# Upravljanje robotskih ruku putem razvojnog alata UNITY

**Hanuljak, Ivan**

**Master's thesis / Diplomski rad**

**2022**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Pula / Sveučilište Jurja Dobrile u Puli**

*Permanent link / Trajna poveznica:* https://urn.nsk.hr/urn:nbn:hr:137:154329

*Rights / Prava:* In copyright / Zaštićeno autorskim pravom.

*Download date / Datum preuzimanja:* **2024-04-17**

*Repository / Repozitorij:*

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli

Fakultet informatike

Ivan Hanuljak

# Upravljanje robotskih ruku putem razvojnog alata UNITY

Diplomski rad

Pula, 2022.

Sveučilište Jurja Dobrile u Puli

Fakultet informatike

Ivan Hanuljak

# Upravljanje robotskih ruku putem razvojnog alata UNITY

Diplomski rad

JMBAG: 0303061325, redovni student

Studijski smjer: Sveučilišni diplomski studij informatike, nastavni smjer

Predmet: Računalna grafika

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: izv. prof. dr. sc. Sven Maričić

Pula, 2022.

Fakultet informatike u Puli

Ime i prezime studenta/ice Ivan Hanuljak

JMBAG 0303061325 (4282-E)

Status: ☒ redoviti ☐ izvanredni

**PRIJAVA TEME DIPLOMSKOG RADA**

Izv. prof. dr. sc. Sven Maričić

Ime i prezime mentora

Nastavni smjer informatike

Studij

Računalna grafika

Kolegij

Potvrđujem da sam prihvatio/la temu završnog/diplomskog rada pod naslovom:

Upravljanje robotskih ruku putem razvojnog alata UNITY

(na hrvatskom jeziku)

Robotic arm control via UNITY platform

(na engleskom jeziku)

Datum: 21.2.2022.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani _____, kandidat za magistra _____ ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljeni način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, _____

IZJAVA O KORIŠTENJU AUTORSKOG DJELA

Ja, _____ dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom _____ _____ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama. Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____

Potpis

_____

ZAHVALA

Ovim putem se zahvaljujem mentoru izv. prof. dr. sc. Svenu Maričiću i asistentu mag. ing. mech. Ivanu Veljoviću na omogućenoj opremi vezanoj za rad, osiguranim radnim stolom u njihovom uredu, te savjetima i odgovorima na sva moja postavljena pitanja.

Sve aktivnosti vezane uz izradu modela i 3D modeliranje izvršene su unutar projekta ERASMUS+ PROJECT 3D and Virtual Reality Technologies for VET, projektni broj: 2019-1-HR01-KA202-061006.

Također bi se zahvalio svojim prijateljima koji su mi pravili društvo svih ovih godina. Najveće hvala mojoj obitelji i djevojci koji su mi bili potpora i bez kojih ništa od ovoga ne bi bilo moguće.

Tata, nadam se da se ponosiš sa mnom.

SADRŽAJ


| | |
|---|---|
| UVOD | 1 |
| RAČUNALNA GRAFIKA | 1 |
| 3D i 2D RAČUNALNA GRAFIKA | 4 |
| RAČUNALNA ANIMACIJA | 5 |
| RAZVOJ CAD ALATA | 5 |
| ALATI ZA RAZVOJ PROJEKATA I VIDEOIGARA | 12 |
| UNITY 3D | 13 |
| DETALJAN PRIKAZ IZRADE I RADA S ROBOTSKIM RUKAMA | 17 |
| PRIKAZ KORIŠTENIH NAREDBI, BIBLIOTEKA I KODA | 20 |
| PROGRAMSKO RJEŠENJE - DODATAK | 25 |
| ZAKLJUČAK | 28 |
| LITERATURA | 29 |
| INTERNET IZVORI | 29 |
| POPIS SLIKA | 29 |
| SAŽETAK | 31 |
| SUMMARY | 32 |

# 1. UVOD

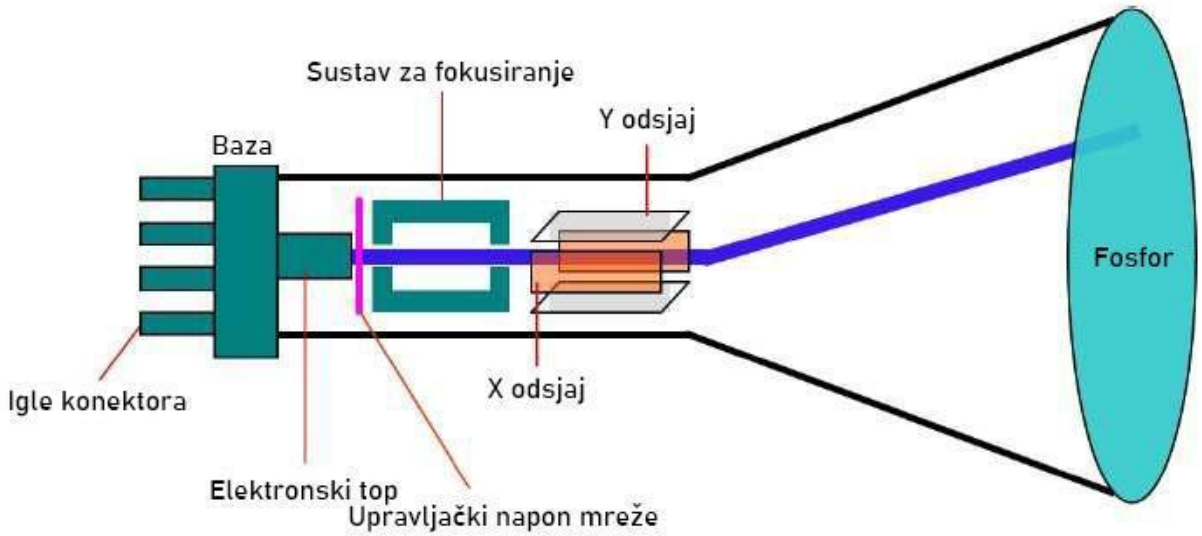
U vremenu digitalnih informacijskih tehnologija, tehnologije 3D modeliranja i računalne grafike ne primjenjuju se samo na razvoj virtualnih modela za računalne simulacije, umjetnu inteligenciju i drugo, već se mogu primijeniti i u mnogim različitim aplikacijama u virtualnoj stvarnosti (VR). Svrha rada je upoznati se s programom Unity i njegovim radom, opisati cijeli razvojni proces i računalne programe koji su se koristili. Diplomski rad ima sljedeću strukturu; u drugom odlomku će se opisati povijest nastanka i razvoja računalne grafike te objasniti što je to računalna grafika i gdje se ona sve koristi. Treći odlomak govori o 3D i 2D računalnoj grafici, dok četvrti odlomak opisuje na koje sve načine se može koristiti animacija pomoću računalne grafike. Peto poglavlje govori o CAD alatima i modeliranju te o njihovim prednostima i nedostacima. Osnovne informacije o programu Unity te njegove funkcije i svrhu korištenja opisane su šestom poglavlju. Zatim se analizira izrada i rad s robotskom rukom, odnosno prikazuje se na koji način je korišten program Unity (verzija 2020.3.32f1) pri izradi 3D modela robotske ruke. U zadnja dva poglavlja se opisuju naredbe, biblioteke i kodovi koji su se koristili pri izradi robotske ruke. U zaključku će se opisati i neke druge prednosti programa Unity te u kojim se ostalim segmentima program upotrebljava pored izrade video igara. Cilj ovog diplomskog rada je prikazati načine i metode razvoja 3D modela robotske ruke počevši od modeliranja, kodova i animacije.

# 2. RAČUNALNA GRAFIKA

Grafika se definira kao svaka skica, crtež ili posebna mreža koja slikovito predstavlja neku smislenu informaciju.[1] Razvoj računalne grafike omogućio je lakšu interakciju s računalima i bolje razumijevanje i tumačenje brojnih vrsta podataka, a imao je dubok utjecaj na različite vrste medije. Revolucionirao je animaciju i industrije filma i videoigara. Izraz računalna grafika korišten je u širokom smislu za opisivanja „gotovo svega na računalima što nije tekst ili zvuk“.[2] Pojavila se kao potpodručje računalne znanosti koje proučava metode za digitalnu sintezu i manipuliranje vizualnim sadržajem. Tijekom prošlog desetljeća razvijena su druga specijalizirana područja poput vizualizacije informacija i znanstvene vizualizacije koja se više bavi „vizualizacijom trodimenzionalnih fenomena“[3], gdje je naglasak na realističnom

[1] https://www.geeksforgeeks.org/introduction-to-computer-graphics/ (“Introduction to Computer Graphics”, pristupljeno: 12.7.2022.)
[2] https://www.explainthatstuff.com/computer-graphics.html (“Computer graphics”, pristupljeno: 12.7.2022.)
[3] https://surveyinggroup.com/what-is-3d-visualization/ (“What is 3D Visualization?”, pristupljeno: 30.8.2021.)

prikazivanju volumena, površine i izvora osvjetljenja. Danas računala i računalno generirane slike dotiču brojne aspekte svakodnevnog života, a slike generirane na računalu nalaze se na televiziji i u novinama (primjerice u vremenskim izvješćima ili u nekim vrstama medicinskih istraživanja). Koristi se tamo gdje je potrebno manipulirati skupom slika ili kreirati sliku u obliku piksela te se crta na računalu. Računalna grafika može se koristiti u digitalnoj fotografiji, filmu, zabavi, elektroničkim napravama i svim ostalim osnovnim tehnologijama koje su potrebne.[4] To je golem predmet i područje u sferi računalnih znanosti. Računalna grafika može se koristiti u dizajnu korisničkog sučelja, renderiranju, geometrijskim objektima, animaciji i još mnogo toga. Postoji nekoliko alata koji se koriste za implementaciju računalne grafike. Osnovna je datoteka zaglavlja <*graphics.h*> u Turbo-C, Unity za napredne, pa čak i OpenGL koji se može koristiti za njegovu implementaciju.

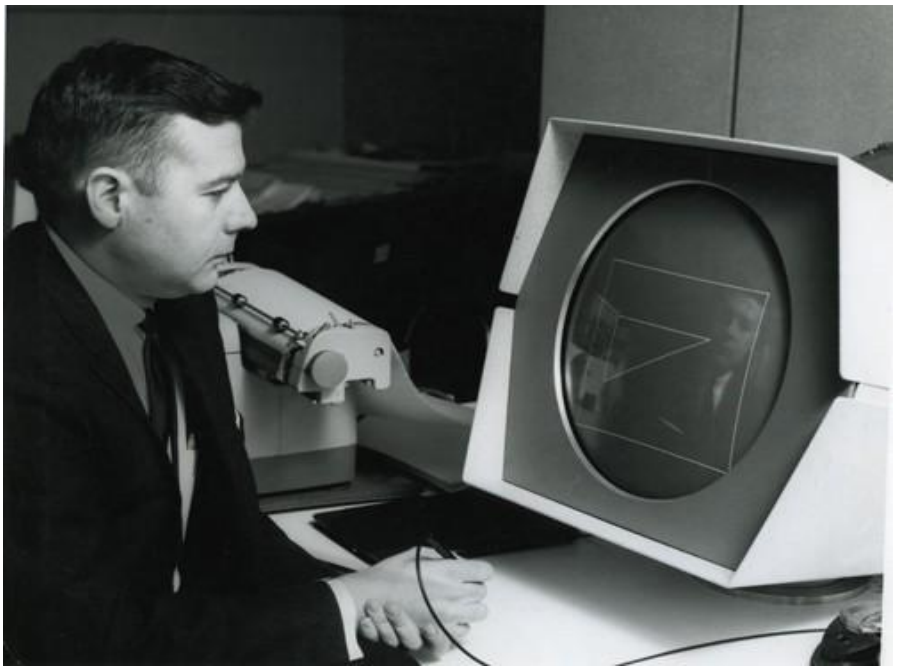



*Slika 1: CRT monitor s katodnom cijevi*[5]

Područje računalne grafike razvilo se s pojavom hardvera računalne grafike, a 1960. godine su ju izumili istraživači Verne Hudson i William Fetter iz Boeinga. Rani projekti kao što su *Whirlwind* i SAGE predstavili su CRT kao održivi zaslon i sučelje za interakciju te predstavili svjetlosnu olovku kao ulazni uređaj. Godine 1959. u MIT-ovom Lincoln Laboratoriju razvijeno je računalo TX-2, koji je integrirao niz novih sučelja čovjek - stroj. Također 60-ih godina drugi student na MIT-u, Steve Russell, stvorio je prvu video igru Spacewar. Napisan za DEC PDP-1, Spacewar je postigao trenutni uspjeh i kopije su počele pritjecati drugim

[4] https://www.tutorialspoint.com/computer_graphics/computer_graphics_quick_guide.htm („Computer Graphics - Quick Guide“, pristupljeno: 12.7.2022.)
[5] https://www.javatpoint.com/computer-graphics-cathode-ray-tube (Pristupljeno: 25.6.2022.)

vlasnicima PDP- 1, a na kraju je čak i DEC dobio kopiju. Inženjeri tvrtke DEC koristili su ga kao dijagnostički program na svakom novom PDP-1 prije isporuke.



*Slika 2: Crtanje na PDP Type 30*[6]

Područje računalne grafike je široko i raznoliko polje koje služi kao presjek između računalnih znanosti i dizajna. To je cijeli proces od stvaranja računalno generiranih slika, stvaranja digitalnih trodimenzionalnih modela, procesa teksturiranja, renderiranja, osvjetljavanja tih modela, do digitalnog prikaza tih renderiranja na ekranu. Ovaj proces započinje jednostavnim tehnikama renderiranja objekata za transformaciju matematičkih prikaza trodimenzionalnih objekata u dvodimenzionalnu sliku na ekranu. To se postiže izračunavanjem projekcijskih transformacija vrhova. Detalji i realizam dodani su kroz simulaciju materijala, tekstura i osvjetljenja. Najtočnije i najrealističnije tehnike uključuju razumijevanje načina na koji svjetlost stupa u interakciju s objektima u fizičkom svijetu. Fenomeni poput refleksije, prozirnosti ili difuzne rasvjete mogu se modelirati korištenjem niza različitih algoritama, neki dizajnirani da budu fizički točni, drugi da budu računalno učinkoviti, ovisno o različitim potrebama. Slike virtualne stvarnosti moraju se generirati u roku od nekoliko milisekundi, dok naprimjer za završetak detaljnog arhitektonskog renderiranja može trebati i nekoliko sati računanja. S razvojem hardvera GPU-a i softvera motora za prikaz, razvoj računalne grafike nastavlja pomicati granice točnosti i brzine računalno generiranih slika.

[6] https://www.computerhistory.org/collections/catalog/102652246 (Pristupljeno: 25.6.2022.)

Računalna grafika odnosi se na nekoliko stvari:

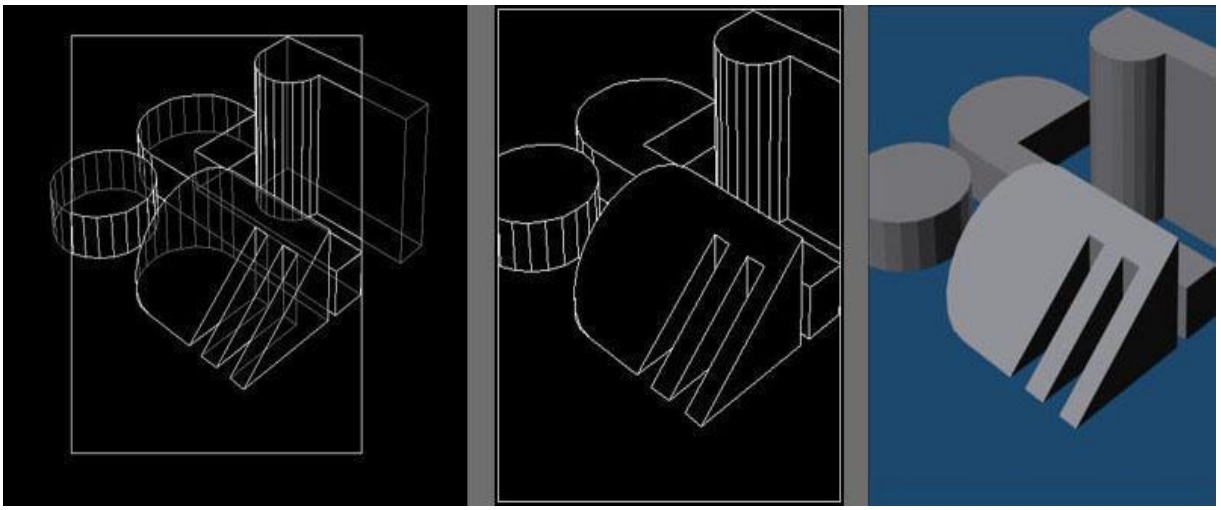
- Manipulacija i prikaz slike ili podataka na grafički način.
- Za izradu i manipulaciju potrebna je različita tehnologija.
- Digitalna sinteza i njezina manipulacija.

Vrste računalne grafike:

- Rasterska grafika: u rasteru se grafički pikseli koriste za sliku koju treba nacrtati. Također je poznata kao bitmap slika u kojoj je slijed slika u manjim pikselima. U osnovi, bitmapa označava veliki broj piksela zajedno.
- Vektorska grafika: u vektorskoj grafici, matematičke formule koriste se za crtanje različitih vrsta oblika, linija, objekata i tako dalje.

Primjena računalne grafike:

- Računalna grafika se koristi za potpomognuto projektiranje inženjerskih i arhitektonskih sustava - koriste se u električnim automobilima, elektro-mehaničkim, mehaničkim, elektroničkim uređajima. Na primjer zupčanici i vijci.
- Računalna umjetnost - MS Paint, Adobe Photoshop, Gimp...
- Prezentacijske grafike - Koristi se za sažimanje financijskih statističkih znanstvenih ili ekonomskih podataka. Na primjer - trakasti grafikon, linijski grafikon…
- Zabava - koristi se u filmovima, glazbenim spotovima, igrama…
- Obrazovanje i obuka - koristi se za razumijevanje operacija složenih sustava. Također se koristi za specijalizirane sustave kao što su kadriranje za kapetane, pilote i tako dalje…
- Vizualizacija - za proučavanje trendova i obrazaca. Na primjer - analiziranje satelitske fotografije zemlje.



*Slika 3: Prikaz renderiranja slike*[7]

## 3. 3D i 2D RAČUNALNA GRAFIKA

John Warnock osnovao je Adobe Systems i stvorio je revoluciju u svijetu izdavaštva sa svojim *PostScript* jezikom za opis stranica, dok je Tom Stockham vodio grupu za obradu slika na University of Utah (UU) koja je blisko surađivala s laboratorijem računalne grafike. Jim Clark je također sudjelovao u ovoj revoluciji, gdje kasnije osniva Silicon Graphics, INC.

[7] https://www.graphics.cornell.edu/online/tutorial/objrender/ (Pristupljeno: 25.6.2022.)

Prvi veliki napredak u 3D računalnoj grafici napravili su ti rani inovatori na UU, algoritam skrivene površine. Kako bi nacrtao prikaz 3D objekta na ekranu, računalo mora odrediti koje su površine „iza“ objekta iz perspektive gledatelja, a zbog toga bi trebale biti „skrivene“ kada računalo stvara sliku. Od tada je računalna grafika postala samo detaljnija i realističnija, zahvaljujući snažnijem grafičkom hardveru i softveru za 3D modeliranje. Dvodimenzionalna 2D računalna grafika računalno je generiranje digitalnih slika, uglavnom iz dvodimenzionalnih modela, kao što su 2D geometrijski modeli, tekst i digitalne slike. 2D računalna grafika se uglavnom koristi u aplikacijama koje su izvorno razvijene na tradicionalnim tehnologijama ispisa i crtanja, kao što su tipografija, kartografija, tehničko crtanje i oglašavanje. U tim primjenama, dvodimenzionalna slika nije samo prikaz objekta iz stvarnog svijeta, već neovisni artefakt s dodanom semantičkom vrijednošću. Dvodimenzionalni modeli su stoga poželjniji jer daju izravniju kontrolu nad slikom od 3D računalne grafike, čiji je pristup sličniji fotografiji nego tipografiji. Trodimenzionalna 3D računalna grafika za razliku od 2D računalne grafike je grafika koja koristi trodimenzionalni prikaz geometrijskih podataka koji su pohranjeni u računalu za potrebe izvođenja izračuna i renderiranja 2D slika. Takve slike mogu biti za kasniji prikaz ili za gledanje u stvarnom vremenu. Unatoč ovim razlikama, 3D računalna grafika oslanja se na brojne iste algoritme kao 2D računalna vektorska grafika. 3D model je matematički prikaz bilo kojeg trodimenzionalnog objekta. Model tehnički nije grafika dok nije vizualno prikazan. Zbog 3D printanja, 3D modeli nisu ograničeni na virtualni prostor. Model se može vizualno prikazati kao dvodimenzionalna slika kroz proces koji se naziva 3D renderiranje ili koristiti u negrafičkim računalnim simulacijama i izračunima.

## 4. RAČUNALNA ANIMACIJA

Računalna animacija je umjetnost stvaranja pokretnih slika korištenjem računala, a potpodručje je računalne grafike i animacije. Sve se više stvara pomoću 3D računalne grafike, iako se 2D računalna grafika još uvijek naširoko koristi za stilske potrebe niske propusnosti i bržeg prikazivanja u stvarnom vremenu. Također se naziva CGI (*Computer-generated imagery* ili računalno generirana slika), posebno kada se koristi u filmovima.

*Slika 4: Računalno generirana slika*[8]

Postoji više metoda za postizanje animacije, a rudimentarni oblik temelji se na stvaranju i uređivanju ključnih kadrova, od kojih svaki pohranjuje vrijednost u danom trenutku, po atributu koji treba animirati. Softver za 2D/3D grafiku će interpolirati između ključnih kadrova, stvarajući krivulju vrijednosti koja može uređivati mapiranu sliku tijekom vremena, što rezultira animacijom. Ostale metode animacije uključuju proceduralne tehnike temeljene na izražavanju (prva reformira povezane elemente animiranih entiteta u skupove atributa, korisnih za stvaranje učinaka čestica i simulacije gomile), a kasnije omogućuje procijenjeni rezultat koji se vraća iz korisnički definiranog logičkog izraza, za automatizaciju animacije na predvidljiv način (pogodno za kontrolu ponašanja kostiju izvan onoga što hijerarhija nudi u postavljanju skeletnog sustava). Kako bi se stvorila iluzija kretanja, slika se prikazuje na zaslonu računala, a zatim se brzo zamjenjuje novom slikom koja je slična prethodnoj slici, ali malo pomaknuta. Ova tehnika je identična iluziji kretanja na televiziji i filmovima. Koncepti i principi slike obično proizvode optički uređaji kao što su kamere, ogledala, leće, teleskopi, mikroskopi te prirodni objekti i pojave, kao što su ljudsko oko ili vodene površine.

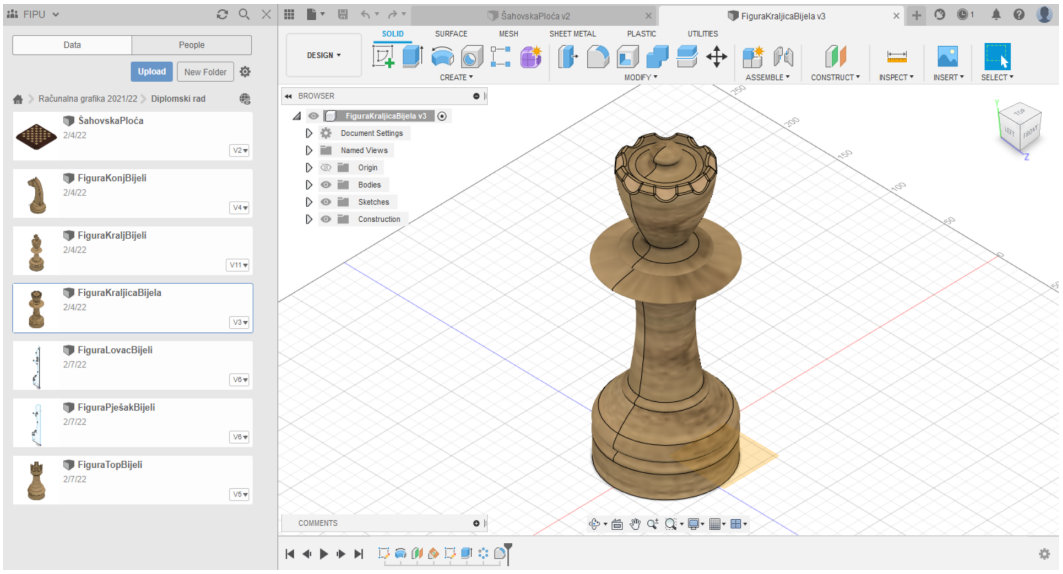
[8] https://www.pinterest.com/pin/784611566326521649/ (Pristupljeno: 25.6.2022.)

## 5. RAZVOJ CAD ALATA

Korištenje računala za olakšavanje ili pomoć pri dizajnu postoji od sredine 20. stoljeća. Računalno podržano projektiranje (CAD - *Computer Aided Design*) je korištenje računalnog softvera za projektiranje i dokumentiranje procesa dizajna proizvoda.[9] Tehničko crtanje podrazumijeva korištenje grafičkih simbola kao što su točke, linije, krivulje, ravnine i oblici. U osnovi, daje detaljan opis bilo koje komponente u grafičkom obliku. Tehnički crteži u upotrebi su više od 2000 godina. Međutim, korištenje ortografskih projekcija službeno je uveo francuski matematičar Gaspard Monge u osamnaestom stoljeću. Budući da vizualni objekti nadilaze jezike, inženjerski crteži su se razvili i postali popularni tijekom godina. Izraz "Računalno podržano projektiranje" (CAD - *Computer Aided Design*) pojavio se 1950-ih i obično se pripisuje Douglasu Rossu, informatičkom znanstveniku koji je radio na MIT-u kada je akronim osmišljen. U 1970-ima prvi CAD programi bili su sposobni stvarati samo 2D crteže, koji su u to vrijeme bili slični ručno izrađenim crtežima. Bez obzira na to, prvi jednostavni programi počeli su mijenjati način na koji proizvodna i građevinska industrija gledaju na dizajn. Velik dio istraživanja prešao je s 2D na 3D stvoreni dizajn. Do kasnih 1970-ih počeo se pojavljivati rani softver za čvrsto modeliranje, dajući dizajnerima mogućnost kombiniranja osnovnih geometrijskih oblika. Tijekom 1980-ih, 3D modeliranje se pojavilo kako su se razvile nove teorije i algoritmi. Do 1990-ih, računala su bila dovoljno moćna da podržavaju CAD softver kao što je AutoCAD, čineći takve programe poznatim imenom za dizajnere u svim industrijama. CAD obuka postala je sinonim za arhitektonsku obuku i programi su od tada postali sve prilagođeniji korisnicima. Dok su raniji inženjerski crteži bili ručno izrađeni, studije su pokazale da su inženjerski dizajni prilično komplicirani. Rješenje mnogih inženjerskih problema zahtijeva kombinaciju organizacije, analize, principa rješavanja problema i grafičkog prikaza problema. Objekti u inženjerstvu predstavljeni su tehničkim crtežom (koji se naziva i nacrt) koji predstavlja nacrte i specifikacije fizičkog objekta i odnosa podataka. Tehnički crtež mora biti precizan jer on prenosi jasno sve podatke o predmetu. Ovdje CAD dolazi do izražaja.

---

9 https://medium.com/technical-illustration/evolution-of-cad-from-light-pens-to-synchronous-technology-549cc8eef5d0 („Evolution of CAD — From light pens to Synchronous Technology!“, pristupljeno: 25.6.2022.)




*Slika 5: Prikaz sučelja CAD alata Fusion 360-a*

CAD se koristi za dizajn, razvoj i optimizaciju proizvoda. Iako je vrlo svestran, CAD se intenzivno koristi u dizajnu alata i opreme potrebnih u procesu proizvodnje, kao i u domeni izgradnje. CAD omogućuje inženjerima dizajna da postave i razviju svoj rad na zaslonu računala, pohrane ga za buduće korištenje, te ispišu za prikaz. Kad je prvi put predstavljena, računalno potpomognuta proizvodnja (CAM - *Computer Aided Manufacturing*) nije bila baš ekonomski isplativa jer su strojevi u to vrijeme bili vrlo skupi.[10] Sve veća snaga računala u kasnijem dijelu dvadesetog stoljeća, s dolaskom miniračunala, a zatim i mikroprocesora, omogućila je inženjerima korištenje CAD datoteka koje služe kao točan prikaz dimenzija odnosno svojstava objekta. CAD se koristi za postizanje preliminarnog dizajna i izgleda, detalja dizajna i proračuna, stvaranje 3D modela, stvaranje i objavljivanje crteža, kao i za povezivanje s analizom, marketingom, proizvodnjom i krajnjim korisničkim osobljem. CAD olakšava proces proizvodnje prijenosom detaljnih informacija o proizvodu u automatiziranom obliku koji obučeno osoblje može univerzalno interpretirati. Može se koristiti za izradu dvodimenzionalnih i trodimenzionalnih dijagrama. Korištenje CAD softverskih alata omogućuje promatranje objekta iz bilo kojeg kuta, čak i iznutra gledajući prema van. Jedna od glavnih prednosti CAD crteža je da je uređivanje brz proces u usporedbi s ručnom metodom. Osim detaljnog inženjeringa 2D ili 3D modela, CAD se široko koristi od idejnog dizajna i izgleda proizvoda do definiranja proizvodnje komponenti. CAD smanjuje vrijeme projektiranja dopuštajući preciznu simulaciju umjesto izrade i testiranja fizičkih prototipova. Integracija CAD-a s CAM-om još više usmjerava razvoj proizvoda. CAD se trenutno široko koristi za industrijske proizvode, animirane filmove i druge primjene. Za ispis profesionalnih

[10] https://www.technicalfoamservices.co.uk/blog/blog-history-of-cad-cam/ (Pristupljeno: 25.6.2022.)

nacrta dizajna obično je potreban poseban pisač ili crtač. CAD programi koriste ili vektorsku grafiku ili rastersku grafiku koja pokazuje kako će objekt izgledati. CAD softver omogućuje; učinkovitost u kvaliteti dizajna, povećanje produktivnosti inženjera, poboljšano vođenje evidencije boljom dokumentacijom i komunikacijom. Danas je korištenje CAD-a zahvatilo gotovo sve industrije. Od zrakoplovstva, elektronike do proizvodnje, CAD se koristi u svim industrijskim vertikalama. Budući da CAD potiče kreativnost i ubrzava produktivnost, postaje sve korisniji kao važan alat za vizualizaciju prije stvarne implementacije proizvodnog procesa. To je također jedan od razloga zašto CAD obuka postaje sve važnija. U području inženjerstva, CAD omogućuje projektiranje i izradu nacrta alata i strojeva koji se koriste u proizvodnji komponenti. Osim toga, detaljan inženjering 3D modela i 2D crteža fizičkih komponenti može se lako generirati primjenom CAD-a. U području arhitekture zgrada, CAD je smanjio potrebu za ručnim crtanjem. Pomaže u generiranju svih detalja zgrade kao što su tlocrti, presjeci, elevacije, pogledi iz perspektive i prikazi modela u kraćem vremenu. Softver omogućuje izradu nacrta i projektiranje svih vrsta građevina. Osim projektiranja, razvoja i optimizacije proizvoda u građevinarstvu i inženjerstvu, CAD se također koristi u poslovne svrhe kao što su sljedeće:

- Testiranje na sadržajnim proizvodima

CAD eliminira potrebu za fizičkim testiranjem cjelokupnog dizajna proizvoda. Umjesto provođenja fizičkog testiranja, programeri mogu izvoditi virtualne testove koristeći CAD koji može simulirati isto stanje. Primjena virtualnih testova često je prihvaćena u zrakoplovnim tvrtkama čije testiranje može biti dugotrajno i skupo.

- Generiranje ideja

CAD omogućuje fleksibilan proces stvaranja ideja kada se ukloni ograničavajući faktor proizvodnje prototipa. Poduzeća, zaposlenici i potencijalni kupci izloženi su otvorenijim i novim idejama i prijedlozima. Prijedlozi za nove proizvode tada se mogu brzo testirati po nižoj cijeni.

- Poboljšanje dizajna

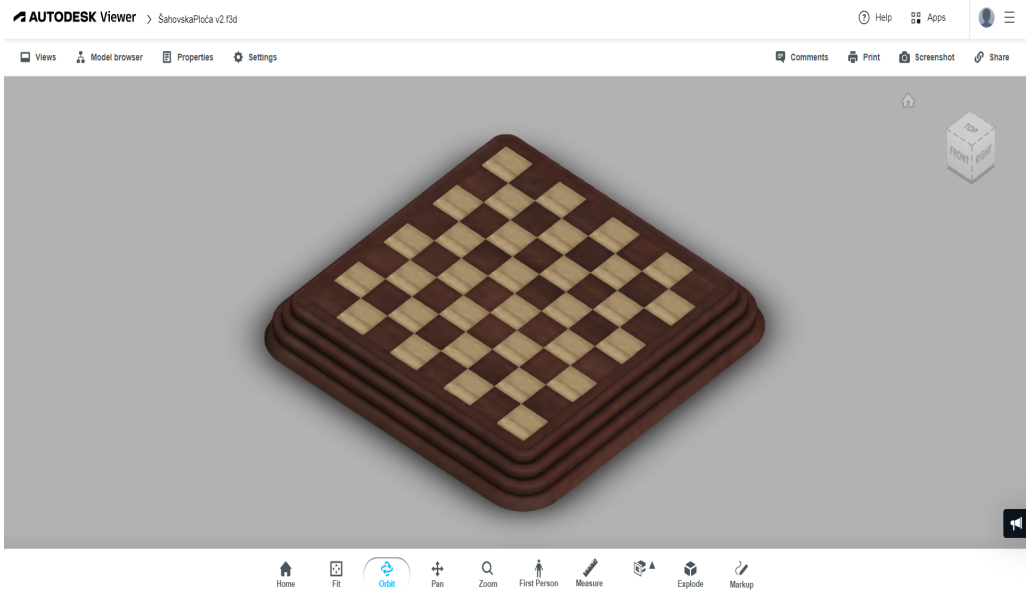
CAD trenutno omogućuje lagano poboljšanje dizajna novih proizvoda. Korisno je u istraživanju mogućih poboljšanja postojećih proizvoda.

- Testiranje tržišta

CAD omogućuje raniji proces testiranja tržišta jer se virtualni novi proizvodi mogu predstaviti brže nego fizički prototipovi, a povratne informacije mogu se dobiti trenutno.

Također se može primijeniti u području medicine za manipulaciju medicinskim slikama kao što je digitalna rendgenska slika (*Digital Imaging and Communications in Medicine* - DICOM). Primjena CAD-a u prezentaciji medicinskih slika još uvijek je nova u usporedbi s CAD-om koji se naširoko koristi u područjima koja uključuju projektiranje proizvoda proizvodne industrije i građevinarstva. Od svog uvođenja u kasnim 1960-ima, CAD softver je napredovao velikim koracima. Vrste prikaza CAD-a su; 2D CAD, 3D CAD, 3D žičani okvir i

modeliranje površina, modeliranje čvrstih tijela. Uz sve više i više tvrtki koje se okreću CAD/CAE (*Computer aided engineering* – Računalno potpomognuto inženjerstvo)/CAM kako bi postigle učinkovitost, točnost i smanjeno vrijeme do plasmana proizvoda na tržište, raste potražnja za CAD softverom. Industrijski lideri u ovom prostoru uključuju AutoCAD, Dassault Systems, Altar i AutoDesk-ov Fusion 360.

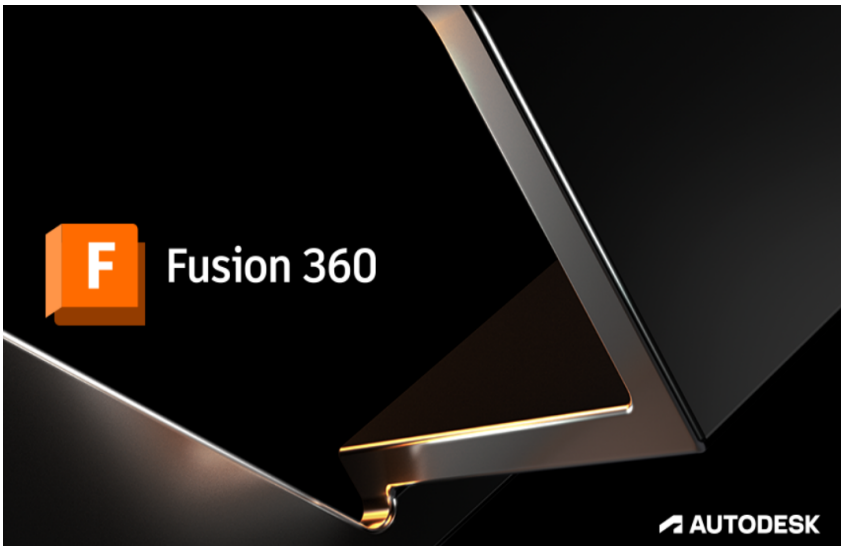



*Slika 6: Prikaz šahovske ploče u Autodesk Viewer-u*

CAD softver nudi nekoliko prednosti koje privlače brojne ljude u industriji. U osnovi, nudi prednosti koje mogu dovesti do uštede vremena i troškova, a to se posebno odnosi na poslovni svijet jer u pogledu komercijalnog čovjeka vrijeme je novac. Prije 1970-ih, gotovo svaki crtež proizveden u svijetu rađen olovkom ili tušem na papiru i svaka promjena zahtijeva brisanje i ponovno crtanje što će proizvesti otpad. CAD softver omogućuje dizajnerima da razviju svoj rad na ekranu, te lakše buduće uređivanje. Softver osigurava kratki ciklus dizajna jer eliminira potrebe za ponovnim crtanjem kada se dogode manje promjene, donosi prednosti u uštedi vremena. Druge uštede troškova i vremena u razvoju proizvoda također se mogu postići uklanjanjem potrebe za fizičkim testiranjem cijelog dizajna proizvoda. Umjesto provođenja fizičkog testiranja, programeri mogu izvoditi virtualne testove koristeći CAD koji može prikazati simulaciju proizvoda. Primjena virtualnih testova često je prihvaćena u zrakoplovnim tvrtkama čije testiranje može biti dugotrajno i skupo. Ušteda se može postići smanjenjem broja projektanata. Korištenjem CAD-a, profesionalni dizajneri mogu brže obaviti projektiranje i eliminirati potrebu za dodatnim dizajnerima. Osim toga, ušteda se također događa kada CAD softver nudi bolje i standardizirane crteže koji zatim olakšavaju komunikaciju. Štoviše, CAD omogućuje brzu i jednostavnu promjenu dizajna, što također

dovodi do bržeg razvoja proizvoda, a brži razvoj proizvoda dovest će do povećanja produktivnosti dizajnera. Softver omogućuje razumijevanje kako stvarni proizvod radi, dajući animaciju proizvodu tako da se mogu napraviti trenutne izmjene, ako je to potrebno. CAD softver pomaže dizajneru u sintetiziranju, analizi i dokumentiranju dizajna, stoga se produktivnost dizajnera može poboljšati zahvaljujući brzom projektiranju, niskim troškovima projektiranja i kraćim vremenima završetka. Zapravo, CAD može poboljšati kvalitetu dizajna jer softver nudi velik broj alata koji pomažu u istraživanju i analizi predloženog dizajna. Iako većina korisnika razumije prednosti koje donosi CAD, u međuvremenu dolazi s nekoliko nedostataka koje inženjer ili poduzetnik mora razumjeti. Ove prednosti mogu utjecati na pristup izlasku proizvoda na tržište. Osim toga, njegova primjena zahtjeva skupe početne troškove uključujući hardver, softver i obuku. Računalna tehnologija koja se koristi u CAD-u skupa je za implementaciju i podršku softvera. Kao takva tehnologija nije konstantna budući da novi hardverski i softverski programi stalno dolaze na tržište i stoga korisnici moraju pratiti te promjene zbog kompatibilnosti. Prilikom implementacije CAD-a mora se uzeti u obzir rizik od kvara tehnologije jer se korišteno računalo može pokvariti. To zatim utječe na napredak razvoja proizvoda koji nije pogodan u strogom rasporedu proizvodnje. Rizik je najveći kada se tvrtka oslanja na strukturu pokretne trake jer će kvar utjecati na sve točke proizvodnje umjesto na izolirano proizvodno područje. Još jedan od nedostataka je taj kada se računalo pokvari, izgubit će se svi spremljeni podaci uključujući CAD kopiju. CAD donosi prednosti korisnicima, ali u međuvremenu također donosi ograničenja ili nedostatke. U osnovi, ušteda troškova i vremena glavne su prednosti jer se dizajn i crteži mogu izraditi brže nego ručno. Međutim, kada dođe do kvara tehnologije, posebice računala, to će utjecati na proizvodnu liniju i uzrokovati kašnjenja. Stoga se prilikom usvajanja CAD-a u razvoju proizvoda mora donijeti ispravna odluka i ona mora uzeti u obzir čimbenike kao što su vrijeme i trošak prijenosa tehnologije, kao i raspoloživu stručnost. Fusion 360 je CAD platforma temeljena na oblaku koja je danas vrlo pristupačna. Jednostavan je za korištenje i ima sve zajedničke značajke koje se mogu očekivati od popularnih CAD paketa. Fusion 360 izgrađen je od temelja da bude sveobuhvatno rješenje za razvoj proizvoda i ima za cilj ponuditi jednostavan tijek rada od konceptualnog dizajna pa sve do proizvodnje. Fusion 360 ima vrlo veliku bazu znanja koja temeljito pokriva svaku značajku softvera, a ovim se vodičima može pristupiti putem Fusion 360 kao i putem web stranice Autodesk. Softver se često nadograđuje, a nove značajke stižu svakih nekoliko mjeseci. Fusion 360 idealan je za tvrtke s velikim prometom, kao i za novoosnovane tvrtke koje traže profesionalni alat za izlazak na tržište. Fusion 360 može izvoditi operacije koje zahtijevaju velike resurse u

oblaku, uključujući renderiranje, simulaciju, optimizaciju oblika i generativni dizajn. To znači da se rad može nastaviti dok se sav težak posao obavlja na oblaku.




*Slika 7: Autodesk Fusion 360*

## 6. ALATI ZA RAZVOJ PROJEKATA I VIDEOIGARA

Alati za razvoj projekata i videoigara pomažu u svim aspektima stvaranja projekata kao što su; teksturiranje, modeliranje, animacije i zvuk. Neki od ovih alata namijenjeni su razvoju specifičnih žanrova igara, dok drugi imaju svestranu primjenu. Alati za razvoj projekata i igara baza su softverskog okruženja i okvira s postavkama i konfiguracijom za dizajniranje i optimiziranje u različitim programskim jezicima, a neki od njih će se prikazati u nastavku. Na primjer umjetnički i vizualni alati olakšavaju dizajn i koriste se za stvaranje umjetničkih aspekata igara i projekata. Cjelokupni razvoj počinje s grubim skicama okruženja, objekata, postavki ili likova. Kompleti za razvoj softvera skup su softverskih alata, programa, biblioteka, uzoraka izvornog koda, dokumentacije i aplikacija koje programeri igara mogu koristiti za izradu vlastitih projekata.

Programski alati za razvoj projekata i videoigara:

- Adobe Photoshop

Adobe Photoshop je softver koji se intenzivno koristi za uređivanje rasterskih slika, grafički dizajn i digitalnu umjetnost. Koristi raslojavanje kako bi omogućio dubinu i fleksibilnost u procesu dizajna i uređivanja, kao i pružanje snažnih alata za uređivanje koji su u kombinaciji sposobni za gotovo sve. Napravila su ga braća Thomas i John Knoll 1988 godine. 1989. godine John je prodao program tvrtki Adobe Systems, koja ga je reklamirala kao "Photoshop". Od tada je program postao de facto industrijski standard za uređivanje rasterske grafike.

- Visual Studio

Microsoft Visual Studio je IDE (*Intergrated development environment* – Integrirano razvojno okruženje) koji je napravio Microsoft i koristi se za različite vrste razvoja softvera kao što su računalni programi, web stranice, web aplikacije, web usluge i mobilne aplikacije. Sadrži

alate za dovršavanje, prevoditelje i druge značajke za olakšavanje procesa razvoja softvera. Visual Studio postoji od 1997. godine, a njegova prva verzija bila je Visual Studio 97. Od tada je bilo puno različitih verzija, a trenutna je Microsoft Visual Studio 2022.

- Unreal Engine

Unreal Engine je alat za razvoj igara, a razvio ga je Epic Games 1988. godine. Trenutno se koristi za izradu borbenih igara, RPG-ova, Stealtha i drugih MMORPG-ova. Alat koristi kodiranje jezika C++.

Većina programera koristi Unity za kreiranje igara jer on nudi osnovne tehnike razvoja koje uvelike pomažu početnicima.

7. UNITY 3D

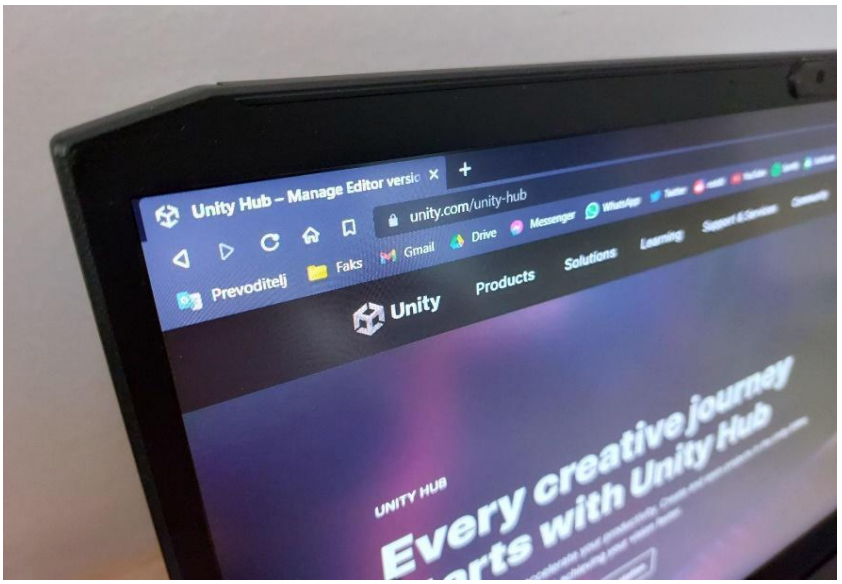
Unity je najpopularniji programski alat namijenjen između ostalog za projektiranje i dizajn igara. Sadrži mnoštvo značajki i dovoljno je fleksibilan da napravi gotovo svaku igru koja se može zamisliti. Sa značajkama na različitim platformama, Unity je jednako popularan među hobi programerima i AAA studijima.[11] Korišten je za stvaranje igara poput Pokemon Go, Hearthstone, Rimworld, Cuphead i još mnogih drugih.

*Slika 8: Unity logo*[12]

Unity također sadrži i alate za razvoj 2D igara. Programeri ga vole zbog C# (C Sharp) API-ja (*Application programming interface* – Programsko sučelje aplikacije) za skriptiranje i Visual Studio integracije. Unity također nudi JavaScript kao skriptni jezik i MonoDevelop kao IDE za one koji žele alternativu Visual Studiju. Ali i umjetnici ga vole jer dolazi s alatima za animaciju koji olakšavaju stvaranje vlastitih 3D scena ili izgradnju 2D animacija od nule. U Unityju se može animirati gotovo sve. Unity 3D također nudi besplatnu verziju tako da programeri mogu objaviti igre napravljene u besplatnoj verziji Unity-a, sve dok zarađuju manje od 100.000 dolara od njih. Za one koji su spremni platiti, Unity nudi neke dodatne značajke i fleksibilan plan licenciranja u okviru modela pretplate na više razina. Premium

[11] https://www.alltechbuzz.net/hr/2d-ili-3d-koji-bi-smjer-trebao-odabrati-po%C4%8Detnik-unity-developer-igre/ ("2D ili 3D: koji bi smjer trebao odabrati početnik Unity programer igara", pristupljeno: 25.6.2022.)
[12] https://1000logos.net/unity-logo/ (Pristupljeno: 3.9.2022.)

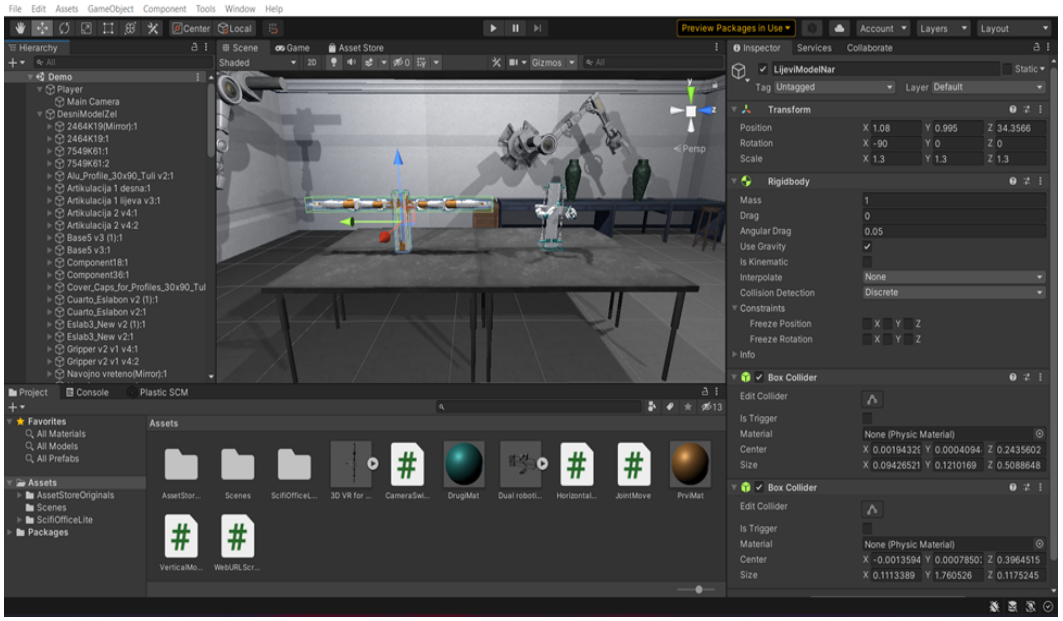
korisnici će također imati pristup Unity-ovom izvornom kodu i podršci za razvojne programere.[13]




*Slika 9: Unity Hub*

Budući da Unity postoji od 2005. godine, razvio je ogroman broj korisnika i veliku biblioteku resursa. Ne samo da Unity ima odličnu dokumentaciju, već je i broj samih video zapisa i praktičnih vodiča na mreži velik. Početnicima je prigodno početi s Unity-om samo iz tog razloga. Među sve većim popisom alata za izradu video igara Unity služi kao portal znanja i resursa izgrađen isključivo na njihovoj velikoj zajednici. Unity 3D dolazi s gomilom profesionalnih alata za programere i umjetnike. Unity pruža radni prostor koji kombinira alate prilagođene umjetnicima s dizajnom vođenim komponentama koji razvoj igara čini prilično intuitivnim. I 2D i 3D razvoj moguć je u Unity-u, uz fiziku kojom upravlja popularni *Box2D engine*. Unity se vrlo jednostavno preuzima i instalira. Sve se nalazi na Unity-evoj stranici za preuzimanje. Na njoj se preuzima Unity Hub, koji služi kao upravitelj preuzimanja koji će omogućiti upravljanje različitim verzijama softvera Unity, zajedno sa svim dodatnim značajkama. Potrebno je napraviti profil te se zatim s istim prijaviti. Nakon otvaranja Unity Hub-a, može se odabrati najnovija verzija za preuzimanje. Instalacijski program provodi korisnika kroz jednostavne korake, te nudi dodatne buildove za Android, iOS, Windows, ovisno o željenoj platformi. Putem Unity Hub-a se također otvaraju projekti.[14]

[13] https://it-s.com/what-is-unity-3d/ ("What Is Unity 3D?", pristupljeno: 25.6.2022.)
[14] https://code.tutsplus.com/tutorials/introduction-to-unity3d--mobile-10752 ("Introduction to Unity3D", pristupljeno: 25.6.2022.)



*Slika 10: Unity Sučelje*

Ovo su glavni prozori te njihovo svojstvo u programu;

- Hijerarhija (*hierarchy*) - Prema zadanim postavkama nalazi se krajnje lijevo, a ona prikazuje popis svih *GameObjects* u sceni. Služi brzom lociranju i odabiru bilo kojeg aspekta projekta kako bi mu se promijenila njegova svojstva.
- Scena (*scene*) - Najveći prozor u sredini programa Unity. Na njemu se nalazi prikaz trenutne razine, izbornika ili svijeta s kojim se trenutno radi. Ovo je prostor u kojem se slobodno može povlačiti, ispuštati, povećavati i smanjivati *GameObjects*. Ikone koje se nalaze u gornjem lijevom kutu programa Unity mijenjaju način na koji se komunicira s GameObject-ima i scenom. Ruka služi pomicanju pogleda u bilo kojem smjeru, dok strelice omogućuju pomicanje objekata u 3D prostoru duž tri osi.
- Igra (*Game*) - Ovaj dio je obično skriven iza prozora scene i može mu se pristupiti preko kartice na vrhu. Igra prikazuje pogled na scenu na način kako se zapravo vidi kad se pokrene program. Ona daje perspektivu kao kamera i prikazivati će dio gdje je kamera i postavljena. Služi za testiranje projekta.
- Trgovina sa sredstvima (*Asset Store*) - također se nalazi na kartici i omogućuje pristup sredstvima koja je razvila zajednica.
- Inspector (inspektor) - ovaj se prozor nalazi na najdaljem desnom dijelu korisničkog sučelja. Svrha mu je pregled i uređivanje svojstava odabranog *GameObject-a*. Naprimjer promjena veličine (razmjera), položaja (transformacija) ili dodavanju komponenti kao što je C# skripta.
- Projekt (*project*) - nalazi se na dnu zaslona i prikazuje sve datoteke koje čine projekt. Ovdje se mogu praviti C# skripte koje se zatim koriste u Unity-u. Ovdje se također ispuštaju 3D datoteke ili teksture.
- Konzola (*Console*) - mjesto gdje se mogu vidjeti informacije iz samog Unity-a. Ovdje će se prikazati moguće pogreške ili upozorenja vezana za kod ili ako postoje problemi koji se mogu riješiti samom postavkom programa Unity.[15]

[15] https://www.androidauthority.com/what-is-unity-1131558/ ("What is Unity? Everything you need to know", pristupljeno: 25.6.2022.)

Unity se sastoji od skupa elemenata ili komponenti koji zajedno čine projekt. Ovi elementi igraju glavnu ulogu u interaktivnosti samog projekta. Ti elementi se sastoje od;

- Imovine (*Assets*) - prikaz stavki projekta koji se mogu implementirati. To može biti datoteka uvezena izvan Unity-a, poput 3D modela, audio i zvučne datoteke, slike, teksture ili bilo koje druge vrste datoteka koje podržava Unity. Također postoji kolekcija vrsta sredstava koja se mogu koristiti unutar Unity-a. Primjeri takve imovine su: slike, audio mikser, upravljač animatora.
- Projekt (*Project*) - mapa ili lokacija koja sadrži kompletan projekt zajedno sa svom povezanom imovinom koja također može sadržavati biblioteku i podmapu imovine. Paketi - unaprijed kompilirani klaster sredstava projekta.
- *GameObject* - svaki objekt prisutan u projektu. Tehnički, oni ne dodaju nikakvu funkcionalnost projektu, već samo djeluju kao držači komponenti. Neki od njih su; *Script, Transform, RigidBody i Light.* Komponente - osnovno građevni blokovi, tj. matice i vijci objekata i njihovih aktivnosti u projektu. Svaki *GameObject* ima automatski postavljen *Transform Component,* jer on diktira gdje će biti pozicioniran unutar Unity okruženja i kako će se rotirati i skalirati.
- Scene (*Scenes*) - mogu se definirati kao osnovni ili nadređeni (roditelj) objekt, a one se koriste u svrhu dodavanja razina projektu. Jedna ili više scena (odnosno razina) stavljaju se u projekt i međusobno povezuju. Logika prelazaka razina će biti postavljena putem metoda koja će se odvijati zajedno s korisnikom unutar scene.
- *Prefab* - komponente *GameObject-a* koje se mogu ponovno koristiti i koje su postavljene u prozoru *Project View.* Prefabovi se mogu uvesti u bilo koji broj scena, koliko god je to puta poželjno. Svaka naknadna promjena na *Prefab-u* se odražava kroz cijeli projekt.
- Građa (*Build*) - izvedena adaptacija projekta koja će sadržavati sve scene za reprodukciju na izabranoj platformi.

*Slika 11: Rad u Unity-u*

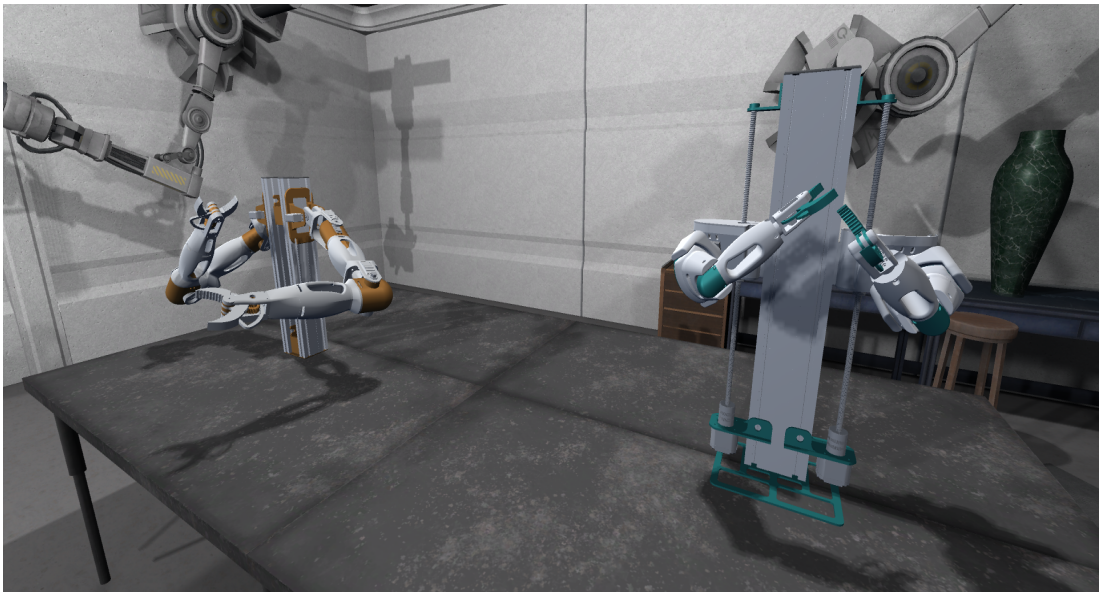
Unity koristi komponentni pristup razvoju igara koji se vrti oko prefabova. Uz *prefabove*, dizajneri igara mogu učinkovitije graditi objekte i okruženja, te iste brže skalirati. S snažnim *shaderima*, materijalima temeljenim na fizici, naknadnom obradom i sustavima rasvjete visoke razlučivosti, Unity može isporučiti impresivnu grafiku u cijelosti. Svi od Ubisoft-a do

NASA-e također koriste Unity-evu VR (*Virtual reality*) tehnologiju. Za samu platformu: Unity je izgrađen u C++ i tijekom godina optimiziran za performanse. Uz podršku za sve glavne konzole i operativni sustav, igre razvijene u Unity-u mogu se implementirati na apsolutno bilo koju platformu. Uz Unity alate za uređivanje možete istovremeno upravljati ulazima za miševe, tipkovnice i kontrolere za igre. Postoji i prilično snažna podrška za rješenja temeljena na oblaku za igre za više igrača s *hostingom* poslužitelja i skalabilnim povezivanjem, što ga čini rješenjem sve u jednom za višestruka iskustva. Timska suradnja uvelike je poboljšana u novijim verzijama Unity-a. Ugrađena kontrola verzija i integracija oblaka čine rad s drugima lakšim nego ikad prije. Unity ima prilagodljiv uređivač s punom API podrškom za izradu vlastitih alata za uređivanje i skripti. Također je moguće kreirati alate koji se kasnije mogu koristiti u programu. A svakako je vrijedno spomenuti i *Asset store* koji sadrži tisuće modela, scenarija, scena, materijala i svega ostalog što se može preuzeti u vlastiti projekt. Moguće je čak kreirati vlastitu imovinu u trgovini koja se onda može prodati drugim korisnicima.[16]

## 8. DETALJAN PRIKAZ IZRADE I RADA S ROBOTSKIM RUKAMA

Robot može obavljati razne vrste poslova ovisno o tome o kojoj se vrsti robota govori. Neki od njih su jednostavni, a neki prilično složeni. Roboti su dizajnirani za obavljanje zadataka koje ljudi mogu ili ne moraju obaviti umjesto njih. Koriste se za izradu automobila, rastavljanje bombi, pakiranje kutija, javljanje na telefone, košenje dvorišta i istraživanje olupina brodova u dubokom moru i mnoge druge stvari. Naravno, svi se roboti po svojoj prirodi mogu reprogramirati da se ponašaju drugačije, ali neki su ograničeni svojim fizičkim oblikom. Robot može biti autonomni robot opće namjene ili namjenski za određeni posao. Ovo poglavlje se usredotočuje na dizajn i konstrukciju robotske ruke. Modeli su učitani kao novi *asseti* u razvojnom programskom alatu Unity 3D u *F3D* formatu. Dodani su im novi materijali, odnosno boje da osim oblika budu različiti i po bojama, jednom modelu su dodijeljeni narančasti detalji dok drugi model ima zelene detalje. U Unity *Asset Store-u* su pretražena okružja u koji su postavljeni modeli. Pronađeno je okruženje pod nazivom *„Free Sci-Fi Office Pack“* od korisnika *„Terresquall“* koje je pokazalo dobre rezultate tijekom rada. Modeli su postavljeni u prostor na stol koji se nalazi na sredini prostorije.

[16] https://conceptartempire.com/what-is-unity/ (“What is Unity 3D & What is it Used For?”, pristupljeno: 25.6.2022.)

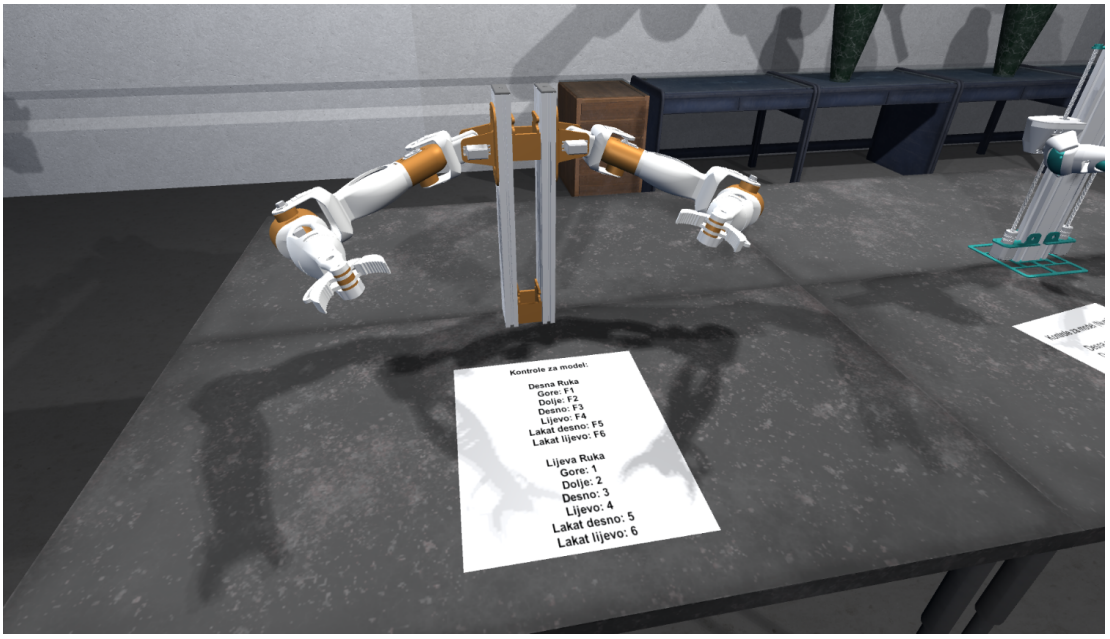


*Slika 12: Prikaz modela*

Nakon toga su sortirane sve komponente oba modela kako bi se olakšao budući rad. To je napravljeno na način da su grupirani sadržaji modela u podmape ovisno o kojem dijelu modela se radi, naprimjer lijevi model je u mapi pod nazivom *„LijeviModelNar“* koja u sebi sadrži podmape; *„Tijelo“, „JointLijevoRameVertical“* i *„JointDesnoRameVertical“*. Nakon toga su dodane mogućnosti pokreta ruku modela pomoću tipkovnice. Pri tome su pomogle skripte za vertikalno i horizontalno kretanje, rotaciju i kretanje u gornjem i donjem smjeru.

- Lijevi model (narančasti)

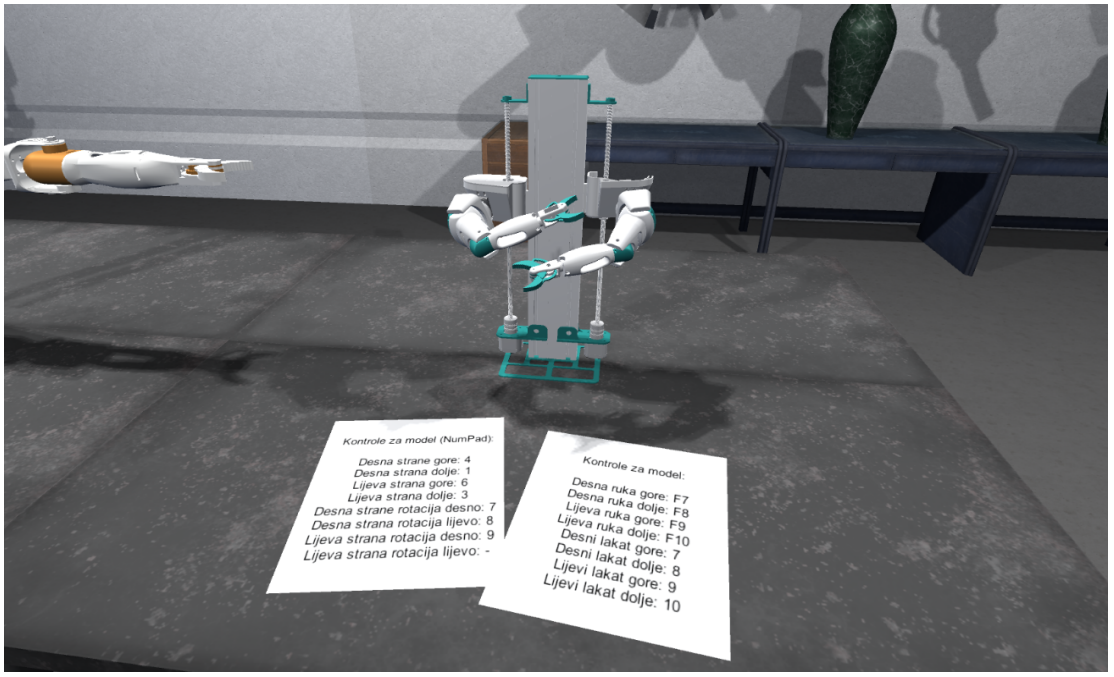
Ruke lijevog modela, odnosno narančastog mogu se kretati pritiskom na tipke s brojevima, odnosno funkcijskim tipkama (F-tipke). Pritiskom na tipku F1 desna ruka se pomiče u gornjem smjeru, dok pritiskom na F2 se kreće u donjem smjeru. Pritiskom na tipku F3 desna ruka lijevog modela se kreće u desnom smjeru, a lijevo pritiskom na tipku F4. Lakat desne ruke se pomiče pritiskom na tipku F5 u desnom smjeru, a pritiskom na F6 u lijevom smjeru.




*Slika 13: Prikaz lijevog (narančastog) modela*

- Desni model (zeleni)

Ruke desnog modela se mogu kretati gore i dolje po ovojnicama sa strana modela. Postavljene tipke za kretanje desne ruke u gornjem smjeru su 4 (NumPad), dok se ruka kreće u donjem smjeru tipkom 1 (NumPad). Lijeva ruka se kreće u gornjem smjeru pomoću tipke 6 (NumPad), a dolje sa tipkom 3 (NumPad). Desni model također ima mogućnost rotacije ruku u području ramena. Desna ruka se rotira u desno pomoću tipke 7 (NumPad), a lijevo 8 (NumPad). Lijeva ruka se rotira pomoću tipke 8 (NumPad) i tipke - (NumPad). U nastavku ću navesti postavljene tipke za kretanje ruku u području ramena i lakta. Desna ruka se kreće gore pomoću tipke F7, a dolje F8. Lijeva ruka se kreće u gornjem smjeru sa tipkom F9, a donjem F10. Desni lakat se kreće gore sa tipkom 7, a dolje sa tipkom 8. Lijevi lakat se pomiče u gornjem smjeru sa tipkom 9, a donjem sa 10.



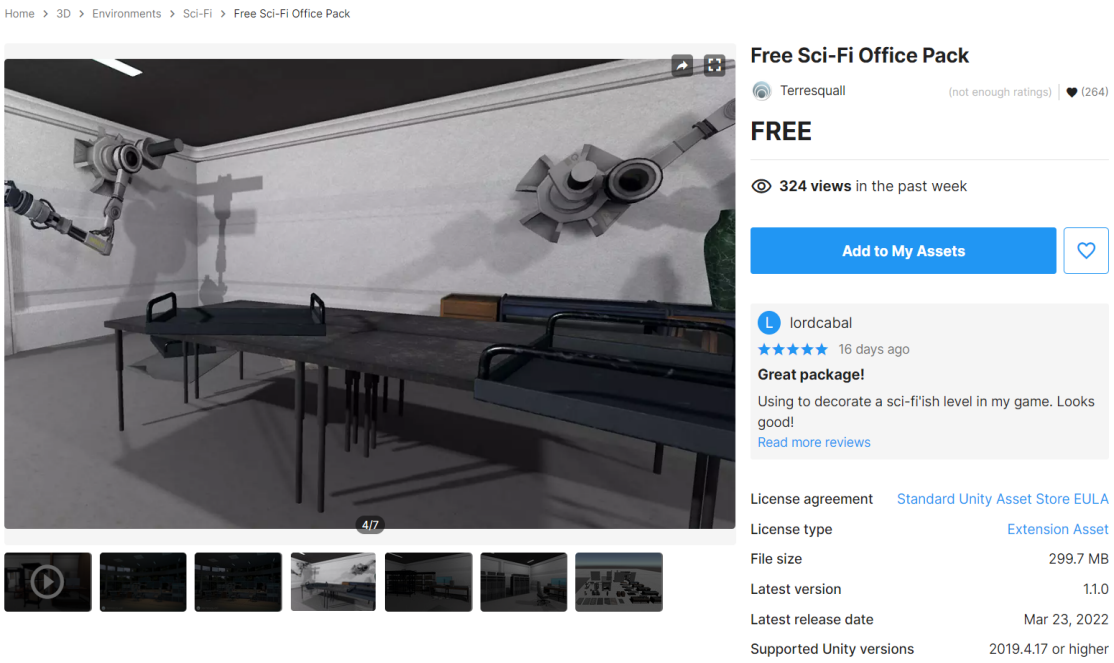

*Slika 14: Prikaz desnog (zelenog) modela*

- Papiri na kojima su prikazane kontrole

Papiri sa kontrolama koji se nalaze ispred modela su dodani u okružje odnosno mapu *Office1* pod nazivima *"KontroleLijeviModel"* i *"KontroleDesniModel"*. Dodani su kao novi *GameObject* u kojem se nalazi *3D object Plane* koji je oblikovan u oblik papira te su zatim postavljeni na mjesta ispred modela. Tekst je dodan na način da je postavljen kao dijete od kontrola, te je zatim dodan *Text mesh* pomoću kojeg je napisan tekst, oblikovan i postavljen na sredinu papira.

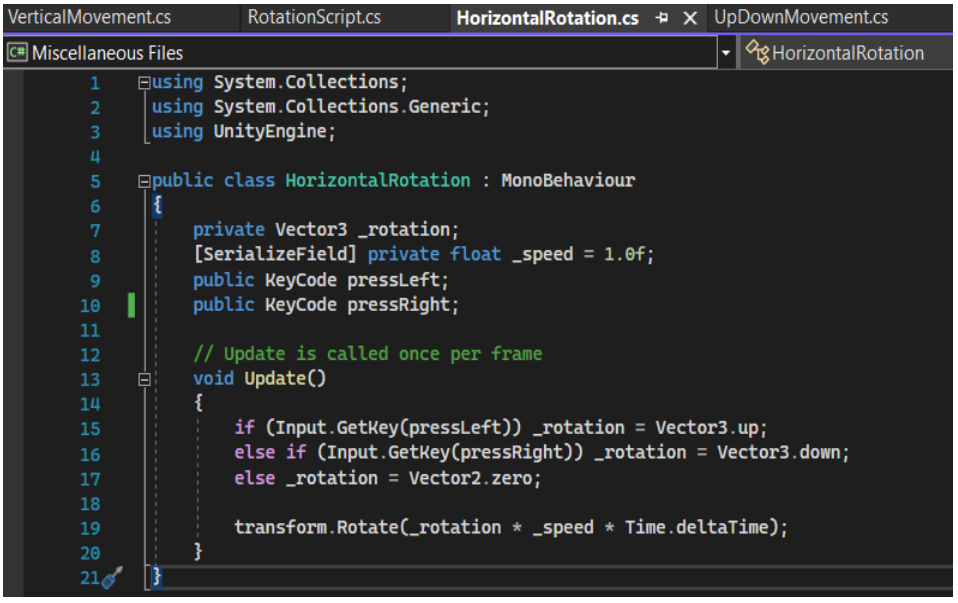
## 9. PRIKAZ KORIŠTENIH NAREDBI, BIBLIOTEKA I KODA

Za okružje je korišten paket koji je pronađen u Unity *Asset Store-u*. Ovaj paket je besplatna verzija *Sci-Fi Office* paketa. Uključuje 43 ručno odabrana *asseta* od preko 350 uredskih i znanstvenofantastičnih materijala iz plaćene verzije. Sva sredstva u paketu dizajnirana su za kreiranje interijera i stvorena su s obzirom na modularnost - lako ih je sastaviti u raznim kombinacijama. Paket se sastoji od podmapa; Mrežice (*Meshes)* - sadrži komponente za osvjetljenje, materijale, uredski interijer, *sci-fi* okružje, teksture, zidove. *Prefabs* - sadrži komponente za pod, stolove, stolice, vrata, ladice, svjetlo, tehničke dodatke, televizije i zidove. *Scene* - sadrži komponente *Asset Display, Demo.unity i Demo. Scripts* - sadrži skripte *DemoDoor i DemoFirstPersonController.* Te priloženu dokumentaciju koja se može pronaći i online na sljedećoj poveznici: terresquall.com/games/sci-fi-office-pack/#guide.




*Slika 15: Okružje Free Sci-Fi Office Pack*[17]

Pri izradi projekta korištene su četiri različite skripte. Skripte služe za kretanje ruku pomoću tipkovnice. Prva skripta se koristi za okretanje ruke u horizontalnom smjeru.

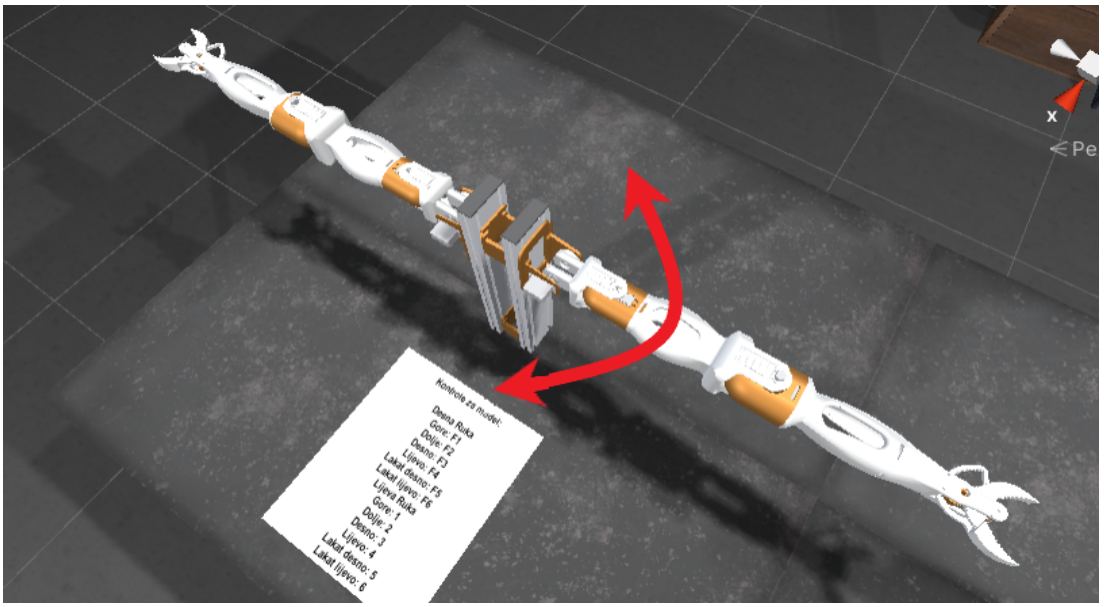



*Slika 16: Skripta HorizontalRotation*

U javnoj klasi *„HorizontalRotation“* nalaze se privatna varijabla *Vector3 _rotation* koja služi za pohranu pozicije ruke. *Private float _speed* služi za određivanje brzine kretanja. *Public KeyCode pressLeft* služi za postavljanje tipke sa tipkovnice koja će služiti za kretnju u lijevom smjeru, dok *public KeyCode pressRight* služi za desni smjer. Kada dođe do pritiska

[17] https://assetstore.unity.com/packages/3d/environments/sci-fi/free-sci-fi-office-pack-195067 (Pristupljeno: 25.6.2022.)

postavljene tipke za kretnju u lijevom smjeru, pozicija *Vector3* će mijenjati dokle god je tipka pritisnuta, isto vrijedi i za tipku za kretnju u desnom smjeru. Ako ni jedna tipka nije pritisnuta, pozicija ruke se neće promijeniti. *Transform.Rotate* nam služi za određivanje brzine kretanja ruke.




*Slika 17: Prikaz horizontalnog kretanja ruke*

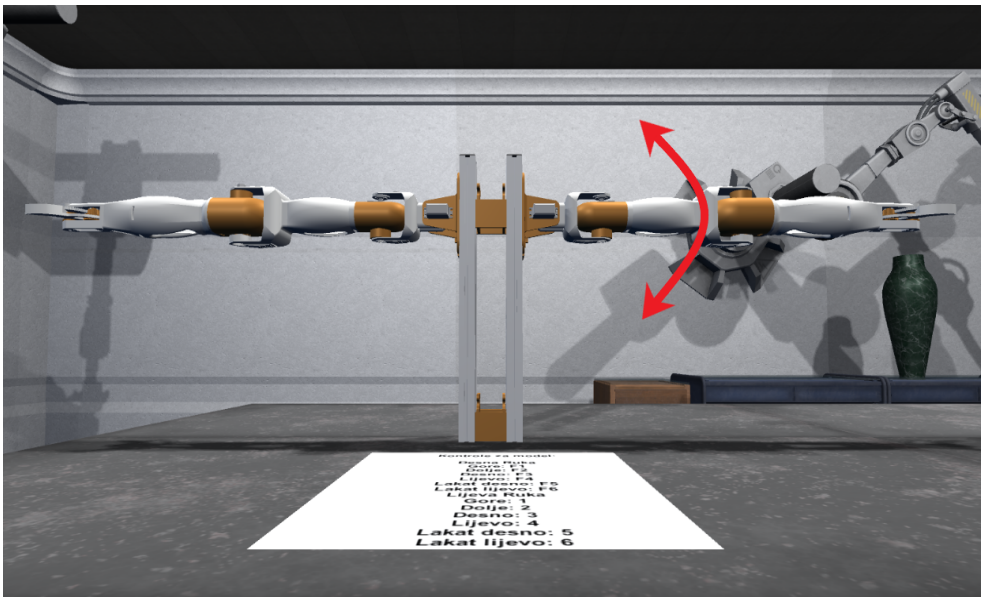
Druga skripta služi za okretanje ruke u vertikalnom smjeru.

```
VerticalMovement.cs | RotationScript.cs | HorizontalRotation.cs | UpDownMovement.cs
Miscellaneous Files                                              VerticalMovement
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class VerticalMovement : MonoBehaviour
6   {
7       private Vector3 _rotation;
8       [SerializeField] private float _speed = 1.0f;
9       public KeyCode pressUp;
10      public KeyCode pressDown;
11
12      // Update is called once per frame
13      void Update()
14      {
15          if (Input.GetKey(pressUp)) _rotation = Vector3.left;
16          else if (Input.GetKey(pressDown)) _rotation = Vector3.right;
17          else _rotation = Vector2.zero;
18
19          transform.Rotate(_rotation * _speed * Time.deltaTime);
20      }
21  }
```

*Slika 18: Skripta VerticalMovement*

Slično kao i kod horizonalne rotacije, private *Vector3 _rotation* služi za pohranu pozicije ruke. *Private float _speed* služi za postavljanje brzine kretanja ruke. *Public KeyCode pressUp* služi za postavljanje tipke na tipkovnici za kretnju u lijevom smjeru, dok *public KeyCode*

*pressDown* služi za kretnju u desnom smjeru. Ako se pritisne postavljena tipka za pokret u gornjem smjeru, pozicija *Vector3* se pomiče dokle god je tipka pritisnuta. Isto vrijedi i za tipku za donji pokret. Ako ni jedna tipka nije pritisnuta, pozicija ostaje nepromijenjena. *Transform.Rotate* služi za brzinu kretnje ruke koja je postavljena ranije u klasi.




*Slika 19: Prikaz vertikalnog kretanja ruke*

Treća i četvrta skripta su korištene kod desnog odnosno zelenog modela.

VerticalMovement.cs | RotationScript.cs | HorizontalRotation.cs | **UpDownMovement.cs**

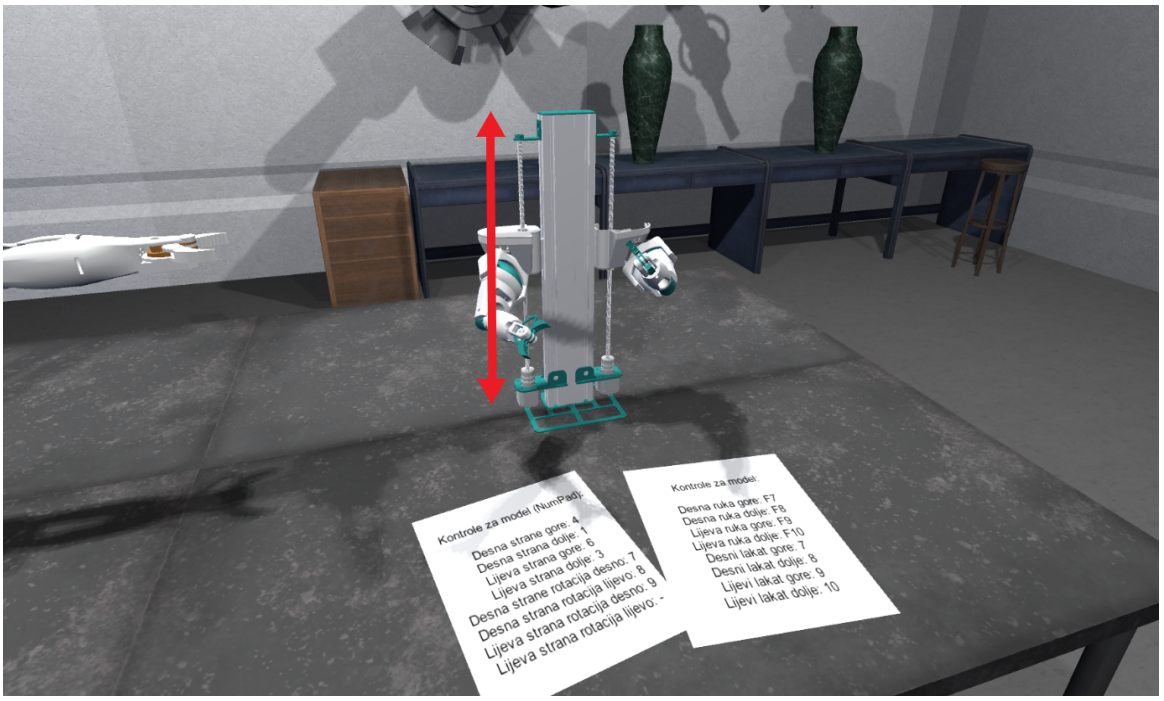
Miscellaneous Files — UpDownMovement

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class UpDownMovement : MonoBehaviour
6   {
7       private Vector3 _position;
8       [SerializeField] private float _speed = 1.0f;
9       public KeyCode pressUp;
10      public KeyCode pressDown;
11
12      // Update is called once per frame
13      void Update()
14      {
15          if (Input.GetKey(pressUp)) _position = Vector3.up;
16          else if (Input.GetKey(pressDown)) _position = Vector3.down;
17          else _position = Vector2.zero;
18
19          transform.Translate(_position * _speed * Time.deltaTime);
20      }
21  }
```

*Slika 20: Skripta UpDownMovement*

Za razliku od ostalih skripti, ova skripta nema svrhu rotiranja objekta nego služi za kretanje ruke u gornjem i donjem smjeru po tijelu modela. U *private Vector3 _position* se pohranjuje položaj ruke, dok *private float _speed* služi za određivanje brzine kretanja objekta odnosno

ruke. *Public KeyCode pressUp* i *pressDown* su postavljeni jer se objekt kreće u gornjem i donjem smjeru. Korisnik ima mogućnost kretanja ruke pomoću postavljenih tipki.




*Slika 21: Prikaz kretanja ruke u smjeru gore i dolje*

Zadnja skripta služi za rotaciju ruke u području ramena.

VerticalMovement.cs | RotationScript.cs | HorizontalRotation.cs | UpDownMovement.cs

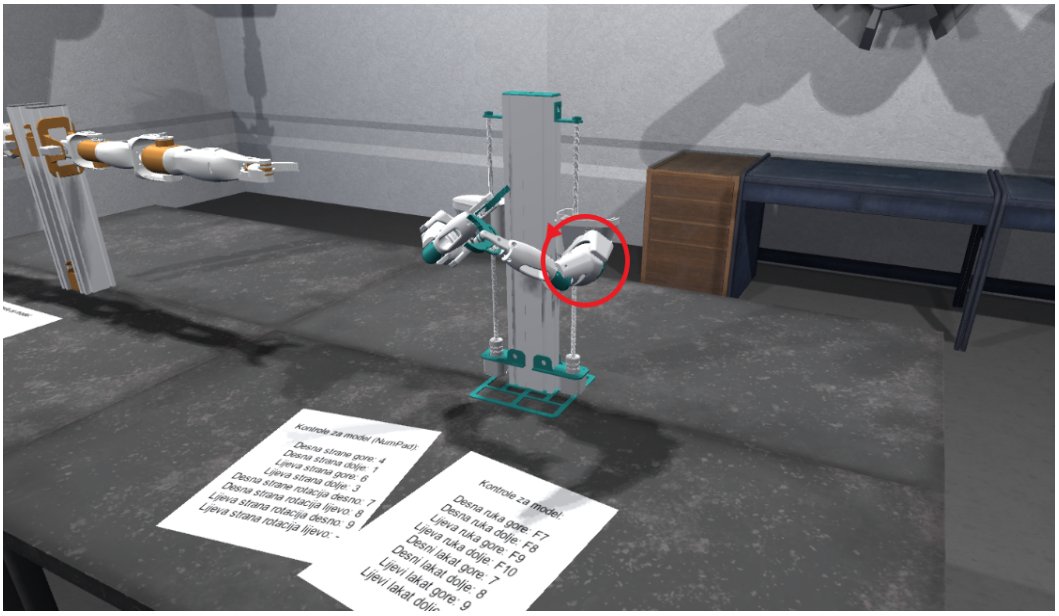
Miscellaneous Files | RotationScript

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class RotationScript : MonoBehaviour
6   {
7       private Vector3 _rotation;
8       [SerializeField] private float _speed = 1.0f;
9       public KeyCode pressLeft;
10      public KeyCode pressRight;
11
12      // Update is called once per frame
13      void Update()
14      {
15          if (Input.GetKey(pressLeft)) _rotation = Vector3.left;
16          else if (Input.GetKey(pressRight)) _rotation = Vector3.right;
17          else _rotation = Vector2.zero;
18
19          transform.Rotate(_rotation * _speed * Time.deltaTime);
20      }
21  }
```

*Slika 22: Skripta RotationScript*

U *private Vector 3 _rotation* se pohranjuje položaj objekta. *Private float _speed* određuje brzinu okretanja. *Public KeyCode pressLeft* služi za postavljanje tipke za okretanje u lijevom smjeru, dok public *KeyCode pressRight* služi za desni smjer. U *void Update* je postavljena

petlja *if, else if, else,* gdje *if* služi za kretanje objekta, odnosno *vectora* u lijevom smjeru, a *else if* u desnom smjeru. Ako niti jedna tipka nije pritisnuta pozicija *Vector-a* ostaje nepromijenjena.




*Slika 23: Prikaz rotacije ruke*

## 10. PROGRAMSKO RJEŠENJE - DODATAK

- Horizontal rotation

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

public class HorizontalRotation : MonoBehaviour
{
    private Vector3 _rotation;
    [SerializeField] private float _speed = 1.0f;
    public KeyCode pressLeft;
    public KeyCode pressRight;

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKey(pressLeft)) _rotation = Vector3.up;
        else if (Input.GetKey(pressRight)) _rotation = Vector3.down;
        else _rotation = Vector2.zero;

        transform.Rotate(_rotation * _speed * Time.deltaTime);
    }
}
```

- Vertical movement

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

public class VerticalMovement : MonoBehaviour
{
   private Vector3 _rotation;
   [SerializeField] private float _speed = 1.0f;
   public KeyCode pressUp;
   public KeyCode pressDown;

   // Update is called once per frame
   void Update()
   {
      if (Input.GetKey(pressUp)) _rotation = Vector3.left;
      else if (Input.GetKey(pressDown)) _rotation = Vector3.right;
      else _rotation = Vector2.zero;

      transform.Rotate(_rotation * _speed * Time.deltaTime);
   }
}
```

- UpDown movement

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

public class UpDownMovement : MonoBehaviour
{
   private Vector3 _position;
   [SerializeField] private float _speed = 1.0f;
   public KeyCode pressUp;
   public KeyCode pressDown;

   // Update is called once per frame
   void Update()
   {
      if (Input.GetKey(pressUp)) _position = Vector3.up;
      else if (Input.GetKey(pressDown)) _position = Vector3.down;
      else _position = Vector2.zero;

      transform.Translate(_position * _speed * Time.deltaTime);
   }
}
```

- Rotation

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

```
public class RotationScript : MonoBehaviour
{
    private Vector3 _rotation;
    [SerializeField] private float _speed = 1.0f;
    public KeyCode pressLeft;
    public KeyCode pressRight;

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKey(pressLeft)) _rotation = Vector3.left;
        else if (Input.GetKey(pressRight)) _rotation = Vector3.right;
        else _rotation = Vector2.zero;

        transform.Rotate(_rotation * _speed * Time.deltaTime);
    }
}
```

# 11. ZAKLJUČAK

U ovom radu prikazani su osnovni i napredni koncepti upravljanja robotske ruke. U kinematici robota, ruke se mogu pomicati gdje je potrebno, korištenjem rotacije karika i zglobova. Kinematika robota se dijeli na dvije vrste: kinematika naprijed i inverzna kinematika. Rad u programskom alatu Unity 3D je bio veoma intuitivan, a ako je postojao neki problem isti je bio riješen uz pomoć mentora ili pretragom problema na internetu, točnije na Unity-evim forumima i YouTube-u. Želja je bila spojiti umjetnost (srednja škola) sa informatikom (fakultetsko obrazovanje), a Unity kao programski alat pruža tu mogućnost. Međutim Unity-u upotreba nije isključivo za izradu video igara, postoje mnoge druge namjene za Unity. Unity je moćan multimedijski alat sposoban za implementaciju na gotovo svaku modernu platformu. Neke od ostalih upotreba su; interaktivnost - uređaji sa zaslonom osjetljivim na dodir su posvuda oko nas. Tradicionalne karte u trgovačkim centrima zamjenjuju interaktivne karte s dodirnim zaslonom. Za takva rješenja potrebna je bogata multimedijska platforma za rad, a ovakva rješenja postaju sve više popularna. Skalabilnost Unity-a omogućuje integraciju u sve postojeće pozadinske mreže koje neka tvrtka ima, a Unity-eve moćne grafičke mogućnosti omogućuju vizualno oduševljenje ljudi. Uz Unity i njegov *Asset Store* to postaje moguće. Radom u programu mogu se napraviti sjajne animirane sekvence koje služe za previzualizaciju u filmskoj produkciji. Arhitektonska vizualizacija - programi za video igre kao što je Unity već su sposobni nositi se s ogromnim količinama složene geometrije i dati realističan izgled rasvjete i površina. Prilično je lako uvoziti podatke izravno u Unity, a zatim ih i prikazati u programu. Umjesto da samo prikazuju koncepte i nacrte svojim klijentima, arhitekti mogu koristiti Unity za stvaranje 3D doživljaja, koji se može primijeniti na klijentov mobitel, *desktop* ili *web* preglednik. Animacija - programski alati kao što je Unity postaju ozbiljna opcija za autore animiranih filmova. Uskoro ćemo doći do točke kada nećemo moći primijetiti razliku između unaprijed generirane grafike i grafike generirane u stvarnom vremenu. Isto tako kroz nekoliko godina počet ćemo vidjeti potpuno novu vrsta snimanja filmova, sve uz pomoć VR tehnologije. Gledatelj će postati dio filma jer će biti u središtu događanja. Renderiranje u stvarnom vremenu ključno je za to. Simulacije - video igre više nisu samo izvor zabave. Mogu se ozbiljno koristiti za obuku u medicinskoj i vojnoj industriji. Suočavanje s različitim scenarijima prije nego što se dogode u pravom životu poboljšava učinkovitost, a proširivost, realistične grafičke mogućnosti i jednostavnost korištenja čine ga idealnim za izradu industrijskih simulatora.

LITERATURA

ERASMUS+ PROJECT 3D and Virtual Reality Technologies for VET, projektni broj: 2019-1-HR01-KA202-061006, Rad na računalnim modelima virtualne stvarnosti (VR) (Ivan Hanuljak, matični broj: 0303061325)

INTERNET IZVORI

https://www.tutorialspoint.com/computer_graphics/computer_graphics_quick_guide.htm („Computer Graphics - Quick Guide“, pristupljeno: 12.7.2022.)
https://www.geeksforgeeks.org/introduction-to-computer-graphics/ (“Introduction to Computer Graphics”, pristupljeno: 12.7.2022.)
https://www.explainthatstuff.com/computer-graphics.html (“Computer graphics”, pristupljeno: 12.7.2022.)
https://www.javatpoint.com/computer-graphics-tutorial (“Computer Graphics Tutorial”, pristupljeno: 12.7.2022.)
https://www.technicalfoamservices.co.uk/blog/blog-history-of-cad-cam/ („The history of computer-aided design and computer-aided manufacturing (CAD/CAM)“, pristupljeno: 25.6.2022.)
https://medium.com/technical-illustration/evolution-of-cad-from-light-pens-to-synchronous-technology-549cc8eef5d0 („Evolution of CAD — From light pens to Synchronous Technology!“, pristupljeno: 25.6.2022.)
https://surveyinggroup.com/what-is-3d-visualization/ (“What is 3D Visualization?”, pristupljeno: 30.8.2021.)
https://www.alltechbuzz.net/hr/2d-ili-3d-koji-bi-smjer-trebao-odabrati-po%C4%8Detnik-unity-developer-igre/ (“2D ili 3D: koji bi smjer trebao odabrati početnik Unity programer igara”, pristupljeno: 25.6.2022.)
https://code.tutsplus.com/tutorials/introduction-to-unity3d--mobile-10752 (“Introduction to Unity3D”, pristupljeno: 25.6.2022.)
https://conceptartempire.com/what-is-unity/ (“What is Unity 3D & What is it Used For?”, pristupljeno: 25.6.2022.)
https://it-s.com/what-is-unity-3d/ (“What Is Unity 3D?”, pristupljeno: 25.6.2022.)
https://www.androidauthority.com/what-is-unity-1131558/ (“What is Unity? Everything you need to know”, pristupljeno: 25.6.2022.)

POPIS SLIKA

Slika 1: CRT monitor s katodnom cijevi: https://www.javatpoint.com/computer-graphics-cathode-ray-tube (Pristupljeno: 25.6.2022.) 2
Slika 2: Crtanje na PDP Type 30: https://www.computerhistory.org/collections/catalog/102652246 (Pristupljeno: 25.6.2022.) 3
Slika 3: Prikaz renderiranja slike: https://www.graphics.cornell.edu/online/tutorial/objrender/ (Pristupljeno: 25.6.2022.) 4
Slika 4: Računalno generirana slika: https://www.pinterest.com/pin/784611566326521649/ (Pristupljeno: 25.6.2022.) 5
Slika 5: Prikaz sučelja CAD alata Fusion 360-a 7
Slika 6: Prikaz šahovske ploče u Autodesk Viewer-u 9
Slika 7: Autodesk Fusion 360 11
Slika 8: Unity logo: https://1000logos.net/unity-logo/ (Pristupljeno: 3.9.2022.) 12
Slika 9: Unity Hub 13
Slika 10: Unity Sučelje 16

Slika 11: Rad u Unity-u 16
Slika 12: Prikaz modela 18
Slika 13: Prikaz lijevog (narančastog) modela 19
Slika 14: Prikaz desnog (zelenog) modela 20
Slika 15: Okružje Free Sci-Fi Office Pack: https://assetstore.unity.com/packages/3d/environments/sci-fi/free-sci-fi-office-pack-195067 (Pristupljeno: 25.6.2022.) 21
Slika 16: Skripta HorizontalRotation 21
Slika 17: Prikaz horizontalnog kretanja ruke 22
Slika 18: Skripta VerticalMovement 22
Slika 19: Prikaz vertikalnog kretanja ruke 23
Slika 20: Skripta UpDownMovement 23
Slika 21: Prikaz kretanja ruke u smjeru gore i dolje 24
Slika 22: Skripta RotationScript 24
Slika 23: Prikaz rotacije ruke 25

SAŽETAK

Rad robotske ruke može se simulirati pomoću računala. Ovaj rad pokazuje kako pojednostaviti komplicirane funkcije i važnost programskih pravila. Računalno okruženje (sučelje i programski jezik) razvijeno je za simulaciju robotske ruke korištenjem Unity *Enginea* i C# za programiranje. Ovaj rad objašnjava preduvjete i razvoj projekta od početka do kraja. Daje se objašnjenje korištenih metoda i napisanih skripti. Također je predstavljen alat koji je korišten u radu. Unity 3D je višeplatformski programski alat koji je razvio Unity Technologies, a koji se prvenstveno koristi za razvoj videoigara, simulacija za računala, konzole i mobilne uređaje. Brzo se razvio u najkorišteniji alat te je od 2005. godine proširen na 27 platformi. Iskustvo sa razvojnim alatom Unity 3D je pozitivno, jer sadrži puno materijala i praktičnih primjera koji pomažu u radu. Kada se i našilo na problem, uz nekoliko pretraga pronađeno je rješenje. Sve to je moguće sa velikom zajednicom koja je spremna pitati, a i pomoći na svako pitanje pojedinca. Unity kao programski alat je jednostavan za korištenje i korisnikova kreativnost se lako može prenijeti u bilo koji projekt.

Ključne riječi: Unity 3D, CAD alati, Računalna grafika

SUMMARY

The operation of the robotic arm can be simulated using a computer. This paper shows how to simplify complicated functions and the importance of programming rules. A computing environment (interface and programming language) was developed to simulate the robotic arm using Unity Engine and C# for programming. This paper explains the prerequisites and development of the project from start to finish. An explanation of the methods used and the scripts written is given. The tool used in the work is also presented. Unity 3D is a cross-platform programming tool developed by Unity Technologies, which is primarily used for the development of video games, simulations for PCs, consoles and mobile devices. It quickly developed into the most used tool and since 2005 has been expanded to 27 platforms. The experience with the development tool Unity 3D is positive, because it contains a lot of materials and practical examples that help in the work. When a problem was encountered, a solution was found after several searches. All this is possible with a large community that is ready to ask, and also help with every question of an individual. Unity as a programming tool is easy to use and the user's creativity can easily be transferred to any project.

Keywords: Unity 3D, CAD tools, Computer graphics