Prema imerzivnom projektiranju proizvodnih procesa korištenjem tehnika virtualne stvarnosti

Buzjak, Domagoj

Master's thesis / Diplomski rad

2017

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:235:990908

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-25**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Domagoj Buzjak

Zagreb, 2017.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor: Student:

Prof. dr.sc. Zoran Kunica, dipl. ing. Domagoj Buzjak

Zagreb, 2017.

ZADATAK

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur. broj:	

DIPLOMSKI ZADATAK

Student: **Domagoj Buzjak** Mat. br.: 0035187268

Naslov rada na hrvatskom jeziku: **Prema imerzivnom projektiranju proizvodnih procesa korištenjem tehnika virtualne stvarnosti**

Naslov rada na engleskom jeziku: **Towards Immersive Designing of Production Processes Using Virtual Reality Techniques**

Opis zadatka:

Virtualna stvarnost ima sve veći značaj u inženjerskim djelatnostima projektiranja proizvoda, procesa i sustava. Riječ je o uporabi imerzivnog – snažnog interaktivnog alata, koji uključuje vizualizaciju i haptiku za simulaciju – razmatranje i vrednovanje prethodno vjerno oblikovanih tvorevina, u jednako vjerno dočaranom okružju. Virtualna stvarnost tako omogućava kvalitetnije rezultate projektiranja; pa je s jedne strane, prirodan tehnološki razvoj djelatnosti projektiranja, dok s druge podrazumijeva prevrednovanje i provjeru tradicionalnih postavki i načina projektiranja.

U radu je potrebno:

1. opisati pojam virtualne stvarnosti i njezin značaj za djelatnost inženjerskog projektiranja
2. opisati trenutačno stanje razvijenosti sustava virtualne stvarnosti
3. opisati opremu za virtualnu stvarnost koja je na raspolaganju za korištenje
4. odabrati sadržaje projektiranja proizvodnih procesa za koje bi se istražila mogućnost primjene virtualne stvarnost (naprimjer, radnje rukovanja pri montaži ili demontaži: jednoruki i dvoruki rad te timski rad, manualni i čovjek-robot)
5. za sadržaje odabrane u prethodnoj točki, razviti virtualne procese povezujući ih sa stvarnim
6. usporediti rezultate dobivene primjenom virtualne stvarnosti s onima koji se dobivaju klasičnim projektiranjem.

Zadatak zadan:	Rok predaje rada:	Predviđeni datum obrane:
11. svibnja 2017.	13. srpnja 2017.	19., 20. i 21. srpnja 2017.

Zadatak zadao:

Prof. dr.sc. Zoran Kunica

Predsjednica Povjerenstva:

Prof. dr. sc. Biserka Runje

IZJAVA

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

Zahvaljujem brojnim ljudima koji su na bilo koji način pomogli u realizaciji ovog projekta. Hvala Darianu Škarici iz tvrtke DivIT na pozajmljenoj opremi, Fakultetu i djelatnicima Računskog centra, posebice Nikoli Zduniću, na osposobljavanju računala preporučenih specifikacija i svom uloženom vremenu.

Posebnu zahvalu iskazujem mentoru, profesoru Zoranu Kunici na brojnim savjetima, zalaganju, iskazivanju interesa, pomoći s opremom i pruženom vremenu, a najviše na izvrsnim idejama i zamislima koje su pomogle u koncepciji ovog rada i njegovoj realizaciji.

Za kraj, zahvalu upućujem svim prijateljima koji su mi bili podrška na bilo koji način tijekom studiranja, kao i mojoj obitelji koja je bezuvjetno stajala uz mene u svakom trenu mog života.

Od srca Vam hvala!

U Zaboku, 11. srpnja 2017.

_____

Domagoj Buzjak

SAŽETAK

Rad donosi opis i pregled područja virtualne stvarnosti, uz naglasak na imerzivno projektiranje radnih procesa. Korištena platforma u prvom dijelu rada uključuje niskobudžetni Google Cardboard i mobilni telefon sustava Android. U drugom dijelu rada korišteni su naglavne naočale virtualne stvarnosti Oculus Rift te kao senzor pokreta kontroler Leap Motion. Kako bi se definiralo virtualno okruženje, opisana je problematika pretvorbe formata i proširenja biblioteke programskog paketa Unity. Također, detaljno je prikazan postupak razvoja imerzivnog okruženja u Unityu, u kojem su izvršeni zadaci umetanja osovine u provrt i sklapanja zidne priključnice. Na primjeru zidne priključnice mjereno je vrijeme trajanja sklapanja u virtualnom okruženju, koje je poslužilo za usporedbu s vremenom dobivenim sustavom unaprijed određenih vremena MTM-2, kao tradicionalnim načinom projektiranja ručnih radnji sklapanja. Uz analizu rezultata, iznijete su prednosti i nedostaci imerzivnog projektiranja putem tehnika virtualne stvarnosti.

Ključne riječi: virtualna stvarnost, imerzivno projektiranje, MTM, Oculus Rift, Leap Motion, Unity

SUMMARY

The diploma thesis provides a description and an overview of virtual reality, with an emphasis on immersive designing of work processes. In the first part of the work, the low-end platform consisting of Google Cardboard and Android mobile phone was used. In the second part, Oculus Rift virtual reality glasses were used, as well as the motion sensor controller Leap Motion. In order to define the virtual environment, the issue of format conversion and library expansion of Unity software is described. Also, the process of developing of an immersive environment in the Unity software is depicted in detail, including solving of a peg in hole and wall socket assembly tasks. In the example of an assembly of the wall socket, the task execution time in the virtual environment was measured and then compared with the results of a traditional planning approach which uses predetermined motion time system MTM-2. In addition to the analysis of the results, the advantages and disadvantages of the immersive designing using virtual reality techniques are presented.

Key words: virtual reality, immersive planning, MTM, Oculus Rift, Leap Motion, Unity

S A D R Ž A J

ZADATAK 1
IZJAVA 2
SAŽETAK 3
SUMMARY 4
Popis oznaka i mjernih jedinica fizikalnih veličina 7
Popis slika 8
Popis tablica 10
Popis izlista 11

1. UVOD 12

2. SUSTAVI VIRTUALNE STVARNOSTI I NJIHOVA PERSPEKTIVA 16

2.1. IMERZIVNOST 16
2.2. SUSTAVI VIRTUALNE STVARNOSTI 20
2.2.1. Google Cardboard 20
2.2.2. Oculus VR 22
2.3. PERSPEKTIVA SUSTAVA VIRTUALNE STVARNOSTI 23
2.4. RAČUNALNO POSREDOVANA STVARNOST 24
2.5. ZDRAVSTVENE POTEŠKOĆE – BOLEST VIRTUALNE STVARNOSTI 26

3. POVEZIVANJE VIRTUALNE STVARNOSTI I CAD-a 28

3.1. STVARANJE GEOMETRIJE I PRETVORBA FORMATA 30
3.2. Autodesk 3ds Max 32
3.3. SOFTVER UNITY 37
3.3.1. Proširenje biblioteke i definiranje razvojne platforme 38
3.3.2. Uvoz konvertirane geometrije 45
3.4. IZRADA APLIKACIJE ZA PLATFORMU GOOGLE CARDBOARD 48
3.4.1. *Android Developers* 48
3.4.2. *Unity app builder* 52
3.5. RAZVOJ APLIKACIJE VIRTUALNE STVARNOSTI ZA PLATFORMU OCULUS 68
3.5.1. Oculus DK2 68
3.5.2. Leap Motion 70

3.5.3. Povezivanje Unityja, Oculusa i kontrolera Leap Motion 72

4. PROJEKTIRANJE PROIZVODNIH PROCESA TEHNIKAMA VIRTUALNE STVARNOSTI 83

4.1. UMETANJE OSOVINE U PROVRT 84

4.2. SKLAPANJE ZIDNE PRIKLJUČNICE 90

4.2.1. Opis i struktura zidne priključnice 90

4.2.2. Elementi rada i plan montaže 94

4.2.3. Definiranje radnog okruženja za virtualno sklapanje 97

4.2.4. Analiza i određivanje vremena sklapanja sustavom MTM 106

4.2.5. Virtualno sklapanje 112

4.2.6. Usporedba rezultata dobivenih sustavom MTM i virtualnim sklapanjem 114

4.3. JEDNORUKI I DVORUKI RAD 117

4.4. TIMSKI RAD 117

4.5. RAD ČOVJEK-ROBOT 118

5. ZAKLJUČAK 120

6. LITERATURA 122

PRILOG: CD

Popis oznaka i mjernih jedinica fizikalnih veličina

Oznaka	**Opis**	**Mjerna jedinica**
	duljina	mm, cm, m
	napon	V
	frekvencija	Hz
	jakost struje	A
	vrijeme	s
CAD	konstruiranje pomoću računala	
MTM	eng. *Methods-Time Measurement* (Metode – mjerenje vremena; sustav unaprijed određenih vremena)	
eng.	engleski	
USD	američki dolar	USD
VR	eng. *Virtual Reality* –- virtualna stvarnost	
TMU	eng. *Time Measurement Unit* (mjerna jedinica vremena sustava MTM)	

Popis slika

Slika 1. Oculus Touch [5] 19
Slika 2. Detaljni prikaz postupka sastavljanja *Google Cardboarda* [10] 21
Slika 3. Sklopljeni sustav Google Cardboard [11] 21
Slika 4. Oculus Rift CV1 [5] 22
Slika 5. Prihodi potrošačkih sustava VR u 2016. godini i projekcija za 2020. godinu [14] 24
Slika 6. Računalno posredovana stvarnost [18] 25
Slika 7. Shema konverzije podataka od CAD-a do VR-a 31
Slika 8. Autodesk 3ds Max 32
Slika 9. Uvoz CAD geometrije u Autodesk 3ds Max 33
Slika 10. Postavke uvoza u Autodesk 3ds Max 34
Slika 11. Dio uvezen u Autodesk 3ds Max 35
Slika 12. Postavke izvoza 36
Slika 13. Odredišni direktorij 37
Slika 14. Uvoz paketa u Unity 39
Slika 15. Lokacija paketa 40
Slika 16. Paket GoogleVRForUnity 41
Slika 17. Postavke *Build* 42
Slika 18. Promjena platforme 43
Slika 19. Odabir *Google Cardboard* platforme 44
Slika 20. Lokacija datoteke na disku 45
Slika 21. Koljenasto vratilu u Unityju 46
Slika 22. Definiranje dimenzija 47
Slika 23. *Build* broj 49
Slika 24. Programerske opcije 50
Slika 25. *USB Debugging* 51
Slika 26. Prikaz koljenastog vratila u Unity *Game*u 52
Slika 27. Skaliranje podloge 53
Slika 28. Dodavanje značajke *GvrMain* 55
Slika 29. Kreiranje nove skripte 56
Slika 30. Programski kod u Visual Basicu 57
Slika 31. Dodavanje skripte objektu *Pomicanje* 61
Slika 32. Odabir mehanizma kretanja 62
Slika 33. Simulacija Cardboard 3D prikaza 1 63
Slika 34. Simulacija Cardboard 3D prikaza 2 63
Slika 35. Simulacija Cardboard 3D prikaza 3 64
Slika 36. Odabir odredišnog direktorija 64

Slika 37. Izrađena aplikacija 65
Slika 38. Kontroler Leap Motion [34 i 35] 67
Slika 39. Kontroler Leap Motion montiran na OSVR headset [34] 67
Slika 40. Oculus DK2 [5] 69
Slika 41. Kamera za praćenje položaja 69
Slika 42. Zajedničko korištenje kontrolera Leap Motion i Oculusa [38] 71
Slika 43. Ilustracija radnog pojasa kontrolera Leap Motion [38] 71
Slika 44. Međusobno povezivanje komponenti sustava za stvaranje imerzivnog okruženja . 73
Slika 45. Uključivanje podrške za virtualnu stvarnosti 74
Slika 46. Prikaz koordinatnog sustava kontrolera Leap Motion [38] 75
Slika 47. Proširivanje biblioteke Unity 76
Slika 48. Paket za proširivanje biblioteke 76
Slika 49. LMHeadMountedRig prefab 77
Slika 50. Prikaz Mesh i Box Collidera 80
Slika 51. Omogućivanje interakcije u imerzivnom okruženju 81
Slika 52. Umetanje osovine u provrt 83
Slika 53. Sklapanje zidne priključnice 83
Slika 54. Odstupanje u pozicioniranju [41] 85
Slika 55. Lociranje radnih komponenti u virtualnom okruženju 86
Slika 56. Prihvat i pozicioniranje osovine iznad provrta 86
Slika 57. Poravnanje dijelova u smjeru vertikalne *Y* osi 87
Slika 58. Nemogućnost izvršavanja zadatka umetanja 88
Slika 59. Pokušaj umetanja nakon smanjenja dimenzije osovine 88
Slika 60. Zračnost između komponenata 89
Slika 61. Ostvareni sklop 90
Slika 62. Sklapanje ugradbenih elemenata zidne priključnice [42] 96
Slika 63. Definirano virtualno radno okruženje 97
Slika 64. Virtualna radna površina 97
Slika 65. Pozicija sklapanja i već sklopljena zidna priključnica (prikazana i uvećano) 99
Slika 66. Okvir vanjskog kućišta u kutiji s dijelovima 102
Slika 67. Hvatanje okvira vanjskog kućišta 103
Slika 68. Pozicioniranje vanjskog kućišta na radnoj površini 104
Slika 69. Radna površina s odloženim Sklopom 5 104
Slika 70. Spajanje vanjskog kućišta 105
Slika 71. Umetanje vijka 105
Slika 72. Simulacija pritezanja vijka 106
Slika 73. Modeli odlučivanja za: uzimanje (lijevo) i postavljanje (desno) [43] 108
Slika 74. Serija slika iz video isječka virtualnog sklapanja zidne priključnice 113
Slika 75. Robot Fanuc LR-Mate 200iC 118

Popis tablica

Tablica 1. Razine virtualne stvarnosti [4] 18
Tablica 2. Strukturna sastavnica zidne priključnice, List 1/2 [44] 92
Tablica 3. Strukturna sastavnica zidne priključnice, List 2/2 [44] 93
Tablica 4. Plan montaže zidne priključnice [44] 95
Tablica 5. Elementi sustava MTM-2 [43] 107
Tablica 6. Normativne vrijednosti sustava MTM-2 [43] 108
Tablica 7. Podjela duljine pokreta u sustavu MTM-2 [43] 109
Tablica 8. Analiza rada putem MTM-2 postupka, List 1/2 110
Tablica 9. Analiza rada putem MTM-2 postupka , List 2/2 111
Tablica 10. Mjerenje trajanja imerzivnog procesa sklapanja, element rada 7. 114

Popis izlista

Izlist 1. Skripta Autowalk [33] 60
Izlist 2. Skripta kolizije 101
Izlist 3. Suprotna skripta kolizije 102

1. UVOD

Virtualna stvarnost[1] je računalno stvoreno sučelje koje imitira zbilju te omogućuje interakciju korisnika u tom zamišljenom svijetu. Logično je postaviti temeljna pitanja poput: zašto bi netko realan svijet mijenjao za virtualni svijet koje stvara računalo, te ako to već čini, koje su prednosti i pogodnosti koje taj svijet pruža u odnosu na realnost, odnosno svijet koji poznajemo. Na ta, ali i mnoga druga pitanja pokušat će odgovoriti ovaj rad, uključujući i konkretan prikaz odabranih radnih – proizvodnih procesa izvršenih u potpunosti u virtualnom okruženju, korištenjem odgovarajuće opreme.

Iz temeljne definicije virtualne stvarnosti vidljivo je kako se uz što vjerniju imitaciju zbilje naglasak stavlja i na mogućnost interakcije u stvorenome svijetu, koja predstavlja ključni element za što vjerniju replikaciju realnog svijeta. Interakcijski uređaji, od kojih su u prošlosti bili najzastupljeniji izrazito skupi haptički uređaji, u posljednje vrijeme bivaju zamijenjeni znatno jeftinijim i praktičnijim uređajima koji omogućuju interakciju gestikulacijom putem korisnikovih ruku i prstiju. Spomenuti jeftiniji uređaji funkcioniraju na potpuno drugačiji način od haptičkih uređaja, gdje se senzori sile i vibro-taktilni aktuatori koji su montirani na korisnikove ruke kod haptičkih uređaja, u ovom slučaju u potpunosti izostavljaju, te su za interakciju, bez potrebe za dodatnom opremom, dostatne samo korisnikove ruke i prsti. Konkretno, ruke i prsti prepoznati su u virtualnom okruženju putem svojih specifičnih geometrijskih značajki koje pak prepoznaje zaseban softver. Takvi uređaji nemaju fizičku povezanost s korisnikovim rukama i prstima, već koriste načela računalnog vida, gdje će korisnikove gestikulacije biti prepoznate sve dok se njegove ruke nalaze u radnom polju kamere uređaja. Jedan od najpoznatijih opisanih uređaja je kontroler *Leap Motion*, koji će se koristiti u ovome radu. Važno je spomenuti kako nedostatak fizičke veze s korisnikovim prstima kod takvih uređaja ima i značajna ograničenja, naročito u područjima gdje je potrebno poznavanje sila koje djeluju od i na korisnika pri njegovoj interakciji s objektima u virtualnom okruženju.

[1] engl. *Virtual Reality* (VR)

Nadalje, pojam koji se neizostavno veže uz područje virtualne stvarnosti jest pojam imerzivnosti, koji podrazumijeva generiranje trodimenzionalne slike za koju se pričinja da okružuje korisnika. Ono što je bit imerzivnosti, to je naglasak na implementaciju korisnikovih osjetila u umjetno stvoreno okruženje, za što snažnije dočaravanje uključenosti korisnika u virtualno okruženje, što je općenito i temeljna svrha virtualno stvorenih okruženja.

U ovome radu bit će obrađeni različiti radni tj. proizvodni procesi za koje će se koristiti prethodno spomenuti alati i tehnike. Da bi se uopće moglo krenuti s povezivanjem problematike strojarstva (naprimjer, uspostave i vizualizacije objekata i određivanja značajki njihovoga gibanja prostorom djelovanjem ljudske ruke ili hvataljke robota) i područja virtualne stvarnosti, potrebno je krenuti s temeljnim elementima koji ta dva sustava povezuju, a to su interakcijski objekti, tj. dijelovi nekoga sklopa (proizvoda), s kojima se vrši interakcija. Dijelovi za interakciju, u području strojarstva prvotno su stvoreni računalno, u obliku CAD[2] dijela, generiranog unutar nekog od CAD alata. Virtualno okruženje također koristi dijelove, koji su međutim generirani na drugačiji način, ali načelno vrlo slično kao i dijelovi generirani u CAD-u. Ipak, dijelovi generirani u softverima za stvaranje virtualnog okruženja općenito su manje kompleksni i ne moraju zadovoljavati usko polje tolerancija i određene površinske značajke koje su vrlo bitne u slučaju kada se geometrija generira unutar CAD sustava. S obzirom na očiti problem generiranja kompleksne geometrije u softverima za stvaranje virtualnog okruženja, potreba koja se javlja jest prebacivanje generirane geometrije u CAD-u u softver za stvaranje virtualnog okruženja, kako bi se kompleksni dijelovi mogli uključiti u virtualno okruženje kao željeni interakcijski elementi sustava.

Povezivanje CAD geometrije i općenito CAD sustava sa sustavima virtualne stvarnosti područje je koje je malo ili nimalo istraženo u trenutku pisanja ovoga rada. To ne treba čuditi s obzirom na činjenicu kako su tehnike i uređaji virtualne stvarnosti za komercijalnu uporabu na tržištu dostupni relativno kratko vrijeme, što ilustrira činjenica da je kompanija Oculus svoju komercijalnu verziju zaslona montiranog na glavu, poznatijeg pod nazivom *Oculus Rift*, stavila na tržište u prvom kvartalu 2016. godine. Prije toga, sustav virtualne stvarnosti sličnih karakteristika nije bilo moguće nabaviti u komercijalnoj verziji, već su u opticaju bile različite

[2] engl. *Computer Aided Design* – konstruiranje pomoću računala

verzije uređaja namijenjene isključivo razvojnim programerima, koje su zahtijevale svojevrsni entuzijazam korisnika jer su aplikacije za korištenje na takvim sustavima bile ograničene, te je za njihovo pokretanje bilo potrebno napraviti niz rutina koje će tek kasnije biti uključene unutar instalacijskih paketa kod komercijalne verzije uređaja.

Tehnike virtualne stvarnosti mogu dovesti do poboljšanja i unapređenja postojećih procesa i usluga, naročito u okruženjima gdje je važna interakcija između čovjeka i elemenata sustava. Kako tehnike virtualne stvarnosti imaju sve veći značaj, odnosno kako je njihova primjena u različitim područjima ljudskog života sve veća, logično je zaključiti kako bi se nova načela i prednosti koje virtualno okruženje pruža mogla u bližoj budućnosti početi koristiti i u različitim granama strojarstva. Osim primjene primjerice u području montaže i kontrole kvalitete, simulacijama virtualnih postrojenja i zadacima virtualnog sklapanja, gdje se tehnike virtualne stvarnosti naširoko koriste [1], biti će zanimljivo vidjeti hoće li spomenute tehnike zahvatiti i druga temeljna područja strojarstva poput 3D modeliranja, te na koji način. Virtualna stvarnost mogla bi poništiti ograničenja koja se susreću kod klasičnog 3D modeliranja u CAD sustavima, poput limitirane vizualizacije procesa i upravljanja velikim sklopovima, te dovesti do jednostavnijeg i intuitivnijeg manipuliranja dijelovima.

Rad će biti podijeljen u dva dijela: u prvom će se dijelu razvijati aplikacija za niskobudžetnu platformu VR, Google Cardboard, te će se krenuti s malim brojem pojedinačnih dijelova, modeliranih u nekom od CAD softverskih paketa. Drugi dio rada uključivat će više pojedinačnih dijelova i sklopove, te će se koristiti oprema naprednijih specifikacija koju će činiti zaslon montiran na glavu – Oculus Rift i senzor pokreta – kontroler Leap Motion. Virtualno okruženje za obje platforme biti će stvoreno unutar softvera Unity koji ima podršku stvaranja aplikacija za velik broj različitih platformi. Platforma Google Cardboard, cjenovno je vrlo pristupačna što sa sobom nosi i određena ograničenja u odnosu na naprednije sustave VR. Najveći nedostatak je prikaz lošije kvalitete jer se za pokretanje sadržaja koristi mobilni telefon, za razliku od Oculus Rifta koji koristi resurse računala. Nadalje, mogućnost interakcije smanjena je zbog ograničenog broja ulaznih uređaja te u konačnici, ali i ne manje važno, uređaj je konstrukcijski loše izrađen.

Problem koji zahvaća obje platforme je nemogućnost izravnog povezivanja CAD-a i okruženja VR bez softvera – posrednika, koji služi za pretvorbu i prilagođavanje CAD formata i uključivanje CAD geometrije u virtualno okruženje. Za konverziju geometrije iz CAD sustava u geometriju odgovarajućeg formata za softver virtualne stvarnosti koristit će se Autodesk 3ds Max. Stoga je **prvi cilj ovog rada povezati generiranu CAD geometriju i sustave virtualne stvarnosti, kao osnova za drugi, a to je istraživanje mogućnosti primjene virtualne stvarnosti u projektiranju radnih odnosno proizvodnih procesa, gdje će interakcijski elementi virtualnog okruženja biti upravo CAD dijelovi**. [2]

2. SUSTAVI VIRTUALNE STVARNOSTI I NJIHOVA PERSPEKTIVA

Iako do razvoja prvih uređaja virtualne stvarnosti (VR) dolazi pojavom *Sensoram*e 50-ih godina prošlog stoljeća [3], čiji je osnivač patentirao i prvi zaslon montiran na glavu[3], razvoj tako naprednih uređaja za to vrijeme nije omogućio adekvatnu i široku primjenu, prvenstveno zbog skromnih mogućnosti tadašnjih računala ili opreme pomoću koje se prikazivao virtualan svijet. S razvojem tehnologije to se mijenja, te je u današnje vrijeme tehnologija VR široko dostupna i primjenjiva širokoj populaciji zbog iskoraka u računalnoj industriji i komercijalizaciji popratnih sustava VR. Cijene suvremenih uređaja VR pale su na vrlo pristupačne iznose, pa se sustavi poput Google Cardboarda i sličnih modifikacija kvalitetnije izvedbe mogu nabaviti po cijeni već od par desetaka USD. Za razvoj virtualnog okruženja na platformi Google Cardboard potrebno je napraviti aplikaciju za mobilni telefon, zbog toga što navedena platforma koristiti pametni telefon kako bi se projicirao željeni sadržaj. Napredniji, a samim time i skuplji uređaji virtualne stvarnosti, poput Oculus Rifta, ne projiciraju sadržaj putem mobilnog telefona, već se izravno spajaju na stolno ili prijenosno računalo preko priključka HDMI[4] i/ili USB. Na najnovijim verzijama uređaja VR, priključak HDMI nerijetko biva zamijenjen priključkom *DisplayPort*[5].

2.1. IMERZIVNOST

Uz pojam virtualne stvarnosti, nezaobilazan termin koji se susreće jest imerzivnost. Pojam imerzivnosti podrazumijeva generiranje trodimenzionalne slike virtualne stvarnosti za koju se pričinja da okružuje korisnika, gdje je slika generirana putem računalnog zaslona ili sustava. Imerzivnost se također može definirati kao percepcija fizičke prisutnosti u

[3] engl. HMD – *head mounted display*

[4] *High-Definition Multimedia Interface* [48] – veza za istovremeni prijenos slike i zvuka u nekomprimiranom obliku; digitalna zamjena za starije analogne sustave poput SCART/VGA/S-Video priključka.

[5] DisplayPort je sučelje koje se primarno koristi za prijenos video signala, iako se također može koristiti za prijenos audio, USB i ostalih vrsta podataka. Karakterizira ga povratna kompatibilnost s priključcima VGA, DVI i HDMI pomoću bilo aktivnih bilo pasivnih adaptera. [50]

nefizičkom svijetu. Dojam percepcije stvoren je putem sustava virtualne stvarnosti koji stimulira čovjekova osjetila preko vizualnih, zvučnih ili drugih podražaja koji služe za stvaranje okruženja. Da bi okruženje što više podsjećalo na realan svijet, potrebno je stimulirati što više osjetila i to na što realniji način. Za stimulaciju vida koriste se uređaji za projekciju sadržaja, kao što su 3D zasloni, hologrami i zasloni montirani na glavu, dok se kod stimulacije sluha koriste 3D zvučni efekti, imerzivan zvuk i tehnike zvučnog okruženja. Stimulacija dodira vrši se putem haptičkih uređaja koji služe kako bi čim realnije osjet opipa uključili u virtualno okruženje. Posebni sustavi čak pružaju stimulaciju njuha, dok sustavi koji stimuliraju jedini preostali osjet, a to je osjet okusa, u ovome trenutku nisu uključeni u postojeće sustave virtualne stvarnosti.

Stimulacija njuha potječe još od začetaka virtualne stvarnosti i kinematografa Mortona Heiliga, koji je svoju *Sensoramu* opremio ventilatorima i sustavima za generiranje mirisa. Za interakciju u virtualnom, imerzivnom okruženju, najčešće se koriste tehnike poput prepoznavanja kretnji i prepoznavanja govora dok je u budućnosti svakako cilj vršiti interakciju putem naprednih metoda komunikacije između čovjeka i stroja, od kojih je svakako jedna od zanimljivijih tematika interakcija putem prepoznavanja misli čovjeka.

U imerzivnom okruženju vrlo je važan stupanj osjećaja simuliranog prostora, tzv. *sense of immersion.* Što je veći stupanj osjećaja, imerzivno okruženje manje odstupa od stvarnoga. S obzirom na razine, virtualna stvarnost se sve do nedavno dijelila na neobuhvatnu, poluobuhvatnu i sveobuhvatnu[6] virtualnu stvarnost, međutim, zbog prodaje komercijalnih verzija sustava VR situacija se mijenja te se razina poluobuhvatne virtualne stvarnosti praktički izgubila. Nova podjela virtualne stvarnosti mogla bi se izvršiti samo u dvije kategorije: neobuhvatnu i obuhvatnu. Tablica 1. predočuje razlike između tih dviju razina.

[6] engl. *non-immersive*, *semi-immersive*, *fully immersive*

Tablica 1. Razine virtualne stvarnosti [4]

Razine VS	**Neobuhvatna VS**	**Obuhvatna VS**
Ulazni uređaji	Miš, tipkovnica, *joystick*	Rukavice *data glove*, uređaji za praćenje i prepoznavanje pokreta, zvučne naredbe
Izlazni uređaji	Standardni visokorezolucijski monitori, više monitora, projektori	Zaslon montiran na glavu
Rezolucija	Visoka	Srednja
Stupanj uključenosti čovjeka u simuliranom prostoru	Nema - nizak	Visok
Stupanj interakcije	Nizak	Visok
Cijena	Niska	Niska-srednja

Svakodnevni rast i razvoj u računalnoj industriji a naročito lansiranje na tržište komercijalnih sustava virtualne stvarnosti, značajno su promijenili područje poimanja obuhvatne virtualne stvarnosti. Rezolucija kod takvih uređaja i dalje je manja od primjerice standardnih monitora, ali se više ne može definirati kao niska što je bio slučaj do prije par godina, jer je ta razlika znatno smanjena. Isto tako, bitna razlika najviše je vidljiva u segmentu cijene, gdje se sustavi obuhvatne virtualne stvarnosti više ne svrstavaju u rang cjenovno vrlo skupih proizvoda, već bi se mogli svrstati u kategoriju između jeftinih i srednje skupih uređaja, prvenstveno zbog mogućnosti prikaza navedenog sadržaja na mobilnim platformama, koje za prikaz koriste uređaje poput Google Cardboarda ili slične modificirane verzije kvalitetnije izvedbe, a čija cijena je niska. Za projiciranje sadržaja na takvoj platformi znatno više sredstava potrebno je izdvojiti za odgovarajući mobilni uređaj koji će pripadajući sadržaj reproducirati u zadovoljavajućoj kvaliteti. Postoji i druga krajnost koju čine uređaji miješane stvarnosti poput Microsoft HoloLensa, a koji cjenovno znatno odstupa od bilo kojeg sustava virtualne

stvarnosti namijenjenog bilo za mobilnu ili računalnu platformu. Cijena uređaja Microsoft HoloLens iznosi 3000 USD za verziju namijenjenu za razvojne programere, dok je za potrošačku verziju potrebno izdvojiti čak 5000 USD. Međutim, zbog svoje specifičnosti prikaza, navedeni uređaj ne spada u kategoriju virtualne stvarnosti kao što se često pogrešno navodi, već se radi o uređaju koji spada u kategoriju miješane stvarnosti, koja se po mnogočemu razlikuje od virtualne stvarnosti, što će biti opisano na kraju ovog poglavlja.

Nadalje, što se tiče kategorije ulaznih uređaja, osim *data glove* haptičkih rukavica koje su vrlo skupe a koje omogućuju osjet opipa u virtualnom okruženju, sve češće se koriste drugačije vrste ulaznih uređaja za interakciju s dijelovima, koje su višestruko jeftinije od haptičkih rukavica a dovoljno vjerno predočuju interakciju unutar virtualnog okruženja, zbog čega njihova raširena primjena ne treba izazvati čuđenje. Osim kontrolera Leap Motion čija se primjena u većini slučajeva očituje u sustavima VR, znatno veće tržište pokriva komercijalna verzija uređaja Oculus Rift koja sa sobom sadrži Oculus Touch, sustav za integraciju dodira u okruženje virtualne stvarnosti (Slika 1.). Oculus Touch sastoji se od dva joystick uređaja, po jedan za svaku ruku, koji sadrže gumbe i dva okidača – jedan za hvatanje te drugi za ispaljivanje/pucanje. Sustav je u potpunosti praćen u 3D prostoru preko Oculus Constellationa te posjeduje sustav za detekciju gestikulacije prstiju u trenutku držanja kontrolera. Međutim, potreba za korištenjem fizičkih komponenti kod sustava Oculus Touch očituje se kao veliki nedostatak u odnosu na kontroler Leap Motion koji ne zahtijeva nikakvu interakciju s fizičkim komponentama za rad.

Slika 1. Oculus Touch [5]

Imerzivno okruženje koje će biti stvoreno u sklopu ovog rada imat će naglasak na interakciji između dijelova unutar proizvodnog procesa uz prikaz trenutnih mogućnosti u području virtualnog sklapanja i virtualne simulacije postrojenja. Cilj je što vjernije prikazati odabrani proizvodni proces, uz što manje odstupanje od istog proizvodnog procesa koji se vrši u zbilji. Svrha takvog pristupa jest izvući sve prednosti koje virtualno okruženje pruža u odnosu na klasični pristup projektiranja proizvodnih procesa. [4 do 8]

2.2. SUSTAVI VIRTUALNE STVARNOSTI

U nastavku slijedi prikaz sustava virtualne stvarnosti koji će se koristiti u sklopu ovog rada. U prvom dijelu to će biti niskobudžetni Google Cardboard, dok će u drugom dijelu rada biti korišten Oculus Rift.

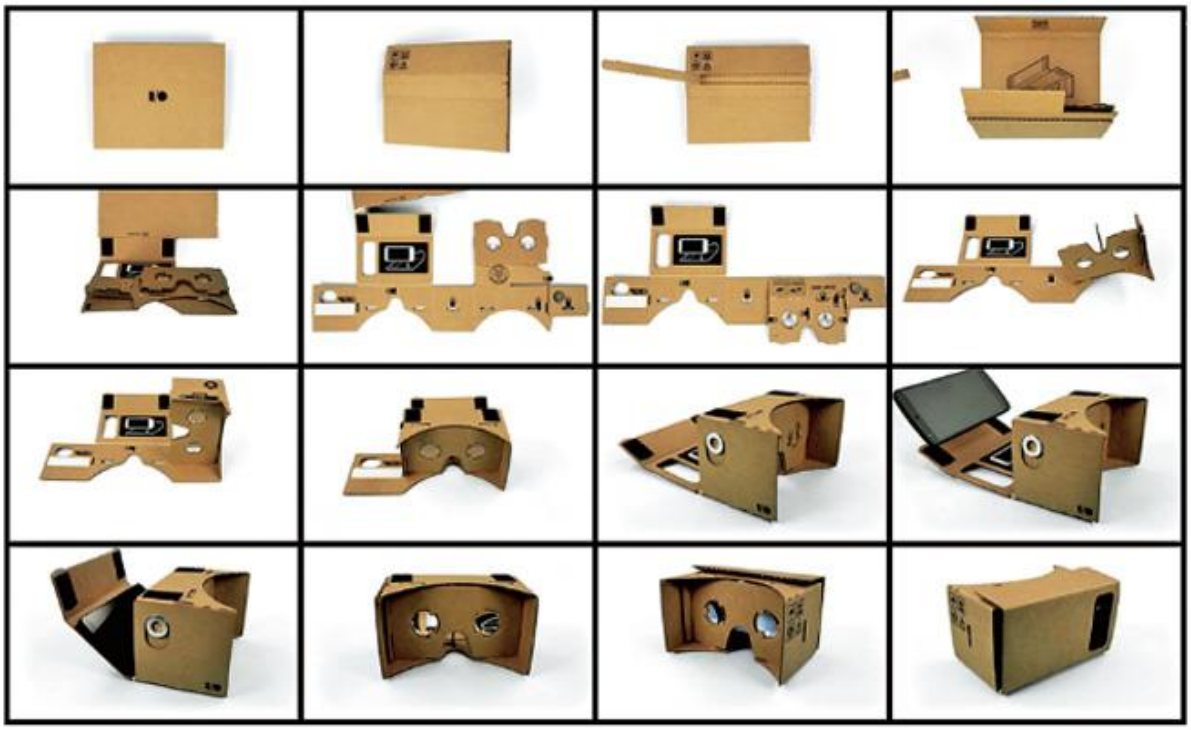
2.2.1. Google Cardboard

Google Cardboard je platforma VR razvijena od strane kompanije Google, koja koristi kartonski zaslon montiran na glavu (prema kojemu je dobila ime) i pametni telefon za prikaz VR sadržaja. Sustav karakterizira niska cijena (otprilike 15 USD) i široka dostupnost, gdje sam korisnik može izraditi svoj *Cardboard* iz uputa objavljenih na stranici proizvođača. Mobilni telefon služi za projiciranje sadržaja te se umeće u poseban pretinac na prednjoj strani *Cardboarda*. Važno je napomenuti kako se kod spominjanja platforme Google Cardboard nikako ne misli samo na kartonski zaslon montiran na glavu, već na cjelinu između zaslona, pametnog telefona, ali i popratne aplikacije Google VR SDK[7] koja omogućuje projiciranje sadržaja na poseban, imerzivan način. [9]

Upute za sklapanje Google Cardboarda poprilično su jednostavne (Slika 2.), a sastoje se od umetanja više pojedinačnih kartonskih dijelova u numerirane prostore u obliku provrta, gdje

[7] *Software Development Kit*

se izvršavajući upute i slijedeći korake dane nizom brojeva od 1 do 9, postepeno dolazi do sklopa u cjelini.



Slika 2. Detaljni prikaz postupka sastavljanja *Google Cardboarda* [10]

Sklopljeni sustav Google Cardboard poprima oblik kao na slici 3.

Slika 3. Sklopljeni sustav Google Cardboard [11]

2.2.2. Oculus VR

Oculus je američka tehnološka kompanija osnovana od strane Palmer Luckyja 2012. godine u Kaliforniji, specijalizirana za razvoj hardvera i softvera virtualne stvarnosti. Najpoznatiji proizvod kompanije je Oculus Rift headset VR (Slika 4.), koji je kroz prošlost doživio različite verzije. Počevši kao projekt na Kickstarteru, Oculus Rift je doživio pet različitih vrsta prototipova, od kojeg su dvije verzije bile moguće za kupnju kao tzv. razvojni set (*development-kit*), Oculus DK1 krajem 2012. i Oculus DK2 sredinom 2014. godine. Iako su razvojni setovi namijenjeni za programere, velik broj entuzijasta se svejedno odlučio na kupnju takvog seta. U početku, kompanija je predviđala dobit od 250 000 USD, što je višestruko premašeno jer je dobit u kratkom vremenu iznosila više od 2,4 milijuna USD. U ožujku 2014. Facebook je kupio Oculus VR za dio udjela u dionicama Facebooka i dvije milijarde američkih dolara u gotovini. Komercijalna verzija Oculus Rifta (poznatija kao CV1) na tržište je puštena 28. ožujka 2016. godine te se trenutno može nabaviti po cijeni od 598 USD.

Slika 4. Oculus Rift CV1 [5]

CV1 posjeduje značajne prednosti u odnosu na sve prijašnje verzije *Rifta*, pa tako i na verziju DK2 koja će se koristiti u ovom radu. O usporedbi, prednostima i nedostacima CV1 verzije i verzije DK2 biti će govora u nastavku. Osim primjene u području računalnih igara, *Rift* nalazi primjenu u područjima poput društvenih mreža, virtualnog kina, sporta, TV industrije ali i u području industrije, gdje je *Rift* izazvao interes u području poboljšanja produktivnosti,

vizualizacije procesa i prodaje. U području arhitekture dolazi do eksperimentiranja u vizualizaciji dizajna dok se u području autoindustrije primjena očituje u raznim simulatorima ili dizajniranju automobila prema kupčevim željama. Koncem 2015. godine kompanija širi svoje vidike, tu u suradnji sa Samsungom razvija Samsung Gear VR, *headset* virtualne stvarnosti podržan za Samsung mobilne telefone, koji je prodan u više od 5 milijuna primjeraka. Samsung Gear VR se funkcionalno ne razlikuje od Google Cardboarda, ali je konstrukcijski bolje razrađen i ima podršku samo za određene vrste mobilnih telefona čije komponente podržavaju prikaz VR sadržaja u visokoj kvaliteti. [5, 12 i 13]

2.3. PERSPEKTIVA SUSTAVA VIRTUALNE STVARNOSTI

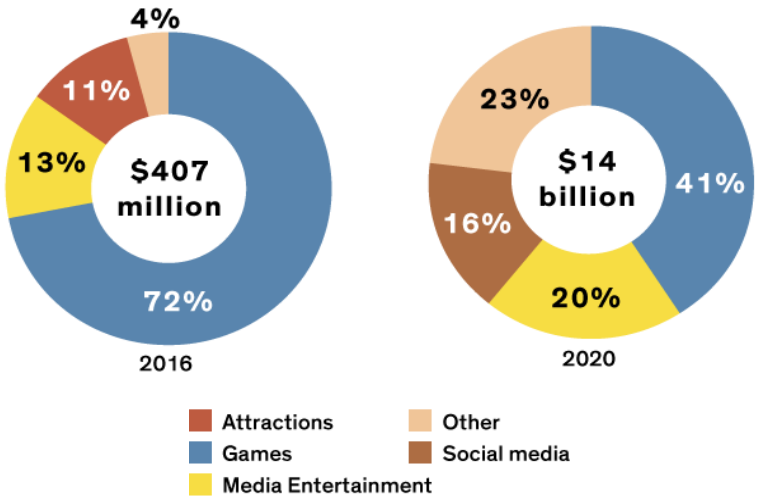
U prošlosti je naglasak tehnologije VR bio prvenstveno fokusiran na primjenu u zabavnoj industriji, dok se u proteklih par godina situacija mijenja, te novi uređaji poput Microsoft HoloLensa i Oculus Rifta nalaze primjenu i u područjima koja dotiču širi spektar ljudskih djelatnosti, poput medicine, stomatologije te raznih inženjerskih grana.

Dobru usporedbu sadašnjeg stanja i eventualne skore budućnosti pruža slika 5. Iz slike je vidljivo kako je prošle, 2016. godine, prihod industrije VR od strane potrošača iznosio 407 milijuna USD. Do 2020. godine, SuperData[8] predviđa kako će prihodi narasti na čak 14 milijardi USD. Ono što je bitnije, jest činjenica kako trenutno praktički ¾ prihoda u industriji virtualne stvarnosti dolazi od strane računalnih igara, dok samo 4 % prihoda dolazi iz kategorije ostalo, u koju primjerice spada i strojarstvo. Do 2020. godine primjena virtualne stvarnosti u različitim granama poput strojarstva, medicine i stomatologije bi se trebala znatno povećati, sudeći po povećanju potrošačkog prihoda u kategoriji ostalo, na 23 %. Jasno je vidljivo kako bi se trenutno stanje u kojem se virtualna stvarnost većinom oslanja i nalazi primjenu u području zabave i računalnih igara trebalo promijeniti, u prilog čemu ide i procjena Goldman Sachsa[9] prema kojoj se navodi kako je virtualna stvarnost vjerojatno sljedeća napredna platforma koja će izazvati revoluciju. Prema toj procjeni, do 2025. godine

[8] tvrtka za istraživanje tehnološkog tržišta

[9] američka multinacionalna financijska kompanija [49]

tržište virtualne stvarnosti pokrivalo bi vrijednost od 80 milijardi USD, što je otprilike iznos vrijednosti tržišta koje u današnje vrijeme drže stolna računala. [14 i 15]

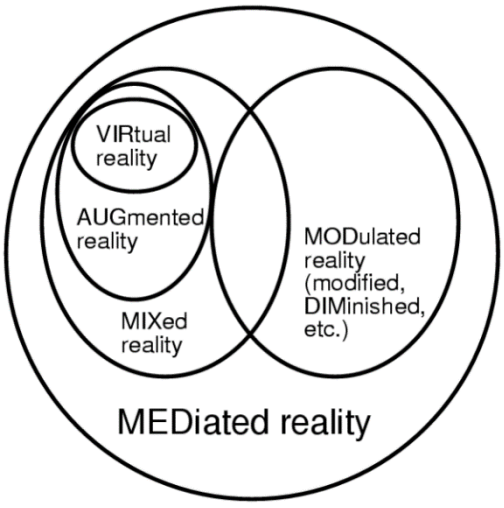


Slika 5. Prihodi potrošačkih sustava VR u 2016. godini i projekcija za 2020. godinu [14]

2.4. RAČUNALNO POSREDOVANA STVARNOST

Zbog činjenice kako sustavi virtualne stvarnosti nisu uvelike raširenи te trenutačno ne nalaze primjenu kod široke populacije, kao što je to slučaj primjerice s tržištem pametnih telefona, vrlo često se može naići na pogrešnu interpretaciju i preklapanje termina poput virtualne/proširene/miješane stvarnosti za opisivanje određenih uređaja. Iako se površnom poznavatelju navedeni termini mogu činiti vrlo slični zbog nedovoljne informiranosti o navedenim sustavima, ipak postoji značajna razlika između svakog od tih sustava. Virtualna stvarnost, kako je opisana u ovom radu, uvelike se razlikuje od proširene stvarnosti jer ona realan svijet u potpunosti zamjenjuje sa simuliranim svijetom. Za razliku od virtualne, kod proširene stvarnosti korisnik nije dio u potpunosti stvorenog novog, umjetnog okruženja, već on predstavlja dio realnog svijeta u kojem je prikaz onoga što on uistinu vidi poboljšan elementima koji su realizirani putem sustava proširene stvarnosti. Za realizaciju tih elemenata najčešće se koriste kamere koje uz pomoć strojnog vida ili prepoznavanja objekata poboljšavaju korisnikov prikaz koji on vidi u stvarnosti, te omogućuju interakciju i digitalnu manipulaciju. Zbog potrebe korištenja kamere, a vrlo često i GPS sustava, kompasa i uređaja poput akcelerometra i/ili žiroskopa, za sustave proširene stvarnosti vrlo su pogodni mobilni uređaji poput pametnih telefona ili tableta. Nadalje, potpuno neovisni zasloni za

sustave proširene stvarnosti, temeljeni su na tzv. HUD[10] uređaju. Uređaj HUD predstavlja proziran zaslon koji projicira podatke bez da zahtijeva korisnikovo usmjeravanje pogleda izvan uobičajenih točaka gledišta. U današnje vrijeme, najpoznatiji takav zaslon je Microsoft Hololens, za koji se naročito često upotrebljava pogrešna terminologija kako je to sustav virtualne stvarnosti. U svojoj osnovi Microsoft Hololens je sustav proširene stvarnosti, jer upravo počiva na karakteristikama koje krase uređaj HUD, dok korisnik u svakom trenutku percipira stvari iz onoga što doista trenutno vidi u stvarnome svijetu. Razlog pogrešnoj interpretaciji djelomično je nedostupnost uređaja za krajnje korisnike, jer još uvijek na tržištu ne postoji komercijalna verzija uređaja. Također, u posljednje vrijeme vidljiv je i svojevrsni odmak Microsofta od definiranja *HoloLensa* kao uređaja proširene stvarnosti, tako da je u ovom trenu na službenim stranicama proizvođača *HoloLens* preciznije definiran kao uređaj miješane stvarnosti. Miješana stvarnost, za koju se ponekad koristi i termin hibridna stvarnost, predstavlja spajanje stvarnog i virtualnog svijeta kako bi se stvorila nova okruženja gdje fizički i digitalni objekti koegzistiraju i vrše interakciju u realnome vremenu. Odvijanje radnje miješane stvarnosti ne odvija se isključivo u fizičkom ili virtualnom svijetu, već se radi o mješavini fizičkog i virtualnog svijeta, istodobno obuhvaćajući i segmente proširene stvarnosti. S obzirom na nedostatak adekvatne riječi koja bi obuhvatila sve navedene vrste umjetno stvorenih stvarnosti, predlaže se termin računalno posredovane stvarnosti koji uključuje vrste poput virtualne stvarnosti, proširene stvarnosti i miješane stvarnosti u svoj podskup, kao što je vidljivo na slici 6. [16 i 17]



Slika 6. Računalno posredovana stvarnost [18]

[10] engl. *head-up display*

2.5. ZDRAVSTVENE POTEŠKOĆE – BOLEST VIRTUALNE STVARNOSTI

Korištenje bilo koje stvari, proizvoda i alata, pa tako i uređaja za virtualnu stvarnost, podrazumijeva prethodno razmatranje sigurnosnih uključivo i zdravstvenih aspekata.

Korištenje nekog od uređaja virtualne stvarnosti u vrlo kratkom vremenu može izazvati osjećaje poput neugodnosti, mučnine, glavobolje, vrtoglavice, znojenja, pospanosti, dezorijentacije, bljedila, boli i peckanja u očima što su simptomi bolesti virtualne stvarnosti. Bolest virtualne stvarnosti javlja se u trenutku kada je korisnik izložen virtualnom okruženju, te posjeduje neke od prije navedenih simptoma, a koji su vrlo slični simptomima bolesti kretanja[11] koja je sindrom neslaganja između vizualne percepcije pokreta i centra za ravnotežu. Međutim, bolest virtualne stvarnosti razlikuje se od bolesti kretanja po tome što može biti uzrokovana vizualno induciranom percepcijom vlastitog kretanja gdje nije nužno potrebno da se korisnik kreće, iako je njegovo kretanje moguće unutar okruženja stvorenog tehnikama virtualne stvarnosti.

Postoji više teorija zašto dolazi do bolesti virtualne stvarnosti. Teorija senzorskog konflikta smatra kako se bolest javlja kada je korisnikova percepcija vlastite kretnje bazirana na nepodudaranju senzorskih podražaja vizualnog sustava i sustava ravnoteže, pogotovo kada se senzorski podražaji ne podudaraju s korisnikovim očekivanjima utemeljenim na prethodnom iskustvu. Nadalje, postoje očiti indikatori koji utječu na bolest virtualne stvarnosti. Jedan od glavnih problema je stopa osvježavanja[12], koja često nije dovoljno visoke frekvencije, što može pogodovati nekim od prije navedenih simptoma. Stopa osvježavanja je manja tj. sporija od brzine kojom mozak obrađuje podatke što rezultira problemima prikaza.

Kod praćenja pokreta glave korisnici mogu doživjeti sličan fenomen. Naime, ako korisnik slobodno okreće glavu a prikaz na njegovu zaslonu kasni, to može rezultirati nekom vrstom bolesti kretanja. Primjerice, u virtualnom okruženju korisnikove oči šalju mozgu informaciju

[11] engl. *motion sickness*. Drugi nazivi koji se koriste u hrvatskom jeziku su kinetoza, morska bolest ili bolest putovanja.

[12] engl. *refresh rate*

da on trči kroz prostor, međutim centar za ravnotežu odašilje signale da se osoba ne kreće, te da miruje, što također može utjecati na bolest virtualne stvarnosti. Iako, važno je napomenuti kako se svi ne slažu s ovom teorijom. Druga teorija, tzv. teorija nestabilnosti, drži kako je razlog bolesti loša prilagodba tijela na neuobičajenu vizualnu stimulaciju. Ova teorija predviđa kasniji razvoj simptoma bolesti virtualne stvarnosti. Specijalisti u igrama virtualne stvarnosti navode kako su simptomi tek manji problem koji se može prebroditi s više vremena provedenog u korištenju zaslona montiranog na glavu. Zbog iznesenih problema, većina sustava virtualne stvarnosti posjeduje upozorenje za zdravlje i sigurnost osobe koja rukuje sustavom VR. Upozorenje se najčešće prikazuje već kod samog spajanja zaslona montiranog na glavu na računalo, ili netom prije pokretanja igre ili aplikacije. [19 i 20]

3. POVEZIVANJE VIRTUALNE STVARNOSTI I CAD-a

Realizacija povezivanja CAD-a i VR sustava je relativno nova jer su komercijalni sustavi VR došli na tržište prije svega par mjeseci. Doduše, Oculus VR je već prije par godina omogućio nabavu Oculus Rifta u vidu razvojnog kita za programere (*development kit*), što nije rezultiralo pretjeranom primjenom u inženjerskom području. Spomenuta tehnologija koristila se prvenstveno za povezivanje računalnih igara i simulacija te njihov realističniji prikaz na zaslonu montiranom na glavu. Zbog svega navedenog, ukoliko se razmatra šira primjena navedenih sustava, primjerice povezivanje klasičnih CAD softvera i sustava VR, odmah se može očekivati kako će navedena interakcija rezultirati brojnim problemima poput nekompatibilnosti i nedostatka međusobne programske podrške.

Primjerice, CAD sustav poput CATIA-je razvijen je još 80-ih godina prošlog stoljeća. Ostali CAD sustavi nešto su novijeg datuma, ali su svejedno relativno stari u odnosu na novu tehnologiju VR. Iako CAD softvere generalno odlikuje mogućnost spremanja modelirane geometrije u mnoštvo različitih izlaznih formata, od kojih su neki standardni poput STEP-a i IGES-a, svejedno dolazi do nemogućnosti prikaza takvih formata na platformi VR, zbog njezinog nedostatka podrške za spomenute formate. Da bi se CAD geometrija prikazala na platformi VR (u ovom slučaju na Google Cardboardu) potrebno je izlaznu CAD geometriju prilagoditi za određenu vrstu softvera koji omogućuje stvaranje virtualnog okruženja za neku od platformi VR. Razvojni softveri za aplikacije VR također su novijeg datuma, pa se i u tom aspektu javljaju određena ograničenja. Zbog dostupnosti besplatne studentske licence, te intuitivnog korisničkog sučelja, razvojni softver koji će se koristiti u ovome radu je Unity, preko kojeg je moguće stvoriti aplikacije za različite platforme VR, pa tako i za Google Cardboard i Oculus Rift.

Unity je softver namijenjen za izgradnju video igara na više različitih platformi. Razvijen je od strane Unity Technologies, 2005. godine, te je do današnjeg dana na njemu moguće razviti aplikacije za više od 20 različitih platformi. Unity ima direktnu podršku za prepoznavanje

formata .FBX, .dae, .3DS, .dxf i .obj. Budući da konvencionalni CAD softveri ne pružaju mogućnost spremanja izlazne geometrije u nekom od navedenih formata, morat će se koristiti pretvarač tj. konverter formata kako bi se postigla željena kompatibilnost. Važno je spomenuti da Unity pruža i indirektnu podršku za neke od formate poput .MAX, .MA, .MB nastale u drugim softverima kao što su Max, Maya, Blender, Cinema4D, Modo, Lightwave i Cheetah 3D. Podrška je indirektna, jer je uvoz podataka moguć jedino pomoću konvertera. Za uvoz podataka iz CAD softvera koristit će se dodatni program koji omogućuje konverziju CAD datoteke u format podržan u Unityju. Najpogodniji program za takvu konverziju je Autodesk 3ds Max (u prošlosti 3D Studio), profesionalni softver koji služi za izradu 3D računalne grafike koja uključuje animacije, filmove, igre i slike. Glavna prednost spomenutog softvera je mogućnost uvoza datoteka različitih formata, od kojih će za ovaj rad biti najznačajnija podrška za formate nastale u CAD softveru poput CATIA-je (podrška za inačicu V4 i V5), SolidWorksa (SLDPRT, SLDASM), Pro/ENGINEER (PRT, ASM) ali i za univerzalne formate nastale u drugim verzijama komercijalnih CAD softvera poput STEP-a ili IGES-a. Mogućnost uvoza praktički svih vrsta formata iz najzastupljenijih CAD softvera u Autodesk 3ds Max omogućuje spremanje i izvoz željene geometrije u formatu podržanom od strane Unityja. Zbog manjeg opterećenja računalnih resursa izvoz geometrije je optimalno vršiti u .FBX formatu za razliku od unaprijed predloženog .3DS formata.

Nakon što se uspješno provede konverzija podataka između CAD softvera i Autodesk 3ds Maxa, željena CAD geometrija sprema se u format pogodan za otvaranje u softveru Unity. Promjenom okoline, definiranjem okruženja i instaliranjem odgovarajućih dodataka potrebnih za stvaranje aplikacije za platformu Google Cardboard, u konačnici će biti moguće prikazati geometriju modeliranu u CAD sustavu na niskobudžetnom sustavu Google Cardboard VR. [21 do 23]

3.1. STVARANJE GEOMETRIJE I PRETVORBA FORMATA

Za generiranje geometrije u CAD-u, tj. modeliranje, koristit će se CATIA[13] V5. CATIA je višeplatformsko softversko sučelje za CAD (konstruiranje pomoću računala), CAM (proizvodnju pomoću računala), PLM (životni ciklus proizvoda), razvijeno od strane francuske kompanije Dassault Systemes. Prvo izdanje CATIA-e došlo je na tržište 1977. godine, što ju čini jednom od najstarijih CAD softvera čija se aktivna primjena u raznim sferama inženjerskih zanimanja održala sve do današnjice.

Za modeliranje je moguće koristiti bilo koju drugu vrstu 3D CAD softvera, gdje je u ovom slučaju važno da odabrani softver ima pogodne izlazne formate koji se mogu učitati u okruženje Autodesk 3ds Max, kako bi se uspješno obavila pretvorba formata i na taj način kasnije omogućilo uključivanje modelirane geometrije u Unity. Kao što je već spomenuto u ovom poglavlju, 3ds Max ima podršku za učitavanje datoteka iz mnoštvo različitih CAD softvera, poput CATIA-je, SolidWorks-a te Pro/ENGINEER-a. U ovome primjeru će se koristiti softver CATIA prvenstveno zbog svoje dostupnosti.

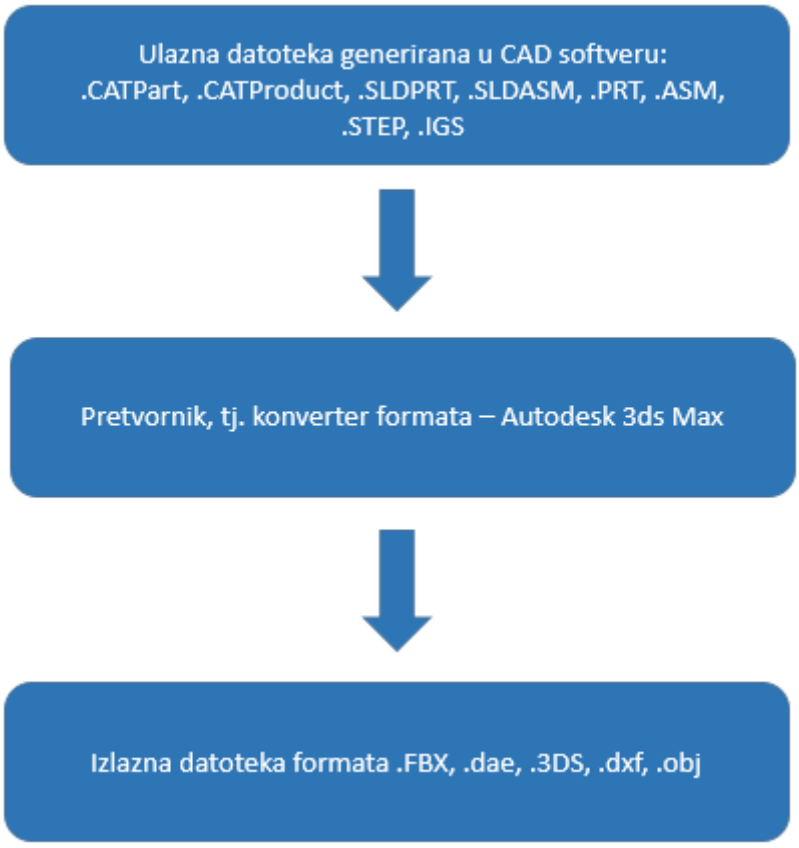
Kako samo modeliranje geometrije ne predstavlja temeljnu svrhu ovog rada, kao gotova geometrija mogu se koristiti već modelirani dijelovi dostupni na raznim internetskim sjedištima[14]. U ovome konkretnom slučaju, kao pokazni primjer generirane geometrije koristit će se model koljenastog vratila dostupan na GrabCAD-u. GrabCAD je internetsko središte koje objedinjuje razne entuzijaste i inženjere različitih struka koji modele svojih radova pohranjuju u online bazu podataka, koja je besplatno dostupna svim korisnicima interneta, uz prethodnu registraciju.

U nastavku će biti prikazana shema pretvorbe formata: konkretno, nakon što je dio tj. *part* oblikovan u CATIA-ji, klikom na ikonicu *spremi* softver automatski sprema *part* u formatu CATPart. Navedeni format neće se moći direktno uvesti u softver za virtualno okruženje, kao

[13] *Computer Aided Three-dimensional Interactive Application*

[14] Naprimjer: https://grabcad.com/

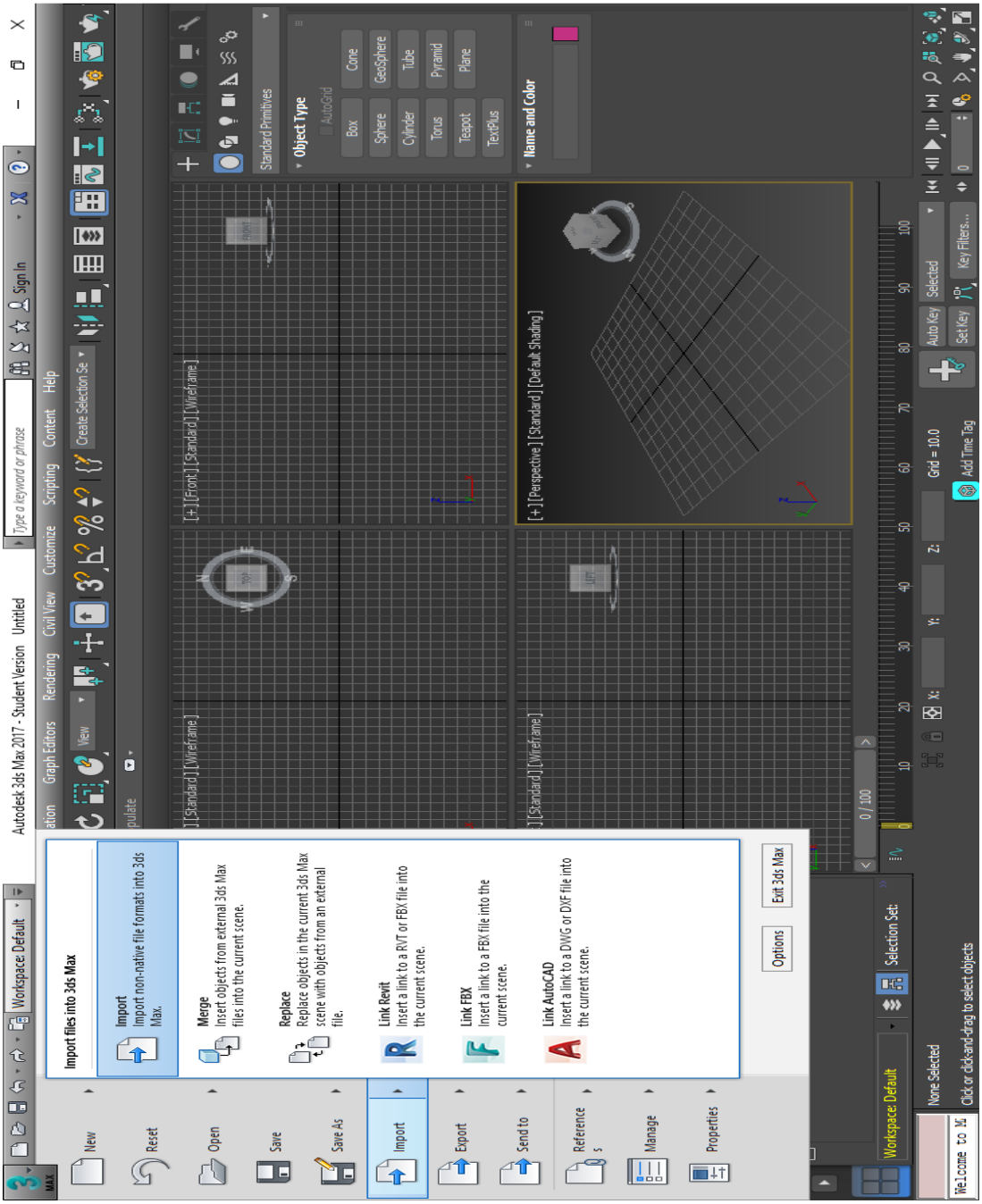
što je već i prije napisano, već će biti potrebna dodatna konverzija putem pretvornika tj. konvertera formata u format pogodan za čitanje u Unityju. Pretvorba će se napraviti putem programa Autodesk 3ds Max, koji omogućuje spremanje izlaznih datoteka u raznim formatima, od čega će .FBX i .3DS formati biti pogodni za uvoz u Unity. [24 i 25]



Slika 7. Shema konverzije podataka od CAD-a do VR-a

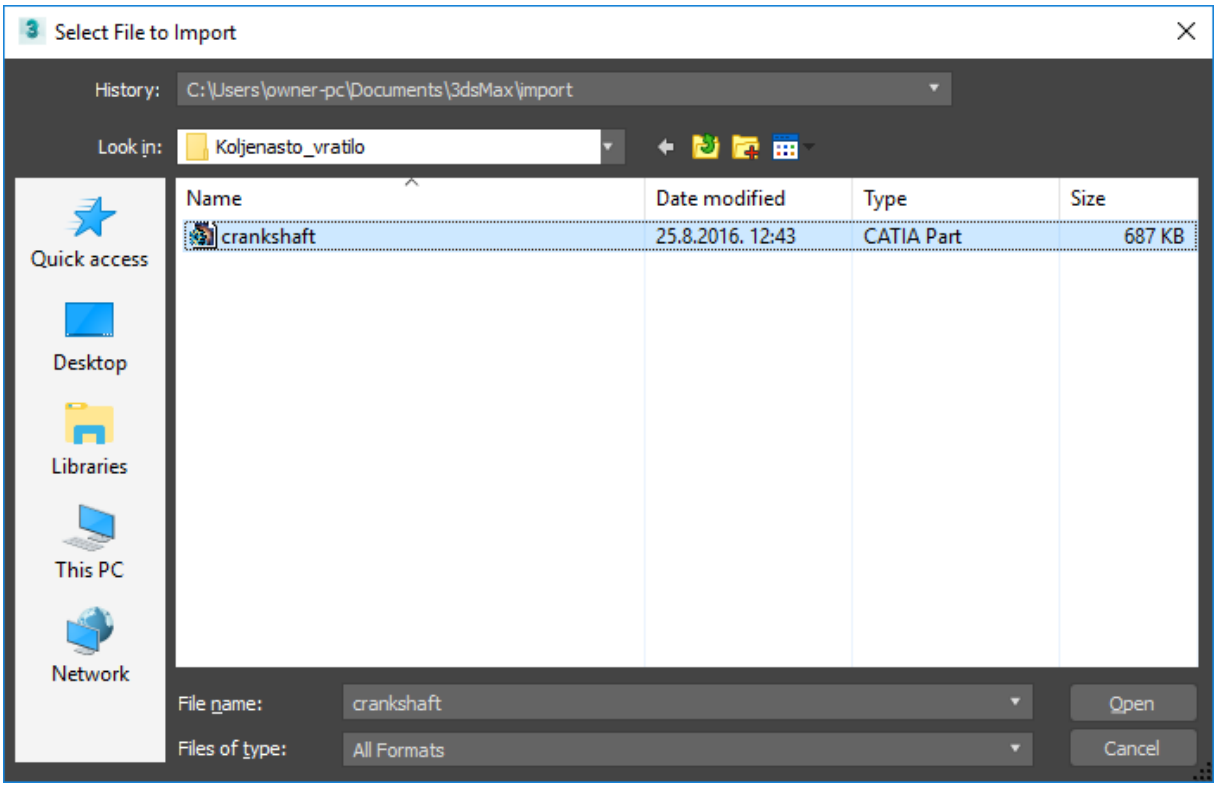
3.2. Autodesk 3ds Max

Part koljenasto vratilo koje je spremljeno u CATIA-ji otvara se u Autodesk 3ds Max softveru (Slika 8.). To se čini na način da se u padajućem 3MAX izborniku odabere opcija *import* te klikne na *Import – Import non native file formats into 3ds Max.* Odabrani izbornik omogućuje učitavanje datoteka koje nisu izvorno nastale u Autodesk softveru, te će se preko njega uvesti *part* koljenasto vratilo u programsko okruženje.



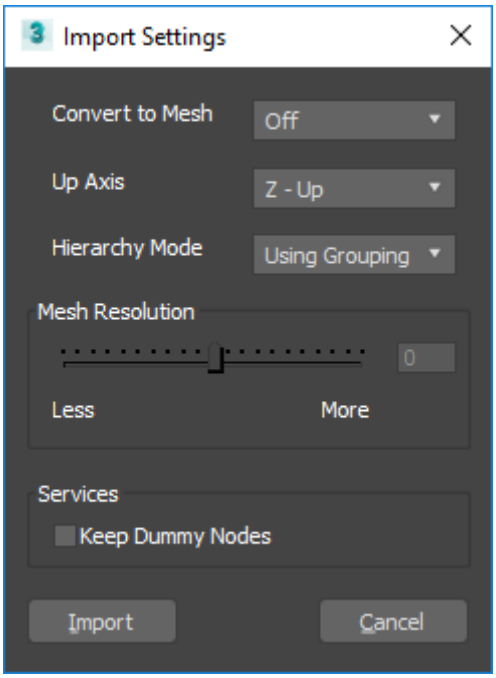
Slika 8. Autodesk 3ds Max

Nakon nekog vremena otvorit će se skočni prozor gdje je potrebno doći do direktorija gdje se nalazi CAD model (Slika 9.).



Slika 9. Uvoz CAD geometrije u Autodesk 3ds Max

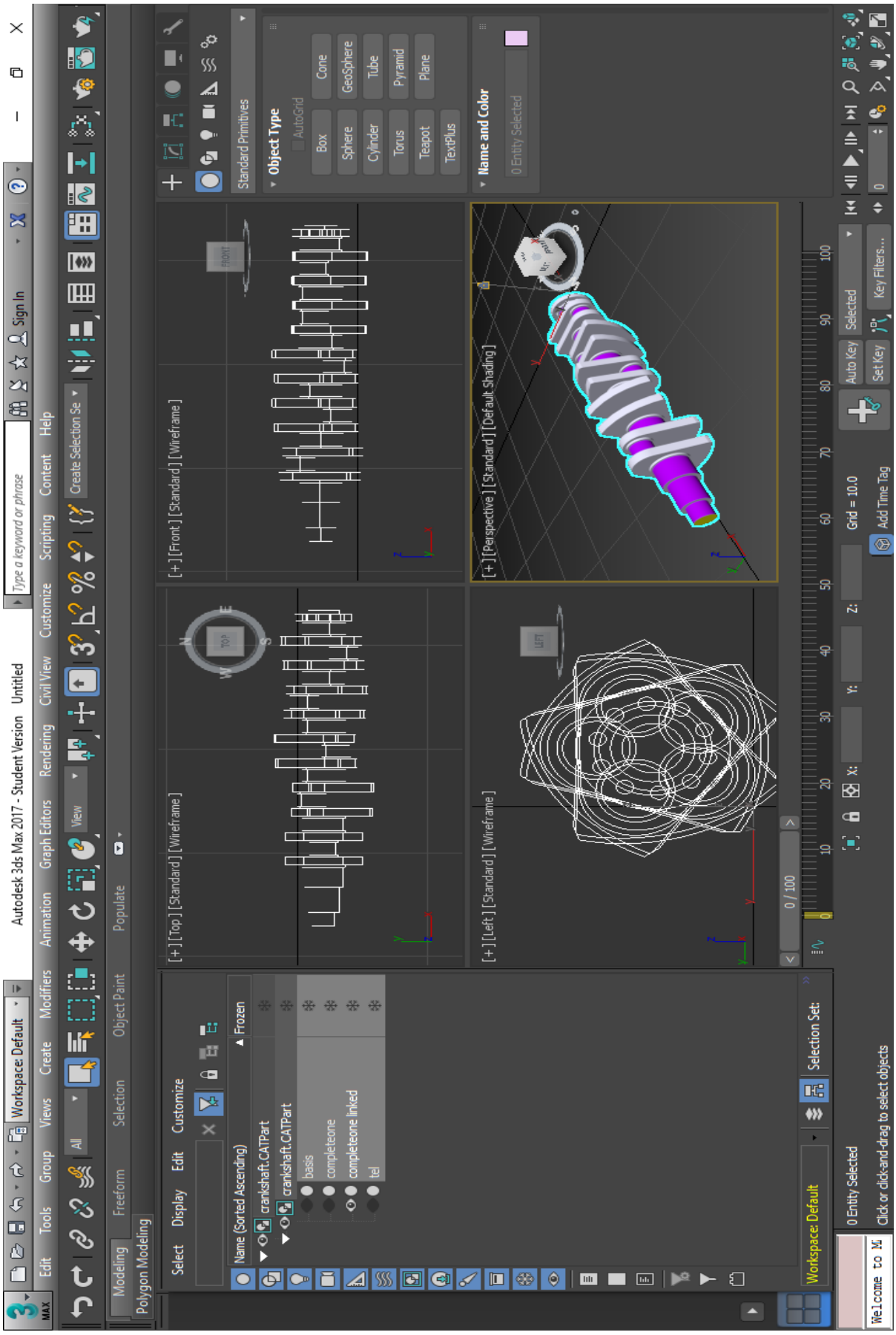
Odabirom *parta crankshaft* tj. koljenastog vratila, i klikom na *Open* otvorit će se dodatni izbornik koji omogućuje definiranje postavki uvoza odabranog dijela (Slika 10.). Zbog bržeg uvoza *parta* u okruženje postavka *Convert to Mesh,* koja iz odabranog modela generira mrežu, postavlja se na *off*. Usto, moguće je definirati i položaj osi *Z* (u ovom slučaju prema gore). Ostale postavke u ovome trenutku neće biti bitne. Ono što preostaje je kliknuti na gumb *Import* kako bi se dio učitao u programskom okruženju 3ds Max.



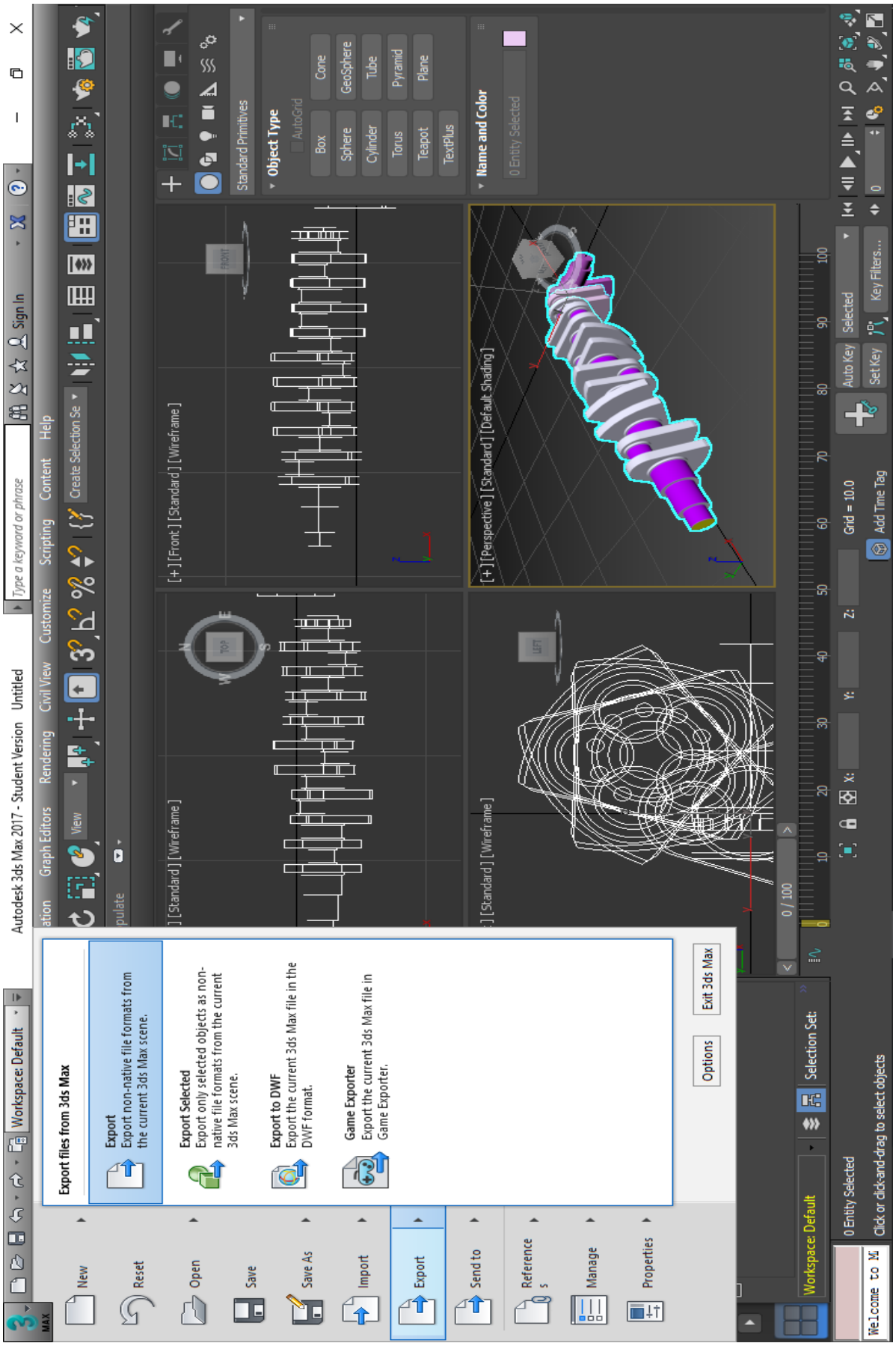
Slika 10. Postavke uvoza u Autodesk 3ds Max

3ds Max će nakon što je dio uvezen automatski prikazati njegove ortogonalne projekcije u četiri različita pravokutna prozorčića (Slika 11.). Prozorčić u donjem desnom uglu prikazuje izometrijski prikaz dijela.

Budući da se neće vršiti nikakva interakcija i modifikacija na učitanom *part*u, odmah se može krenuti na izvoz *part*a, kako bi se stvorio format pogodan za otvaranje u Unityju. Klikom na padajući izbornik i odabirom opcije *Export – Export non-native formats from the current 3ds Max scene* otvorit će se skočni prozor (Slika 12.). Odabrana opcija omogućuje izvoz formata koji nisu izvorno nastali u softveru Autodesk 3ds Max, u neki od željenih formata koje navedeni softver podržava.

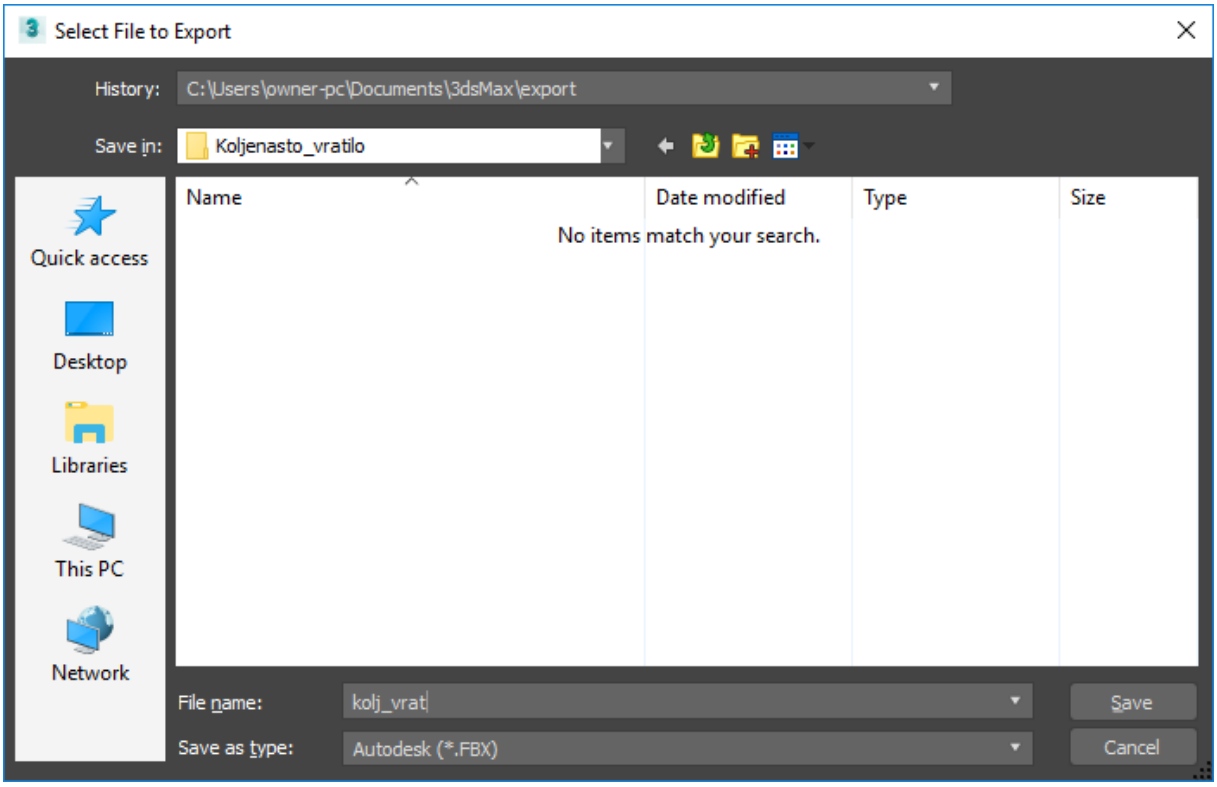


Slika 11. Dio uvezen u Autodesk 3ds Max



Slika 12. Postavke izvoza

U novootvorenom skočnom prozoru (Slika 13.) potrebno je definirati ime datoteke, što će u ovom slučaju biti kolj_vrat, te odrediti odredišni direktorij. Na dnu slike vidljivo je kako će datoteka biti spremljena u formatu .FBX, koji omogućuje interakciju u Unityju.



Slika 13. Odredišni direktorij

3.3. SOFTVER UNITY

Neke od spomenutih platformi koje su podržane od strane Unityja su naprimjer: Android, iOS, Windows Phone, Steam VR, Tizen, Mac, Linux, Gear VR, Playstation 4, Playstation VR, Oculus Rift, Microsoft HoloLens itd. Iako postoji mnoštvo razvijenih igara pomoću navedenog softvera, svakako je dobro spomenuti neke među najpoznatijima kao što su Angry Birds Epic, Angry Birds 2, Hitman Go, Pokemon Go. Ovo poglavlje će osim prikaza načina na koji se uvozi konvertirana geometrija u Unity detaljno prikazati i korake proširenja biblioteke koje je potrebno načiniti kako bi se programsko okruženje podesilo na ispravan način za razvoj aplikacije za željenu platformu VR. [21 i 26]

Interakcija CAD-a i Unityja u ovome trenutku nije moguća bez posrednika, što je i razumljivo jer Unity s klasičnim CAD sustavima nema nikakvog doticaja. Vjerojatno će tako ostati i ubuduće, međutim razlog odabira ovog softverskog paketa leži u činjenici da Unity omogućuje relativno lako i intuitivno kreiranje virtualnog okruženja. Uz Unity, Unreal Engine jedan je od rijetkih softvera koji također posjeduje mogućnost stvaranja aplikacija za VR. Bez korištenja takvih softvera, trenutačno ne bi bilo moguće prikazati dijelove generirane u bilo kojem CAD softveru na niskobudžetnom uređaju virtualne stvarnosti kao što je Google Cardboard. Prije ubacivanja konvertirane geometrije u Unity, potrebno je prilagoditi razvojnu platformu za mobilne sustave Android.

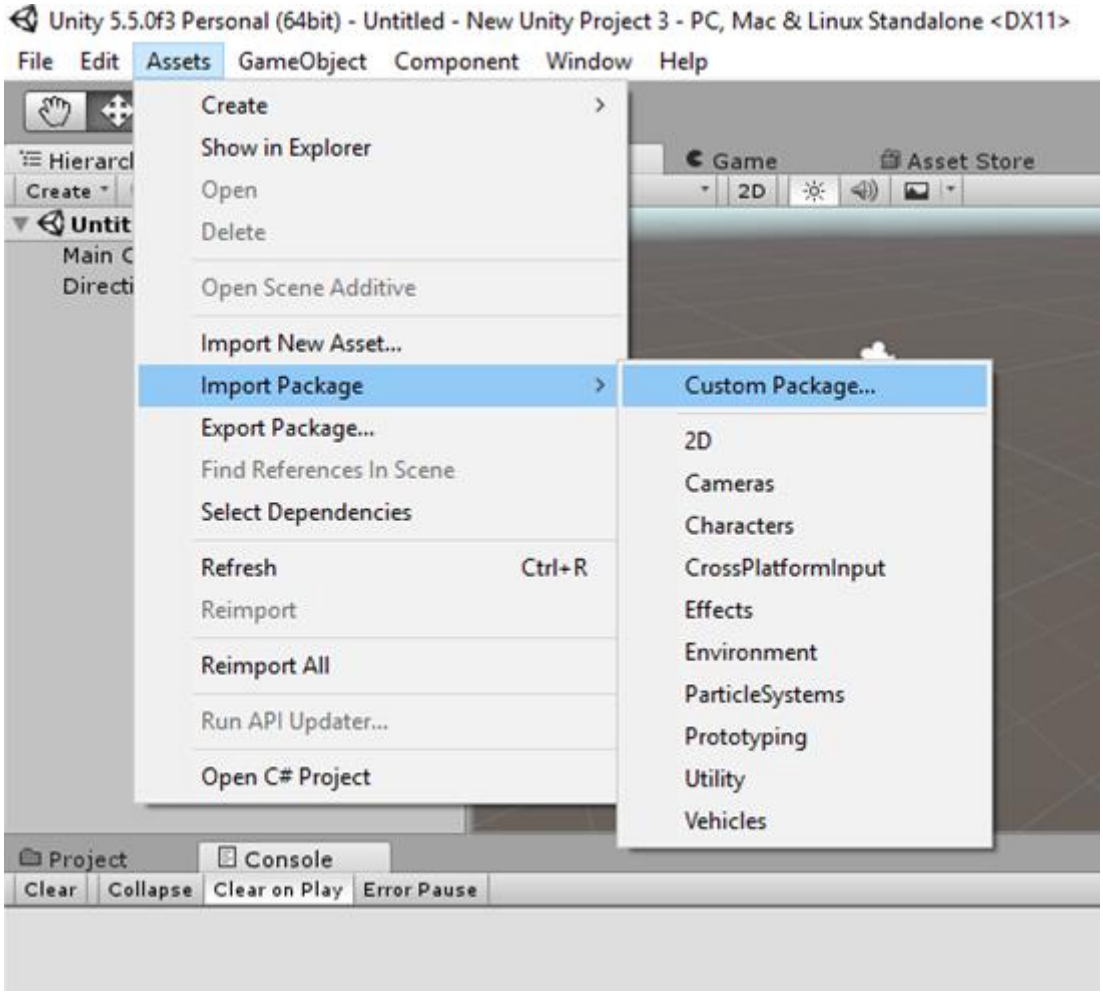
Nakon pokretanja Unityja, unaprijed zadane postavke automatski će pretpostaviti da se razvija aplikacija za Windows platformu. Kako bi se to promijenilo, potrebno je promijeniti postavke, što će biti prikazano u nastavku. Budući da je tehnologija VR vrlo nova, iako se koristi posljednja inačica softvera Unity, svejedno je potrebno instalirati određene dodatke kako bi se omogućio razvoj aplikacije za VR platformu, koja će biti pokretana na mobilnom sustavu Android putem aplikacije Google Cardboard.

3.3.1. Proširenje biblioteke i definiranje razvojne platforme

Kako bi se razvila aplikacija za Google Cardboard ili Daydream potrebno je instalirati *Daydream technical preview*. Daydream je platforma Google za visokokvalitetni mobilni VR. Isto tako, ukoliko se želi razviti aplikaciju za platformu Android, potrebno je prije svega instalirali Android SDK i omogućiti USB ispravljanje pogrešaka (*USB debugging*) na mobilnom telefonu, što je prikazano u točki *3.4.1*.

Nadalje, potrebno je preuzeti Google VR SDK za Unity. Navedeni paket omogućuje razvijanje mobilnih aplikacija za Android i iOS mobilne sustave, uz priložene demo scene koje je moguće isprobati prije generiranja bilo kakvog koda ili proizvoljne geometrije definirane od strane korisnika. Paket je moguće preuzeti na Google stranicama za razvojne programere (*developers.google.com*).

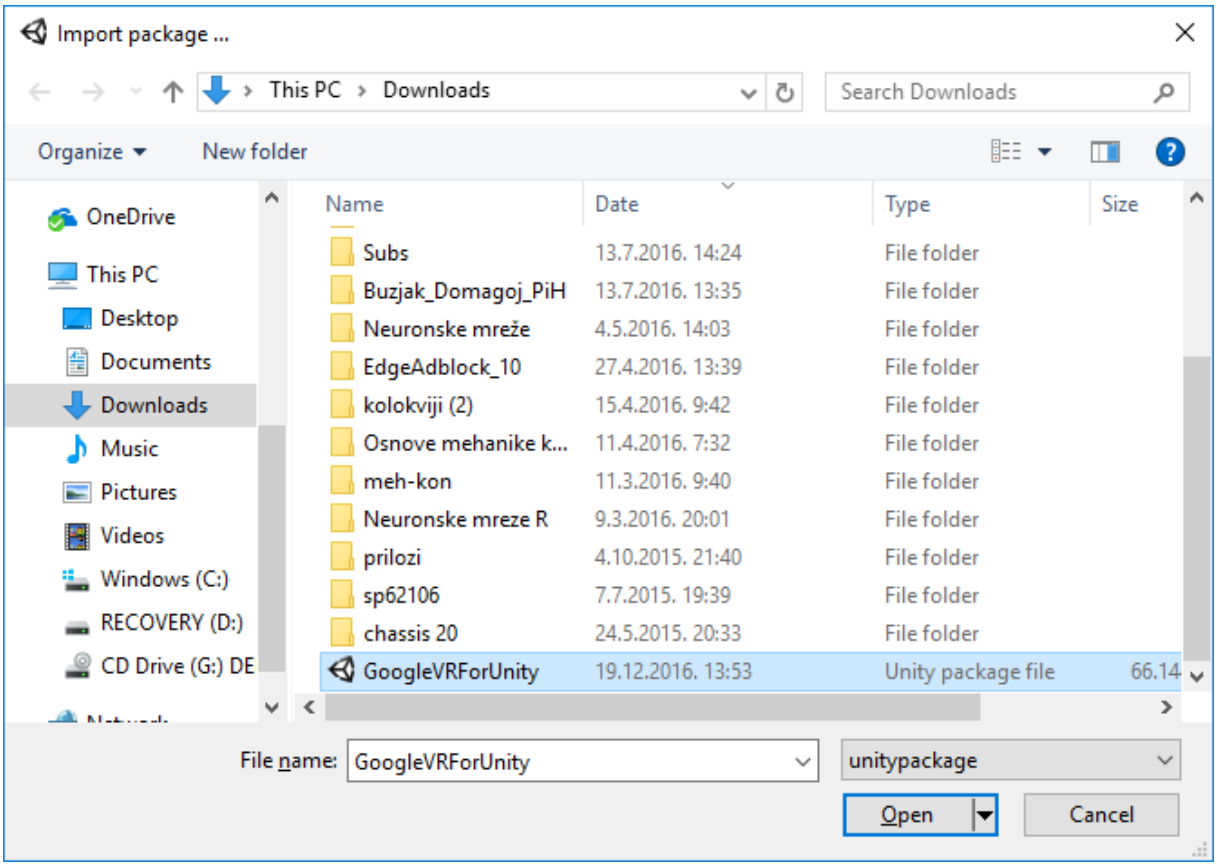
Nakon preuzimanja, navedeni paket potrebno je uključiti u biblioteku Unity kako bi softver dobio potrebne informacije o razvojnoj platformi. Nakon pokretanja Unityja i odabira 3D projekta, paket se dodaje na način prikazan slikom 14. Otvara se izbornik *Assets – Import Package – Custom Package.*



Slika 14. Uvoz paketa u Unity

Klikom na *Custom Package* otvara se skočni (*pop-up*) prozor (Slika 15.) u kojem je potrebno odabrati lokaciju na disku na kojoj je spremljena datoteka *GoogleVRForUnity.unitypackage*. Navedena opcija služi za proširenje biblioteke Unity te otvara dodatne mogućnosti za razvoj aplikacija za određenu vrstu platforme. U ovome slučaju, proširenje će omogućiti opcije poput praćenja kretnji korisnikove glave, detekcije korisnikove interakcije sa sustavom (primjerice putem gumba ili kontrolera), distorzijske korekcije za leće virtualnog sustava,

razdvajanja ekrana na dva dijela kod *game* prikaza (kako bi se simulirao prikaz istovjetan onome na pametnom telefonu) i slično.



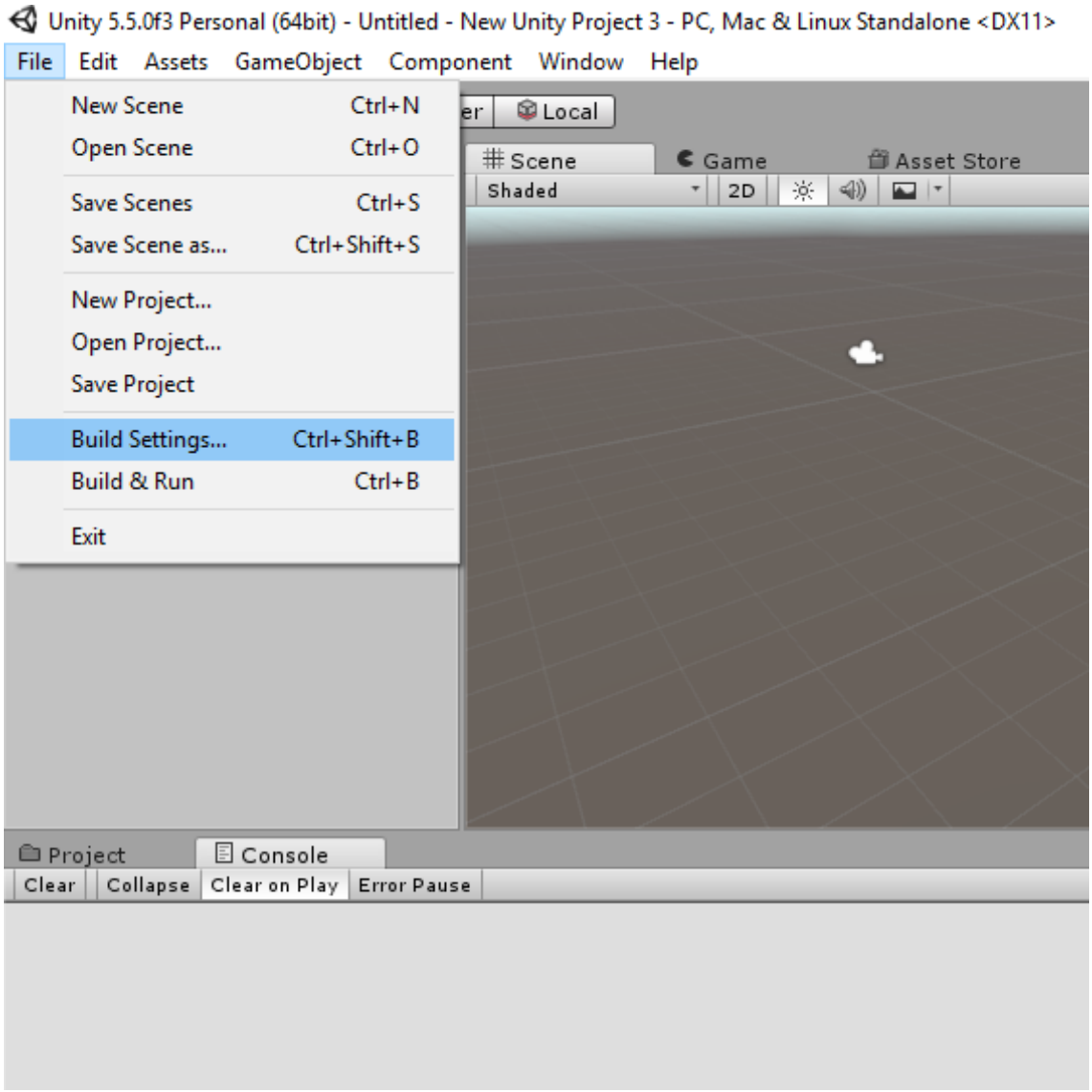
Slika 15. Lokacija paketa

Pronalaskom navedene datoteke na disku i klikom na *Open* Unity će započeti raspakiranje *package* datoteke te će se nakon nekog vremena na ekranu prikazati prozor *Import Unity Package* (Slika 16.).

Slika 16. Paket GoogleVRForUnity

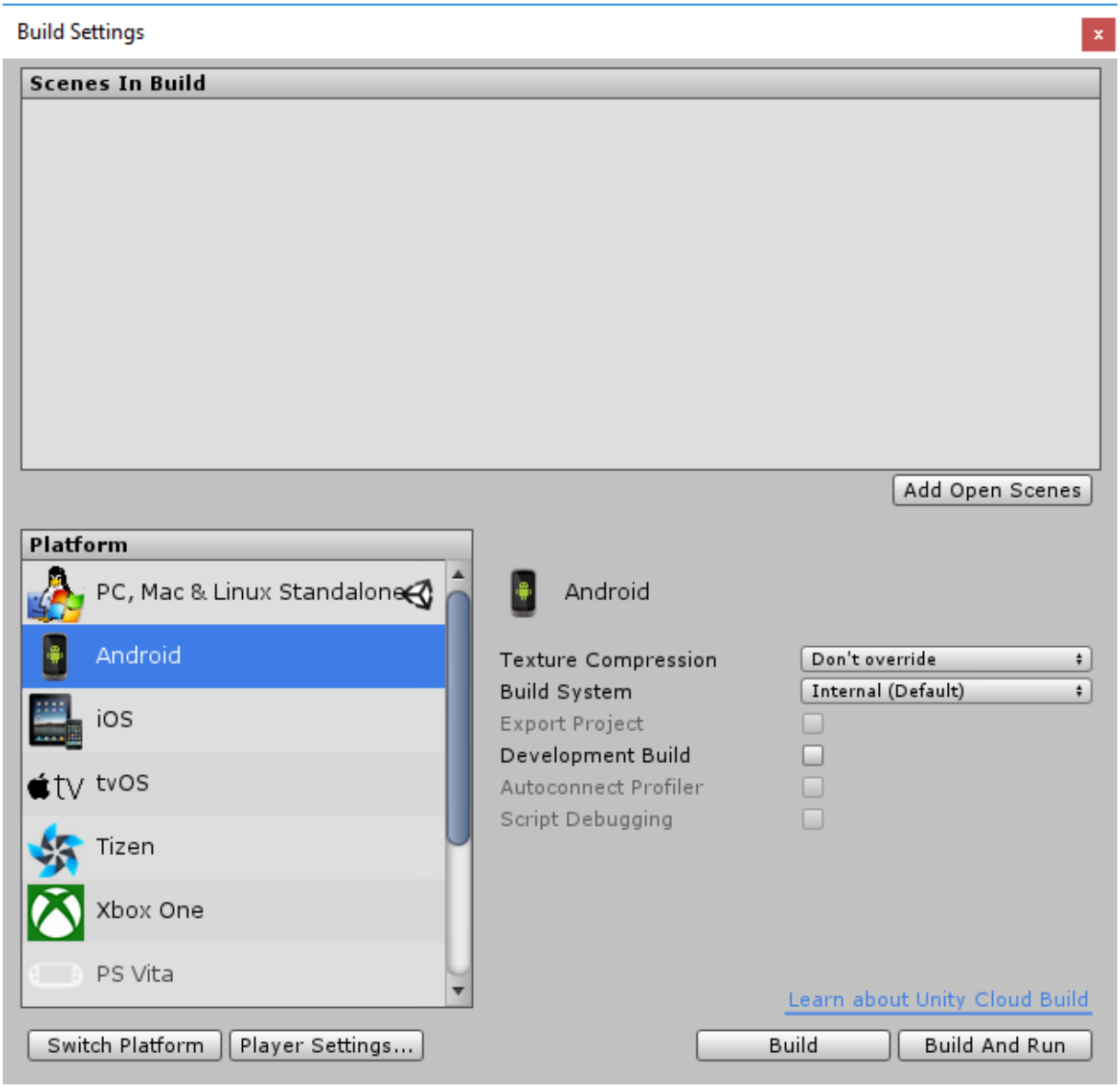
Jedino što preostaje je kliknuti na *Import* imajući u vidu da sve kućice na slici imaju kvačicu. Ukoliko bi izostala kvačica na nekoj od kućica, *package* datoteka ne bi u potpunosti bila uključena u Unity, što može rezultirati greškom u radu programa, naročito ako se kvačica izostavi na važnim *script* datotekama koje sadrže kod za generiranje okruženja VR.

Uvoz navedenog paketa u Unity omogućio je proširenje biblioteke i dao mogućnost stvaranja aplikacije za željenu platformu Google Cardboard. Međutim, da bi se promijenila unaprijed zadana platforma također je potrebno izmijeniti *Build* postavke na način prikazan u nastavku (Slika 17.). Klikom na izbornik *File – Build Settings* otvorit će se prozor u kojem je moguće definirati razvojnu platformu i njene karakteristike.



Slika 17. Postavke *Build*

Nakon otvaranja izbornika *Build Settings* pod sekcijom *Platform* (Slika 18.) odabere se Android te se u donjem lijevom uglu klikne na opciju *Switch Platform*. Na taj način promijenit će se platforma s unaprijed definirane razvojne platforme namijenjene za PC, Mac i Linux operativne sustave na željenu razvojnu platformu Android.

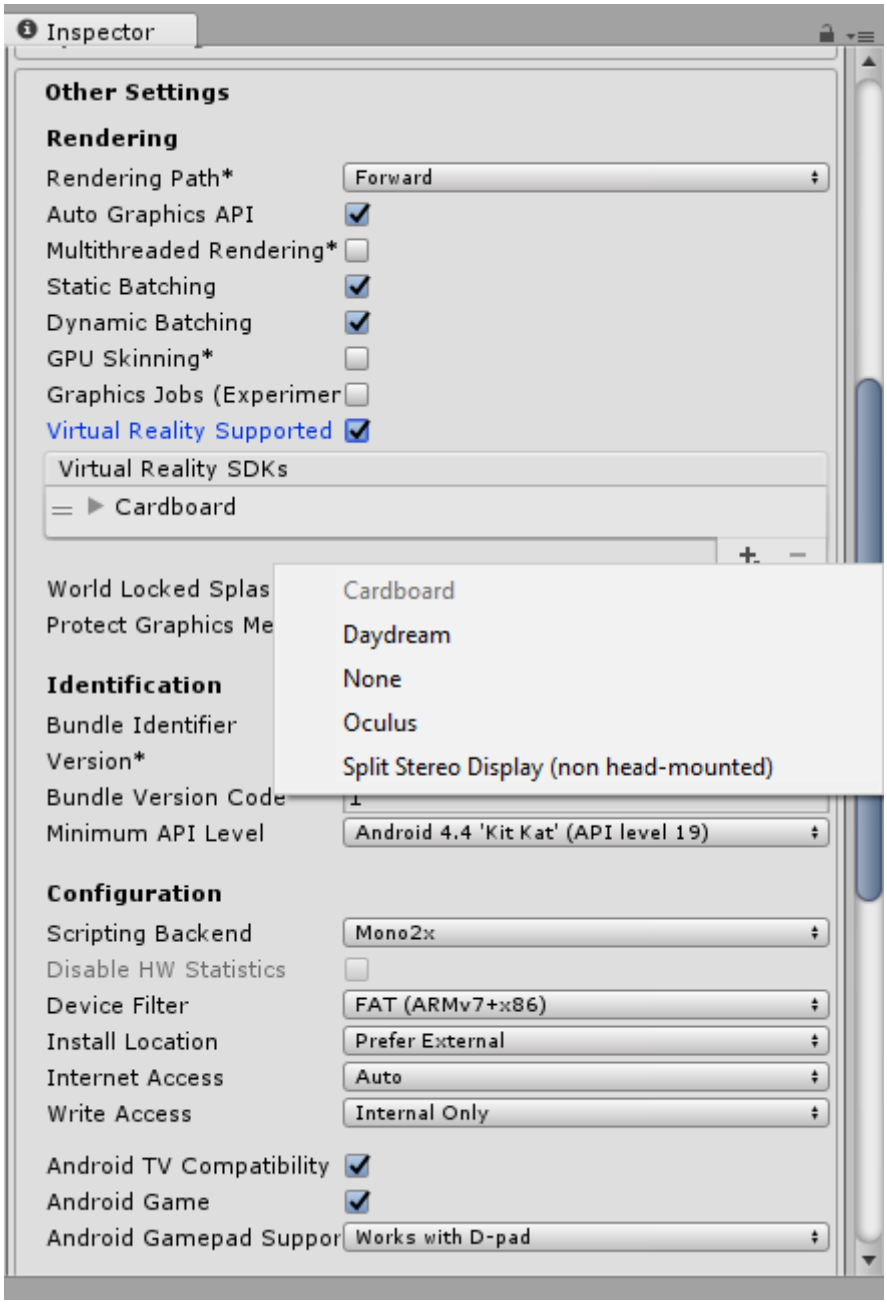


Slika 18. Promjena platforme

Nakon što je platforma uspješno promijenjena, potrebno je kliknuti na gumb *Player Settings* u kojem će se omogućiti podrška za sustave virtualne stvarnosti. Otvorit će se izbornik kao na slici 19.

U podsekciji *Ostale postavke* potrebno je omogućiti razvoj aplikacije za platformu VR klikom na *Virtual Reality Supported*. U podizborniku Virtual Reality SDKs moguće je odabrati željenu platformu. S obzirom da se razvija aplikacija za Google Cardboard, odabere se *Cardbord*. U istom izborniku moguće je definirati i ostale postavke poput minimalnog podržanog

operativnog sustava. U ovom slučaju odabrana je podrška za Android 4.4 Kit Kat verziju (API level 19). [27 i 28]

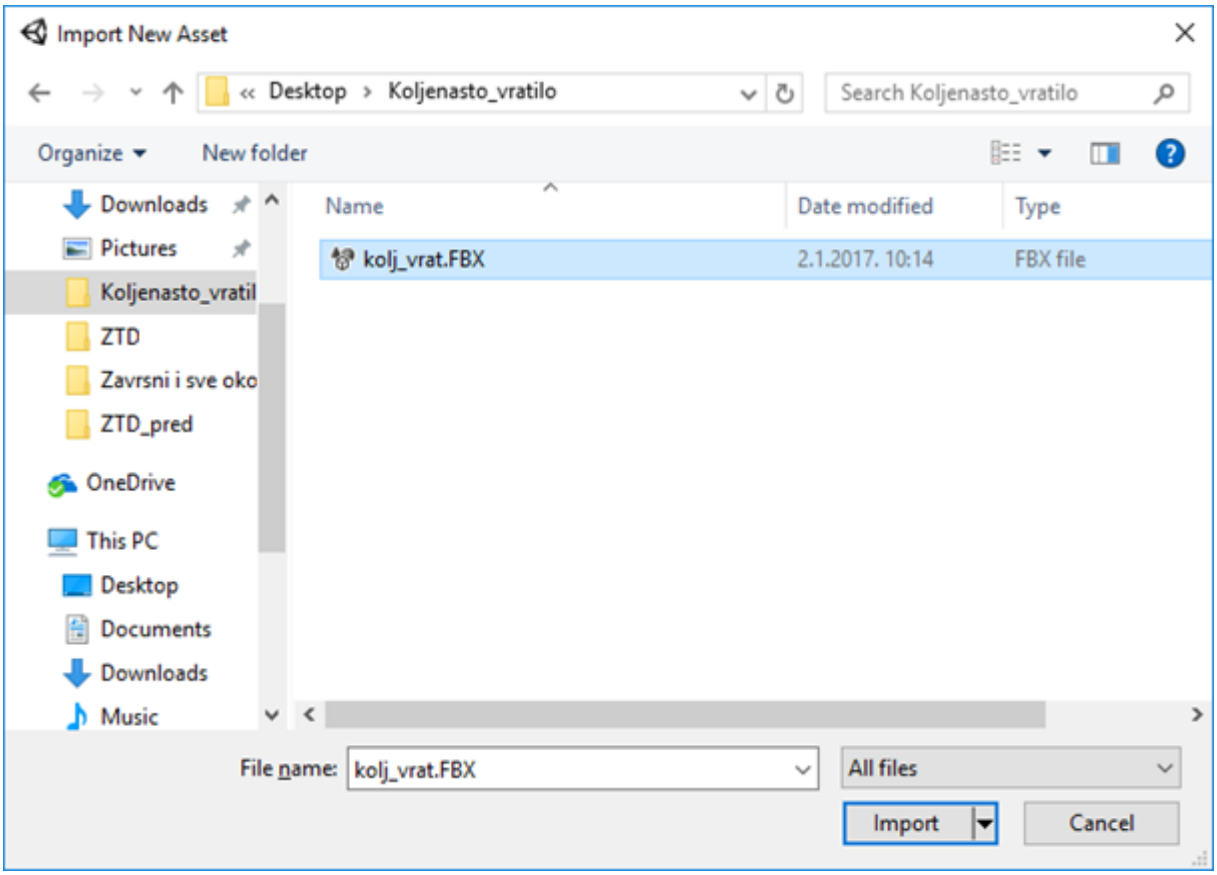


Slika 19. Odabir *Google Cardboard* platforme

Tek nakon što su definirane prethodno navedene postavke moguće je krenuti s uvozom konvertirane CAD geometrije u Unity.

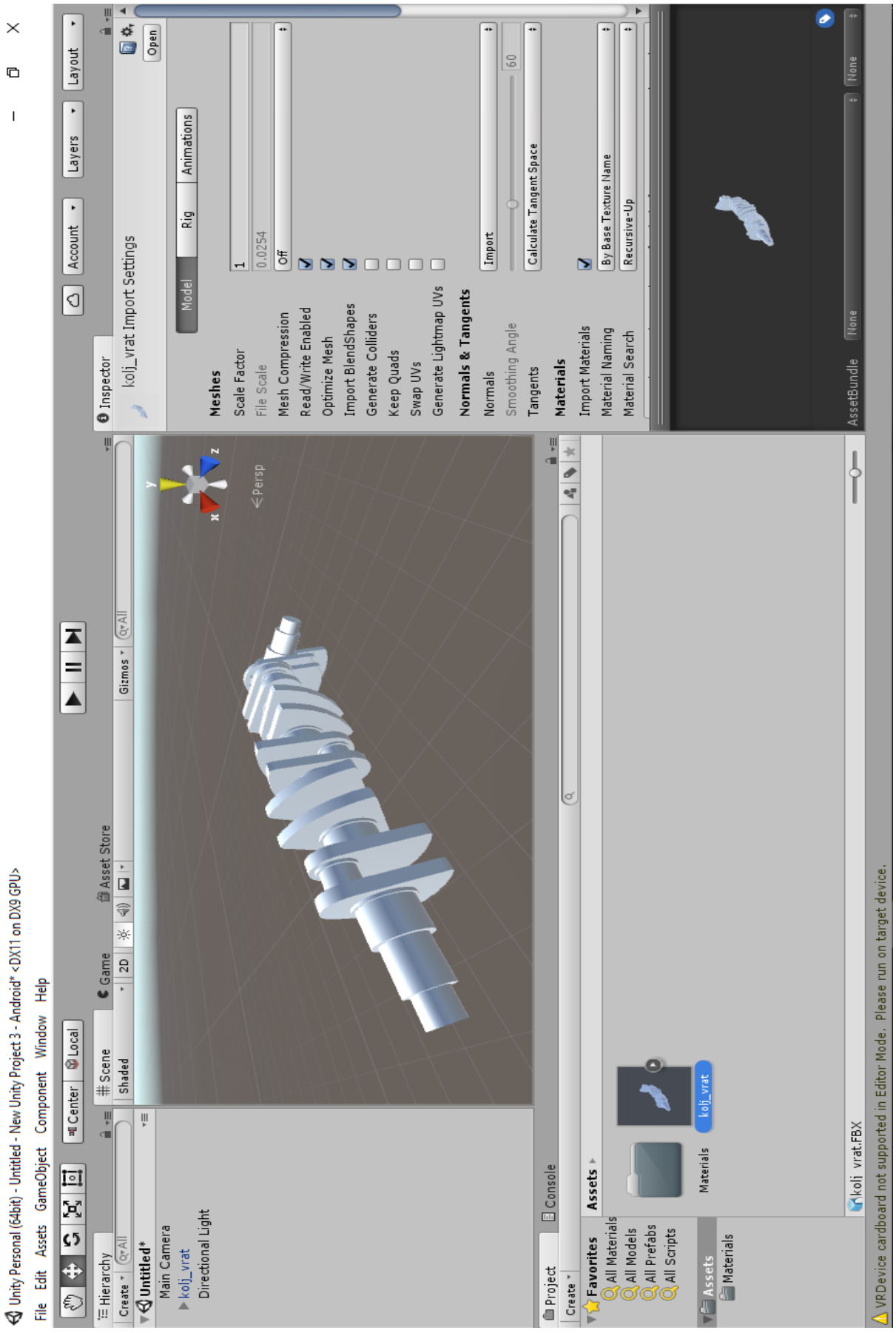
3.3.2. Uvoz konvertirane geometrije

Uvoz datoteke moguće je napraviti na više načina. Najjednostavniji način je *drag and drop* princip, prema kojem se željena datoteka povuče mišem bilo gdje u prozor Unity te se na taj način automatski uključuje u radno okruženje. Drugi način preko odredišnog direktorija bit će prikazan u nastavku. Klikom na *Assets-Import New Asset* otvorit će se skočni prozor (Slika 20.).



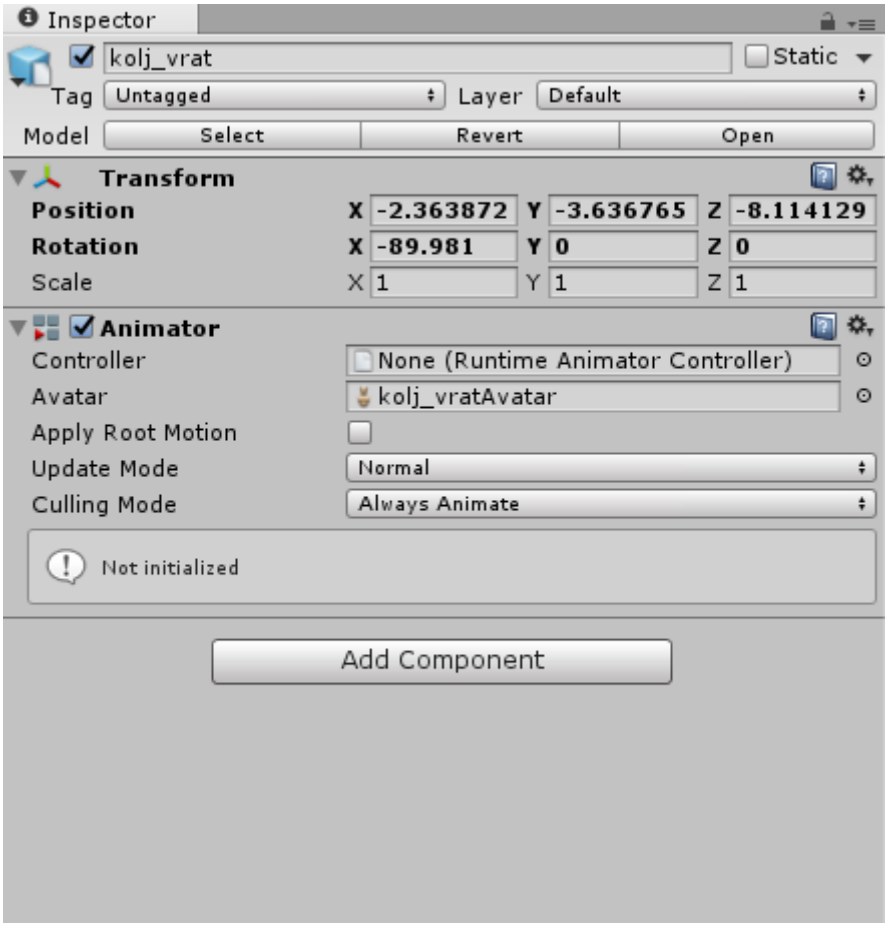
Slika 20. Lokacija datoteke na disku

Potrebno se pozicionirati u direktorij u kojem je spremljena konvertirana datoteka u formatu .FBX. Nakon pozicioniranja, potrebno je odabrati željenu datoteku i kliknuti na *Import*. Ukoliko je sve napravljeno u redu i prema uputama, koljenasto vratilo bi trebalo biti učitano u sekciju Unity Assets. Da bi se koljenasto vratilo prikazalo u 3D projekciji kao na slici 21., potrebno je datoteku kolj_vrat.FBX učitanu u *Assets* sekciji jednostavno mišem povući bilo gdje u prozorčić *Scene*. Na taj način će koljenasto vratilo biti prikazano na sceni koja je vidljiva u trenutku pokretanja aplikacije (Slika 21.).



Slika 21. Koljenasto vratilu u Unityju

Prilikom premještanja koljenastog vratila i puštanja u prozorčić *Scene*, na desnoj strani ekrana otvorit će se dodatni izbornik (Slika 22.). Pozicija koljenastog vratila automatski je određena pozicijom ispuštanja na sceni. U *Transform* sekciji, pod dijelom *Position* određene su koordinate ispuštanja dijela. Zbog jednostavnosti i lakšeg određivanja buduće pozicije vratila, vrijednosti koordinata *X*, *Y* i *Z* postavljaju se na 0. Isto se čini i u dijelu *Rotation* koji određuje rotaciju komponente u odnosu na određenu koordinatnu os. U dijelu *Scale* moguće je skalirati odnosno mijenjati dimenzije predmeta po pojedinoj koordinatnoj osi. Ukoliko se primjerice vrijednost koordinatne osi *X* postavi na 5 umjesto 1, koljenasto vratilo bit će uvećano za pet puta u smjeru osi *X,* što će u ovome slučaju značiti da će Unity povećati duljinu dijela za pet puta od one definirane u izvornom CAD modelu. Vrijednosti svih koordinatnih osi na dijelu *Scale* ostavljaju se na numeričkoj vrijednosti jednakoj 1.



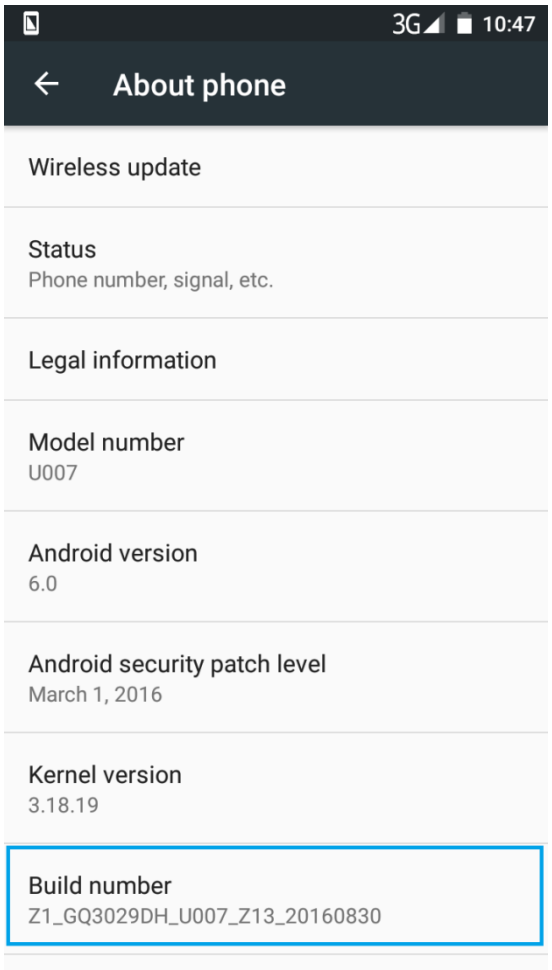
Slika 22. Definiranje dimenzija

3.4. IZRADA APLIKACIJE ZA PLATFORMU GOOGLE CARDBOARD

Nastavak detaljno prikazuje korake i rutine koje je potrebno napraviti kako bi se stvorila aplikacija za platformu Google Cardboard. Kroz tekst su izneseni problemi do kojih može doći kod stvaranja aplikacije, ali su opisana i njihova rješenja. S obzirom da je u dijelu 3.3. Softver UNITY prikazan način pripreme platforme za razvoj aplikacije VR na mobilnom sustavu Android, ovdje je sadržan popis rutina koje je potrebno načiniti na mobilnom telefonu kako bi se otključale opcije za razvojne programere, te samim time omogućio prijenos aplikacije na mobilnu platformu.

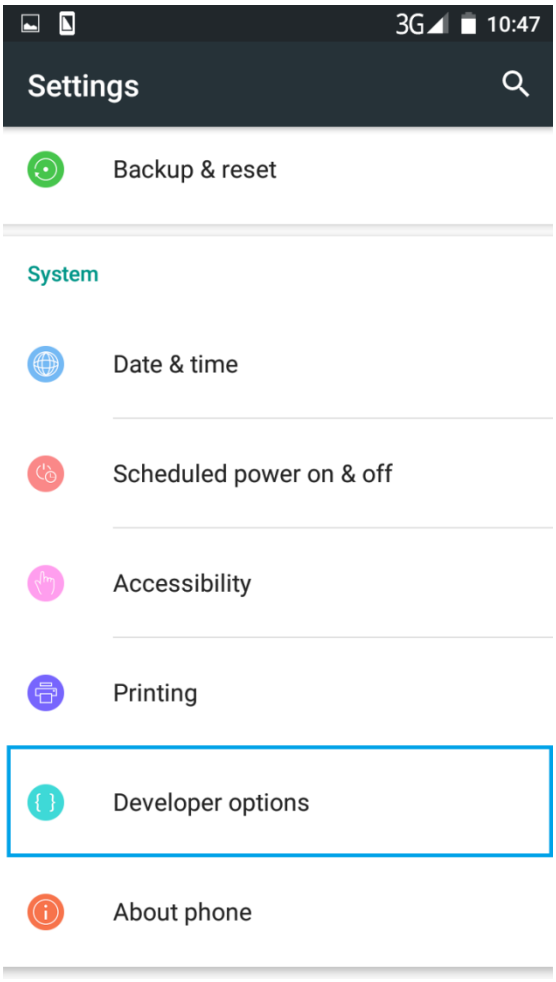
3.4.1. *Android Developers*

Da bi uopće bilo moguće napraviti mobilnu aplikaciju za mobilni sustav Android, potrebno je instalirati Android Studio na računalo na kojem će se razvijati aplikacija. Nadalje je potrebno na mobilnom telefonu omogućiti tzv. *USB Debugging* tj. uklanjanje pogrešaka. Ovisno o vrsti mobitela i proizvođaču, ne postoji jedinstven način otključavanja *USB Debugging* moda koji bi bio univerzalan za sve vrste mobilnih uređaja, međutim za veliku većinu pametnih telefona koristi se pristup koji će biti prikazan u nastavku (Slika 23.). Potrebno je ući u postavke telefona (eng. *Settings*), te doći u izbornik O telefonu (eng. *About phone*). Izbornik se obično nalazi pri dnu ekrana. Nakon ulaska u izbornik u kojem su prikazane informacije o telefonu, potrebno je pronaći dio nazvan Build number (označen plavo na slici 23.). Da bi se došlo do naprednih razvojnih opcija, potrebno je sedam puta zaredom pritisnuti na *Build number*. Odmah će se prikazati skočni prozor na kojem će biti prikazana informacija koliko je još puta potrebno pritisnuti na *Build number* kako bi se otključale dodatne opcije za programere.



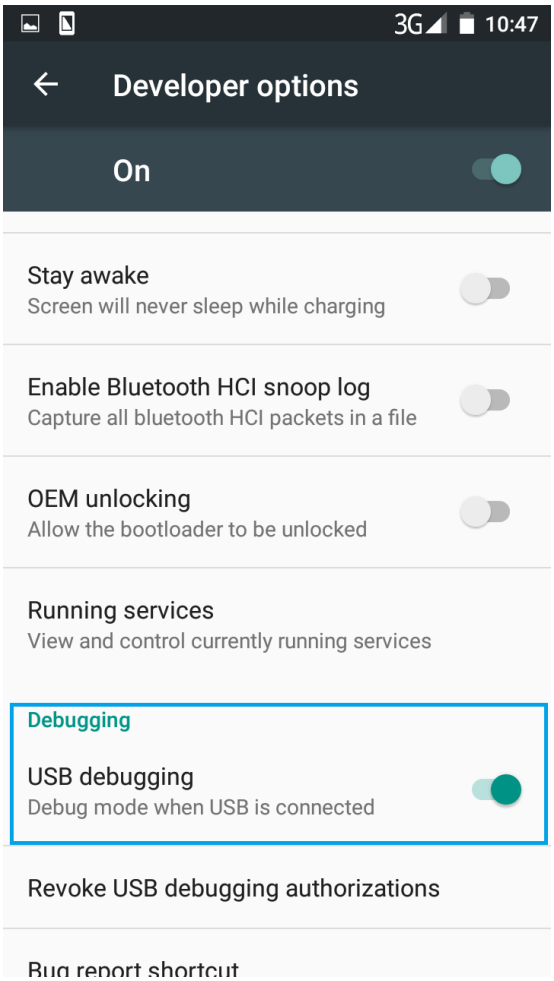
Slika 23. *Build* broj

Nakon sedmog pritiska otključane su programerske opcije. Kako bi se došlo do dodatnog izbornika potrebno se vratiti u roditeljski direktorij s postavkama. Pod izbornikom *System* nalazi se novi podizbornik koji nosi ime *Developer options* (Slika 24.). Navedeni izbornik prije nije bio vidljiv, jer je bio zaključan, te se prikazanim postupkom došlo do otključavanja izbornika a samim time i do dodatnih opcija.



Slika 24. Programerske opcije

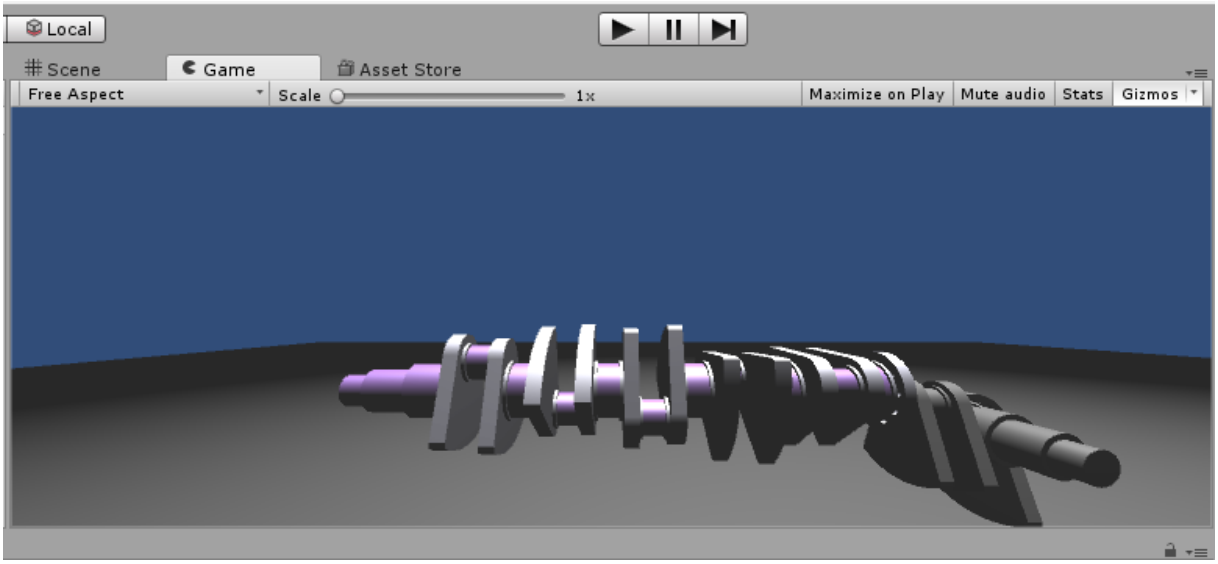
Potrebno je ući u navedeni izbornik te uključiti napredne opcije koje će omogućiti razvoj i pokretanje aplikacije na mobilnom telefonu. U podizborniku *Debugging* potrebno je uključiti opciju *USB Debugging* (Slika 25.). Time je dovršeno modificiranje postavki na mobilnom telefonu koje omogućuju razvoj i pokretanje novih aplikacija. [29]



Slika 25. ***USB Debugging***

3.4.2. *Unity app builder*

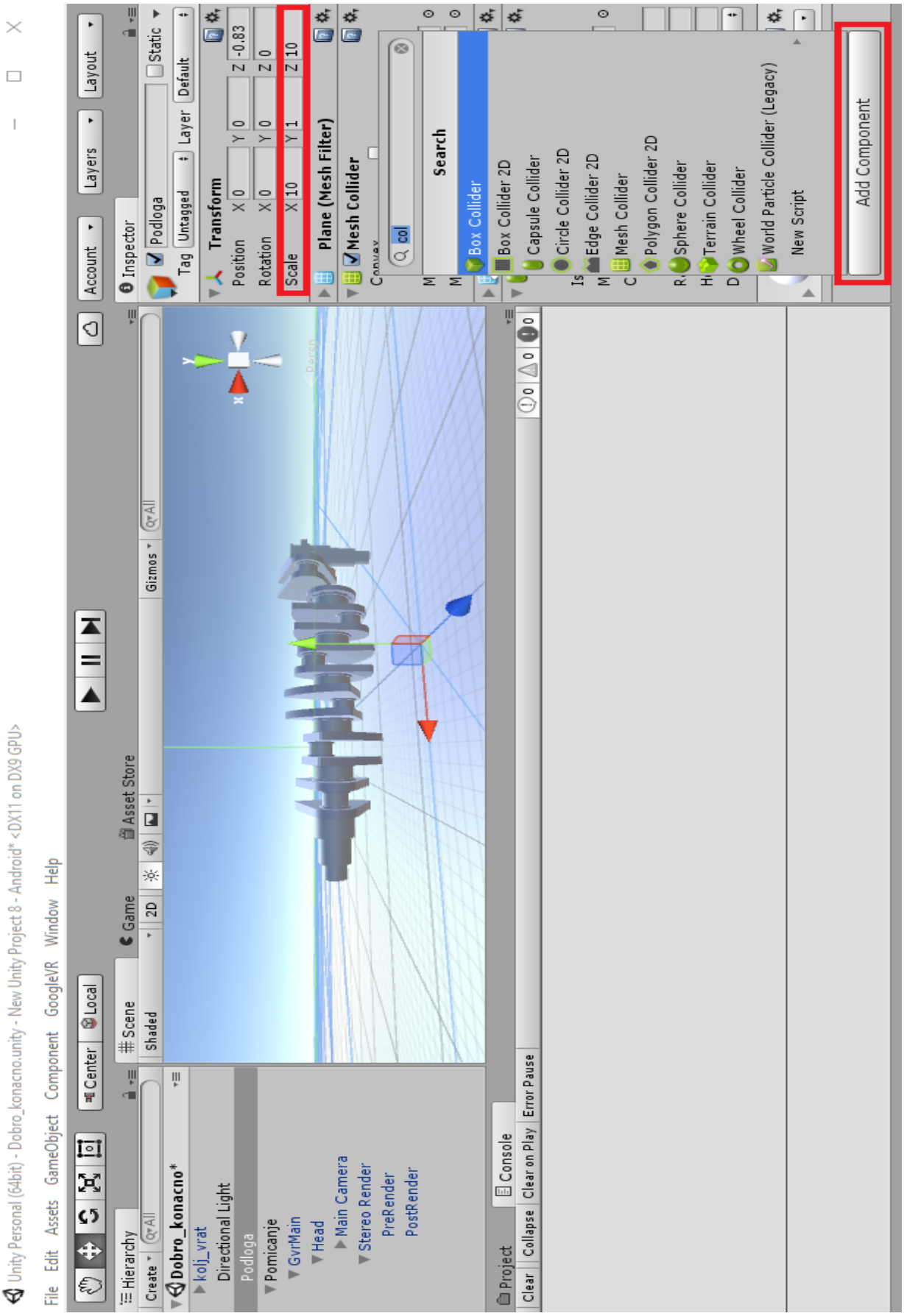
S obzirom na specifičnost Unityja odmah nakon što je koljenasto vratilo ubačeno u program, moguće je pokrenuti scenu klikom na gumb *Play* koji se nalazi na vrhu prozora. Klikom na *Play* generirat će se poseban (*Game*) prikaz koji će simulirati okruženje potpuno istovjetno onome kakvo bi bilo na željenoj razvojnoj platformi. Konkretno, u ovome slučaju na monitoru će se stvoriti prikaz istovjetan onome kakav će biti na Google Cardboardu (Slika 26.).



Slika 26. Prikaz koljenastog vratila u Unity *Game*u

Nakon klika na *Play* vidljivo je kako je koljenasto vratilo učitano u novom okruženju, međutim bez mogućnosti posebne manipulacije i interakcije. Da bi se to promijenilo, odnosno omogućilo kretanje u prostoru, potrebno je u okruženju napraviti određene modifikacije. Modifikacije će uključivati definiranje zakonitosti između objekata, dodavanje skripte za određivanje rutine kretanja ali i definiranje podloge kao krutog tijela preko *Collider* opcije kako u trenutku uključivanja lika (eng. *character*) u prostor u prikazu *Game* ne bi dolazilo do propadanja lika u smjeru osi *Y* zbog djelovanja gravitacije i pozadinske fizike koja je sastavni dio programa.

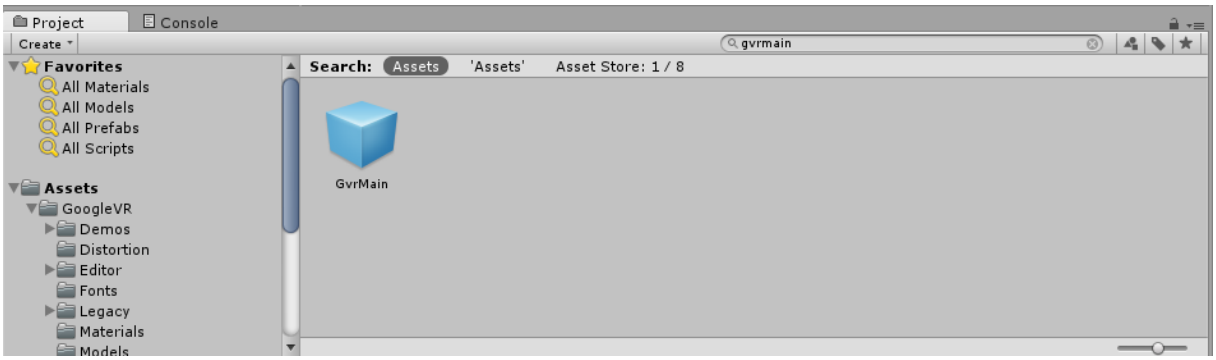
Na slici 27. vidljivo je kako je podloga izvučena u smjeru ravninskih osiju *X* i *Z*, te je skalirana za 10. To znači da su dimenzije u smjeru odabranih osi uvećane 10 puta od unaprijed zadanih dimenzija. Veličina podloge u ovom slučaju određuje prostor koji je dostupan za kretanje



Slika 27. Skaliranje podloge

unutar virtualnog okruženja. Da bi kretanje bilo moguće, potrebno je dodati ograničenje koje će onemogućiti propadanje. Kada bi se u okruženje dodao lik bez definiranog ograničenja na podlozi, došlo bi do njegovog slobodnog pada u smjeru vertikalne osi, što ne bi rezultiralo željenim prikazom. Zbog toga je podlozi potrebno pridodati atribut koji sprječava propadanje. To će se učiniti dodjeljivanjem svojstva *Collider*. Postupak dodavanja *Collider*a je sljedeći. Nakon što je podloga označena u stablu *Hierarchy*, u izborniku *Inspector* odabere se opcija *Add Component*. U tražilicu se upisuje *Collider* zbog lakšeg pronalaska opcije. Moguće je dodati bilo koji 3D *Collider*, pa je u ovome slučaju odabran *Box Collider*. Na taj način podlozi je dodijeljeno svojstvo krute površine, što rezultira onemogućivanjem propadanja lika u smjeru osi *Y* u trenutku pokretanja *Game* prikaza.

U sekciji *Hierarchy*, potrebno je izraditi novi objekt kojem će se dodijeliti svojstva za interakciju u kasnije generiranom virtualnom okruženju. Desnim klikom u odabranoj sekciji, otvara se izbornik u kojemu se odabire *Create Empty*. Na taj način u stablo je dodan novi *Game Object* bez unaprijed definiranih svojstava. Standardan naziv *Game Object* promijenit će se u naziv *Pomicanje*. Stvorenom objektu će se pridijeliti određena svojstva koja će služiti za kretanje po prostoru. Da bi prikaz bio realističan i definiran iz prvog lica, iz stabla je potrebno komponentu *Main Camera* prebaciti u objekt *Pomicanje*. Time je unutar strukture definirana relacija *Parent–Child*. Da bi prikaz bio razdvojen kao na *Google Cardbord*u, tj. da se generirano okruženje podijeli na razdvojene zaslone (tzv. *split screen*), potrebno je iz prije dodane Googleove biblioteke dodati značajku *GvrMain* kao *Child* komponentu objektu *Pomicanje*. Najlakši način za pronalazak značajke *GvrMain* je koristeći tražilicu u sekciji *Project* (Slika 28.). *GvrMain* sadrži rutine za praćenje postavki virtualnog okruženja. Sastoji se od dvije podrutine - *Head* i *Stereo Render*. U podrutini *Head* sadržane su informacije za praćenje pokreta korisnikove glave, što će rezultirati promjenom prikaza i zakretanjem kamere u smjeru ovisnom o smjeru gledanja korisnika. U podrutini *Stereo Render* sadržane su značajke za generiranje prikaza. [30]

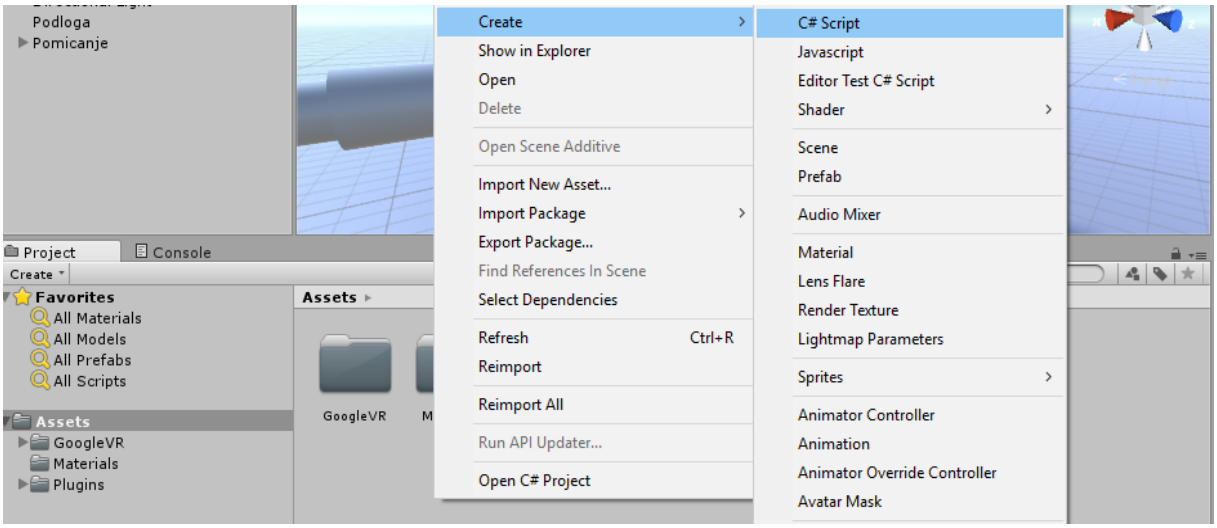


Slika 28. Dodavanje značajke *GvrMain*

Da bi se u virtualnom okruženju simuliralo kretanje, Unity pruža mogućnost korištenja skripte.

Skripte su tekstualne datoteke koje sadrže programski kod, napisan u nekim od programskih jezika (C#[15] i JS), čije korištenje omogućuje prilagođavanje okruženja specifičnim potrebama korisnika. Na taj način moguće je definiranje kretanja objekta *Pomicanje* koji je napravljen u programskom stablu. Velika prednost Unityja je besplatna online dostupnost različitih vrsta skripata za definiranje prikaza, geometrije, fizike i ostalih zakonitosti koje su potrebne kod razvoja aplikacija za sve podržane vrste platformi. Kako je u virtualnom okruženju bitna interakcija i dojam uključenosti u prostor, cilj je omogućiti pristup objektu sa svih strana. Zbog toga će se objektu *Pomicanje* dodijeliti nova skripta, imena *Autowalk* u kojoj će biti definiran kod za kretanje unutar virtualnog okruženja. Nova skripta može se izraditi u *Project* sekciji, desnim klikom miša bilo gdje u praznom prostoru, te klikom na *Create – C# Script* (Slika 29.). Na taj način kreirana je C# skripta.

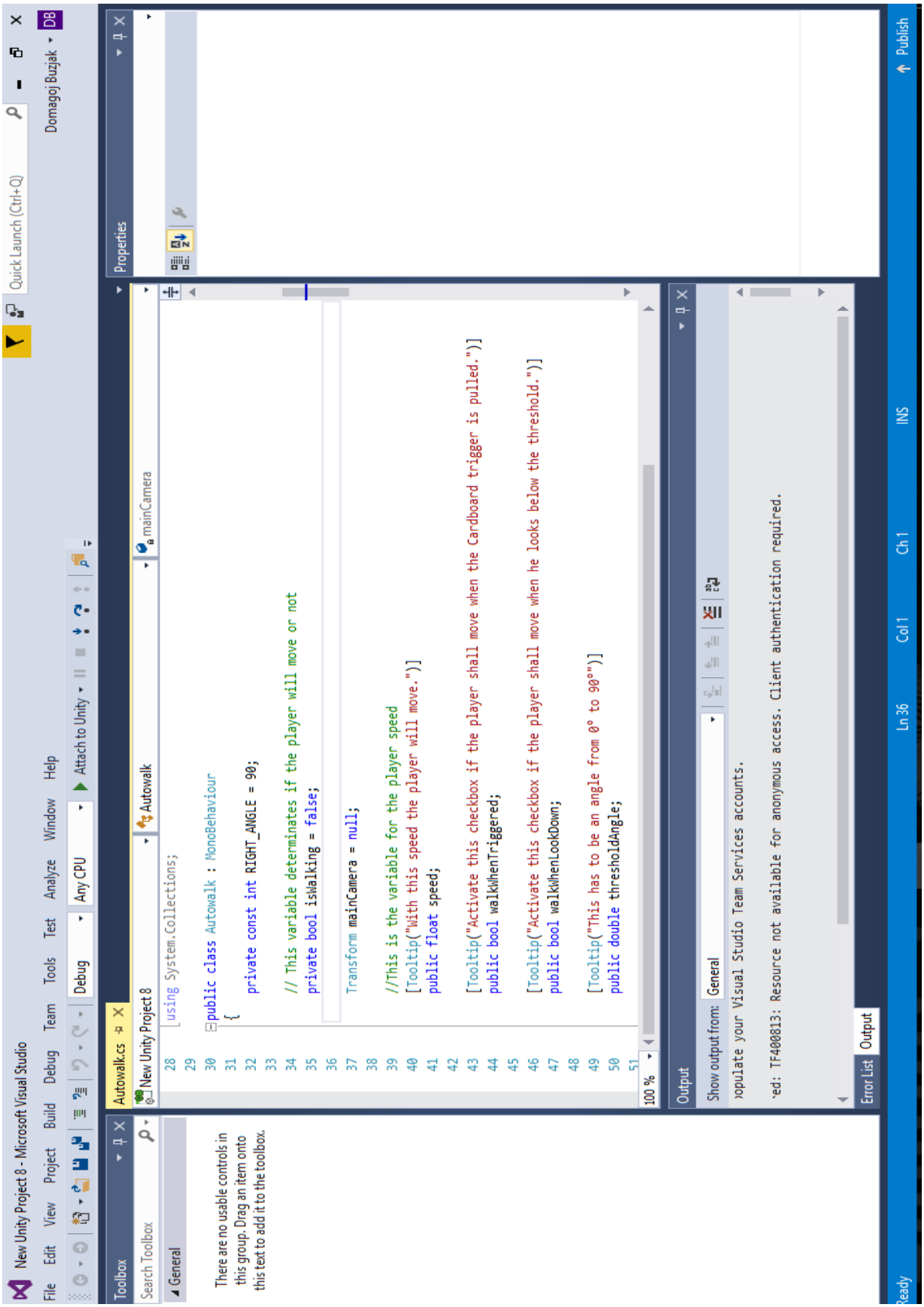
[15] Čita se *C Sharp*.



Slika 29. Kreiranje nove skripte

Imenovanjem skripte u naziv *Autowalk* te dvostrukim klikom miša ili klikom na gumb *Open* u izborniku *Inspector*, otvorit će se Microsoft Visual Studio koji omogućuje pisanje koda za programski jezik C#. C# je objektno-orijentiran programski jezik nastao 2000. godine u sklopu Microsoftove .NET inicijative, koji je kasnije prihvaćen kao standard po ISO i ECMA[16] normi. Microsoft Visual Studio predstavlja integrirano razvojno sučelje koje služi za razvoj programa na Windows platformi. Visual Studio ima podršku za razne programske jezike poput C, C++, VB.NET, C#, gdje je u ovome slučaju najbitnija podrška za C# u kojem će biti napisana skripta za kretanje lika unutar virtualnog okruženja (Slika 30.). [31 i 32]

[16] Europska udruga za standardizaciju informacijskih i komunikacijskih sustava. [46]



Slika 30. Programski kod u Visual Basicu

Budući da se kretanje može realizirati na više načina, ovdje će biti korištena kombinacija koja omogućuje kretnju na principu više različitih mehanizama, koji po potrebi mogu raditi zasebno ili zajedno, ovisno o želji korisnika. Putem prvog mehanizma, kretanje je omogućeno pritiskom na tipku *Cardboard*a. Drugi mehanizam složeniji je i teži za realizaciju, ali omogućuje automatsko kretanje u trenutku kada korisnik zakrene glavu za određeni kut. U ovome primjeru graničnu vrijednost kuta može proizvoljno odrediti korisnik. Primjerice, u trenutku kada je kut manji od 30 ° u odnosu na normalu, prikaz, tj. kretanje se neće odvijati automatski, dok će se za slučaj kada je kut veći od 30 ° kretanje odvijati automatski. Pregled skripte i opisi pojedinih funkcija dani su izlistom 1.

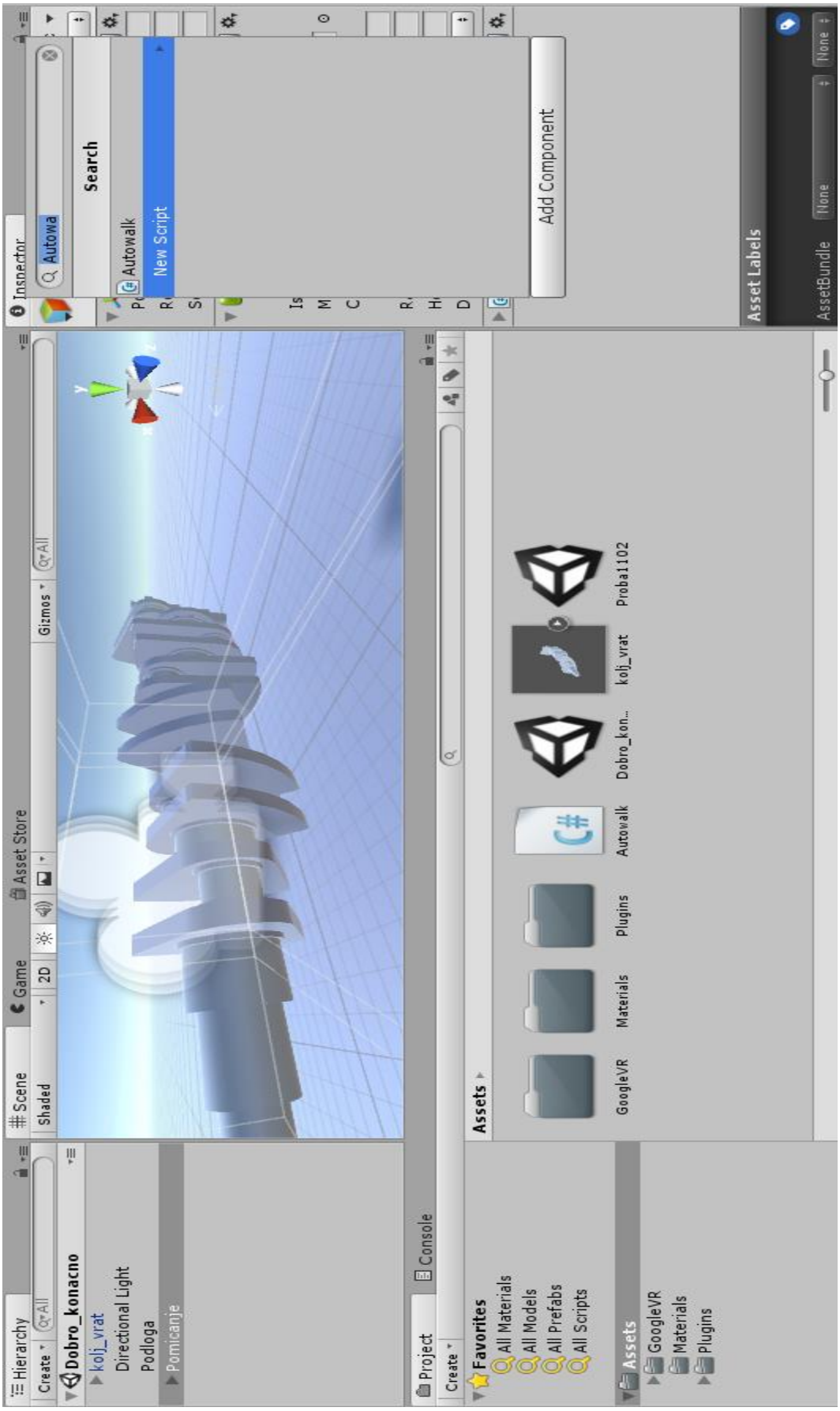
```
using UnityEngine;
using System.Collections;

public class Autowalk : MonoBehaviour
{
    private const int RIGHT_ANGLE = 90;

    // Varijabla određuje hoće li se lik micati ili ne
    private bool isWalking = false;

    Transform mainCamera = null;

    //Varijabla koja određuje brzinu lika
    [Tooltip("Ovom brzinom će se lik micati.")]
    public float speed;

    [Tooltip("Aktivirajte ovu opciju ako se lik treba micati kada je
pritisnuta tipka Cardboarda.")]
    public bool walkWhenTriggered;

    [Tooltip("Aktivirajte ovu opciju ako se lik treba micati kada je kut
ishod granice preliva (trehshold).")]
    public bool walkWhenLookDown;

    [Tooltip("Kut između 0° i 90°")]
    public double thresholdAngle;

    [Tooltip("Aktivirajte ovu opcije ukoliko želite zamrznuti y os." +
             "Primjerice ukoliko podlozi nisu dodijeljena svojstva
Collidera")]
```

```
    public bool freezeYPosition;

    [Tooltip("Fiksirana y koordinatna os.")]
    public float yOffset;

    void Start()
    {
        mainCamera = Camera.main.transform;
    }

    void Update()
    {
        // Kretanje kada je pritisnut gumb na Cardboardu
        if (walkWhenTriggered && !walkWhenLookDown && !isWalking &&
GvrViewer.Instance.Triggered)
        {
            isWalking = true;
        }
        else if (walkWhenTriggered && !walkWhenLookDown && isWalking &&
GvrViewer.Instance.Triggered)
        {
            isWalking = false;
        }

        // Kretanje kada korisnik gleda ispod kuta thresholda
        if (walkWhenLookDown && !walkWhenTriggered && !isWalking &&
            mainCamera.transform.eulerAngles.x >= thresholdAngle &&
            mainCamera.transform.eulerAngles.x <= RIGHT_ANGLE)
        {
            isWalking = true;
        }
        else if (walkWhenLookDown && !walkWhenTriggered && isWalking &&
                 (mainCamera.transform.eulerAngles.x <= thresholdAngle ||
                 mainCamera.transform.eulerAngles.x >= RIGHT_ANGLE))
        {
            isWalking = false;
        }

        // Kretanje kada je pritisnut gumb Cardboarda i kada korisnik gleda
ispod kuta thresholda
        if (walkWhenLookDown && walkWhenTriggered && !isWalking &&
            mainCamera.transform.eulerAngles.x >= thresholdAngle &&
            GvrViewer.Instance.Triggered &&
            mainCamera.transform.eulerAngles.x <= RIGHT_ANGLE)
```

```
        {
            isWalking = true;
        }
        else if (walkWhenLookDown && walkWhenTriggered && isWalking &&
                 mainCamera.transform.eulerAngles.x >= thresholdAngle &&
                 (GvrViewer.Instance.Triggered ||
                 mainCamera.transform.eulerAngles.x >= RIGHT_ANGLE))
        {
            isWalking = false;
        }

        if (isWalking)
        {
            Vector3 direction = new Vector3(mainCamera.transform.forward.x,
0, mainCamera.transform.forward.z).normalized * speed * Time.deltaTime;
            Quaternion rotation = Quaternion.Euler(new Vector3(0, -
transform.rotation.eulerAngles.y, 0));
            transform.Translate(rotation * direction);
        }

        if (freezeYPosition)
        {
            transform.position = new Vector3(transform.position.x, yOffset,
transform.position.z);
        }
    }
}
```

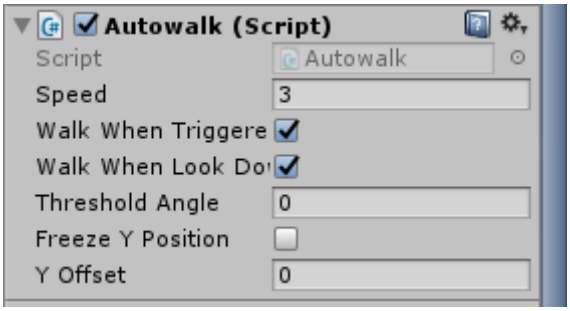
Izlist 1. Skripta Autowalk [33]

Spremanjem skripte u Visual Basicu i vraćanjem u Unity potrebno je objektu *Pomicanje* dodijeliti napravljenu skriptu. Na taj način, *Pomicanje* će poprimiti rutine koje su sadržane u programskom kodu, što će omogućiti kretanje lika pomoću prije spomenutih mehanizama. Zapravo, ono što će se kretati je kamera, koja je *child* komponenta stvorenog objekta *Pomicanje*. Klikom na *Pomicanje* u *Hierarchy* sekciji, u *Inspector* izborniku odabire se *Add Component* opcija. Postupak je sličan kao i dodavanje *Collidera*, samo što će se ovdje drugom objektu dodati drugačiji set funkcija, sadržan u skripti *Autowalk*. Klikom na *Add Component* potrebno je dodati napisanu skriptu. Najlakši način za to učiniti je preko tražilice, gdje će se kod upisivanja naziva skripte ona automatski pojaviti u padajućem izborniku (Slika 31.), zbog toga što Visual Studio automatski sprema napravljenu skriptu u Unityjev odredišni direktorij.



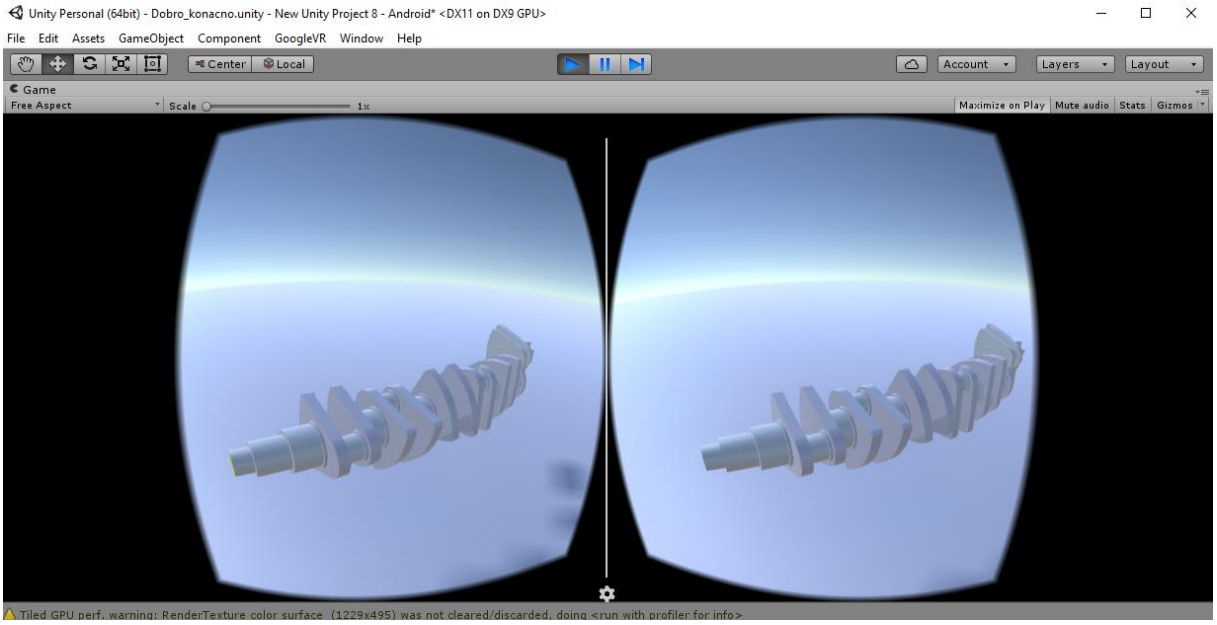
Slika 31. Dodavanje skripte objektu *Pomicanje*

Nakon što je dodana skripta, u Inspector izborniku korisnik može odabrati željeni mehanizam kretanja. U ovom slučaju, odabrane su opcije *Walk When Triggered i Walk When Look Down* kako bi se kretanje omogućilo putem oba opisana mehanizma zajedno. Korisnik može proizvoljno odabrati brzinu kretanja, granični kut i odmak od osi *Y* (Slika 32.).

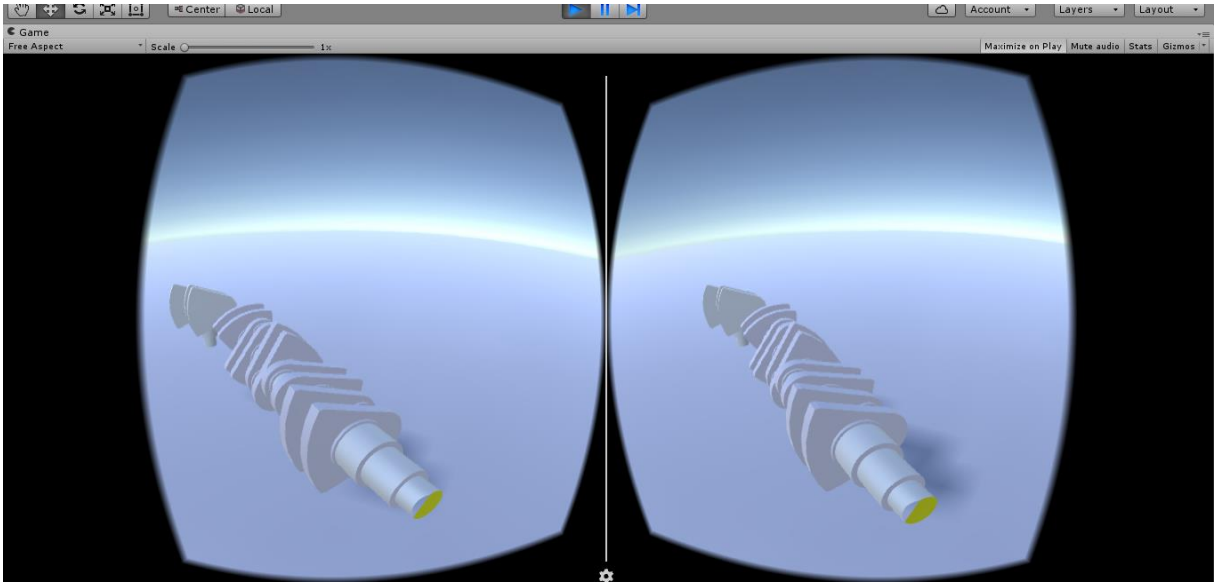


Slika 32. Odabir mehanizma kretanja

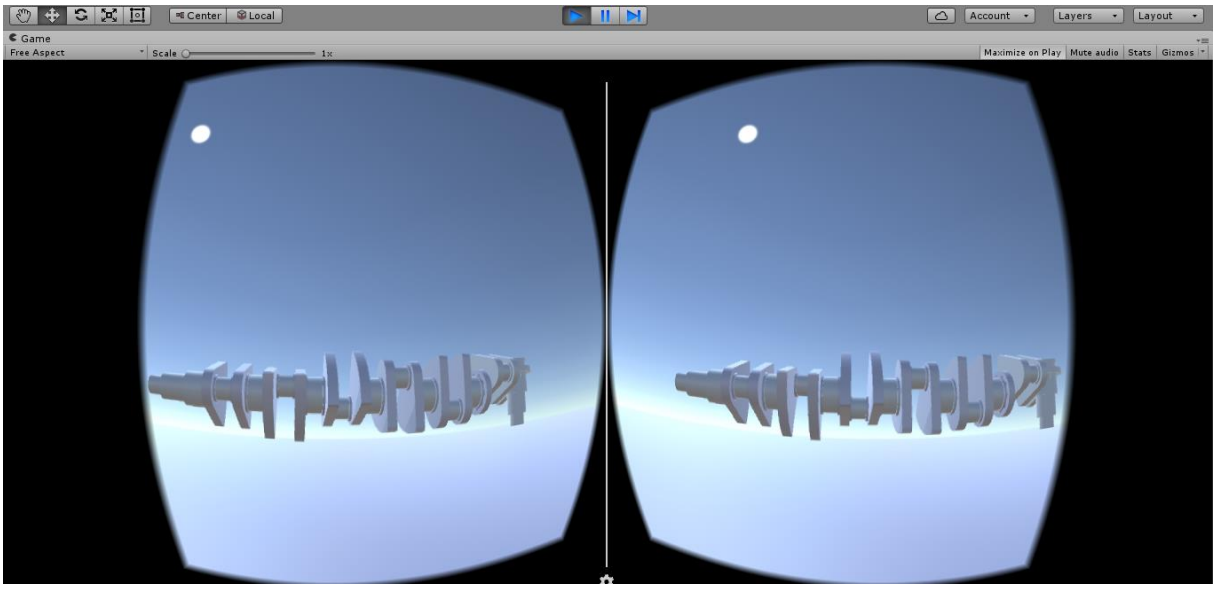
Time su definiranje sve rutine koje će omogućiti realniji dojam uključenosti u virtualni prostor. Klikom na tipku *Play* dobiva se konačni *Game* prikaz koji simulira *Cardboard 3D* prikaz. Budući da je ovo još uvijek prikaz na računalnom monitoru, približavanje simulaciji *Cardboard*a može se dočarati pritiskom na tipku Alt ili Ctrl te istodobnom rotacijom miša. Na taj način simuliraju se kretnje korisnikove glave koje u ovom trenutku predstavljaju rotaciju u virtualnom okruženju. Lijevim klikom miša simulira se tipka *Cardboard*a, što u ovome slučaju znači da će se korisnik kretati po virtualnom okruženju, zbog definirane rutine u *Autowalk* skripti. Više različitih prikaza koljenastog vratila moguće je vidjeti na slikama 33. do 35.



Slika 33. Simulacija Cardboard 3D prikaza 1

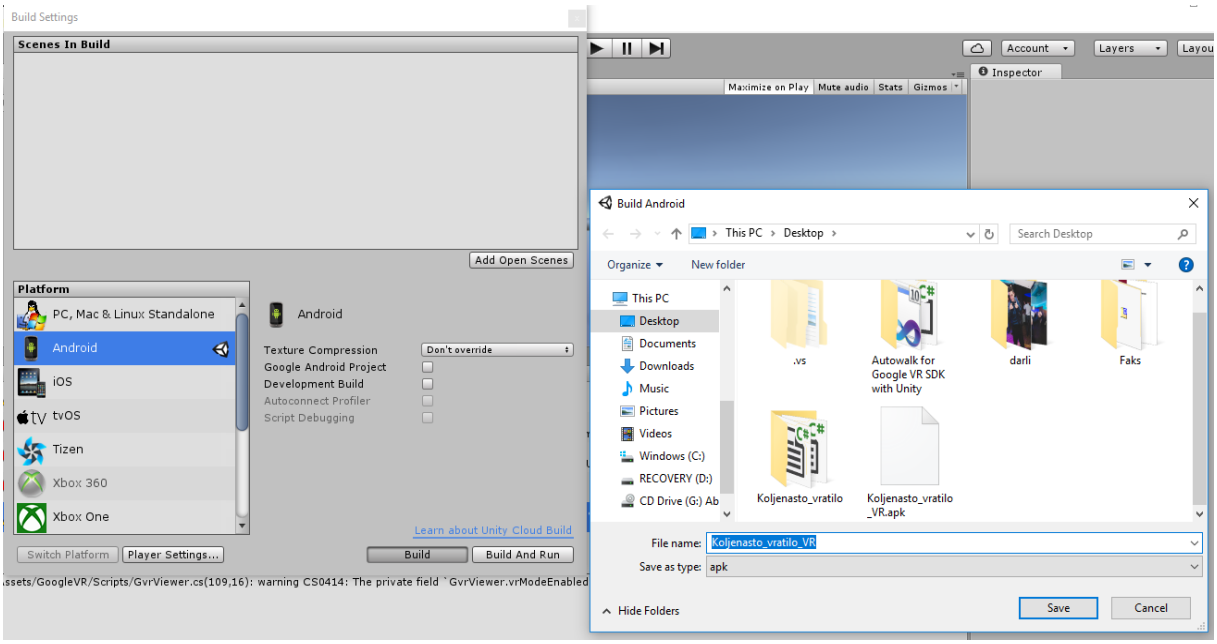


Slika 34. Simulacija Cardboard 3D prikaza 2



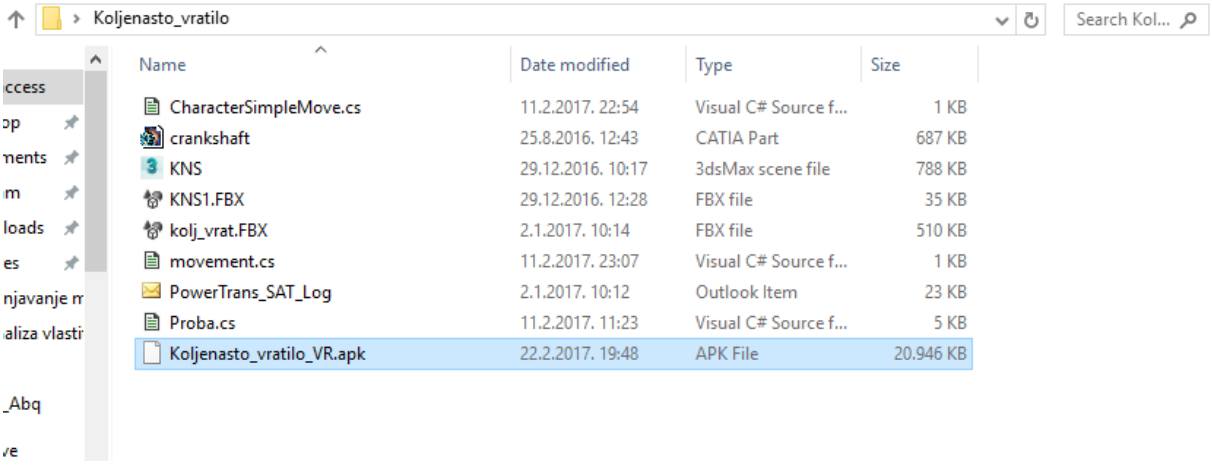
Slika 35. Simulacija Cardboard 3D prikaza 3

Da bi se napravljeno okruženje moglo prikazati na mobilnom sustavu Android, a kasnije i na Google Cardboardu, potrebno je iz generiranog okruženja napraviti aplikaciju preko tzv. *Build* opcije u Unityju. Klikom na *File – Build & Run* otvara se novi prozor u kojem je potrebno u donjem desnom kutu pritisnuti tipku *Build* te odabrati odredišni direktorij u kojem će Unity spremiti napravljenu aplikaciju (Slika 36). Ukoliko dođe do grešaka kod generiranja aplikacije, potrebno je instalirati najnoviju verziju *Java Development Kit*a.



Slika 36. Odabir odredišnog direktorija

Nakon određenog vremena aplikacija će biti spremljena u odredišni direktorij (Slika 37.), što će značiti kako je njeno korištenje moguće putem prebacivanja na mobilni telefon i učitavanja u aplikaciji *Cardboard* na Android telefonu. Kako bi aplikacija radila na korisnikovom mobilnom telefonu, potrebno je da je najniža instalirana inačica sustava API 19 – Android 4.4 'Kit Kat'.



Slika 37. Izrađena aplikacija

Nakon što aplikacija napravljena, mogu se sagledati sljedeće prednosti i nedostaci.

Prednosti:

- Učitavanje jednostavne CAD geometrije u virtualno okruženje omogućuje **3D imerzivno iskustvo**, koje rezultira uključenošću korisnika u virtualno okruženje.
- **Lakša vizualizacija i stvaranje dojma realnosti.**
- **Mogućnost skaliranja predmeta**, gdje se predmeti glomaznih dimenzija mogu smanjiti što rezultira lakšom predodžbom i manipulacijom dijelovima.
- Uključivanjem sustava za praćenje pokreta u virtualno okruženje dolazi se do **lakše interakcije s elementima okruženja**.

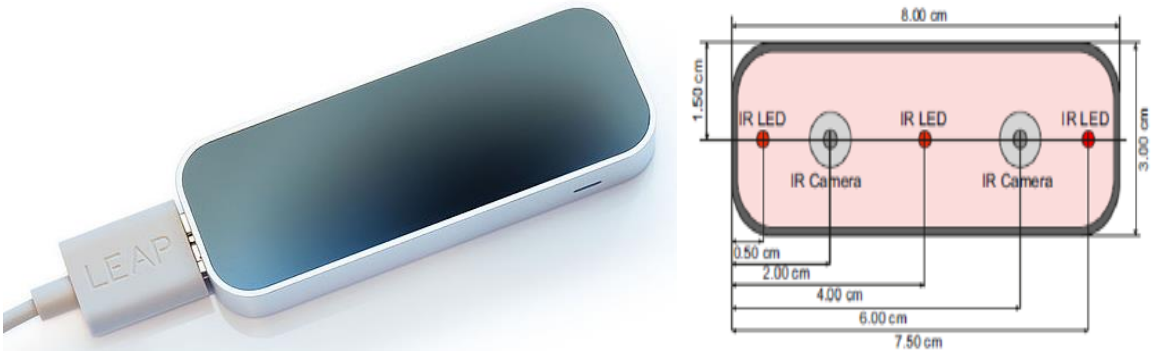
Nedostaci:

- Potrebno je napraviti **mnogo posla** kako bi se geometrija uopće mogla prikazati na platformi koja omogućuje stvaranje aplikacije za uređaje virtualne stvarnosti. **S povećanjem broja dijelova** i definiranjem međusobnih veza, **postupak postaje još dugotrajniji i složeniji**.
- Da bi se omogućila konkretnija interakcija s geometrijom, **potrebno je izvršiti napredno programiranje**, tzv. *Scripting*.
- Unity vrlo često doživljava nove verzije pa pojedine skripte **nisu kompatibilne** s novijim verzijama softvera što može **rezultirati raznim pogreškama.**
- Moguće su **geometrijske nedosljednosti ili nedostaci** pojedinih dijelova geometrije kod prikaza na uređaju virtualne stvarnosti, zbog postupka konverzije formata.
- Kao i kod većine drugih uređaja virtualne stvarnosti, **korisnik može osjetiti laganu mučninu i nelagodu kod korištenja uređaja**.

Iz svega navedenog vidljivo je kako su **u ovome trenutku nedostaci primjene ove tehnologije čak blago premašuju prednosti** koje su iznesene. Međutim, razvoj u industriji virtualne stvarnosti sigurno će s vremenom poboljšati trenutno stanje i nedostatke kod interakcije između CAD-a i sustava VR. Činjenica koja bi tu tezu mogla dokazati je ta da će SolidWorks 2017 imati uključen modul VR koji će omogućiti spajanje uređaja virtualne stvarnosti i CAD sučelja. Za sada nije poznato na koji način će modul VR funkcionirati, te hoće li dovesti do određenih noviteta u modeliranju surovom CAD geometrijom, ili će pak, s druge strane, omogućiti lakšu vizualizaciju i prikaz određenih simulacija koje su moguće u novom izdanju SolidWorksa.

Ono što bi u ovome trenutku svakako predstavljalo napredak kod povezivanja CAD geometrije i sustava VR su sustavi za praćenje pokreta korisnika. Primjerice, korištenjem sustava za praćenje pokreta ruku (eng. *hand tracking*), omogućila bi se interakcija dijelovima u trenutku kada korisnik napravi određenu gestu sa svojom rukom ili prstima. Na taj način, ne bi bilo potrebe koristiti računalnu periferiju, te bi se olakšala interakcija. Najveći iskorak u sustavima za praćenje pokreta ruku napravila je korporacija Leap Motion, Inc. iz SAD-a.

Korporacija je 2016. godine napravila softver Orion koji služi za praćenje pokreta ruku u virtualnoj stvarnosti. Softver radi uz kontroler Leap Motion (Slika 38.) koji se s računalom spaja preko USB veze.



Slika 38. Kontroler Leap Motion [34 i 35]

Kod korištenja sa sustavima virtualne stvarnosti, spomenuti kontroler se montira direktno na *headset* (Slika 39.). Za rad koristi dvije monokromatske infracrvene kamere i tri infracrvene LED žaruljice. Analizirani podaci se šalju preko USB veze na računalo, gdje u softveru pomoću kompleksne matematike dolazi do prepoznavanja pokreta i definiranja određenih funkcija. [34]

Slika 39. Kontroler Leap Motion montiran na OSVR headset [34]

3.5. RAZVOJ APLIKACIJE VIRTUALNE STVARNOSTI ZA PLATFORMU OCULUS

Zbog nedostataka koje pruža prikaz sadržaja na platformi Google Cardboard, u nastavku će biti prikazan razvoj aplikacije za Oculus DK2[17]. Osnovni principi razvoja aplikacije neće se mnogo razlikovati u odnosu na platformu *Cardboard*, jer će se i dalje koristiti svi do sada opisani programski paketi. Razlika će biti u tome da razvojna mobilna platforma više neće biti sustav Android, već će se aplikacija razvijati za platformu Windows s podrškom za sustave virtualne stvarnosti, konkretno za Oculus. Oculus pruža brojne prednosti prvenstveno u aspektu kvalitete i realnosti prikaza sustava virtualne stvarnosti u odnosu na mobilnu platformu. Dakako, prednost je i ta što Oculus direktno koristi računalne resurse za prikaz sadržaja, za razliku od *Cardboard*a gdje se koriste resursi pametnog telefona, koji po performansama značajno odstupaju od računalnih komponenti. Za lakšu interakciju i intuitivniji pristup aplikaciji, u razvoju će se koristiti i već spomenuti kontroler Leap Motion. Korištenje kontrolera Leap Motion omogućit će intuitivniju interakciju jer za manipulaciju dijelovima neće biti potrebno koristiti periferijalne uređaje poput tipkovnice i miša, već će se radnje koje uključuju manipulaciju dijelovima moći izvršiti samo s rukama, bez fizičkog pritiskanja na bilo koju od tipku kao što je to slučaj kod miša ili tipkovnice.

3.5.1. Oculus DK2

Oculus DK2 (Slika 40.) ima rezoluciju 960x1080 po oku, *refresh rate* od 75 Hz te širinu kuta gledanja[18] od 100 ° za razliku od potrošačke verzije Oculus CV1. Potrošačka verzija ima kvalitetniji zaslon koji se očituje u većoj rezoluciji od 1080x1200 po oku, većoj frekvenciji osvježivanja (90 Hz), većem kutu gledanja (110 °) i integriranim 3D slušalicama.

[17] verzija Development Kit 2

[18] engl. FOV – *field of view*

Slika 40. Oculus DK2 [5]

Principi spajanja računala i Oculusa su vrlo slični bez obzira o kojoj se verziji Oculus Rifta radi. Za spajanje uređaja Oculus DK2 na računalo koristi se HDMI i USB *port*. HDMI *port* služi za prijenos slike i zvuka, dok se USB sučelje koristi za napajanje Oculusa. Nadalje, uz Oculus DK2 dolazi i kamera za praćenje položaja (Slika 41.), koja prepoznaje korisnikove kretnje, preko identifikacije IR[19] točaka koje se nalaze na površini uređaja. Kamera za praćenje pozicije također koristi USB priključak za napajanje, ali i 2.5“ konektor za sinkronizaciju podataka između kamere i uređaja.

Slika 41. Kamera za praćenje položaja

[19] engl. *Infra Red* – infracrvene

Kako bi Oculus radio, potrebno je instalirati popratne programe koji služe za prepoznavanje uređaja. U prošlosti je bilo potrebno instalirati više različitih vrsta programa, dok je nedavno došlo do integracije *Oculus Runtimea* i drugih rutina u jedinstveni softver koji je dostupan na stranicama proizvođača. Budući da je Oculus DK2 uređaj namijenjen za razvojne programere, koji je izašao na tržište već prije tri godine, a u međuvremenu je došlo do proizvodnje i prodaje potrošačke verzije, Oculus navodi kako uređaj DK2 više nije podržan kao potrošački uređaj iako će podržavati rad aplikacija poput *Oculus Home*, *Oculus Store* i *Oculus app* za Windowse. Međutim, ako se korisnik odluči za kupnju pojedine aplikacije u trgovini *Oculus Store* može doći do toga da aplikacija ne radi pouzdano, ili ne radi uopće. Ta ograničenja u ovom slučaju neće predstavljati problem, jer je kod razvoja aplikacije u ovom radu bitna isključivo međusobna podrška softvera za razvoj aplikacije VR (Unity) i Oculusa DK2, što i dalje radi bez problema.

Kod skidanja instalacijskog programa sa službene Oculus stranice, te pokretanja instalacije preporučljivo je preskočiti određene korake u kojima se vrši prepoznavanje uređaja, baš zbog činjenice kako je dostupan softver prvenstveno namijenjen za CV1 potrošačku verziju *Rifta*, koji posjeduje taktilne senzore koji nisu uključeni u prijašnje verzije Oculusa. Preskakanje pojedinih koraka neće rezultirati greškama u radu jer će naknadno spajanje nakon što je završena instalacija rezultirati normalnim radom. Nadalje, da bi Oculus radio u optimalnom režimu rada, potrebno je posjedovati računalo naprednih specifikacija. Preporučene specifikacije za Oculus Rift jesu: procesor Intel i5-4590 ili ekvivalent, grafička kartica NVIDIA GTX 1060/AMD Radeon RX 480 ili bolja, 8+GB radne memorije te USB 3.0 ulazni *portovi*. Zbog starije DK2 verzije specifikacije računala mogu biti i nešto niže od navedenih, iako treba imati na umu činjenicu kako to može rezultirati niskokvalitetnim prikazom i povremenim zamrzavanjem slike što može dovesti do mučnine i nelagode kod prikaza virtualnog okruženja. [5,12, 36 i 37]

3.5.2. Leap Motion

Leap Motion, Inc. je američka kompanija koja proizvodi i stavlja na tržište senzorske uređaje koji omogućuju prepoznavanje pokreta ruku i prstiju, poput računalnog miša, ali bez potrebe za fizičkim kontaktom s perifernim uređajem. 2012. godine kompanija stavlja na tržište

uređaj nazvan *The Leap*[20] za kojeg je koncem godine razvila softver za razvojne programere. Do ožujka 2014. okvirno je prodano više od pola milijuna uređaja što je bilo znatno manje od početnih očekivanja. U kolovozu 2014. kompanija razvija softver za praćenje pokreta na zaslonima montiranim na glavu, poput primjerice Oculus Rifta (Slika 42.). U novije doba, kompanija se fokusirala na virtualnu stvarnost pa je tako početkom 2016. godine razvijen softver Orion, koji je posebno dizajniran za praćenja pokreta ruku u aplikacijama virtualne stvarnosti.

Slika 42. Zajedničko korištenje kontrolera Leap Motion i Oculusa [38]

Kontroler Leap Motion obuhvaća polukružno[21] radno područje (Slika 43.) ispred uređaja na udaljenosti od otprilike 1 metar. U istraživanju provedenom 2013. godine ispostavilo se da je ukupna prosječna točnost očitavanja pokreta 0,7 milimetara.

Slika 43. Ilustracija radnog pojasa kontrolera Leap Motion [38]

[20] U nastavku rada koristit će se naziv kontroler Leap Motion.

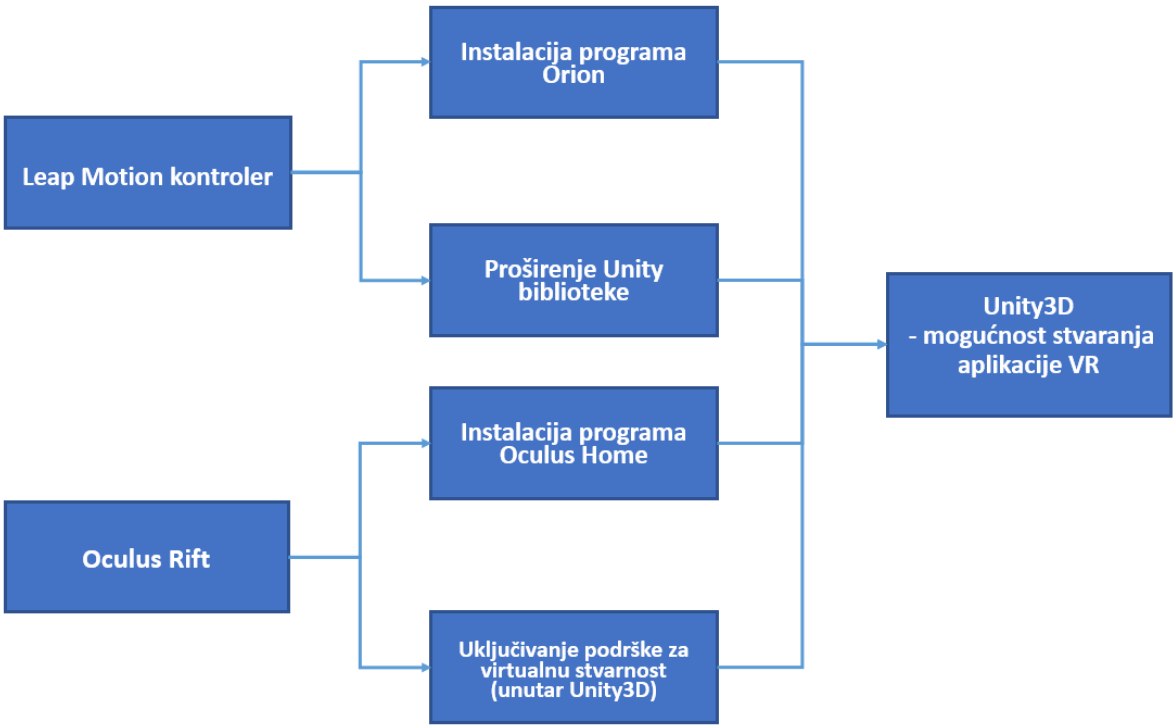
[21] Točnije: 150 °.

Većina korisnika uređaj koristi za precizno crtanje, manipuliranje kompleksnim 3D podacima ili pak upravljanje gestama na raznim mapama, web sjedištima ili drugim zabavnim aplikacijama koje su dostupne u sklopu trgovine Leap Motion App Home. Kontroler Leap Motion koristi se i kod istraživanja u medicini, automobilskoj i glazbenoj industriji. U ovome radu, uređaj će se koristiti za interakciju s dijelovima koji su nastali u CAD okruženju kako bi se povećala intuitivnost i realnost virtualnog okruženja, što će omogućiti značajnu prednost u interakciji zbog činjenice kako će se u potpunosti izbaciti potreba za korištenjem klasičnog miša i tipkovnice. Cijena kontrolera Leap Motion iznosi svega 80 USD što rezultira širokom dostupnošću ali i određenim nedostacima koji će biti izneseni u nastavku ovog rada.

Da bi kontroler Leap Motion bio prepoznat u Windowsima, potrebno je instalirati softver Orion, koji se trenutno nalazi u beta fazi. Za razliku od prijašnjih verzija softvera, Orion pruža veliki napredak i poboljšanje u tehnikama praćenja pokreta ruku i prstiju, uključujući poboljšano rukovanje kad je pojedina ruka nedovoljno izložena senzoru, ili pod velikim kutom koji nije u potpunosti prekriven senzorom. Robustan dizajn omogućuje velika poboljšanja kod interakcije u okruženjima koja se očituju prekomjernom svjetlošću ili velikim brojem komponenata. Važno je spomenuti kako uz Orion dolazi niz besplatnih primjera demo aplikacija koje korisniku omogućuju upoznavanje s opremom ali i pozadinom koja stoji iza nastale aplikacije. Dakako, u vrlo kratkom roku korištenja uređaja vidljivo je kako postoje određena ograničenja kod rukovanja dijelovima, a koja se jednostavno ne mogu zaobići zbog samog načina na koji je kontroler konstruiran. [34 i 39]

3.5.3. Povezivanje Unityja, Oculusa i kontrolera Leap Motion

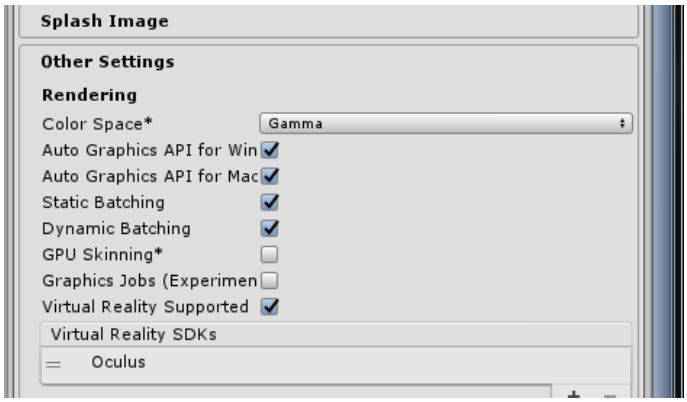
Da bi došlo do stvaranja imerzivnog okruženja, u Unityju je potrebno uključiti rutine koje će omogućiti međusobno prepoznavanje i povezivanje između Unityja, platforme virtualne stvarnosti te kontrolera Leap Motion (Slika 44.).



Slika 44. Međusobno povezivanje komponenti sustava za stvaranje imerzivnog okruženja

Programi koji omogućuju korištenje kontrolera Leap Motion (Orion) i Oculus Rifta (Oculus Home) trebaju biti valjano instalirani na računalo, jer se u protivnom neće moći razviti aplikacija virtualne stvarnosti.

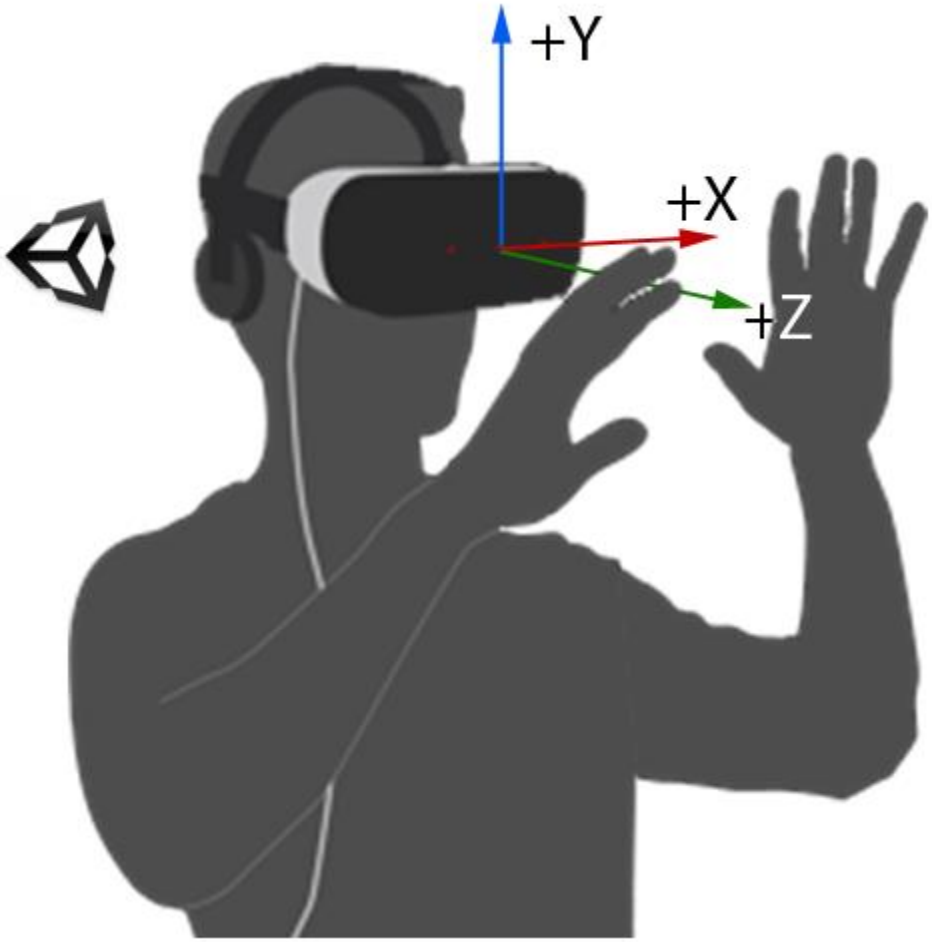
U nastavku slijedi prikaz radnji koje je potrebno učiniti unutar Unityja za razvoj imerzivne aplikacije. U početku je potrebno definirati razvojnu platformu, koja će u ovom slučaju biti Windows, uz podršku za virtualnu stvarnost. Postupak odabira analogan je postupku odabira platforme Google Cardboard u točki 3.3.1. Proširenje biblioteke i definiranje razvojne platforme, uz razliku što u ovom slučaju nije potrebno mijenjati razvojnu platformu, već je potrebno omogućiti stvaranje virtualnog okruženja na platformi Oculus (Slika 45.).



Slika 45. Uključivanje podrške za virtualnu stvarnosti

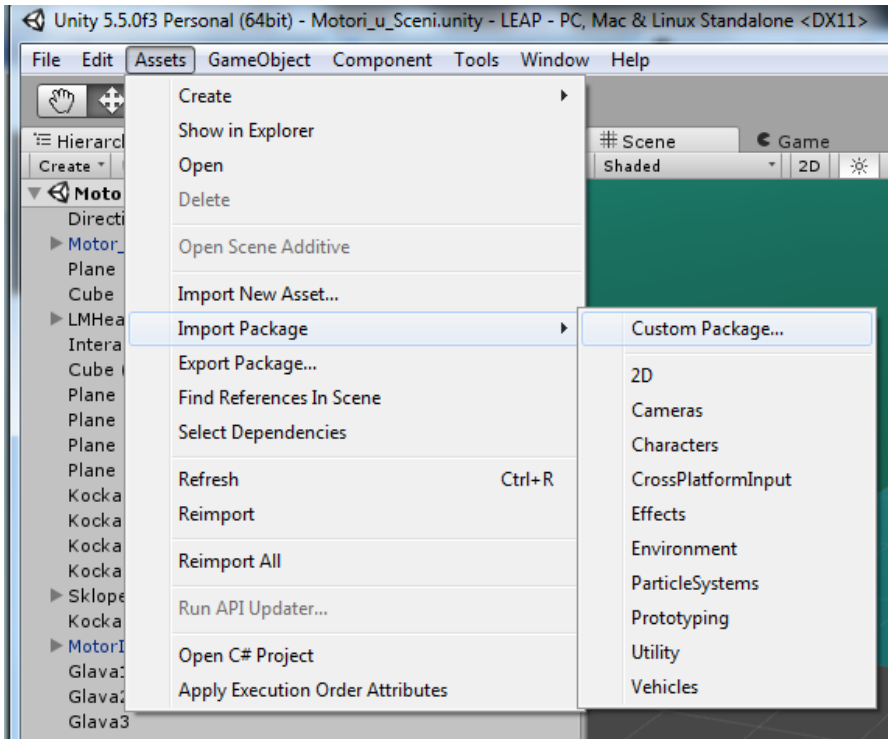
Unity VR omogućuje pokretanje VR uređaja bez potrebe za korištenjem vanjskih proširenja. Ako je u ovome trenutku Oculus spojen na računalo, klikom na gumb *Play* i zaokretanjem Oculusa doći će do zaokretanja prikaza tj. kamere u generiranom Unity okruženju. Time je definirana rutina koja će u svakom trenutku kada korisnik pritisne na tipku *Play* omogućiti generiranje virtualnog okruženja koje će istovremeno biti prikazano i na Oculusu i na računalnom monitoru. To je moguće zbog rutine koja definira automatski prikaz sadržaja na zaslonu virtualne stvarnosti a koja je integrirana unutar Unityja. Praćenje pokreta glave i definiranje veličine kuta gledanja također je automatski sadržano u ovoj rutini. Osim računalne konfiguracije koja mora zadovoljiti preporučene zahtjeve, važno je da je instaliran i tzv. *runtime* uređaja virtualne stvarnosti, Oculus Home, prema uputama u točki 3.5.1.

Korištenje kontrolera Leap Motion u Unityju omogućuje praćenje položaja ruku i prstiju te bilježenje pozicija, brzina i orijentacije uz ključ da se navedeno odvija s dobrom točnošću i niskom latencijom kako bi prikaz i interakcija s dijelovima u okruženju bili što realniji. Programsko sučelje kontrolera Leap Motion kao osnovnu jedinicu duljine koristi milimetre, dok je koordinatni sustav definiran preko pravila desne ruke (Slika 46.). S druge strane, Unity za definiranje koordinatnog sustava koristi pravilo lijeve ruke a za osnovnu jedinicu duljine koriste se metri. Navedeni problem rješava se proširivanjem biblioteke i pravovaljanom orijentacijom kontrolera Leap Motion na zaslonu virtualne stvarnosti.



Slika 46. Prikaz koordinatnog sustava kontrolera Leap Motion [38]

Proširena biblioteka sadrži skripte koje definiraju zakonitosti i izmjenu informacija između Unityja i kontrolera Leap Motion. To će konkretno u ovom slučaju omogućiti unutarnju transformaciju koordinatnog sustava kontrolera Leap Motion u lijevokretni koordinatni sustav, te automatsku promjenu jedinica u metre. Nadalje, dodatna proširenja grafičke biblioteke uz korištenje posebnih skripti omogućit će uključivanje i 3D prepoznavanje korisnikovih ruku na sceni. Navedena *package* datoteka koja služi za proširivanje biblioteke nosi naziv Unity Core Assets, te je dostupna na službenim stranicama kompanije Leap Motion. Nakon preuzimanja datoteke potrebno je uvesti *package* u Unity preko izbornika prikazanog na slici 47.



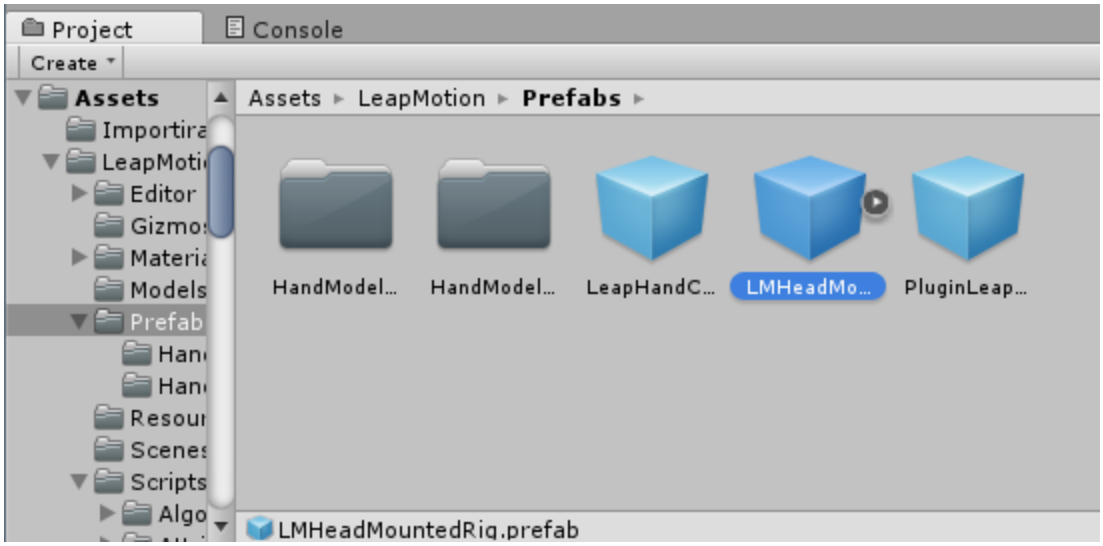
Slika 47. Proširivanje biblioteke Unity

Lociranjem paketa na disku i klikom na *import* pojavit će se prozor kao na slici 48.

Slika 48. Paket za proširivanje biblioteke

Nakon klika na *Import*, u *Project* izborniku u datoteci *Assets* pojavit će se *LeapMotion* datoteka koja sadrži rutine[22] koje omogućuju prepoznavanje kontrolera Leap Motion u Unity okruženju.

Unutar *LeapMotion* datoteke nalazi se datoteka *Prefabs* u kojoj je sadržan *LMHeadMountedRig.prefab* (Slika 49.). Pojam *prefab* predstavlja unaprijed definiran objekt kojeg krase određene komponente i svojstva pridijeljena tim komponentama. Premještanje navedenog *prefaba* u scenu omogućit će prepoznavanje pokreta ruku i prstiju u specifičnom položaju, tj. u trenutku kada je kontroler Leap Motion montiran na zaslon virtualne stvarnosti. To je moguće jer se u *prefabu LMHeadMountedRig* nalaze rutine koje između ostalog omogućuju prepoznavanje svake ruke pojedinačno, ali i neke dodatne opcije poput prepoznavanja zvuka.



Slika 49. LMHeadMountedRig prefab

Kod generiranja okruženja, potrebno je pripaziti na veličinu navedenog *prefab*a. Ako je *prefab* premalen u odnosu na dio kojim se želi manipulirati, potrebno je napraviti skaliranje, odnosno promijeniti odnose veličina (najčešće povećati *prefab*) kako bi se dobilo na realnosti.

[22] Tzv. *prefab*ovi, skripte i scene.

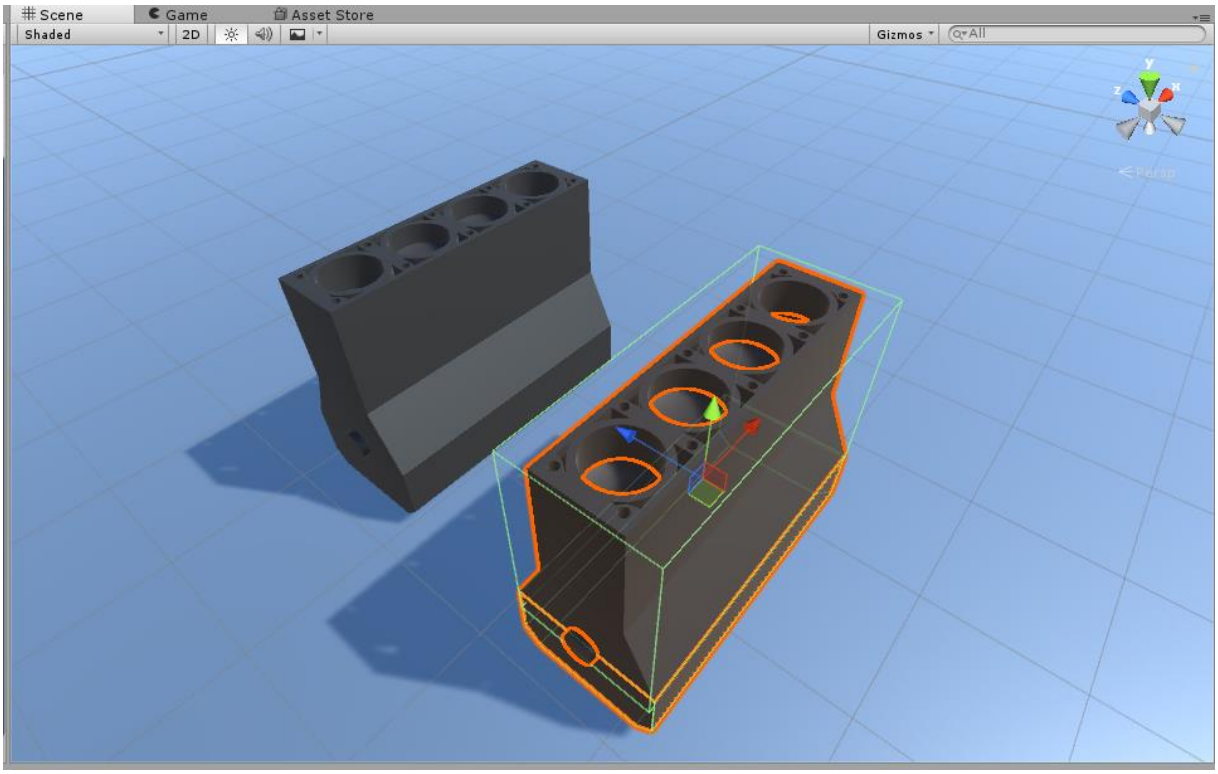
Kao što je već spomenuto, uključivanje navedenog *prefab*a rezultirat će prikazom korisnikovih ruku u virtualnom okruženju, međutim to neće automatski značiti kako je moguće vršiti interakciju s dijelovima koji se nalaze unutar virtualnog okruženja. Svakom dijelu, bilo da je generiran u CAD softveru ili direktno u Unityju[23] potrebno je pridijeliti svojstva koja će definirati njegovo ponašanje u okolini, ali i u trenutku interakcije sa korisnikovim rukama preko kontrolera Leap Motion. Kada dijelovi ne bi bili tako definirani, ili bi propadali u *Game* prikazu (primjer Google Cardboarda) ili bi u slučaju dodira dolazilo do prolaska prstiju kroz dio. Zato je potrebno definirati fiziku objekata.

Dakle, da bi interakcija korisnika u virtualnom okruženju djelovala realistično, potrebno je pridijeliti određena svojstva i primijeniti određene zakonitosti na objekte stvorene u virtualnom okruženju. Kako bi objekti prikazivali uvjerljivo fizičko ponašanje, potrebno je definirati odgovarajuće rutine koje će omogućiti njihovo ponašanje u skladu s fizičkim zakonitostima. Primjerice, primjenom akceleracije, gravitacije ili drugih sila, objekt bi se trebao ponašati u skladu sa zakonitostima mehanike. Kod radnji poput hvatanja, pozicioniranja ili međusobnog sudaranja dijelovima vrlo je važno svojstvo kolizije, koje će pravovremeno detektirati međusobnu interakciju komponenata i na taj način rezultirati realističnim prikazom odnosa između komponenata. Zbog oponašanja fizičkog svijeta, bitno je da se kolizija odvija pravovremeno i točno, kako prikaz ne bi odstupao od realnosti. Unity ima ugrađen sustav za fiziku koji omogućuju relativno jednostavno definiranje fizičkih zakonitosti između pojedinih dijelova u stvorenom okruženju. Jedno od najčešće korištenih svojstava je svojstvo krutog tijela, tzv. *Rigidbody*. Pridjeljivanjem komponente *Rigidbody* objektu, ono će automatski reagirati na gravitaciju. Da bi se omogućilo međusobno sudaranje dijelova, potrebno je definirati svojstva kolizije preko opcije *Collider*. Svojstvo *Collider* funkcionira na principu stvaranja nevidljive mreže oko objekta koja omogućuje fizičko sudaranje s drugim dijelova u okruženju koji također imaju definirano to isto svojstvo. Mreža ne mora biti iste konture kao i objekt definiran u okruženju, ali to može rezultirati nerealnim prikazom i nerealnom kolizijom između dijelova, zbog preklapanja površina. Unity koristi više vrsta takozvanih primitivnih *Collider*a koji zahtijevaju najmanje računalnih resursa. Takvi *Collider*i mogu biti pravokutnog, sfernog ili cilindričnog oblika. Geometrija

[23] Unity omogućuje generiranje samo primitivne 3D geometrije, za razliku od CAD-a.

pojedinih sklopova stvorenih u CAD sustavu gotovo uvijek je znatno kompleksnija od primitivnih oblika *Collider*a koje pruža Unity, stoga je potrebno svojstva *Collider* definirati na drugačiji način, preko opcije *Mesh Collider*, ukoliko se želi po cijelom volumenu dijela pridijeliti svojstvo *Collider*.

Kao što i samo ime govori, *Mesh Collider* stvara mrežu koja se u potpunosti poklapa s geometrijom dijela. Čim je dio kompleksnije geometrije, mreža oko dijela poprima kompleksniji oblik što u konačnici rezultira znatno većim utroškom računalnih resursa u odnosu na primitivne *Collider*e. Ukoliko se u Unity ubaci sklop koji sadrži mnoštvo pojedinačnih kompleksnih dijelova njegovo svojstvo *Collider*a moglo bi se definirati preko nekih od prije spomenutih primitivnih *Collider*a koji zahtijevaju znatno manje računalnih resursa. U tom slučaju mreža koju stvara softver neće u potpunosti obuhvaćati cjelokupnu geometriju sklopa, ili će obuhvaćati površinu veću od sklopa, što će rezultirati nerealnim prikazom naročito kod radnji poput hvatanja dijelova ili međusobnog sudaranja. Točka dodira nalaziti će na rubu mreže, bez obzira je li mreža premalena pa se nalazi unutar dijela, ili se nalazi na nekoj vanjskoj točki geometrije dijela. Kod desnog dijela (Slika 50.), svojstvo *Collider*a definirano je preko opcije primitivnog oblika *Box Collider*. Zelenim rubom vidljiva je mreža koju obuhvaća *Box Collider*. Mreža se može i smanjiti, međutim zbog geometrijskog ograničenja *Box Collider*a nemoguće je u potpunosti obuhvatiti geometriju dijela, jer je dio apriori kompleksnijeg geometrijskog oblika u odnosu na opciju *Box Collider*. Želi li korisnik uhvatiti dio, do kolizije između prstiju i dijela doći će na zelenom rubu, a ne na rubu geometrije dijela, što će rezultirati nerealnim prikazom hvatanja. Isto će vrijediti i kod eventualnog sudara između dijelova. Detekcija kolizije izvršit će se za na zelenom rubu, što će rezultirati lošim i nerealnim prikazom gdje se dijelovi sudaraju zapravo bez kontakta, u praznom prostoru, na rubu koji je loše definiran.



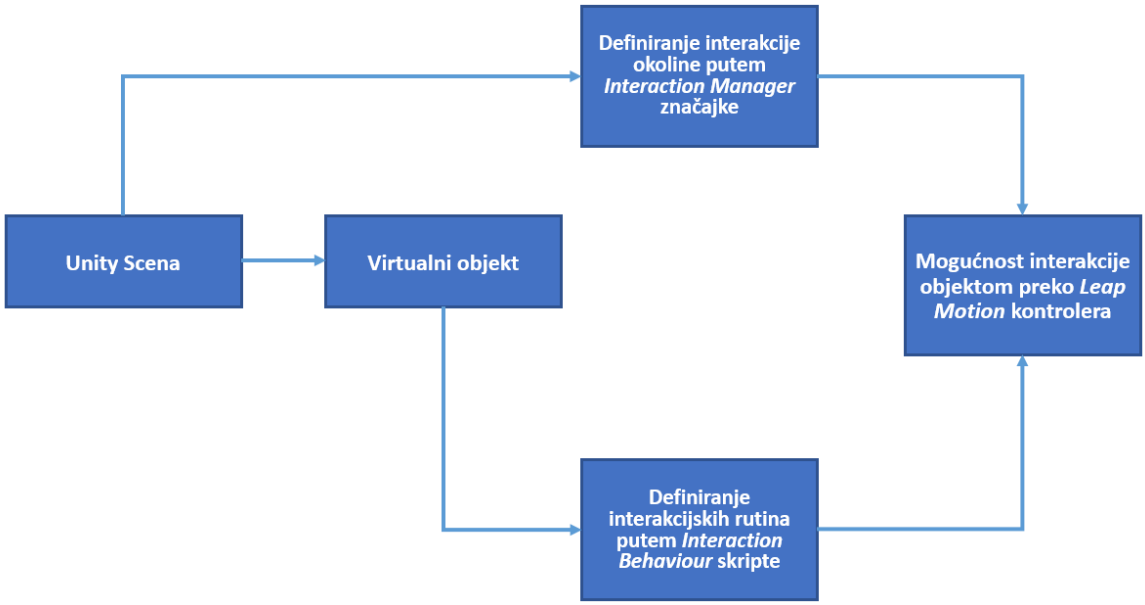
Slika 50. Prikaz Mesh i Box Collidera

Zbog iznesene problematike, sklopovlje i kompleksnije oblike moguće je definirati preko spomenute opcije *Mesh Collider*, kao što je slučaj s lijevim dijelom (Slika 50.), gdje se mreža *Collider*a u potpunosti poklapa sa rubnom geometrijom dijela. Definiranje *Collider*a preko opcije *Mesh Collider* kod slabijih računalnih konfiguracija može rezultirati zastajanjem prikaza kod pokušaja hvatanja pojedinačnih komponenti sklopa. Kod područja montaže dijelova, gdje se sklapaju primjerice dijelovi koji se teško dovede u međusoban odnos zbog specifičnih geometrijskih značajki, ponekad se koriste primitivni *Collider*i koji su definirani na udaljenim rubovima dijelova, kako bi do međusobne kolizije načelno došlo što teže, tek u trenutku kad se sudare prethodno definirani rubovi *Collider*a. Takav pristup korišten je u nastavku kod sklapanja zidne priključnice (točka 4.2.).

Za definiranje okruženja i zakonitosti između objekata vrlo je važno koristiti skriptiranje. Ponašanje objekata unutar okruženja definirano je komponentama koje su tim istim objektima pridijeljene. Osim prije spomenutih komponenti koje su definirale fizičko ponašanje dijela, objektima je moguće pridijeliti i skripte. Skripte se koriste za izradu grafičkih efekata, kontrolu fizičkog ponašanja objekata i definiranje bilo kojih drugih zakonitosti koje nisu standardno uključene unutar Unityja. Iako Unity posjeduje širok raspon

ugrađenih komponenata koje definiraju zakonitosti između dijelova, vrlo brzo se dolazi do ograničenja u realizaciji pojedinih naprednih funkcija te je zbog toga moguće stvoriti skripte prema vlastitoj želji u podržanim programskim jezicima poput C# ili UnityScript (JavaScript). U slučaju da se skripta ne pridijeli željenom objektu u okruženju, izvršavanje njenog koda neće biti aktivno.

Unity, ali i primjerice kontroler Leap Motion sadrže velik broj skripata koje su javno dostupne za korištenje i koje sadrže primjere za lakšu interakciju u virtualnom okruženju. Kada bi primjerice uz pomoć kontrolera Leap Motion željeli rukovati objektom u virtualnom okruženju kojemu nije pridijeljena skripta preko koje će taj objekt prepoznavati radnju hvatanja, korisnikovi prsti bi jednostavno samo prošli kroz navedeni objekt. U dokumentaciji kontrolera Leap Motion zbog toga su dostupne interakcijske skripte koje su sadržane unutar *package* datoteke Unity Core Assets koja služi za proširenje biblioteke Unityja. Skripte *Interaction Manager* i *Interaction Behaviour* nalaze se u *Asset* području u *Project* izborniku. Dodavanje *Interaction Manager*a na scenu omogućit će interakcijske rutine na objektima kojima se pridijeli *Interaction Behaviour* skripta. Navedene skripte služe za prepoznavanje specifičnih pokreta korisnikovih ruku i prstiju, što će značiti da će u trenu kada primjerice korisnik odluči uhvatiti objekt svojim prstima to biti i omogućeno. Shematski prikaz (Slika 51.) prikazuje koje je radnje potrebno izvršiti kako bi se u imerzivnom okruženju omogućila interakcija virtualnim objektima.

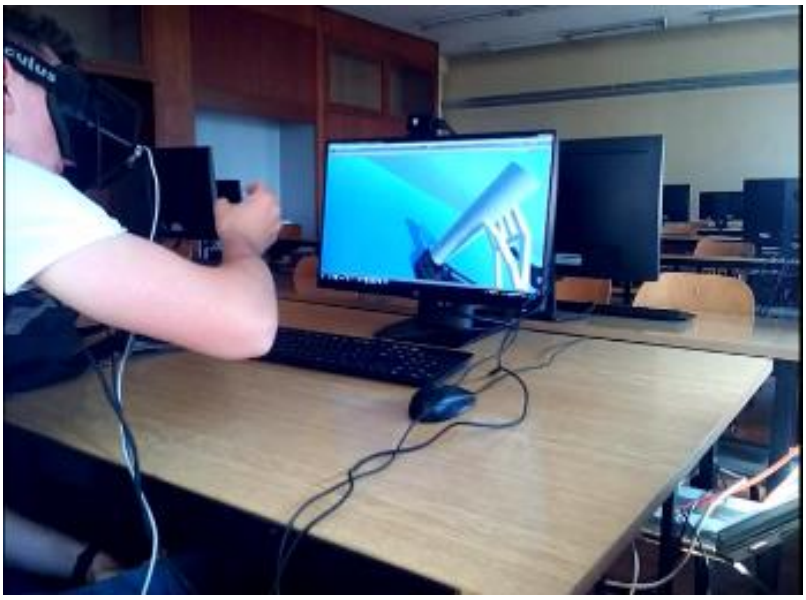


Slika 51. Omogućivanje interakcije u imerzivnom okruženju

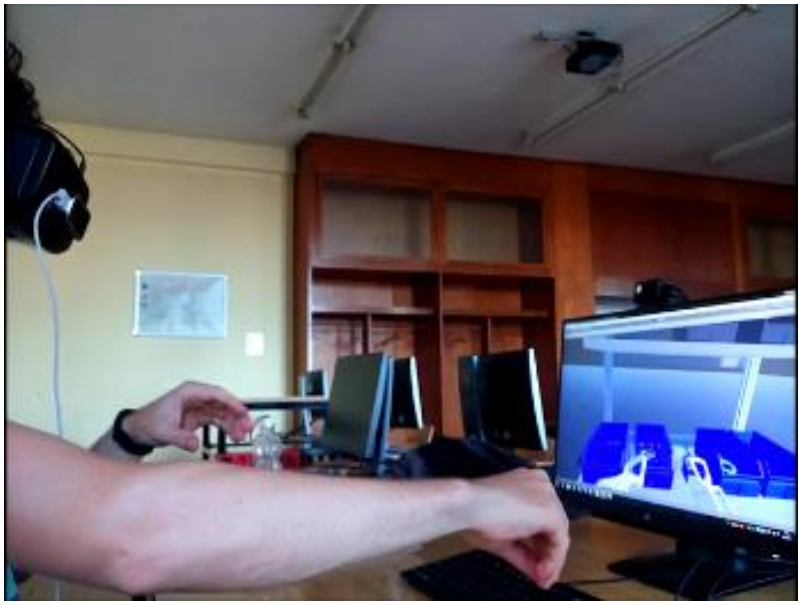
Prvo je na scenu potrebno dodati objekt *Interaction Manager* koji će zbog sadržanih komponenata i zakonitosti unutar objekta omogućiti definiranje interakcijske okoline. Nadalje, bilo kojem virtualnom objektu na kojem se želi vršiti interakcija potrebno je pridijeliti skriptu *Interaction Behaviour* koja će definirati ponašanje dijela u trenutku kada korisnik izvršava radnje poput hvatanja, pomicanja ili pozicioniranja dijelova, uz uvjet da je pridijeljen odgovarajući *prefab*. [38 i 40]

4. PROJEKTIRANJE PROIZVODNIH PROCESA TEHNIKAMA VIRTUALNE STVARNOSTI

Ovo poglavlje omogućit će širi prikaz trenutnih mogućnosti projektiranja proizvodnih procesa tehnikama virtualne stvarnosti na različitim primjerima. Prvi primjer koji slijedi u nastavku bavit će se problemom umetanja osovine u provrt (Slika 52.), dok će drugi primjer prikazati proces sklapanja zidne priključnice (Slika 53.).



Slika 52. Umetanje osovine u provrt



Slika 53. Sklapanje zidne priključnice

4.1. UMETANJE OSOVINE U PROVRT

Problem točnog umetanja osovine u provrt[24] jedan je od glavnih problema u procesima sklapanja dijelova, bez obzira radi li se o ručnoj ili automatskoj montaži (koju primjerice vrši robot), za koji su provedena brojna istraživanja (naprimjer, [41]). Tehnologija virtualne stvarnosti omogućuje razvijanje specifičnog područja virtualnog sklapanja dijelova koje uključuje i iznova aktualizira ovaj problem. Umetanje u stvarnosti mora biti često izvedeno s velikom točnošću i preciznošću, kako ne bi došlo do zabijanja osovine ili njezinog umetanja pod krivim kutom, što bi moglo rezultirati zaglavljivanjem osovine u provrtu, te njezinim naknadnim teškim izvlačenjem uz mogućnost oštećivanja površina ugradbenih elemenata.

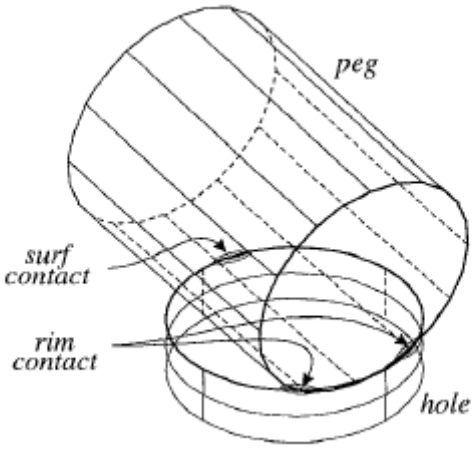
Kako bi se uspješno izvršio zadatak sklapanja, potrebno je obaviti nekoliko radnji:

1. locirati osovinu i provrt u virtualnom okruženju
2. uhvatiti osovinu
3. pozicionirati osovinu iznad provrta[25]
4. poravnati osi osovine i provrta
5. umetnuti.

Najkompleksnije radnje kod ove vrste sklapanja su navedene pod točkama 3 i 4. Točno pozicioniranje iznad provrta utječe na uspješnost sklapanja, gdje je bitno da se prije umetanja, na dovoljnoj udaljenosti između dijelova izvrši poravnavanje osi osovine i provrta, kako bi za umetanje osovine bilo potrebno izvršiti samo linearno gibanje za izvršavanje radnje pod točkom 5. Situacija (Slika 54.) prikazuje veliko odstupanje u poravnanju osi osovine i provrta, što rezultira potrošnjom vremena i potrebom za dodatnim gibanjem koje bi smanjilo grešku uslijed pozicioniranja.

[24] Ovaj problem poznat je pod engleskim nazivom *peg in hole*.

[25] Provrt će u ovom slučaju biti fiksiran, tj. uklješten, kako ne bi došlo do njegovog pomicanja.

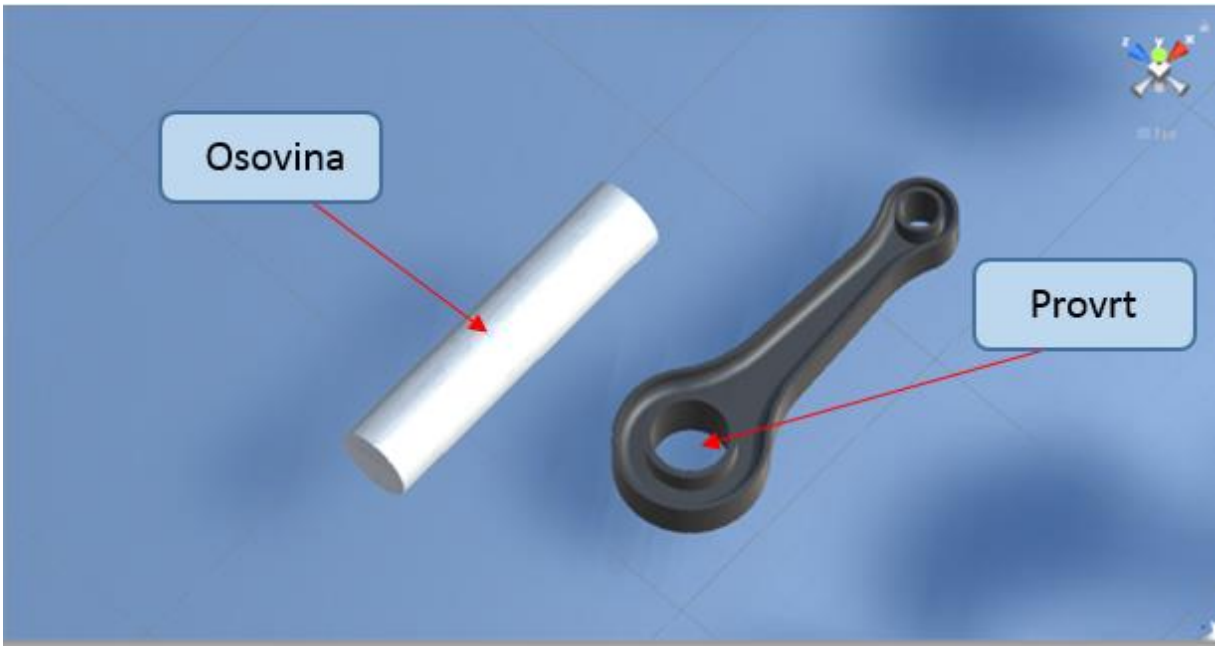


Slika 54. Odstupanje u pozicioniranju [41]

Izvršavanje zadataka sklapanja u virtualnom okruženju ne može toliko realno replicirati[26] proces stvarnog sklapanja, da bi došlo primjerice do oštećenja površina te uslijed toga do teškog izvlačenja ili potpune nemogućnost vađenja osovine iz provrta, ali može dovoljno dobro dočarati radnju otežanog umetanja osovine u provrt, prvenstveno zbog fizike koja je sastavni dio Unityja. Tako će umetanje osovine prevelikih dimenzija u premali provrt rezultirati nemogućnošću sklapanja, dok će primjerice kod ostvarenog prijelaznog dosjeda sklapanje biti moguće, iako u jednoj mjeri i otežano u odnosu na slučaj kada je provrt znatno veći od dimenzija osovine (u slučaju labavog dosjeda), što će odgovarati procesu sklapanja i u stvarnosti. Definiranje takvog sklapanja moguće je zbog opcije *Collider* koja je sastavni dio Unityja i koja vrlo vjerno omogućuje detekciju sudara između radnih objekata određenih dimenzija. Logično je zaključiti kako će u virtualnom okruženju dijelovi biti definirati kao idealni geometrijski oblici, koji će u potpunosti odgovarati predviđenim nazivnim mjerama, što dakako nije slučaj u realnom svijetu. U ovome primjeru umetat će se osovina cilindričnog oblika u kružni provrt koji se nalazi na klipnjači.

Za izvršavanje zadatka potrebno je izvršiti radnje pod prije definiranim točkama 1. do 5. Početni korak uključuje upoznavanje virtualnog okruženja i lociranje radnih komponenti, koje su u ovom slučaju osovina i provrt (Slika 55.).

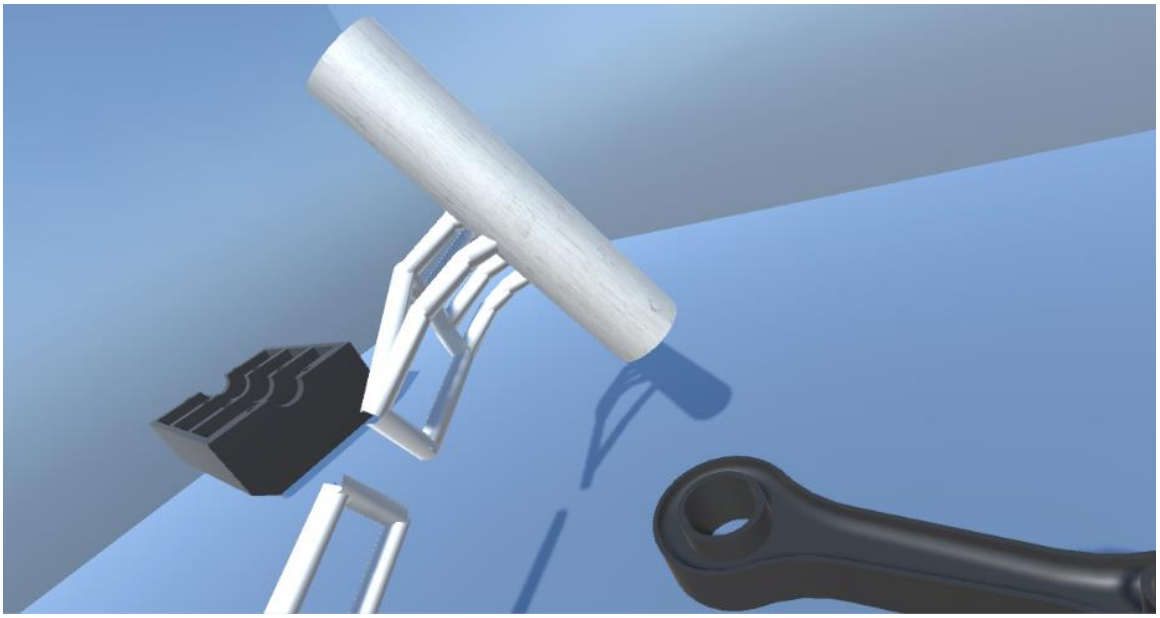
[26] tzv. *real physics*, naprimjer softver COMSOL



Slika 55. Lociranje radnih komponenti u virtualnom okruženju

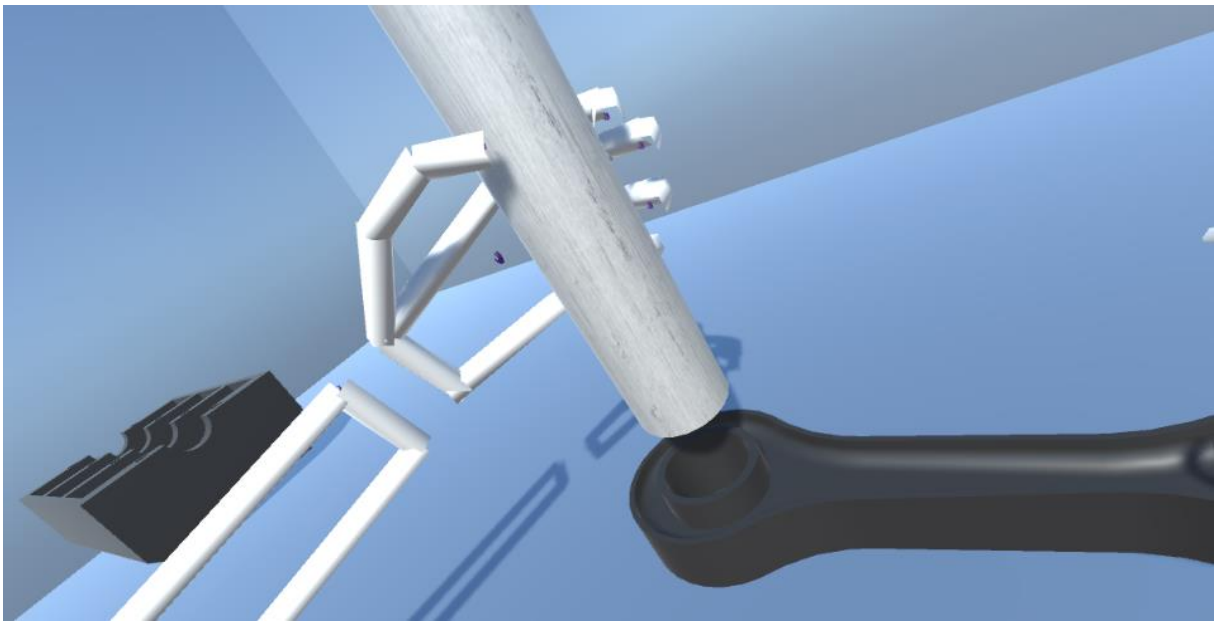
Za lakše izvršavanje zadatka sklapanja, translacije i rotacije na klipnjači bit će onemogućene u smjeru svih koordinatnih osi, kako bi se simuliralo uklještenje, odnosno kako ne bi došlo do pomicanja dijela (klipnjače).

Naredni koraci sastoje se od prihvata osovine i njezinog pozicioniranja iznad provrta u koji se ona umeće (Slika 56.).



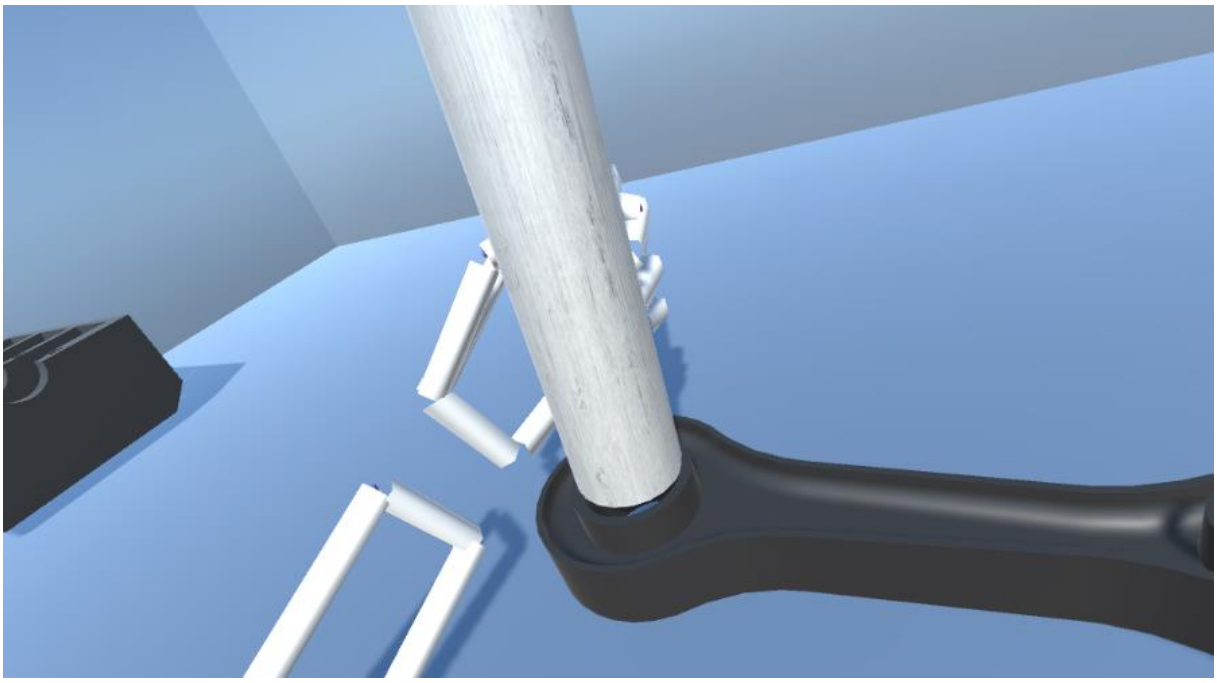
Slika 56. Prihvat i pozicioniranje osovine iznad provrta

Nadalje, potrebno je izvršiti poravnanje vertikalne *Y* osi osovine i provrta kako bi se moglo izvršiti umetanje (Slika 57.).



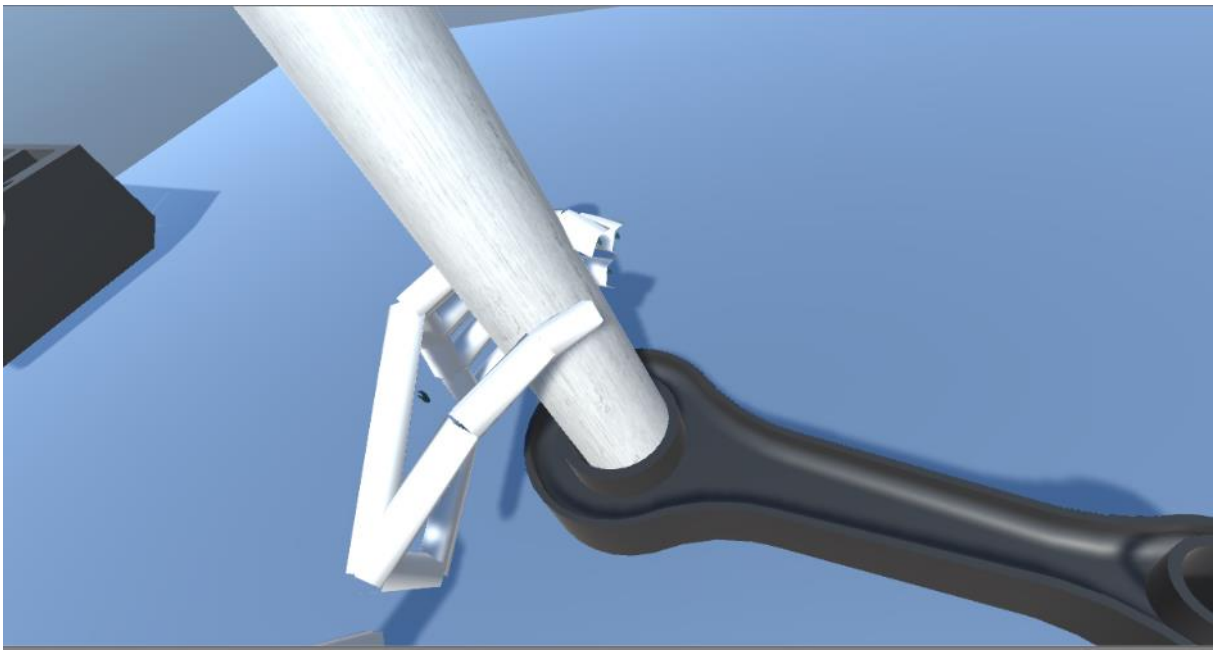
Slika 57. Poravnanje dijelova u smjeru vertikalne *Y* osi

Umetanje u ovom slučaju nije moguće izvršiti zbog nekompatibilnosti dimenzija radnih komada. Naime, dimenzija osovine veća je od dimenzije provrta, pa tako radnju pod točkom 5. u ovom slučaju nije moguće izvršiti. Zbog pozadinske fizike, dojam poteškoće umetanja bio je vrlo dobro simuliran, jer umetanje osovine u provrt nije bilo moguće izvršiti ni nakon više uzastopnih pokušaja umetanja (Slika 58.).



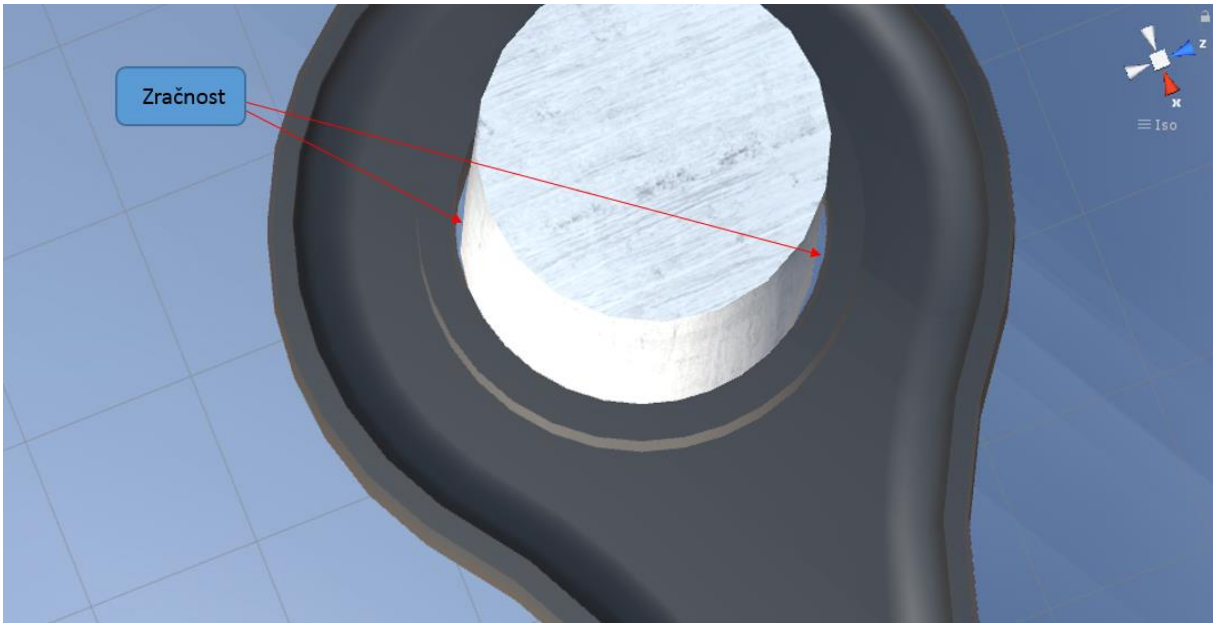
Slika 58. Nemogućnost izvršavanja zadatka umetanja

Smanjivanjem dimenzija osovine čime je cilj definirati prijelazni dosjed između komponenti, pokušat će se ponovno izvršiti proces sklapanja. Prve četiri točke zadatka su izvršene, te je potrebno provjeriti hoće li se moći izvršiti linearno gibanje koje će rezultirati pravilnim umetanjem i dobrim izvršenjem zadatka sklapanja (Slika 59.).



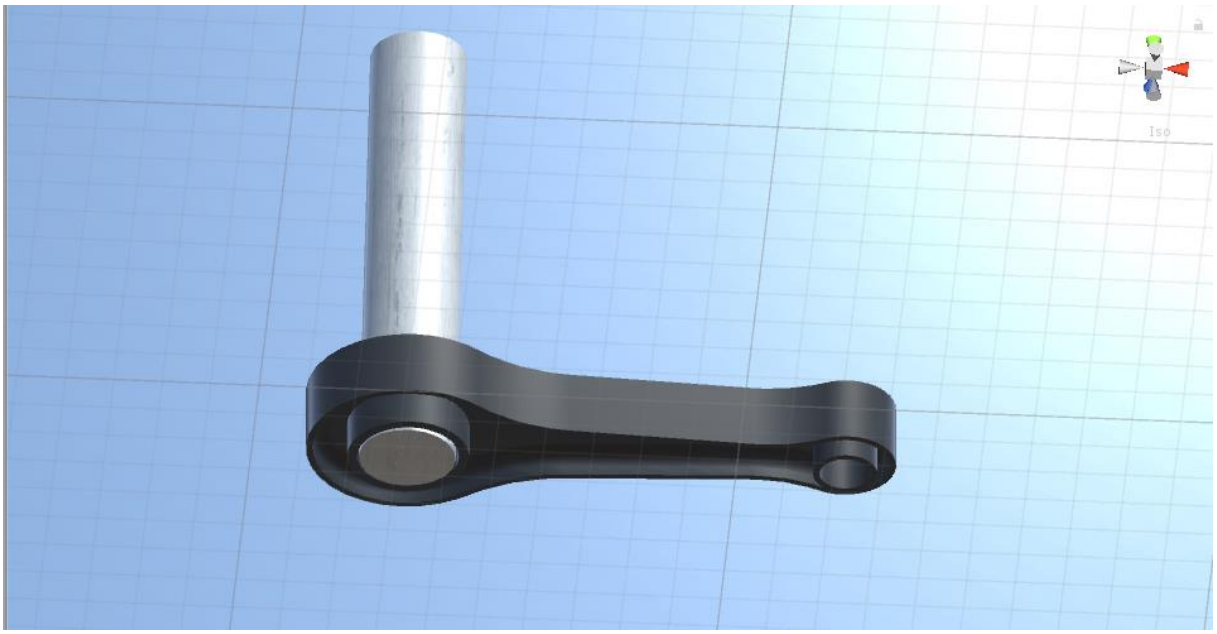
Slika 59. Pokušaj umetanja nakon smanjenja dimenzije osovine

Zbog smanjivanja dimenzija osovine, ovog puta nije bilo problema s umetanjem osovine u pripadajući provrt. Zadatak sklapanja bilo je moguće izvršiti bez poteškoća zbog zračnosti između komponenata (Slika 60.).



Slika 60. Zračnost između komponenata

Izvršavanje zadatka sklapanja u virtualnom okruženju omogućuje direktno upoznavanje radnika s radnim dijelovima bez prethodne potrebe za proizvodnjom fizičkih prototipova. Izrada fizičkih prototipova često je skupa, naročito ako se radi o komponentama kompleksne geometrije, ili proizvodima koji trebaju biti izrađeni od posebnih materijala. Pristup procesu sklapanja kroz tehnike virtualne stvarnosti automatski rezultira uštedama, ali i stjecanjem iskustva osoba koje će vršiti sklapanje, te njihovim upoznavanjem s radnom okolinom i radnim zadacima koji ih čekaju. U konkretnom primjeru, dojam realnosti bi se u potpunosti postigao korištenjem haptičkih rukavica koje bi mogle simulirati silu na rukama korisnika, te bi na taj način korisnik imao povratnu informaciju kada dođe do kolizije dijelova. [41]



Slika 61. Ostvareni sklop

4.2. SKLAPANJE ZIDNE PRIKLJUČNICE

Ovaj primjer prikazat će proces projektiranja putem virtualne stvarnosti, gdje će se ručno sklapati zidna priključnica. Krajnji cilj (projektiranja) je omogućiti da radnik učinkovito sklopi više pojedinačnih dijelova u sklop, uz pomoć odgovarajućeg alata i prikladnim razmještajem opreme na radnom mjestu. Izvođenjem sklapanja u virtualnom okruženju istražit će se mogućnost definiranja složenih radnji sklapanja u virtualnom okružju, te posebno, omogućiti usporedba vremena izvođenja radnji u virtualnom okružju s vremenima koja se dobivaju tradicionalnim postupcima projektiranja ručnih radnji sklapanja -- sustavom unaprijed određenih vremena, MTM[27].

4.2.1. Opis i struktura zidne priključnice

Budući da je primjena zidne priključnice neizbježna u svakom objektu gdje se koriste jednofazna trošila, te je stoga jedna od osnovnih električnih ugradbenih proizvoda, javlja se zahtjev za velikim proizvodnim količinama. Funkcija je proizvoda osigurati vezu električnog

[27] engl. *Methods-Time Measurement*

aparata s naponskom mrežom od 220 V i frekvencijom od 50 Hz uz najveću dopuštenu jakost struje od 16 A. Priključnica se spaja s mrežom preko dvaju radnih kontakata koji prenose struju i napon do električnog aparata koji je trenutno priključen. Vodič uzemljenja spaja se sa zaštitnim kontaktom zidne priključnice. Uz sklopove radnog i zaštitnog kontakta, ostali su dijelovi priključnice:

- vanjsko kućište,
- okvir kućišta,
- unutrašnji limeni okvir,
- nosač kontakata,
- držač kontakata,
- vijci i matice.

Razmatrana zidna priključnica proizvod je tvrtke TEP – Tvornice elektrotehničkih proizvoda, iz Zagreba. Izrađena je u skladu s važećim međunarodnim standardima IEC[28] 83 i VDE 0620[29]. Stupanj mehaničke zaštite reguliran je standardima IEC 529 i CEI 70-1 koji propisuju stupanj zaštite koju osigurava kućište proizvoda. Strukturna sastavnica zidine priključnice dana je tablicama 2 i 3[30]. [42 i 43]

[28] Međunarodno povjerenstvo za elektroniku

[29] njemačka elektrotehnička norma

[30] Crteži tj. njihovi brojevi navedeni u strukturnoj sastavnici nalaze se u [42] odnosno u [44].

Tablica 2. Strukturna sastavnica zidne priključnice, List 1/2 [44]

FSB
Fakultet strojarstva i brodogradnje Zagreb

STRUKTURNA SASTAVNICA

Naziv:

Identitet:

Datum izrade: 2001-02-18

Broj crteža: SR-00-00

K:

Kl:

List: 1

Listova: 2

Stupanj ugradnje	Identitet ugradbenog dijela	Naziv ugradbenog dijela	Format crteža	Broj crteža	Broj pozicije na sklopom crtežu	K	Kl	JM	Količina	Broj izmjene	Datum izmjene	S	Datum stupanja
1		Vanjsko kućište	A3	SR-00-01	1	D	10	11	1				
1		Okvir kućišta	A3	SR-00-02	2	D	10	11	1				
1		Središnji vijak M3,5 x 10	-	Standard	3	D	20	11	1				
1		Sklop 5	-	-	-	S	30	11	1				
2		Stezni lim	A3	SR-00-05	6	D	20	11	2				
2		Sklop 4	-	-	-	S	30	11	1				
3		Unutrašnji limeni okvir	A3	SR-00-06	7	D	20	11	1				
3		Sklop 3	-	-	-	S	30	11	1				
4		Središnja raskivna matica	A3	SR-00-03	4	D	20	11	1				
4		Nosač kontakata	A3	SR-00-11	13	D	10	11	1				
4		Sklop 1	-	-	-	S	30	11	1				
5		Radni kontakt	A3	SR-00-09	10	D	20	11	2				
5		Opruga	A4	SR-00-08	9	D	20	11	2				
5		Kombi vijak M3,5 x 8	-	Standard	11	D	20	11	2				
5		Klizna matica M3,5	A3	SR-00-10	12	D	20	11	2				
4		Držač kontakata	A3	Sr-00-12	14	D	10	11	1				
4		Sklop 2	-	-	-	S	30	11	1				
5		Zaštitni kontakt	A3	SR-00-07	8	D	20	11	1				
5		Kombi vijak M3,5 x 8	-		11	D	20	11	1				

JM - jedinica mjere:
11 - komad, 20 - gram, 21 - dkg, 22 - kg, 31 - mm, 33 - cm, 41 - mm^2, 42 - cm^2, 44 - m^2, 52 - cm^3, 53 - dm^2, 54 - m^3, 61 - dcl, 62 - l

K - karakter dijela:
D - dio u užem smislu, F - fiktivni sklop, M - materijal (sirovina), P- gotov proizvod, S – sklop, T - standardni dio

Kl - ključ nositelja izrade:
10-ljevaonica, 15-teška obrada, 20-laka obrada, 30-montaža, 40-pogon održavanja, 54-nabava, 55-kooperacija domaća, 56-kooperacija inozemna

S - status ključ:
U - ubacivanje dijela, B - brisanje dijela, R - izvedeni dio, * - alternativni dio

Broj sastavnice:

Tablica 3. Strukturna sastavnica zidne priključnice, List 2/2 [44]

FSB
Fakultet strojarstva i brodogradnje Zagreb

STRUKTURNA SASTAVNICA

Naziv:

Identitet:

Datum izrade: 2001-02-18

Broj crteža: SR-00-00

K:

Kl:

List: 2

Listova: 2

Stupanj ugradnje	Identitet ugradbenog dijela	Naziv ugradbenog dijela	Format crteža	Broj crteža	Broj pozicije na sklopom crtežu	K	Kl	JM	Količina	Broj izmjene	Datum izmjene	S	Datum stupanja
5		Klizna matica M3,5	A3	SR-00-10	12	D	20	11	1				
3		Bočna raskivna matica	A3	SR-00-04	5	D	20	11	2				
2		Vijak M3 x 12	-	Standard	15	D	20	11	2				

JM - jedinica mjere:
11 - komad 31 - mm 44 - m^2 54 - m^3
20 - gram 33 - cm 52 - cm^3 61 - dcl
21 - dkg 41 - mm^2 53 - dm^2 62 - l
22 - kg 42 - cm^2

K - karakter dijela:
D - dio u užem smislu P- gotov proizvod
F - fiktivni sklop S – sklop
M - materijal (sirovina) T - standardni dio

Kl - ključ nositelja izrade:
10-ljevaonica 40-pogon održavanja
15-teška obrada 54-nabava
20-laka obrada 55-kooperacija domaća
30-montaža 56-kooperacija inozemna

S - status ključ:
U - ubacivanje dijela
B - brisanje dijela
R - izvedeni dio
* - alternativni dio

Broj sastavnice:

4.2.2. Elementi rada i plan montaže

Element rada je najmanji racionalni dio rada, koji se može nezavisno izvoditi, i kojim se definira takvo stanje djelomično montiranog sklopa da se može pomicati na drugo radno mjesto bez neželjenog rasklapanja. Element rada dijeli se na pokrete. Pokreti se dadu opisati i izmjeriti, ali se u okviru montažnoga procesa ne mogu izolirano izvesti (naprimjer. posegnuti, uhvatiti). Podjelom montažnoga procesa na elemente rada, utvrđuje se točan sadržaj rada i vremena za njegovo izvođenje, te se mogu definirati točne upute za izvršenje rada i odrediti troškovi montažnoga procesa. [43]

Grupiranjem elemenata rada po radnim mjestima oblikuju se montažne operacije. Nadalje, plan montaže je osnovni dokument koji sadrži jednoznačan opis procesa montaže i sve potrebne informacije za njegovo izvođenje i kontrolu. Izrađuje se za svaki sklop i završnu montažu proizvoda. Funkcija mu je tehničko i organizacijsko definiranje procesa montaže. Tehnička funkcija se odnosi na izvođenje procesa i sadrži informacije o: potrebnim ugradbenim elementima, elementima rada u najpovoljnijem redoslijedu i opremi radnih mjesta (alat, naprave, strojevi i slično). Organizacijska se funkcija odnosi na planiranje, ustroj, upravljanje i obračun troškova montažnoga procesa, a izražava se informacijama o mjestu troška, norma vremenu, grupi plaćanja i veličini serije.

Plan montaže zidne priključnice prikazan je u tablici 4. [44]

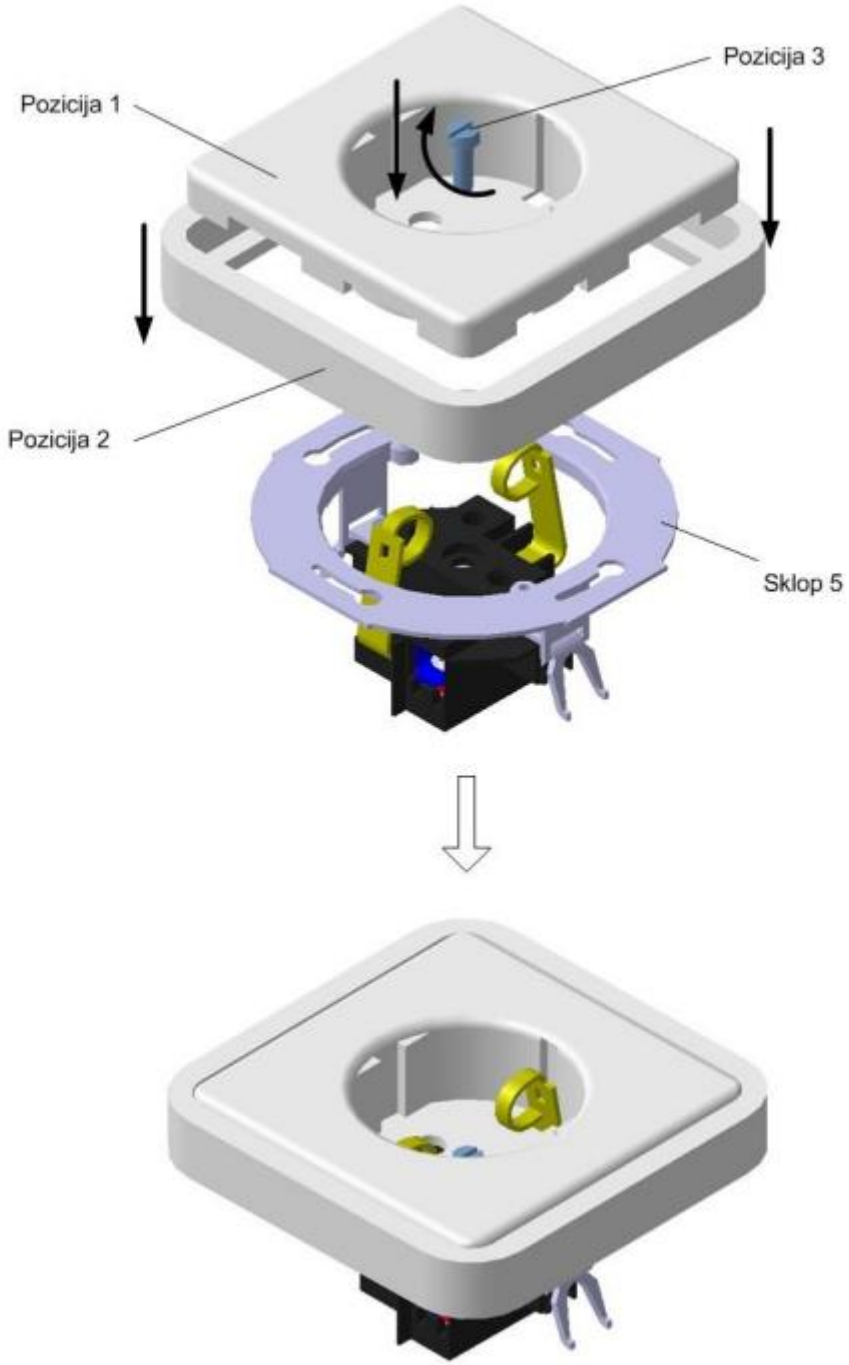
U virtualnom okruženju biti će prikazano sklapanje elementa rada označenog brojem 7, koji se odnosi na sklapanje pozicija 1, 2 i 3 te Sklopa 5. Sredstvo za montažu će biti vijčalo, gdje je potrebno poziciju 3 preko središnjeg provrta postaviti na početak navoja, uvrnuti i izvršiti stezanje, nakon što su u pravilan odnos uvedene pozicije 1 i 2, te Sklop 5[31]. [42]

[31] S obzirom da se radi s dijelovima koji su modelirani u prijašnjim radovima, moguće su određene nedosljednosti u geometriji dijelova.

Tablica 4. Plan montaže zidne priključnice [44]

FSB Fakultet strojarstva i brodogradnje, Zagreb	PLAN MONTAŽE		Datum	Mjesto troška:	List: 1
Zamjena za:	Naziv: **ZIDNA PRIKLJUČNICA**	Izradio	2001-02-20	F. Hlebec	Broj radnog mjesta:	Listova: 1
Zamijenjen sa:	Identitet:	Kontrolirao		Dr.sc. Zoran Kunica	Takt (minuta):	T_{pz} (minuta):
Vrijedi, od-do:	Broj crteža:	Optimalna serija, od-do (komada):		Veličina serije (komada):	Komada/smjena:	

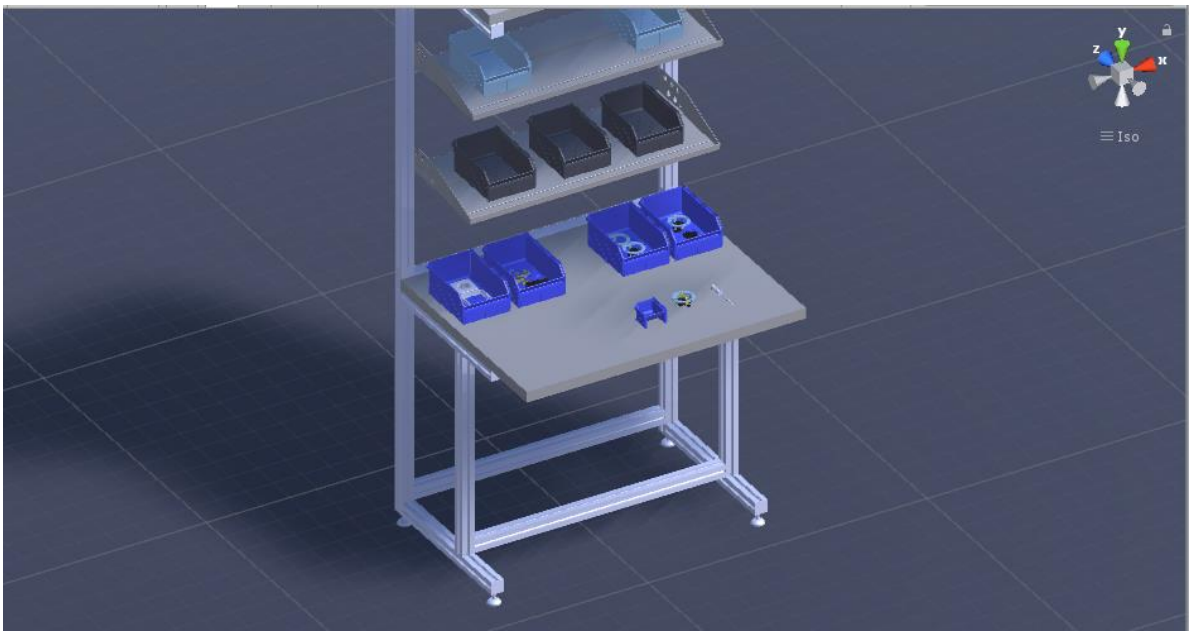
Oznaka radnog mjesta/ stroja	Broj radnika	Oznaka elementa rada	Opis elemenata rada	Ugradbeni elementi, materijal	Broj crteža/ standard	Kom.	Sastavnica		Sredstva za montažu, kontrolu i ispitivanje	Vrijeme izrade t_i (minuta)	Norma vrijeme t_1 (minuta)	Kvali. radnika
							List	Broj poz.				
MM1		1	Utisnuti poziciju 9 u poziciju 10 (P9+P10)	Pozicija 9 Pozicija 10	SR-00-08 SR-00-09	1 1	10 11	9 10				
MM2		2	Umetnuti poziciju 11 u provrt pozicije 10, poziciju 12 uvrnuti na poziciju 11 i pritegnuti vijčalom (Sklop1)	P9+P10 Pozicija 11 Pozicija 12	- HRN M B1.051 SR-00-10	2 2 2	- - 12	- 11 12	Vijčalo			
MM3		3	Umetnuti poziciju 11 u provrt pozicije 8, poziciju 12 uvrnuti na poziciju 11 i pritegnuti vijčalom	Pozicija 8 Pozicija 11 Pozicija 12	SR-00-07 HRN M B1.051 SR-00-10	1 1 1	9 - 12	8 11 12	Vijčalo			
MM4		4	Umetnuti sklopove 1 i sklop 2 u utore pozicije 13, zatvoriti pozicijom 14, umetnuti poziciju 4 kroz provrt pozicija 13 i 14, te je zakovati (Sklop 3)	Sklop 1 Sklop 2 Pozicija 4 Pozicija 13 Pozicija 14	- - SR-00-03 SR-00-11 SR-00-12	2 1 1 1 1	- - 5 13 14	- - 4 13 14	Uređaj za zakivanje			
MM5		5	Staviti poziciju 7 na sklop 3, kroz njihove provrte utisnuti pozicije 5, te ih zakovati (Sklop 4)	Sklop 3 Pozicija 5 Pozicija 7	- SR-00-04 SR-00-06	1 1 1	- 6 8	- 5 7	Uređaj za zakivanje			
MM6		6	Umetnuti pozicije 6 u pravokutne otvore sklopa 4, pozicije 15 postaviti preko provrta pozicija 6 do početka navoja, uvrnuti ih i pritegnuti vijčalom (Sklop 5)	Sklop 4 Pozicija 6 Pozicija 15	- SR-00-05 HRN M B1.103	1 2 2	- 7 -	- 6 15	Vijčalo			
MM7		7	Poziciju 1 umetnuti u poziciju 2, te ih preko pozicije 8 utisnuti na sklop 5, poziciju 3 preko središnjeg provrta postaviti na početak navoja, uvrnuti i pritegnuti vijčalom (Proizvod)	Sklop 5 Pozicija 1 Pozicija 2 Pozicija 3	- SR-00-01 SR-00-02 HRN M B1.130	1 1 1 1	- 3 4 -	- 1 2 3	Vijčalo			



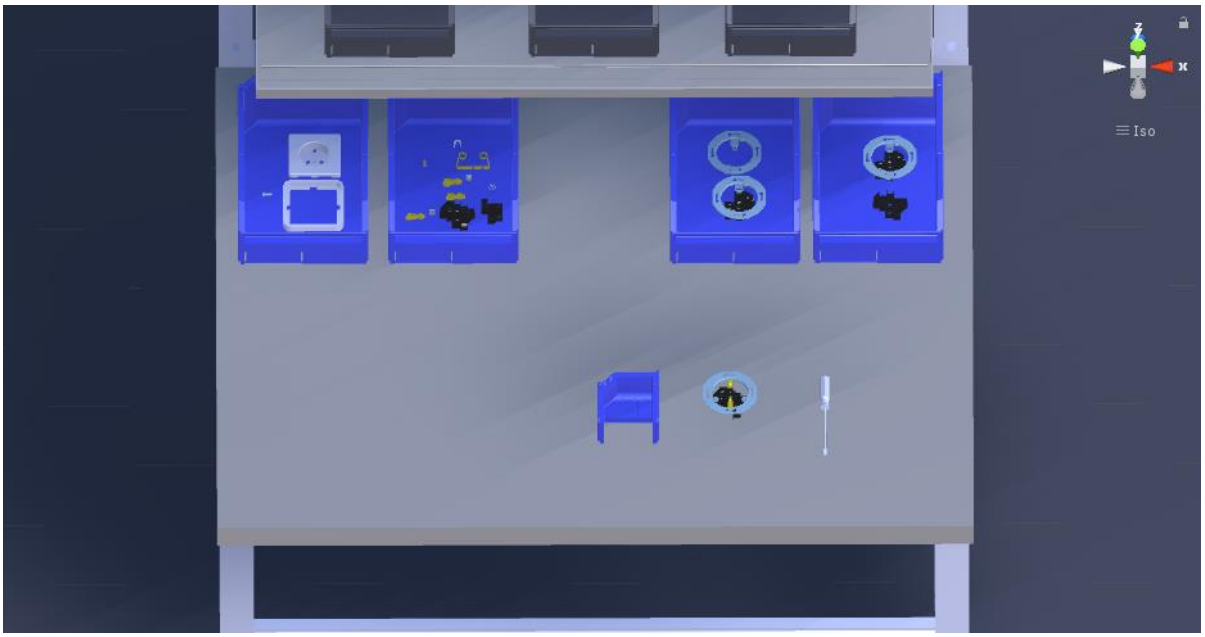
Slika 62. Sklapanje ugradbenih elemenata zidne priključnice [42]

4.2.3. Definiranje radnog okruženja za virtualno sklapanje

Da bi se u virtualnom okruženju izvršilo sklapanje, potrebno je definirati radno okruženje. Za definiranje okruženja koristit će se gotova oprema dostupna iz *on-line* kataloga Rexroth, te dijelovi već prije definirane zidne priključnice [42]. Radno mjesto sadržavat će potrebne kutije u kojima se nalaze pojedinačni dijelovi, te adekvatan prostor ispred radnika za izvršavanje montaže (slike 63. i 64.).



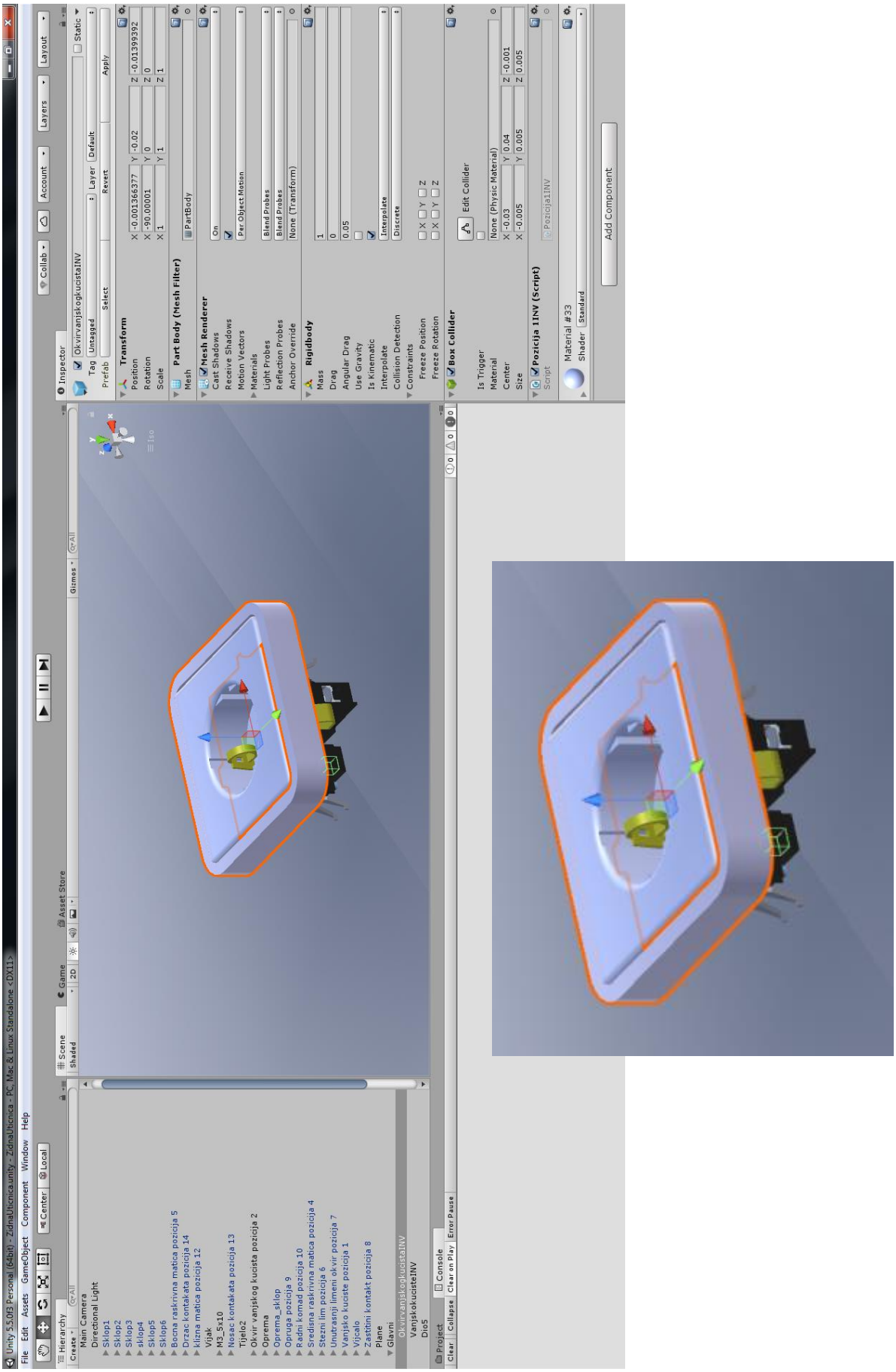
Slika 63. Definirano virtualno radno okruženje



Slika 64. Virtualna radna površina

Na mjestu gdje će se vršiti sklapanje potrebno je unaprijed odrediti poziciju sklopa, jer će se za simulaciju sklapanja koristiti jednostavne funkcije koje uključuju odnosno isključuju grafički prikaz pojedinačnih komponenti u trenutku kada dođe do detekcije sudara između dijelova. Pomoću sudara tj. kolizije dijelova zapravo će se prikazivati dovođenje pojedinačnih dijelova u međusoban odnos. Primjerice, dio koji uhvati montažer mora se prenijeti na odgovarajuće mjesto za montažu, gdje je bitno da se u trenutku pravilnog pozicioniranja i izvršenja sklapanja dijelovi vizualno što realnije međusobno dotaknu. Da bi uopće moglo doći do detekcije sudara, potrebno je pridijeliti (već poznata) svojstva *Collider* pojedinačnim dijelovima, koja će reagirati na detekciju sudara između komponenti. Zbog intuitivnijeg i realnijeg osjećaja sklapanja, neće se koristiti značajka *Mesh Collider* jer će kod uključivanja te opcije već kod malog kontakta između površina dijelova doći do kolizije i isključivanja grafičkog prikaza jednog dijela, odnosno uključivanja drugog, što neće pridonijeti realnom prikazu sklapanja. Zbog toga će se koristiti primitivni *Collideri* koji će biti najčešće smješteni na udaljenim rubovima dijelova, kako bi do kolizije načelno došlo što teže, odnosno kako bi se korisnik trebao potruditi što bolje pozicionirati dio unutar okruženja, kako bi sklapanje djelovalo što realnije.

Na slici 65. vidljiva je pozicija na kojoj će se vršiti sklapanje sklopa zidne priključnice (odnosno prikazana je već sklopljena zidna priključnica).



Slika 65. Pozicija sklapanja i već sklopljena zidna priključnica (prikazana i uvećano)

Sklop je u scenskom prikazu doveden na poziciju sklapanja. U trenutku klika na gumb *Play*, pokrenut će se *Game* prikaz, istovjetan onome na *headset*u virtualne stvarnosti. Po pokretanju *Game* prikaza vizualni prikaz pojedinih dijelova sklopa sa slike 65. će se isključiti u svrhu izvršenja zadatka sklapanja. Dijelovi sklopa čiji će prikaz biti isključen (vanjsko kućište, okvir vanjskog kućišta i vijak) trebat će se sklopiti kako bi se ponovno prikazali. To će biti ostvareno preko sudara između istih takvih dijelova iz kutije (vanjsko kućište, okvir vanjskog kućišta i vijak), koje izuzima korisnik, i grafički nevidljivih dijelova čiji je prikaz prethodno isključen, a za koje su u pozadini aktivne *Collider* komponente, koje u svakom trenu provjeravaju međusobnu koliziju dijelova. Zbog toga se u lijevom kutu okvira vanjskog kućišta (Slika 65.) nalazi zelena kockica koja predstavlja *Box Collider* svojstvo. Isto tako, u trenutku pokretanja aplikacije, navedena kockica se također neće vidjeti na headsetu virtualne stvarnosti niti unutar *Game* prikaza. Takvo ponašanje je omogućeno preko pridijeljene skripte (Izlist 2.) koja isključuje grafički prikaz dijela u trenutku pokretanja aplikacije te ga uključuje u trenutku kada dođe do aktivacije sudara između *Collidera* (zelene kockice) grafički nevidiljivog okvira vanjskog kućišta i okvira vanjskog kućišta (pozicija 2), kojeg korisnik uzima unutar odgovarajuće kutije s dijelovima u trenutku kada vrši sklapanje.

Što se tiče dijela s kojim se vrši sklapanje, a koji se nalazi u kutiji, kod njega su svojstva definirana drugačije. Tom dijelu se također pridijeli *Collider* svojstvo, ali je njegov grafički prikaz u trenutku hvatanja dijela uključen, sve do trenutka kolizije s *Collider*om istog takvog dijela na sklopu, čiji je grafički prikaz prethodno isključen. Zbog toga je druga skripta (Izlist 3.) pridijeljena dijelu koji se izuzima iz kutije. Istovremenom kombinacijom dviju skripti u trenutku kolizije uključuje se dakle grafički prikaz do tada nevidljivog dijela na sklopu, odnosno isključuje se grafički prikaz dijela kojeg korisnik drži u rukama, bez obzira o kojem se dijelu radi, čim je zadovoljen uvjet da se sudaraju isti dijelovi (što je definirano preko naziva dijela unutar programskog koda). Konkretno na primjeru vanjskog kućišta, u trenutku pokretanja aplikacije, okvir vanjskog kućišta na sklopu neće biti vidljiv sve dok korisnik ne dovede u međusobni odnos Sklop 5 i prikazanu poziciju 2. U trenutku kada pozicija 2 iz kutije, odnosno okvir vanjskog kućišta, koji se nalazi u korisnikovim rukama, dotakne grafički nevidljivi *Box Collider* (tj. zelenu kockicu) isključit će se prikaz komponente u korisnikovim rukama, te će se uključiti prikaz komponente u sklopu, što će rezultirati dojmom da je

korisnik doveo u odnos poziciju 2 i Sklop 5, odnosno da je izvršio sklapanje komponente. Sve to je moguće zahvaljujući pozadinskoj fizici ali i jednostavnim skriptama koje reagiraju na koliziju i tako uključuju/isključuju grafički prikaz pojedine komponente. Okviru vanjskog kućišta koji se nalazi u kutiji s dijelovima pridijeljen je naziv *Tijelo2*, dok je istom takvom okviru koji je u početku programa grafički nevidiljv, a nalazi se na sklopu, pridijeljen naziv *OkvirvanjskogkucistaINV*[32].

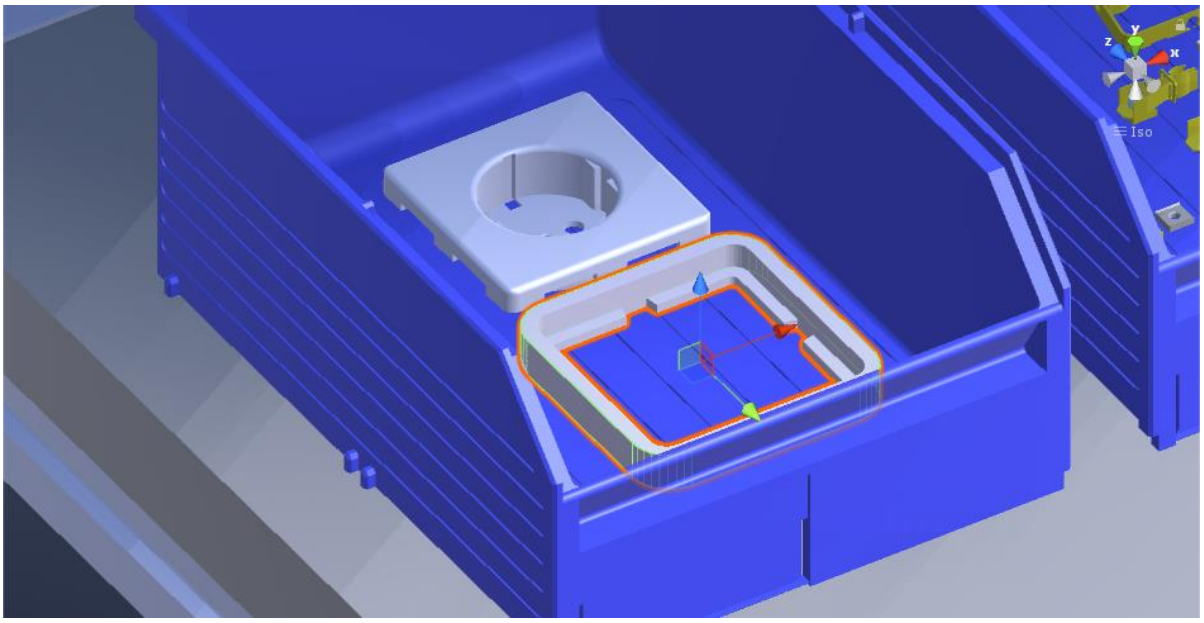
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Pozicija2 : MonoBehaviour {

    void Start()
    {
        GetComponent<MeshRenderer>().enabled = false;
    }

    // Update is called once per frame
    void OnCollisionEnter(Collision pozicija2)
    {
        if (pozicija2.gameObject.name == "Tijelo2"
)
        {
            GetComponent<MeshRenderer>().enabled = true;
        }
    }
}
```

Izlist 2. Skripta kolizije

Iz izlista 2. je vidljivo kako je grafički prikaz komponente u trenutku pokretanja skripte onemogućen (*MeshRenderer.enabled=false*), međutim, preko detekcije sudara i *if* funkcije definiran je uvjet koji u trenutku sudara uključuje grafički prikaz komponente ako je naziv objekta s kojim je došlo do sudara *Tijelo2*. Dakle, u tom trenutku će *MeshRenderer.enabled* biti jednak *true* što će uključiti grafički prikaz komponente. Nadalje, okvir vanjskog kućišta koji se nalazi u kutiji s dijelovima (Slika 66.), imat će pridijeljenu skriptu *Pozicija2INV.cs* (Izlist 3.) koja će vršiti suprotnu stvar. Tamo će grafički prikaz biti uključen, sve do trenutka dok ne dođe do sudara između dijelova, pod uvjetom da su se sudarila tijela naziva

[32] Unity je osjetljiv na mala i velika slova te razmake između riječi, stoga je ovo karakterističan oblik naziva objekta bez korištenja razmaka i dijakritičkih znakova.

OkvirvanjskogkucistaINV i *Tijelo 2*, nakon čega će se aktivirati komponenta u sklopu te isključiti grafički prikaz okvira vanjskog kućišta u rukama korisnika.



Slika 66. Okvir vanjskog kućišta u kutiji s dijelovima

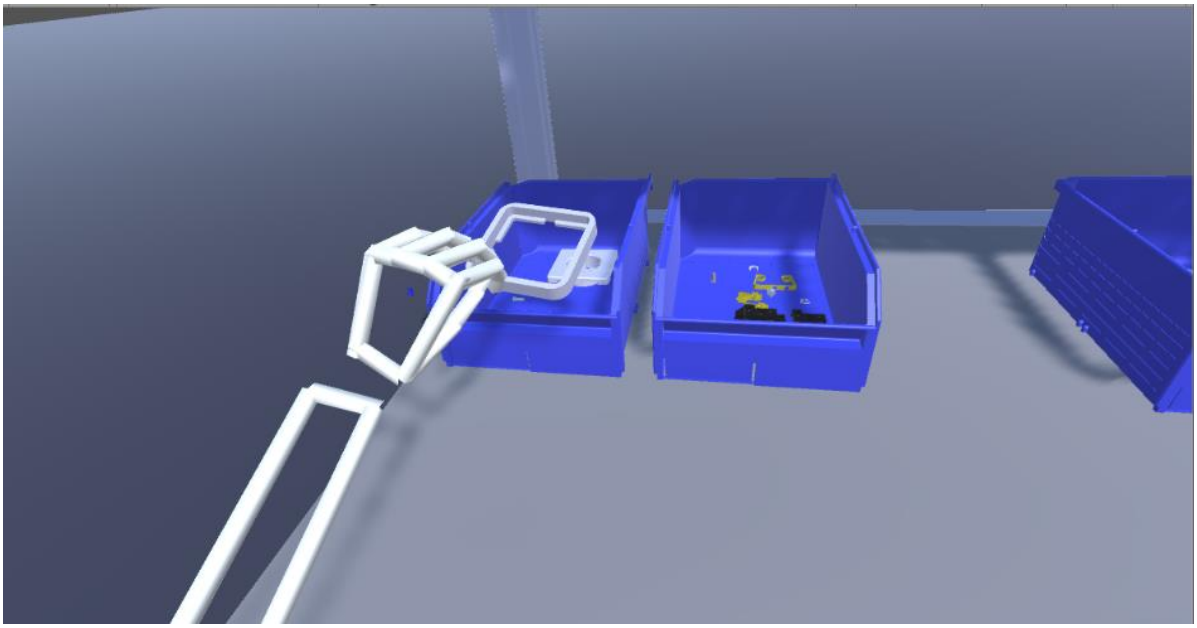
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Pozicija2INV : MonoBehaviour {

    void Start()
    {
        GetComponent<MeshRenderer>().enabled = true;
    }

    // Update is called once per frame
    void OnCollisionEnter(Collision pozicija2)
    {
        if (pozicija2.gameObject.name == "OkvirvanjskogkucistaINV"
)
        {
            GetComponent<MeshRenderer>().enabled = false;
            GetComponent<Collider>().enabled = false;
        }
    }
}
```

Izlist 3. Suprotna skripta kolizije

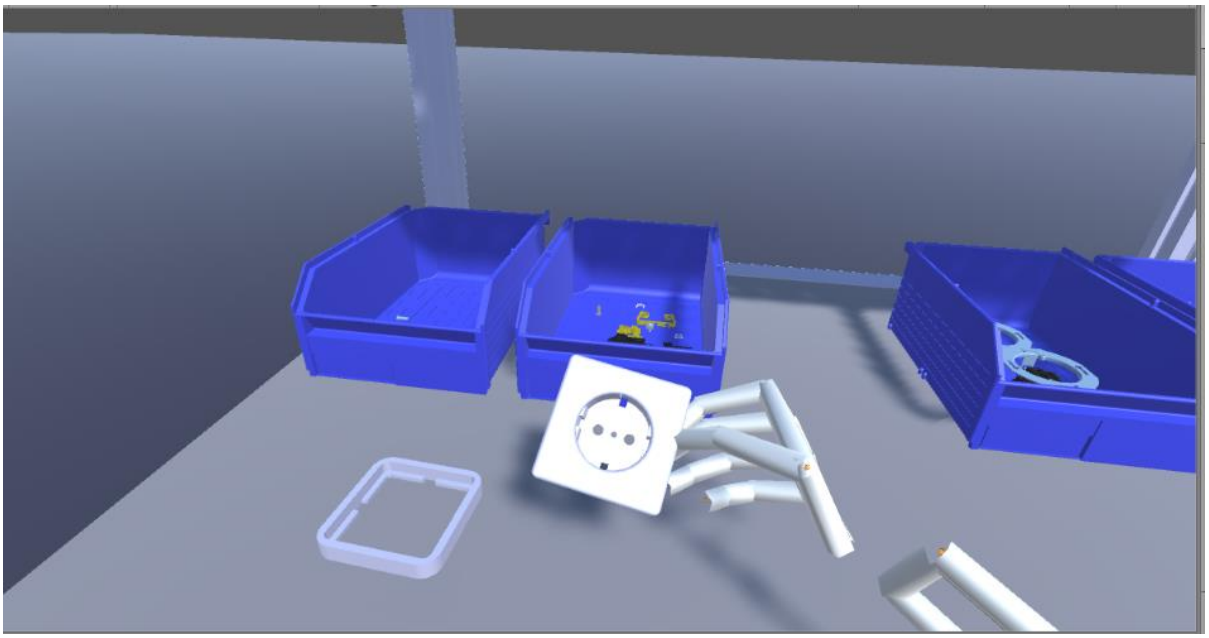
Na isti način definirane su i ostale pozicije u virtualnom okruženju na kojima se vrši proces sklapanja, a koje uključuju vanjsko kućište i vijak.

Kada je okruženje u potpunosti definirano, može se početi s izvršenjem zadatka sklapanja koristeći dostupan plan montaže [42]. Nakon pokretanja aplikacije i pravovaljanog pozicioniranja, kreće se s izuzimanjem dijelova s kojima će se vršiti sklapanje. Sklapanje će biti prikazano za element rada označen brojem 7, što znači da je u početku potrebno u pravilan odnos dovesti pozicije 1 i 2, tj. vanjsko kućište i okvir vanjskog kućišta u odnosu na Sklop 5, te nakon toga izvršiti stezanje komponenti preko vijka, zajedno sa Sklopom 5.

Prva radnja zahtijeva uzimanje pojedinačnih komponeta iz dviju kutija, te njihovo pozicioniranje na radnoj površini. Na slici 67. vidljivo je hvatanje okvira vanjskog kućišta. Potom, iz prve kutije izuzima se vanjsko kućište i pozicionira (odlaže) na radnoj površini, pored okvira vanjskog kućišta (Slika 68.).

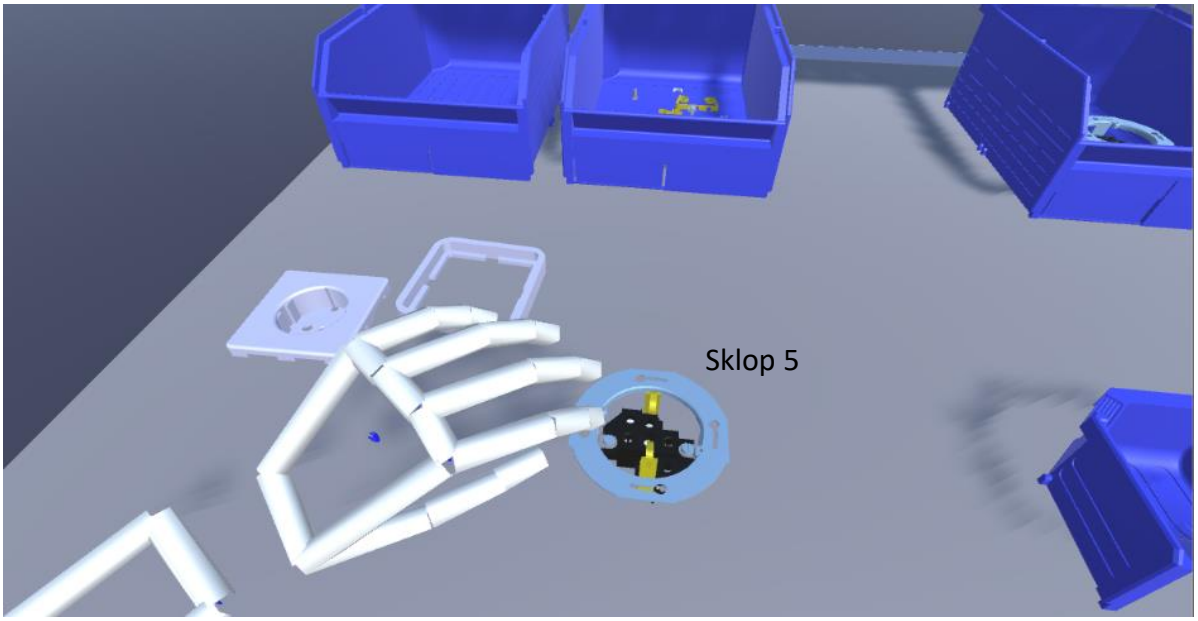


Slika 67. Hvatanje okvira vanjskog kućišta



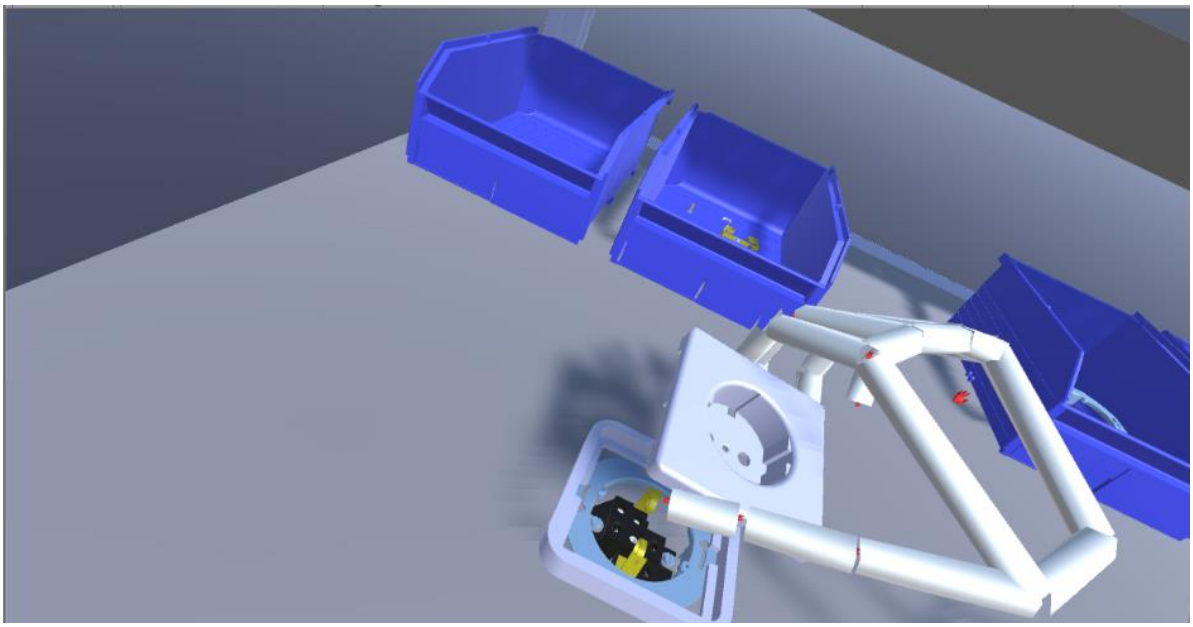
Slika 68. Pozicioniranje vanjskog kućišta na radnoj površini

Pozicija 1 i pozicija 2, tj. vanjsko kućište i okvir vanjskog kućišta, kasnije je potrebno dovesti u međusoban odnos, a također i sa Sklopom 5 (Slika 69.).



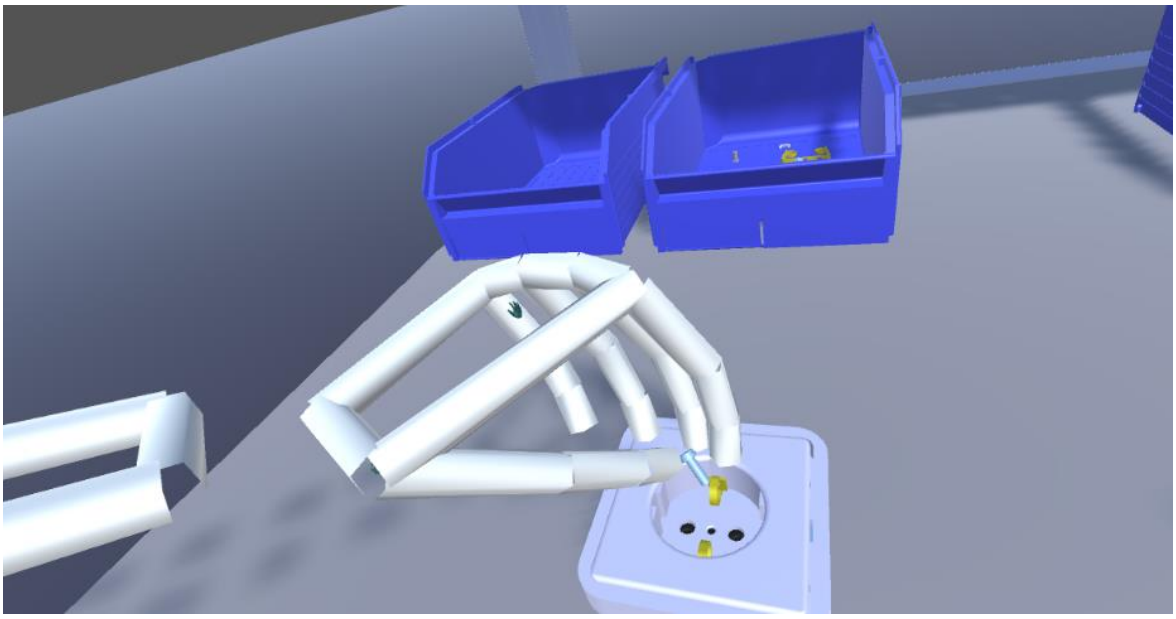
Slika 69. Radna površina s odloženim Sklopom 5

Prvo će se sklopiti okvir vanjskog kućišta sa Sklopom 5, a zatim novonastali sklop spojiti s vanjskim kućištem (Slika 70.).



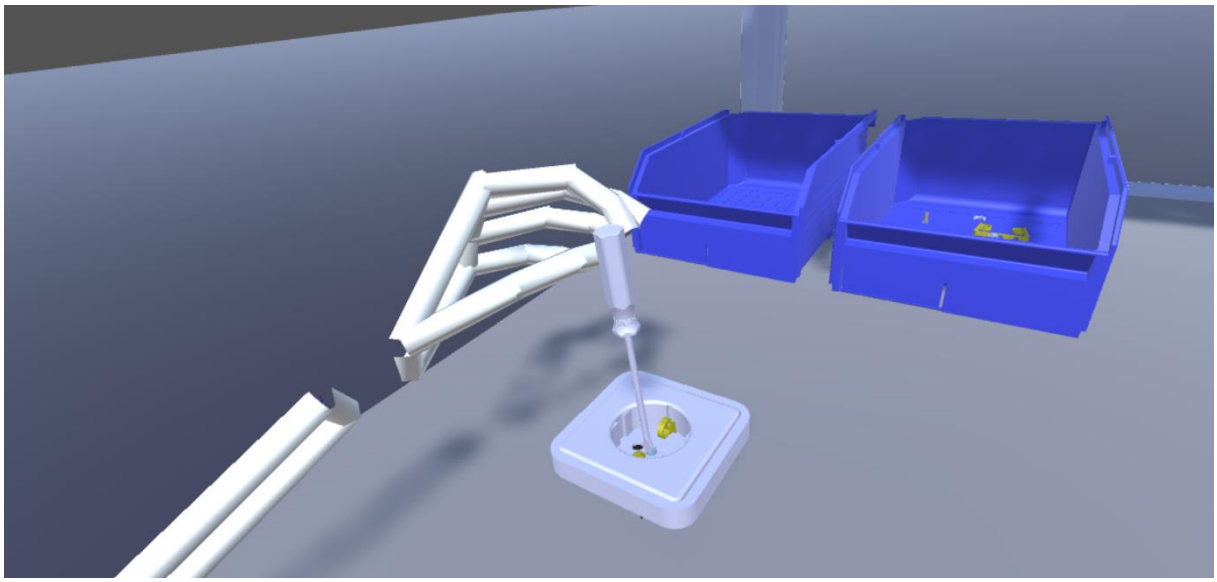
Slika 70. Spajanje vanjskog kućišta

Nakon što su prethodne tri komponente pravilno sklopljene, potrebno je pričvrstiti vanjsko kućište i njegov okvir preko vijka, čije je umetanje vidljivo na slici 71.



Slika 71. Umetanje vijka

Vijak je potrebno pritegnuti pomoću odvijača kako bi proces sklapanja bio dovršen u potpunosti, tj. montirana cijela zidna priključnica (Slika 72.).



Slika 72. Simulacija pritezanja vijka

Prethodno sklapanje komponenti izvedeno je u potpunosti intuitivno, bez uvježbanosti i bez ikakve analize rada[33]. U nastavku će biti prikazano projektiranje procesa sklapanja elementa rada označenog brojem 7 tradicionalnim postupkom projektiranja, dok će se zadatak virtualnog sklapanja izvršiti istovremeno s dvije ruke, kad god je to moguće, uz pokušaj za što vjerniju replikaciju postupka sklapanja s obzirom na prkasu i analizu rada koja će biti vidljiva u tablicama 8. i 9. Na kraju će biti prikazana usporedba vremena izvršavanja zadatka dobivenog projektiranjem procesa sklapanja tradicionalnim postupkom te vremena koje je potrebno za izvršavanje zadatka u virtualnom okruženju, i to slijeđenjem (pokreta iz) analize rada napravljene tradicionalnim postupkom. Pojedine radnje koje će biti vidljive u analizi neće biti moguće izvršiti u virtualnom okruženju, o čemu će biti više govora u nastavku.

4.2.4. Analiza i određivanje vremena sklapanja sustavom MTM

U virtualnom okruženju biti će definiran brojač vremena koji će poslužiti za usporedbu s tradicionalnim postupcima projektiranja koji uključuje primjenu sustava MTM. Sustav unaprijed određenih vremena, MTM, služi za mjerenje (normiranje) ljudskog rada, a sastoji se od postupaka raščlane ručnih (tjelesnih) radnji na osnovne pokrete, gdje se svakom pokretu dodjeljuje unaprijed standardno vrijeme, a na osnovi vrste pokreta i uvjeta pod

[33] U potpunosti intuitivno sklapanje uzelo je više vremena od sklapanja koje je slijedilo analizu rada.

kojima se izvodi. Vrijeme automatske operacije i tehnološko vrijeme utvrđuje se snimanjem kronometrom ili izračunavanjem pomoću formula.

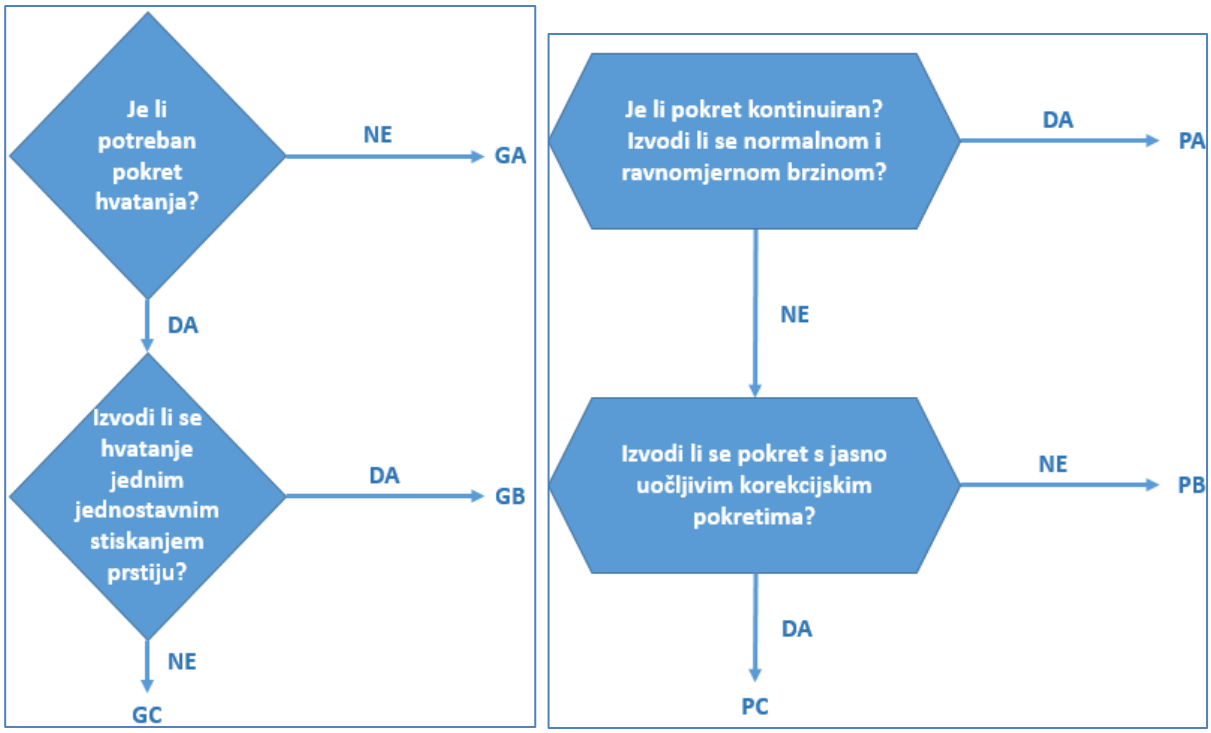
Ovisno o proizvodnim količinama i kontekstu primjene, razvijeno je više inačica sustava unaprijed određenih vremena MTM (MTM-1, MTM-2, MTM-3 i druge). Sustav MTM-2 koji će se koristiti u ovom slučaju, sastavljen je od elemenata prikazanih u tablici 5., koji predstavljaju grupe osnovnih pokreta ili pojedinačni MTM osnovni pokret.

Pojedini elementi iz tablice 5. ne mogu se ostvariti u virtualnom okruženju. Primjerice, radnja pritiskivanja koja znači primjenu snage mišića bilo kojim dijelom tijela, sadrži radnje stvaranja sile, minimalnog vremena zadržavanje sile i otpuštanje sile, što nije realno ostvarivo bez korištenja haptičkih rukavica. Također, pokret stopala i korak u ovom slučaju su ograničeni duljinom žica koje spajaju popratnu opremu VR s računalom, pa je u svakom slučaju u budućnosti cilj ostvariti bežičnu komunikaciju između opreme VR i računala.

Tablica 5. Elementi sustava MTM-2 [43]

Element	Izvorni naziv	Simbol
1.Uzimanje	*Get*	GA,GB,GC
2.Dodatak za težinu kod uzimanja	*Get weight*	GW
3.Postavljanje	*Put*	PA, PB, PC
4.Dodatak za težinu kod postavljanja	*Put weight*	PW
5.Ponovno hvatanje	*Regrasp*	R
6.Pritiskivanje	*Apply pressure*	A
7.Okretanje ručice	*Crank*	C
8.Pokret očiju	*Eye action*	E
9.Pokret stopala	*Foot motion*	F
10.Korak	*Step*	S
11.Saginjanje i uspravljanje	*Bend and raise*	B

Slika 73. daje modele odlučivanja za pravilno određivanje slučaja uzimanja i postavljanja.



Slika 73. Modeli odlučivanja za: uzimanje (lijevo) i postavljanje (desno) [43]

Važno je napomenuti kako se kod MTM postupaka vrijeme mjeri u TMU[34], gdje je:

1 TMU = 1/100 000 sati, odnosno 1 TMU = 0,036 sekunda.

Tablica 6. daje normativne vrijednosti sustava MTM-2.

Tablica 6. Normativne vrijednosti sustava MTM-2 [43]

M T M - 2						
GA	**GB**	**GC**	**OZNAKA**	**PA**	**PB**	**PC**
3	7	14	**5**	3	10	21
6	10	19	**15**	6	15	26
9	14	23	**30**	11	19	30
13	18	27	**45**	15	24	36
17	23	32	**80**	20	30	41
GW: 1 TMU/kg				**PW**: 1 TMU/(5 kg)		
R	**A**	**C**	**E**	**F**	**S**	**B**
6	14	15	7	9	18	61

[34] engl. *Time Measurement Unit*

Veliki utjecaj na određivanja vremena trajanja pojedine radnje imat će duljina pokreta koji je potrebno izvršiti. Obzirom na duljinu, prema MTM-2 postupku pokreti su podijeljeni u pet razreda (Tablica 7.).

Tablica 7. Podjela duljine pokreta u sustavu MTM-2 [43]

Udaljenost [cm]		**Oznaka**
Više od	Zaključno sa	
0	5	5
5	15	15
15	30	30
30	45	45
45	-	80

Tako će primjerice pokret uzimanja predmeta koji se nalazi na udaljenosti većoj od 15 cm, a manjoj od 30 cm dobiti oznaku GA30, GB30 ili GC30, ovisno o modelu odlučivanja (Slika 73., lijevo). Prema sustavu MTM-2, trajanje radnje uzimanja u tom će slučaju za radnju GA30 iznositi 9 TMU, za radnju GB30 14 TMU te za radnju GC30 27 TMU, gdje su podaci očitani iz tablice 6. Međusobne udaljenosti komponenata i njihova pravovaljana orijentacija te općenito prostorni raspored elemenata sustava uvelike će utjecati na brzinu izvršenja zadatka sklapanja i u virtualnom okruženju.

Za potpuno opisivanje postupka ručne montaže zidne priključnice koristit će se tablice 6. i 7., kako bi se u potpunosti analizirao rad. Prvo će se raščlaniti pokreti lijeve i desne ruke na osnovne pokrete, gdje će se svakom osnovnom pokretu unaprijed dodijeliti standardno vrijeme, kao što je opisano u prethodnom primjeru, prema vrsti pokreta i uvjetu pod kojim se on izvodi. Analiza rada za primjer sklapanja zidne priključnice za element rada 7 (ER7) dana je tablicama 8 i 9.

Tablica 8. Analiza rada putem MTM-2 postupka, List 1/2

FSB Fakultet strojarstva i brodogradnje, Zagreb	**MTM-2 LIST ZA ANALIZU RADA**	Oznaka: ER7

Naziv	ZAVRŠNA MONTAŽA VANJSKOG KUĆIŠTA – GOTOVI PROIZVOD	
Početak	Desnom rukom prema Sklopu 5	
Sadržaj	Postavljanje pozicije 1 u poziciju 2, te utiskivanje pozicija 1 i 2 na Sklop 5. Uzimanje pozicije 3 (vijka), postavljanje u središnji provrt pozicije 1, uvrtanje te pritezanje vijčalom.	
Završetak	Odložen proizvod	
Ograničenje	Vrijedi za jedan proizvod	

Opis rada lijeve ruke	*f*	Simbol	TMU	Simbol	*f*	Opis rada desne ruke
			14	GB30		Uzimanje Sklopa 5
			19	PC30		Postavljanje Sklopa 5
Pokret blizu kutije - pozicija 2		G-	23	GC30		Uzimanje pozicije 1
Uzimanje pozicije 2		GC5	14			
			30	PC30		Postavljanje poz1 u poz2
Sklop 1+2 na Sklop 5		PC15	26			
Pritiskivanje		A	14			
			23	GC30		Uzimanje vijka
			6	R		Ponovno hvatanje
			30	PC30		Prijenos vijka do provrta
			6	2PA5		Traženje početka navoja
			14	2GB5		Obuhvaćanje prstima
			6	2PA5		Uvrtanje
			14	GB30		Uzimanje vijčala
			6	R		Ponovno hvatanje

Napomena:

Standardno vrijeme:

Izradio:	Datum:	Odobrio:	Datum:	List:	Listova:
Domagoj Buzjak	2017-07-11	Zoran Kunica		1	2

Tablica 9. Analiza rada putem MTM-2 postupka , List 2/2

FSB Fakultet strojarstva i brodogradnje, Zagreb	**MTM-2 LIST ZA ANALIZU RADA**					Oznaka: ER7
Naziv	ZAVRŠNA MONTAŽA VANJSKOG KUĆIŠTA – GOTOVI PROIZVOD					
Početak	Desnom rukom prema Sklopu 5					
Sadržaj	Postavljanje pozicije 1 u poziciju 2, te utiskivanje pozicija 1 i 2 na Sklop 5. Uzimanje pozicije 3 (vijka), postavljanje u središnji provrt pozicije 1, uvrtanje te pritezanje vijčalom.					
Završetak	Odložen proizvod					
Ograničenje	Vrijedi za jedan proizvod					
Opis rada lijeve ruke		*f*	Simbol	TMU	Simbol	*f*	Opis rada desne ruke
				30	PC30		Postavljanje na glavu vijka
				70	10GB5		Obuhvaćanje
				30	10PA5		Uvrtanje
				14	A		Pritezanje
				11	PA30		Odlaganje vijčala
				Σ=400			
Napomena:							
Standardno vrijeme (neto): 400 TMU							
Izradio: Domagoj Buzjak	Datum: 2017-07-11	Odobrio: Zoran Kunica	Datum:	List: 2	Listova: 2		

Dobiveno **vrijeme rada korištenjem tradicionalnog planiranja** procesa putem sustava MTM-2 **iznosi 400 TMU, što odgovara trajanju od 14,40 sekundi**.

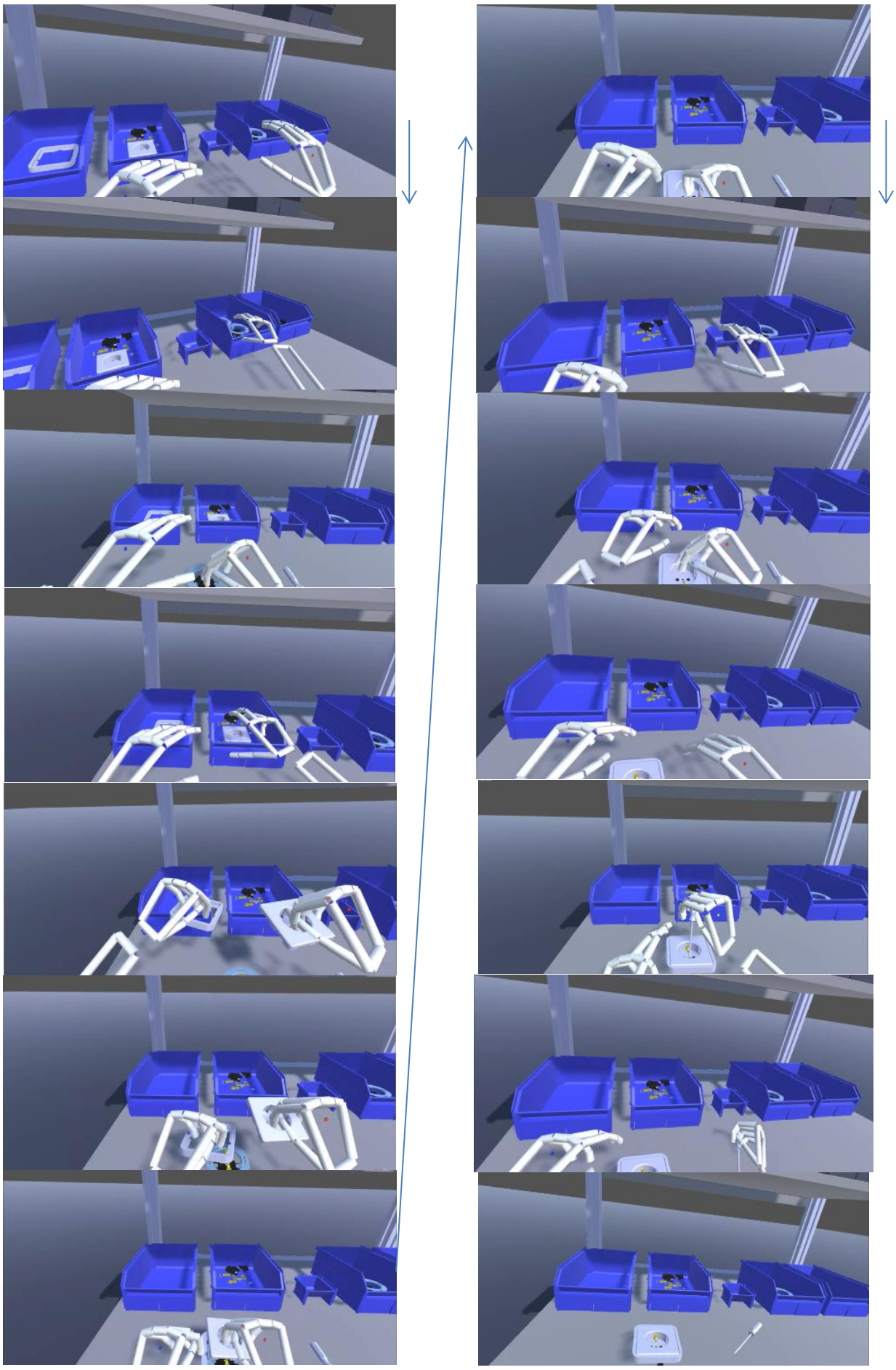
4.2.5. Virtualno sklapanje

Izvršavanje zadatka sklapanja u virtualnom okruženju pokušat će se izvršiti što vjernijom replikacijom analiziranog rada koji je naveden u tablicama 8 i 9. Dakako, virtualno okruženje neće omogućiti potpunu replikaciju svih radnji, kao što su primjerice traženje početka navoja, uvrtanje i pritezanje vijka, gdje se za te radnje naknadno dodaje tehnološko vrijeme potrebno za izvršenje operacije[35].

Za usporedbu, kao što je već spomenuto, koristit će se brojač vremena unutar virtualnog okruženja, kako bi se mogla usporediti dobivena vremena s tradicionalnim postupkom planiranja koji je izvršen. Važno je naglasiti kako i unutar samog virtualnog okruženja pravovaljana orijentacija komponenti te njihova udaljenost, u odnosu na korisnika koji će vršiti sklapanje, igra važnu ulogu na vrijeme trajanja sklapanja. Vrlo je bitno da su radne komponente smještene dovoljno blizu korisnika kako pokreti ne bi bili nepotrebno predugi, odnosno kako bi odgovarali udaljenostima između 15 cm i 30 cm kao što je prikazano u analizi rada.

Sklapanje u virtualnom okruženju izvršit će se ponovno, ali ovoga puta u cilju što vjernije replikacije optimiranog procesa dobivenog analizom rada sustavom MTM-2. Ovaj postupak razlikovat će se od prijašnjeg gdje je sklapanje bilo izvedeno u potpunosti intuitivno, bez uvježbanosti i slijeđenja pravila definiranih analizom rada. Zadatak će se pokušati izvršiti u što kraćem vremenu.

[35] Tehnološko vrijeme za izvršenje pojedine operacije neće se razmatrati u sklopu ovog rada.



Slika 74. Serija slika iz video isječka virtualnog sklapanja zidne priključnice

Nakon što je zadatak izvršen (Slika 74.), zabilježeno je vrijeme izvršenja pojedinih radnji za element rada 7 uz prikaz dobivenih rezultata u tablici 10.

Tablica 10. Mjerenje trajanja imerzivnog procesa sklapanja, element rada 7.

Radnja	**Trajanje, s**
Uzimanje Sklopa 5 i njegovo pozicioniranje[36]	1,83
Dvoruki rad, uzimanje vanjskog kućišta (pozicija 1) i okvira vanjskog kućišta (pozicija 2), njihovo međusobno pozicioniranje i zajedničko pozicioniranje u odnosu na Sklop 5	3,70
Uzimanje vijka i njegovo pozicioniranje u odnosu na poziciju 1, poziciju 2 i Sklop 5	3,34
Uzimanje vijčala i pozicioniranje na glavu vijka	2,16
Odlaganje vijčala	0,67
Ukupno trajanje:	**11,70[37]**

4.2.6. Usporedba rezultata dobivenih sustavom MTM i virtualnim sklapanjem

Ukupno vrijeme trajanje procesa sklapanja zidne priključnice u virtualnom okruženju iznosi 11,70 sekundi. Za usporedbu, vrijeme sklapanja potpuno istovjetnog procesa kod tradicionalnog planiranja gdje je korištena tehnika MTM-2 procijenjeno je na 400 TMU, što je jednako vremenu trajanja od 14,40 sekundi. Sklapanje u virtualnom okruženju je dakle izvršeno za 2,70 sekundi manje, što znači da je **proces izvršen okvirno 19 % brže u odnosu na** vrijeme predviđeno za izvršavanje radnje sklapanja putem **sustava unaprijed definiranih vremena.** Iz tablice 10. i prikaza rezultata trajanja pojedinih radnji može se zaključiti sljedeće:

- **Kraće vrijeme trajanja** izvršavanja zadatka sklapanja u virtualnom okruženju može na neki način biti i **očekivano** zbog činjenice kako se **proces sklapanja** slijedeći radnje iz analize rada **nije mogao u potpunosti replicirati** zbog manjkavosti virtualnog

[36] Zbog pojednostavnjenja i velikog broja komponenti koje sadrži Sklop 5, u okruženju virtualne stvarnosti izveden je prazan pokret.

[37] Prema ISO 8601 za rastavljanje vremena na dijelove mogu se koristiti decimalna točka ili decimalni zarez, gdje se prednost daje decimalnom zarezu prema ISO 8601:2004. [51]

okruženja u pogledu teškog ostvarivanja pojedinih radnji poput uvrtanja i pritezanja vijka zbog nedostatka povratne veze sile.

- Također, **kod sitnijih dijelova**, kao što je primjerice vijak, **mnogo je teže izvršiti hvatanje** s obzirom na točnost i toleranciju kontrolera Leap Motion. Naime, kod hvatanja manjih dijelova poput vijka iz primjera koji ima promjer svega 3,5 mm, mnogo je teže točno izvršiti zadatak zbog ograničenja kontrolera Leap Motion koji **pri tako finim pozicioniranjima može odstupati od željenog ponašanja i tolerancija**, iako bi trebao biti postojan na pod milimetarskom području. Tako uzimanje vijka često zahtijeva **radnje ponovnog hvatanja (višestrukog zbog neuspješnosti ili zbog neželjenog ispuštanja)** što rezultira sa **znatno više utrošenog vremena** nego da se sklapanje vrši u realnom svijetu. Razlika u vremenu izvršavanja zadatka stoga je nastupila kod radnji traženja početka navoja, obuhvaćanja i uvrtanja vijka koje u virtualnom okruženju s postojećom opremom nisu ostvarive a rezultirale su kraćim izvođenjem procesa sklapanja nego u stvarnosti.
- Izvršavanje zadatka sklapanja u imerzivnom virtualnom okruženju **ne zahtijeva predugo učenje i stvara intuitivno okruženje** koje može poslužiti za stjecanje iskustva i stimulaciju rada, te njegovo lakše izvršavanje u realnosti. S obzirom na činjenicu kako zidna priključnica nije sklop koji se sastoji od kompleksne geometrije, posebnih materijala ili posebno teškog zadatka sklapanja za montažere, prednost ovakvog pristupa možda nije vidljiva u potpunosti. Virtualni trening rezultirao bi **znatno većim uštedama kod kompleksnijih sklopova** gdje se zahtijevaju naprednije radnje sklapanja i gdje se javlja mogućnost eventualnog oštećivanja vrlo vrijednih komponenti ukoliko montažer ne posjeduje određeno predznanje.

Nadalje, iz ovog poglavlja mogu se izvući i neki općenitiji zaključci vezani uz projektiranje proizvodnih procesa tehnikama virtualne stvarnosti.

Prednosti:

- Virtualno okruženje stvara **intuitivan prostor** u kojem je **lako snalaženje korisnika**
- Korisnik u vrlo **kratkom vremenu**, u roku svega par minuta, može **naučiti koristi uređaj** virtualne stvarnosti i kontroler Leap Motion

- **Izvršavanje zadataka sklapanja u virtualnom okruženju** najčešće **će biti kraće** od sklapanja u stvarnosti
- Izgledne su velike **uštede u projektiranju** zbog lakog definiranja okoline i radnih zadataka
- Pravilnim pozicioniranjem unutar okruženja omogućena je **odlična vizualizacija geometrije iz bilo kojeg kuta gledišta**
- Osim realne vizualizacije, kao velika prednost ovog sustava ističe se **mogućnost interakcije** sa svim vrstama dijelova bez obzira na veličinu
- **Upoznavanje** s proizvodom **prije izrade fizičkog prototipa**
- Nakon što su dijelovi spremljeni u odgovarajućem formatu, vrlo **brzo se može definirati scena** za različitu geometriju
- Moguće je **izvršavanje raznolikih zadataka**, zbog fleksibilnosti sustava
- **Sklapanje** se može vršiti na **različitim lokacijama** (naprimjer kod kuće)
- Lako ostvarivanje **virtualnih treninga** za različite djelatnosti
- Sustav **je pogodan za provedbu ergonomskih istraživanja** (biometrika)
- Virtualno okruženje je **sigurno okruženje bez opasnosti od mogućih ozljeda** djelatnika i oštećenja radnih dijelova.

Nedostaci:

- Korisnik može osjetiti laganu **mučninu i nelagodu** kod korištenja uređaja
- Za svaki poseban zadatak **potrebno je nanovo podesiti scenu**
- **Potrebna konverzija formata** CAD geometrije u format pogodan za razvojnu platformu VR
- **Nemogućnost ostvarivanja povratne veze sile** bez skupih haptičkih uređaja
- **Ograničenost kretanja** korisnika zbog geometrijskih značajki opreme
- Mogućnost **pogrešne detekcije** korisnikovih ruku
- **Pregrijavanje uređaja** Leap Motion.

4.3. JEDNORUKI I DVORUKI RAD

Korištenje softvera Orion s kontrolerom Leap Motion **omogućuje rad s obje korisnikove ruke**. U istom trenu moguće je posezanje za više različitih dijelova unutar virtualnog okruženja, te istodobno izvršavanje njihovog pomicanja ili drugih potrebnih radnji. Istovremeni rad s obje ruke vidljiv je na videu na kojem je prikazan proces sklapanja zidne priključnice, a koji je dostupan kao Prilog ovom radu. Što se tiče **jednoruke interakcije** i interakcije pojedinog prsta ruke, *Interaction* manager **ne omogućuje interakciju s jednim korisnikovim prstom samo jedne ruke.** S dva prsta jedne ruke moguće je hvatanje i pomicanje dijelova. Isto je moguće i s jednim prstom lijeve i jednim prstom desne ruke. **Za radnje hvatanja ili pomicanja nema razlike koristi li korisnik dva, tri i četiri prsta, ili cijelu ruku.**

4.4. TIMSKI RAD

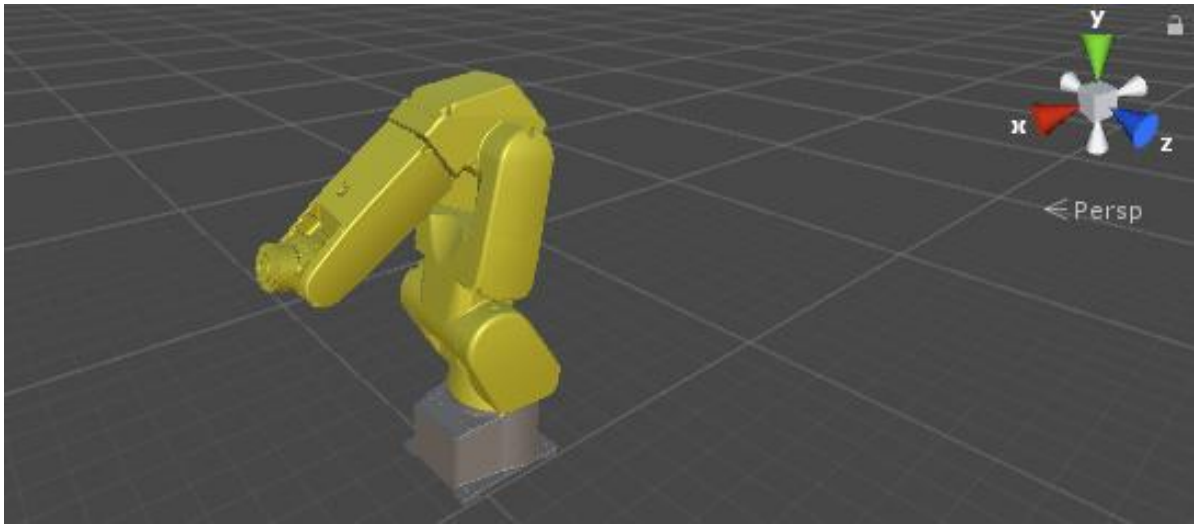
Timski rad koji bi uključivao korištenje više od jednog kontrolera Leap Motion nameće se kao dobra **ideja kako bi se poništila ograničenja** koja se javljaju kod korištenja kontrolera Leap Motion. Naime, u trenutku kada se između kamere nalazi prepreka poput žice (naprimjer iz uređaja VR) **moguće je da ne dođe do detekcije ruku korisnika**, zbog toga jer blokadom bilo koje kamere kontroler ne može percipirati dubinu[38]. Nadalje, ako se teoretski nalazi **više od dviju ruku u vidnom polju** kontrolera, često dolazi do pogrešne detekcije korisnikovih ruku, tj. **moguće je netočno prepoznavanje ruku**. Najveći problem dakako je situacija u kojoj dolazi do **preklapanja korisnikovih prstiju.** U tom trenutku, ukoliko se primjerice jedna ruka nalazi ispod druge ruke, logično je da kontroler neće moći percipirati **donju ruku**, te **ona neće biti detektirana**. Zbog navedenih problema kao logičan zaključak **nameće se ideja korištenja više kontrolera Leap Motion unutar zajedničke scene.** To bi omogućilo **veće vidno polje uređaja**, ali i eventualni **timski rad više korisnika** te njihovo **zajedničko izvršavanje zadataka** u virtualnom okruženju. Korištenje više kontrolera Leap Motion na

[38] Slično kao kod čovjeka, koji ne može percipirati dubinu ako zatvori jedno oko

jednom korisničkom računalu unutar softvera Unity **u ovom trenutku nije moguće bez** određenih **modifikacija** koje zahtijevaju stvaranje virtualnog računala.

4.5. RAD ČOVJEK-ROBOT

Zbog ugrađene fizike Unity se može koristiti primjerice i u području robotike. U početku je potrebno CAD model robota ubaciti u Unity, gdje je važno da robot posjeduje sve konstrukcijske značajke iz CAD-a, odnosno da se ne konvertira tzv. mrtva geometrija koja ne sadrži međusobne zakonitosti između pojedinih zglobova robota. Modeliranje robota unutar Unityja također je moguće, međutim zbog kompleksne geometrije korištenje CAD softvera brže je i jednostavnije za izvedbu.



Slika 75. Robot Fanuc LR-Mate 200iC

Kada je robot uspješno ubačen u Unity potrebno je definirati ograničenja i odnose između pojedinih zglobova robota preko ugrađenih opcija *Joints*. Nakon što su zglobovi dovedeni u međusoban odnos, potrebno je pristupiti rješavanju kinematičkog problema. Kinematički problem robota ne bavi se uzrocima gibanja, već proučava geometriju gibanja koju robot treba izvršiti da bi se pozicionirao u prostoru. Rješavanju kinematičkog problema može se pristupiti na direktan ili inverzan način. Kod direktnog kinematičkog problema polazi se od poznatih položaja zglobova robota u prostoru, čije su informacije dobivene na temelju

senzora na robotu, a cilj je saznati poziciju prihvatnice ili točke u prostoru. Kod inverznog kinematičkog problema pristupa se sa suprotnog stajališta, odnosno, polazi se od poznate pozicija prihvatnice ili točke u prostoru, dok nisu poznati položaji zglobova robota. Pristup rješavanju kinematičkog problema unutar Unityja uključuje korištenje naprednog programiranja u C# jeziku. Kada su međusobno točno definirane zakonitosti između pojedinih zglobova robota, model robota u Unityju može ostvariti komunikaciju sa stvarnim robotom preko TCP/IP[39] protokola. Navedeni princip koristio bi se za direktno učenje robota, orijentirano gibanju, gdje bi robot mogao vršiti replikaciju gibanja koje vrši čovjek korištenjem kontrolera Leap Motion. Pristup je u potpunosti sukladan postupku upravljanja robota putem sustava Microsoft Kinect, na čiju temu je napravljeno više radova na Fakultetu strojarstva i brodogradnje u Zagrebu (naprimjer, [45]). Ovakav pristup upravljanju posebice bi bio pogodan za primjenu u područja učenja robota gdje bi se izbjegla potreba za programiranjem putem principa indirektnog učenja pomoću privjeska za učenje, koji kod pojedinih proizvođača zahtijeva dosta vremena za upoznavanjem sa svim funkcijama. Međutim, navedeni pristup zbog brojnih ograničenja najvjerojatnije ne bi bio odmah industrijski primjenjiv.

Bez obzira na gornje, očit je potencijal koji virtualna stvarnost pruža za projektiranje hibridnih i interaktivnih sustava čovjek-stroj. Naročita prednost interakcije čovjek – robot bila bi vidljiva u pojedinim radnjama određenih proizvodnih zadataka gdje je potrebno kombinirati ljudski rad (koji je kod specifičnih radnji ponekad isplativiji od robota) i rad robota. Kod takvih situacija, robot bi izvršavao zadatke u kojima postiže bolje rezultate od ljudi, dok bi čovjek izvršavao zadatke kod kojih je neisplativo koristiti robota (primjerice zbog kompleksne i promjenjive radne okoline te velikih troškova). Takav rad rezultirao bi većom fleksibilnošću i produktivnošću.

[39] engl. TCP/IP – Transmission Control Protocol / Internet Protocol

5. ZAKLJUČAK

Virtualna stvarnost ima sve veći značaj i primjenu u raznolikim područjima ljudskih djelatnosti, pa tako i u području strojarstva. Trenutačna korist virtualne stvarnosti ponajviše je vidljiva kod vizualizacije proizvoda, provedbe ergonomskih istraživanja, marketinških aktivnosti, oblikovanju značajki proizvoda prema specifičnim željama korisnika i slično. Osim navedenih područja, zbog cjenovne dostupnosti opreme i nedavne komercijalizacije uređaja virtualne stvarnosti, vidljivo je kako je bez prevelikih ulaganja moguće vršiti simulaciju proizvodnih postrojenja, pogona i procesa te izvršavati različite proizvodne zadatke unutar stvorenog okruženja. Oprema koja je u ovom radu uključivala zaslon montiran na glavu, senzor za detekciju pokreta te popratni softver, poslužila je za stvaranje naprednog virtualnog okruženja, u kojem su izvršeni proizvodni zadaci umetanja osovine u provrt i sklapanja zidne priključnice. U stvorenom imerzivnom okruženju, izvedeno virtualno sklapanje provjereno je i uspoređeno s tradicionalnim načinom projektiranja radnih odnosno proizvodnih procesa.

Za primjer sklapanja zidne priključnice napravljena je analiza rada tradicionalnim načinom putem sustava unaprijed određenih vremena MTM-2, određeni su pokreti i njihova vremena za odvijanje radnji sklapanja odabranog elementa rada, te je izračunato ukupno vrijeme trajanja procesa ručne montaže. U virtualnom okruženju potom je izvršeno sklapanje istih dijelova zidne priključnice, i to slijedeći analizu rada (pokrete) napravljenu tradicionalnim pristupom. Sklapanje u virtualnom okruženju bilo je 19 % brže od sklapanja po sustavu MTM-2. Razlika u vremenu posljedica je ograničenja dostupne opreme VR: radnje poput traženja početka navoja, uvrtanja i pritezanja vijka nisu mogle biti vjerno replicirane zbog nedostatka povratne veze sile koja bi se mogla ostvariti korištenjem haptičkih rukavica. Korištenjem haptičkih rukavica najvjerojatnije bi došlo do poklapanja vremena sklapanja.

Dakle, imerzivnim projektiranjem došlo se do vrlo kvalitetnih rezultata te su ustanovljene značajne pogodnosti (uštede) korištenja virtualnog okruženja za projektiranje. Neke od prednosti ovakvog pristupa jesu: sigurno radno okruženje bez opasnosti od ozljeda,

intuitivno okruženje koje ne zahtijeva dugotrajno učenje, upoznavanje s radnom okolinom, upoznavanje dijelova prije izrade fizičkog prototipa, izvršavanje virtualnih treninga sklapanja bez obziru na lokaciju na kojoj se nalazi radnik te učenje sklapanja prije no što su dijelovi uopće proizvedeni.

Virtualno okruženje posjeduje i određene nedostatke. Kako bi se stvorilo željeno okruženje, trenutačno je potrebna pretvorba CAD modela u format pogodan za čitanje u softveru VR, što za veći broj dijelova rezultira većim utroškom vremena što je ujedno i glavni nedostatak ovog pristupa. Nadalje, u rijetkim slučajevima korištenje zaslona montiranog na glavu može imati posljedice na zdravlje pojedinca, gdje su simptomi slični kao kod bolesti kretanja, tzv. kinetoze, a očituju se kao mučnina i nelagoda. S napretkom tehnologije i kvalitetnijim zaslonima za glavu, štetne posljedice će se najvjerojatnije svesti na minimum. Ostala ograničenja rezultat su konstrukcijskih značajki opreme.

Korištenjem haptičkih rukavica i većeg broja kontrolera Leap Motion ograničenja postojeće opreme mogla bi se gotovo u potpunosti poništiti, što bi rezultiralo vjernijom replikacijom proizvodnih procesa i njihovim daljnjim poboljšanjem. Još veći napredak u ovom području ostvarit će se u trenutku kada vodeći proizvođači CAD programa u svoj softver uključe modul virtualne stvarnosti gdje više neće biti potrebna pretvorba formata već će oprema VR moći ostvariti izravnu komunikaciju sa CAD softverom. To će rezultirati lakšim virtualnim sklapanjem, boljim prikazom konstrukcija i proizvoda (općenito, boljom vizualizacijom) te poboljšanim prikazom ostalih značajkih koje su sastavni dio CAD programa. Nadalje, područje međusobne interakcije između čovjeka i robota također otvara brojne mogućnosti za budući rad.

Tehnike virtualne stvarnosti predstavljaju dakle moderan i prije svega kvalitetan pristup rješavanju problema koji pruža brojne prednosti u odnosu na tradicionalan pristup. S protjecanjem vremena, područje računalno posredovane stvarnosti doživjet će daljnju ekspanziju. Naročit pomak mogao bi se dogoditi kada uređaji miješane stvarnosti dožive komercijalizaciju. To bi moglo rezultirati temeljitom promjenom trenutačkog načina života ljudi, ali i uvođenjem do sad neviđenih pristupa rješavanju različitih vrsta problema.

6. LITERATURA

[1] http://www.3ders.org/articles/20150730-ford-reduces-assembly-line-injury-rate-3d-printed-simulations-and-virtual-manufacturing.html, Pristupljeno 2017-07-08.

[2] Buzjak, D., Virtualna stvarnost i njezine primjene -Završni rad, FSB Zagreb, 2016.

[3] https://en.wikipedia.org/wiki/Virtual_reality, Pristupljeno 2017-03-02.

[4] Hamid, N. S. S., Aziz, F. A. i Azizi,A., Virtual Reality Applications in Manufacturing System, *Science and Information Conference*, 2014..

[5] https://en.wikipedia.org/wiki/Oculus_Rift, Pristupljeno 2017-04-19.

[6] http://fortune.com/2017/02/21/microsoft-hololens-update-delay/, Pristupljeno 2017-03-01.

[7] https://en.wikipedia.org/wiki/Immersion_(virtual_reality), Pristupljeno 2017-05-08.

[8] https://en.wikipedia.org/wiki/Immersive_technology, Pristupljeno 2017-05-08.

[9] https://en.wikipedia.org/wiki/Google_Cardboard, Pristupljeno 2017-01-17.

[10] https://recombu.com/mobile/article/recombu-vr-goggles-vs-google-cardboard_M20570.html, Pristupljeno 2017-03-17.

[11] https://store.google.com/product/google_cardboard, Pristupljeno 2017-03-17.

[12] https://en.wikipedia.org/wiki/Oculus_VR, Pristupljeno 2017-04-19.

[13] https://en.wikipedia.org/wiki/Samsung_Gear_VR, Pristupljeno 2017-04-19.

[14] http://spectrum.ieee.org/telecom/internet/beyond-second-life-philip-rosedales-gutsy-plan-for-a-new-virtualreality-empire, Pristupljeno 2017-03-02.

[15] http://www.goldmansachs.com/our-thinking/pages/virtual-and-augmented-reality.html, Pristupljeno 2017-03-02.

[16] https://en.wikipedia.org/wiki/Augmented_reality, Pristupljeno 2017-05-10.

[17] https://en.wikipedia.org/wiki/Mixed_reality, Pristupljeno 2017-05-15.

[18] https://en.wikipedia.org/wiki/Computer-mediated_reality, Pristupljeno 2017-05-10.

[19] https://en.wikipedia.org/wiki/Virtual_reality_sickness, Pristupljeno 2017-05-08.

[20] https://en.wikipedia.org/wiki/Motion_sickness, Pristupljeno 2017-05-08.

[21] https://en.wikipedia.org/wiki/Unity_(game_engine), Pristupljeno 2017-02-12.

[22] https://docs.unity3d.com/Manual/3D-formats.html, Pristupljeno 2017-02-12.

[23] https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/3DSMax/files/GUID-566E59EE-8221-4AC6-824B-5062C5AE0B32-htm.html, Pristupljeno 2017-02-15

[24] https://en.wikipedia.org/wiki/CATIA , Pristupljeno 2017-02-12.

[25] https://grabcad.com/library/tag/crankshaft, Pristupljeno 2017-02-12.

[26] https://unity3d.com/unity/multiplatform, Pristupljeno 2017-02-12.

[27] https://developers.google.com/vr/unity/get-started, Pristupljeno 2017-02-17.

[28] https://developers.google.com/vr/unity/, Pristupljeno 2017-02-17.

[29] https://docs.unity3d.com/Manual/android-sdksetup.html, Pristupljeno 2017-02-20.

[30] https://developers.google.com/vr/unity/plugin#gvrmain, Pristupljeno 2017-02-20.

[31] https://en.wikipedia.org/wiki/C_Sharp_(programming_language), Pristupljeno 2017-03-15.

[32] https://en.wikipedia.org/wiki/Microsoft_Visual_Studio, Pristupljeno 2017-03-15.

[33] https://github.com/JuppOtto/Autowalk-for-Google-VR-SDK-with-Unity, Pristupljeno 2017-03-19.

[34] https://en.wikipedia.org/wiki/Leap_Motion, Pristupljeno 2017-03-28.

[35] https://www.researchgate.net/profile/Salihu_Oladimeji/publication/266614710/figure/fig1/AS:295744327176192@1447522350024/Fig-1-Schematic-view-of-leap-motion-controller-LMC.png, Pristupljeno 2017-03-28.

[36] https://www3.oculus.com/en-us/dk2/, Pristupljeno 2017-04-21.

[37] https://support.oculus.com/1267582579925778, Pristupljeno 2017-04-21.

[38] https://developer.leapmotion.com/documentation/unity/index.html, Pristupljeno 2017-04-27.

[39] https://developer.leapmotion.com/orion/#105, Pristupljeno 2017-04-21.

[40] https://docs.unity3d.com/Manual/, Pristupljeno 2017-04-21.

[41] Bruyninckx, H., Dutre, S. i Schutter, J. D. »Peg-on-Hole: A Model Based Solution to Peg and Hole Alignment,« u *IEEE International Conference on Robotics and Automation*, Nagoya, Japan, 1995.

[42] Jezl, D., Završni rad, FSB, Zagreb, 2009.

[43] Kunica, Z., http://titan.fsb.hr/~zkunica/nastava/ZiM. FSB, Zagreb - nastavni materijali, Pristupljeno 2017-05-25.

[44] Hlebec, F., Diplomski rad, FSB, Zagreb 2001..

[45] Bučević, A., Vođenje robota pokazivanjem, FSB, Zagreb 2012.

[46] http://www.ecma-international.org/memento/history.htm, Pristupljeno 2017-06-01

[47] https://www.microsoft.com/microsoft-hololens/en-us/buy, Pristupljeno 2017-03-01.

[48] https://en.wikipedia.org/wiki/HDMI, Pristupljeno 2017-03-02.

[49] https://en.wikipedia.org/wiki/Goldman_Sachs, Pristupljeno 2017-03-04

[50] https://en.wikipedia.org/wiki/DisplayPort, Pristupljeno 2017-05-15.

[51] https://en.wikipedia.org/wiki/ISO_8601, Pristupljeno 2017-06-29.