

# Interakcija s objektom od interesa u sklopu miješane stvarnosti

---

Štrkalj, Natalija

Master's thesis / Diplomski rad

2024

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:063953>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-13**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

**Natalija Štrkalj**

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

Mentor:

Izv. prof. dr. sc. Tomislav Stipančić, dipl. ing.

Student:

Natalija Štrkalj

Zagreb, 2024.

Izjavljujem da sam ovaj rad izradila samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se prof. dr.sc. Tomislavu Stipančiću na mentorstvu, susretljivosti i riječima podrške te asistentu mag. ing. Leonu Korenu na pomoći s učitavanjem projekta na *HoloLens* uređaj. Također, zahvaljujem se svojim roditeljima što su mi pružali podršku prilikom studiranja te prijateljima koji su uvijek bili spremni uskočiti u pomoć, koji su samnom proživjeli sve lijepe, vesele, smiješne, tužne i ružne trenutke i tolerirali moje „povremeno“ jadanje. Za kraj, zahvaljujem se najboljem dečku na svijetu, na pozitivnoj energiji koju je uvijek spreman podijeliti sa mnom, na svim lijepim riječima i djelima te što je uvijek bio tu za mene i vjerovao da ću uspjeti, hvala ti od srca.

Natalija Štrkalj



SVEUČILIŠTE U ZAGREBU

FAKULTET STROJARSTVA I BRODOGRADNJE

Središnje povjerenstvo za završne i diplomske ispite

Povjerenstvo za diplomske ispite studija strojarstva za smjerove:

Proizvodno inženjerstvo, inženjerstvo materijala, industrijsko inženjerstvo i menadžment,  
mehatronika i robotika, autonomni sustavi i računalna inteligencija



Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 - 04 / 24 - 06 / 1	
Ur.broj: 15 - 24 -	

## DIPLOMSKI ZADATAK

Student: **Natalija Štrkalj** JMBAG: 0035215980

Naslov rada na hrvatskom jeziku: **Interakcija s objektom od interesa u sklopu miješane stvarnosti**

Naslov rada na engleskom jeziku: **Interaction with an object of interest in mixed reality**

Opis zadatka:

Primjene virtualne, miješane i proširene stvarnosti se kreću u različitim sferama ljudskog djelovanja. Kao spona između fizičkog, virtualnog i proširenog svijeta koriste se različita sučelja u vidu ekrana i naočala za virtualnu ili proširenu stvarnost. Za kreiranje 3D svjetova moguće je koristiti različite razvojne i interakcijske okoline.

U radu je potrebno:

- proučiti komponente sustava miješane stvarnosti
- razviti 3D svijet koji uključuje ciljane objekte od interesa koristeći prikladnu programsku podršku
- povezati i podesiti hardverske te softverske komponente uključujući naočale za proširenu stvarnost MS HoloLense 2
- razviti interakcijsko sučelje koje omogućuje interakciju s objektima u virtualnom svijetu
- razvijeno cjelovito rješenje potrebno je eksperimentalno evaluirati u sklopu Laboratorija za projektiranje izradbenih i montažnih sustava.

U radu je potrebno navesti korištenu literaturu te eventualno dobivenu pomoć.

Zadatak zadan:

7. ožujka 2024.

Zadatak zadao:

Izv. prof. dr. sc. Tomislav Stipančić

Datum prednje rada:

9. svibnja 2024.

Predviđeni datumi obrane:

13. – 17. svibnja 2024.

Predsjednik Povjerenstva:

Prof. dr. sc. Ivica Garašić

## SADRŽAJ

SADRŽAJ .....	I
POPIS SLIKA .....	III
POPIS TABLICA.....	VI
POPIS KRATICA .....	VII
SAŽETAK.....	VIII
SUMMARY .....	IX
1. UVOD.....	1
2. POVIJESNI PREGLED RAZVOJA URANJAJUĆIH TEHNOLOGIJA.....	2
3. VIRTUALNA, PROŠIRENA I MJEŠOVITA STVARNOST.....	8
3.1. Općenito o sustavima virtualne, proširene i mješovite stvarnosti.....	8
3.2. Značajke sustava virtualne, proširene i mješovite stvarnosti.....	9
4. KOMPONENTE SUSTAVA MJEŠOVITE STVARNOSTI.....	13
4.1. Hardverske komponente .....	13
4.1.1. Opis glavnih senzora smještenih na uređaje za mješovitu stvarnost .....	14
4.2. Softverske komponente.....	15
4.2.1. Unreal Engine .....	16
5. UREĐAJI ZA MJEŠOVITU STVARNOST.....	17
5.1. <i>Microsoft HoloLens 2</i> .....	17
5.2. Preostali popularniji HMD-ovi današnjice [28].....	19
5.3. Primjena <i>Microsoft HoloLens 2</i> uređaja .....	21
6. RAZVOJ 3D SVIJETA ZA MJEŠOVITU STVARNOST .....	24
6.1. Osnovne informacije o <i>Unreal Engine</i> -u .....	24
6.2. Instalacija <i>Unreal Engine</i> -a .....	26
6.3. Instalacija <i>Visual Studio Installer</i> -a te <i>Visual Studio Code</i> -a .....	27
6.3.1. Javljanje pogrešaka i upozorenja prilikom kreiranja projekta te pakiranja za HoloLens uređaj.....	28
6.4. Izrada aplikacije u <i>UnrealEngine</i> -u .....	28
6.4.1. Instalacija potrebnih MR alata za Unreal Engine .....	29
6.4.2. Stvaranje Unreal Engine projekta te prikaz osnovnih postavki i radnji za omogućavanje sustava mješovite stvarnosti .....	30
6.4.3. Učitavanje 3D modela zrakoplova te omogućavanje interakcije s istim .....	35
6.4.4. Izrada početnih virtualnih tipkala za stvaranje zrakoplova i korisničkog sučelja... ..	37
6.4.5. Izrada korisničkog sučelja za zrakoplov .....	41
6.4.5.1. Tipkalo About Airplane .....	41
6.4.5.2. Tipkala Rotate i Move .....	42
6.4.5.3. Tipkalo Scale .....	44
6.4.5.4. Tipkalo Show Parts .....	45
6.4.6. Prikaz dijelova zrakoplova.....	47
6.4.7. Uporaba glasovne naredbe „Explore“ .....	48

---

7. EKSPERIMENTALNA EVALUACIJA RAZVIJENOG 3D SVIJETA .....	51
7.1. Pakiranje projekta za <i>HoloLens 2</i> .....	51
7.2. Učitavanje paketa na <i>HoloLens 2</i> uređaj putem <i>Windows Device Portal</i> -a .....	53
7.3. Pokretanje aplikacije mješovite stvarnosti na <i>HoloLens 2</i> uređaju.....	53
7.4. Kritički osvrt .....	57
7.5. Unaprjeđivanje aplikacije .....	58
8. ZAKLJUČAK.....	59
LITERATURA.....	60

**POPIS SLIKA**

Slika 1.	Prikaz prvih virtualnih naočala <i>Sensorama</i> [5] .....	2
Slika 2.	Prikaz prvog VR zaslona za glavu pod nazivom „ <i>Sword of Damocles</i> “ [8] .....	3
Slika 3.	Primjer jednog od niza projekata Myron W. Kruegera pod nazivom „ <i>Metaplay</i> “ koji je doveo do razvoja projekta „ <i>Videoplace</i> “ [9] .....	4
Slika 4.	Primjena naočala i rukavica u sklopu projekta „ <i>VIEW</i> “ [7] .....	5
Slika 5.	Prikaz modernog VR HMD-a <i>Oculus Rift</i> [12] .....	6
Slika 6.	Prikaz <i>Google</i> -ovih „pametnih naočala“ [13] .....	6
Slika 7.	Prve <i>Microsoft</i> -ove AR/MR naočale <i>Microsoft HoloLens</i> [15] .....	7
Slika 8.	Slikoviti prikaz raspona mješovite stvarnosti kao kombinacije proširene i virtualne [19] .....	9
Slika 9.	Primjer uporabe sustava virtualne stvarnosti putem HMD-a [20] .....	10
Slika 10.	Primjer uporabe sustava proširene stvarnosti putem pametnog telefona [21] .....	10
Slika 11.	Primjer uporabe sustava mješovite stvarnosti putem HMD-a [22] .....	11
Slika 12.	Prozirni optički zaslon AR i MR sustava [24] .....	14
Slika 13.	Video zaslon AR i MR sustava [24] .....	14
Slika 14.	<i>Microsoft HoloLens 2</i> [29] .....	18
Slika 15.	<i>Microsoft HoloLens 2</i> – mogućnost praćenja ruku [30] .....	18
Slika 16.	<i>Microsoft HoloLens 2</i> – mogućnost praćenja očiju [30] .....	19
Slika 17.	<i>Microsoft HoloLens 2</i> – mogućnost mapiranja prostora [30] .....	19
Slika 18.	<i>Magic Leap 2</i> [33] .....	20
Slika 19.	<i>Meta Quest Pro</i> [34] .....	20
Slika 20.	<i>Pico 4</i> [35] .....	21
Slika 21.	Primjena <i>HoloLens 2</i> uređaja u proizvodnji te radu na daljinu [36] .....	22
Slika 22.	Primjena <i>HoloLens 2</i> uređaja u proizvodnji [36] .....	23
Slika 23.	Prikaz <i>Viewport</i> -a (sredina), <i>Outliner</i> -a (desno) te <i>Content Browser</i> -a (dolje) <i>Unreal Engine</i> -a .....	25
Slika 24.	Različite klase elemenata <i>Unreal Engine</i> -a .....	25
Slika 25.	Prikaz instaliranih verzija <i>Unreal Engine</i> -a .....	26
Slika 26.	Postavke instalacije <i>Unreal Engine</i> -a .....	27
Slika 27.	Modifikacije <i>Visual Studio Build Tools</i> -a .....	27
Slika 28.	Potrebni <i>Microsoft</i> alati za mješovitu stvarnost dostupni na <i>Unreal Engine Marketplace</i> -u .....	29
Slika 29.	Prikaz nekih značajki <i>Microsoft</i> -ovog dodatka za mješovitu stvarnost <i>Mixed Reality UX Tools</i> .....	30
Slika 30.	Odabir postavki prilikom kreiranja <i>Unreal Engine</i> projekta za mješovitu stvarnost .....	30
Slika 31.	Prikaz stvaranja <i>Data Asset</i> -a za <i>ARSessionConfig</i> unutar <i>Unreal Engine</i> -a .....	31
Slika 32.	Dodavanje <i>Player Start</i> -a unutar <i>Unreal Engine</i> -a .....	32
Slika 33.	Prikaz čvorova za pokretanje i završavanje sesije unutar <i>Level Blueprint</i> -a u <i>Unreal Engine</i> -u .....	32
Slika 34.	Prikaz <i>Viewport</i> -a klase <i>Pawn</i> , kategorije <i>Default Pawn</i> , unutar <i>Unreal Engine</i> -a .....	33
Slika 35.	Prikaz <i>Event Graph</i> -a klase <i>Pawn</i> , kategorije <i>Default Pawn</i> , unutar <i>Unreal Engine</i> -a .....	33
Slika 36.	Prikaz <i>Viewport</i> -a, klase <i>Game Mode Base</i> unutar <i>Unreal Engine</i> -a .....	34
Slika 37.	Prikaz postavki projekta unutar <i>Unreal Engine</i> -a .....	34
Slika 38.	Prikaz računalno generiranih ruku nakon pokretanja sesije unutar <i>Unreal Engine</i> -a .....	35



Slika 39.	Prikaz učitano modela zrakoplova <i>Airbus A340</i> u projekt <i>Unreal Engine</i> -a .....	36
Slika 40.	Prikaz učitano modela trupa zrakoplova <i>Airbus A340</i> u projekt <i>Unreal Engine</i> -a .....	37
Slika 41.	Prikaz <i>Event Graph</i> -a modela trupa zrakoplova u <i>Unreal Engine</i> -u .....	37
Slika 42.	Prikaz <i>Event Graph</i> -a <i>UxtUIElement</i> -a koje sadrži tipkalo kategorije <i>Switch</i> za prikaz zrakoplova u <i>Unreal Engine</i> -u .....	38
Slika 43.	Prikaz modifikacija komponente <i>UxtPalmUpConstraint</i> -a lijeve ruke u <i>Unreal Engine</i> -u .....	38
Slika 44.	Prikaz komponenata za dohvaćanje i spremanje lokacije korisnika u <i>Unreal Engine</i> -u .....	39
Slika 45.	Prikaz <i>Event Graph</i> -a <i>Uxt Pressable Switch Button Actor</i> -a za dohvaćanje i spremanje lokacije korisnika u <i>Unreal Engine</i> -u .....	39
Slika 46.	Prikaz <i>Event Graph</i> -a <i>Uxt Pressable Switch Button Actor</i> -a za stvaranje i uklanjanje zrakoplova u <i>Unreal Engine</i> -u.....	39
Slika 47.	Prikaz zrakoplova nakon pritiska na tipkalo unutar <i>Unreal Engine</i> -a .....	40
Slika 48.	Prikaz korisničkog sučelja za zrakoplov nakon pritiska tipkala unutar <i>Unreal Engine</i> -a.....	40
Slika 49.	Prikaz korisničkog sučelja za zrakoplov unutar <i>Viewport</i> -a u <i>Unreal Engine</i> -u ..	41
Slika 50.	Prikaz <i>Event Graph</i> -a tipkala <i>About Airplane</i> u <i>Unreal Engine</i> -u.....	41
Slika 51.	Prikaz <i>Event Graph</i> -a tipkala <i>Rotate</i> u <i>Unreal Engine</i> -u.....	42
Slika 52.	Prikaz sučelja <i>Rotate</i> za rotaciju zrakoplova nakon pokretanja sesije u <i>Unreal Engine</i> -u .....	43
Slika 53.	Prikaz <i>Event Graph</i> -a sučelja <i>Rotate</i> za rotaciju zrakoplova (dio rotacije oko negativne <i>x</i> osi) u <i>Unreal Engine</i> -u.....	43
Slika 54.	Prikaz sučelja <i>Move</i> za translaciju zrakoplova nakon pokretanja sesije u <i>Unreal Engine</i> -u .....	44
Slika 55.	Prikaz <i>Event Graph</i> -a sučelja <i>Move</i> za translaciju zrakoplova (dio translacije po negativnoj <i>x</i> osi) u <i>Unreal Engine</i> -u .....	44
Slika 56.	Prikaz sučelja <i>Scale</i> za skaliranje zrakoplova nakon pokretanja sesije u <i>Unreal Engine</i> -u .....	45
Slika 57.	Prikaz <i>Event Graph</i> -a sučelja <i>Scale</i> za skaliranje zrakoplova u <i>Unreal Engine</i> -u	45
Slika 58.	Prikaz sučelja oko zrakoplova nakon pokretanja sesije u <i>Unreal Engine</i> -u (1)....	46
Slika 59.	Prikaz sučelja oko zrakoplova nakon pokretanja sesije u <i>Unreal Engine</i> -u (2)....	46
Slika 60.	Prikaz <i>Event Graph</i> -a početnih postavka zrakoplova u <i>Unreal Engine</i> -u.....	47
Slika 61.	Prikaz <i>Event Graph</i> -a tipkala <i>Show Parts</i> u <i>Unreal Engine</i> -u.....	47
Slika 62.	Prikaz repa zrakoplova stvorenog u prostoru nakon pokretanja sesije u <i>Unreal Engine</i> -u .....	48
Slika 63.	Prikaz desnog krila zrakoplova stvorenog u prostoru nakon pokretanja sesije u <i>Unreal Engine</i> -u .....	48
Slika 64.	Prikaz modifikacije <i>Project Settings</i> prozora za kreiranje glasovne naredbe u <i>Unreal Engine</i> -u .....	49
Slika 65.	Prikaz dijela <i>Level Blueprint</i> skripte za omogućavanje glasovne naredbe u <i>Unreal Engine</i> -u .....	49
Slika 66.	Prikaz ispisa izvršene glasovne naredbe nakon pokretanja sesije u <i>Unreal Engine</i> -u .....	50
Slika 67.	Prikaz internetske stranice [literatura] kojoj se pristupilo nakon izvršene glasovne naredbe nakon pokretanja sesije u <i>Unreal Engine</i> -u.....	50
Slika 68.	Prikaz dodavanja imena projekta za pakiranje unutar <i>Project Settings</i> prozora u <i>Unreal Engine</i> -u .....	51

---

Slika 69.	Unos potrebnih informacija za <i>Company Name</i> te <i>Company Distinguished Name</i> unutar <i>Project Settings</i> prozora u <i>Unreal Engine</i> -u .....	51
Slika 70.	Označivanje opcije <i>Start in VR</i> unutar <i>Project Settings</i> prozora u <i>Unreal Engine</i> -u .....	52
Slika 71.	Modifikacija potrebnih informacija za <i>HoloLens</i> te <i>Signing Certificate</i> unutar <i>Project Settings</i> prozora u <i>Unreal Engine</i> -u .....	52
Slika 72.	Prikaz <i>Windows</i> ikonice <i>HoloLens</i> uređaja nakon pokazivanje unutarnjeg dijela zgloba .....	53
Slika 73.	Prikaz virtualnog tipkala <i>Airplane Visibility</i> u sustavu mješovite stvarnosti, nakon pokazivanja lijevog dlana kameri .....	54
Slika 74.	Prikaz virtualnog modela zrakoplova u sustavu mješovite stvarnosti, nakon klika na tipkalo <i>Airplane Visibility</i> .....	54
Slika 75.	Prikaz virtualnog tipkala <i>Menu Visibility</i> u sustavu mješovite stvarnosti, nakon pokazivanja desnog dlana kameri .....	55
Slika 76.	Prikaz virtualnog sučelja zrakoplova u sustavu mješovite stvarnosti nakon klika na tipkalo <i>Menu Visibility</i> te prozora za rotaciju, translaciju i skaliranje zrakoplova .....	55
Slika 77.	Prikaz virtualnih sučelja koji pokazuju dijelove zrakoplova u sustavu mješovite stvarnosti nakon klika na tipkalo <i>Show Parts</i> .....	56
Slika 78.	Prikaz virtualne internetske stranice u sustavu mješovite stvarnosti nakon klika na tipkalo <i>About Airplane</i> .....	56
Slika 79.	Prikaz virtualnog repa zrakoplova u sustavu mješovite stvarnosti nakon klika na tipkalo <i>Tail</i> .....	57

---

**POPIS TABLICA**

Tablica 1. Značajke VR sustava [16] .....	9
Tablica 2. Značajke AR sustava [16] .....	10
Tablica 3. Značajke MR sustava [16].....	11
Tablica 4. Usporedba AR i MR sustava [23] .....	11

---

## POPIS KRATICA

<b>Kratika</b>	<b>Puni naziv</b>
AR	<i>Augmented Reality</i>
HCI	<i>Human-Computer Interaction</i>
HMD	<i>Head Mounted Display</i>
HUD	<i>Head Up Display</i>
MR	<i>Mixed Reality</i>
SDK	<i>Software Development Kit</i>
VR	<i>Virtual Reality</i>

---

**SAŽETAK**

U sklopu ovog diplomskog rada opisan je koncept uranjajućih tehnologija (engl. *Immersive Technologies*), pri čemu je naglasak na sustavu mješovite stvarnosti, njezinoj primjeni, usporedbi s proširenom stvarnosti te opisom njenih komponenti. Rad također obuhvaća razvoj te detaljan opis *Unreal Engine* aplikacije namijenjene za mješovitu stvarnost, koja uključuje razne mogućnosti poput interakcije s virtualnim elementima, glasovnih naredbi te praćenja pozicije korisnika. U završnom je dijelu razvijeni projekt učitani na *HoloLens 2* uređaj te eksperimentalno evaluiran.

Ključne riječi: uranjajuće tehnologije, proširena stvarnost, mješovita stvarnost, virtualni element, *Unreal Engine*, *HoloLens 2*, eksperiment

---

**SUMMARY**

This master's thesis describes the concept of immersive technologies, with an emphasis on the mixed reality system, its application, comparison with augmented reality, and description of its components. The thesis also encompasses the development and detailed description of an Unreal Engine application designed for mixed reality, which includes various features such as interaction with virtual elements, voice commands, and user position tracking. In the final part, the developed project is loaded onto the HoloLens 2 device and experimentally evaluated.

Key words: immersive technologies, augmented reality, mixed reality, virtual element, Unreal Engine, HoloLens 2, experiment

## 1. UVOD

Cjelokupnim razvojem znanosti i tehnologije raste potreba za proučavanjem te unaprjeđivanjem sustava virtualne (engl. *Virtual Reality*, VR), a shodno tome i sustava proširene (engl. *Augmented Reality*, AR) i mješovite stvarnosti (engl. *Mixed Reality*, MR). Njihovo široko područje primjene uvelike može olakšati neke od svakodnevnih poslova koje obavlja pojedinac, ubrzati proces učenja, posljedično smanjiti vjerojatnosti za pogreškama u realnom svijetu te, na kraju krajeva, služiti za razonodu.

Industrija za proizvodnju videoigara bila je prva koja je popularizirala sustave virtualne stvarnosti, no oni se danas upotrebljavaju i u mnogim drugim sferama industrije. [1] Nakon sustava virtualne stvarnosti nastali su sustavi proširene stvarnosti te je, kombinacijom elemenata oba svijeta, stvorena mješovita stvarnost. Navedene vrste spadaju u skupinu uranjajućih tehnologija (engl. *Immersive Technologies*).

Virtualna stvarnost omogućuje interakciju korisnika s umjetnim trodimenzionalnim vizualnim ili drugim osjetilnim okruženjem, zahvaljujući uređajima i aplikacijama specifičnim za to područje. [2] Za razliku od virtualne stvarnosti koja stvara potpuno umjetno okruženje, u proširenoj stvarnosti dolazi do preklapanja digitalnih informacija s korisnikovom okolinom u stvarnome vremenu. [3] Korisnik s virtualnim elementima proširene stvarnosti može imati ograničenu interakciju, a sami elementi ne mogu imati interakciju s okolišem jer se u njemu ne nalaze. Stoga, mješovita stvarnost predstavlja najnapredniji oblik uranjajuće tehnologije, budući da su virtualni elementi u potpunosti integrirani u okruženje korisnika, što mu omogućuje realističniju interakciju s istima. Osim navedenog, posebna je po tome što virtualni objekti mogu imati interakciju s fizičkim okolišem u kojem su smješteni. [3]

U nastavku ovog rada naglasak je na primjeni i opisu komponenata sustava proširene i mješovite stvarnosti te postupku izrade i implementacije virtualnog trodimenzionalnog svijeta namijenjenog za *Microsoft*-ov uređaj *HoloLens 2*.

## 2. POVIJESNI PREGLED RAZVOJA URANJAJUĆIH TEHNOLOGIJA

Spekulacije o virtualnoj stvarnosti datiraju još iz prve polovice 20. stoljeća, no pravi razvoj započinje 1957. godine kada američki redatelj i kinematograf Morton Heilig na temelju svog snimateljskog iskustva razvija prve virtualne naočale *Sensorama*, čime je dobio titulu „oca virtualne stvarnosti“. Uređaj je nazvan stereoskopskim televizorom za individualnu uporabu te je bio prijenosan i poprilično glomazan, [4] a Heilig ga je nazvao i kinom „budućnosti“ jer je uključivao zvučnike, ventilatore, generatore mirisa i vibrirajući stolac koji gledatelja „uranjaju“ u film [Slika 1]. [6]



Slika 1. Prikaz prvih virtualnih naočala *Sensorama* [5]

U nastavku su prikazane neke od značajnijih godina kroz povijest, od 1960e do danas:

- 1960.: Morton Heilig izumio je telesfernu masku (engl. *Telesphere Mask*) koja je bila prvi primjer zaslona montiranog na glavu (engl. *Head Mounted Display*, HMD), a omogućavao je trodimenzionalne širokovidne stereoskopske slike, no bez mogućnosti praćenja pokreta. [7]
- 1961.: Inženjeri Charles Comeau i James Bryan, iz američke tvrtke za proizvodnju elektroničkih uređaja *Philco*, razvili su prvu preteču HMD-a kakvu poznajemo danas. Uređaj je sadržavao video ekran za svako oko kao i magnetski sustav za praćenje pokreta povezan s kamerom za video nadzor. Pokreti glave pomicali bi udaljenu kameru, dopuštajući tako korisniku da promatra prostor koji ga okružuje. Ovaj uređaj



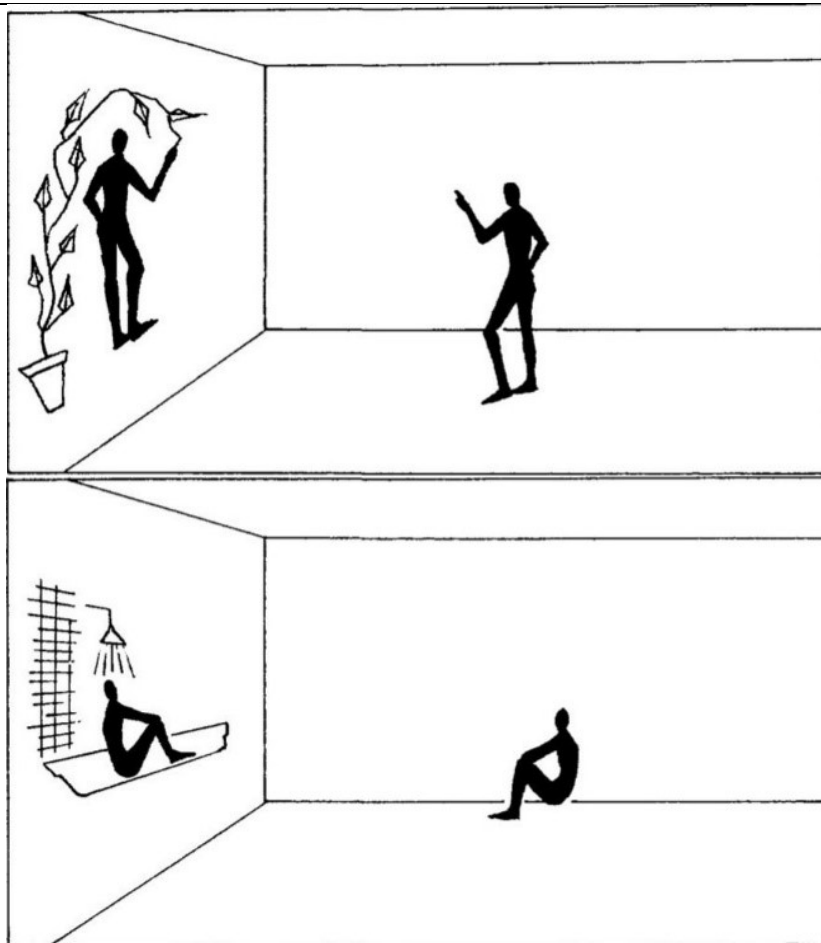
bio je prvi korak u evoluciji HMD-a za virtualnu stvarnost, no nedostajale su mu integracija računala te generiranje slike. [7]

- 1965.: Ivan Edward Sutherland, američki informatičar te pionir interneta i računalne grafike, opisao je koncept pod nazivom „*Ultimate Display*“ koji može simulirati stvarnost do točke neprepoznatljivosti između prave i simulirane. Njegov je rad postao temeljni nacrt za današnje koncepte koji obuhvaćaju virtualnu stvarnost. [7]
- 1968.: Ivan Edward Sutherland i njegov student Bob Sproull stvorili su prvi VR, odnosno AR zaslon za glavu (engl. *Headset*), pod nazivom „*Sword of Damocles*“. Uređaj je za razliku od prethodnih bio spojen na računalo, a ne na kameru, no bio je relativno primitivnog izgleda, budući da je visio o strop i bio veoma neergonomičan [Slika 2]. [7]



Slika 2. Prikaz prvog VR zaslona za glavu pod nazivom „*Sword of Damocles*“ [8]

- 1969.: Myron W. Krueger, američki računalni umjetnik, upotrijebio je naziv „umjetna stvarnost“, a obuhvaćala je računalno generirana okruženja s kojima su korisnici, koji su se nalazili u tom prostoru, mogli imati „interakciju“ [Slika 3]. [7]



Slika 3. Primjer jednog od niza projekata Myron W. Kruegera pod nazivom „Metaplay“ koji je doveo do razvoja projekta „Videoplace“ [9]

- 1975.: Projekt „Videoplace“ Myron W. Kruegera naširoko se smatra prvim interaktivnim sustavom virtualne stvarnosti. Korištenjem kombinacija računalne grafike, svjetlosne projekcije, kamera i zaslona mogao je mjeriti položaj korisnika. [7] Korisnici su mogli vidjeti kako njihove računalno generirane siluete oponašaju vlastite pokrete i radnje; pokreti korisnika snimani su kamerom te prenošeni na siluetu. Također, korisnici u preostalim sobama su mogli komunicirati sa siluetama drugih korisnika u istom virtualnom svijetu. Navedeno je potaknulo ideju da ljudi mogu komunicirati u virtualnom okruženju čak i ako nisu fizički blizu. [10]
- 1987.: Prvi puta se u stručnoj terminologiji upotrijebio naziv „virtualna stvarnost“. [7]
- 1989.: NASA je započela s korištenjem sustava virtualne stvarnosti te je, uz pomoć američke tehnološke tvrtke *Crystal River Engineering*, razvila projekt „VIEW“, odnosno simulaciju za uvježbavanje astronauta. U projektu su bile korištene i rukavice za

interakciju dodirrom, a isti se smatra modernim primjerom virtualne stvarnosti [Slika 4].

[7]



**Slika 4. Primjena naočala i rukavica u sklopu projekta „VIEW“ [7]**

- 1991.: Počeli su se pojavljivati uređaji za virtualnu stvarnost kojima je javnost imala pristup, iako je njihovo posjedovanje u kućanstvu još uvijek bilo daleko od dosega. Tvrtka *Virtuality* lansirala je niz arkadnih igara i strojeva s impresivnim stereoskopskim trodimenzionalnim vizualima, pri čemu su igrači koristili naočale. [7]
- 1992.: Prvi puta se u stručnoj terminologiji upotrijebio naziv „proširena stvarnost“. Iste se godine po prvi puta pojavio *Virtual Fixture*, prvi uranjajući sustav proširene stvarnosti s ciljem poboljšanja doživljaja. Sam termin odnosi se na digitalno stvoreni element koji se pojavljuje u stvarnom okruženju korisnika. [10]
- 1994.: Znanstvenici Paul Milgram i Fumio Kishino objavljuju rad pod nazivom „*A Taxonomy of Mixed Reality Visual Displays*“ gdje po prvi puta službeno upotrebljavaju termin „mješovita stvarnost“. [11]
- 1999.: Stvoren *ARToolKit*, biblioteka otvorenog koda koja služi za stvaranje aplikacija proširene stvarnosti. Trenutno se nalazi na *GitHub*-u. [10]

- 2007.: *Google* uvodi *Street View*, na način da je poboljšao svoju uslugu *Maps* koristeći slike od 360 stupnjeva na razini ulice, snimljene automobilima opremljenim za kameru. [7]
- 2010.: Nakon nekoliko, ne toliko uspješnih, pokušaja lansiranja proizvoda na tržište od strane kompanija poput američke *SEGA*-e te japanskog *Nintendo*-a, američki poduzetnik Palmer Luckey napravio je prototip *Oculus Rift*-a, prvog modernog VR HMD-a, koji se oslanjao na procesorsku snagu računala za isporuku slika [Slika 5]. Iste je godine *Google* uveo stereoskopski 3D pogled za *Street View*. [7]



**Slika 5. Prikaz modernog VR HMD-a *Oculus Rift* [12]**

- 2013.: Nakon relativno neuspješnog pokušaja iz 2011., *Google* proizvodi „pametne naočale“, tzv. *Google Glass* [Slika 6], te ih godinu dana nakon plasira na tržište. Zaslon ovih naočala korisnicima omogućuje prikaz digitalnih informacija u njihovom vidnom polju. [10]



**Slika 6. Prikaz *Google*-ovih „pametnih naočala“ [13]**

- 2014.: Bivša američka tvrtka *DAQRI*, koja se bavila sustavima proširene stvarnosti, proizvodi „pametnu kacigu“, tzv. *Smart Helmet*. To je bio novi stereoskopski uređaj za gledanje stvarnog okruženja s ugrađenim računalnim programom unutar kamere koji snima, obrađuje i prikazuje informacije o korisnikovoj okolini. [10] [14] Iste godine *Facebook* kupuje *Oculus VR* tvrtku, nakon čega je razvoj uređaja za virtualnu stvarnost znatno porastao te tvrtke poput *Sony*-a, *Google*-a i *Samsung*-a objavljuju svoje nove prototipe. [7]
- 2015.: *Microsoft*-ove naočale *Microsoft HoloLens* postaju dostupne javnosti [Slika 7]. To je bila prva oprema za proširenu i mješovitu stvarnost s ugrađenim *Windows 10* operativnim sustavom i mogućnošću mapiranja dubine. [10]



Slika 7. Prve *Microsoft*-ove AR/MR naočale *Microsoft HoloLens* [15]

- 2016.: Američka tehnološka tvrtka *Magic Leap* proizvodi *Magic Leap* naočale za proširenu stvarnost koje postaju konkurencija *Microsoft HoloLens*-u. [10]
- 2017.: Mnoge tvrtke su dinamičnije krenule s razvojem vlastitih HMD-a za virtualnu, proširenu i mješovitu stvarnost, uključujući *Google*, *Apple*, *Amazon*, *Sony*, *Samsung*, *Microsoft* itd. [7]
- 2019.: *Microsoft*-ove naočale *HoloLens 2* te *Facebook*-ov HMD za virtualnu stvarnost *Oculus Quest* postaju dostupni javnosti. [7][10]
- 2020. – danas: Nadograđuju se te javnosti postaju dostupne nove verzije HMD-a poput *Oculus 2*, *Pico Neo 3* HMD-a za virtualnu stvarnost koji konkurira *Facebook*-ovom *Oculus*-u te *Apple Vision Pro* tvrtke *Apple*. [7]

### 3. VIRTUALNA, PROŠIRENA I MJEŠOVITA STVARNOST

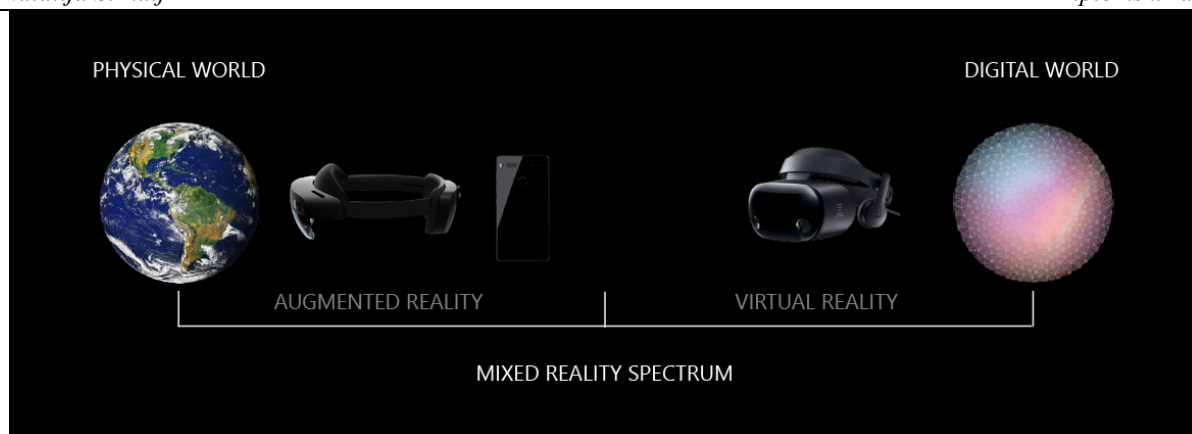
Glavna razlika uranjajućih sustava je u okruženju koje vidi korisnik te samoj interakciji korisnika s digitalnim objektima, kao i digitalnih objekata s okruženjem. U nastavku su navedene opće informacije i karakteristike sva tri svijeta te po čemu se razlikuje proširena u odnosu na mješovitu stvarnost.

#### 3.1. Općenito o sustavima virtualne, proširene i mješovite stvarnosti

Virtualna stvarnost produkt je korištenja računalnog modeliranja i simulacije koja korisnika uranja u potpuno umjetno generirano okruženje s kojim može ostvariti interakciju. [2] Virtualni svijet omogućuje prilagođavanje i stvaranje onoliko različitih iskustava i scenarija koliko je potrebno. [16]

Proširena stvarnost je „poboljšana“ izvedba stvarnog svijeta postignuta kroz, većinskim dijelom, digitalne vizualne elemente i zvukove putem holografske tehnologije. Proširena stvarnost uključuje tri značajke, a to su kombinacija digitalnog i fizičkog svijeta, interakcije koje se odvijaju u stvarnom vremenu te preciznu trodimenzionalnu identifikaciju stvarnih i virtualnih objekata. [17] Dakle, glavna prednost sustava proširene stvarnosti je njegova mogućnost spajanja digitalnih, trodimenzionalnih komponenti s percepcijom stvarnog svijeta pojedinca. [18]

Mješovita stvarnost ima sve navedene karakteristike kao i proširena, no za razliku od proširene u kojoj su digitalni objekti odvojeni od realnog okruženja, odnosno ne njega „preslikani“, mješovita stvarnost pruža dodatne mogućnosti poput potpune komunikacije digitalnih objekata s okolinom u kojoj se nalaze, budući da na neki način postaju njen sastavni dio [Slika 8]. Uz navedeno, korisnik može realistično manipulirati objektima zahvaljujući raznim sensorima koji prate pokrete ruku. Mješovita stvarnost temelji se na napretku u računalnom vidu, grafičkoj obradi, tehnologijama prikaza, sustavima unosa i računalnom oblaku. [19]



**Slika 8. Slikoviti prikaz raspona mješovite stvarnosti kao kombinacije proširene i virtualne [19]**

Danas sustavi mješovite stvarnosti uključuju sljedeće: [19]

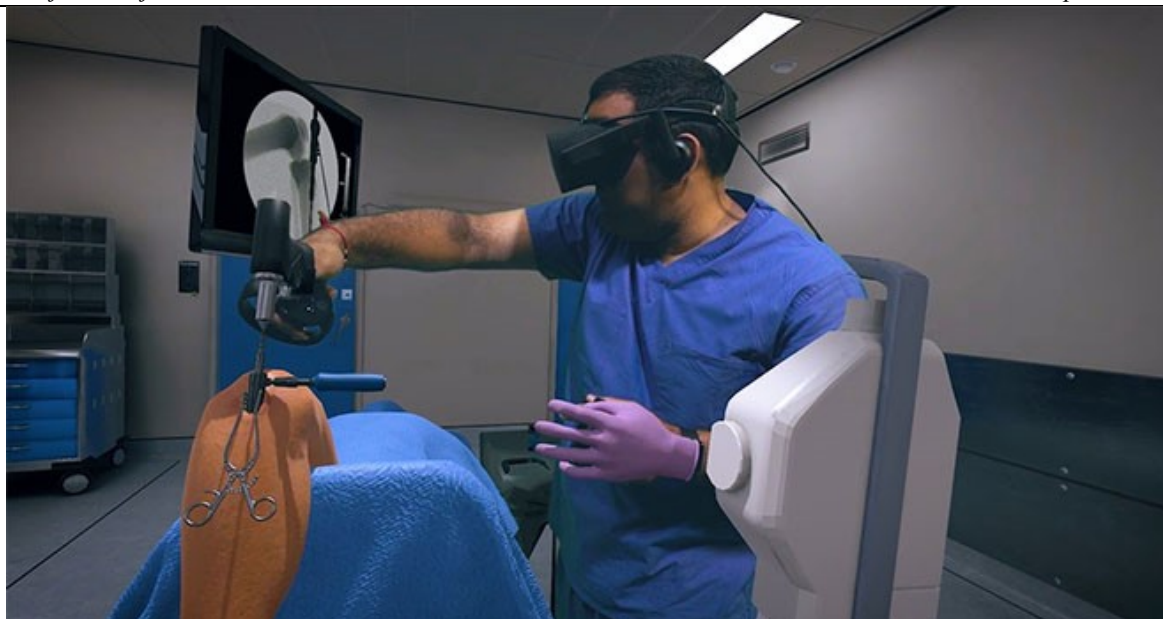
- Razumijevanje okoline koje uključuje mapiranje prostora (engl. *spatial mapping*) te pozicioniranje digitalnih 3D objekata na točno određenu poziciju u prostoru (engl. *anchors*).
- Praćenje pokreta ruku, očiju te razumijevanje glasovnih naredba korisnika.
- Prostorni zvuk (engl. *spatial sound*), odnosno stvaranje iluzije zvukova koji dolaze iz različitih smjerova i udaljenosti kako bi korisnik imao što realističniji doživljaj.
- Mogućnost kolaboracije glede digitalnih objekata.

### 3.2. Značajke sustava virtualne, proširene i mješovite stvarnosti

U nastavku su tablično prikazane neke od pozitivnih i negativnih značajki VR [Tablica 1], AR [Tablica 2] te MR [Tablica 3] sustava kao i usporedba AR i MR sustava [Tablica 4].

**Tablica 1. Značajke VR sustava [16]**

Pozitivne značajke VR sustava	Negativne značajke VR sustava
Potpuna fleksibilnost glede stvaranja digitalnog iskustva	Iskustvo je „izolirano“, odnosno odvojeno od stvarnog okoliša i ljudi
Potpuno iskustvo „uranjanja“	
Stvaranje beskrajnih virtualnih scenarija i iskustava [Slika 9]	



Slika 9. Primjer uporabe sustava virtualne stvarnosti putem HMD-a [20]

Tablica 2. Značajke AR sustava [16]

Pozitivne značajke AR sustava	Negativne značajke AR sustava
Prenosivi i najčešće bežični	Elementi djeluju holografski i nerealno
Manje cijene u odnosu na MR sustave	Otežan rad s velikim objektima uslijed uskog vidnog polja
Mogućnost interakcije korisnika s realnim okruženjem	Elementi nisu sastavni dio korisnikovog realnog okruženja
Izvrstan za prikazivanje jednostavnog sadržaja poput „slojeva informacija“ („preslika“ digitalnih elemenata u realno okruženje) [Slika 10]	



Slika 10. Primjer uporabe sustava proširene stvarnosti putem pametnog telefona [21]



Tablica 3. Značajke MR sustava [16]

Pozitivne značajke MR sustava	Negativne značajke MR sustava
Digitalni elementi su sastavni dio korisnikovog okruženja	Nešto veći i teži uređaji
Najbolje odgovara simulacijama koje trebaju odražavati stvarne scenarije	Potrebno je namjensko okruženje za korištenje
Pružuje potpunu fleksibilnost virtualnog svijeta uz pouzdanost stvarnog [Slika 11]	
Neograničena računalna sposobnost	



Slika 11. Primjer uporabe sustava mješovite stvarnosti putem HMD-a [22]

Tablica 4. Usporedba AR i MR sustava [23]

Aspekt	AR sustavi	MR sustavi
Interakcija	Obično nema izravne interakcije između komponenti fizičkog i digitalnog svijeta	Sposobnost korisnika da komunicira s digitalnim zaslonom te manipulira i ima interakciju s virtualnim objektima i informacijama

Uređaji	Obično dostupni putem pametnog telefona, bez potrebe za dodatnom opremom	Zahtijeva HMD s prozirnomo lećom ili kamerom kako bi se stvorilo bolje iskustvo uranjanja
Uporaba	Širok raspon primjene uključujući igre, obrazovanje, dizajn i zdravstvena skrb; često se koristi za prikaz informativnog sadržaja na objekte iz stvarnog svijeta	Podiže uporabu AR sustava na novu razinu omogućujući interaktivnija i sveobuhvatnija iskustva

## 4. KOMPONENTE SUSTAVA MJEŠOVITE STVARNOSTI

Mješovita i proširena stvarnost idu rame uz rame te svaki uređaj za mješovitu stvarnost može biti i uređaj za proširenu stvarnost. Proces ovih sustava može se podijeliti u tri koraka: [24]

- Ulaz: senzori prikupljaju informacije iz realnog svijeta.
- Obrada: procesor interpretira podatke prikupljene pomoću različitih hardverskih i softverskih izvora.
- Izlaz: prikaz informacija koji korisniku pruža osjećaj kao da su virtualni objekti dio stvarnog svijeta.

Komponente sustava proširene i mješovite stvarnosti dijele se na hardverske i softverske [24], a neki od izvora navode kako je i korisnik sastavni dio sustava.

### 4.1. Hardverske komponente

Ulazne hardverske komponente čine različiti tipovi senzora i kamera koji reagiraju na fizičke ili kemijske podražaje iz stvarnog okruženja te pružaju potrebne informacije za razvoj sustava. Senzori mogu biti optički, magnetski, inercijski te ostali tipovi, a u nastavku su navedeni neki od primjera iz svake grupe: [24]

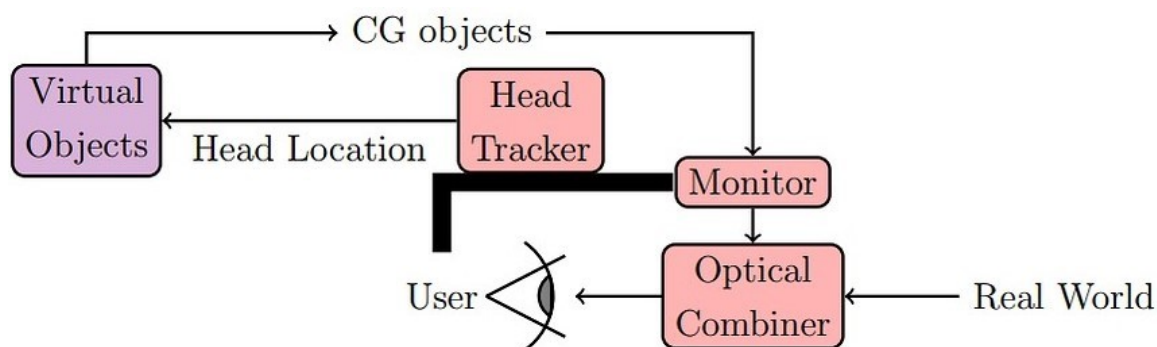
- optički senzori – infra-crveno,
- magnetski senzori – kompas,
- inercijski senzori – akcelerometri, žiroskopi,
- ostali senzori – GPS, dubinski.

Kako bi se informacije mogle obraditi te prikazati kao realističan 3D sadržaj, potrebni su snažni procesori te jedinice za grafičku obradu. [23]

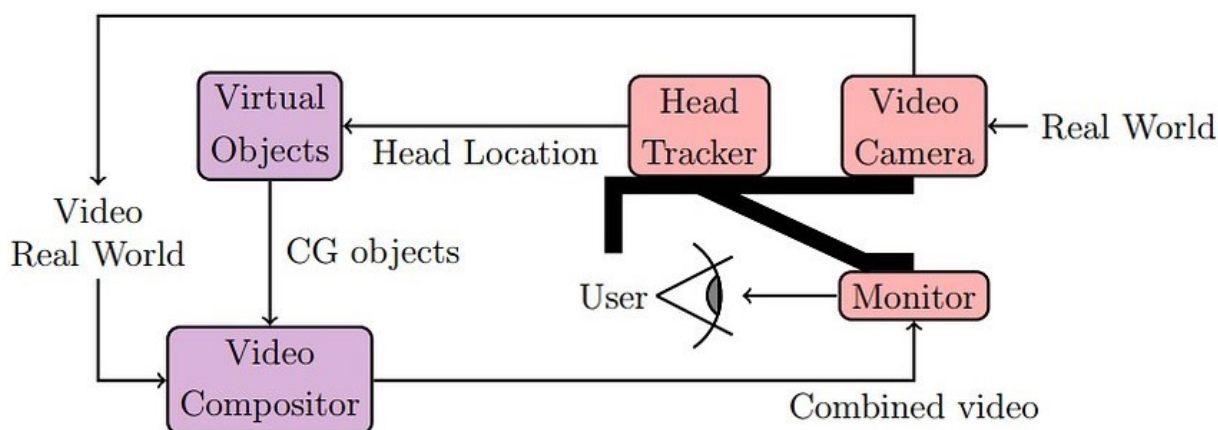
Izlaznu hardversku komponentu čini zaslon za prikaz informacija koji se može podijeliti na nosivi i nenosivi, pri čemu su za sustave mješovite stvarnosti adekvatniji nosivi kako bi korisnik mogao ostvariti bolju interakciju s virtualnim objektima. Osim spomenutih kategorija, zaslone se mogu podijeliti na optičke i video zaslone (uglavnom namijenjene za proširenu stvarnost) te projekcije kao što je navedeno u nastavku: [24]

- optički zaslone – kacige, naočale,
- video zaslone – *Head-Up Display* (HUD), pametni telefoni, tableti, PC-evi,
- projekcije – laserski projektori.

Kod prozirnih optičkih zaslona, virtualni sadržaji se projiciraju na sučelje kako bi se optički „pomiješali“ sa stvarnim okruženjem [Slika 12]. Kod video zaslona upotrebljavaju se dva mehanizma, pri čemu jedan koristi HMD uređaje, a drugi kameru i zaslon uređaja poput pametnih telefona i tableta [Slika 13]. Jedan od uređaja koji koristi proziran optički zaslon je *Microsoft HoloLens*. [24]



Slika 12. Prozirni optički zaslon AR i MR sustava [24]



Slika 13. Video zaslon AR i MR sustava [24]

#### 4.1.1. Opis glavnih senzora smještenih na uređaje za mješovitu stvarnost

Četiri su glavna senzora koja se nalaze na naočalama ili kacigama, a to su žiroskopi, akcelerometri, magnetometri i senzori dubine. Uz senzore, ključne su i kamere. [25]

Žiroskopi mjere kutnu brzinu objekta u više različitih smjerova. U MR uređajima koriste se za praćenje pokreta, odnosno izračun rotacije glave korisnika na temelju primijenjenog okretnog momenta. [25]

Glavna uloga akcelerometra je praćenje nagiba i rotacije glave korisnika kako bi se virtualna scena mogla pomicati u skladu s time. Neki uređaji koriste ovaj tip senzora za jednostavne geste poput kimanja glavom kako bi se ostvarila interakcija s virtualnim objektima. Uz navedeno,

akcelerometri surađuju sa žiroskopima i magnetometrima kako bi kalibrirali interni sustav praćenja uređaja te stabilizirali virtualnu scenu, sprječavajući podrhtavanje i dezorijentaciju. [25]

Magnetometri primarno služe za točnost praćenja glave, odnosno poboljšavaju preciznost praćenja glave koju postižu akcelerometri. Dok akcelerometri daju informacije o tome koliko se glava korisnika pomaknula, magnetometri određuju točan smjer, eliminirajući dvosmislenosti i sprječavajući nepotrebna pomicanja u virtualnoj sceni. Magnetometri korigiraju skretanje prateći rotacije glave oko okomite osi i osiguravajući da virtualni svijet neprimjetno prati pokrete u stvarnom. U nekim naprednim MR uređajima, magnetometri mogu pomoći u mapiranju realnog okoliša na način da lociraju željezne objekte. Ovakav tip senzora može naići na probleme ukoliko se nađe u području jakih magnetskih polja zbog potencijalnih interferencija. [25]

Dubinski senzori mjere udaljenost između korisnika i virtualnih objekata u prostorijski stvaraju 3D mapu prostorijske. Tip dubinskih senzora *Time-of-Flight* (ToF) mjeri vrijeme koje je potrebno svjetlosnom pulsu da pogodi objekt i od njega se odbije. Sastoji se od emitera za projiciranje, obično infracrvene, svjetlosti te prijemnika za hvatanje reflektirane svjetlosti. Senzor zatim, na temelju izmjerenog vremena koje je potrebno svjetlosti da prođe od emitera do točke refleksije i natrag, izračunava udaljenost od uređaja kojeg korisnik nosi na glavi do objekta. [26]

Postoje dvije vrste kamere u MR uređajima, a to su tzv. *Pass-through* kamere te *Inside-out* kamere za praćenje. *Pass-through* kamere su okrenute prema vanjskoj strani te snimaju stvarno okruženje, a *Inside-out* kamere su okrenute prema korisniku te prate njegove pokrete. Podaci dobiveni *Inside-out* kamerama se koriste zajedno s podacima dobivenima od senzora. [25]

## 4.2. Softverske komponente

Glavna karakteristika softverskih komponenti je interpretacija prikupljenih podataka. Dije se na one visoke razine (engl. *High-level*) te niske razine (*Low-level*) [24]. Kod softvera visoke razine, neki od poznatijih alata koji pomažu u razvoju programa su *Vuforia*, *ARKit*, *Wikitude*, *ARToolKit*, *EasyAR*, *LayAR*, *OpenXR*, *Unity* i *Unreal Engine*, a spadaju u kategoriju poznatiju kao *Software Development Kit* (SDK). Softveri niske razine sastoje se od različitih područja koja se mogu razlikovati ovisno o potrebama programa, a to su programske biblioteke, računalni vid, računalna grafika, obrada slike, interakcija čovjeka i računala (engl. *Human-Computer Interaction*, HCI) itd. [24] Većina sustava proširene i mješovite stvarnosti zahtijeva

i umjetnu inteligenciju koja, primjerice, omogućuje izvršavanje radnji putem glasovnih naredbi ili pomaže u obradi informacija. [17]

#### 4.2.1. *Unreal Engine*

*Unreal Engine* je alat za razvoj videoigara i softvera tvrtke *Epic Games*. Ovaj alat omogućuje programerima izradu simulacija, uređivanje videozapisa ili zvuka te renderiranje animacija, a razvojni programeri koristili su ga za izradu nekih od najpopularnijih videoigara na današnjem tržištu. Jedna od glavnih prednosti je ta da je besplatan i prilagođen korisniku. Kao programski jezik za razvoj aplikacija unutar ovog alata koristi se C++, budući da pruža stabilnost te izvrsnu raspodjelu memorije. [27] *Unreal Engine* je kreiran 90ih godina prošlog stoljeća, no danas je javnosti već dostupna verzija 5.4., a stare su verzije razvojem tehnologije zamrle. Najbolji pokazatelj brzine razvoja su razni dodaci (engl. *plugins*) te alati (engl. *toolkits*) za *Unreal Engine* koji više nisu dostupni ni za neke verzije 4. Osim razvoja videoigara koristi se i za razvoj 3D svijeta proširene i mješovite stvarnosti, zahvaljujući specijaliziranim dodacima i alatima, te je prilagođen *HoloLens 2* HMD-u.

Proces instalacije *Unreal Engine*-a te postupak izrade aplikacije opisani su u nastavku ovog rada.

## 5. UREĐAJI ZA MJEŠOVITU STVARNOST

Glavni uređaji za mješovitu stvarnost su naočale i kacige, odnosno zasloni za glavu (*Headsets*, tj. *Head Mounted Displays*). Zaslone za glavu su ključni u pružanju uranjajućih iskustava mješovite stvarnosti jer projiciraju vizualni sadržaj izravno u korisnikov vidokrug. S HMD-om mogu se povezati i igraće konzole, no taj je segment još uvijek u fazi razvoja. Korištenje pametnih telefona i tableta se većinskim dijelom koristi za sustave proširene stvarnosti jer ne mogu pružiti pravi osjećaj dubine koji je potreban u sustavima mješovite stvarnosti. [23]

Postoje dvije kategorije HMD-ova za mješovitu stvarnost, a to su privezani (engl. *tethered*) i nevezani (engl. *untethered*). [28]

Privezani HMD-ovi izravno su povezani s računalom ili igraćom konzolom te se za generiranje MR iskustva oslanjaju na računalnu snagu povezanog uređaja. Ovakav tip obično nudi kvalitetniju grafiku i šire vidno polje, no mana im je manja mobilnost jer zahtijevaju kabelsku vezu s potrebnom operativnom platformom. Nevezani HMD-ovi su samostalni uređaji te za rad ne zahtijevaju računalo ili igraću konzolu. Oslanjaju se na interni procesor te senzore koji se nalaze na samom uređaju. Ovakav tip je mobilniji i praktičniji za korištenje, no nudi grafiku niže kvalitete te užu vidno polje zbog manje snage od onih koji su vezani na računalo. [28]

U nastavku su navedene karakteristike *Microsoft HoloLens 2* te usporedbe preostalih, prema [28] popularnijih HMD-ova današnjice.

### 5.1. *Microsoft HoloLens 2*

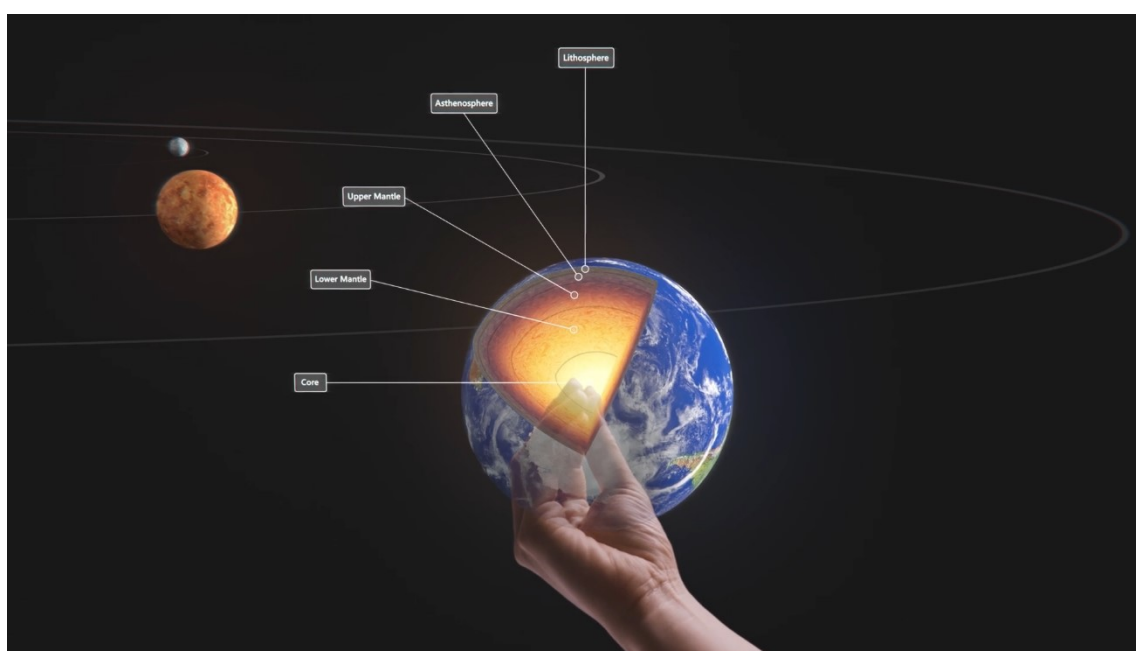
*Microsoft HoloLens 2* [Slika 14] je samostalan HMD dizajniran za poslovnu i industrijsku primjenu. Uključuje i paket softvera i alata usmjerenih na poduzeća poput *Dynamics 365*, *Guides* te *Remote Assist*.

*HoloLens 2* radi na *Windows Holographic* operativnom sustavu temeljenom na *Windows*-ima 10. Vidno polje ovih naočala je relativno usko i iznosi 52 stupnja. Ima mogućnost praćenja pokreta ruku [Slika 15] i očiju korisnika [Slika 16], sposobnost prepoznavanja glasa te ne zahtijeva korištenje kontrolera. Uz navedene sposobnosti razumijevanja pokreta i govora čovjeka, ima sposobnost razumijevanja okoliša, odnosno praćenje položaja u svih šest stupnjeva slobode, mogućnost mapiranja prostora [Slika 17] te slikanje i snimanje sustava mješovite stvarnosti. [31] Brzina osvježavanja (engl. *refresh rate*) iznosi 90 Hz. [28] Vanjske komponente od kojih se sastoji su vizir, traka za glavu, gumbi za svjetlinu i glasnoću, gumb za paljenje i gašenje te USB-C priključak, a na prednjoj se strani pri vrhu uređaja nalaze senzori

(akcelerometar, žiroskop, magnetometar te optički infracrveni) i kamere (vidljivog svjetla, dubinske) [Slika 14]. [31][32] U pakiranju dolazi zajedno s jastučićem za prednji dio glave, remenom za dodatnu fiksaciju na glavi te USB-C punjačem i kabelom. [31] Mane su mu relativno visoka cijena te zamor korisnika prouzročen težinom uređaja, nakon duljeg perioda nošenja. [28]



Slika 14. *Microsoft HoloLens 2* [29]

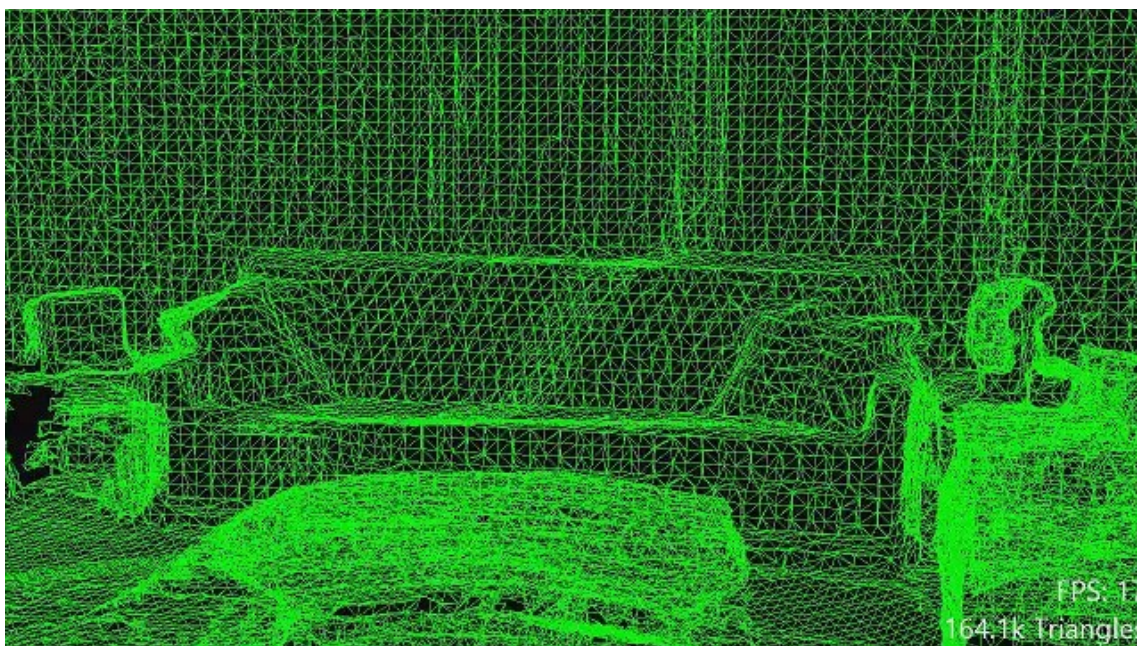


Slika 15. *Microsoft HoloLens 2* – mogućnost praćenja ruku [30]





Slika 16. *Microsoft HoloLens 2* – mogućnost praćenja očiju [30]



Slika 17. *Microsoft HoloLens 2* – mogućnost mapiranja prostora [30]

## 5.2. Preostali popularniji HMD-ovi današnjice [28]

1. *Magic Leap 2* [Slika 18] je tip HMD-a koji je povezan s nosivim i prijenosnim *Compute Pack*-om koji, između ostalog, smanjuje težinu te odvodi toplinu.

Vidno polje iznosi 70 stupnjeva te ima mogućnost praćenja pokreta ruku i očiju korisnika kao i sposobnost prepoznavanja govora. Ovaj HMD dolazi u paketu s

haptičkim kontrolerom pokreta. Brzina osvježavanja iznosi 120 Hz te je relativno lagan i ugodan za korištenje. Mane su mu relativno visoka cijena te slabija rezolucija od ostalih HMD-ova.



Slika 18. *Magic Leap 2* [33]

2. *Meta Quest Pro* [Slika 19] je samostalan HMD za virtualnu stvarnost koji također nudi funkcionalnost mješovite stvarnosti te je prvenstveno osmišljen u svrhu zabave.

Vidno polje iznosi 105 stupnjeva, ima mogućnost praćenja pokreta ruku i očiju korisnika te sposobnost prepoznavanja govora, iako jedino na engleskom te jedino u području SAD-a. Dolazi u paketu s haptičkim kontrolerom pokreta. Brzina osvježavanja iznosi 90 Hz. Iako je cjenovno povoljniji od ostalih HMD-ova, teži je te ima probleme sa zaslonom zbog pojave distorzije okoliša, slabiju rezoluciju u sustavu mješovite stvarnosti te zahtijeva *Meta* račun kako bi se mogao upotrebljavati.



Slika 19. *Meta Quest Pro* [34]

3. *Pico 4* [Slika 20] je samostalan HMD za virtualnu stvarnost koji poput *Meta Quest Pro* nudi iskustvo mješovite stvarnosti.

Vidno polje iznosi 105 stupnjeva, ima mogućnost praćenja pokreta ruku i očiju korisnika, no praćenje pokreta ruku je nešto ograničenije nego kod ostalih HMD-ova. Prepoznavanje govora nije ugrađeno u HMD, ali se može ugraditi u prilagođene aplikacije. Dolazi u paketu s haptičkim kontrolerom pokreta. Brzina osvježavanja iznosi 90 Hz. Najjeftiniji je od navedenih HMD-ova te je ugodan za korištenje. Kao i kod *Meta Quest Pro*, postoje problemi sa zaslonom zbog pojave distorzije okoliša te ima slabiju rezoluciju u sustavu mješovite stvarnosti, a uz navedeno i ograničenu podršku od strane programera.



Slika 20. *Pico 4* [35]

Uz navedene uređaje postoji još mnogo popularnih proizvođača HMD-ova za mješovitu stvarnost poput *Samsung*-ovog *Odyssey +*, *Apple*-ovog *Apple Vision Pro* koji je na tržište stupio u veljači 2024. godine, *Varjo* tvrtke *Varjo Technologies* itd.

### 5.3. Primjena *Microsoft HoloLens 2* uređaja

Prvotna ideja uporabe *Microsoft HoloLens*-a je njegova primjena u industriji te rad na daljinu, pri čemu biiskusni inženjeri davali korisne savjete zaposlenicima [Slika 21].



**Slika 21. Primjena *HoloLens 2* uređaja u proizvodnji te radu na daljinu [36]**

Široko područje primjene ovog uređaja navedeno je u nastavku. [37]

- Zdravstvena njega: obuka medicinskih stručnjaka te pomoć pri operacijama.
- Obrazovanje: stvaranje uranjajućih iskustava te mogućnost praktičnog učenja.
- Proizvodnja: pomoć u raznim zadacima poput dizajna proizvoda, montiranja i održavanja [Slika 22].
- Arhitektura i građevinarstvo: pomoć pri dizajnu i vizualizaciji.
- Udaljena suradnja (engl. *Remote Colaboration*): omogućavanje suradnje različitih timova u istom virtualnom okruženju.
- Maloprodaja: pružanje trgovcima jednostavan pristup o podacima i modelima njihovih proizvoda dok se nalaze u trgovini.
- Razonoda.



Slika 22. Primjena *HoloLens 2* uređaja u proizvodnji [36]

## 6. RAZVOJ 3D SVIJETA ZA MJEŠOVITU STVARNOST

Glavni dio ovog rada bio je razviti 3D svijet mješovite stvarnosti koji uključuje ciljane objekte od interesa, načiniti sučelje te povezati razvijeno okruženje s *HoloLens* HMD-om. Razvoj aplikacije vršio se u već spomenutom softveru *Unreal Engine*, a postupak je detaljno razrađen po koracima.

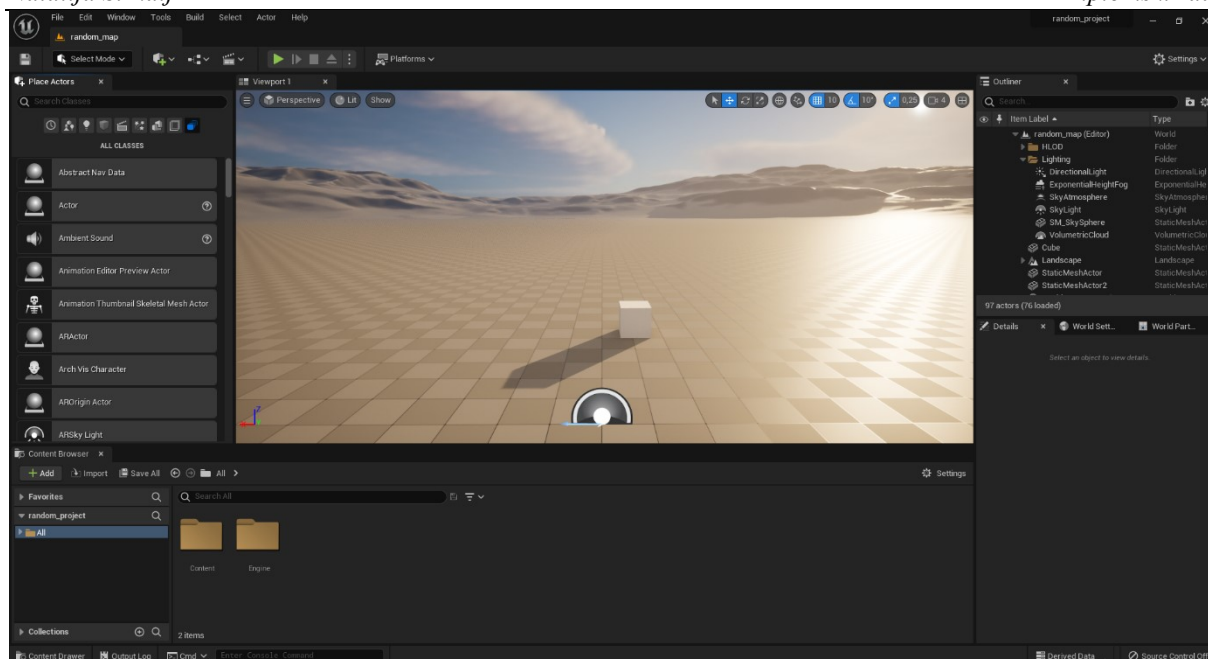
Prije same instalacije, pokretanja te izrade aplikacije unutar ovog programa, potrebno je razumjeti neke njegove osnovne karakteristike opisane u nastavku.

### 6.1. Osnovne informacije o *Unreal Engine*-u

Prilikom pokretanja *Unreal Engine*-a potrebno je stvoriti vlastiti projekt te u prozoru *Unreal Project Browser*, pod dijelom *Project Defaults*, odabrati postavke s kojima se želi raditi. U istom se prozoru s lijeve strane nalaze predlošci koji su inicijalno različiti po nekim postavkama glede unaprijed napisanog koda, kamera, osvjetljenja te aktivnih dodataka.

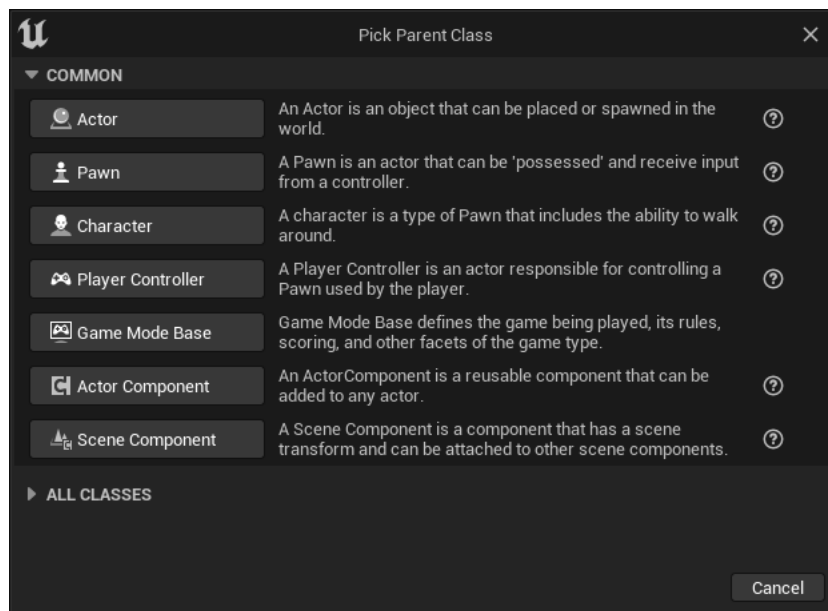
Kada se otvori novi projekt [Slika 23], pojavi se korisničko sučelje u čijem se središtu nalazi prozor za prikaz svega što želimo vidjeti u našem projektu (engl. *Viewport*), bilo da se radi o videoigri, animaciji ili simulaciji. *Viewport* se sastoji od razina (engl. *Levels*), a svaka razina unutar istog projekta može imati potpuno isto ili različito okruženje. Dakle, *Viewport* prikazuje sve što se nalazi na određenoj razini. Nazivi elemenata te karakteristike svakog koji se trenutno nalazi u razini koju kreiramo, prikazani su u prozoru desno koji se naziva *Outliner*. U *Outliner*-u možemo manipulirati nekim osnovnim karakteristikama elementa poput lokacije, rotacije, veličine, vidljivosti, materijala od kojeg se sastoji, je li element pokretljiv ili ne, želimo li primijeniti zakone fizike te mnogim drugim [Slika 23]. U *Content Browser*-u koji je smješten na dnu sučelja [Slika 23], nalaze se, najjednostavnijim riječima, svi programski kodovi za manipulaciju elementima ili bilo kakvim radnjama unutar *Viewporta* te svi elementi koje smo uvezli u program. Primjerice, ukoliko želimo uvesti model zrakoplova, isti će se pojaviti u dijelu *Content Browser*, no, ukoliko ga još nismo smjestili u *Viewport*, neće se pojaviti u dijelu *Outliner*.

Kretanje unutar *Viewporta* moguće je korištenjem tipki W,A,S,D na tipkovnici u kombinaciji s pritisnutom lijevom ili desnom tipkom miša.



Slika 23. Prikaz *Viewport*-a (sredina), *Outliner*-a (desno) te *Content Browser*-a (dolje) *Unreal Engine*-a

Klase elemenata koje možemo smjestiti na svaku razinu mogu biti tipa *Actor*, *Character*, *Pawn*, *Player Controller*, *Game Mode Base*, *Actor Component* te *Scene Component*. [Slika 24]



Slika 24. Različite klase elemenata *Unreal Engine*-a

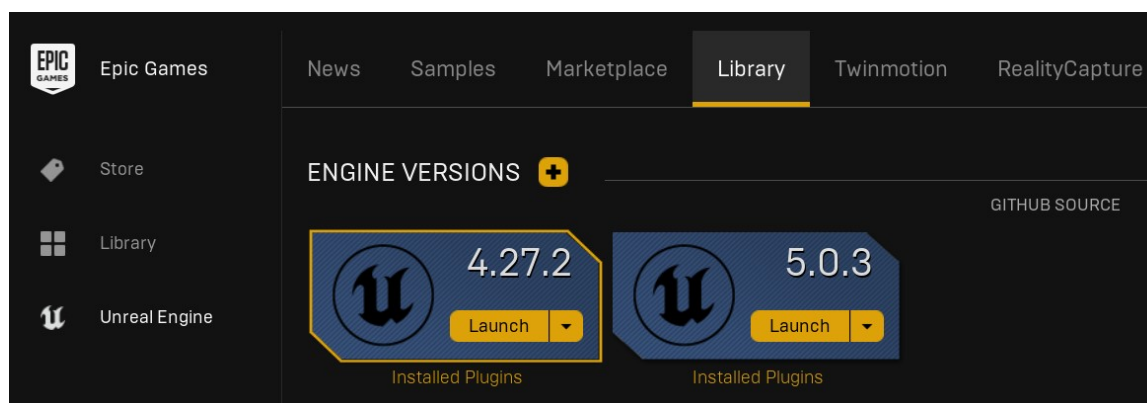
U svrhu ovog rada potrebne su klase *Actor*, *Pawn* te *Player Controller*. Klasa *Actor* je svaki objekt koji može biti inicijalno smješten u prostor ili stvoren programskim kodom. Svaki *Actor* može imati roditelj-dijete (engl. *Parent-Child*) odnos s drugim *Actor*-om te radnje koje utječu

na roditelja mogu utjecati i na dijete. Klasa *Pawn* je svaki *Actor* kojeg možemo „posjedovati“ te njime upravljati putem kontrolera. Klasa *Player Controller* je također *Actor* koji je odgovoran za manipulaciju elementa klase *Pawn*. Primjerice, ukoliko bi kreirali videoigru, osoba koja upravlja likom unutar videoigre je klasa *Player Controller*, lik je klasa *Pawn*, a svi ostali elementi s kojima lik može stupiti u interakciju su klasa *Actor*.

Svaka od spomenutih klasa može se programirati pomoću C++ programskog jezika ili *Blueprints*-a. *Blueprint Visual Scripting* je sustav za stvaranje skripti temeljen na konceptu sučelja zasnovanog na čvorovima za stvaranje elemenata, a upotrebljava se za definiranje objektno orijentiranih klasa ili objekata. Sustav je prilagođen korisniku te olakšava proces kreiranja programa u odnosu na pisanje C++ linija koda, no C++ je znatno brži te je prijeko potreban ukoliko su u pitanju neke zahtjevnije radnje. Svaki C++ kod može se proširiti *Blueprint* skriptom. *Blueprint Class* može se kreirati klikom na desnu tipku miša, klikom na ikonicu koja se nalazi na gornjoj alatnoj traci te klikom na ikonicu koja se nalazi unutar prozora *Details* (otvara se klikom na element u prozoru *Outliner*). Osim *Blueprint* klase koju može posjedovati svaki element zasebno, postoji i *Level Blueprint* čija skripta vrijedi za sve elemente koji se nalaze na trenutnoj razini. Nakon svake izmjene unutar *Blueprints*-a potrebno je kliknuti *Compile*.

## 6.2. Instalacija Unreal Engine-a

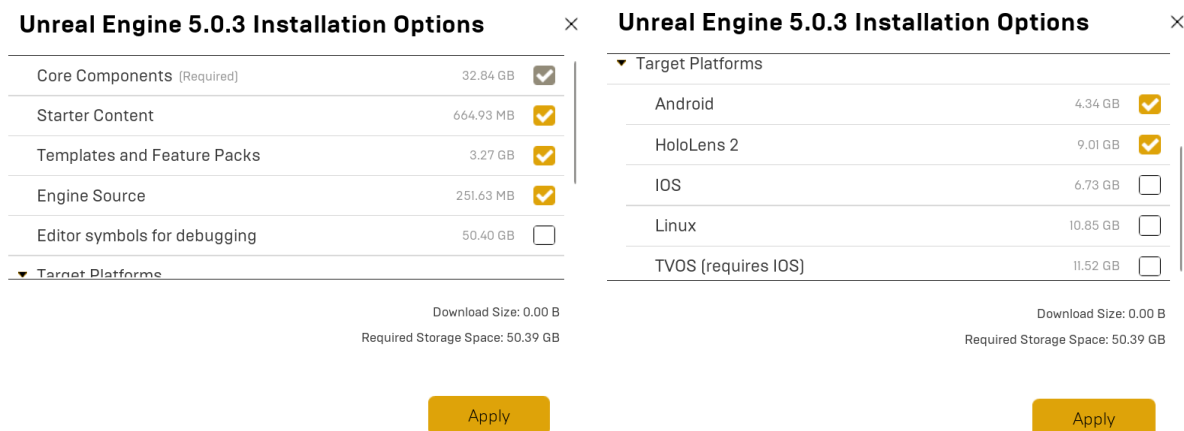
Prvi korak je instaliranje i pokretanje *Unreal Engine* softvera koji se može preuzeti s *Epic Games* trgovine [Slika 25]. Potrebno je pratiti razvoj softvera, kako se ne bi koristila prestara verzija zbog mogućnosti pojave greške pri pakiranju aplikacije za *HoloLens* uređaj, te prije same instalacije provjeriti podržavaju li potrebni dodaci željenu verziju *Unreal Engine*-a. U svrhu ovog rada upotrijebila se verzija 5.0.3.



Slika 25. Prikaz instaliranih verzija *Unreal Engine*-a



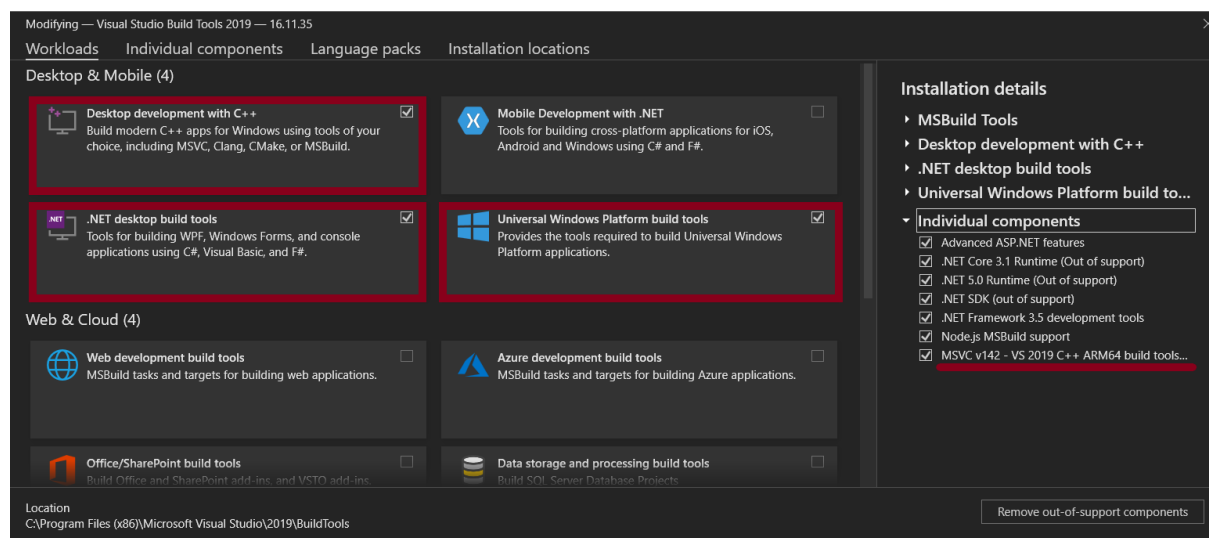
Ako se radi u operativnom sustavu *Windows*, u postavkama instalacije potrebno je odabrati opcije kako prikazuje slika [Slika 26].



Slika 26. Postavke instalacije *Unreal Engine-a*

### 6.3. Instalacija *Visual Studio Installer-a* te *Visual Studio Code-a*

Uz *Unreal Engine* potrebno je instalirati *Visual Studio Installer* te *Visual Studio Code*. Za verziju 5.0.3 poželjna je instalacija *Visual Studio 2022*, no za određene namjene može raditi i s verzijom *Visual Studio 2019*. *Unreal Engine* dizajniran je za interakciju s *Visual Studio-m*, omogućujući da se brzo i jednostavno naprave izmjene koda u projektima te vide rezultati nakon kompajliranja. Specifično za svrhe ovog diplomskog rada, potrebno je modificirati *Visual Studio Build Tools* unutar *Visual Studio Installer-a* kako bi se mogao stvoriti projekt u kojem će se dizajnirati svijet za mješovitu stvarnost te kako bi se projekt mogao spakirati za učitavanje na *HoloLens* uređaj [Slika 27].



Slika 27. Modifikacije *Visual Studio Build Tools-a*

### **6.3.1. Javljanje pogrešaka i upozorenja prilikom kreiranja projekta te pakiranja za HoloLens uređaj**

Osim slikom [Slika 27] prikazanih modifikacija, postoji mogućnost da se uslijed korištenja novijih verzija *Unreal Engine*-a moraju instalirati još neki od potrebnih alata unutar *Visual Studio Build Tools*-a. Dodatne alate je najbolje instalirati ukoliko se pojave pogreške (engl. *errors*) prilikom učitavanja projekta ili pakiranja istog za *HoloLens*. Problemi najčešće nastaju prilikom pakiranja, a potencijalne se greške ili upozorenja mogu javljati ovisno o ažuriranjima softvera, alata s kojima se radi, verziji *Unreal Engine*-a itd. Najčeći problemi se javljaju s alatima poput *Windows 10 SDK*, *.NET SDK*, *.NET Frameworks* i *NET Core Runtime* pa treba pripremiti koje su od verzija *Unreal Engine*-a programu potrebne za rad i pakiranje projekta. Uz navedeno, prilikom pakiranja projekta za *HoloLens*, često dolazi do situacija u kojima program pokušava pristupiti nekim datotekama koje su nebitne za rad, stoga ih, na neki način treba ukloniti. Uklanjanje „nepotrebnih“ datoteka dolazi s iskustvom te se radnja izvodi na vlastitu odgovornost. S druge strane, mogu se dogoditi i pogreške prilikom instalacije alata, stoga treba pripremiti jesu li kreirane sve potrebne mape.

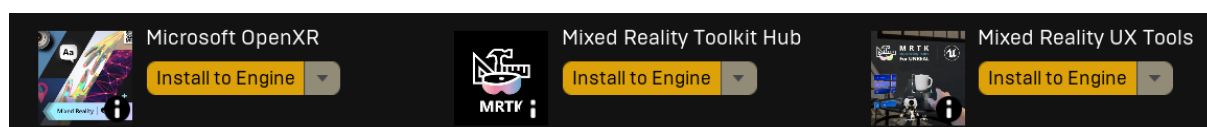
### **6.4. Izrada aplikacije u UnrealEngine-u**

Ideja za izradu ove aplikacije proizašla je iz činjenice da se MR naočale vremenom sve više upotrebljavaju u području proizvodnje, montaže, održavanja i sl. Stoga je u sklopu ovog rada osmišljen konceptualni 3D svijet u kojem bi inženjeri zrakoplovstva, održavanja, mehaničari ili studenti imali mogućnost analize dijelova virtualnog zrakoplova pritom obilazeći stvarni. Dijelovi virtualnog zrakoplova mogu imati oznake o oštećenjima pri čemu bi se klikom na svako od njih otvorio virtualni prozor s opisom i lokacijom tog oštećenja, je li potrebna zamjena ili popravak dijelova te mnoge druge opcije. Za potpunu izradu ove aplikacije potrebno samostalno dizajnirati zrakoplov u nekom od CAD alata te je potrebno imati uvid u razne dokumentacije kao i priručnik za popravak konstrukcije (engl. *Structural Repair Manual*), što se u sklopu ovog projekta, zbog ograničenog vremena i nedostatka potrebnih sredstava za rad, nije provelo.

U aplikaciji je detaljno razrađena manipulacija virtualnim zrakoplovom i njegovim dijelovima, s mogućnošću translacije, rotacije i promjene volumena (ručno ili klikom na virtualna tipkala), pristupa internetu za detaljnije informacije o zrakoplovu te uporabe glasovnih naredbi. Osim same ideje glede potencijalnog industrijskog projekta, težište je i na uporabi i opisu različitih mogućnosti unutar samog *Unreal Engine*-a koje se mogu upotrijebiti u bilo koje druge svrhe.

### 6.4.1. Instalacija potrebnih MR alata za Unreal Engine

Prije pokretanja *Unreal Engine*-a potrebno je instalirati besplatne *Microsoft*-ove dodatke za mješovitu stvarnost dostupne na *GitHub*-u i *Unreal Engine Marketplace*-u. Potrebni alati su *Microsoft OpenXR*, *Mixed Reality Toolkit Hub* te *Mixed Reality UX Tools* [Slika 28].

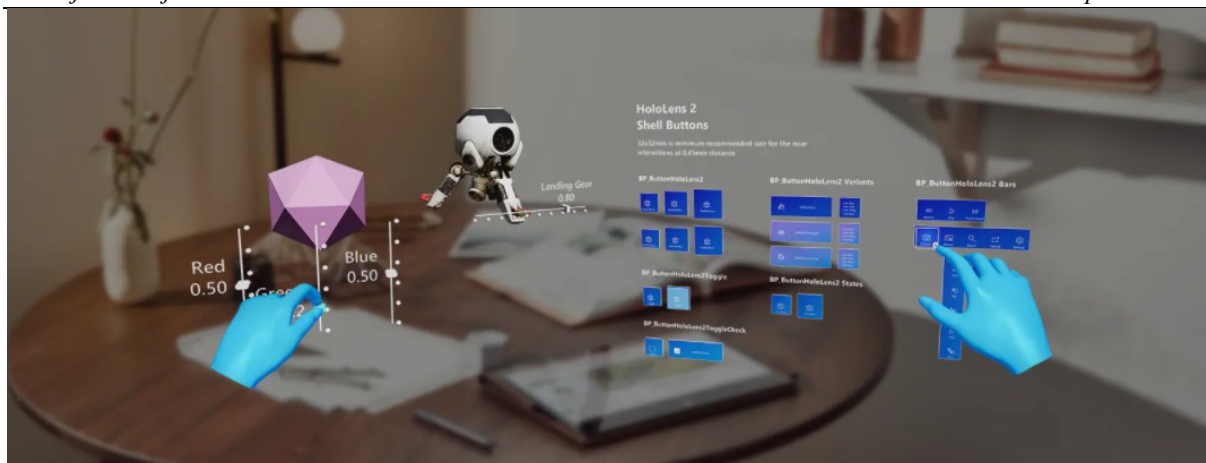


**Slika 28.** Potrebni *Microsoft* alati za mješovitu stvarnost dostupni na *Unreal Engine Marketplace*-u

*Mixed Reality Toolkit Hub* je polazna točka *Mixed Reality Toolkit*-a za *Unreal Engine* te se, zahvaljujući ovom dodatku, u projekte mogu ugraditi različiti *Microsoft Reality* dodaci. Instalacijom ovog dodatka nije potrebno dodatno instalirati *Mixed Reality UX Tools* jer dolaze zajedno u paketu, no *Mixed Reality UX Tools* se može instalirati i zasebno. Unutar *Mixed Reality Toolkit*a nalaze se i ostali dodaci poput *Mixed Reality Graphics Tools*, *Mixed Reality UX Tools Examples* te *Mixed Reality Graphics Tools Examples*.

*Mixed Reality UX Tools* je skup alata koji razvojnim programerima omogućuje stvaranje interaktivnih, vizualno dotjeranih te učinkovitih aplikacija mješovite stvarnosti. Karakteristike ovog alata su simulacija ruku te opća podrška za interakciju ruku na temelju modela bliske i daleke interakcije, 3D virtualna tipkala, klizači, izbornici i prozori, te mogućnost raznih oblika komunikacije s istim [Slika 29].

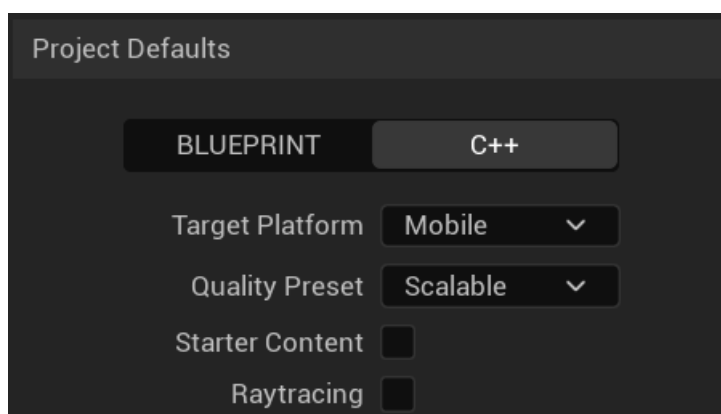
*Microsoft OpenXR* je dodatak koji sadrži skup proširenja (engl. *extensions*) koja otključavaju *Microsoft*-ove značajke namijenjene za mješovitu stvarnost. Dodatak je potreban kako bi se pristupilo značajkama poput mapiranja prostora (engl. *spatial mapping*), razumijevanja prizora i događaja (engl. *scene understanding*), holografsko daljinsko upravljanje (engl. *holographic remote*), glasovni unos (engl. *voice input*), mreža čvorova koja predstavlja šaku (engl. *hand mesh*) itd. Ovaj dodatak radi zajedno s ugrađenim *OpenXR* dodacima u *Unreal Engine*-u koji pružaju funkcionalnost za praćenje očiju (engl. *eye tracking*) i zglobova šake (engl. *joint tracking*), praćenje gesta rukom (engl. *hand gesture*) itd. Alternativa ovom dodatku je ugrađeni dodatak unutar *Unreal Engine*-a, *Microsoft Windows Mixed Reality*, te se može aktivirati jedan ili drugi, no *Microsoft OpenXR* pruža dodatne mogućnosti.



Slika 29. Prikaz nekih značajki *Microsoft*-ovog dodatka za mješovitu stvarnost *Mixed Reality UX Tools*

#### 6.4.2. Stvaranje *Unreal Engine* projekta te prikaz osnovnih postavki i radnji za omogućavanje sustava mješovite stvarnosti

Prilikom kreiranja novog projekta za mješovitu stvarnost, pod *Project Defaults* potrebno je odabrati opcije prikazane na slici [Slika 30].

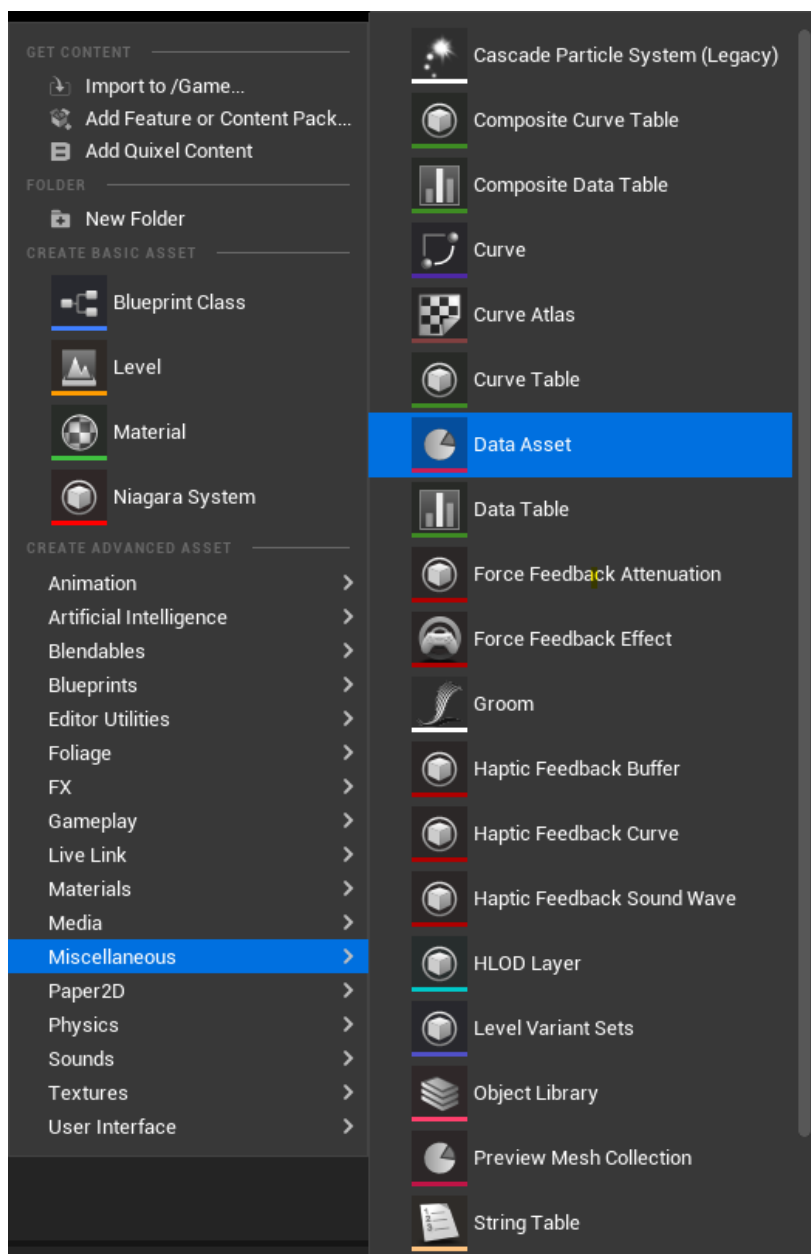


Slika 30. Odabir postavki prilikom kreiranja *Unreal Engine* projekta za mješovitu stvarnost

Nakon što se učita novi projekt potrebno je klikom na *Edit* zatim *Plugins* odabrati potrebne dodatke kako bismo mogli izraditi program za sustav mješovite stvarnosti. Od dodatka je potrebno označiti spomenute u poglavlju 6.4.1. koji se nalaze u dijelu *Installed*, pri čemu se *Microsoft Open XR* može ili ne mora označiti, no ovisno o tome potrebno je označiti njegovu alternativu *Microsoft Windows Mixed Reality* koji se nalazi unutar dijela *Virtual Reality*. Za prijašnje verzije *Unreal Engine*-a koje podupiru *HoloLens* bilo je potrebno označiti i dodatak *HoloLens* unutar dijela *Augmented Reality*. Ukoliko su dodaci iz poglavlja 6.4.1. preuzeti s

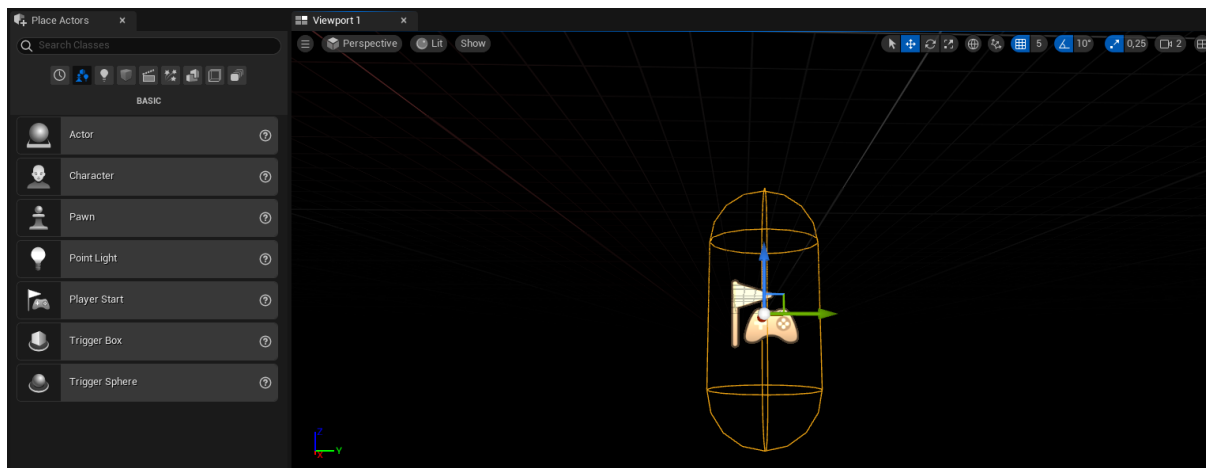
*GitHub*-a, potrebno ih je smjestiti u mapu *Plugins* koja je spremljena na istoj lokaciji kao i *Unreal Engine* projekt.

Sljedeći korak je dodavanje *ARSessionConfig*-a, na način da se unutar *Content Browser*-a klikne na *Add* ili desnu tipku miša te pod *Miscellaneous* odabere *Data Asset* [Slika 31]. Klikom na novonastali *Data Asset*, otvara se prozor u kojem je potrebno odabrati opciju *ARSessionConfig*. *ARSessionConfig* zaslužan je za pohranjivanje sesija proširene ili mješovite stvarnosti te je unutar njega potrebno spremi već zadane postavke.



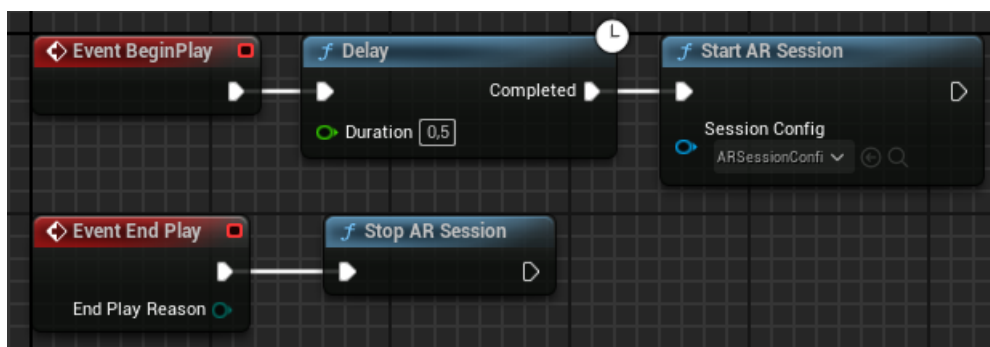
Slika 31. Prikaz stvaranja *Data Asset*-a za *ARSessionConfig* unutar *Unreal Engine*-a

Treći korak je kreiranje prazne razine (engl. *Level*) klikom na *File* te *New Level* i dodavanje *Player Start* unutar prozora *Place Actors* [Slika 32]. Pozicija *Player Start*-a određuje naše početne koordinate prilikom pokretanja sesije. Potrebno je zatim spremiti trenutnu razinu klikom na *File* te *Save Current Level As*.



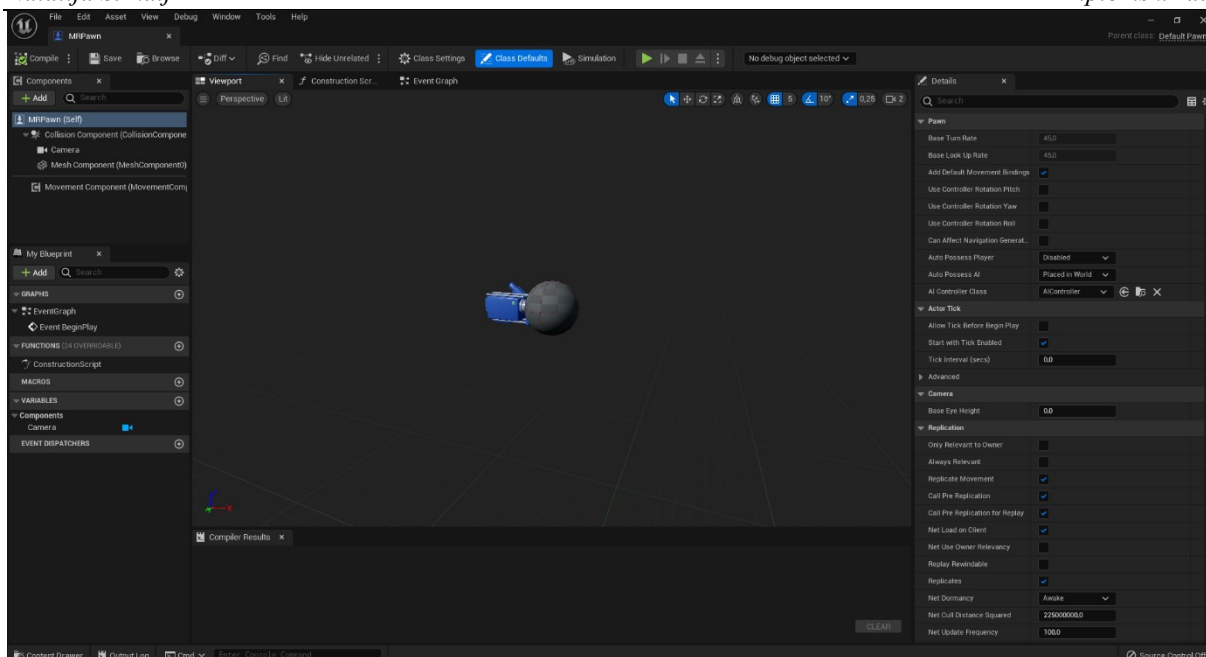
Slika 32. Dodavanje *Player Start*-a unutar *Unreal Engine*-a

Kako bismo uspješno pokrenuli sesiju, u *Level Blueprint*-u, unutar *Event Graph*, potrebno je napraviti određene modifikacije prikazane slikom [Slika 33]. Funkcija *Delay* sprječava potencijalno rušenje programa prilikom pokretanja istog na *HoloLens*-u, no nije ju nužno postaviti. Čvorovi crvene boje, u ovom slučaju *Event Begin Play* te *Event End Play* imaju ključnu ulogu u izvršavanju funkcija unutar skripte prilikom pokretanja sesije.

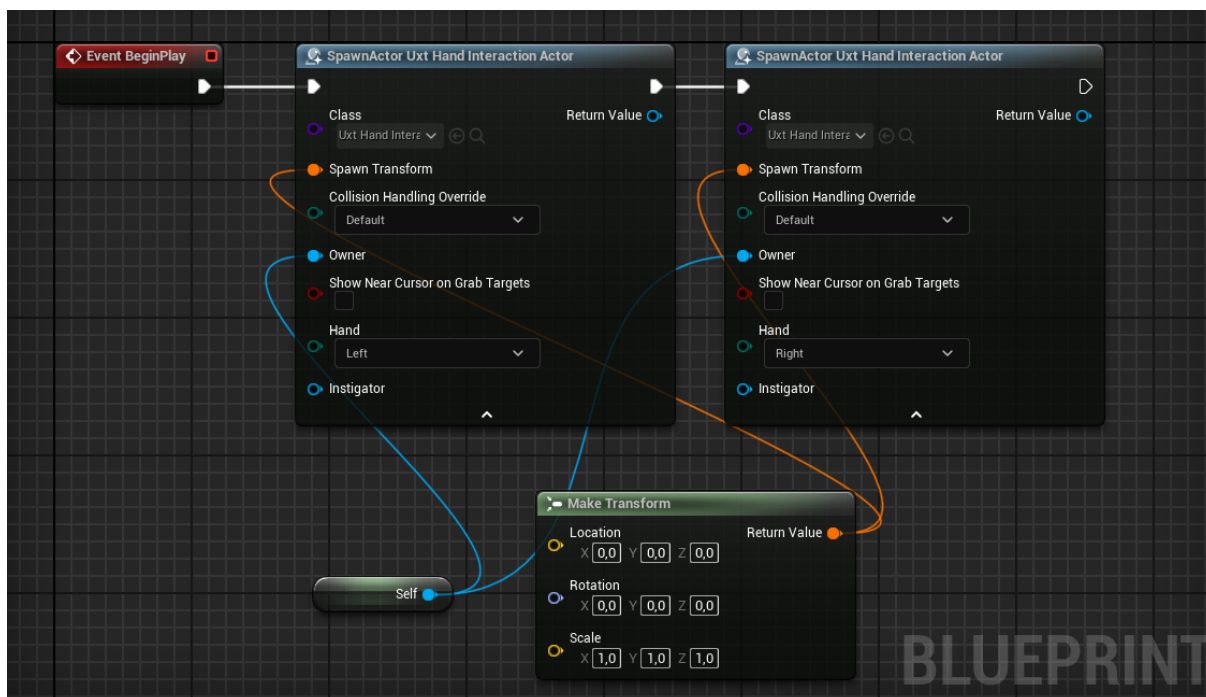


Slika 33. Prikaz čvorova za pokretanje i završavanje sesije unutar *Level Blueprint*-a u *Unreal Engine*-u

Sljedeći korak je kreiranje klase *Pawn*, kategorije *DefaultPawn* (nalazi se u izborniku pod *All Classes*), koji je pokretan te sadrži kameru za promatranje scene [Slika 34]. Unutar *Event Graph*-a potrebno je dodati čvorove za prikaz lijeve i desne šake [Slika 35].

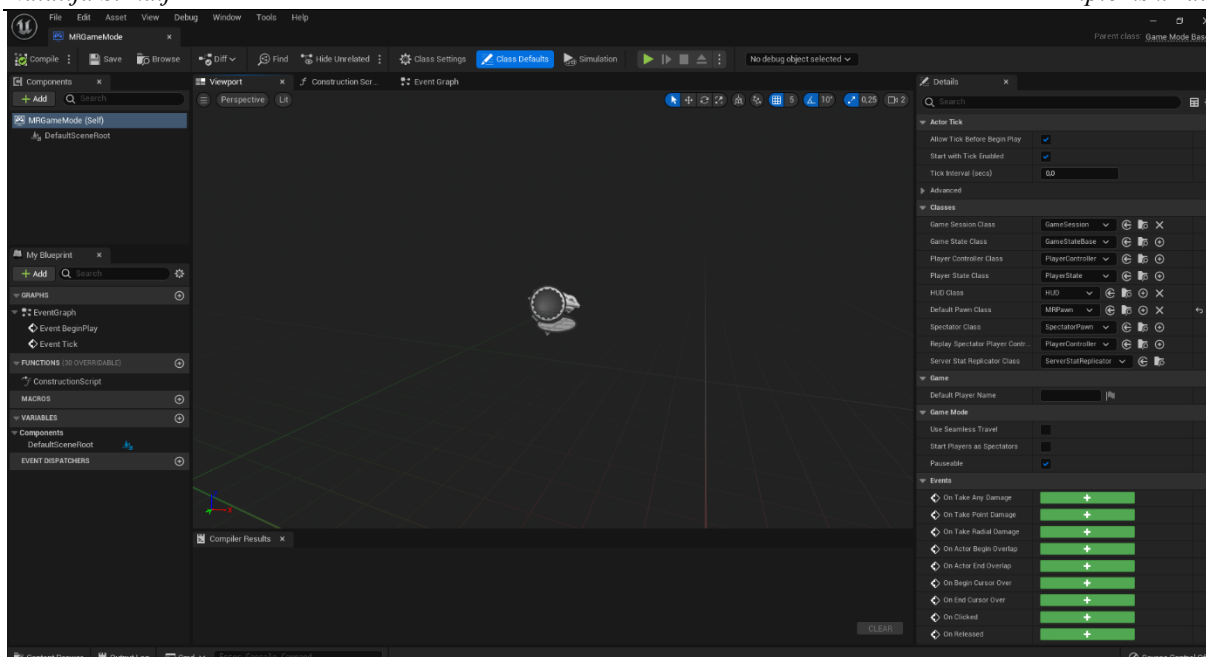


Slika 34. Prikaz Viewport-a klase *Pawn*, kategorije *Default Pawn*, unutar *Unreal Engine-a*



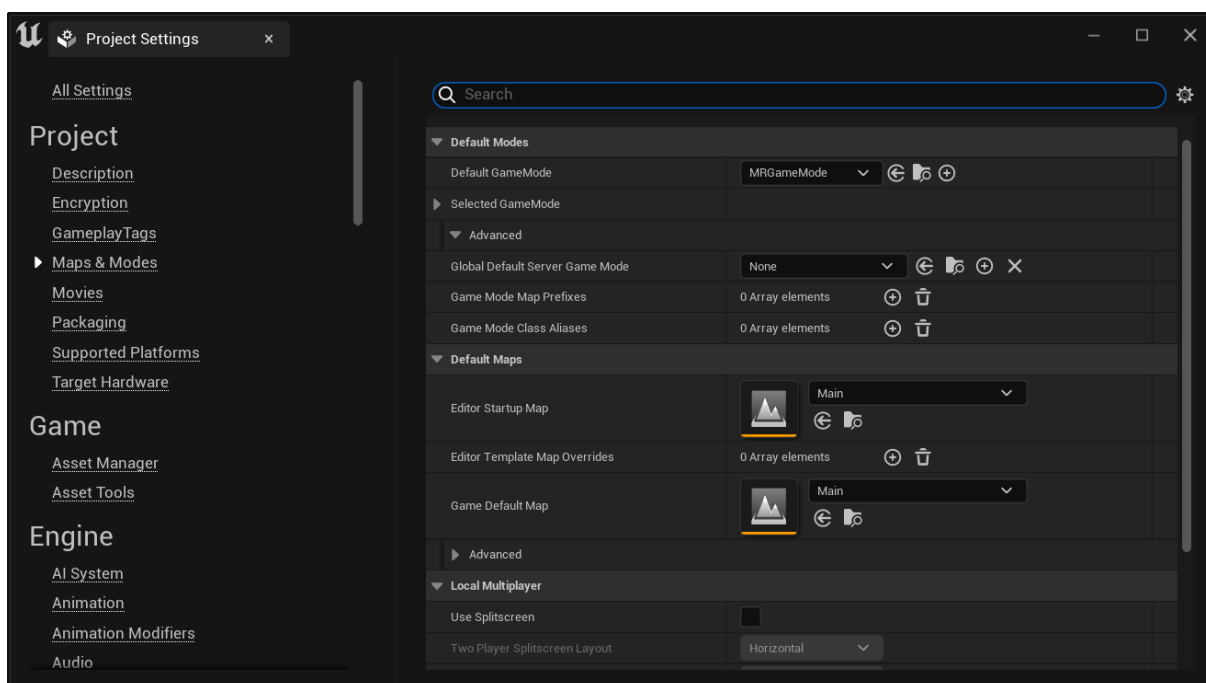
Slika 35. Prikaz *Event Graph-a* klase *Pawn*, kategorije *Default Pawn*, unutar *Unreal Engine-a*

Zadnji korak početnog dijela aplikacije za sustav mješovite stvarnosti je stvaranje klase *Game Mode Base*, u kojoj je u prozoru *Classes* s desne strane, dijelu *Default Pawn Class*, potrebno odabrati *Default Pawn* (imena *MRPawn*) kojeg smo prethodno stvorili [Slika 36].



Slika 36. Prikaz Viewport-a, klase *Game Mode Base* unutar *Unreal Engine-a*

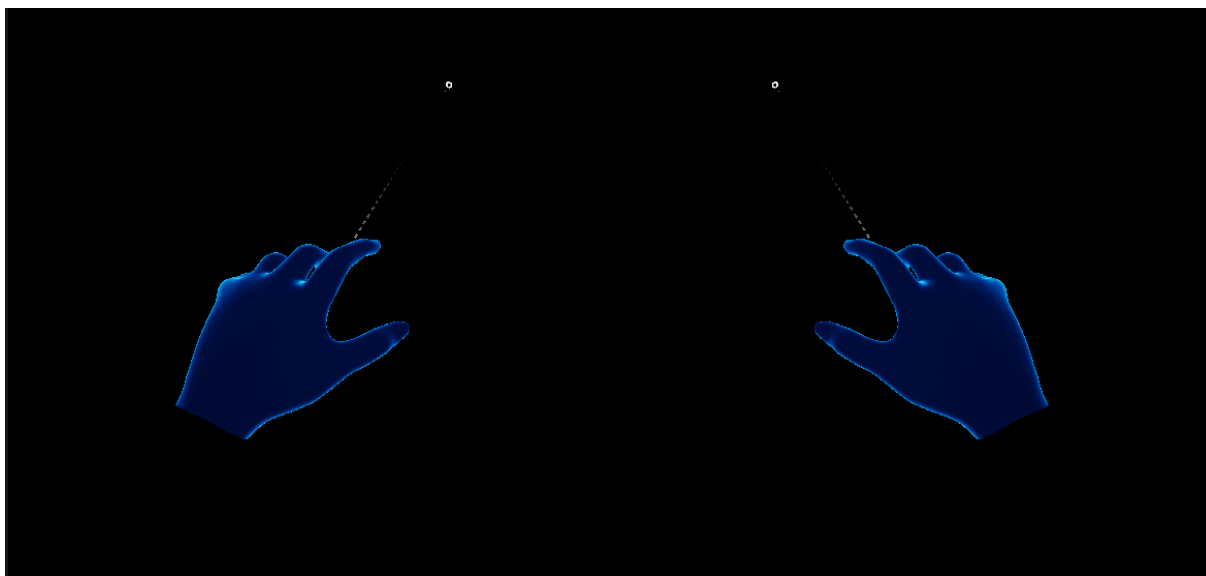
Kako bi se prilikom pokretanja projekta uvijek prikazala ova razina, potrebno je u postavkama pod *Edit*, zatim *Project Settings* te *Maps & Modes* napraviti određene modifikacije [Slika 37]. Za *Default GameMode* potrebno je postaviti prethodno kreiranu klasu *Game Player Name Mode Base*, a za *Editor Startup Map* te *Game Default Map* razinu na kojoj razvijamo aplikaciju.



Slika 37. Prikaz postavki projekta unutar *Unreal Engine-a*



Konačan rezultat ovog postupka je prikaz računalno generiranih šaka nakon pokretanja sesije, odnosno klika na zelenu ikonicu trokuta [Slika 38]. Gibanje šaka lijevo i desno vrši se pritiskom tipki *ALT* za desnu šaku te *SHIFT* za lijevu šaku zajedno s pomicanjem miša. Gibanje šaka naprijed nazad vrši se na isti način, no umjesto pomicanja miša, pomičemo njegovu srednju tipku. Hvatanje virtualnih objekata moguće je pritiskom lijeve tipke miša, rotacija pritiskom na *ALT* ili *SHIFT* s paralelnim pritiskom desne tipke miša nakon klika lijeve, a pokazivanje dlanova pritiskom na *ALT* ili *SHIFT* s paralelnim pritiskom tipke *HOME*. Promjena volumena objekta vrši se na način da se objekt jednom rukom uhvati za jedan kraj, a drugom za drugi, pri čemu se pritom pomiču obje ruke ili jedna od njih.

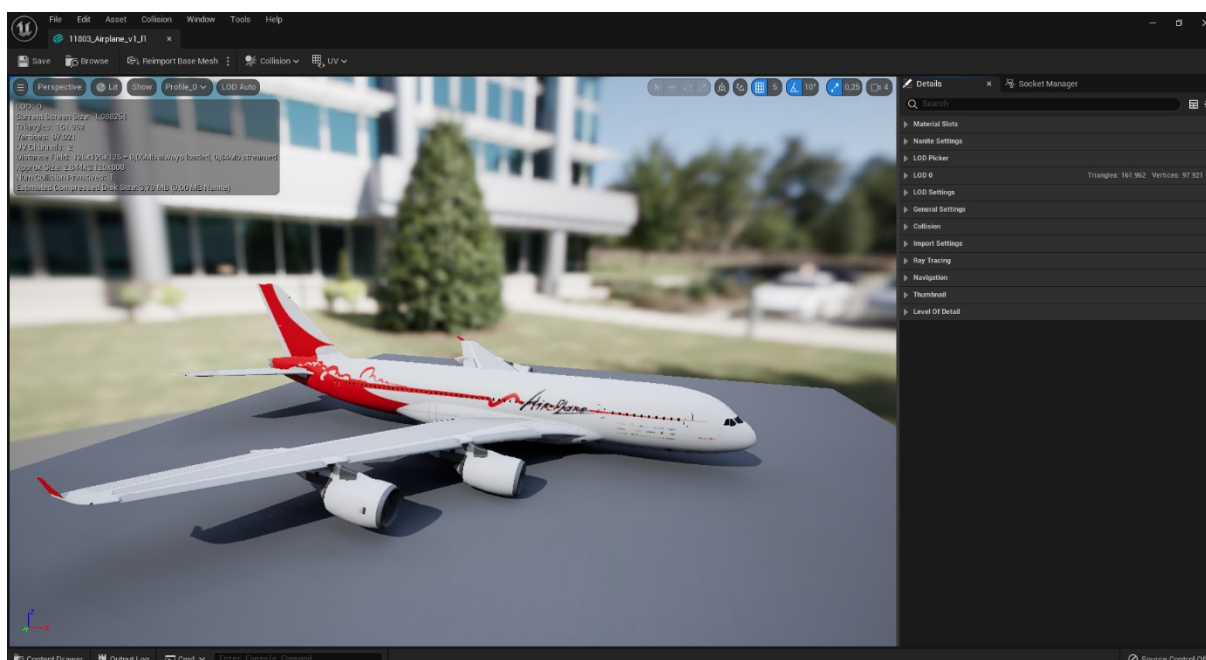


Slika 38. Prikaz računalno generiranih ruku nakon pokretanja sesije unutar *Unreal Engine-a*

#### 6.4.3. Učitavanje 3D modela zrakoplova te omogućavanje interakcije s istim

Inicijalna ideja ovog rada je bila mogućnost interakcije s cjelovitim modelom zrakoplova te mogućnost da se ručno uzme svaki njegov dio i odvoji od ostatka. Zbog nastalih poteškoća, zrakoplov *Airbus A340* preuzet s interneta [Slika 39][38], datoteke je tipa *3D OBJECT*, u čiju kategoriju spadaju *.obj*, *.stl*, *.fbx*, *.amf* oblici datoteka, te je u projekt učitana kao cjeloviti model čijim se dijelovima ne može pristupiti. Razlog tome je što se sa modelima učitanim dodatkom *Datasmith*, primjerice, *.step* datotekama, nije mogla ostvariti ikakva interakcija, unatoč svim dobro postavljenim postavkama. Oblik datoteka tipa *.step* sastoji se od cjelovite geometrije unutar koje se se može pristupiti svakom dijelu zasebno te je idealan za ostvarivanje inicijalne zamisli ovog projekta. Postoje razne mogućnosti zbog kojih se nije mogla ostvariti interakcija, a jedna od njih je *beta* verzija dodataka koji moraju međusobno surađivati te njihova

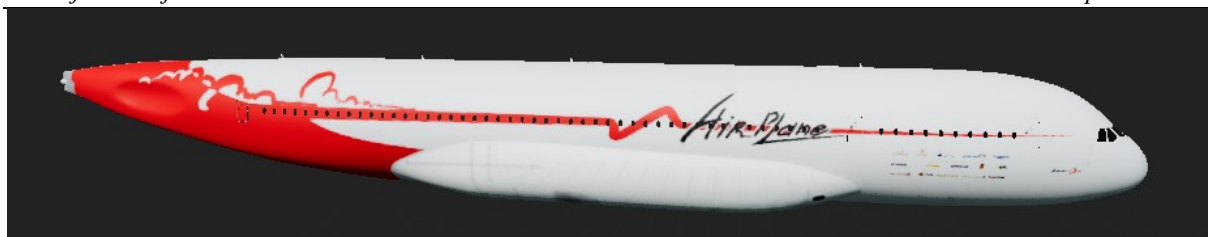
nekompatibilnost u vidu sustava mješovite stvarnosti. Unatoč tome, pronašlo se jednako dobro rješenje opisano u nastavku. Model zrakoplova se prvo učita s uključenom opcijom *Complete Mesh*, a zatim s isključenom kako bi se učitali samo njegovi dijelovi, međusobno odvojeni.



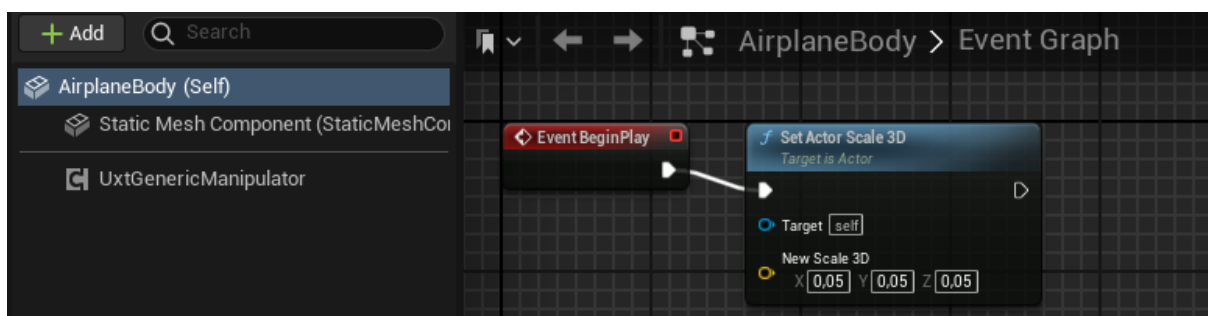
Slika 39. Prikaz učitano modela zrakoplova *Airbus A340* u projekt *Unreal Engine-a*

Potpuni model te svaki njegov dio su klasa *Actor* koji će klikom na virtualna tipkala biti stvoreni u prostoru te na isti način uklonjeni iz istog. Postoji opcija da se modeli već nalaze u prostoru i nevidljivi su pa klikom na tipkala postanu vidljivi, no u sklopu projekta mješovite stvarnost javljale su se situacije u kojima bi, ukoliko bi došlo do direktnog preklapanja vidljivog i nevidljivog elementa, računalno generirane šake imale naispravnu interakciju ili uopće ne bi imale interakciju s vidljivim elementom. Kako bi se to spriječilo, većinskim su se dijelom upotrebljavali čvorovi *Spawn Actor from Class* te *Destroy Actor*.

Za model zrakoplova i svaki njegov dio potrebno je stvoriti *Blueprint* klasu, u kojoj će se odrediti skala elementa nakon pojavljivanja u okruženju, te dodati komponentu *UxtGenericManipulator* koja omogućava interakciju između šaka i objekata unutar *Unreal Engine-a* (kasnije virtualnih objekata). Kako bi se objekti uistinu mogli pomicati, dio *Mobility* unutar *Viewport-a* mora biti označen kao *Movable*. Slika [Slika 40] prikazuje trup zrakoplova, a slika [Slika 41] *Blueprint* klasu analognu preostalim dijelovima.



Slika 40. Prikaz učitano modela trupa zrakoplova *Airbus A340* u projekt *UnrealEngine-a*



Slika 41. Prikaz *Event Graph-a* modela trupa zrakoplova u *Unreal Engine-u*

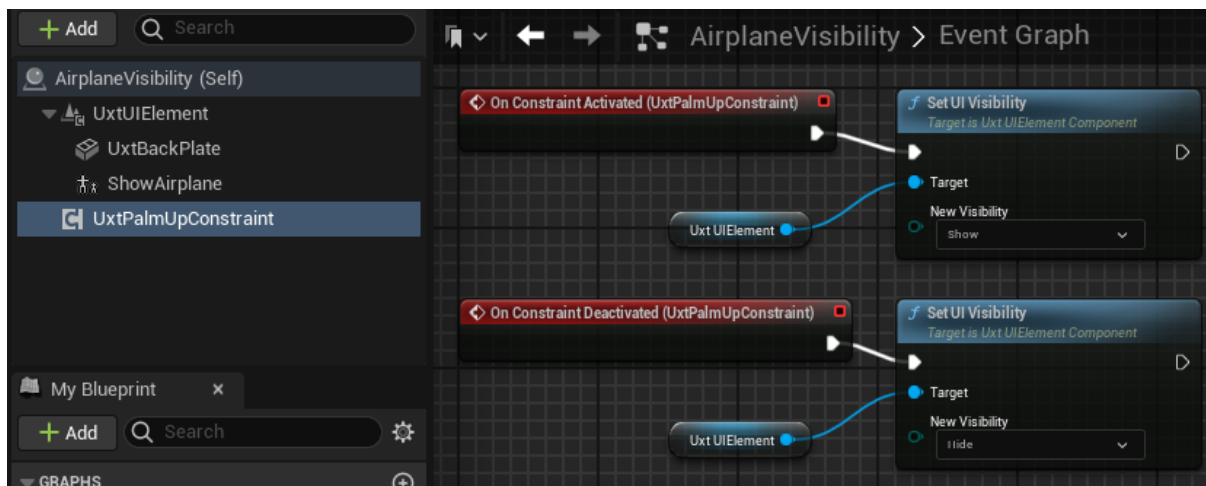
#### 6.4.4. Izrada početnih virtualnih tipkala za stvaranje zrakoplova i korisničkog sučelja

Prije svega, u *Viewport*, odnosno razinu, potrebno je dodati neki oblik osvjetljenja kako bi elementi koji se nalaze u prostoru bili vidljivi. Stoga je dodano četiri *Directional Light* osvjetljenja iz prozora *Place Actors* te su postavljena na željene lokacije.

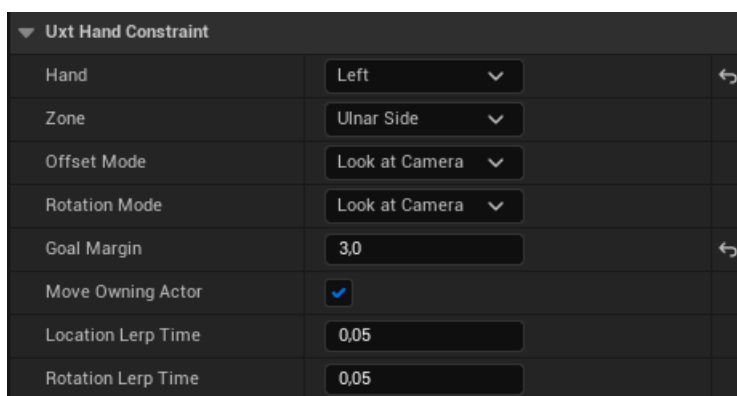
Kako bi se u prostoru stvorili zrakoplov te korisničko sučelje (engl. *User Interface*) s informacijama o zrakoplovu, opcijom za prikaz dijelova itd., potrebno je napraviti neki oblik naredbe, bilo glasovne putem glasovnog unosa, bilo dodirne putem tipkala. U tu svrhu kreirala su se tipkala koja se prikazu ako korisnik licu pokaže lijevi i/ili desni potpuno otvoreni dlan. Prikazom lijevog dlana pojavi se tipkalo kategorije *Switch* za odabir stvaranja ili uklanjanja zrakoplova iz prostora, a prikazom desnog dlana pojavi se tipkalo iste kategorije s mogućnošću odabira stvaranja ili uklanjanja korisničkog sučelja za zrakoplov. Ovo su jedina tipkala koja su inicijalno smještena u prostor te rade na principu prikaži/sakrij, budući da su malih dimenzija te ne ometaju rad. U prostor ih možemo smjestiti daleko van radijusa kretanja te će se uvijek pojaviti ispred nas ukoliko licu pokažemo otvorene dlanove.

Prvi korak je kreiranje klase *Actor* te, unutar *Components*, dodavanje komponenti *UxtUIElement* te *UxtPalmUpConstraint* [Slika 42] [Slika 43] koja je potrebna za zadavanje navedene naredbe prilikom pokazivanja otvorenog dlana. S komponentom *UxtUIElement* potrebno je stvoriti relaciju roditelj-dijete, pri čemu je dijete tipkalo kategorije *Switch* i direktno je vezano za komponentu *UxtUIElement* koja se aktivira ili deaktivira ovisno o dlanu. Relacija

se može načiniti dodavanjem *Child Actor* komponente te, unutar prozora *Details*, pod *Child Actor Component*, dio *Child Actor Class*, odabrati naziv prethodno kreirane komponente za koju želimo da postane dijete (*ShowAirplane* na slici [Slika 42]). Dok god je dlan otvoren i orijentiran prema kameri, ovaj virtualni element prati lokaciju korisnika te se pomiče zajedno s njim.



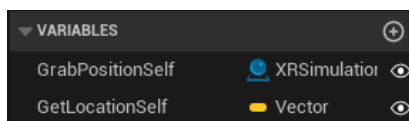
Slika 42. Prikaz *Event Graph*-a *UxtUIElement*-a koje sadrži tipkalo kategorije *Switch* za prikaz zrakoplova u *Unreal Engine*-u



Slika 43. Prikaz modifikacija komponente *UxtPalmUpConstraint*-a lijeve ruke u *Unreal Engine*-u

Drugi korak je kreiranje klase *Actor*, kategorije *Uxt Pressable Switch Button Actor*, odnosno djeteta *UxtUIElement*-a iz prethodnog odlomka. Svrha ovog tipkala je prikazivanje zrakoplova ili korisničkog sučelja za zrakoplov koji se trebaju stvoriti blizu lokacije na kojoj se nalazimo, bilo da se nalazimo na početnoj lokaciji (prilikom pokretanja sesije) ili smo se pomakli na neku drugu lokaciju u odnosu na početnu. To je moguće ostvariti kreiranjem varijable *GrabPositionSelf* [Slika 44] koja prilikom pokretanja sesije dohvaća sve potrebne informacije o kategoriji *XRSimulation Actor*, uključujući i njenu lokaciju. *XRSimulation Actor* predstavlja

korisnika, odnosno nas. Kada se u *Blueprint* klasu za *Uxt Pressable Switch Button Actor* dohvati *XRSimulation Actor*, čvorom *Event Tick* potrebno je svakog trenutka pratiti njegovu lokaciju te pohranjivati u vektor *GetLocationSelf*, kao što prikazuje slika [Slika 45].



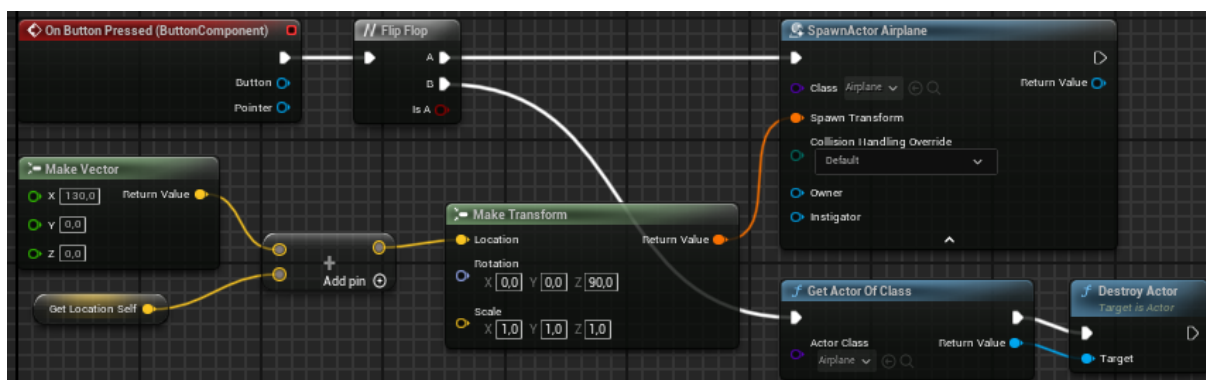
Slika 44. Prikaz komponentata za dohvaćanje i spremanje lokacije korisnika u *Unreal Engine-u*



Slika 45. Prikaz *Event Graph*-a *Uxt Pressable Switch Button Actor*-a za dohvaćanje i spremanje lokacije korisnika u *Unreal Engine-u*

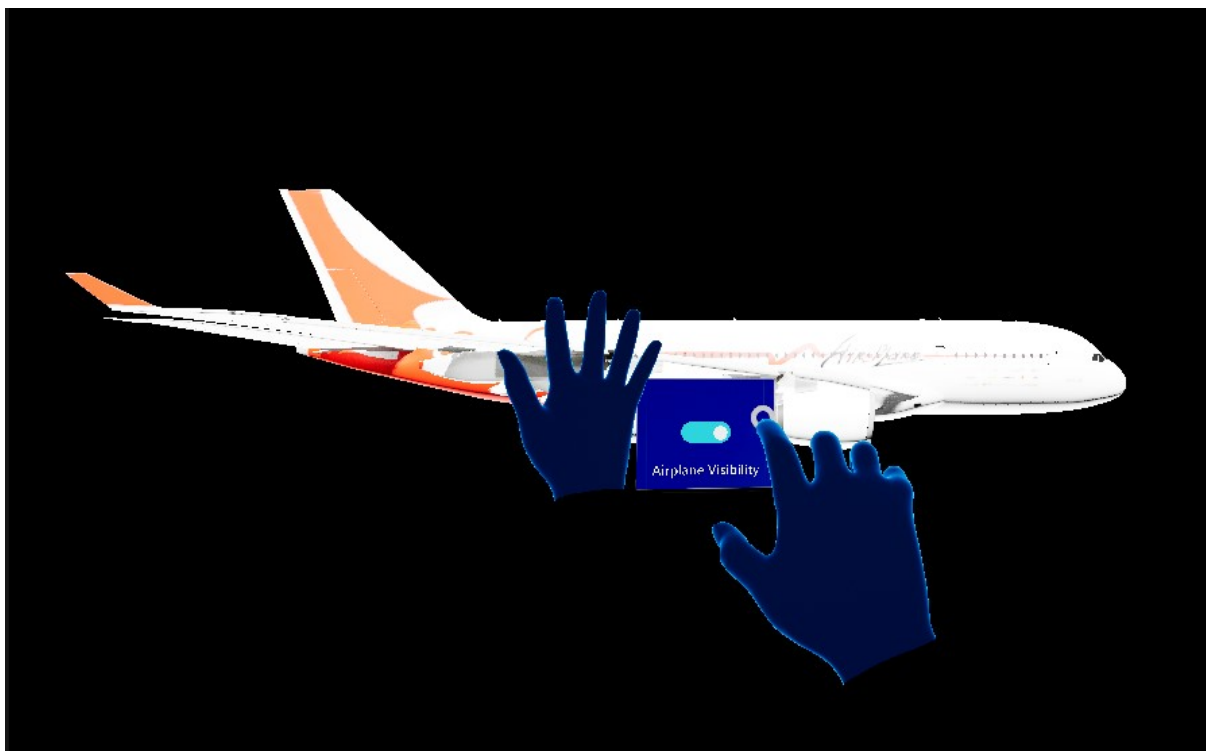
Slika u nastavku [Slika 46] prikazuje stvaranje zrakoplova (unutar istog *Event Graph*-a) prilikom pritiska na tipkalo, koje se izvršava zahvaljujući *Event* čvoru *On Button Pressed*. Zrakoplov se u okruženju pojavi udaljen 130 cm po osi *x* od trenutne lokacije korisnika, a ponovnim se pritiskom na tipkalo zahvaljujući čvoru *Flip Flop* uklanja iz prostora.

*Blueprint* skripta korisničkog sučelja za zrakoplov analogna je *Blueprint* skripti za zrakoplov, no potrebno ju je prethodno kreirati (opisano u poglavlju 6.4.5.).

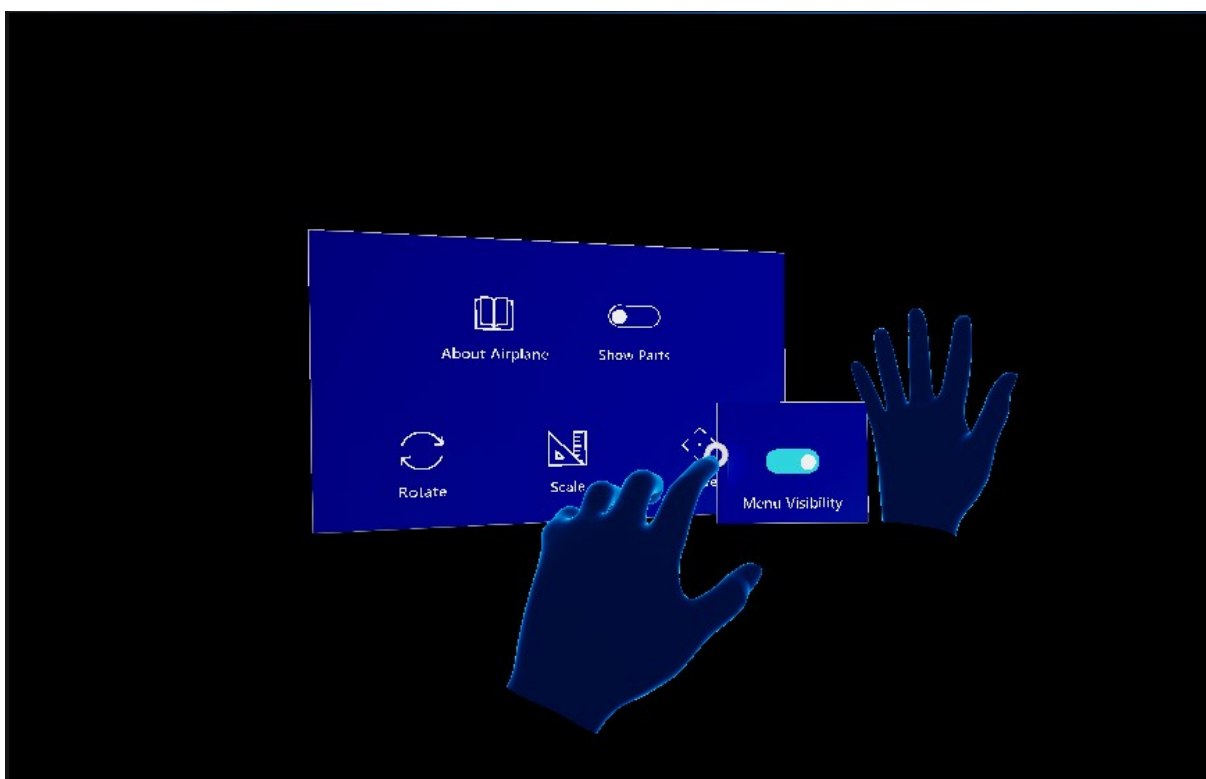


Slika 46. Prikaz *Event Graph*-a *Uxt Pressable Switch Button Actor*-a za stvaranje i uklanjanje zrakoplova u *Unreal Engine-u*

Nakon pokretanja sesije, rezultat dosadašnjih skripti prikazan je slikama [Slika 47] [Slika 48].



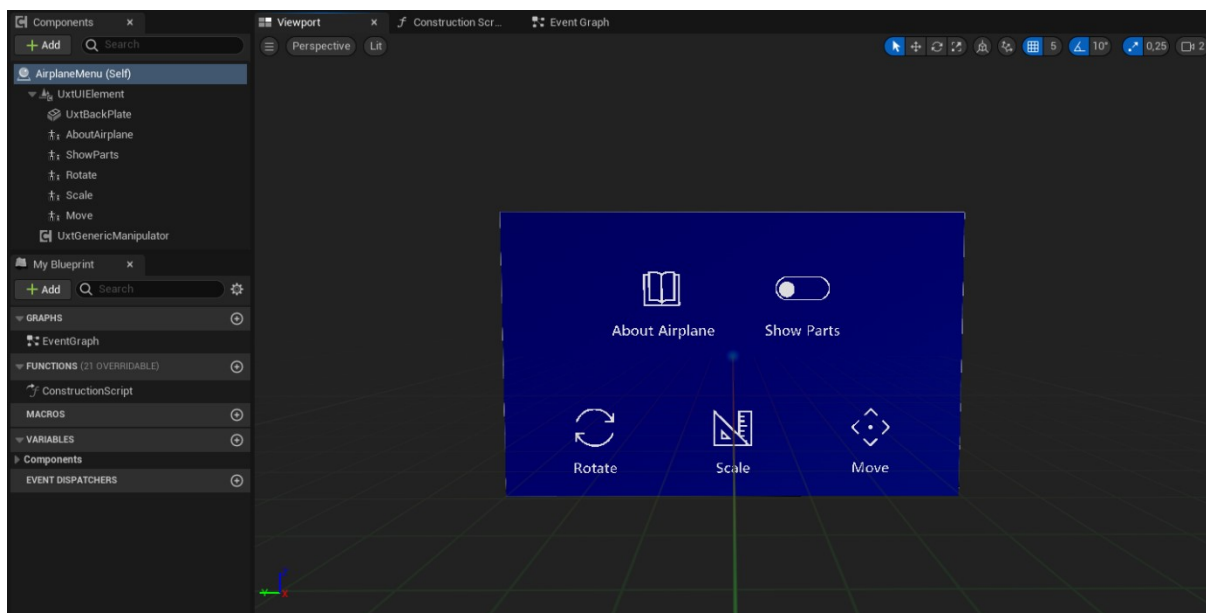
Slika 47. Prikaz zrakoplova nakon pritiska na tipkalo unutar *Unreal Engine-a*



Slika 48. Prikaz korisničkog sučelja za zrakoplov nakon pritiska tipkala unutar *Unreal Engine-a*

### 6.4.5. Izrada korisničkog sučelja za zrakoplov

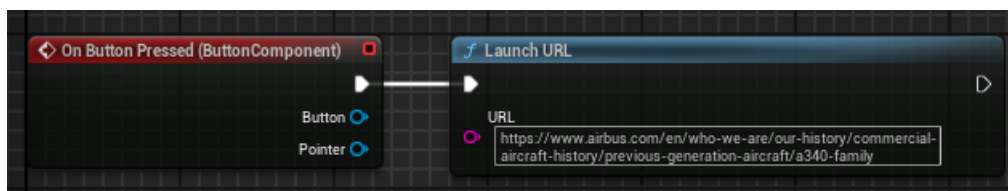
Korisničko sučelje za zrakoplov sastoji se od raznih tipkala klase *Actor*, kategorije *Uxt Pressable Button Actor* te jedne kategorije *Uxt Pressable Switch Button Actor* [Slika 49]. Kao i u prethodnom poglavlju, komponenta *UxtUIElement* predstavlja roditelja, a sva tipkala koja se nalaze na korisničkom sučelju predstavljaju njegovu djecu (postupak opisan u poglavlju 6.4.4.). Izgled ikonica kao i same dimenzije tipkala mogu se odrediti u prozoru *Details*, klikom na *Child Actor Component*, zatim *Child Actor Template* te *Uxt Pressable Button*.



Slika 49. Prikaz korisničkog sučelja za zrakoplov unutar *Viewport*-a u *Unreal Engine*-u

#### 6.4.5.1. Tipkalo *About Airplane*

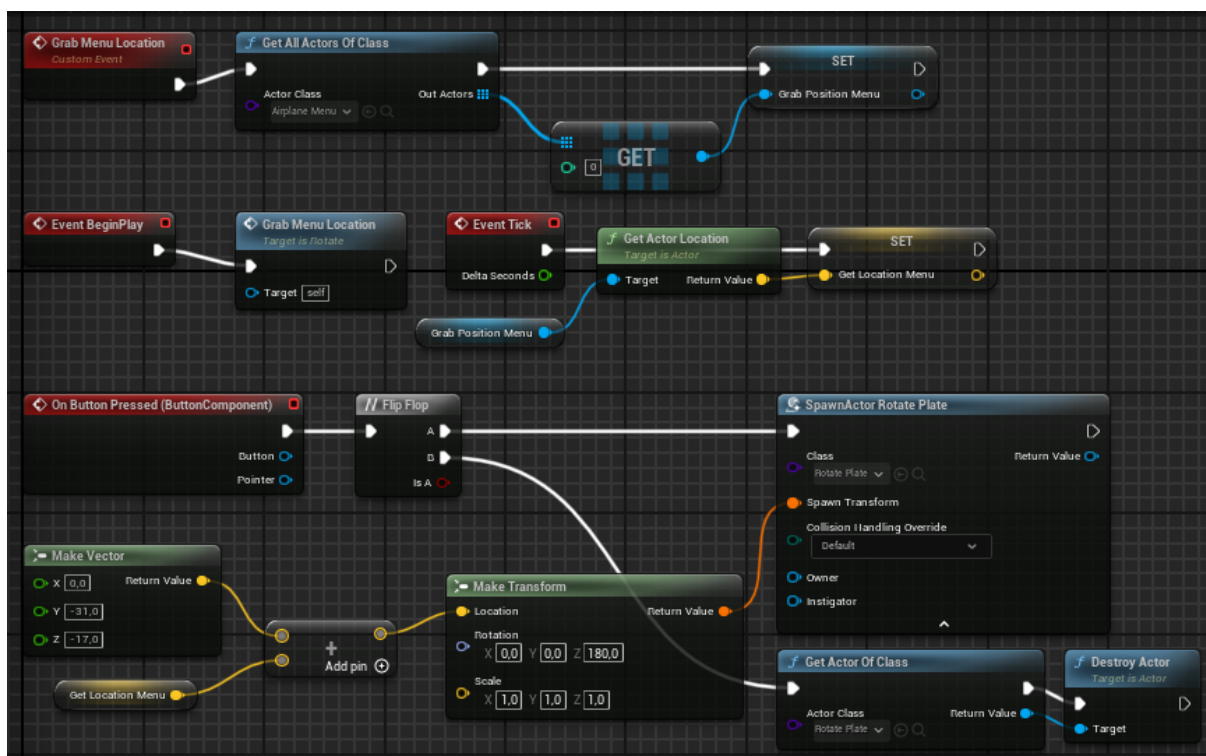
Tipkalo *About Airplane* je *Uxt Pressable Button Actor* tipkalo koje korisnika vodi na službenu internetsku stranicu zrakoplova *Airbus A340* što je moguće zahvaljujući funkcijskom čvoru *Launch URL* unutar kojeg je potrebno upisati željenu poveznicu [Slika 50][39].



Slika 50. Prikaz *Event Graph*-a tipkala *About Airplane* u *Unreal Engine*-u

## 6.4.5.2. Tipkala Rotate i Move

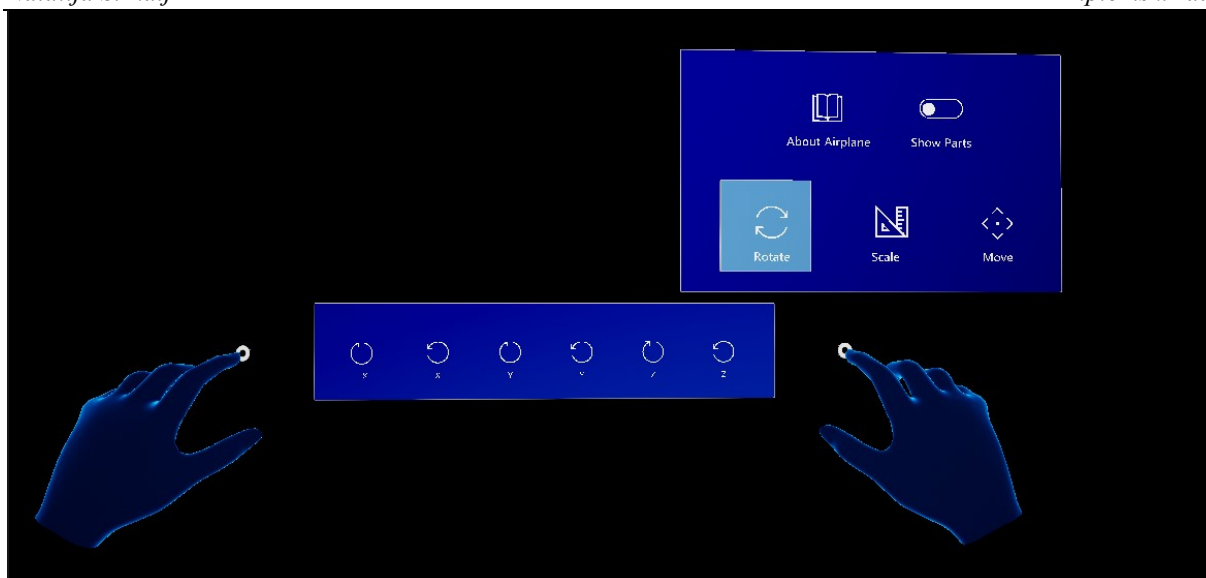
Tipkalo *Rotate* je *Uxt Pressable Button Actor* tipkalo koje korisnicima omogućuje rotaciju zrakoplova, ukoliko s njime ne žele imati direktnu interakciju, budući da rotacija elemenata rukama, za neiskusne korisnike, može biti zamorna. Pritiskom na ovo tipkalo stvara se novo sučelje *RotatePlate* te se nalazi na uvijek istoj relativnoj udaljenosti od korisničkog sučelja za zrakoplov, po principu dohvaćanja i spremanja lokacije opisane u poglavlju 6.4.4. [Slika 51].



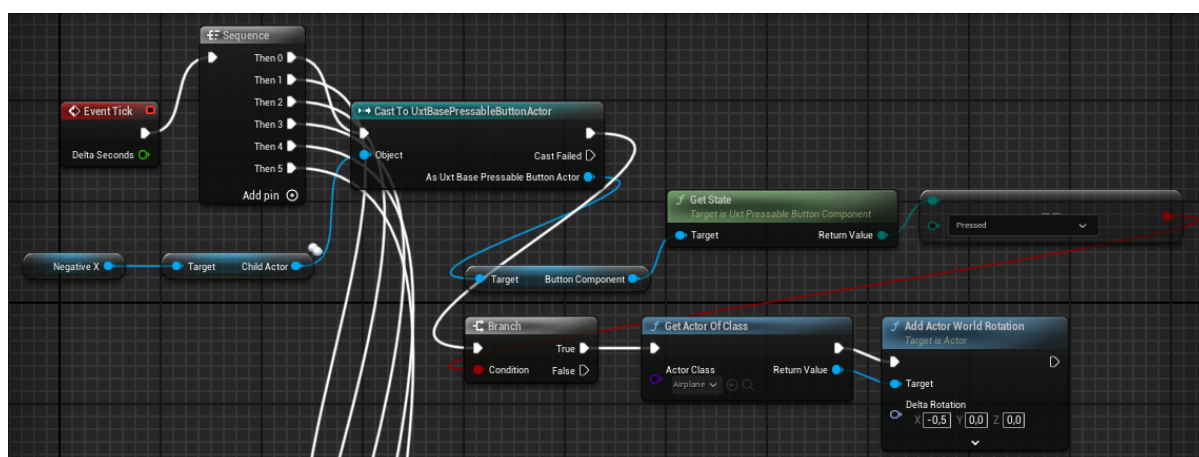
Slika 51. Prikaz *Event Graph*-a tipkala *Rotate* u *Unreal Engine*-u

*RotatePlate* je sučelje na kojem se nalazi šest tipkala za rotaciju zrakoplova oko svake osi [Slika 52]. Kao *Event* čvor upotrijebio se *Event Tick*, budući da djeluje kao svojevrsna *while* petlja koja provjerava stanje tipkala, odnosno je li pritisnuto ili ne. Dok je tipkalo pritisnuto, zrakoplov se rotira, a kada je otpušteno, rotacija prestaje [Slika 53].



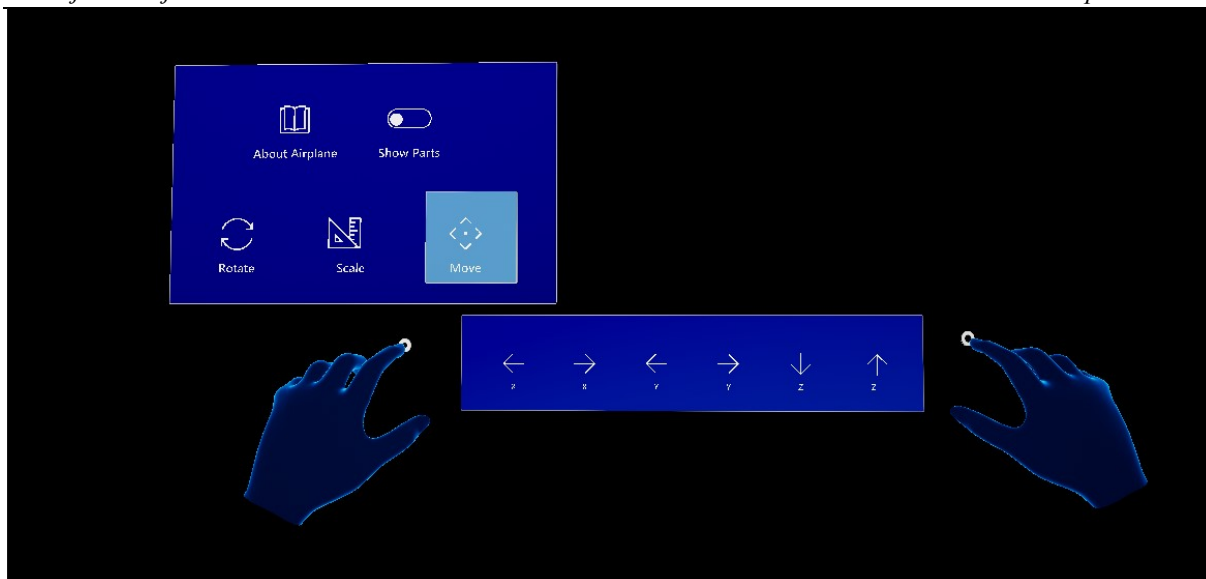


Slika 52. Prikaz sučelja *Rotate* za rotaciju zrakoplova nakon pokretanja sesije u *Unreal Engine-u*

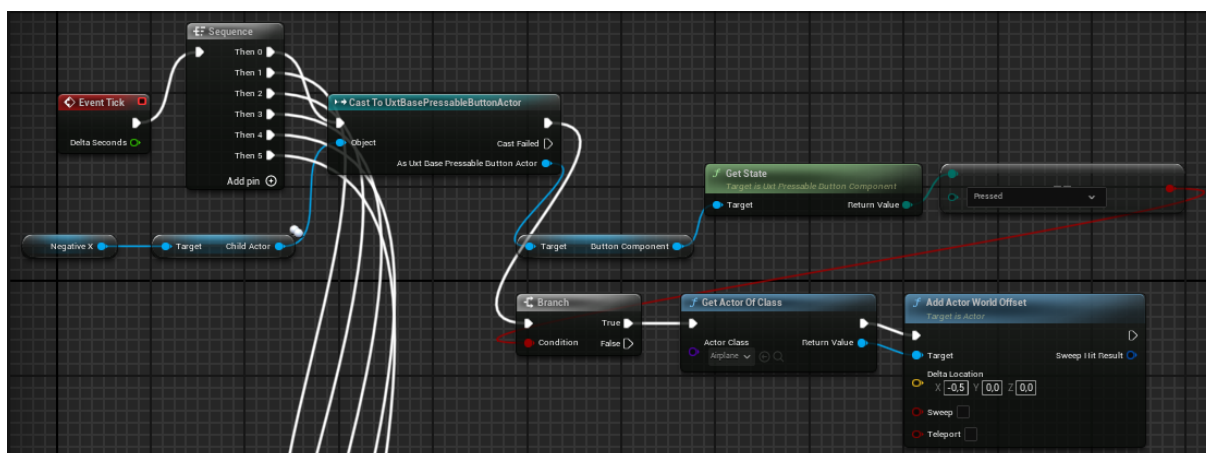


Slika 53. Prikaz *Event Graph*-a sučelja *Rotate* za rotaciju zrakoplova (dio rotacije oko negativne *x* osi) u *Unreal Engine-u*

Po istom se principu kreira i sučelje *MovePlate* na kojem se nalazi šest tipkala za translaciju zrakoplova po svakoj osi [Slika 54] kao i odgovarajuća *Blueprint* skripta [Slika 55].



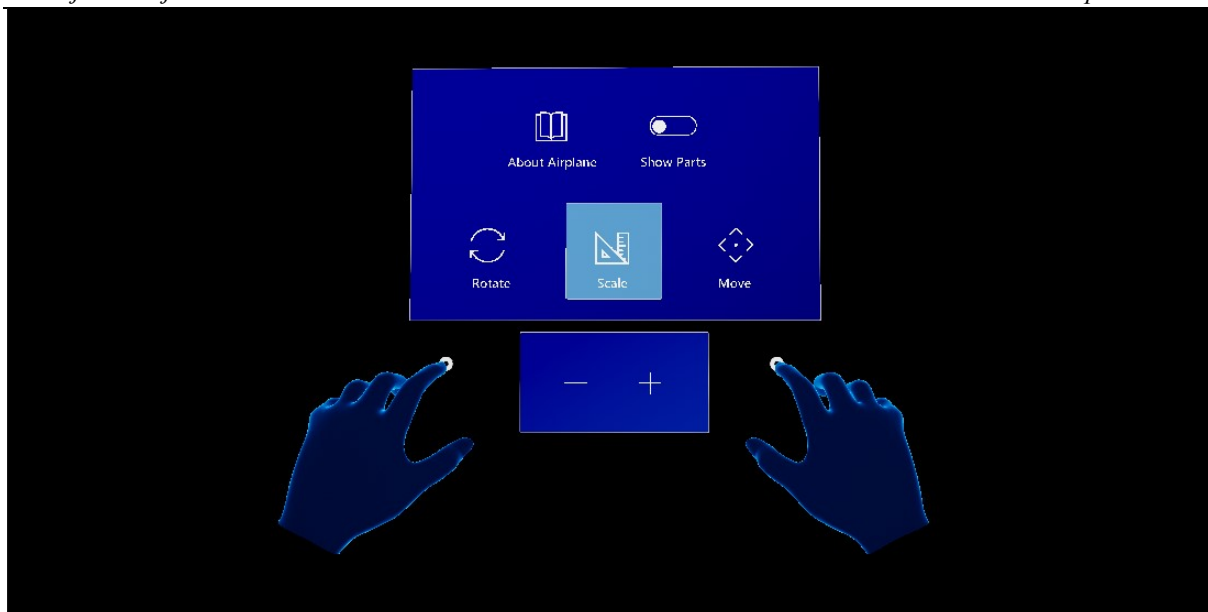
Slika 54. Prikaz sučelja *Move* za translaciju zrakoplova nakon pokretanja sesije u *Unreal Engine-u*



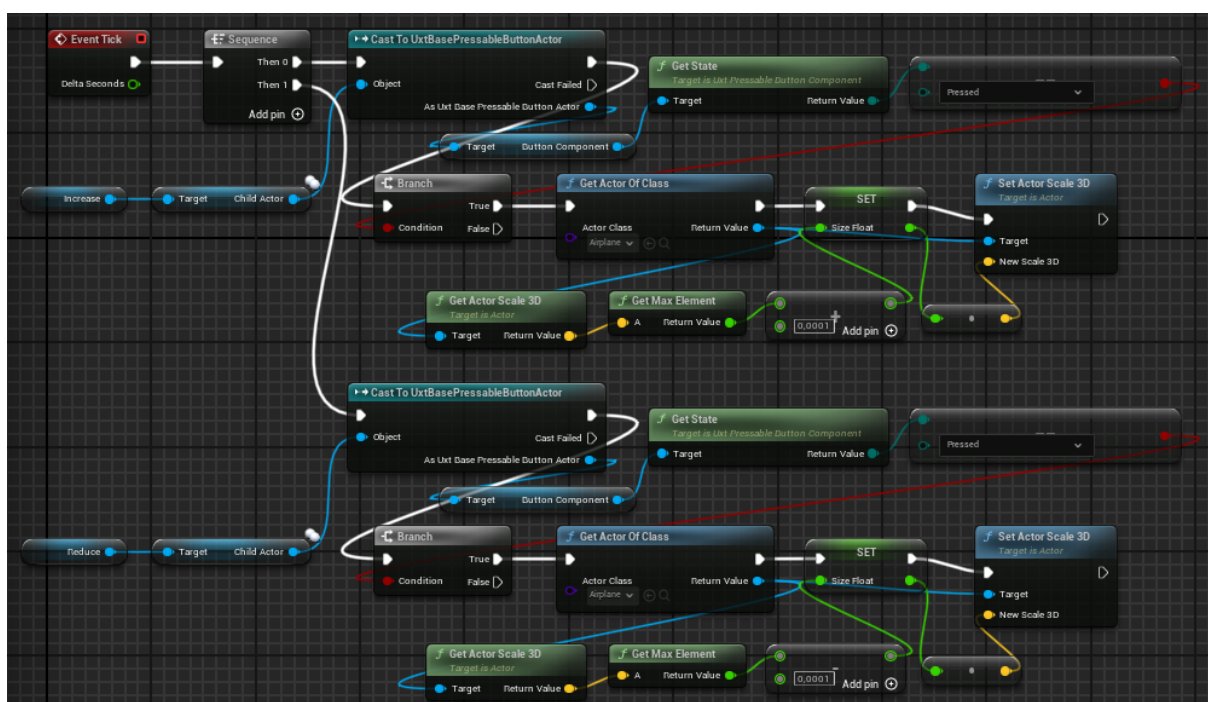
Slika 55. Prikaz *Event Graph*-a sučelja *Move* za translaciju zrakoplova (dio translacije po negativnoj *x* osi) u *Unreal Engine-u*

#### 6.4.5.3. Tipkalo *Scale*

Tipkalo *Scale* je *Uxt Pressable Button Actor* tipkalo koje korisnicima omogućuje skaliranje, odnosno promjenu volumena zrakoplova, ukoliko, kao i u slučaju rotacije i translacije, ne žele imati direktnu interakciju sa zrakoplovom. Poput korisničkih sučelja *RotatePlate* te *MovePlate*, sučelje *ScalePlate* stvara se na uvijek istoj relativnoj udaljenosti od korisničkog sučelja zrakoplova [Slika 56]. *ScalePlate* je sučelje na kojem se nalaze dva tipkala za smanjivanje i povećavanje volumena zrakoplova, a *Event Tick* čvor ponovno djeluje kao svojevrsna *while* petlja koja provjerava stanje tipkala [Slika 57].



Slika 56. Prikaz sučelja *Scale* za skaliranje zrakoplova nakon pokretanja sesije u *Unreal Engine-u*



Slika 57. Prikaz *Event Graph*-a sučelja *Scale* za skaliranje zrakoplova u *Unreal Engine-u*

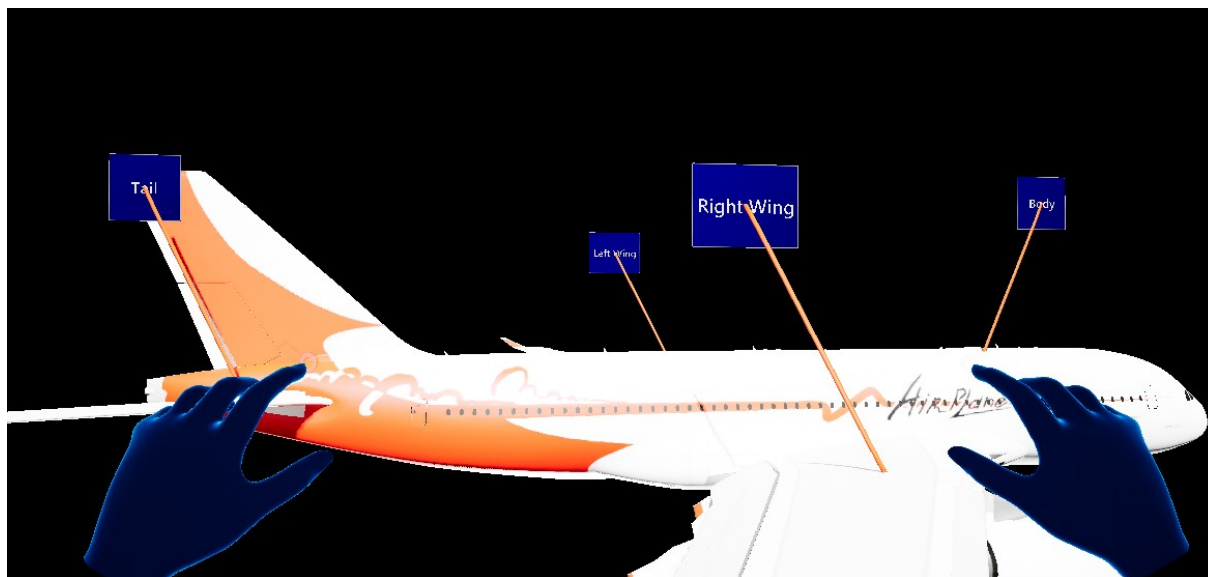
#### 6.4.5.4. Tipkalo *Show Parts*

Posljednje od tipkala koje se nalazi na korisničkom sučelju za zrakoplov je tipkalo na koje se klikom prikažu sučelja oko zrakoplova, s krivuljom usmjerenom prema svakom od njegovih dijelova [Slika 58][Slika 59]. Ova se sučelja kreiraju zajedno sa zrakoplovom te su inicijalno

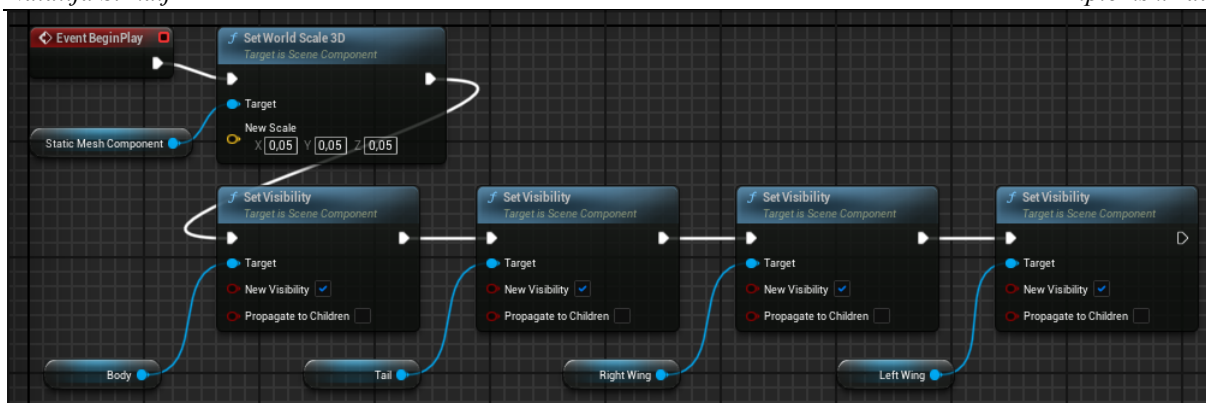
skrivena u prostoru [Slika 60]. Svako od sučelja također sadrži tipkalo te se pritiskom na neko od njih pojavi željeni dio zrakoplova (detaljnije opisano u poglavlju 6.4.6.). Prikazivanje ovih sučelja vrši se funkcijom *Set Visibility*, pri čemu zrakoplov predstavlja roditelja, a sučelja koja ga okružuju predstavljaju djecu [Slika 60].



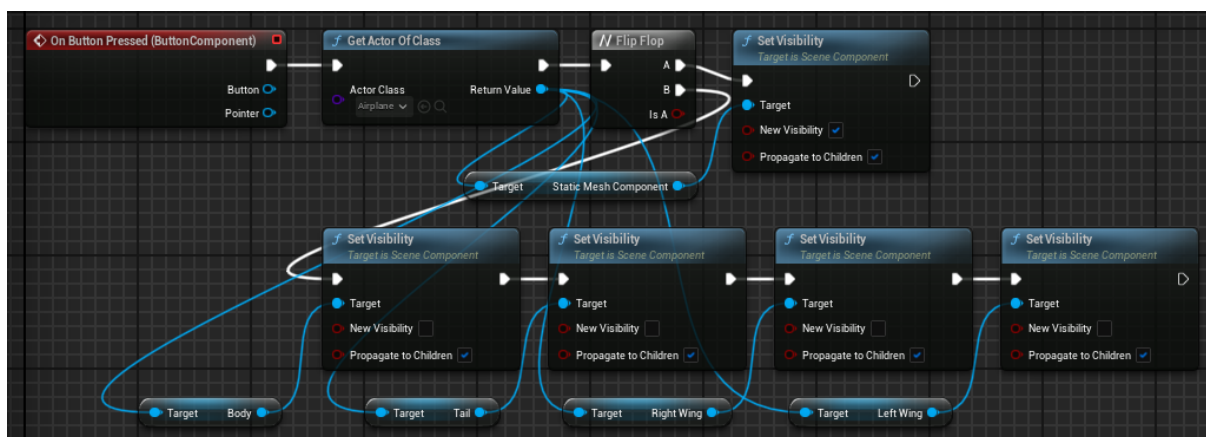
Slika 58. Prikaz sučelja oko zrakoplova nakon pokretanja sesije u *Unreal Engine-u* (1)



Slika 59. Prikaz sučelja oko zrakoplova nakon pokretanja sesije u *Unreal Engine-u* (2)



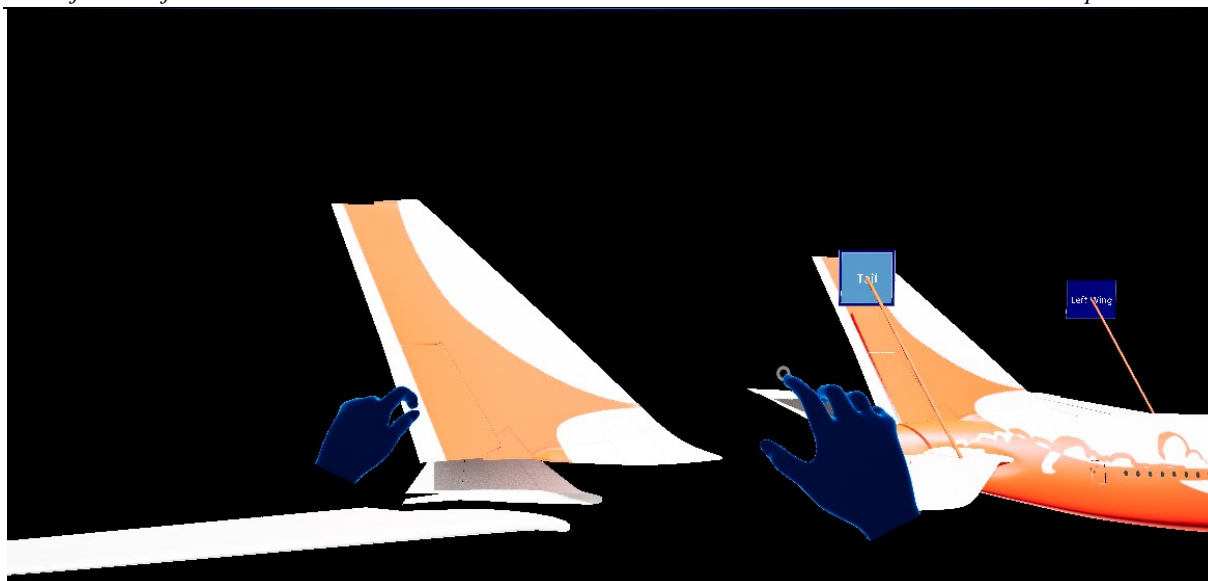
Slika 60. Prikaz *Event Graph*-a početnih postavka zrakoplova u *Unreal Engine*-u



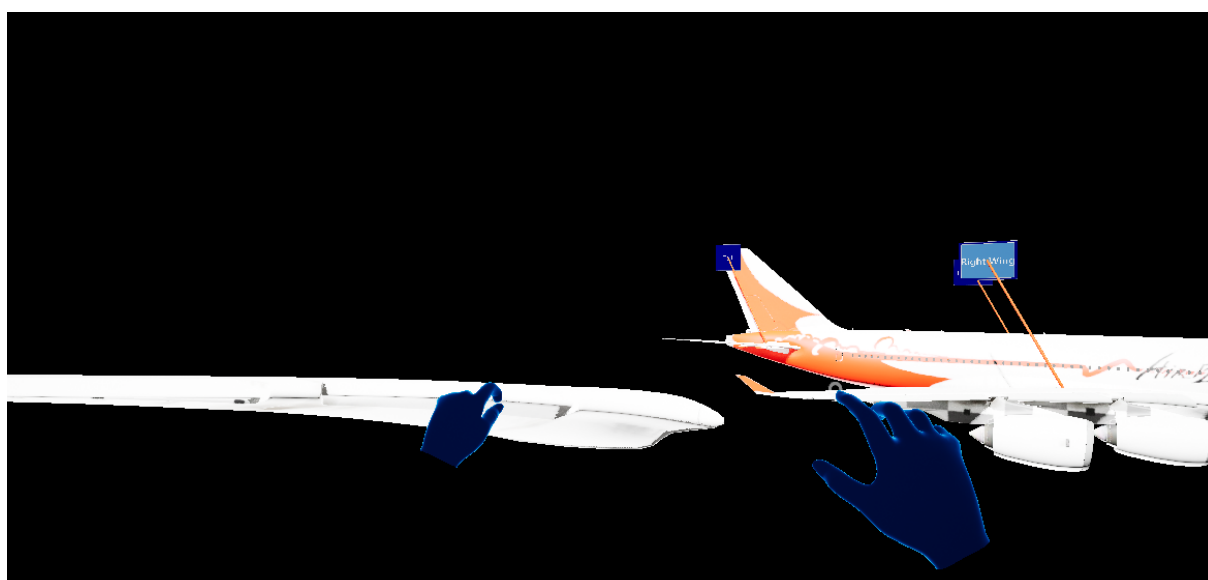
Slika 61. Prikaz *Event Graph*-a tipkala *Show Parts* u *Unreal Engine*-u

#### 6.4.6. Prikaz dijelova zrakoplova

Svaki se dio zrakoplova pojavljuje na uvijek istoj relativnoj udaljenosti od cijelog modela. Radnja se ponovno vrši na način da se pomoću globalne varijable, u ovom slučaju *GetAirplaneLocation*, dohvati lokacija zrakoplova te pohrani u vektor *GetLocation* (opisano u poglavlju 6.4.4.). Dijelovima je potrebno omogućiti mobilnost te dodati komponentu *UxtGenericManipulator*. U nastavku su prikazani rep i desno krilo zrakoplova stvoreni pomoću čvora *Spawn Actor from Class*, klika na tipkalo *Tail* [Slika 62] te klika na tipkalo *Right Wing* [Slika 63].



Slika 62. Prikaz repa zrakoplova stvorenog u prostoru nakon pokretanja sesije u *Unreal Engine-u*



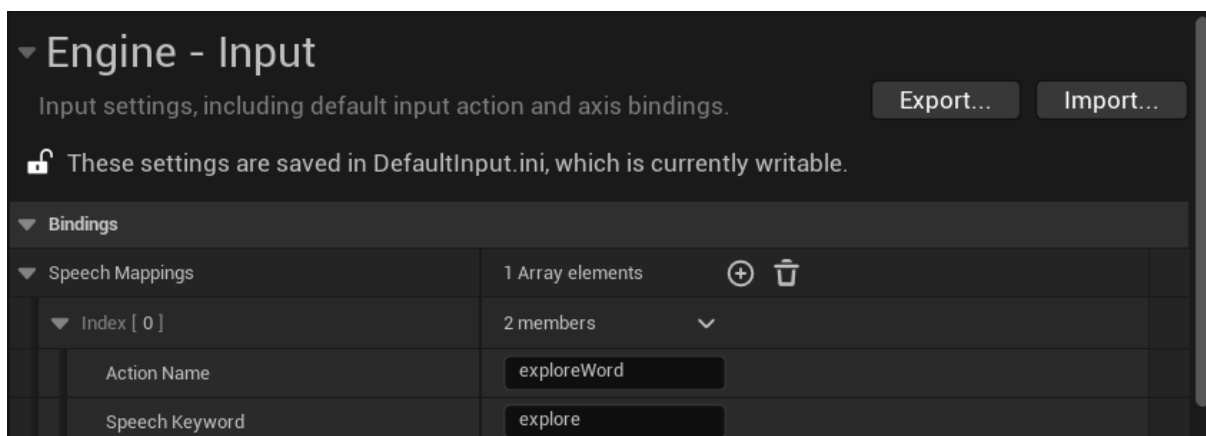
Slika 63. Prikaz desnog krila zrakoplova stvorenog u prostoru nakon pokretanja sesije u *Unreal Engine-u*

#### 6.4.7. Uporaba glasovne naredbe „Explore“

Posljednji korak u sklopu izrade ove aplikacije je uporaba glasovne naredbe kako bi se pristupilo službenoj internetskoj stranici zrakoplova *Airbus A340* [39]. Glasovne naredbe su dobra alternativa naredbama koje se ostvaruju dodiranjem ukoliko su nam ruke trenutno zauzete, no postoji problem ako je okruženje u kojem se nalazimo bučno. Za potrebe glasovnih naredba potrebno je osposobiti prepoznavanje glasa (engl. *Speech Recognition*) koje se nalazi u postavkama *Windows-a*. Ako se *Speech Recognition* nije prethodno upotrebljavao, postoji

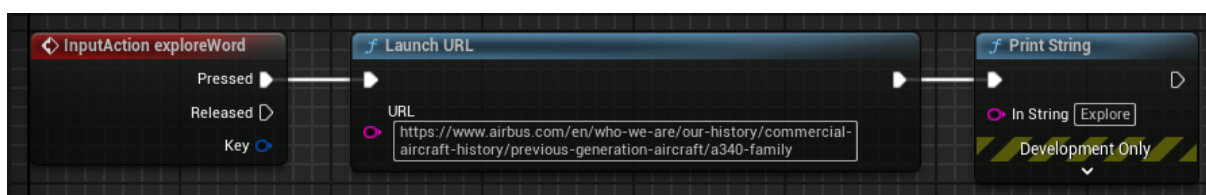
možnost da će trebati proći neko vrijeme dok sustav nauči prepoznavati glasovne unose, budući da je u pitanju umjetna inteligencija.

Klikom na *Edit*, a zatim *Project Settings* otvara se prozor s postavkama projekta, pri čemu je potrebno modificirati dio *Input* koji se nalazi unutar postavke *Engine*. Kako bi se naredbi moglo pristupiti unutar *Blueprint* skripte, potrebno joj je dati naziv te napisati riječ koja se želi upotrijebiti u sklopu glasovne naredbe [Slika 64].

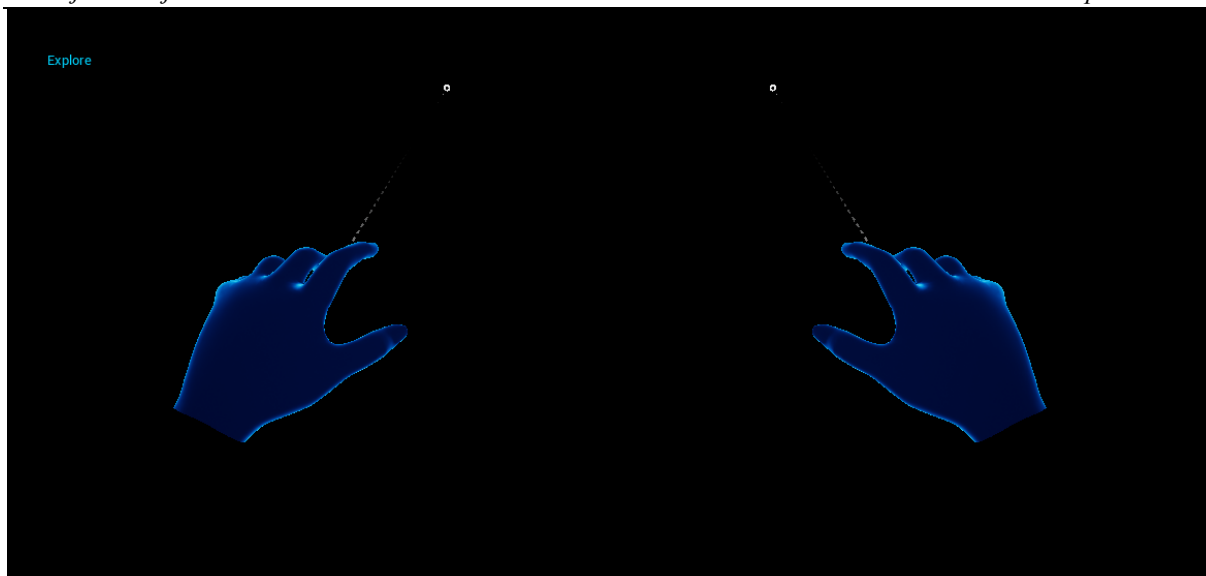


**Slika 64.** Prikaz modifikacije *Project Settings* prozora za kreiranje glasovne naredbe u *Unreal Engine-u*

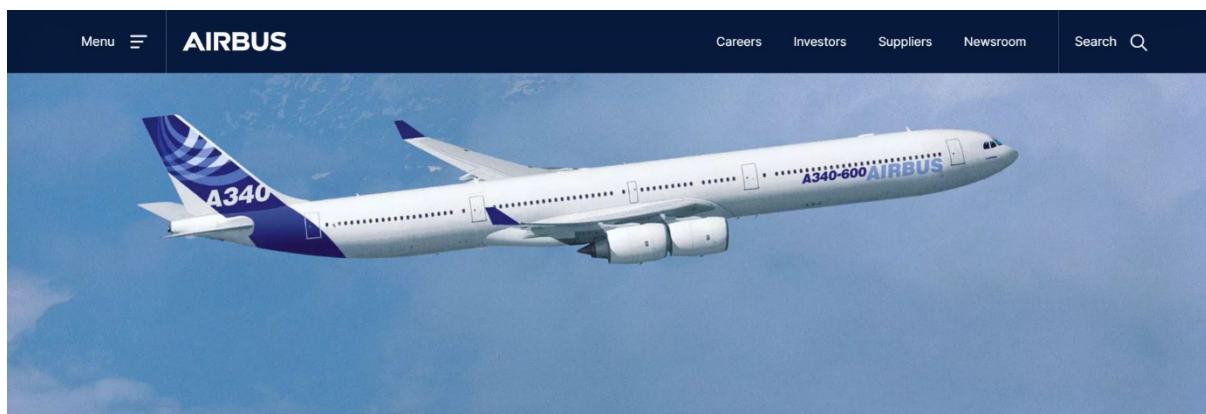
Unutar *Level Blueprint* skripte potrebno je dodati *InputAction* čvor naziva kojeg smo stavili pod *Action Name* u postavkama projekta te funkcijski čvor *Launch URL* unutar kojeg je potrebno upisati željenu poveznicu [Slika 65]. Funkcija *Print String* postavljena je za potrebu izvršene naredbe. Ispis riječi *Explore* te slika pristupljenoj internetskoj stranici prikazane su u nastavku [Slika 66][Slika 67].



**Slika 65.** Prikaz dijela *Level Blueprint* skripte za omogućavanje glasovne naredbe u *Unreal Engine-u*



Slika 66. Prikaz ispisa izvršene glasovne naredbe nakon pokretanja sesije u *Unreal Engine-u*



Slika 67. Prikaz internetske stranice [literatura] kojoj se pristupilo nakon izvršene glasovne naredbe nakon pokretanja sesije u *Unreal Engine-u*

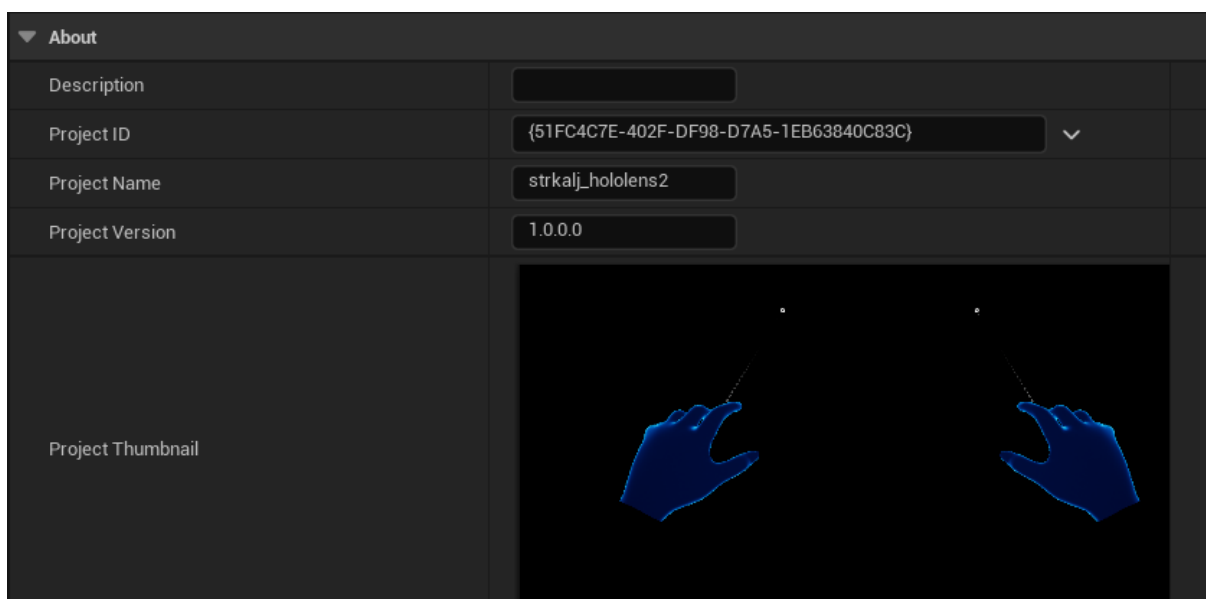


## 7. EKSPERIMENTALNA EVALUACIJA RAZVIJENOG 3D SVIJETA

### 7.1. Pakiranje projekta za *HoloLens 2*

Kako bi se kreirani projekt za mješovitu stvarnost mogao spakirati za *HoloLens 2* uređaj, potrebno je napraviti određene korake koji prethode samom pakiranju.

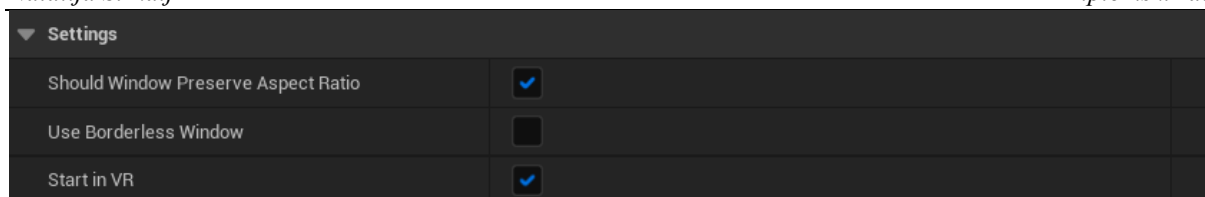
Unutar *Project Settings*-a, *Projects* te *Description*, u dijelu *About (Project Name)*, potrebno je dati ime projektu [Slika 68], a zatim je u dijelu *Publisher (Company Name* te *Company Distinguished Name)* potrebno unijeti potrebne informacije prikazane slikom [Slika 69]. U dijelu *Settings* gdje se nalazi i *Publisher*, potrebno je označiti *Start in VR* [Slika 70].



Slika 68. Prikaz dodavanja imena projekta za pakiranje unutar *Project Settings* prozora u *Unreal Engine-u*

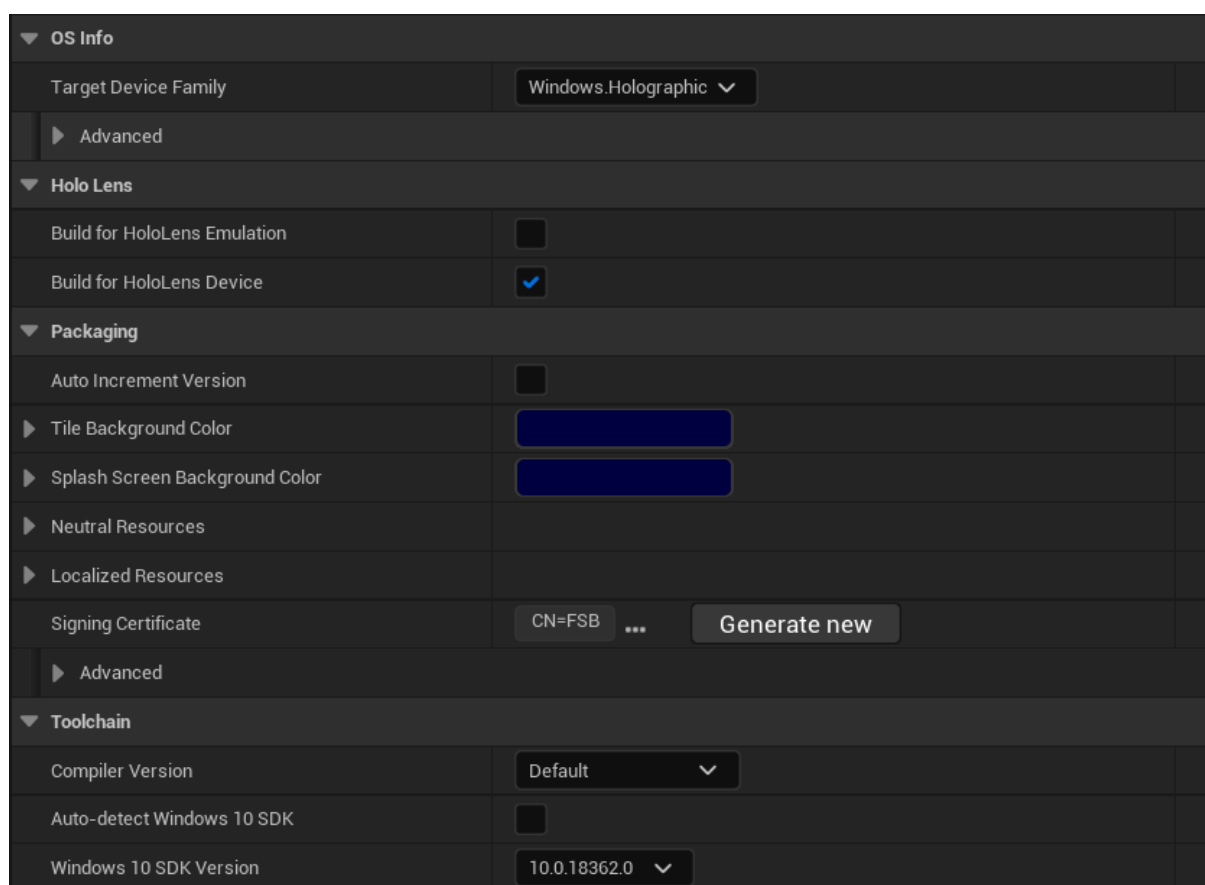


Slika 69. Unos potrebnih informacija za *Company Name* te *Company Distinguished Name* unutar *Project Settings* prozora u *Unreal Engine-u*



**Slika 70. Označavanje opcije *Start in VR* unutar *Project Settings* prozora u *Unreal Engine-u***

Sljedeći je korak da se unutar *Platforms* te *HoloLens* u dijelu *Packaging* pod *Signing Certificate* klikne *Generate new* te se, unutar novog prozora, odabere *None* (ukoliko se ne želi kreirati privatna lozinka). U dijelu *Holo Lens* potrebno je odabrati *Build for HoloLens Device*, budući da pakiramo projekt za učitavanje na HMD [Slika 71]. Ukoliko se želi raditi s *HoloLens* emulatorom koji je računalna zamjena za HMD, označi se i dio *Build for HoloLens Emulation*. Valja obratiti pozornost i na verziju *Windows 10 SDK* koja se nalazi na dnu prikazane slike.



**Slika 71. Modifikacija potrebnih informacija za *HoloLens* te *Signing Certificate* unutar *Project Settings* prozora u *Unreal Engine-u***

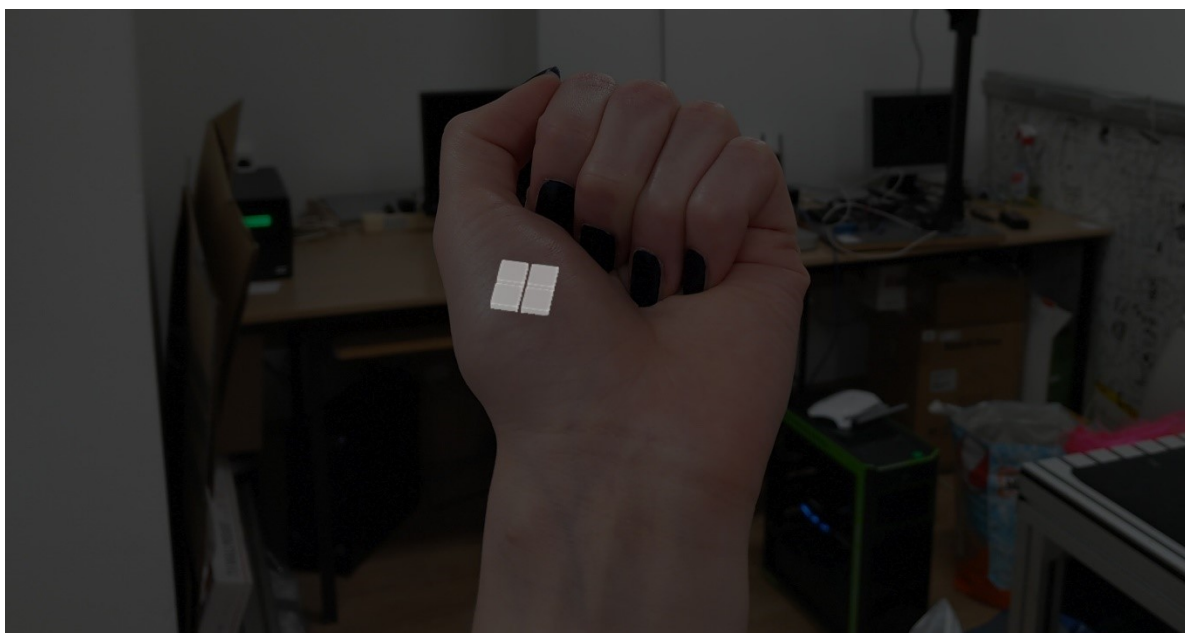
Nakon što se provedu svi potrebni koraci, aplikacija je spremna za pakiranje. Klikom na tipkalo *Platforms* smješteno na alatnoj traci pri vrhu, otvara se izbornik u kojem je potrebno odabrati *HoloLens*, a zatim *Package Project*. Paket se sprema na željenu lokaciju.

## 7.2. Učitavanje paketa na *HoloLens 2* uređaj putem *Windows Device Portal-a*

Unutar postavka *Windows-a* 10, klikom na *Privacy & Security*, zatim *For Developers*, na *HoloLens* uređaju je potrebno omogućiti *Developer Mode* i *Device Portal*. Spajanjem računala i *HoloLens-a* na istu mrežu unosom *Wi-Fi* adrese (ili uporabom *USB-a*), može se pristupiti *Windows Device Portal-u*, pri čemu je prvo potrebno kreirati račun, ukoliko je nepostojeći. Zatim se u dijelu *Apps* unutar *Views*, vrši prebacivanje paketa na *HoloLens*, na način da se klikom na *Choose File* odabere *APPXBUNDLE* tip datoteke te je, nakon označivanja opcije za odabir izbornih paketa, ponovnim klikom na *Choose File* potrebno odabrati *APPX* tip datoteke i u konačnici kliknuti na *Install*. Detaljniji postupak učitavanja paketa na *HoloLens* uređaj kao i uporaba *Windows Device Portal-a* opisan je na službenoj *Microsoft-ovoj* stranici [40].

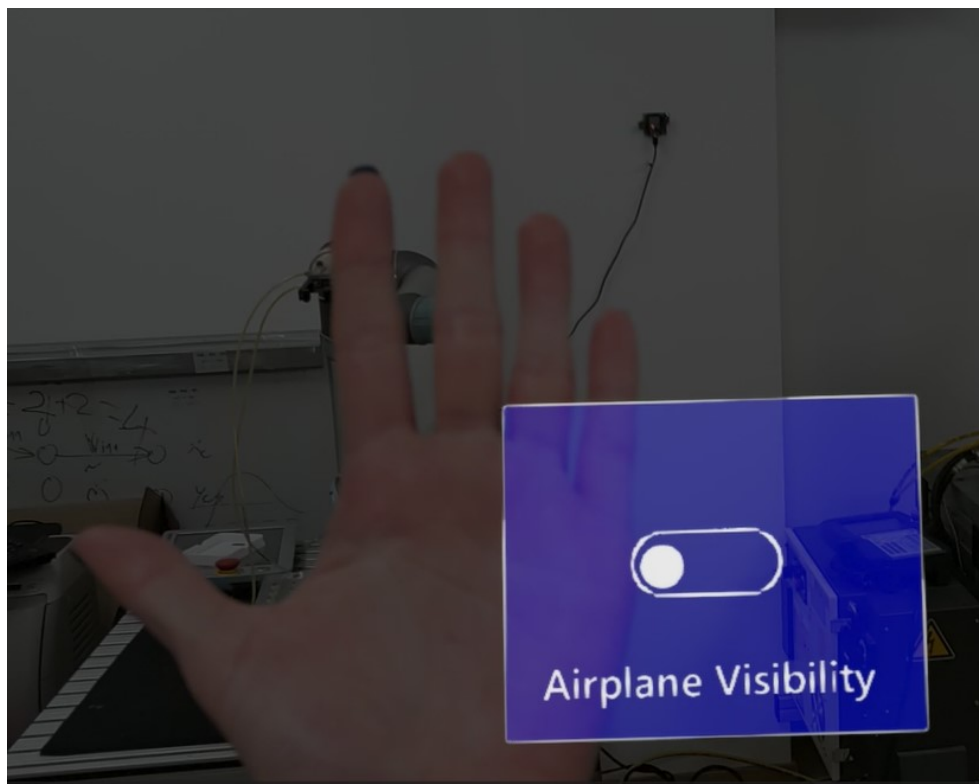
## 7.3. Pokretanje aplikacije mješovite stvarnosti na *HoloLens 2* uređaju

Nakon učitavanja paketa, potrebno je pokrenuti aplikaciju mješovite stvarnosti na *HoloLens 2* HMD. Aplikaciji se može pristupiti klikom na *Windows* ikonicu koja se pojavi u području unutarnjeg dijela zgloba ruke nakon što se isti okrene prema kameri [Slika 72]. Klikom na spomenutu ikonicu, prvo se otvara početno korisničko sučelje u kojem se nalazi tipkalo za postavke, *Microsoft Edge*, *Microsoft Store*, kameru, slike itd., kao i tipkalo *All Apps* koje vodi do željene aplikacije.



Slika 72. Prikaz *Windows* ikonice *HoloLens* uređaja nakon pokazivanje unutarnjeg dijela zgloba

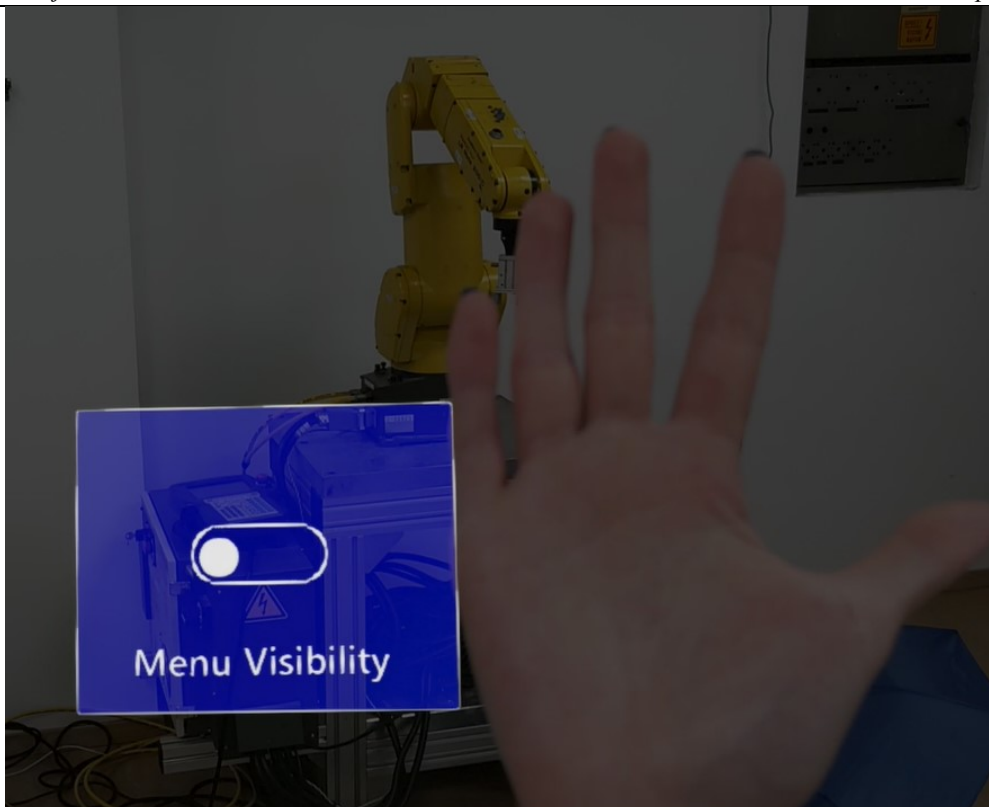
U nastavku su prikazane slike [Slika 73][Slika 74][Slika 75][Slika 76][Slika 77][Slika 78][Slika 79] snimljene pomoću kamera smještenih na prednjoj strani uređaja.



**Slika 73.** Prikaz virtualnog tipkala *Airplane Visibility* u sustavu mješovite stvarnosti, nakon pokazivanja lijevog dlana kameri



**Slika 74.** Prikaz virtualnog modela zrakoplova u sustavu mješovite stvarnosti, nakon klika na tipkalo *Airplane Visibility*



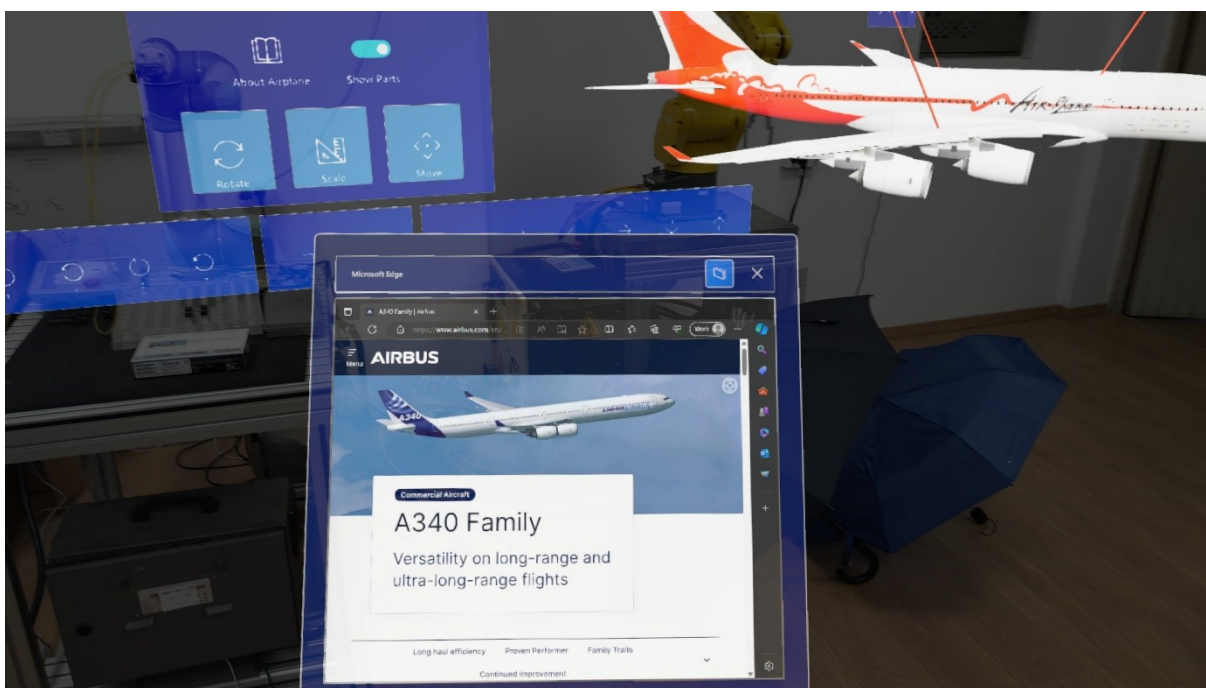
Slika 75. Prikaz virtualnog tipkala *Menu Visibility* u sustavu mješovite stvarnosti, nakon pokazivanja desnog dlana kameri



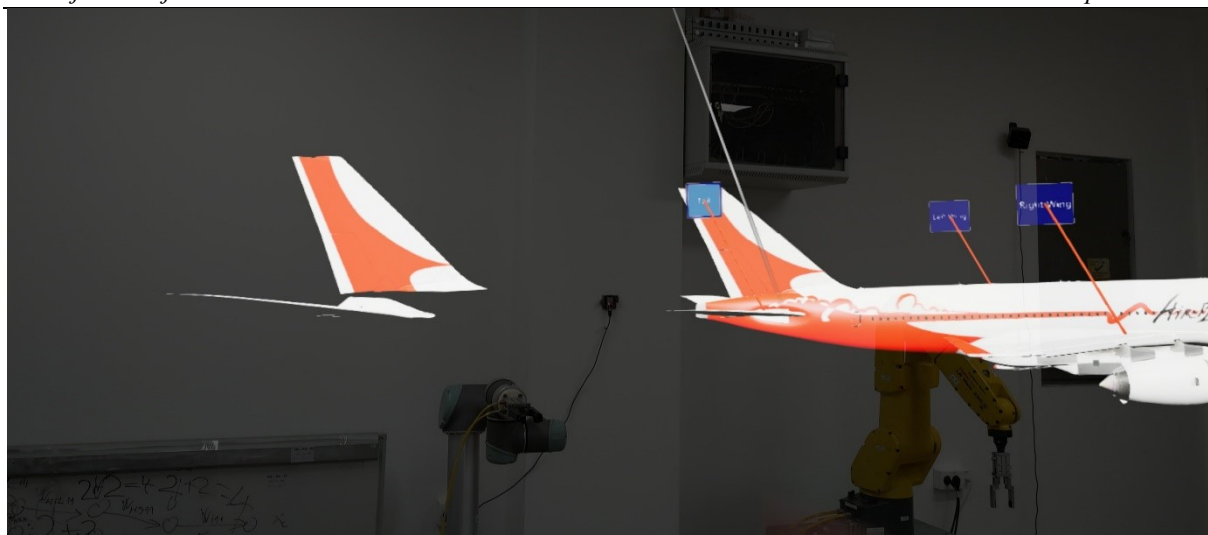
Slika 76. Prikaz virtualnog sučelja zrakoplova u sustavu mješovite stvarnosti nakon klika na tipkalo *Menu Visibility* te prozora za rotaciju, translaciju i skaliranje zrakoplova



Slika 77. Prikaz virtualnih sučelja koji pokazuju dijelove zrakoplova u sustavu mješovite stvarnosti nakon klika na tipkalo *Show Parts*



Slika 78. Prikaz virtualne internetske stranice u sustavu mješovite stvarnosti nakon klika na tipkalo *About Airplane*



**Slika 79. Prikaz virtualnog repa zrakoplova u sustavu mješovite stvarnosti nakon klika na tipkalo *Tail***

#### 7.4. Kritički osvrt

Prilikom pakiranja aplikacije za *HoloLens* uređaj, pojavilo se mnogo upozorenja, grešaka ili informacija kako se neke radnje nisu uspjele dovršiti, unatoč tome što je u *Unreal Engine*-u sve uredno funkcioniralo. Neuspjele radnje uglavnom su bile vezane za komponente instalirane putem *Visual Studio Build Tools*-a te *Microsoftov*-e dodatke za mješovitu stvarnost, no mogu se javiti i greške vezane uz sam operativni sustav računala, *driver*-e itd. Često se događalo da sustav prilikom pakiranja nije mogao pristupiti raznim datotekama i mapama zbog određenih problema s kompatibilnosti. Teško je popratiti proces otklanjanja pogrešaka, budući da se nakon otklanjanja jedne potencijalno pojavi još nekoliko novih, pri čemu se rijetko kad dobije konkretna informacija o tome kako ih otkloniti. Često se pojavljuju pogreške vezane uz *Windows 10 SDK*, stoga treba provjeravati koju je od navedenih verzija potrebno naknadno instalirati. Nakon procesa otklanjanja pogrešaka i uspješnog prebacivanja aplikacije na *HoloLens* uređaj, ponovno su se javili problemi prilikom samog učitavanja aplikacije putem *Windows Device Portal*-a. Aplikacija se nije mogla učitati, odnosno konstantno se rušila, te je ponovno trebalo utrošiti vremena kako bi se pronašao uzrok ove situacije, a sam postupak je zbog kompleksnosti i dugotrajnosti teško opisati te se treba poslužiti informacijama dostupnima na internetu.

Nakon uklanjanja svih pogrešaka, aplikacija se uspješno učitala. Ručna interakcija s virtualnim elementima često je nezahvalna, budući da *HoloLens* uređaj nekada ne može raspoznati želimo li uhvatiti element ili kliknuti na neki njegov dio. Često je potrebno i po nekoliko puta kliknuti

na virtualno tipkalo kako bi se izvršila željena radnja, što nije nužno do *HoloLens* uređaja, već do *Microsoft*-ovih dodatka za mješovitu stvarnost u sklopu *Unreal Engine*-a. Još jedna negativna karakteristika je ta što se uređaj znatno uspori ukoliko mu se smanji postotak napunjenosti baterije te gotovo da postane nemoguć za rad.

Ukupan dojam je da su *HoloLens* uređaj te *Microsoft*-ovi dodaci za mješovitu stvarnost zanimljivi, a sama zamisao ovog uređaja dobra, no postoji još mnogo prostora za poboljšanje što je pokazatelj koliko su sustavi virtualne i mješovite stvarnosti još uvijek na početku razvoja.

## 7.5. Unaprjeđivanje aplikacije

Aplikacija se može unaprijediti na način da se samostalno izradi CAD model zrakoplova te da se dijelovi grupiraju na prednji, srednji i stražnji dio kao u priručniku za popravak konstrukcije (engl. *Structural Repair Manual*). Svaki bi dio tada mogao biti „obložen“ međusobno okomitim linijama koje bi stvarale svojevrsnu mrežu, čiji bi čvorovi prikazivali lokacije oštećenja. Svako od oštećenja imalo bi dostupne informacije poput onih koje se nalaze u priručniku. Još jedna od zanimljivih karakteristika koja se može nadodati je, primjerice, omogućavanje opcije praćenja unutar *Unreal Engine*-a putem neke glasovne naredbe. Na taj bi se način objekti „izgubljeni“ u prostoru, ukoliko se korisnik puno kreće, mogli odmah stvoriti u njegovom radijusu.



---

## 8. ZAKLJUČAK

Sustavi mješovite stvarnosti predstavljaju veliki korak u vidu tehnološkog napretka, no postoji još mnogo prostora za razvojem i usavršavanjem uređaja koji pružaju ovakva iskustva.

Glede *HoloLens* HMD-a koji se upotrebljavao u skopu ovog diplomskog rada, potrebno je prvenstveno poraditi kod unaprjeđivanja softvera te dodataka za mješovitu stvarnost, budući da se, prema raznim izvorima, *Microsoft* više neće baviti proizvodnjom hardverskih komponenti. Ideja koja leži iza ovog uređaja za mješovitu stvarnost, kao i mnogih drugih uređaja, dobro je zamišljena, no i dalje je poprilično veliki omjer cijene i kvalitete. Ako se uređaji, primjerice, žele upotrebljavati u edukacijske svrhe, potrebno je imati razvojne programere koji će razvijati programe specifičnih namjena, što zahtijeva dodatne financijske troškove.

Unatoč navedenom, potrebno je mnogo znanja i iskustva od strane samih razvojnih programera, koji svojim sposobnostima i idejama mogu učiniti sveukupno iskustvo puno zanimljivijim i ugodnijim za rad, a *HoloLens* bi bio idealna polazna točka kada bi se poradilo na negativnim karakteristikama.

**LITERATURA**

- [1] <https://medium.com/stellarx-en/the-slow-rise-of-mixed-reality-c6e3ef5ad3dd>, pristupljeno 22.4.2024.
- [2] <https://www.britannica.com/technology/virtual-reality>, pristupljeno 22.4.2024.
- [3] <https://www.onirix.com/ar-vs-mr/>, pristupljeno 22.4.2024.
- [4] <https://www.innovae.com/en/the-virtual-reality-technology/>, pristupljeno 22.4.2024.
- [5] <https://www.techradar.com/news/wearables/forgotten-genius-the-man-who-made-a-working-vr-machine-in-1957-1318253>, pristupljeno 22.4.2024.
- [6] <https://www.vistaequitypartners.com/insights/an-introduction-to-immersive-technologies/>, pristupljeno 27.4.2024.
- [7] <https://www.vrs.org.uk/virtual-reality/history.html>, pristupljeno 27.4.2024.
- [8] <https://www.dsourc.in/course/virtual-reality-introduction/evolution-vr/sword-damocles-head-mounted-display>, pristupljeno 27.4.2024.
- [9] <https://medium.com/the-reality-files/the-reality-files-06-63a158e7a894>, pristupljeno 29.4.2024.
- [10] <https://codereality.net/ar-for-eu-book/chapter/introduction/historyar/>, pristupljeno 29.4.2024.
- [11] <https://www.frontiersin.org/articles/10.3389/frvir.2021.647997/full>, pristupljeno 29.4.2024.
- [12] <https://www.amazon.com/Oculus-Rift-Virtual-Reality-Headset-pc/dp/B00VF0IXEY>, pristupljeno 1.5.2024.
- [13] <https://www.bloomberg.com/news/articles/2020-02-04/google-says-developers-can-now-purchase-latest-smart-glasses>, pristupljeno 1.5.2024.
- [14] <https://www.dezeen.com/2016/01/27/daqri-smart-construction-helmet-augmented-reality-wearable-technology/>, pristupljeno 1.5.2024.
- [15] <https://www.bbc.com/news/technology-35686616>, pristupljeno 1.5.2024.
- [16] <https://varjo.com/virtual-augmented-and-mixed-reality-explained/>, pristupljeno 1.5.2024.
- [17] <https://dynamics.microsoft.com/en-us/mixed-reality/guides/what-is-augmented-reality-ar/>, pristupljeno 1.5.2024.
- [18] <https://www.techtarget.com/whatis/definition/augmented-reality-AR>, pristupljeno 1.5.2024.

- 
- [19] <https://learn.microsoft.com/en-us/windows/mixed-reality/discover/mixed-reality>, pristupljeno 1.5.2024.
- [20] <https://virtualspeech.com/blog/vr-applications>, pristupljeno 1.5.2024.
- [21] <https://www.paulreiffer.com/2020/06/ar-see-your-new-fine-art-print-in-your-own-space-with-augmented-reality/>, pristupljeno 1.5.2024.
- [22] <https://www.urtech.ca/2021/03/what-is-holoportation-how-is-it-different-from-augmented-reality-different-mixed-reality-and-microsoft-mesh/>, pristupljeno 1.5.2024.
- [23] <https://www.onirix.com/mixed-reality/>, pristupljeno 1.5.2024.
- [24] <https://mafda.medium.com/ar-101-components-of-the-augmented-reality-system-part-3-878c71e68069>, pristupljeno 1.5.2024.
- [25] <https://medium.com/@beyzael2007/sensors-of-mixed-reality-f49669e890c8>, pristupljeno 2.5.2024.
- [26] <https://blog.vive.com/us/how-depth-sensors-revolutionize-virtual-and-mixed-reality/>, pristupljeno 2.5.2024.
- [27] <https://www.bairesdev.com/blog/what-is-unreal-engine/>, pristupljeno 2.5.2024.
- [28] <https://www.spheregen.com/mixed-reality-headsets/>, pristupljeno 2.5.2024.
- [29] <https://www.amazon.com/Microsoft-Hololens2-64GB-Headset-Device/dp/B08GP98FG7>, pristupljeno 2.5.2024.
- [30] <https://www.microsoft.com/en-us/hololens/hardware#document-experiences>, pristupljeno 2.5.2024.
- [31] <https://learn.microsoft.com/en-us/hololens/hololens2-hardware>, pristupljeno 2.5.2024.
- [32] <https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/research-mode>, pristupljeno 2.5.2024.
- [33] <https://channelxr.com/products/magic-leap-2-enterprise-edition>, pristupljeno 2.5.2024.
- [34] <https://www.forbes.com/sites/moorinsights/2022/10/11/meta-quest-pro--a-high-end-consumer-headset-masquerading-as-enterprise/>, pristupljeno 2.5.2024.
- [35] <https://www.amazon.in/PICO-All-One-128GB-Headset/dp/B0BN49H6WY>, pristupljeno 2.5.2024.
- [36] <https://www.microsoft.com/en-us/hololens/industry-manufacturing>, pristupljeno 2.5.2024.
- [37] <https://www.monarch-innovation.com/microsoft-hololens>, pristupljeno 2.5.2024.
- [38] <https://free3d.com/3d-models/aircraft>, pristupljeno 4.5.2024.
- [39] <https://www.airbus.com/en/who-we-are/our-history/commercial-aircraft-history/previous-generation-aircraft/a340-family>, pristupljeno 4.5.2024.

---

[40] <https://learn.microsoft.com/en-us/windows/uwp/debug-test-perf/device-portal>,  
pristupljeno 4.5.2024.