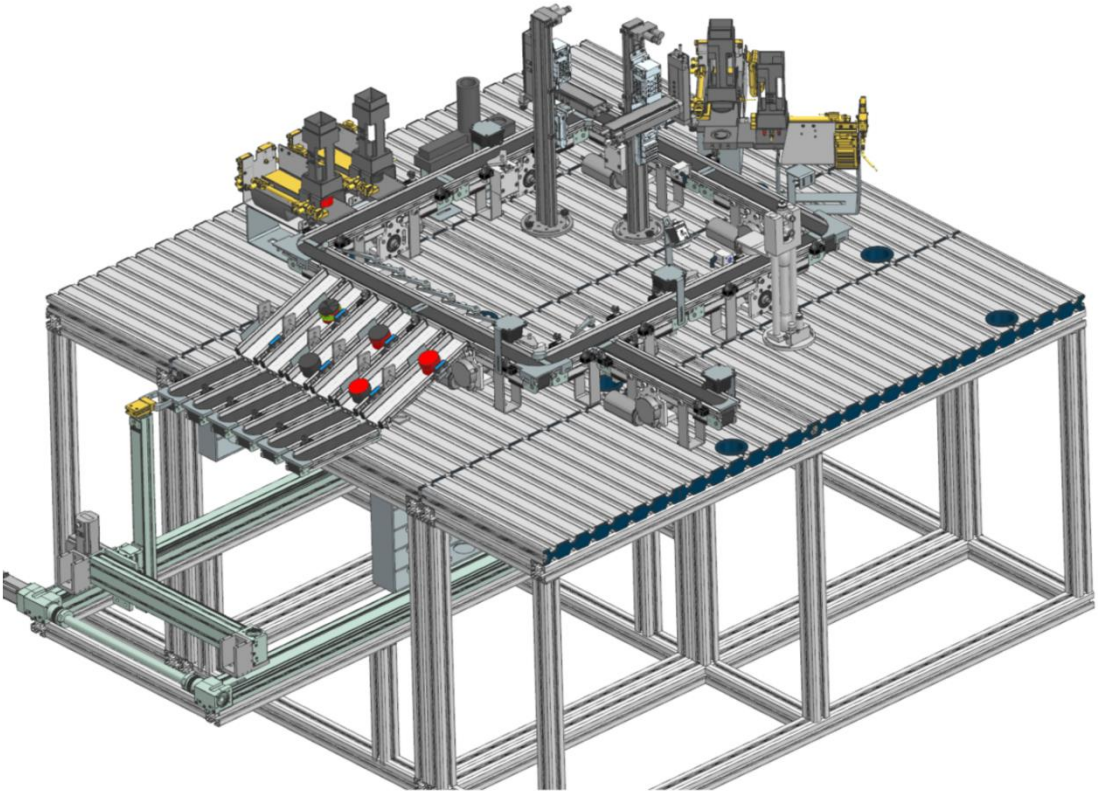
követi a gyártósor optimális elrendezésének és vezérlőlogikájának megállapítása anyagáramszimulációval a Technomatix Plant Simulation szoftverben. A virtuális üzembehelyezés utolsó fázisa a megtervezett berendezés elemeinek és működésének valós idejű szimulációja a Siemens SIMIT szoftverrel, mely által a szimulált irányítástechnikai eszközök konfigurációja és programkódjai a megállapított vezérlőlogika alapján elkészíthetőek a TIA Portal (Totally Integrated Automation Portal) fejlesztői környezetben. Ennek blokkvázlatszerű szemléltetése figyelhető meg 12. ábrán.



12. ábra: A virtuális üzembe helyezés által létrehozott digitális iker valós idejű szimulációja. (HIL: hardware-in-the-loop, SIL: software-in-the-loop) [12]

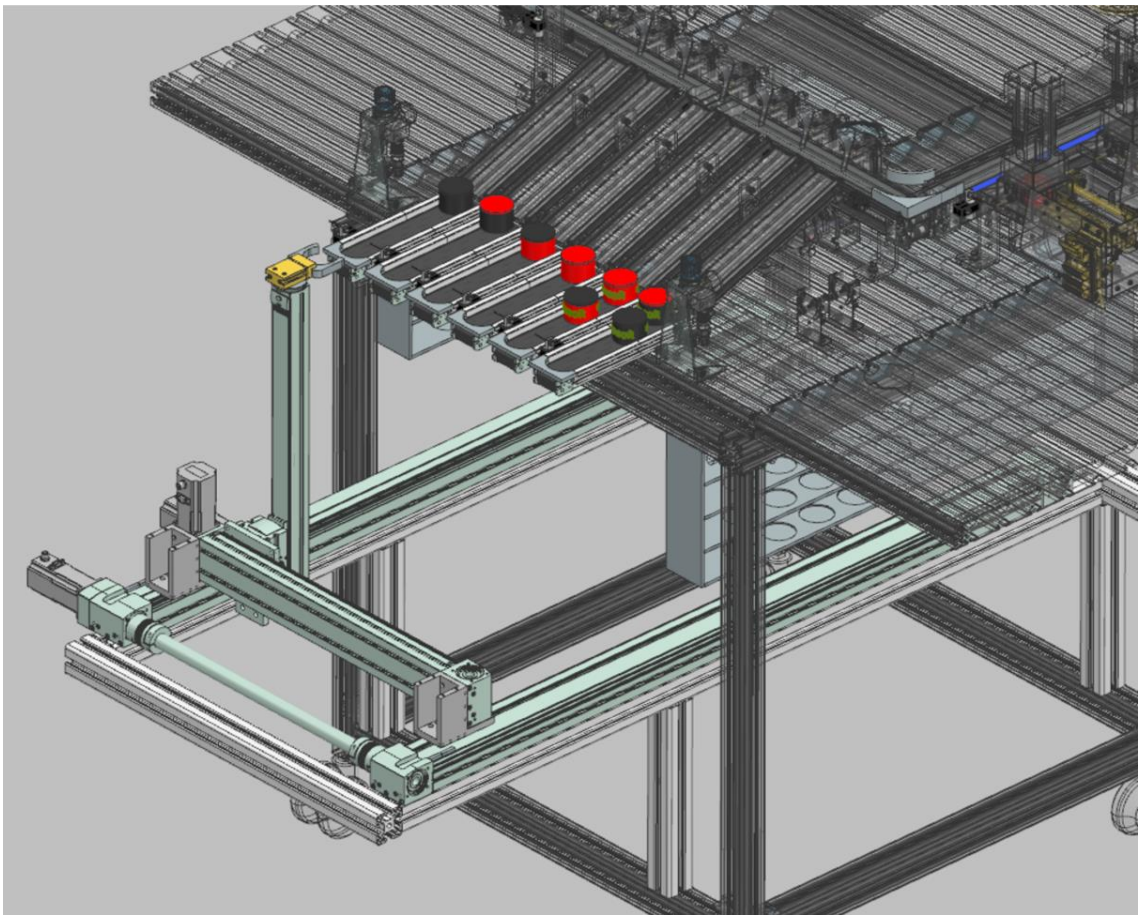
Ennek a digitalizációs tervezési eljárásnak a használata nem csak a fizikai telepítés kivitelezését segítette, hiszen a virtuális üzembe helyezés technológiájának alapvető, de nem másodlagos melléktermék a berendezés digitális ikerpárja, mely alkalmazása nem csak a gyártóüzem megtervezésének fázisban kamatoztatható, hanem annak bővítése során is. Ennek köszönhető ugyanis, hogy míg a megépített fizikai gyártósoron az elkészült termékek egy plexitálcára érkeznek, a digitális ikerben szereplő párjuk egy a virtuális térben már lefejlesztett, viszont a valóságban még meg nem épült háromtengelyű raktározó robotkar által elhelyezésre kerülnek egy szintén virtuális

alsóraktárban. A robotkar három tengelyén való elmozdulást a PLC által egy-egy szervomotor vezérli, melynek köszönhetően a határain belül tetszőleges 3D koordinátába képes mozogni. A kar végén egy forgatható megfogószerv (gripper) található, melynek szárai összeszorításával képes a termékeket magához venni, majd elengedésével letenni az alsóraktár polcainak szabad helyeire.



13. ábra: Az evosoft Hungary Kft. gyártósor modelljének digitális ikre.

A feladatom ennek a 13. ábrán megfigyelhető digitális ikerben funkcionáló, fizikailag meg nem épített raktárrobotnak, melyet a 14. ábra szemléltet, a valós térben való vizualizálása, a berendezéshez való illesztése, valamint a működésének támogatása a kiterjesztett valóság technológiájának segítségével. A cél tehát a gyártósor digitális ikrének továbbfejlesztése AR-támogatott digitális ikerré (AR-assisted Digital Twin) egy olyan alkalmazás készítésével, mely az AR-támogatott digitális ikrek lehetőleg minél magasabb dimenziójának feleljen meg. Az implementálandó felhasználási eseteket tehát ennek figyelembevételével választottam meg.



14. ábra: A gyártósor modell digitális ikrének raktárrobot alrendszere.

Első és kiinduló felhasználási esetnek a raktárrobot méretarányos és realisztikus digitális 3D modelljének a valós térbe való integrációját választottam, mely a gyártósorhoz illeszkedve a digitális ikerbeli működésnek megfelelően végrehajtja a PLC utasításait, vagyis a gyártósorról legördülő termékeket elhelyezi az alsóraktárban.

A vizualizáció ennek a formájának, vagyis a virtuális 3D objektumok fizikai térben való megjelenítésének fontos része a kitakarás (occlusion) jelensége, vagyis annak figyelembevétele, hogy a felhasználó szemszögéből amennyiben egy virtuális objektum egy fizikaival egy vonalban, viszont nála távolabb helyezkedik el, akkor a fizikai objektum által takarásban lévő része az élethűség és jó felhasználói élmény érdekében ne látszódjon, mint ahogy a tárgyak a valóságban is kitakarják egymást.

A raktározó robot és alsóraktár valósághű megjelenítése után a következő szükséges funkció a robotkar mechanikájának definiálása annak érdekében, hogy a fizikai térbe való integráláson túl a digitális ikerben szereplő robot mozgását is realisztikusan

kövesse. Ennek megfelelően implementálandó a három tengely összehangolt mozgása, a megfogószerv forgatása, valamint az elkészült termékek megfogása és elengedése.

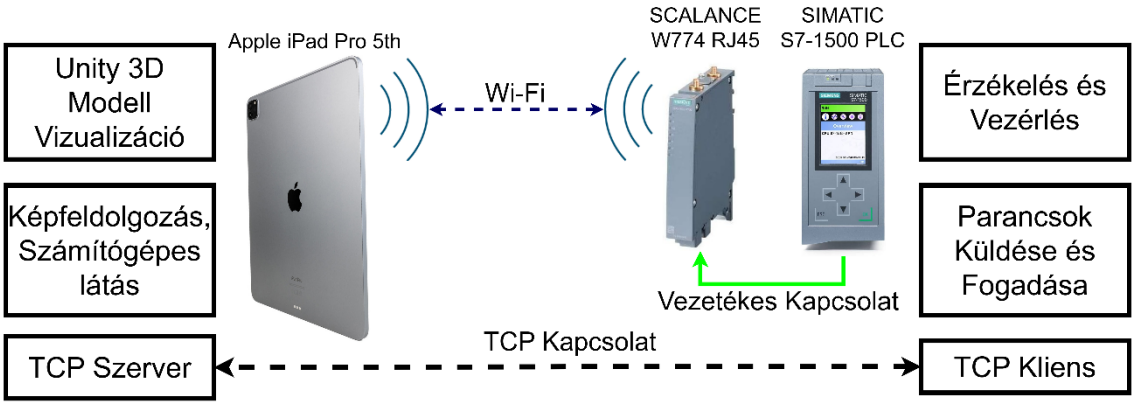
A mozgásfajták definiálásával a robot modellnek tetszőleges utasítássorozatot végre kell tudnia hajtani, ehhez azonban kommunikációs kapcsolatot kell létesítenie a gyártást irányító PLC-vel lehetőleg vezeték nélküli úton. Ezek az utasítások a következőek lehetnek:

- Egzakt pozícióba való mozgás 3D koordinátával.
- A megfogó elem elforgatása az alapálláshoz képest fokban.
- Termékminta megfogása, illetve elengedése.
- Termékminta érkezésének információja valamelyik szortírozó tálcán.

Az imént felsorolt funkciók implementálásával egy AR alkalmazás definíció szerint már megfelel az AR-támogatott digitális ikrek virtuális iker dimenziójának. Érdemes azonban figyelembe venni, milyen felhasználási eset segítené a kiterjesztett valóság applikációt a hibrid iker szintjére lépni. Ez a dimenzió koncepciójának főbb elemei ugyebár a kétirányú információáramlás, a kiberfizikai interakció és a digitális iker alapú vezérlés, vagyis a virtuális analízis alapú visszacsatolás, mely funkcióknak egyik nem titkolt célja a humán-robot kollaboráció (HRC), illetve együttműködés megteremtése. A helyszíni operátorok biztonsága egy ilyen munkakörnyezetben kiemelt fontosságú, emiatt az általam választott implementálandó funkció a virtuális robotkar működésének leállítása emberijelenlét érzékelése esetén, vagyis az AR applikáció legyen képes detektálni az emberi testet, meghatározni annak pozícióját, illetve visszajelezni a PLC-nek, amennyiben túl közel találja a raktárrobot helyzetéhez.

A megvalósítandó felhasználási esetek tisztázását követően, azokat figyelembe véve választottam eszközt az AR applikációmnak. A gyártósor hardveres felszereltségéből adódóan már adott egy SIMATIC S7-1500 CPU 1515TF-2 PN PLC, egy SCALANCE W774 RJ45 Wi-Fi router, az AR applikáció céleszközének hardverspecifikációi és fejlesztői lehetőségei miatt egy rendelkezésemre álló 5. generációs Apple iPad Pro táblagépet választottam. A tablet a Wi-Fi router-en keresztül tud hozzákapcsolódni a PLC alhálózatához, melyen belül már szabadon tudnak adatot cserélni, kommunikációs protokolljuknak pedig a TCP (Transmission Control Protocol) protokollt választottam az egyes utasítások biztos és helyes sorrendben való megérkezésének érdekében.

Az imént tárgyalt működés blokkvázlata látható a 15. ábrán.



15. **ábra:** A dolgozat által bemutatott saját kiterjesztett valóság alkalmazás felhasználási esetének architektúrája, működő tervezete.

3.2 Tervezés környezete, fejlesztői eszközök

Ebben a fejezetben kerül részletezésre, hogy a feladat megvalósításához milyen eszközök, illetve szoftverek lettek választva a felhasználási esetek, illetve a technológia által szabott követelmények értelmében.

3.2.1 Apple iPad Pro 12,9” (5th gen)

Az AR alkalmazásom céleszközének egy 5. generációs iPad Pro [14], magas színvonalú táblagépet választottam, amit az Apple gyárt. A készülék rendelkezik egy élénk Liquid Retina kijelzővel, amely két méretben elérhető: 11 és 12,9 hüvelykes. Az iPad Pro az Apple saját M1 processzorával van ellátva, ami gyors és hatékony teljesítményt biztosít munkához, játékhoz, illetve fejlesztői tevékenységekhez. Tárhelykapacitása 128 GB-tól 1 TB-ig terjed, így nagy mennyiségű adat, valamint program tárolására alkalmas. Az iPad Pro hátlapi kamerarendszere egy 12 megapixeles széles látószögű, és egy 10 megapixeles ultraszéles látószögű objektívvel rendelkezik, melyek lehetővé teszik a magas minőségű fényképezést és a 4K felbontású videófelvételt. Ezen felül a készülék LiDAR (Light Detection And Ranging) [15] érzékelővel van felszerelve, amely javítja a kiterjesztett valóság élményét, pontos tárgy elhelyezést és méréseket tesz lehetővé. A kapcsolódási lehetőségek közé tartozik a Wi-Fi, a mobilinternet, a Bluetooth és az USB-C (Universal Serial Bus type-C) gyors

adatátvitelhez és töltéshez. Az iPad Pro kompatibilis a második generációs Apple Pencil-lel és a Magic Keyboard-dal, ami fizikai billentyűzetet és touchpad-et biztosít a hatékonyabb produktivitáshoz. Az Apple iPad Pro táblagép a 16. ábrán figyelhető meg.

16. **ábra:** Az Apple iPad Pro 5th generáció. [16]

3.2.1.1 Alkalmazásfejlesztés iPadOS környezetre (Xcode)

Az iPadOS 2019 óta az Apple iPad táblagépek dedikált operációs rendszere. Az iPhone mobiltelefonok által használt iOS-re alapul, mely a hasonló megjelenési stílusából is megfigyelhető, fejlesztései a nagyobb kijelző és az erősebb hardver előnyeinek kihasználására fektetik a hangsúlyt. Tartalmazz továbbá Apple Pencil, Magic Keyboard, egér, illetve trackpad támogatást is a kényelmesebb, interaktívabb, kreatívabb felhasználói élmény lehetőségének biztosításához.

Az Apple termékeire való alkalmazásfejlesztést az Xcode [17] integrált fejlesztői környezet (IDE) biztosítja kizárólag macOS platformra, mely teljes körű eszközkészletet nyújt az iOS, iPadOS, macOS, watchOS és tvOS operációs rendszerekre való

szoftverfejlesztéshez. Kódszerkesztője szintaxiselemzéssel, kódnavigációval és -kiegészítéssel támogatja a hatékony fejlesztési folyamatot. A legtöbb népszerű magasszintű nyelv mellett kiemelt támogatást kap a Swift [18], mely az iOS és macOS alapú alkalmazások natív, magasszintű programozási nyelve. Az IDE további kulcsfontosságú eleme az Interface Builder, mely egy grafikus szerkesztő modern és felhasználóbarát UI (User Interface) egyszerű létrehozására, illetve elemeinek összekapcsolására a kóddal. A fejlesztés alatt álló alkalmazások vizsgálatára, tesztelésére az Xcode lehetőséget biztosít a célhardveren történő valós idejű hibakeresésére, azonban az eszköz jelenlétének hiányában az IDE beépített emulátort tartalmaz, hogy applikációnkat szimulált Apple termékek segítségével virtuális iOS, macOS és egyéb operációs rendszerek környezetében próbálhassuk ki, melyet a 17. ábra is szemléltet. A környezet hibakeresője támogat töréspontokat, lépésenkénti programvégrehajtást, illetve a változók és a memória futásidejű vizsgálatát.

17. ábra: Az Xcode fejlesztői környezet kódszerkesztő és szimulációs felülete. [19]

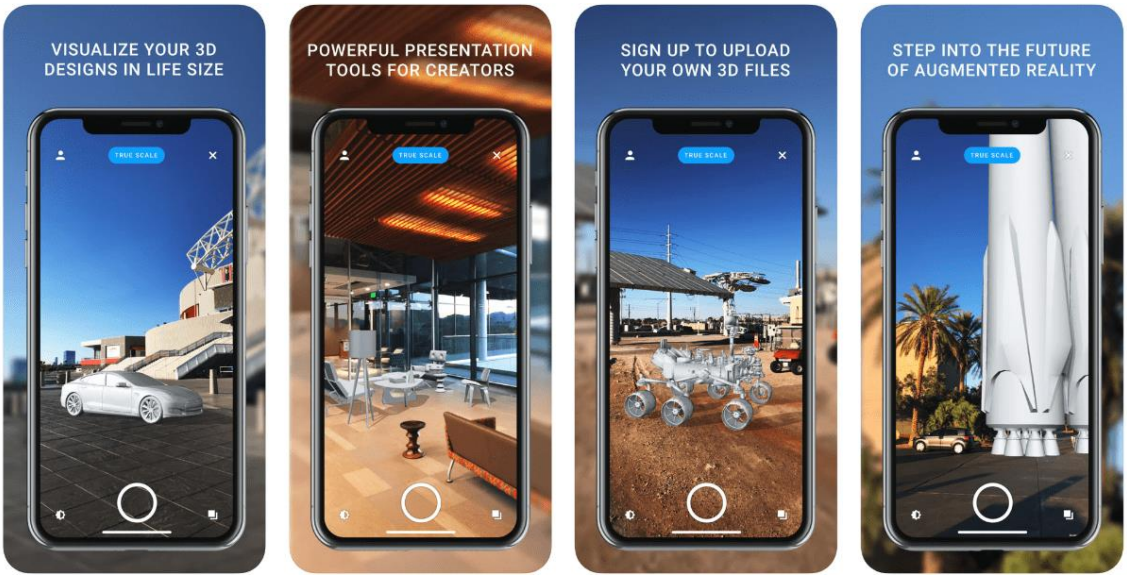
Az Xcode részét képzi továbbá az Instruments nevű eszköz, mely lehetőséget biztosít az alkalmazások teljesítményének elemzésére és optimalizálására. Részletes információkat nyújt a CPU (Central Processing Unit) használatról, a memóriafelhasználásról és az energiahatékonyságról. Az Instruments segítségével a fejlesztők azonosíthatják a teljesítményproblémákat és optimalizálhatják a kódot annak érdekében, hogy hatékonyabb és reaktívabb alkalmazásokat hozzanak létre.

Az Apple éves előfizetéses üzleti modellel tartja számon a termékeire fejlesztő developer-eket az Apple Developer Program keretein belül, akiknek egyéb plusz szolgáltatások mellett az Xcode köz közvetlen megosztási lehetőséget kínál az elkészült alkalmazások számára az App Store-ba. Az Apple termékekre való szoftverfejlesztés egyébként ingyen is kivitelezhető kisebb korlátozásokkal, például egy eszközre maximum csak 3 saját alkalmazás telepíthető, azok is csak néhány napig elérhetőek tesztelés céljából.

3.2.1.2 AR alkalmazásfejlesztés Apple eszközökre (ARKit)

Az ARKit [20] az Apple hivatalos kiterjesztett valóság szoftverfejlesztési keretrendszere kifejezetten iOS és iPadOS környezetre szánt alkalmazásokhoz. Általa teljes hozzáférésünk nyílik az iPhone-unk, illetve iPad-ünk kameráihoz, szenzoraihoz és képfeldogozó erőforrásaihoz, így biztosítva lehetőséget a fejlesztőknek, hogy látványos és interaktív AR applikációt készíthessenek.

Az ARKit a gépi látás fejlett algoritmusait és képfeldolgozási technológiáit használja a környezet lehető legjobb detektálásához, mint például a VIO (Visual Inertial Odometry) [21], a SLAM (Simultaneous Localization And Mapping) [22], képjellemzők felismerése és követése, mélységkép előállítása, de számos előre tanított mély neurális háló is rendelkezésére áll komplexebb objektumok detektálására, valamint mozgásuk figyelemmel kísérésére. Az Apple eszközök minőségi szenzorait és erős számítási kapacitását kihasználva az ARKit olyan bonyolult képfeldolgozási feladatok valós idejű megvalósítására nyújt megoldást, mint az eszköz térbeli pozíciójának megállapítása, síkfelületek felismerése, képek, tárgyak, emberek detektálása és mozgásuk követése, fényviszonyok megállapítása fényforrások pozícionálásával, virtuális vagy valós objektumok egymás által adott perspektívából való takarása, vagy a LiDAR (Light Detection And Ranging) szenzorral felszerelt eszközök esetében a környezet térhálójának poligonokból való rekreálása a mélységkép alapján. Az ARKit felhasználási lehetőségeit illusztrálja a 18. ábra.



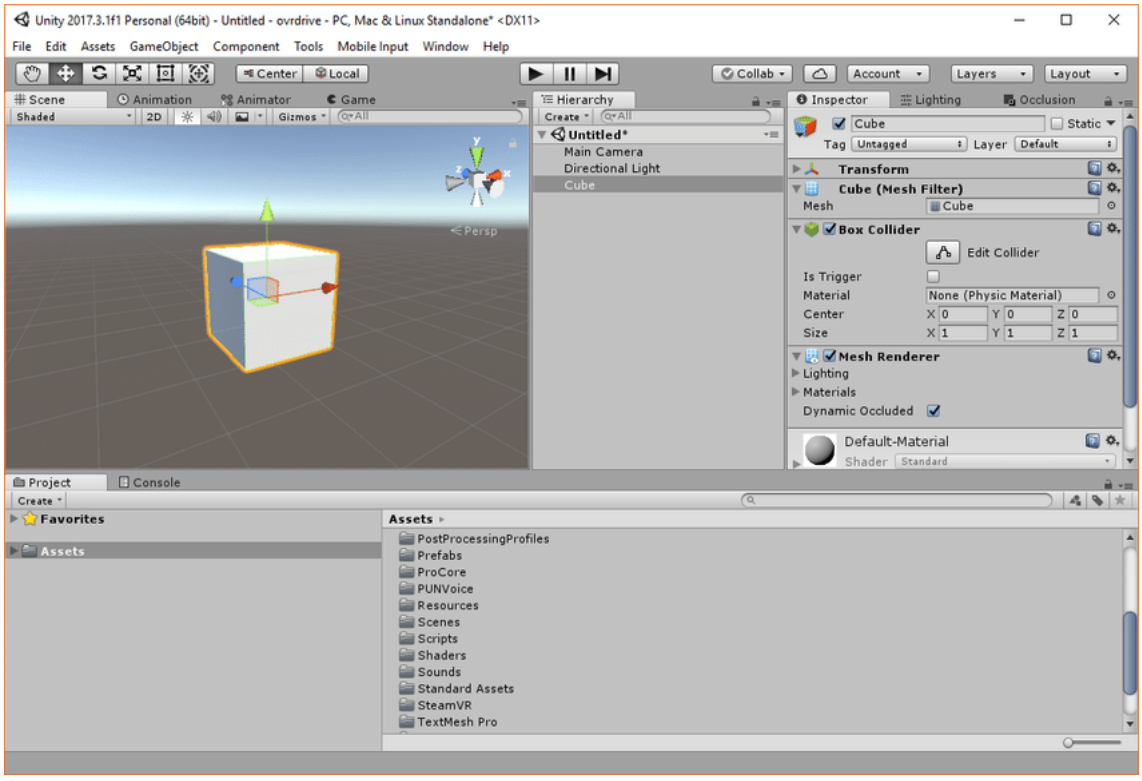
18. ábra: Az ARKit által nyújtott lehetőségek szemléltetése. [23]

Az ARKit tehát rendkívül sok képfeldolgozási technológiát kínál, melyek sok esetben alapfeltételei a robusztus kiterjesztett valóság alkalmazás fejlesztésének, hiszen a virtuális információk, illetve modell elemek pontos illesztéséhez, valóságba való integrálásához, majd az azokkal való interakcióhoz elengedhetetlen a környezet sokoldalú vizsgálata. Az általa készíthető applikációk ennek köszönhetően látványos és interaktív kiterjesztett valóság élményt nyújthatnak a játékiparban, oktatásban, tervezésben, navigációban, a szociális médiában és egyéb vizualizációs alkalmazásokban.

3.2.2 Unity (Real-Time Development Platform)

Az Apple iPad Pro-ra való AR szoftverfejlesztéshez Unity-ra [24] esett a választásom, mely egy sokoldalú, ingyenes, cross-platform (többplatformú, platformfüggetlen) játékfejlesztő motor és integrált fejlesztői környezet. Multifunkcionalitásának köszönhetően lehetőséget kínál különböző interaktív alkalmazások létrehozására 2D-ben vagy 3D-ben beleértve játékokat, szimulációkat, virtuális valóság élményeket, kiterjesztett valóság applikációkat és egyéb vizuális tartalmakat. Átfogó eszközkészlet és széles funkcióbázis áll rendelkezésre, hogy változatos és látványos szoftvereket készítsünk Windows, Linux, Mac, iOS/iPadOS, Android, játékkonzol (Xbox, PlayStation Nintendo Switch), VR/AR szemüvegek, okos TV-k, böngészők és más környezetek platformjára.

A Unity egyik kulcsfontosságú jellemzője a vizuális szerkesztője, amely lehetővé teszi a fejlesztőknek virtuális objektumok, jelenetek és felhasználói felületek tervezését, manipulálását. A beépített fizikai motor lehetőségek széles tárházát biztosítja a valósághű mechanikai folyamatok szimulálásához, objektumainkat felparaméterezhetjük tömeggel, gravitációs gyorsulással, súrlódással vagy akár szilárdsággal, melyekhez erővektorokat, elmozdulás időfüggvényeket és objektuminterakciókat definiálhatunk. Ezeknek az eszközöknek a segítségével animálhatunk jeleneteket, melyeket megvilágítással, hanghatásokkal tehetünk gazdagabb élménnyé. A grafikus szerkesztői felület kinézete a 19. ábrán figyelhető meg.



19. ábra: A Unity játékmotor grafikus szerkesztője. [25]

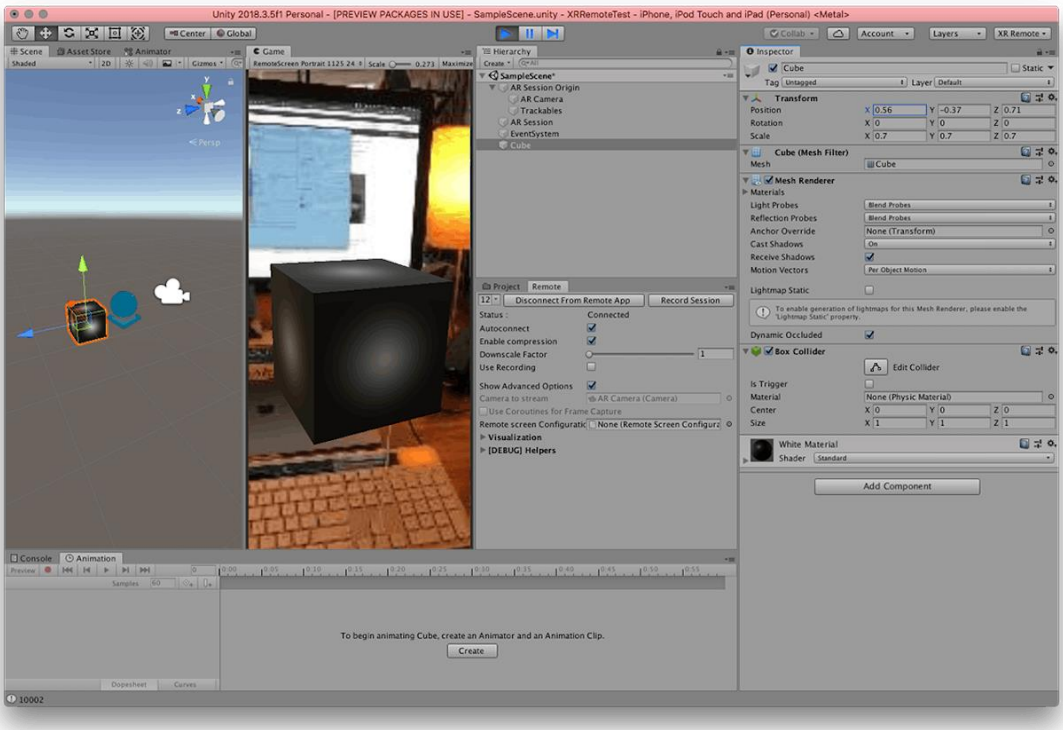
Az intuitív grafikus felület mögött a Unity egy magasszintű szkriptelési lehetőséggel rendelkezik. Elsődlegesen támogatott programnyelve a C# annak legtöbb nyelvi elemével és könyvtárával együtt. Beépített szövegszerkesztő helyett kódjainkat Visual Studio-ban írhatjuk kihasználva annak a legtöbb fejlesztést segítő funkcióját. Szkriptjeinket vizuális objektumokhoz rendelhetjük, melyekhez így egyedi viselkedést, logikát, illetve bonyolult játékmechanikákat társíthatunk.

A fejlesztők együttműködésének támogatására a Unity az Asset Store szolgáltatását kínálja, mely egy nyílt piacteret biztosít 3D modellek, textúrák, audió anyagok, szkriptek megosztására, illetve igénybevételére. Ezek a tartalmak könnyen integrálhatóak saját projektjeinkbe, mely sok időt és erőfeszítést takaríthat meg a fejlesztési folyamatban.

3.2.3 AR alkalmazásfejlesztés Unity környezetben (AR Foundation)

A Unity AR Foundation [26] egy fejlett keretrendszer, amely széles eszköztárral rendelkezik kiterjesztett valóság applikációk fejlesztésérem mobil, és más dedikált eszközökre. Egységes API-t biztosít az iOS és Android felületekre egyaránt, így kompatibilis az ARKit és ARCore framework-ökkel, ezzel biztosítva lehetőséget az AR alkalmazások platformfüggetlen fejlesztésére.

Az AR Foundation számos képfeldolgozási és megjelenítési funkciót kínál, mely hozzájárul egy látványos és interaktív kiterjesztett valóság élmény megalkotásához. Az eszköz környezetének részletes és gyors vizsgálata által az alkalmazásunk képes lehet valós időben felismerni és interakcióba lépni síkfelületekkel, detektálni és követni képeket vagy objektumokat, realizálni és reprodukálni fényviszonyokat. Ezek a lehetőségek nagymértékben javítják a virtuális tartalmak integrációját a valós környezetbe. Az AR Foundation használatának példája a 20. ábrán figyelhető meg.



20. ábra: Az AR Foundation használata Unity környezetben. [27]

A kiterjesztett valóság applikációk nagyobb élményt nyújtanak intuitív interakciós lehetőségek biztosításával, ezért az AR Foundation számos lehetőséget kínál a céleszköz hardveres képességeitől függően az alkalmazással való kapcsolatteremtésre. Az érintőképernyőkön elhelyezett UI (felhasználói felület) elemeken keresztül, vagy akár a targetálható objektumokkal közvetlenül is lehetőség van interakciós opciók implementálására. A felhasználó nagyobb mértékű bevonását eredményezik azonban az olyan népszerű funkciók, mint a gesztusalapú irányítás, mely lehetőséget kínál például a virtuális objektumok kijelölésére, mozgatására, forgatására, méretezésére. Ezt többnyire kezünkkel végezhetjük, de a fejlett arcfelismerő és elemző szoftverek segítségével akár szemmozgásunk, vagy arckifejezéseink is bemenetéül szolgálhatnak az applikáció működésének. Az interaktív élmény további bővítését szolgálja a hangalapú vezérlés, mely szintén implementálható eszköz.

3.2.4 Siemens SIMATIC S7-1500 PLC (Programable Logical Controller)

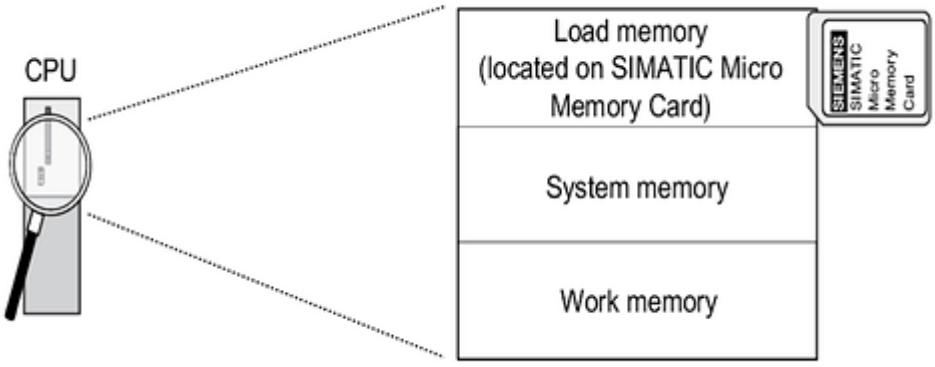
Az iPad Pro-ra készülő AR applikációnak kapcsolatot kell tudnia létesíteni a gyártósort vezérlő PLC-vel, melyhez annak programkódját módosítani kell a kiterjesztett valóság alkalmazás kiszolgálására, ennek okán fontos tisztázni, hogy mi az a PLC, és milyen programozási lehetőségei vannak.

A PLC egy programozható logikai vezérlő, mely funkcióját tekintve egy robusztus irányítástechnikai számítógép. Elsődleges célja az ipari automatizálási és vezérlőrendszerek kontrollálása, mint például gyártósorok, erőművek vagy egyéb gépi irányítást igénylő üzemek, így konstrukcióját, architektúráját, működését, programnyelveit és kommunikációját erre optimalizálva tervezték. A PLC-k digitális és analóg jeleken, reléken, illetve hajtásvezérlőkön keresztül kapcsolódnak ipari irányítástechnikai folyamatokhoz.

A PLC-k moduláris szerkezetűek, központi egységük a CPU (Central Processing Unit) egység. Kiépítésük szerint két fő típus között teszünk különbséget: centrális és decentrális. A centrális PLC-k esetében a modulok egy hátlapi buszon keresztül kommunikálnak a CPU-val, míg a decentrális PLC-k esetében valamilyen ipari buszrendszeren keresztül (például PROFIBUS, PROFINET) csatlakoznak hozzá. A 21. ábrán egy CPU egység figyelhető meg centrális elrendezésben.

21. ábra: A SIMATIC S7-1500 PLC termékcsalád egy eleme centrális modulokkal. [28]

A PLC-kben 3 memóriaterületet különböztetünk meg: Load Memory (betöltési memória), System Memory (rendszermemória) és Work Memory (munkamemória). A betöltési memória egy cserélhető memóriakártya (SIMATIC Memory Card, SMC), mely ezáltal bővíthető is. Ide tölthetőek le a PLC hardverkonfigurációját meghatározó beállítások, illetve a vezérlési feladatot megfogalmazó program-, valamint adatblokkok. A rendszermemória egy CPU-ba integrált memóriaterület, mely a PLC operációs rendszerén kívül címezhető regisztereket, merker-eket, időzítőket, számlálókat tartalmaz, továbbá dedikált helyet biztosít a PLC által rögzített ki- és bemeneti értékek tárolására a Process Image Input (PII) és Process Image Output (PIO) területeken. A munkamemória szintén integrált, nem bővíthető memória, mely kód-, és adatterületekre osztható. A betöltési memóriából a használat előtt ide töltődnek be a programblokkok, illetve a hardverkonfiguráció. A PLC-k memóriáit szemlélteti a 22. ábra.

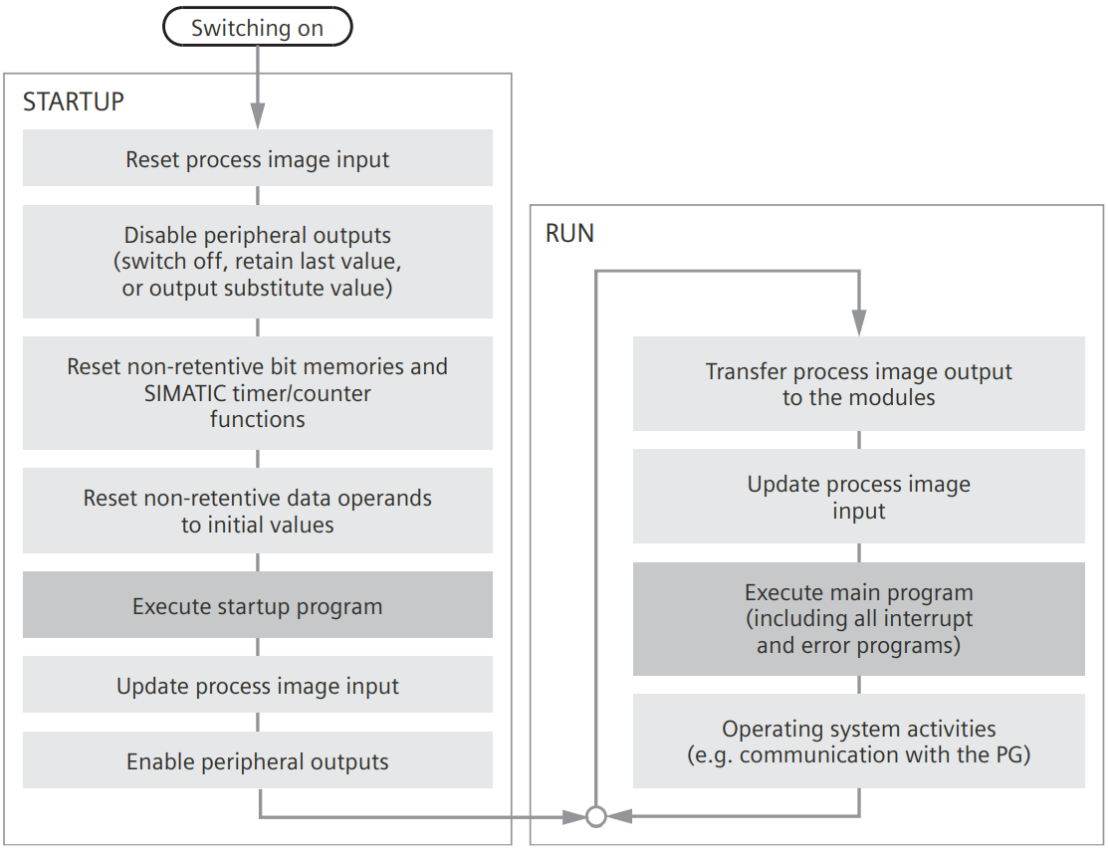


22. ábra: A CPU egységek memóriakezelése. [29]

A PLC-k három fő üzemállapota a STOP (állás), a RUN (futás) és a STOP-ból RUN-ba való váltás esetén a STARTUP (felfutás), előfordulhat azonban FAULT (hiba), vagyis defektív mód, amennyiben a PLC nem lekezelhető utasítást próbál végrehajtani.

A STARTUP során a PLC meghatározott sorrendben fontos feladatokat végez el a megfelelő RUN üzemállapot előkészítésének érdekében. Ezek a feladatok a Process Image Input alaphelyzetbe állítása, a kimeneti perifériák tiltása, a nem remanens memóriaterületek visszaállítása, az OB100 (Startup Organization Block) organizációs programblokk futtatása (amennyiben projektálásra került), a Process Image Input

frissítése, majd végül kimeneti perifériák engedélyezése. A PLC STARTUP, valamint RUN üzemállapotát a 23. ábra szemlélteti.

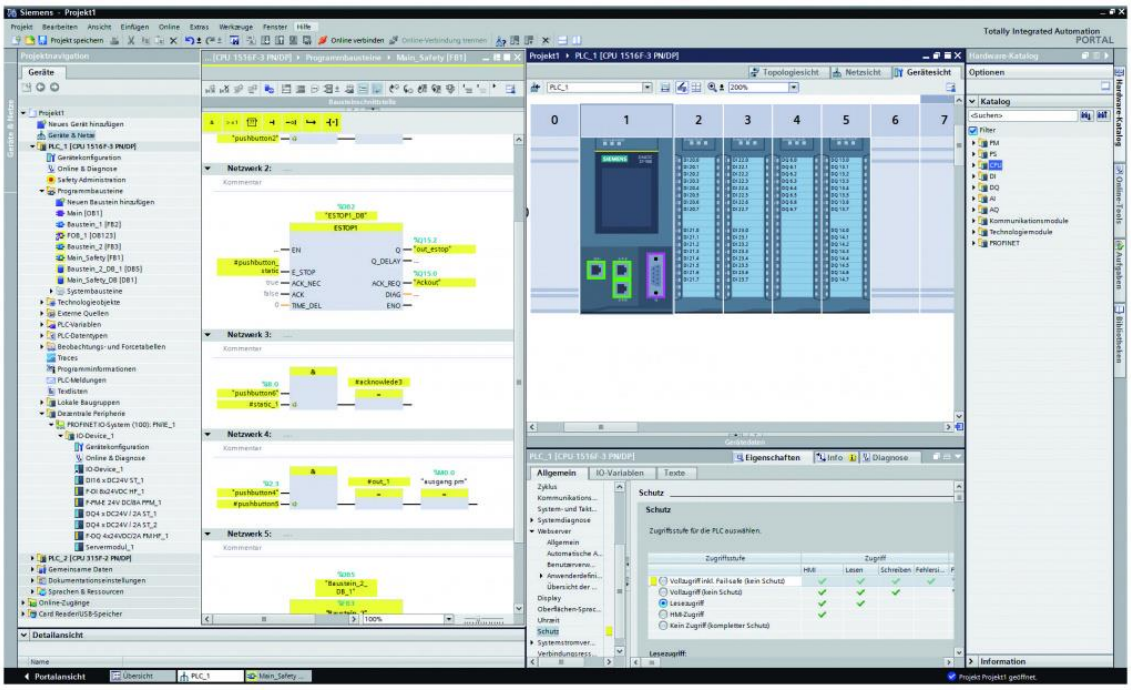


23. ábra: A PLC STARTUP procedúrájának lépései és RUN üzemállapotának ciklusa. [30]

A felfutás lépéseinek sikeres végrehajtását követően a PLC belép a RUN üzemállapotba, amely ciklikus működésű, vagyis programkódjának szekvenciális végrehajtása periodikusan ismétlődik. Minden PLC ciklus a Watchdog-gal (felügyelettel) kezdődik, amely figyeli a ciklusidőt. Ha a ciklusidő meghaladja a hardverkonfigurációban meghatározott maximális értéket, meghívásra kerül egy bizonyos Time Error Interrupt megszakítás organizációs blokk, mely tartalmazza az eset szükséges kezelésének procedúráját, amennyiben projektálva van, ellenkező esetben, vagy, ha a futás a megengedett ciklusidő kétszeresét is meghaladja, a PLC STOP üzemállapotba kerül, melyben a programfeldolgozás nem kerül végrehajtásra. A PLC STARTUP és RUN üzemállapotok folyamatának szemléltetése a 23. ábrán látható.

3.2.5 Siemens PLC programozás (TIA Portal)

A TIA Portal (Totally Integrated Automation Portal) [31] a Siemens által nyújtott tervezői felület, illetve fejlesztői környezet automatizálási és vezérlési rendszerek tervezéséhez és konfigurálásához. A szoftvercsomag számos különböző komponenst magába integrálva szolgál egységes platformként a tervezett irányítórendszerben részt vevő hardverelemek, mint például PLC-k, HMI (Human-Machine Interface) megjelenítők, hajtásvezérlők egy felületen történő felparaméterezéséhez, programozásához, diagnosztizálásához, így szüntetve meg a több szoftvereszközzel való munkavégzés szükségét. Ezek közül a főbb szoftvercsomagok a STEP 7 (Steuerungen Einfach Programmieren, „Vezérlők Egyszerű Programozása”), a WinCC (Windows Control Center), SINAMICS Startdrive (hajtásvezérlés), de számos más komponens is a részét képzi.



24. ábra: A TIA Portal grafikus kódszerkesztői és konfigurációs felülete. [32]

A TIA Portal a 24. ábrán megfigyelhető felhasználóbarát és intuitív grafikus felhasználói felületet, valamint hasznos fejlesztői eszközök széles tárházát biztosítja a PLC programozás kényelmes és átlátható kivitelezéséhez. Az egyes tervezői folyamatoknak külön projekteket hozhatunk létre, melyekben egymástól függetlenül dolgozhatunk, fejleszthetünk.

A szoftver hardverkatalógusa széles portfólióját tartalmazza a Siemens termékek listájának, így megtalálható benne a legtöbb PLC, decentrális periféria és modul típus. A megfelelő eszközöket projektálva, illetve felparaméterezve előállítható az irányítórendszer hardverkonfigurációja, melyet a PLC felprogramozásával lehet vezérelni.

A PLC vezérlését meghatározó kód elkészítésére a STEP 7 rendszerben program- és adatblokkok létrehozásával végezhető el. Ezek organizációs blokkok (OB), funkcióblokkok (FB), funkciók/függvények (FC) vagy adatblokkok (DB) lehetnek. A ciklikus programfeldolgozásért az OB1 (Program Cycle) organizációs blokk felel, vagyis tartalma periodikusan lefutásra kerül PLC ciklusonként. Az FB-k, illetve FC-k tetszőleges bemeneti, kimeneti és temporális változókkal felparaméterezhető programblokkok azzal a különbséggel, hogy az FB-k rendelkezhetnek statikus változókkal is, melyek értéke nem áll vissza a kiindulási értékre a PLC ciklus elején, értékét külön DB tárolja. Praktikusan mindkét blokktípus az ismétlődő kódrészletek redundanciájának csökkentésére, valamint a kód strukturált, átlátható szerkezetének szervezésére használatos. A TIA Portal a leggyakrabban alkalmazott, egyszerűbb, vagy akár kifejezetten bonyolult szoftverkomponensekre részletes rendszerblokk (SFB, SFC) könyvtárat tart fent, mely a fejlesztési folyamat egészén a programozó rendelkezésére állnak. A DB-k tetszőleges típusú, a programkód számára globálisan elérhető, statikus változókat tárolnak. A blokkokban példányosítható alapvető változótípusokon kívül létrehozhatóak struktúrák is, illetve a UDT-k (User-Defined Type), vagyis a felhasználó által készített egyedi, összetett változótípusok.

A különböző programblokkokat a PLC kód megfogalmazására számos alkalmas programozási nyelven van lehetőségünk megírni, mint például a LAD (Ladder logic, létra/relé logika), az FBD (funkcióblokk diagram), az SCL (Structured Control Language) vagy az STL (Statement List, utasításlista).

A TIA Portal a PLC belső memóriaterületének, illetve digitális és analóg be- és kimeneti moduljainak címzésére Tag-táblát tart fent, melyet a programkódból szintén globálisan elérhetünk. A változóink, valamint memóriacímeink strukturált monitorozására Watch-, illetve Force-táblákat készíthetünk, melyek nagy átláthatóságot képesek biztosítani a fejlesztett program felügyeletére, tesztelésére.

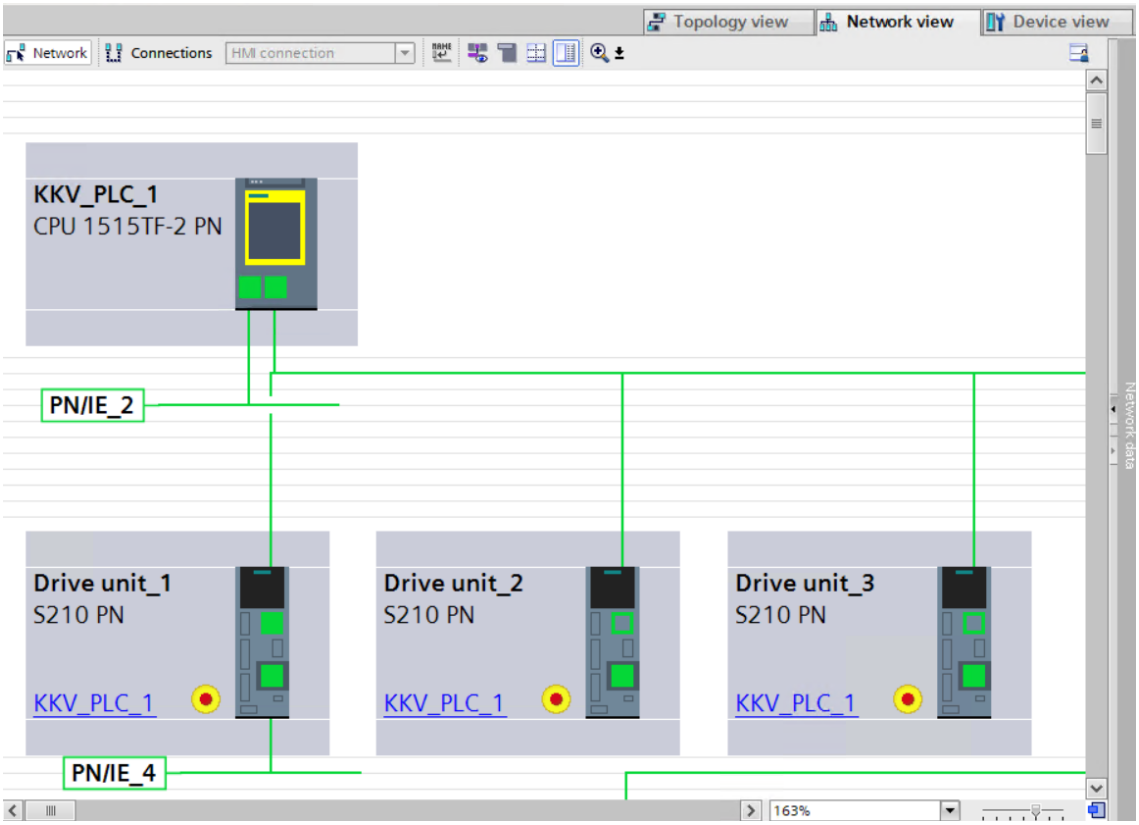
Az elkészült hardverkonfiguráció, illetve programkód hálózati úton keresztül tölthető le a PLC Load memóriájába, mely a RUN üzemállapotban kerül végrehajtásra.

4 Megvalósítás

Ebben a fejezetben kerül bemutatásra az AR-támogatott digitális iker szolgáltatást megvalósító kiterjesztett valóság alkalmazás szoftverkomponenseinek megvalósítása.

4.1 PLC program megtervezése

A gyártósort vezérlő SIMATIC S7-1500 CPU 1515TF-2 PN PLC már rendelkezik a virtuális raktárrobot automatizált működését megvalósító szoftverrel, mellyel annak három tengelyének mozgását működtető SINAMICS S210 PN hajtásvezérlő decentrális perifériákat és szervomotorokat irányítja. Ez a TIA Portal projektben is megfigyelhető a 25. ábrán.



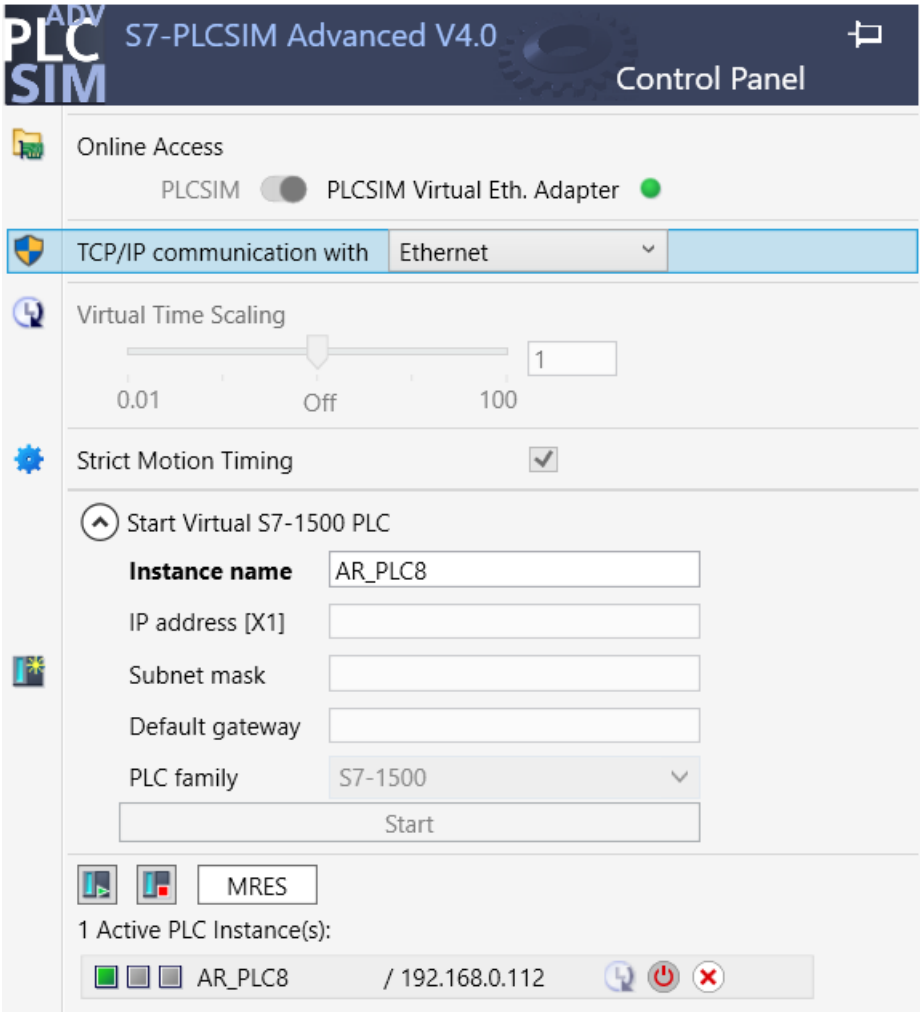
25. ábra: A gyártósor modell TIA Portal projektjének raktárrobotot vezérlő hardverkomponensei.

A terveimnek megfelelően az implementációs feladatom, hogy ennek a PLC-nek a vezérlőszoftverét kiegészítsem az AR applikációm kiszolgálására, vagyis TIA Portal

projektjét kibővítsem olyan programblokkokkal, amelyek a raktárrobotnak címzett utasítások kiadásával egyidőben továbbítsa az iPad Pro-n futó kiterjesztett valóság alkalmazásnak. Ehhez meg kell keresnem azokat a változókat, melyek valós idejűen tárolják a parancsok típusát, paraméterének értékét, és eseményvezérelt kiadásának jelét. A parancsok tartalmát jól olvasható formátumú üzenetté kell formálni a PLC-n, majd egy TCP kliens létesítésével hálózati úton továbbítani kell a táblagépre tervezett TCP szerver felé, melyet követően annak feldolgozása már ott történhet. Ehhez hasonló módon a PLC programnak képesnek kell lennie az AR alkalmazás válasz üzenetének fogadására, vagyis, amennyiben a kiterjesztett valóság alkalmazás emberi jelenlétet érzékel a virtuális raktármodell közelében, és értesíti a PLC-t, annak le kell állítania a vezérlését.

Az imént tisztázott kifejlesztendő funkciókat a TIA Portal-ban SCL programozási nyelven implementálom. A PLC szoftverének elkészítését jelentős mértékben támogatta a 26. ábrán látható SIMATIC S7-PLCSIM Advanced V4.0 program, mely valós időben képes a fejlesztői számítógép hálózati interfészén egy S7-1500 PLC-t szimulálni, mely által otthoni környezetben is képes voltam elkészíteni és tesztelni a funkciók megvalósítására szolgáló kódot, így a helyszínen már csak importálni és illeszteni kellett a valós PLC programjába.

Ez a fejlesztési módszer nagy átfedésben van a korábban tárgyalt virtuális beüzemelés koncepciójával.



26. **ábra:** A SIMATIC S7-PLCSIM Advanced V4.0 program egy S7-1500 PLC szimulálása közben.

4.1.1 Parancs üzenetek előállítása JSON objektumok formájában

A PLC által kiadott parancs üzenetek kiadásának detektálásához, illetve tartalmuk megállapításához érdemes megvizsgálni, milyen típusú instrukciókat kell keresni. A korábban már tárgyaltaknak megfelelően a virtuális raktárrobot négy különböző információt vár a PLC-től, melyek a 3D koordinátába való mozgás, a megfogószerv (gripper) adott szögbe való forgatása, az elkészült termékek megfogása és elengedése, végül maguknak az elkészült termékek a gyártósorról a megfelelő futószalagon való legördülésének információja.

Az egyes elkészült termékek színüknek megfelelően gördülnek le a fizikai gyártósor végét jelentő csúszdákon, melyek mindegyike egy lézeres szenzorral vannak

ellátva. Ezek logikai értékének billenését vizsgálva könnyűszerrel megállapítható a programból, melyik futószalagokon kell virtuális terméket raktározni. A mozgás, forgatás és megfogás utasítások kiadása egymáshoz nagyon hasonló, eseményvezérelt módon lettek implementálva, mindegyiket egy-egy bool változó felfutóéle vezérli, melyek jól detektálható események, billenésük pillanatában könnyen kinyerhetőek a valós idejű parancsba adott paraméterek értéke. A kigyűjtött vezérlési információk tárolására egy tartalmuknak megfelelő struktúrát hozok létre, mely magában foglalja a végrehajtandó utasítás típusát, és az ahhoz kapcsolódó paramétert. Azonban ezeknek a parancsoknak a kiadása adott esetben nagyon gyorsan, pillanatszerűen is követhetik egymást, melyet a TCP kommunikáció nem feltétlenül képes egyidejűleg lekezelni azok elvesztése nélkül, emiatt a detektált parancsok struktúráját egy FIFO (First In, First Out) bufferben gyűjtöm, a vagyis az újonnan érkezőek a tömb végére kerülnek, küldeni pedig a tömb elejéről kell sorban, melynek 10 Hz-es frekvenciát, vagyis 100 ms küldési periódusidőt választottam, melyet már képes lekezelni a TCP kommunikáció, viszont még mindig jó felhasználói élményt ad. Ennek a logikának a kezelésére egy funkcióblokkot hoztam létre, mely feltölti az érzékelt parancsokkal és azok értékeivel a 27. ábrán is látható FIFO buffert, így szolgáltatva azokat a kommunikáció számára.

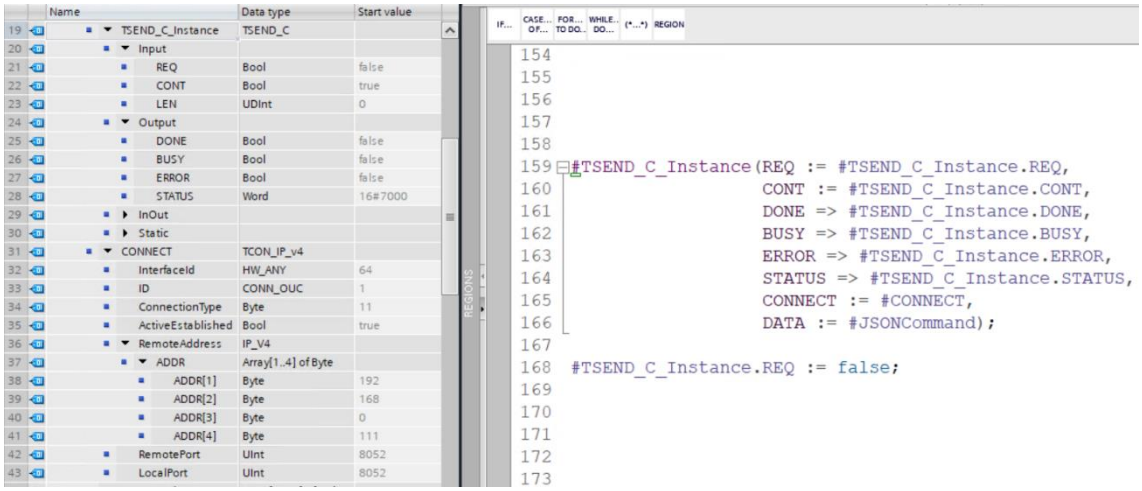
Name	Data type	Default value
CommandBuffer	Array[0..#BUFFERLENGTH] of Struct	
CommandBuffer[0]	Struct	
CommandSelect	Struct	
Move	Bool	false
Rotate	Bool	false
Grip	Bool	false
Spawn	Bool	false
CommandAttributeValues	Struct	
X	Real	0.0
Y	Real	0.0
Z	Real	0.0
Degree	Int	0
Grip	Bool	false
ConveyorID	Int	0
CommandBuffer[1]	Struct	
CommandBuffer[2]	Struct	
CommandBuffer[3]	Struct	
CommandBuffer[4]	Struct	
CommandBuffer[5]	Struct	

27. ábra: A kiadandó parancsokat tároló FIFO buffer a TIA Portal projektben.

4.1.2 TCP kliens parametrizálása, utasítások kiadása

Az iPad Pro-val létesített hálózati kapcsolat megteremtéséhez a PLC oldaláról egy TCP klienst kell létre hozni, illetve azt felparametrizálni, melynek egy külön funkcióblokkot hoztam létre. A blokk célja, hogy kapcsolódási kérelmet indítson a kiterjesztett valóság alkalmazás TCP szervere felé, majd a felépült kapcsolaton keresztül az előzetesen megfelelő formátumban előállított parancs üzeneteket elküldje a szervernek.

A TCP klienst megvalósító kódrészlet a TIA Portal egy beépített könyvtárának funkcióblokkja, a TSEND_C blokk, mely felparaméterezését és meghívását követően kapcsolatot létesít a neki megadott szerverrel, kérésre pedig elküldi a szintén paraméterül kapott változót egy bitstream formájában. A blokk bekonfigurálásához egy CONNECT hálózati struktúrát kell paraméterül kapnia, melyben be kell állítani a kapcsolat típusát TCP-nek, betöltött szerepét pedig kliensnek, továbbá meg kell adni a címzendő kommunikációs fél IP címét, valamint a kapcsolat hálózati portját. Ennek a parametrizálása és meghívása látható a 28. ábrán.



28. ábra: A TSEND_C programblokk TCP kliensként való konfigurációja a TIA Portal-ban.

A TCP kapcsolat létesítését követően a PLC-n futó programnak a parancsokat gyűjtő FIFO buffer első eleméből egy értelmezhető formátumú string-et kell előállítania, melyet küldésre lehet szánni. Ennek a formátumnak a JSON (JavaScript Object Notation) objektumok szintaktikáját választottam, mivel ez az informatikában egy széles körben alkalmazott, emberi szemmel is könnyen értelmezhető adatleíró szerkezet, továbbá a

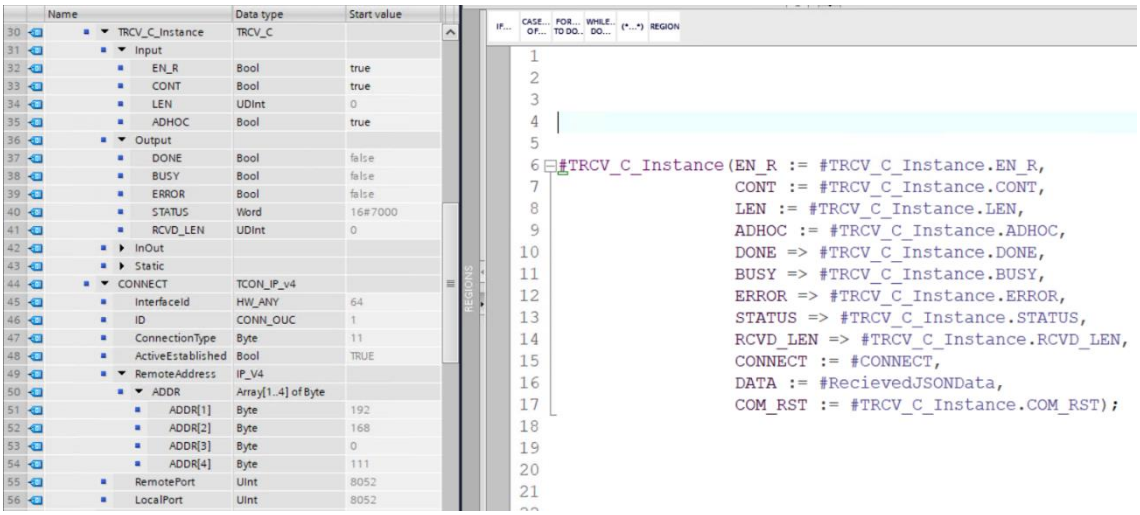
legtöbb programnyelven egyszerű string-kezeléssel deszerializálható, az általa tárolt információ könnyen kinyerhetők lokális változókba, valamint nem mellesleg az általam küldeni szándékozott utasítások is megfeleltethetőek a JSON objektumokban használt kulcs-érték párjainak. Ennek megfelelően ennek a funkcióblokknak a feladata, hogy a kiadni tervezett parancsok tartalmát string-gé konvertálja, majd a megfelelő szintaktikai elemekkel együtt konkatenálja JSON objektummá, majd átadja azt a TSEND_C blokknak. Ez a működés reprezentálódik a 29. ábrán.

5		Send	Bool	false	FALSE
6		CommandTypeSelect	String	''	'move'
7	▶	CommandSelect	Struct		
8		JSONCommand	String	''	'{"type":"move","x":0.100,"y":0.200,"z":0.300}'
9	▶	ParsedCommandString	Array[0..15] of String		
10	▶	SampleParsedComma...	Array[0..15] of String		
11	▼	CommandAttributeVa...	Struct		
12		X	Real	0.1	0.1
13		Y	Real	0.1	0.2
14		Z	Real	0.1	0.3
15		Degree	Int	90	90
16		Grip	Bool	false	FALSE
17		ConveyorID	Int	0	0
18	▶	TSEND_C_Instance	TSEND_C		
19	▶	CONNECT	TCON_IP_v4		

29. ábra: A parancs üzenetek előállítása JSON objektumok formájában a TIA Portal-ban.

4.1.3 Emberi jelenlét információjának fogadása az AR alkalmazástól

A PLC programját az imént tárgyalt feladatokon kívül továbbá el kell látni a virtuális raktárrobotot vészleállító funkcióval, mely aktiválására az iPad Pro-n futó AR applikációtól várja az utasítást. Ehhez implementálni kell egy fordított irányú kommunikáció lehetőségét, vagyis egy TCP üzeneteket fogadó blokkot, mely nem más, mint a TRCV_C, szintén a TIA Portal beépített könyvtárainak blokkja. Ennek felparaméterezéséhez egy új funkcióblokkot hozok létre, melyeben a TSEND_C blokkhoz nagyon hasonló módon konfigurálom be, ami a 30. ábrán megfigyelhető.



30. ábra: A TRCV_C programblokk TCP kliensként való konfigurációja a TIA Portal-ban.

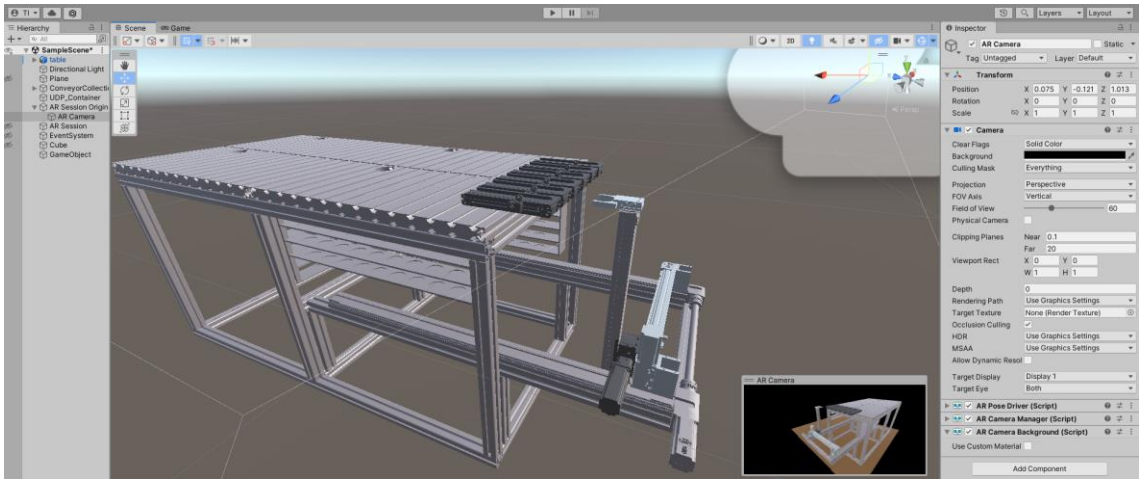
A PLC program a fogadó blokk által a már korábban kifejtett formátumban, JSON objektumokban formájában várja a virtuális raktárrobot működtetésének engedélyezését, a beérkező üzenetekben tehát JSON objektumokat keresek az emberi test közeli pozíciójának detektálására utaló tartalommal. A letiltás, illetve engedélyezés üzenetének beérkezésekor egy már létező bool változó kapcsolgatásával állítja a fogadó blokk a raktárrobot üzemállapotát, melyet a helyszíni operátorok egy HMI panelen már eddig is állíthattak manuálisan, ezt követően azonban már automatizáltam is működhet.

A PLC programját kiegészítő funkcióblokkok elkészültük után meghívhatóak a PLC Main OB-jában (Program Cycle , OB1), majd letölthető a PLC-re, melytől fogva az eddigi kódot kiegészítve funkcionál, így fel lett készítve a kiterjesztett valóság alkalmazás támogatására.

4.2 Kiterjesztett valóság alkalmazás megtervezése Unity környezetben

A gyártósort, illetve a virtuális raktárrobotot vezérlő PLC programja fel lett készítve az AR alkalmazással való közös együttműködésre, így a soron következő feladat ennek az applikációnak a Unity környezetben való lefejlesztése. Ehhez a Unity AR Foundation kiterjesztett valóság szoftverfejlesztési csomagjának könyvtárait veszem segítségül, amely többek között az Apple által kiadott ARKit SDK-t (Software

Development Kit) is tartalmazza, ezáltal képes az AR alkalmazások forráskódjának Swift nyelvű Xcode projektjének előállítására, melyek az Xcode fejlesztői környezetből telepíthetőek és futtathatóak az iOS és iPadOS operációs rendszerű Apple eszközökön, így a saját applikációm céleszközéül szánt iPad Pro-n is.



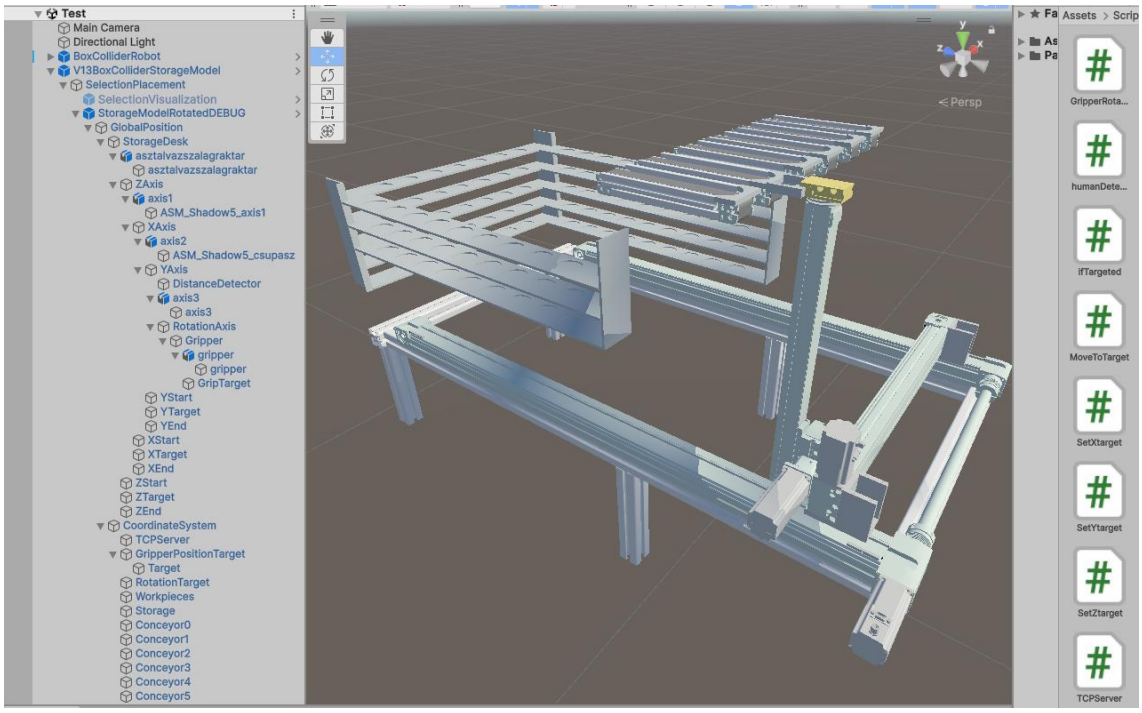
31. ábra: A virtuális raktárrobot 3D modellje egy kezdetleges fejlesztési szakaszban Unity környezetben.

A megvalósítandó kiterjesztett valóság program feladata tehát a gyártósor digitális ikre által jelenleg csak a virtuális térben létező raktárrobot realisztikus és méretarányos 3D modelljének élethű megjelenítése a valós térbe integrálva, illetve annak a fizikai infrastruktúrához való illesztése. A raktárrobotnak egy korai fejlesztési fázisban álló modellje figyelhető meg a 31. ábrán. A megjelenített robot modellnek továbbá tükröznie kell a digitális ikerbeli megfelelőjének mozgását is, ezért függetlenül mozgó elemeinek mechanikáját mozgató C# szkriptekkel való felprogramozással kell definiálni. Mindezek mellett a robotnak tudnia kell fogadni a PLC által kiadott, annak TCP kliensén keresztül továbbított utasításokat, emiatt kell rendelkeznie egy TCP szerverrel, mely a helyes konfigurációval tovább tudja adni a PLC-től érkezett parancsokat a mozgó elemeknek. Végül az alkalmazásnak képesnek kell lennie az emberi test detektálására, valamint pozíciójának meghatározására a 3D-s térben, melyet amennyiben túl közelinek ítél a raktárrobothoz, értesítenie kell a PLC-t, hogy állítsa le a vezérlést.

4.2.1 Az AR raktárrobot modelljének importálása, mechanikájának definiálása az AR alkalmazásban

4.2.1.1 Az AR raktárrobot modelljének importálása Unity környezetbe

A virtuális raktárrobot realisztikus és méretarányos 3D modelljéhez a Siemens NX CAD/CAM/CAE tervezőszoftverben elkészített digitális ikerből tudok hozzáférni, vagyis annak külön mozgó elemenként való kiexportálása által .obj szabványosított formátumba, melyet a Unity is képes kezelni. A megjelenítésének első lépése tehát az elemeknek a Unity projektbe való importálása, melyek az NX kedvező exportálási mechanizmusának köszönhetően egyből a megfelelő orientációban, egymáshoz illeszkedve jelennek meg. Ezek az elemek a raktárrobot vázszerkezete, melyhez a polcok és a gyártósor végén lévő futószalagok is hozzá tartoznak, a három külön mozgó tengely, valamint a megfogószerv (gripper). Ezeknek a komponenseknek az importálását követően meg kell határoznunk azok szerkezeti hierarchiáját, hiszen amennyiben azt szeretnénk, hogy a gripper, valamint az alsóbb rendű tengelyek kövessék felettes, szülőobjektumuk (parent object) mozgását, hierarchiában alájuk kell tartozniuk. A raktárrobot 3D modelljének hierarchiája a 32. ábra bal oldalán tekinthető meg.



32. ábra: A végleges virtuális raktárrobot modell hierarchiája és szkriptjei.

4.2.1.2 Az AR raktárrobot mechanikájának definiálása

Az importált és együtt mozgó elemeinek figyelembevételével hierarchikusan felépített raktárrobot modellje készen áll a vezérelhető mozgási mechanikáinak definiálására. Ezek a mozgási mechanikák tehát a korábban már tisztázott parancsba adható utasítások, vagyis a gripper 3D koordinátába való mozgatása a tengelyek által, a gripper adott szögbe való forgatása, vagy a közrefogott termék megfogása, valamint elengedése, továbbá ne feledkezzünk meg azonban az újonnan elkészült termékek megjelenítéséről sem a futószalagok végén, melyek az előzőleg említett parancsok végrehajtásának valamilyen variációja által kerülhetnek el a végső helyükre az alsóraktár polcain. Annak érdekében, hogy a feladatok külső utasításba adva legyenek végrehajthatóak, ezeket a mechanizmusokat bizonyos bemeneti változókkal irányítható, általános értékű, passzívan követendő szabályokként definiálom, melyeket a robot feltétel nélkül követ, és melyeknek megfelelően a bemeneti változókat aszinkron módon figyelembe véve változtat állapotain, mint a pozíció, vagy forgásirány, így a beérkező utasításoknak csak ezt a néhány inputként szolgáló változót kell módosítania a robot vezérléséhez.

Elsőként a tengelyek mozgásának mechanizmusát foglalom kódba az imént letisztázottaknak megfelelően. Ehhez érdemes néhány szóban megállapítani, hogy mégis milyen mozgást várunk ezektől az objektumoktól. Ahogy a működésüket a valóságban is elvárnánk, az egyes tengelyeknek csak saját sínjeik mentén szabad elmozdulniuk, vagyis szabadsági fokaik mindegyik tengely számára a 3D-s térnek csak egyik bázisvektorának irányára korlátozódnak. Ezt alapul véve a mozgatásuk bemenetéül egy 3D-s pontot érdemes definiálni és a térben elhelyezni, melynek az egyes tengelyek a hozzájuk rendelt iránynak megfelelő koordinátáját fogják célul kitűzni, és afelé haladni, ez egy olyan hatást kelt majd az egyébként egymástól függetlenül a pont megfelelő koordinátája felé tartó tengelyeknek, mintha a megfogószerv közel egyvonalban haladni a cél irányába.

Fontos szempont azonban a realisztikusság és az élvezhető felhasználói élmény érdekében, hogy a tengelyek valamilyen dinamikát követve mozogjanak, valamint egyszerre érkezzenek meg. Ezt a két faktort egyszerre elégíti ki a Unity 3D-s pontjainak Vector3.Lerp() lineáris interpoláción alapuló metódusa, melyet az objektumokhoz rendelt szkriptek Update() függvényében meghívva, valamint egy célpontot paraméterül adva, mely felé dinamikusan megválasztott sebességgel halad egyre lassabban annak függvényében, hogy milyen közel van a célhoz. Sebességét továbbá úgy választja meg,

hogy akármilyen kiinduló pozícióból ugyanannyi idő alatt tegye meg a céltól való távolságot. Ezek a tulajdonságok egy kellemes dinamikájú mozgást kölcsönöznek, mely komfortosabb látvány az egyenes vonalú egyenletes mozgásnál, valamint általa mindhárom tengely valóban egyidőben érkezik meg a céljául kitűzött pontba. Ezt követően a mozgató utasításoknak csak az imént tárgyalt működésű inputként szolgáló pont pozícióját kell ugrásszerűen módosítania, vagyis áthelyeznie, melyet a robotkar kellemes látványú dinamikával követ.

A megfogószerv forgatásának vezérlését a tengelyek utasítható mozgásának már definiált elve alapján implementálom, vagyis létrehozok egy a forgatás bemenetéül szolgáló referencia pontot, melynek orientációját egy a gripper-hez rendelt szkriptben paraméterül adok meg a szkript Update() metódusában meghívott Quaternion.RotateTowards() függvények, ezáltal a gripper egyenletes szögsebességgel fog forogni a referenciát adó bemeneti pont iránya felé, amíg el nem éri azt. Ezzel a legbonyolultabb mozgatási mechanizmusok implementálásra kerültek.

A következő utasításul kapható és végrehajtandó feladat az elkészült termékek gripper általi megfogása, illetve azok elengedése. A valóságban ugyebár ez a megfogószerv szárainak összeszorulása által kifejtett tapadási súrlódási együtthatóval kényszeríti a terméket a közös mozgásra, a virtuális térben ennek a túldefiniálása szükségtelen. Nem ad semmivel rosszabb felhasználói élményt a kiválasztott, vagyis a gripper által közrefogott termékeknek a hierarchiában megfogószerv alá való átrendezése, vagyis a termék szülő objektumának a gripper-t beállítva, melytől fogva az mindenhova követni fogja a szülő mozgásának megfelelően. Ebből kiindulva az elengedés mechanikája is megfeleltethető egy hierarchiai átrendezésnek, mely folyamán a termék szülőobjektumát a gripper-ről egy másik, teljesen attól független objektumot választok, ami alá a helyére került termékeket gyűjtöm. Jogosan merül fel azonban a kérdés, hogy hogy lehet eldönteni futásidőben, hogy melyik terméket fogtuk éppen közre a gripper-rel. Ezt a termékek, valamint a gripper szárai köré helyezett úgynevezett boksz collider-ekkel (ütköző dobozokkal), vagyis ütközésdetektáló láthatatlan téglatestekkel vizsgálom, tehát amelyik termékkel közös metszetet alkot a megfogószerv szárai köré tervezett box collider, megfogás, valamit elengedés utasítás esetén azzal a termékkel kell interakcióba lépnem. Ez egy kódból futásidőben is jól detektálható esemény.

Az utolsó kívülről érkező lehetséges utasítás az az egyes termékek elkészülésének, valamint a színüknek megfelelő csúszdán való legördülésüknek

információja, melyről miután a raktárrobot tudomásul szerzett, meg kell jelenítenie azt szintén a termék színének megfelelő futószalagon. Ezt az adott színű termékek egyszerű, futásidejű példányosításával, valamint a futószalagok végeinek előre ismert pozíciójába való helyezésével könnyen kivitelezhető feladat.

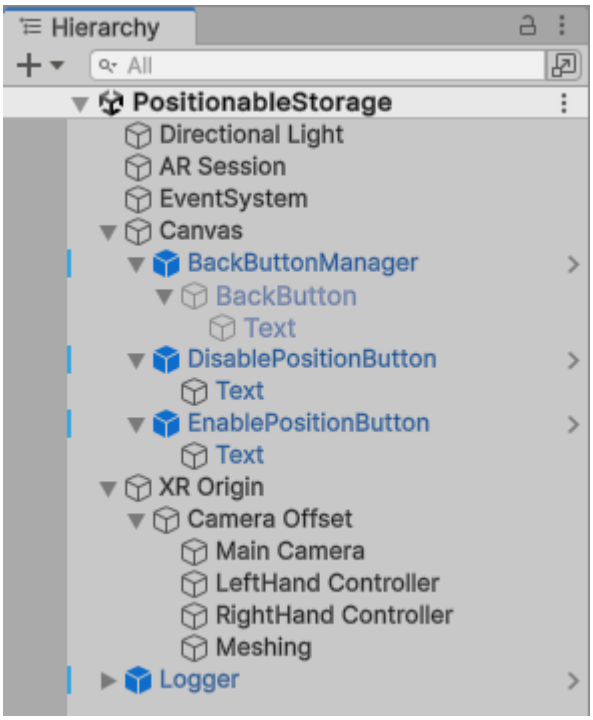
Ezzel a kiterjesztett valóság alkalmazásban megjelenő raktárrobotot felkészítettem minden lehetséges utasítás beérkezésének esetére, így képes lesz azokat végrehajtani. Érdemes rátérni az alkalmazás kezelési módjaira és megjelenésére.

4.2.1.3 Az AR alkalmazással való interakció implementálása

A robot kiterjesztett valóság applikációban való megjelenítésére a számos lehetőség adott. Az egyszerű, egy pontban automatikusan megjelenő és lebegő modellektől az olyan népszerű jelölőalapú (marker-based) megoldásokig sok lehetőséget kipróbálva vizsgáltam az általuk nyújtott felhasználói élményt. Érdekes módon a legélvezhetőbb megjelenítési végeredményt a síkdetektálás, majd az arra manuálisan, felhasználói interakció által való lehelyezése és illesztése adta. Úgy néz ki, a Unity által kínált lokáció alapú kiterjesztett valóság megoldások nem a legrobusztusabbak, legalábbis nem az AR Foundation által támogatott eljárásokkal, hiszen a beépített jelölőalapú modellmegjelenítő funkció csak kifejezetten közelről működött, akkor is a gyakori újrapozícionálás miatt egy remegő hatást keltve. Emiatt döntöttem úgy, hogy a kellemes látvány érdekében a manuális elhelyezését választom a modellnek, ezzel legalább jól reprezentálódnak az AR alkalmazásokkal való felhasználói interakciós lehetőségek.

Az imént letisztázott működés megvalósítására létre is hozok a Unity projektben egy új, üres jelenetet, vagy színhelyet (scene), így hívja a Unity az alkalmazáson belül aktuálisan megjelenített vizuális környezetet, mely jelen esetben a hátlapi főkamera képére alapszik. Hozzáadok egy XR Origin objektumot, mely már hierarchiailag magában foglal egy Camera Offset, az alatt pedig egy Main Camera objektumot. Ez az XR Origin objektum hivatott a jelenet indításakor elhelyezni az AR alkalmazásban értelmezett 3D-s tér koordinátarendszerének globális origóját az applikációt futtató eszköz által használt kamera indítás pillanatabeli pozíciójába, melyet minden további globális pozíció kiszámításához referenciául vesz. Mivel a kiterjesztett valóság alkalmazást sosem lehetünk képesek az eszközt mindig ugyanabban a pozícióban tartva

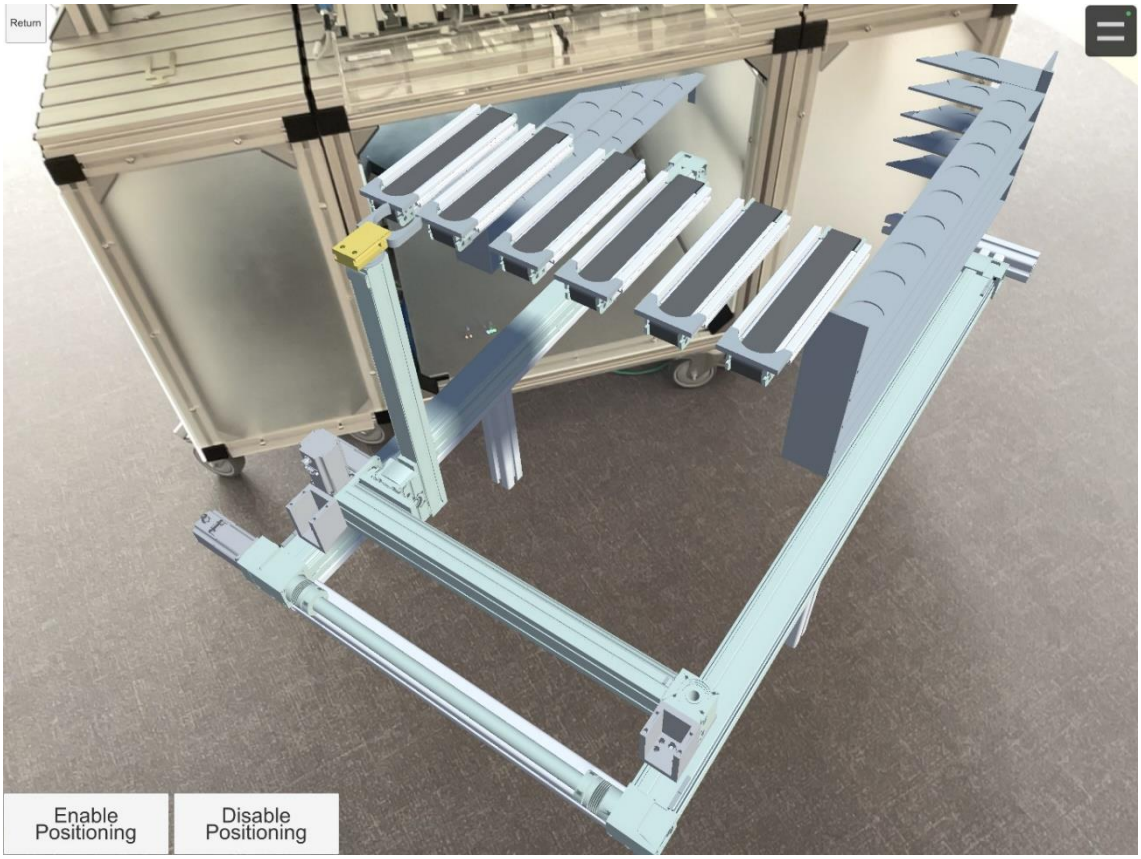
elindítani, ezért az egyes mozgatási mechanizmusoknál érdemes az objektumok globális pozíciója helyett a lokálisat figyelembe venni, mely esetben a szülő objektumhoz képesti pozíciót és orientációt veszi alapul a program. Ezt tudomásul véve adok hozzá a jelenethez egy úgynevezett AR Session objektumot, melynek feladata a virtuális objektumok pozíciójának és orientációjának nyomon követése a programot futtató eszköz elmozdulásának függvényében, mely által a AR modellek a felhasználó mozgásától függetlenül egyhelyben maradnak, enélkül ezek követnék a felhasználó mozgását. Ezzel előkészítettem a programot a virtuális raktárrobot megjelenítésére, és egyhelyben maradására. A PLC által vezérelt AR raktárrobot kiterjesztett valóság alkalmazásának jelenete a 33. ábrán figyelhető meg.



33. ábra: A PLC által vezérelt AR raktárrobot modell kiterjesztett valóság alkalmazásának jelenete.

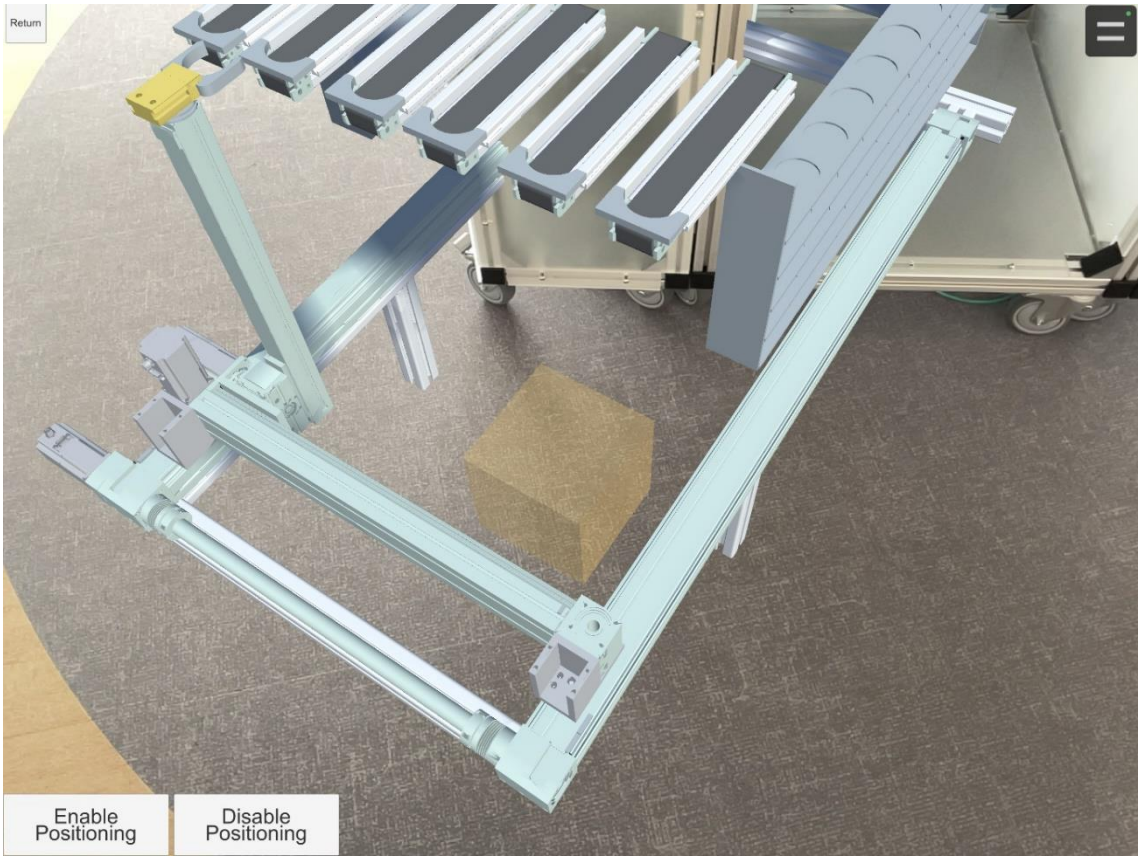
A raktárrobot vizualizálására egy korábban már tárgyalt okból a síkdetektálást, valamint a felhasználó által az azon való szabad mozgathatóság, és a fizikai infrastruktúrához való manuális illesztés lehetőségét választottam. Ehhez a következő AR Foundation-beli szkripteket és objektumokat használtam fel. Először az XR Origin objektumot töltöttem fel az AR Plane Manager, az AR Input Manager, az AR Raycast Manager, majd végül az AR Foundation által szolgáltatott mintaalkalmazásokban található Place On Plane szkripteket, mely utóbbinak paraméterül adtam a raktárrobot

modelljének objektumát. Ehhez a robot modellt is ki kellett egészítenem egy AR Translation Interactable szkripttel, mellyel engedélyeztem rajta a horizontális elmozgatás szándékából történő interakció lehetőségét. Ezt követően a Main Camera objektumhoz hozzáadtam az AR Gesture Interactor scriptet, továbbá az AR Session objektumot is elláttam az XR Interaction Manager szkripttel. Ezek együttes működése azt eredményezi, hogy a kiterjesztett valóság alkalmazás a jelenet indulásakor a kamera betekintési szögébe eső képen elkezd síkdetektálást végezni, majd az úgynevezett Raycasting technológia segítségével a táblagép érintőkijelzőjének minden pontját levetíti, majd hozzárendeli a detektált síkfelületek azon pontjához, mellyel a képernyőn az egybe esik. Ennek, valamint az egyéb interakciós szkripteknek köszönhetően valósul meg az a működés, hogy a felhasználó a hol hozzáér a képernyőn, a síknak annak megfelelő pontján fog megjelenni a raktárrobot, vagyis ujjunkkal húzva, vagy másik ponthoz érve tetszőlegesen helyezhetjük el modellünket a síkfelületeken. Ezt a működést hivatott reprezentálni a 34. ábra.



34. ábra: A valóságba integrált, pozícionálható virtuális raktárrobot modell a fizikai infrastruktúrához való illesztés előtt.

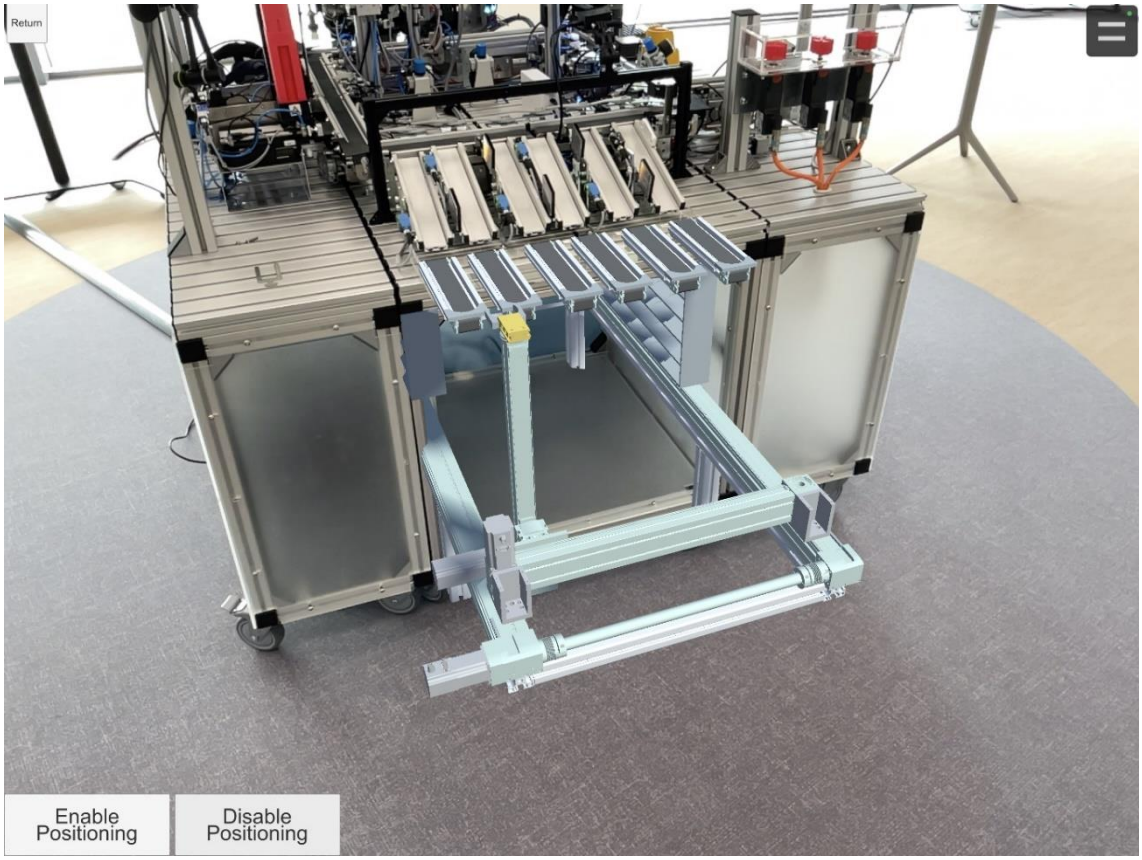
A virtuális raktárrobot így már elhelyezhető a valós térbe integrálva, azonban orientációja még az XR Origin által meghatározott globális referenciapont irányától függ, vagyis attól, hogy merre néztünk a jelenet indításakor. Erre egy kijelölésen, majd kétujjas forgatáson alapuló megoldást alkalmaztam, mely által tetszőlegesen forgatható a modell. Ehhez mindösszesen a raktárrobot modellhez kellett hozzáadnom egy AR Selection Interactable, valamint egy AR Rotation Interactable szkriptet, valamint egy Selection Visualization objektumot, melynek egy kis, transzparens kockát választottam, ami a 35. ábrán látható. Ezeknek köszönhetően a robot modell érintőképernyőn való megérintésével kijelölhető, melyet a kis, transzparens kockának a megjelenése jelez, ezt követően pedig kétujjas technikával elforgatható. A kijelölés eltüntethető a modellről való elkattintással.



35. ábra: A valóságba integrált, pozícionálható raktárrobot modell forgatás közben a fizikai infrastruktúrához való illesztés előtt.

A virtuális raktárrobot modell így már a detektált síkfelületeken megjeleníthető, horizontálisan elmozgatható, el is forgatható, egy tulajdonsága még zavaró lehet azonban, hogy a fizikai infrastruktúrához gondosan odaillesztett modellt könnyű véletlenül

elmozgatni például a képernyő szélének megérintésével, ezért létrehoztam a mozgatás tiltására, valamint engedélyezésére szolgáló gombokat, melyeket az érintőkijelző bal alsó sarkában helyeztem el. Ezek alkalmazásával megelőzhetőek a kellemetlen félrepozícionálások. A helyesen bepozícionált AR raktárrobot modell a 36. ábrán látható.

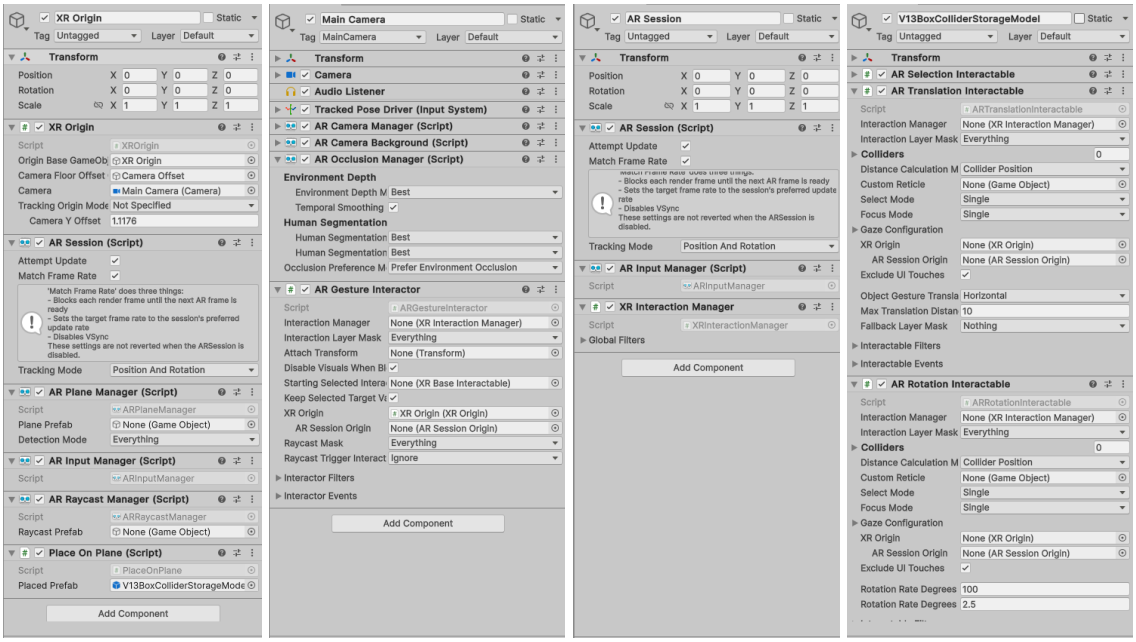


36. ábra: Az AR raktárrobot modell a fizikai infrastruktúrához való illesztést követően.

4.2.1.4 Kitakarás (occlusion)

Korábban a robot 3D modell valós térbe való realisztikus integrációjának egyik kritériumaként már szóba került az occlusion, vagyis a felhasználó szemszögéből a virtuális objektumok fizikaiak által kitakart részének eltűntetése. Ezt egy mélység szenzor nélküli, egyetlen kamerával rendelkező eszközzel nagyon nehezen lehetne kivitelezni, úgyhogy többek között ez a szempont is indokolta az iPad Pro céleszközül való választását, hiszen az LiDAR szenzora által képes mélységkép előállítására, melyet felhasználva el tudja dönteni, hogy a megjelenített virtuális objektumok látszódhatnak-e a felhasználó szemszögéből. Ehhez a mindösszesen a Main Camera objektumhoz hozzá kell adni az AR Occlusion Manager szkriptet, melynek konfigurálható minőséget befolyásoló paramétereinek beállítása után el is végzi ezt a feladatot.

Az előző fejezetek által tárgyalt, a jelenetben alkalmazott szoftverkomponensek konfigurációja a 37. ábrán figyelhető meg.



37. ábra: A PLC által vezérelt AR raktárrobot modell kiterjesztett valóság alkalmazásában a szoftverkomponensek konfigurációja.

4.2.2 TCP szerver beágyazása, FIFO utasításfogadás és végrehajtás

Az előző fejezetben tárgyaltak alapján a virtuális raktárrobot realisztikus modellje élethű módon megjeleníthető a felhasználó környezetébe integrálva, majd manuális pozícionálás és forgatás interakciók által illeszthető a fizikai gyártósorhoz. Jelen állapotban már utasítások végrehajtására is alkalmas a mozgási mechanizmusainak definiálásának köszönhetően. A PLC által kiadott parancsok fogadására alkalmas TCP szerver, valamint az általa beérkezett feladatok feldolgozása és eseményvezérelt végrehajtása azonban még implementációra szorul.

Az eddig tárgyalt szoftverkomponensek alkalmazásánál nem tértem ki rá, hiszen viszonylag magától értetődő, hogy azok a program fő szálán (thread) szekvenciális utasításvégrehajtással futnak. Az olyan hálózati kommunikációs programrészek esetében, mint egy TCP szerver azonban szükséges a többszálasítás (multithreading) szoftvertechnológiáját alkalmazni, amennyiben nem szeretnénk, hogy a program megjelenítésére szolgáló fő szál esetleg egy kommunikációs folyamat végbemenésére

várva megszakítsa az érintőképernyőn kijelzett kép frissítését, a felhasználói interakciók regisztrálását, és az azokra való reagálását, illetve minden egyéb látható, vagy háttérben történő tevékenységét, melyek folyamatos futása fontos a hibamentes működés érdekében. A többszálasítás erre egy új szálon történő párhuzamos programfeldolgozással kínál megoldást, mely a fő száltól aszinkron módon képes végezni a számára delegált feladatokat, a szálak közötti szinkronizációra, vagy valós idejű információcserére azonban lehetőség van.

A többszálasítás imént tárgyalt előnyeinek kihasználása érdekében egy a raktárrobot modellhez rendelt új szkriptben létre is hozok egy másodlagos szálat a TCP szerver számára, melyhez a C# System.Threading könyvtárának osztályait és metódusait hívom segítségül. Az új szálat elindításra utasítom a szkript meghívásával, melyen belül kivételkezelést, valamint hiba, esetleg leállás esetén újraindítást definiálok. Ezen a kommunikációs szálon belül a C# System.Net.Sockets könyvtárának osztályaiból példányosítok, és elindítok egy TCP figyelő (TCP Listener) objektumot, mellyel a PLC által küldött üzenetek portján figyelem az esetlegesen beérkező üzeneteket. A fogadott üzenetek egy hosszú bitstream formájában érkeznek, melyek ASCII kódból string-gé konvertálva válnak olvashatóvá. Ezek adott esetben egyszerre több egymást követő, PLC által kiadott parancsot is tartalmazhatnak, ezért a begyűjtött üzeneteket változóban letárolom további vizsgálatokra. Ezzel sikeresen implementációra került a kiterjesztett valóság alkalmazás TCP szervere.

A beérkezett és letárolt bitstream-ekben a PLC által külött utasításokat JSON objektumok formájában keresem, tehát a kapcsos zárójelek ({ és }) között elhelyezkedő substring-eket. A detektált JSON objektumokból egy saját JSON deszerializáló kódrészlettel nyerem ki az általuk tartalmazott információkat, ezekhez a C# string-kezelő nyelvielemeit alkalmazom. Először trimmeléssel eltávolítom az elején és végén található kapcsos zárójeleket, majd kinyerem a vesszővel elválasztott kulcs-érték párokat, melyeket egy tömbbe töltök. Ezután megvizsgálom minden egyes kulcsot, valamint annak értékét külön, az értékeket a hozzá tartozó kulcs függvényében kasztolom az annak megfelelő formátumba. A JSON objektumokban keresett kulcsok a parancs típusa, az mozgatás X, Y, és Z koordinátái, az elforgatás szöge, a termék megfogása, illetve elengedése, valamint egy elkészült termék megjelenítése egy adott futószalagon. A JSON objektumokban talált kulcs-érték párokból kinyert információkat egy dedikáltan azok rendszerezett tárolására létrehozott osztály példányaiban mentem el, így sikerült

változóban tárolva hozzáférnem a PLC által kiadott parancsok tartalmához, ezzel lehetővé téve azok végrehajtását.

A PLC utasításait eddig tehát képes vagyok sikeresen fogadni, valamint az általa hordozott információkat is tudom értelmezni. Ezek a parancsok azonban egykorábban már tárgyalt okból adott esetben nagyon gyorsan követhetik egymást, így annak érdekében, hogy egy újonnan beérkezett parancs ne írja felül az aktuálisan végrehajtás alatt lévőt, itt is egy FIFO buffert alkalmazok az utasítások sorrendben való elvégzésére. Ennek megfelelően létrehozok egy listát (System.Collections.Generic.List<T>), melybe a kommunikációs szálban a JSON objektumokból kinyert érték letárolására szolgáló osztálypéldányok a lista végére kerülnek. Ezzel párhuzamosan a fő szálban a raktárrobot a feladatokat a lista elejéről kezdi feldolgozni, melyeket ciklikusan, eseményvezérelt módon, sorban hajt végre, továbbá új feladatba csak akkor kezd bele, ha az előzővel végzett, aminek ellenőrzése könnyen kivitelezhető. Ez egy robusztus utasításfogadási és feladatvégrehajtási módszer, azonban egy lekezelendő funkció még implementációra szorul. A négy különböző utasítástípusból háromnál kifejezetten fontos a sorrendben, egymás után való végrehajtás, az újonnan elkészült termékek megjelenítésének viszont a valóságban sem szükséges megvárniuk, hogy végbemenjen egy adott esetben hosszabb, feltorlódott utasítássorozat. Emiatt az új termék megjelenítésének parancsát egy külön FIFO bufferben listázom, mely párhuzamosan képes működni a többi utasítás listájával, így az elkészült termékek azonnal meg tudnak jelenni a színüknek megfelelő futószalagon.

Ezekkel a funkciókkal sikeresen implementálásra került a kiterjesztett valóság alkalmazás TCP szervere, illetve eseményvezérelt utasításvégrehajtása. Összefoglalva tehát a virtuális raktárrobot modell már megjeleníthető, illeszthető a fizikai gyártósorhoz, melynek vezérlő PLC-jével ezentúl képes TCP kapcsolaton keresztül fogadni annak utasításait, valamint sorrendben végre is hajtani azokat. Az applikáció továbbá egy eddig nem tárgyalt módon alkalmas a működés állapotait szöveges bejegyzésekkel, log üzenetekkel a felhasználó tudtára adni. Ennek a stádiumnak és funkcióbázisnak az elérésével kijelenthető, hogy a kiterjesztett valóság alkalmazás definíció szerint elérte az AR-támogatott digitális ikrek dimenzióinak virtuális ikerpár szintjét.

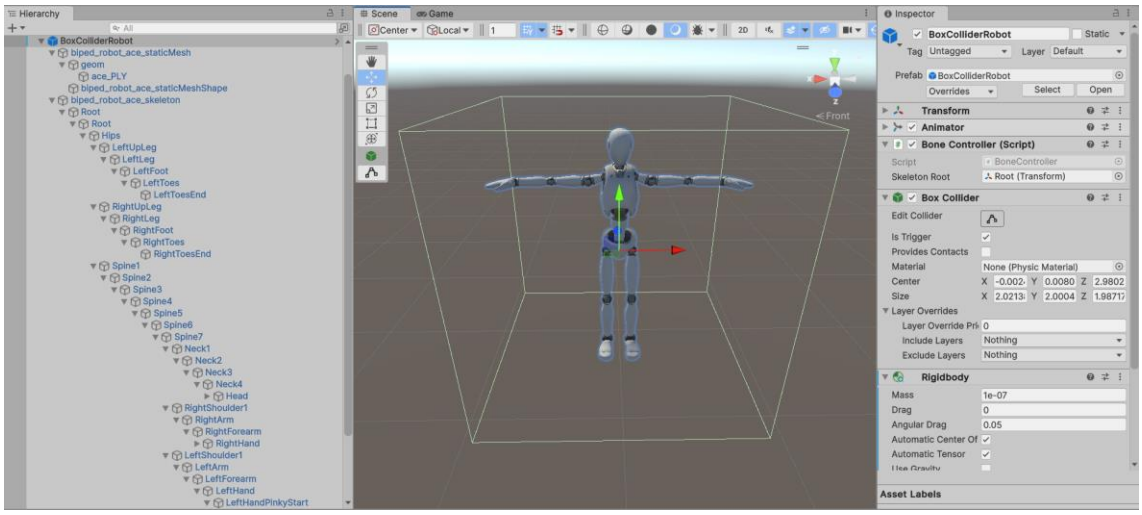
4.2.3 Emberi test detektálása és biztonsági vészjelzés adása a PLC felé

A kiterjesztett valóság alkalmazásban szereplő raktárrobot AR-támogatott digitális iker realizációja eddigi állapota szerint csak egyirányú kommunikációt létesít a fizikai infrastruktúrával, semmilyen a virtuális térben történő esemény nincsen hatással a fizikai vezérlésre, továbbá semmilyen funkció nem támogatja a humán-robot kollaborációt és együttműködést, a kiberfizikai interakció- és ütközésvizsgálatot vagy biztonságot. Ennek megfelelően az alkalmazás jelen stádiumában nem nevezhető hibrid iker dimenziójú AR-támogatott digitális ikerpárnak.

Ezen igyekszik változtatni a következő implementálandó funkció, mely nem más mint az emberi test detektálás, mely pozíciójának a raktárrobot közelében lévő megítélése esetén az AR-támogatott digitális iker azonnal visszajelzést küld a PLC-nek a robot leállításáról. Ennek PLC oldali szoftverkomponensei már lefejlesztésre kerültek, most az iPad Pro oldali fejlesztésen van a sor.

Ennek a problémának a megoldásához a kiterjesztett valóság alkalmazásnak először is valamilyen úton képesnek kell lennie az emberi test detektálására. Ebben nyújt segítséget az AR Foundation által biztosított emberi test követés (human body tracking) funkció. Ehhez a jelenet XR Origin objektumához hozzáadom az AR Human Body Manager szkriptet, majd egy új objektumot létrehozva, melyet Human Body Tracking-nek nevezek el, azt is ellátom egy Human Body Tracker szkripttel, melynek paraméteréül adom az XR Origint, valamint egy emberi testet reprezentáló 3D modellt, melyet megjelenít az érzékelt ember helyén.

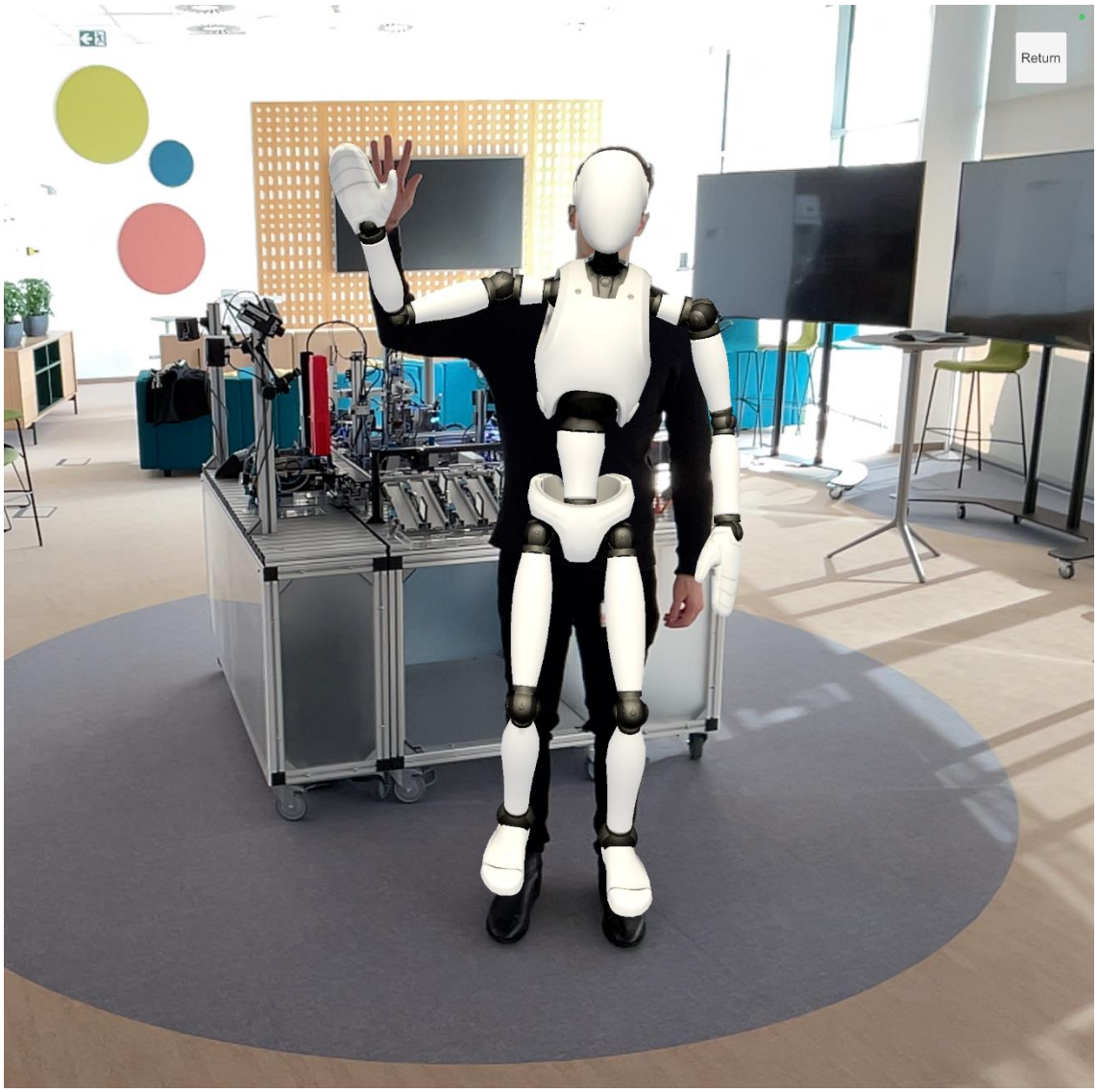
Ezzel a megoldással a kiterjesztett valóság alkalmazás abban az esetben, amikor emberi testet érzékel, rá vetítve megjeleníti rajta az ember modellt a detektált testtartásban. Ezzel az applikáció képes futásidőben detektálni az emberi jelenlétet a kamera betekintési szögében, a ráhelyezett modellel pedig képes annak aktuális 3D-s pozícióját nyomon követni, így alkalmas az érzékelt ember raktárrobottól való távolságának meghatározására.



38. ábra: Az AR Foundation emberalakú robot modellje, melyet emberi test detektáláskor képes a detektált emberre vetíteni. A modell ütköző dobozzal van körbevéve.

A virtuális raktárrobothoz való veszélyes közelséget egy korábban már alkalmazott megoldással, az úgynevezett box collider-ek, vagyis láthatatlan ütközésdetektáló dobozok használatával képes vagyok vizsgálni. Ez a 38. ábrán figyelhető meg. Ennek megfelelően mind a raktárrobotot, mint az emberi modellt ellátom egy-egy ilyen ütközésdetektáló aurával, melyek amennyiben metszetet képeznek, vagyis amikor az ember túlságosan megközelítette a robotot, az egy gyakran mintavételezett logikai változó billentésével detektálhatóvá válik, mely azonnali leállás utasításának kiadását kell, hogy eredményezze a PLC irányába.

A visszairányú kommunikáció megteremtésére a TCP szerver fogadó szoftverkomponense mellé egy üzenetküldő metódust is implementálni kell, mely a szerverrel kapcsolatot létesítő kliensnek, vagyis a PLC-nek címzi üzeneteit. Ennek a funkciónak a hívásával utasítható a PLC a raktárrobot leállítására, illetve újra engedélyezésére. Hívási logikája azonban korántsem egyértelmű, és nem is érdemes egy ilyen épséget veszélyeztető biztonsági kérdést a véletlenre bízni. Állapítsuk tehát meg!

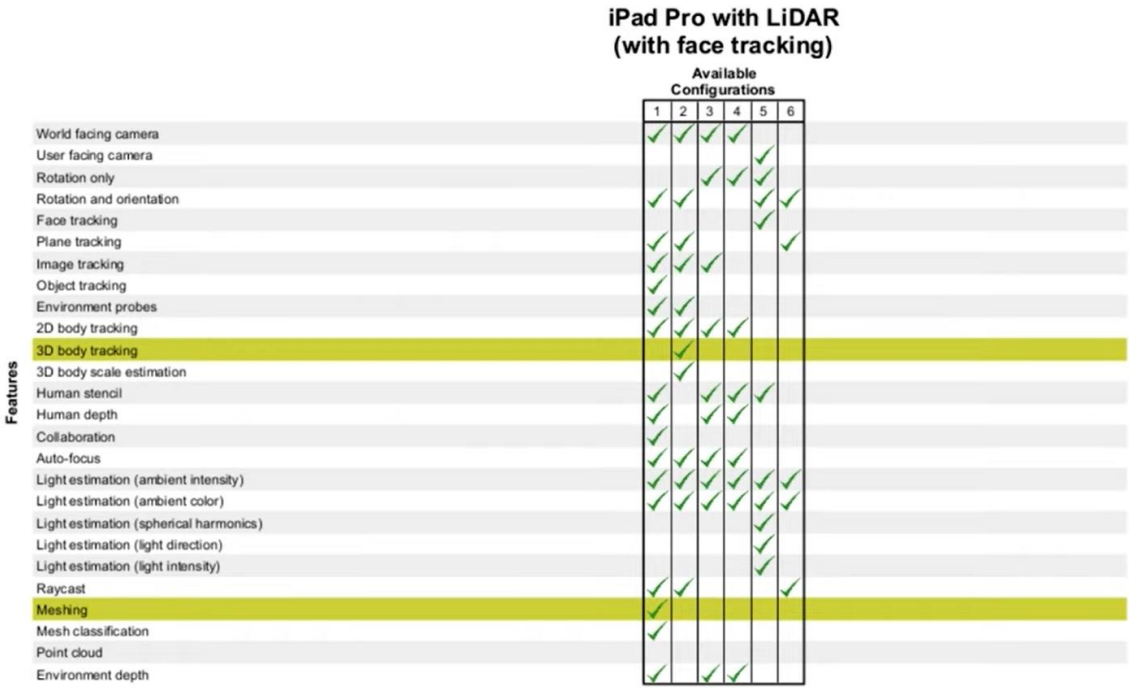


39. ábra: Az AR Foundation emberi test detektálásának funkciója.

Az AR Foundation által kínált emberi test detektálásának lehetősége, mely a 39. ábrán látható, meglehetősen pontos az elvártakhoz képest, azonban az ember testtartásainak valós idejű lekövetése közben néha képes elveszíteni a modell illesztést, melynek hatására rövid időközönként pillanatokra eltűnhet a virtuális ember modell, majd újra megjelenik. Ennek lekezeletlenül hagyásával nagyon gyorsan egymást követő engedélyező és tiltó üzenetekkel áraszthatjuk el a csatornát, melyek akár fel is torlódhatnak, ezzel kockáztatva azt, hogy egy valódi vészhelyzet esetén nem fog megfelelően működni a funkció. Ennek érdekében érdemes egy erre irányuló pergésmentesítést, valamint egy engedélyezés visszaszámlálót közbe iktatni, mely által az adott üzenetek csak egyszer lesznek elküldve, ezzel használható marad a csatorna, továbbá az engedélyező üzenet is csak akkor lesz kiadva, mikor már egy előre definiált

idő leforgása alatt, melyet én 5 másodpercnek határoztam meg, nem történt közeli emberi jelenlét érzékelése.

A funkciót az imént megtárgyalt szempontok alapján implementálva egy robusztus digitális iker vezérelte biztonsági visszacsatolást vagyok képes megvalósítani a HRC környezetben. A felhasználási eset egy kellemetlen tényezőt hordoz magával a felhasználói élményt figyelembe véve, mely nem más, hogy az Apple által kiadott ARKit bizonyos szenzoros számítógépes látás funkciókat szoftveresen nem enged egyidőben alkalmazni, melybe sajnos bele tartozik az occlusion és az emberi test követés funkciók inkompatibilitása, mely azt jelenti, hogy egy jeleneten belül csak az egyik elérhető a 40. ábrán látható táblázat szerint. Ennek következtében a két funkciót két különböző jelenetben fejlesztettem le és demonstráltam.



40. ábra: Az Apple ARKit által támogatott kiterjesztett valóság funkciók kompatibilitási táblázata.

4.3 A tervezett szoftverkomponensek implementációja

Az előző fejezetekben részleteibe menően kerültek tárgyalásra az iPad Pro-ra fejlesztett kiterjesztett valóság alkalmazás tervezett felhasználási eseteit megvalósító szoftverkomponensek, valamint azok használata a kívánt működés eléréséhez. Ezek, illetve az általuk megalkotni hivatott funkciók előkészítését követően az applikáció

alkalmassá válik azok kipróbálására, továbbá üzemszerű viselkedésüknek és együttműködésüknek, egymás rendeltetésszerű támogatásának és kiszolgálásának vizsgálatára.

Ekkor tehát a gyártósort és a virtuális raktárrobotot vezérlő PLC hardverkonfigurációját és vezérlőkódját tartalmazó TIA Portal projekt készen áll a letöltésre, a Unity-ben elkészített kiterjesztett valóság alkalmazás pedig készen áll az őt megvalósító Swift forráskód Xcode projektjének kiexportálására, majd annak telepítésére az iPad Pro-ra. A programokkal feltelepített eszközök között hálózati összeköttetést biztosítva, illetve azokat egy lokális alhálózatba helyezve vagyunk képesek megteremteni kettejük kommunikációját lehetővé téve kettejük kooperációját. Mindenekelőtt azonban nem szabad megfeledkezni az elérni kívánt és hibamenetes működésről való meggyőződésről.

5 Verifikáció és validáció

Az elkészült kiterjesztett valóság alkalmazás, és az azt kiszolgáló PLC program végső telepítése és beüzemelése előtt meg kell győződnöm arról, hogy az azokat alkotó szoftverkomponensek hibamentesek, kellően hatékonyak, valamint, hogy az együttes használatuk valóban azokat a funkciókat, felhasználási eseteket valósítják meg, melyeket eredetileg terveztünk, és melyekre valóban szükségünk van. A következő fejezetekben ennek érdekében tehát verifikációs és validációs eljárások kerülnek tárgyalásra, melyek elvégzésével képesek lehetünk megállapítani, hogy az szoftverünk alkalmas-e a kiadásra, leszállításra és az azt követő üzemszerű használatra.

5.1 A tervezett felhasználási esetek verifikációja szimulációs környezetben

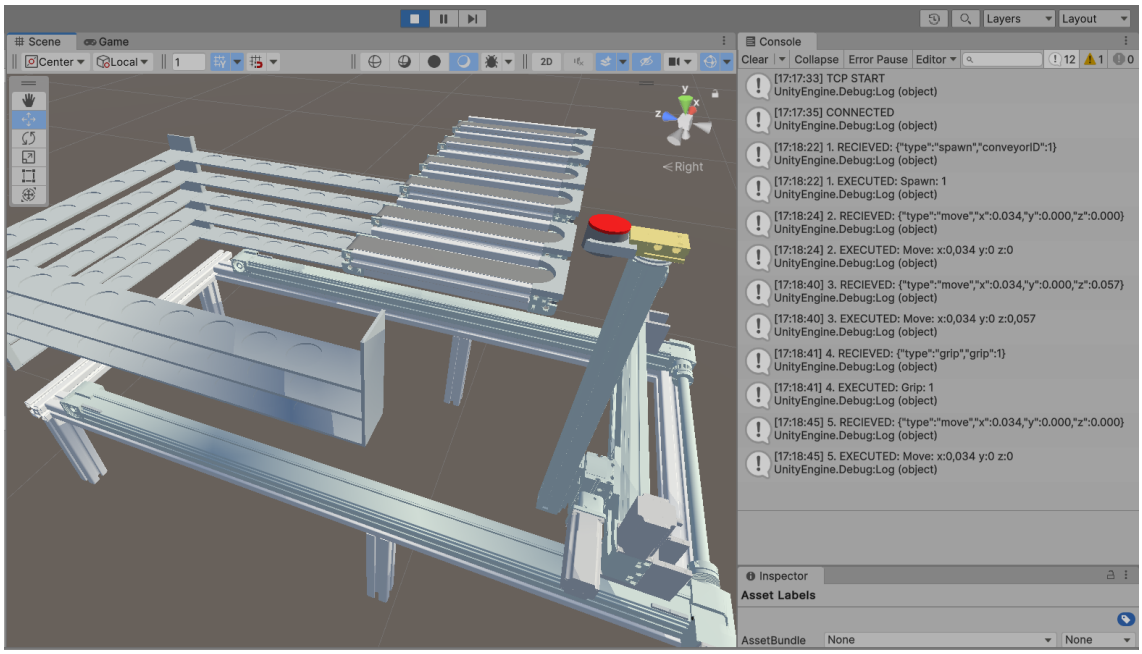
Ennek a kiterjesztett valóság alkalmazásnak fejlesztése annak módjából, valamint az ahhoz használt eszközök és környezetek funkcióiból és szolgáltatásaiból adódóan jelentős értelmezésbeli átfedésben van a virtuális beüzemelés digitalizációs eljárásával, mely ugyanezen okokból igaz a rendszer tesztelhetőségére is. Ez a párhuzam azért állítható, mert mind a virtuális raktárrobotot vezérlő PLC és az azt megjelenítő AR applikáció szoftverét helyszíni munka vagy fizikai eszköz használta nélkül, szimulációs környezetben voltam képes lefejleszteni. A következő fejezetekben ennek a szemléletnek az alkalmazásával, ugyancsak szimulációval igyekszem verifikálni az alkalmazásom funkcióinak hibamentes működését.

A verifikációt ennek megfelelően otthoni környezetben végzem el a PLC programot egy SIMATIC S7-PLCSIM Advanced V4.0 szoftver által szimulált S7-1500 PLC-re való letöltéssel, valamint a kiterjesztett valóság alkalmazást a Unity beépített szimulációs funkciójával. A szimulált működés továbbá lehetőséget ad a programok viselkedésének a fejlesztői környezetek által való futásidejű debug-olására, mely hatékonyabb tesztelési folyamatot eredményezhet. A szimulált felek hálózati kapcsolatának, ezáltal azok TCP kommunikációjának megteremtésére egy tetszőleges router használata alkalmas, így képesek lehetünk még a fizikai telepítés előtt meggyőződni a felhasználási eseteink hibamentességéről.

5.1.1 PLC által vezérelt AR virtuális raktárrobot szimulációja és verifikációja

Az imént tárgyalt szimulációs környezet felállítását, valamint a szimulációk elindítását követően elkezdhető a felhasználási esetek verifikációja, melyek közül az első a kiterjesztett valóság alkalmazásban megjelenő virtuális raktárrobot PLC általi vezérlésének hibamentességéről való meggyőződés. Ehhez a Unity-ben elindítom az ezt megvalósítani hivatott jelenet szimulációját, melyben az ezt követően megjelenő játékablakban az egér kattintásával az érintőképernyő megérintését reprezentálva vagyok képes elhelyezni, illetve további kattintásokkal tetszőlegesen módon horizontálisan áthelyezni a raktárrobot modelljét a jelenetben.

A robot megjelenítésével egyidőben elindul annak kommunikációért felelős szála, általa pedig TCP szervere. Ezzel párhuzamosan a TIA Portal projekttel ellátott szimulált PLC RUN állapotba lépését követően az általa futtatott TCP klienssel automatikusan csatlakozási kérelmet indítványoz a szerver felé, melyet követően szinte azonnal kiépül a kettejük közötti kapcsolat, melytől fogva a rendszer képes parancs üzenetek forgalmazására. Ennek procedúráját, valamint ezek után az egyes üzenetek váltására Unity konzolán megjelenő debug bejegyzésekből követhetem nyomon, melyben az egyes utasítások fogadását és végrehajtását is bejegyzem. Ezzel jól megfigyelhető, mely típusú üzenetek beérkezésének esete nincsen esetleg megfelelően lekezelve. Mindemellett a folyamatról vizuálisan is meggyőződhetek, hiszen a raktárrobot modell mozgását megfigyelve is meg lehet állapítani a viselkedés helyességét és valósághűségét. Ezt a folyamatot szemlélteti a 41. ábra.



41. ábra: A virtuális raktárrobot modell szimulációja Unity környezetben, amely fogadja, és végrehajtja a PLC utasításait.

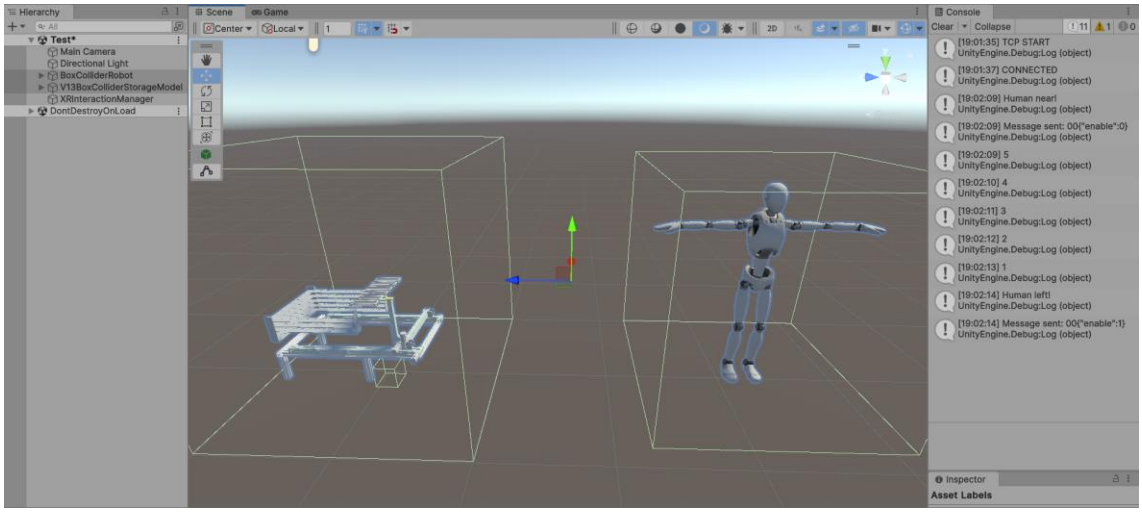
A működés vizsgálatára a PLC TIA Portal projektjében különböző tesztelési utasítássorozatokat definiáltam, melyek elkészült termékek raktározásának vezérlési scenario-it reprezentálja. Ezeket igyekeztem minél hosszabb, változatosabb, összetettebb, sok eset lefedését tartalmazó parancspermutációkkal feltölteni. Ezek futtatása alatt igyekeztem minél frekventáltabb utasításkiadást szimulálni, melynek megfelelően leggyakrabban tizedmásodpercenként indítottam üzeneteket, ezzel terhelve a hálózatot, valamint a fogadó oldal FIFO buffereinek kapacitását. A teszt scenario-k futtatása alatt semmilyen kapacitásbeli problémába nem ütköztem a hosszú időtartamú tesztelés alatt sem, melynek leghosszabb üzemidővizsgálata egy órán keresztül zajlott.

Ezzel a PLC által vezérelt AR virtuális raktárrobot modell felhasználási eset tesztesetei lefedésre kerültek, a szoftver az így felfedezett hibák kijavításával várhatóan a fizikai eszközökre telepítve is problémamentesen fog működni.

5.1.2 Emberi jelenlét biztonsági vészjelzésének szimulációja és verifikációja

A másik verifikálandó felhasználási eset a biztonsági vészjelzés emberi jelenlét érzékelésekor a virtuális raktárrobot modellje közelében. Az ezt megvalósító szoftvert az előző fejezetben tárgyalt módon, szimulációs környezetben tesztelem. Ehhez a Unity szimulációját indítom elsőként, melyet követően a játékablakban kattintással elhelyezem a virtuális raktárrobot modellt a jelenetben. Ezután a szimulált PLC-t RUN állapotba helyezve, ezzel elindítva annak programvégrehajtását az eddigiekhez hasonló módon automatikusan kapcsolódik a robot lehelyezésekor elindult TCP szerverre. Ezzel előállt a tesztelésre alkalmas környezet.

Az emberi jelenlét szimulálására a jelenetben egy emberalakú robot modellt példányosítok, melyet az éles használatban a detektált emberre vetít rá a kiterjesztett valóság alkalmazás. Az emberalakú robot modellje köré, valamint szintúgy a raktárrobot modellje köré a fejlesztési fázisban a 42. ábrán is látható, úgynevezett box collider-eket, vagyis láthatatlan ütköző dobozokat helyeztem el, melyeknek amennyiben metszetet képeznek. Amikor az ember túl közel megy a raktárrobothoz, egy biztonsági üzenet kerül elküldésre a kiterjesztett valóság alkalmazásból a PLC felé, mely esetben a PLC leállítja a vezérlést.



42. ábra: A kiterjesztett valóság alkalmazás emberi test detektálásának szimulációja Unity környezetben, a biztonsági vészjelzés küldése a PLC-nek.

A két lehelyezett modellt a szimulált Unity környezetben képes vagyok tetszőlegesen mozgatni, ezáltal mesterségesen előállítva az ember közelségének és

eltávolodásának esetét, mely a biztonsági vészjelzés küldésének kiváltója. Az üzenetváltások, a pergésmentesítés, valamint az ember távozásáról való biztos meggyőződés érdekében implementált visszaszámláló nyomon követhető a Unity konzolán megjelenő debug bejegyzésekben. Az üzenetek megérkezését a TIA Portal-ban, a szimulált PLC-vel online kapcsolatot létesítve egy tesztelésre definiált bool változó billegésével lehetünk képesek meggyőződni az információk sikeres beérkezéséről.

Az imént tárgyalt verifikációs eljárásokkal maradéktalanul megbizonyosodhattunk a felhasználási eseteket megvalósító szoftverek hibamentességéről, ennek tudatában nyugodt szívvel tölthetjük le őket fizikai céleszközeikre az éles validáció végrehajtása érdekében.

5.2 A tervezett felhasználási esetek telepítése és éles környezetben való validációja

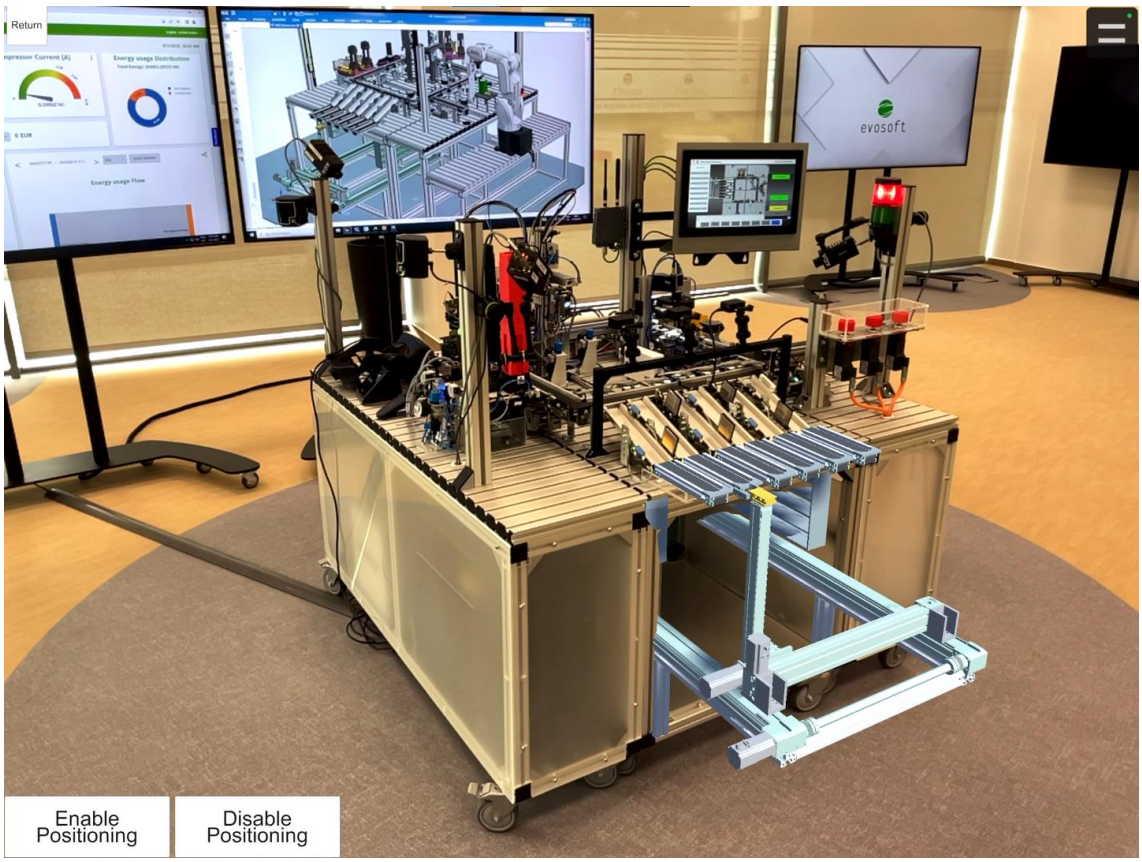
A kiterjesztett valóság alkalmazást megvalósító szoftverkomponensek szimulációs környezetben való verifikációjával megbizonyosodhattunk az azok által megvalósított funkciók hibátlan működéséről. Érdemes azonban a szimulációt követően valós körülmények között is, a programjaink számára dedikált hardverekre telepítve meggyőződni arról, hogy az applikáció valóban a tervezett felhasználási eseteket valósítja meg remélhetőleg jó felhasználási élmény biztosítása mellett. A rendszerbe implementált funkciók validációját ennek tekintetében hajtom végre a gyártósor modell, valamint az azt vezérlő fizikai PLC mellett az iPad Pro-ra telepített kiterjesztett valóság alkalmazással. Ehhez a PLC-re letöltöm a véglegesnek vélt TIA Portal projektet, illetve a tabletre is telepítem a Unity-ből kiexportált Xcode projektet. A táblagépet manuálisan csatlakoztatom a vezérlő PLC lokális hálózatához tartozó Wi-Fi routerhez, ezzel sikeresen előkészítésre került a validációs környezet.

5.2.1 PLC által vezérelt AR virtuális raktárrobot éles validációja

A validációs környezetben elsőként a kiterjesztett valóság alkalmazásban a felhasználói környezetbe integrált virtuális raktárrobot modell PLC általi vezérlésének

esetét vizsgálom meg, hogy valóban a tervezett funkció valósítja-e meg. Ehhez elindítom az iPad Pro-ra telepített kiterjesztett valóság alkalmazást, majd az érintőkijelzőhöz hozzáérve megjelenítem a méretarányos robot modellt az AR applikáció által abban a pontban érzékelt síkfelületen, mely ezesetben a padló. További pozícionálással és forgatással hozzáillesztem a virtuális modellt a fizikai gyártósor megfelelő oldalához.

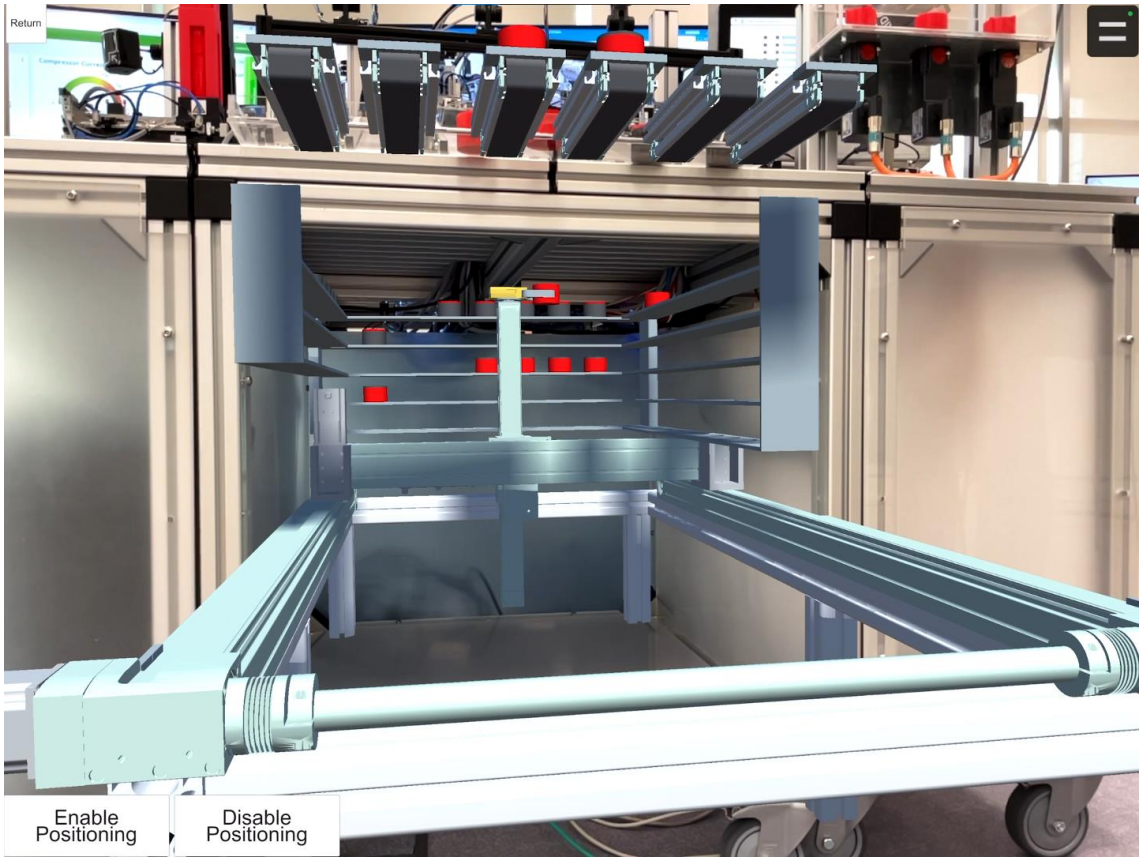
Az így összeállított kiberfizikai rendszer a táblagéppel körbejárható, mely esetben jól megmutatkozik, hogy a robot modell nagy pontossággal egyhelyben marad. Ekkor az is megfigyelhető, hogy az iPad Pro LiDAR szenzorának mélységképe által az alkalmazás az occlusion technológiájának köszönhetően sikeresen meg tudta állapítani, hogy a megjelenített raktárrobot melyik része nem szabad, hogy látszódjon a felhasználó szemszögéből a fizikai objektumok kitakarásától, ezzel egy realisztikus, kellemes látványú felhasználói élményt nyújtva, mely a 43. ábrán is szemügyre vehető.



43. ábra: A PLC által vezérelt AR raktárrobot modell validáció közben. A képen a szürkéskék elemek a virtuális objektum részei, a többi a valós környezet.

A táblagép kijelzőjén megjeleníthető debug bejegyzésekkel meggyőződhetek a TCP kapcsolat Wi-Fi útján való kiépüléséről, melyet követően gyártási megrendeléseket

indítok el a gyártósor HMI paneljéről. Az elkészült termékek futószalagról való legördülését követően láthatóan azonnal megjelennek a virtuális raktárroboton, melyet a robotkar a PLC utasításaira azonnal felkap, és elhelyezi az alsóraktárban, mely a 44. ábrán figyelhető meg. A leghosszabb megszakítás nélküli folyamatos gyártás üzemszerű futtatása 45 percen keresztül zajlott, mely esetben az alkalmazás nem ütközött sem számítási kapacitásbeli, sem memóriabeli korlátokba.



44. ábra: A PLC által vezérelt AR raktárrobot modell alsóraktára validáció közben. A képen a szürkéskék és piros-fekete elemek a virtuális objektum részei, a többi a valós környezet.

Az alkalmazás használata a megállapítottak alapján megfelelően látványosnak, a tervezett funkciókat pedig az elvárásoknak megfelelően tartalmazónak bizonyult, így a PLC által vezérelt AR raktárrobot felhasználási esetének validációja sikeresnek mondható, továbbá ezzel párhuzamosan az is kijelenthető, hogy hibamentesen teljesíti az AR-támogatott digitális ikrek virtuális ikerpár dimenziójának feltételeit.

5.2.2 Emberi jelenlét biztonsági vészjelzésének éles validációja

Az utolsó validálandó funkció nem más, mint a virtuális raktárrobothoz közelmenő ember detektálása esetén a gyártósort vezérlő PLC felé kiadott biztonsági vészjelzés és leállás felhasználási esete. Ehhez a tesztkörnyezet előzőekhez hasonló előkészítését követően szintén elindítom a kiterjesztett valóság applikációt. A korábban már tárgyalt módon elhelyezem a raktárrobotot a valós térben, majd elhelyezem a táblagépet egy fix pozícióba, hogy bele tudjak sétálni kamerájának betekintési szögébe. Ezt a folyamatot egyedül is jól nyomon tudtam követni a táblagép kijelzőjét egy televízióra kivetítve, mely által magam is képes voltam vizsgálni az eseményeket, illetve az általam befolyásolt viselkedést. Egy másik televízión a PLC-vel online kapcsolatot létesített TIA Portal projekt belső, futásidejű állapotváltozásainak megfigyelésével bizonyosodhattam meg az üzenetek tényleges átadásáról.

45. ábra: Az emberi test detektálásának és biztonsági vészjelzésnek verifikációja.

Az imént vázolt, illetve a 45. ábrán megfigyelhető környezetben a gyártósort a testemmel folyamatosan megközelítve, majd attól eltávolodva a táblagép kivetített képernyőjéről jól meg tudtam győződni a raktárrobot viselkedéséről, mely amennyiben a

közelében tartózkodtam, nem szállított tovább termékeket az alsóraktárba, csak távozásom után öt másodperccel folytatta azt. A tesztelés során az emberi test detektálása meglehetősen pontos volt, a jelenlétemben kifejezetten élvezhető felhasználói élményt jelentett, a kamera betekintési szögéből eltávolodva azonban ritkább esetekben, rosszabb fényviszonyokban hajlamos volt hibásan emberi testeket detektálni bizonyos élettelen tárgyakban, amely kevés számú előfordulásban befolyással tudott lenni a működésre. Mindent összevetve az alkalmazás programozási hiba nélkül hajtja végre a tervezett funkciót, így a validáció sikeres megítélésével párhuzamosan érdemes kiemelni, hogy az applikáció megvalósít egy hibrid ikerpár dimenziójú AR-támogatott digitális ikerre jellemző, humán-robot kollaborációt támogató biztonsági funkciót, mely által kategorizálható az említett dimenzióba.

Ezzel a kiterjesztett valóság alkalmazás minden felhasználási esete verifikációra, illetve validációra került, melyek a rendeltetésszerű használat jó demonstrációjaként alkalmasak voltak az esetleges hibák felderítésére, illetve az üzemszerű működés reprezentálására. Ennek a procedúrának az elvégzését követően a hibamentesnek vélt applikáció kiadható, leszállítható.

6 Tudományos eredmény

A dolgozatom a céljaként kitűzött tudományos közéletre gyakorolt hatást az iparban megjelenő digitalizációs eljárások által megalkotott kiberfizikai rendszerek üzemének a kiterjesztett valóság technológiája által való támogatásának kutatásával és demonstrációjával igyekszik elérni. Erre a vizualizációs, immerzív HMI megoldásra mutatkozó igényt, illetve annak szükségességét úgy érzem nem kell bizonygatnom, hiszen a hétköznapokban, és az ipari alkalmazásokban egyaránt folyamatosan növekvő népszerűség övezi. A kiterjesztett valóság technológiájának fejlődése, illetve annak egyre kedvezőbb elérhetősége egyenes utat jelenthet az Ipar 4.0-ban lefektetett digitalizációs alapoktól az Ipar 5.0 által egyik központi elvében megfogalmazott humánközpontú koncepció megvalósításában az AR eszközökkel és szolgáltatásokkal felszerelt operátorok okos operátorokká való fejlesztésével, ezáltal lehetővé téve az emberek kollaboratív robotokkal való munkamegosztását, vagyis a humán-robot kollaborációt (HRC). Ennek tudatában az AR-támogatott kiterjesztett valóság alkalmazások, illetve azok három dimenziójának megvalósításának demonstrációi megkerülhetetlen lépcsőt jelentenek a jövő ipara felé vezető úton. Az ipari szempontokat általánosítva és kiterjesztve a technológiai fejlődés modern társadalomra gyakorolt hatására az AR jól párhuzamba vonható a Társadalom 5.0 által lefektetett emberközpontú, életszínvonalat középpontba helyező koncepcióval is, melyben az egyes közösségi szolgáltatások a fejlett rendszerek által magas minőségben, felhasználói komfortot jelentő módon érhető el a lakosság számára.

Az általam készített AR-támogatott digitális iker alkalmazás egy operátorbarát ember-gép interfészt nyújt, mely amellett, hogy a kiterjesztett valóság technológiája által egy virtuális objektum nagy pontossággal vált illeszthetővé a hozzá tartozó fizikai infrastruktúrához, valós humán-robot kollaborációs felhasználási esettel is rendelkezik, mely egyértelmű megfeleltethetőséget jelent a virtuális ikerpáron túl a hierarchiai modellben afölött álló hibrid iker koncepciójával egyaránt. Az alkalmazásom használata továbbá lehetővé teszi, hogy egy virtuális üzembe helyezés alatt álló gyártórendszernek annak megépítése előtt kipróbálhassuk annak biztonsági funkciót, ezzel kevesebb fejlesztési időt, hatékonyabb fizikai beüzemelést és összességében költséghatékonyabb kivitelezést eredményez.

A kiterjesztett valóság relevanciájának megerősítése mellett az applikációm demonstrációjával annak széleskörű és nagy funkcióbázisú fejlesztői lehetőségeiről is igyekeztem egy objektív álláspont irányának kijelölésében lépéseket tenni. Ennek létrehozása során megbizonyosodhattam róla, hogy a digitális ikreket magasabb szintre helyező AR-támogatott digitális ikrek nagy fejlesztői potenciált rejtenek, a rájuk jellemző felhasználási esetek pedig belépő szinten akár otthoni körülmények között is megvalósíthatóak. A technológia, valamint fejlesztéshez szükséges eszközök rendelkezésre állása tehát tart már olyan szinten, hogy a megfelelő szakmai ismeretek birtokában nagy hatékonyságú alkalmazások fejleszthetőek ipari folyamataink kiszolgálására és támogatására az AR-támogatott digitális ikreknek akár magas dimenzióiban is.

Dolgozatom, illetve annak tárgyául szolgáló kiterjesztett valóság alkalmazásom az imént tárgyalt elérhetőségre és AR-ben rejlő lehetőségekre hivatott felhívni a figyelmet bízva ezáltal abban, hogy az ipar ennek a területének piaci szereplői között a nagy technológiai potenciál következtében kialakuló verseny gyorsabb ütemben támogatja annak bevezetését, széleskörű elterjedését, ezáltal a jövő iparának korábbi eljövetelét.

7 Összefoglalás

A dolgozatomban a kiterjesztett valóság ipari felhasználásait vizsgáltam a technológiát övező modern szemlélet figyelembevételével. Az AR-ban rejlő lehetőségekről és ipari potenciálról a tudományos közéletben fellelhető state-of-the-art kutatások által igyekeztem tájékozottságot szerezni a demonstrálni érdemes felhasználási esetek célul kitűzésének érdekében. Megismerkedtem tehát a kiterjesztett valóság modern paradigmáival, a digitalizációs eljárásoknak nyújtott támogatásával, az AR-támogatott digitális ikrek három dimenziójával, végül az ipar fejlődésében betöltött szerepével, melyek nagy relevanciájú területnek festik le a témakört.

Az irodalomkutatásnak köszönhetően megfelelő irányt sikerült kitűzni a kiterjesztett valóság alkalmazásom számára, mely egy PLC által vezérelt ipari gyártósor digitális ikrében szereplő, azonban fizikailag meg nem épített gyártási komponens működésének vizualizációja, valamint egy hozzá kapcsolódó biztonsági funkció szolgáltatása volt az AR segítségével. A magas szintű tervezési architektúrát, amelyet meg is valósítottam, a 15. ábra mutatja be. Az applikáció elkészítéséhez naprakész kompetencia kiépítésére volt szükség a PLC programozásban és C# nyelvi ismeretekben, hiszen a célul kitűzött felhasználási eseteim által támogatott gyártósort vezérlő PLC programját a TIA Portál integrált fejlesztői környezetben kellett kibővíteni az általam megtervezett funkciókkal, a kiterjesztett valóság alkalmazást pedig a Unity játékmotor és integrált fejlesztői környezetben hoztam létre, melynek felhasználására dedikált eszköz az 5. generációs Apple iPad Pro volt.

Az alkalmazás fejlesztése során számos objektumorientált szoftverfejlesztési, ipari automatizálási, hálózati kommunikációval és szoftverinterfészekkel kapcsolatos, illetve a számítógépes látás és a képfeldolgozás témakörök tudományterületeinek ismereteit kellett vegyítve alkalmaznom számos fejlesztői eszköz és környezet párhuzamos alkalmazásával. Az applikáció megalkotásának folyamata egy szakmai víziót, problémamegoldást és hatékonyságot igénylő, tanulságos folyamat volt, megvalósításának összetettsége, valamint a hozzá szükséges tudás megszerzése pedig széles eszközismerettel és informatikai területeken átívelő kompetenciával ruházott fel.

Az elkészült kiterjesztett valóság alkalmazás a szakirodalmaknak megfelelően az AR-támogatott digitális ikrek virtuális iker és hibrid iker dimenziójából is tartalmaz

felhasználási esetet, mellyel a technológia relevanciáját és elérhetőségét igyekeztem reprezentálni. Reményeim szerint demonstrációmmal hozzá tudok járulni a kiterjesztett valóság technológia piacának fellendüléséhez, ezáltal az általa ígért ipari fejlődés fokozott ütemű bevezetéséhez és elterjedéséhez.

Irodalomjegyzék

[1] A. Renda, S. Schwaag Serger, D. Tataj, A. Morlet, D. Isaksson, F. Martins, M. Mir Roca, C. Hidalgo, A. Huang, S. Dixson-Decl`eve, P. Balland, F. Bria, C. Charveriat, K. Dunlop, and E. Giovannini, "Industry 5.0, a transformative vision for Europe: governing systemic transformations towards a sustainable industry". EC Dir.-General for R.&I., 2021., URL: https://op.europa.eu/en/web/eu-law-and-publications/publication-detail/-/publication/38a2fa08-728e-11ec-9136-01aa75ed71a1

[2] Sihan Huang, Baicun Wang, Xingyu Li, Pai Zheng, Dimitris Mourtzis, Lihui Wang, „Industry 5.0 and Society 5.0—Comparison, complementation and co-evolution”, Journal of Manufacturing Systems, 2022., URL: https://www.sciencedirect.com/science/article/pii/S0278612522001224

[3] Isabell Wohlgenannt, Alexander Simons, Stefan Stieglitz: *Virtual Reality*, 2020., URL: https://link.springer.com/article/10.1007/s12599-020-00658-9#Sec4

[4] *Zeiss Invests Further Into Augmented Reality Creating AR/VR Competence Center,* 2023., URL: https://metrology.news/zeiss-invests-further-into-augmented-reality-creating-ar-vr-competence-center/

[5] Daehyeon Lee, Woosung Shim, Munyong Lee, Seunghyun Lee, Kye-Dong Jung, Soonchul Kwon: *Performance Evaluation of Ground AR Anchor with WebXR Device API*, 2021., URL: https://www.mdpi.com/2076-3417/11/17/7877

[6] Microsoft HoloLens 2, URL: https://www.microsoft.com/hu-hu/hololens

[7] Y. Yin, P. Zheng, C. Li, L. Wang, "A state-of-the-art survey on Augmented Reality-assisted Digital Twin for futuristic human-centric industry transformation." Robotics and Computer-Integrated Manufacturing, 81, 102515. Elsevier. 2023., URL: https://www.researchgate.net/publication/366604991_A_State-of-the-art_Survey_on_Augmented_Reality-assisted_Digital_Twin_for_Futuristic_Human-centric_Industy_Transformation

[8] Dr. Michael Grieves, John Vickers, „Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems (Excerpt)”, 2016., URL: https://www.researchgate.net/publication/307509727_Origins_of_the_Digital_Twin_Concept

[9] Dobaj, J.; Riel, A.; Macher, G.; Egretzberger, M. Towards DevOps for Cyber-Physical Systems (CPSs): Resilient Self-Adaptive Software for Sustainable Human-Centric Smart CPS Facilitated by Digital Twins. Machines 2023, 11, 973. https://doi.org/10.3390/machines11100973

[10] Haraszkó Csaba, „Virtuális üzembe helyezés könnyedén Process Simulate VC Lite segítségével”, 2021., URL: https://digitalisgyar.com/2021/11/04/virtualis-uzembe-helyezes-konnyeden-process-simulate-vc-lite-segitsegevel/

[11] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, and K. Ueda, "Cyber-physical systems in manufacturing", Cirp Annals, 65(2), pp. 621-641. 2016., URL: https://www.sciencedirect.com/science/article/pii/S0007850616301974

[12] Siemens, „SIMATIC/SIMOTION Virtual Commissioning with Hardware in the Loop”, 2018., URL: https://support.industry.siemens.com/cs/document/109758739/simatic-simotion-virtual-commissioning-with-hardware-in-the-loop?dti=0&lc=en-CH

[13] Julie Carmigniani, Borko Furht, Marco Anisetti, Paolo Ceravolo, Ernesto Damiani, Misa Ivkovic: *Augmented reality technologies, systems and applications*, 2010., URL: https://link.springer.com/article/10.1007/s11042-010-0660-6

[14] Apple iPad Pro, 12.9-inch (5th generation) - Technical Specifications, URL: https://support.apple.com/kb/SP844?viewlocale=en_US&locale=hu_HU

[15] Behnam Behroozpour, Phillip A. M. Sandborn, Ming C. Wu, and Bernhard E. Boser: *Lidar System Architectures and Circuits*, 2017., URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8067701

[16] Tech-Protect SC Pen tok Apple iPad Pro 11 2020 / 2021 / 2022, fekete, URL: https://momanio.hu/tech-protect-sc-pen-tok-apple-ipad-pro-11-2020-2021-2022-fekete

[17] Apple: Xcode, URL: https://developer.apple.com/xcode/

[18] Apple: Swift, URL: https://developer.apple.com/swift/

[19] *The Best Software Tools To Use With Cocos*, 2020., URL: https://www.cocos.com/en/post/the-best-software-tools-to-use-with-cocos

[20] Apple: ARKit, URL: https://developer.apple.com/augmented-reality/

[21] Univerity of Zurich, Department of Informatics, Robotics and Perception Group: *Visual and Inertial Odometry*, URL: https://www.ifi.uzh.ch/en/rpg/research/research_vo.html

[22] Hugh Durrant-Whyte, Tim Baliley: *Simultaneous Localizationand Mapping: Part I*, 2006., URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1638022

[23] Andrew Orr: *5 iOS 12 AR Apps to Test Your New Camera*, 2018., URL: https://www.macobserver.com/tips/quick-tip/ios-12-ar-apps/

[24] Unity, URL: https://unity.com/

[25] Sercan Altundaş: *NPC AI System Based on Gameplay Recordings*, 2018., URL: https://www.researchgate.net/publication/326434680_NPC_AI_System_Based_on_Gameplay_Recordings

[26] Unity AR Foundation, URL: https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/index.html

[27] Prayash Thapa: *Cross-platform Augmented Reality with Unity*, 2019., URL: https://www.viget.com/articles/cross-platform-ar-with-unity/

[28] Siemens: SIMATIC S7-1500, URL: https://www.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500.html

[29] sivaranjith, *Simatic S7 memory areas*, 2017., URL: https://forumautomation.com/t/simatic-s7-memory-areas/2695

[30] Hans Berger: *Automating with SIMATIC S7-1500, Configuring, Programming and Testing with STEP 7 Professional*, 2014.

[31] Siemens: Totally Integrated Automation Portal – Always ready for tomorrow, URL: https://www.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html

[32] Darko Živković: Siemens Digital Industries Webinari FA5: SIMATIC Safety, 2020., URL: https://assets.new.siemens.com/siemens/assets/api/uuid:00680ccc-6cd1-43f3-bf2a-93778f2a04da/fa5-simatic-safety.pdf

[33] Siemens: NX software, URL: https://plm.sw.siemens.com/en-US/nx/

Az URL-ek elérési dátuma: 2023.11.02.