SVEUČILIŠTE U SPLITU

FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

MEASUREAR APLIKACIJA

Ana Šundov

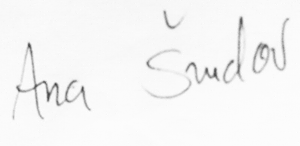
Split, srpanj 2018.

**IZJAVA**

Ovom izjavom potvrđujem da sam završni rad s naslovom MEASUREAR aplikacija pod mentorstvom prof. dr. sc. Mirjane Bonković pisala samostalno, primijenivši znanja i vještine stečene tijekom studiranja na Fakultetu elektrotehnike, strojarstva i brodogradnje, kao i metodologiju znanstveno-istraživačkog rada, te uz korištenje literature koja je navedena u radu. Spoznaje, stavove, zaključke, teorije i zakonitosti drugih autora koje sam izravno ili parafrazirajući navela u završnom radu citirala sam i povezala s korištenim bibliografskim jedinicama.

Studentica

Ana Šundov



SVEUČILIŠTE U SPLITU

FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE

Preddiplomski studij: Fakultet Elektrotehnike, Strojarstva i Brodogradnje

Smjer/Usmjerenje: Komunikacijska i informacijska tehnologija

Oznaka programa: 114

Akademska godina: 2017./2018.

Ime i prezime: **Ana Šundov**

Broj indeksa: 825-2015

**ZADATAK ZAVRŠNOG RADA**

Naslov: **MEASUREAR** **aplikacija**

Zadatak: Upoznati se s teoretskim temeljima vezanim za funkcionalnost uređaja proširene stvarnosti. Ovladati osnovama korištenja ARCore biblioteke za kreiranje aplikacija proširene stvarnosti te razviti mobilnu aplikaciju za precizno mjerenje prostornih veličina.

Prijava rada: 05.03.2018.

Rok za predaju rada: 21.09.2018.

Rad predan: 18.07.2018.

Datum obrane: 23.07.2018.

Mentor:

**prof. dr. sc. Mirjana Bonković**

Sadržaj

[1. UVOD 1](#_Toc519529787)

[2. PROŠIRENA STVARNOST 2](#_Toc519529788)

[2.1. Cilj proširene stvarnosti 2](#_Toc519529789)

[2.2. Osnovna podjela proširene stvarnosti 2](#_Toc519529790)

[2.3. Komponente proširene stvarnosti 5](#_Toc519529791)

[2.4. Proširena stvarnost vs. Virtualna stvarnost 6](#_Toc519529792)

[3. PLATFORME PROŠIRENE STVARNOSTI 7](#_Toc519529793)

[3.1. Wikitude 7](#_Toc519529794)

[3.2. ARKit 8](#_Toc519529795)

[3.3. DeepAR 8](#_Toc519529796)

[3.4. ARToolKit 9](#_Toc519529797)

[3.5. Vuforia 10](#_Toc519529798)

[3.6. Tango 10](#_Toc519529799)

[4. ARCORE PLATFORMA 11](#_Toc519529800)

[4.1. Princip rada ARCore platforme 11](#_Toc519529801)

[4.2. Komponente 12](#_Toc519529802)

[4.2.1. Razumijevanje okoline 12](#_Toc519529803)

[4.2.2. Praćenje pokreta 12](#_Toc519529804)

[4.2.3. Procjena svjetlosti 13](#_Toc519529805)

[4.3. Dijeljenje 13](#_Toc519529806)

[4.4. Razvojna okruženja 13](#_Toc519529807)

[4.5. Podržani uređaji 14](#_Toc519529808)

[5. UNITY 16](#_Toc519529809)

[5.1. Općenito u Unity-u 16](#_Toc519529810)

[5.2. Instalacija 18](#_Toc519529811)

[5.3. Sučelje Unity-a 18](#_Toc519529812)

[5.4. Integracija Android SDK za Unity 20](#_Toc519529813)

[5.5. Koordinatni sustav Unity-a 22](#_Toc519529814)

[6. MEASUREAR APLIKACIJA 23](#_Toc519529815)

[6.1. HelloAR 23](#_Toc519529816)

[6.2. HelloAR Controller skripta 26](#_Toc519529817)

[6.3. MeasureAR razvoj aplikacije 27](#_Toc519529818)

[6.3.1. Sfera 27](#_Toc519529819)

[6.3.2. Linija 30](#_Toc519529820)

[6.3.3. Example Controller skripta 31](#_Toc519529821)

[6.3.4. Tekst 32](#_Toc519529822)

[6.3.5. ARCore okruženje 35](#_Toc519529823)

[7. ZAKLJUČAK 37](#_Toc519529824)

[8. LITERATURA 38](#_Toc519529825)

[9. SAŽETAK 39](#_Toc519529826)

[10. SUMMARY 40](#_Toc519529827)

[11. POPIS KRATICA I REFERENCA 41](#_Toc519529828)

# UVOD

Sve veća želja korisnika za lakšim i jednostavnijim načinom obavljanja različitih poslova dovodi do procvata u razvoju aplikacija proširene stvarnosti. Mogućnost integriranja virtualnih sadržaja u stvarni svijet omogućava projiciranje različitih sadržaja prije same izvedbe te tako olakšava zadatke vezane uz svakodnevni život. Osnovni cilj aplikacija za proširenu stvarnost je neprimjetno integriranje virtualnih objekata u stvarnost kako bi se korisniku omogućio potpuni osjećaj lažne stvarnosti.

Cilj ovog završnog rada je kreirati aplikaciju koja će omogućiti korisniku mjerenja dimenzija prostora pomoću ARCore platforme. Korisnik pomoću kamere mobilnog uređaja detektira ravne površine i doticanjem ekrana u više različitih točaka pokreće program izračunavanja udaljenosti između dotaknutih točki prostora. Aplikacija će biti kreirana unutar Unity Engine razvojnog okruženja verzije 2017.4.3f1 koristeći C++ programski jezik.

Kako bi razvoj aplikacije i njen način rada bio što jasniji, u idućem poglavlju opisana je proširena stvarnost kao glavna motivacija izrade ovakvog tipa aplikacije. Treće poglavlje nas uvodi u neke od najpoznatijih platformi proširene stvarnosti pokazujući nam koliko popularnost proširene stvarnosti raste iz dana u dan. Četvrto poglavlje se koncentrira na ARCore platformu koju smo koristili za izradu aplikacije. Upoznati ćemo se s principom rada platforme, njenim komponentama, razvojnim okruženjima i uređajima koje ARCore podržava. Peto poglavlje nam predstavlja Unity razvojno okruženje, njegov izgled, funkcionalnost i postupak same instalacije. Zadnje i najbitnije poglavlje nas uvodi u sam razvoj MeasureAR aplikacije uključujući njene komponente i HelloAR osnovne aplikacije.

# PROŠIRENA STVARNOST

Proširena stvarnost je jedna od najrazvijenijih tehnologija u današnje vrijeme. Iako usko povezana s virtualnom stvarnošću, proširena stvarnost koristi stvarnu okolinu koju izmjenjuje i nadopunjuje virtualnim sadržajima. Tehnike proširene stvarnosti se izvode u realnom vremenu i semantičkom kontekstu s elementima iz okoliša.

Prvi funkcionalni sistemi koji su integrirali virtualne dijelove u stvarni svijet pojavili su se početkom 1990ih kada je američka zračna vojska osmislila sistem virtualnog čvorišta [1]. Takav sistem je prikupljao informacije o korisnikovoj percepciji stvarnog okruženja i tako poboljšavao ljudski učinak u izravnim i daljinski manipuliranim zadacima. Tijekom godina proširena stvarnost se počela razvijati i iako najčešće korištena u zabavne svrhe, danas ima široku primjenu u obrazovanju, medicini, industriji i sličnim područjima o čemu je detaljnije pisano u potpoglavlju

## Cilj proširene stvarnosti

Cilj proširene stvarnosti je integriranje virtualnog sadržaja u stvarni svijet na što realniji način. Različitim metodama kompjuterske vizije stvarnosti softver određuje koordinate okoline te tako stvara sebi razumljivu sliku. Ovakav proces se sastoji od dva dijela gdje se prvo detektiraju zanimljive točke, markeri ili kretnje a zatim se određuju same koordinate stvarne okoline. Ovakva metoda izračunavanja je preuzeta iz vizualne odometrije[[1]](#footnote-1)

## Osnovna podjela proširene stvarnosti

Postoji nekoliko kategorija proširene stvarnosti razlikujući se u svojim ciljevima i primjenama [2].

1. Marker-based AR[[2]](#footnote-2) koristi se kamerom i vizualnim markerom kako bi davala rezultate samo u slučajevima kada je marker aktiviran od strane čitatelja. Najčešće korišteni oblik ovakve proširene stvarnosti je skeniranje kodova pomoću mobilne kamere prikazano na slici (Slika 2‑1). Kamera detektira površinu, prepoznaje kod i dopušta uređaju daljnji rad u aplikaciji.



Slika 2‑1 Skeniranje koda

1. Markerless AR jedna je od najpopularnijih oblika proširene stvarnosti najčešće korištena kao GPS, digitalni kompas, mjerač brzine ili kao akcelerometar. Bitna stavka ovog oblika proširene stvarnosti je velika dostupnost i rasprostranjenost mobilnih uređaja koji u sebi imaju ugrađene senzore pomoću kojih je moguće pratiti trenutnu lokaciju ili druge željene komponente. Prikaz upotrebe GPS-a nalazi se slici ispod (Slika 2‑2).



Slika 2‑2 GPS

1. Projection Based AR radi tako da projicira umjetno stvorenu svjetlost na stvarnu površinu te detektirajući ljudski dodir, vrši daljnje operacije. Najčešće korišteni primjer ovakvog tipa proširene stvarnosti su virtualne tipkovnice prikazane na slici. (Slika 2‑3).



Slika 2‑3 Virtualna tipkovnica

1. Superimposition Based AR djelomično ili potpuno zamjenjuje izvorni prikaz objekta s novim povećanim prikazom tog istog objekta. Najpoznatija aplikacija koja koristi ovaj tip AR je Ikeina aplikacija za razmještaj namještaja prikazana na slici. (Slika 2‑4)



Slika 2‑4 Ikea aplikacija

## Komponente proširene stvarnosti

Komponente za implementiranje virtualnih sadržaja u stvarni svijet mogu se podijeliti na hardverski i softverski dio.

Hardverske komponente proširene stvarnosti su: procesor, prikazni uređaj i senzor. Njih, u današnje vrijeme, podržava svaki mobilni i računalni uređaj te tako omogućava jednostavan pristup ovoj tehnologiji kao i njen daljnji razvoj.

Tehnologije korištene za prikaz su mnogobrojne, sadržavajući različite optičke projekcijske sustave, monitore i ručne uređaje. Neke od njih su prokomentirane u nastavku [1]:

* Naglavni ekran (HDM[[3]](#footnote-3)) je prikazni uređaj oblika kacige koji, stavljen u vidno polje korisnika, projicira realnu sliku integriranu s virtualnim komponentama. Noviji modeli imaju mogućnost prilagođavanja slike rotaciji i kutu korisnikova vidnog polja.
* AR NAOČALE koriste kamere za presretanje pogleda u stvarni svijet, prikazuju proširenu sliku kroz okulare i uređaje u kojima se virtualne sliku projiciraju ili reflektiraju s površine naočala.
* RUČNI ZASLON (eng. Handheld) ručni je uređaj veličine dlana korišten za prikaz proširene stvarnosti. Zbog svoje veličine omogućava lagan prijenos i dostupnost okoline korisniku. Najčešće korišten u obliku mobilnog uređaja, putem kamere omogućava korisniku uvid u stvarni svijet integriran s virtualnim sadržajima.

Budući da se tijekom godina razvoj aplikacija proširene stvarnosti naglo povećao, pojavili su se i novi programi za razvoj softvera. Neki od najpoznatijih softvera su Unreal Engine i Unity koji je korišten prilikom izrade MeasureAR aplikacije. Razvoj aplikacije, kao i sam Unity predstavljen je u poglavlju 6.

## Proširena stvarnost vs. Virtualna stvarnost

Virtualna stvarnost je oblik računalne simulacije realnog svijeta čiji je cilj dizajnirati sučelje u kojem je korisniku omogućena interakcija s okolinom u virtualnom svijetu jednaka interakciji s okolinom u realnom svijetu. Najčešće se dostavlja korisniku putem ručnog kontrolera ili kontrolera u obliku kacige za glavu. Ova oprema povezuje korisnika s virtualnom stvarnošću dopuštajući im da kontroliraju i određuju kretnje svojih radnji u okruženju koje simulira realni svijet. Najčešće je korišten u zabavne svrhe kao što su video igrice i društvene mreže.

Za razliku od virtualne stvarnosti, proširena stvarnost ne stvara novo, virtualno okruženje već u realnom okruženju stvara virtualne elemente. Omogućuje korisniku bolji doživljaj svijeta u kojem virtualni elementi poboljšavaju i pomažu u čovjekovoj svakodnevici. Koristi se uređajima kao što su mobiteli, laptopi, tableti i slični uređaji kako bi utjecala na interakciju između realnog i digitalnog virtualnog svijeta.

Iako je virtualna stvarnost dosegla veliki uspjeh na području zabavnih sadržaja kao što su video igrice, proširena stvarnost s raznovrsnijom primjenom te samim tim, bržim razvojem omogućava korisniku bolja i korisnija iskustva pogodna za daljnji razvoj i olakšavanje budućnosti.

# PLATFORME PROŠIRENE STVARNOSTI

Kao jedan od najpopularnijih načina spajanja virtualne i realne stvarnosti, proširena stvarnost je tijekom niza godina na tržištu dobila mnogobrojne platforme pogodne za izradu kako mobilnih, tako i programa za dekstop računala. U nastavku su prokomentirane neke od najpoznatijih platformi AR.

## Wikitude

Wikitude je trenutno najbolja mobilna platforma proširene stvarnosti u svijetu. Osnovana 2008. godine u Salzburgu, Austriji, Wikitude se prvenstveno fokusirala na AR baziranu na lokaciji uređaja no 2012. godine objavljuju Wikitude SDK , okružje (eng. framework) koje koristi prepoznavanje i praćenje slika, kao i geolokacijsku tehnologiju[3].

Wikitude je dostupan putem SDK (softvera), Cloud prepoznavatelja koji prepoznaje nekoliko tisuća slika i Wikitude studija koji omogućava stvaranje novog sadržaja metodom prenošenja elemenata na površinu.

Najbitnije značajke Wikitude platforme su:

* Mogućnost prepoznavanja objekta u stvarnom vremenu i prostoru prethodno definiranom od strane korisnika
* Praćenje scene u kojoj se mogu pratiti veći objekti i prostorije. Ova značajka bazirana je na SLAM[[4]](#footnote-4) uređaju koji je korišten za instant praćenje.
* Instant praćenje omogućava integriranje virtualnih sadržaja na realne površine bez korištenja markera.
* Prepoznavanje više od 1000 slika izvan mreže i neprimjetno prebacivanje s jedne lokacije na drugu
* Prošireno praćenje koje dopušta korisniku usidrivanje objekta na određenu lokaciju i daljnji nesmetani rad na platformi iako fokus nije na objektu.

Wikitude je dostupan za sve Android, Apple i Windows uređaje te podržava okružja kao što su: Unity 3D, Cordova, Appcelerator Titanium i Xamarin.

## ARKit

ARKit je Apple-ova platforma za izradu aplikacija proširene stvarnosti. Iako jednostavnijeg hardware-a, ARKit je nastao kao odgovor na Google-ov project Tango 3.6.

Princip rada ARKit-a zasniva se na 3 komponente[4]:

* Dubinska kamera - Koristeći ovaj oblik kamere, aplikacija detektira poziciju u prostoru, topologiju i ekspresije na ljudskom licu. Ovakvi procesi se odvijaju u realnom vremenu što omogućava aplikacijama selfie efekte ili korištenje ljudskih odraza za stvaranje 3D karaktera.
* Vizualna unutarnja odometrija (eng. Visual Inner Odometry) - Koristeći se kombinacijom senzora kamere i Core Motion datoteke, VIO[[5]](#footnote-5) prati svijet oko sebe te tako omogućava osjet kretnje uređaja po prostoru u realnim omjerima.
* Razumijevanje scene i procjena osvjetljenja- Pomoću scene prikazane kroz kameru uređaja, uređaj pronalazi vertikalne i horizontalne površine u prostoriji koje može pratiti ili postavljati objekte na njih. Isto tako, ima mogućnost procjene svjetlosti i postavljanja unaprijed određenog osvjetljenja na scenu.

ARKit podržava Apple A9, A10 i A11 procesore. Ti procesori mogu u kratkom vremenu shvatiti okolinu u kojoj se nalaze te u njoj izgraditi i pokretati virtualne sadržaje. ARKit se može optimizirati u programima kao što su Metal, SceneKit ili u odvojenim okružjima kao što su Unity i Unreal.

## DeepAR

DeepAR platforma koristi se za detekciju lica u stvarnom vremenu na temelju patentiranih modela podataka i tehnika strojnog učenja. Brzog rada i mogućnosti prepoznavanja do 70 točaka lica u 60 okvira po sekundu, DeepAR izvršava precizan prikaz slika optimiziran za mobilne i web aplikacije[5].

DeepAR softver sadrži 4 tipa efekta: kruti objekti, deformabilne maske, morfne maske i efekte nakon obrade koji omogućavaju korisnicima detektiranje lica visoke kvalitete kao i nadodavanje različitih maski i efekata sličnih onima koje koriste aplikacije poput Facebook-a i Snapchat-a kao što je prikazano na slici (Slika 3‑1).



Slika 3‑1 DeepAR efekti i maske

## ARToolKit

ARToolKit je kompjuterska platforma za kreiranje aplikacija proširene stvarnosti koje preslikavaju virtualne slike u stvarni svijet. Razvijena 1999.godine na Washingtonskom fakultetu, danas je jedna od poznatijih platformi s preko 160 tisuća preuzimanja od najnovije verzije objavljene 2004.godine. ARToolKit se trenutno održava kao open-source[[6]](#footnote-6) platforma na GitHub-u i podržava Windows, Mac OS, Linux, iOS i Android [6].

Neke od glavnih značajki ove platforme su praćenje pozicije i orijentacije samostalne ili stereo kamere, praćenje jednostavnih crnih kocki i planarnih slika, kalibracija kamere i optička stereo kalibracija, HMD podrška i mnoge druge. Ovakva raznolikost funkcija otežava integraciju platforme kao i vremenski period izvršavanja funkcija.

## Vuforia

Vuforia je jedna od najpopularnijih platformi proširene stvarnosti. Sadrži funkcionalnosti kao što su: prepoznavanje različitih tipova vizualnih objekata ( kutija, cilindar, ravne površine), prepoznavanje teksta i okoline i VuMark[[7]](#footnote-7) koji se mogu provesti korištenjem baze podataka. Isto tako, koristeći Vuforia Objekt Skener moguće je skenirati i kreirati različite objekte.

Vuforia nudi aplikacijska sučelja za programiranje (API[[8]](#footnote-8)) u C + +, Java, Objective-C++ i. NET jezika kroz proširenje Unity-a. Isto tako podržava širok spektar mobilnih uređaja uključujući iPhone, iPad, Android i tablet s Android OS 2.2. ili višom verzijom.

Svi dodatci i funkcionalnosti platforme mogu se koristiti, ali uključuju vodene žigove Vuforije. Ograničenja se odnose samo na broj korištenja VuMark-a i broj prepoznavanja koje se izvodi preko Clouda [5].

## Tango

3 godine prije izdavanja ARCore-a, Google je objavio prvu platform za proširenu stvarnost nazvanu Projekt Tango. Koristeći računalni vid, omogućio je mobilnim uređajima da detektiraju relativnu poziciju obzirom na stvarni svijet ne koristeći GPS i slične signale.

Ova aplikacija je omogućila korisnicima 3D mapiranje, navigaciju u zatvorenom prostoru, mjerenje prostora, prepoznavanje okoline, proširenu stvarnost i slično [7].

Softver se zasniva na 3 komponente:

1. Praćenje pokreta
2. Učenje okoline i stvaranje modela okoline razumljivog računalu
3. Shvaćanje dubine prostora

Platforma je službeno ugašena 01.03.2018. godine zbog dovođenja nove platforme na tržište – ARCore-a o kojem je detaljnije pisano u idućem poglavlju. (4)

# ARCORE PLATFORMA

ARCore je Google-ova platforma za stvaranje proširene stvarnosti. Koristeći različite API, ARCore omogućava mobitelu osjet okoline, razumijevanje svijeta i interakciju s informacijama. Detekcijom stvarne okoline korisnik može mijenjati percepciju stvarnosti i dodavati virtualne 2D i 3D sadržaje kao što su riječi, objekti i znakovi. Tako nastale scene su pohranjene i imaju mogućnost višestrukog korištenja od strane različitih korisnika[8].

## Princip rada ARCore platforme

Općenito gledano, ARCore radi dvije stvari:

* prati poziciju mobilnog uređaja u pokretu
* stvara vlastito razumijevanje stvarnog svijeta

ARCore tehnologija za praćenje koristi mobilnu kameru kako bi identificirala značajne točke i pratila njihovu kretnju kroz vrijeme. Kad korisnik dotakne ili na bilokoji način posreduje s ekranom mobilnog uređaja, ARCore poprima koordinate te točke i projicira zraku na tu koordinatu preko kamere u stvarni svijet.. Kombinacijom pokreta ovih točaka i očitavanja s unutarnjih mobilnih senzora, ARCore određuje poziciju i orijentaciju mobitela u prostoru.

Dodatno, Arcore može detektirati ravne površine kao što su stol ili pod i procijeniti prosječno osvjetljenje u prostoriji i tako omogućiti ARCore-u stvaranje vlastitog razumijevanja realnog svijeta[8].

Ako se želi odrediti kut pod kojim se nalazi neki objekt koriste se orijentirane točke. Kad se izvrši hit testing [[9]](#footnote-9)koji vraća značajnu točku, ARCore će potražiti obližnje značajne točke i tako pokušati procijeniti kut površine na danoj značajnoj točki. ARCore će tada vratiti pozu koja uzima u obzir taj kut. S obzirom na to da ARCore koristi klastere za detektiranje kuta površine, moguće je da površine bez teksture, kao što je bijeli zid, ne budu detektirane pravilno.

## Komponente

ARCore koristi 3 ključna svojstva kako bi integrirao virtualni sadržaj sa stvarnim svijetom preko mobilne kamere[8]:

### Razumijevanje okoline

ARCore konstantno poboljšava svoje shvaćanje realnog okruženja u svijetu detektirajući značajke točaka i površine. Traži klastere značajnih točaka koje se prikazuju kao da leže na horizontalnim ili vertikalnim površinama kao što su na primjer stolovi ili zidovi, te čini te površine dostupne aplikaciji kao i granice svih detektiranih površina. Ove informacije se mogu koristiti za stavljanje mirujućih virtualnih objekata na ravne površine.

### Praćenje pokreta

Dok se mobitel pomiče, ARCore temeljem COM [[10]](#footnote-10)procesa (istovremeno izvođenje odometrije i mapiranja) procjenjuje položaj mobitela u odnosu na objekte prostora. ARCore zapravo detektira značajke u snimljenoj slici, i prati njihove pomake. Za određivanje položaja stvarnih točaka u prostoru, najprije se te vizualne informacije uparuju s inercijskim mjerenjima koje daje IMU uređaj kako bi se procijenila pozicija (položaj i orijentacija) fotoaparata u odnosu na svijet tijekom vremena.

Usklađivanjem poza (položaja i orijentacije) virtualne kamere koja prikazuje modelirani 3D sadržaj s pozadinom fotoaparata uređaja ARCore (koji prikazuje stvarni svijet), programeri mogu prikazati virtualni sadržaj iz točne perspektive. Prikazana virtualna slika može natkriti sliku iz fotoaparata (ili njen dio) mobitela, što izgleda kao da je virtualni sadržaj dio stvarnog svijeta.

### Procjena osvjetljenja

Ova komponenta se bavi očitavanjem intenziteta piksela kamere, procjenom svjetlosti i skrivanjem 3D objekata. Procjena svjetlosti se sastoji od 3 svojstva: korekcija boje, stanje i intenzitet piksela. Ti podaci omogućuju osvjetljenje virtualnih objekata pod istim uvjetima kao i okruženje oko njih, što povećava osjećaj stvarnosti.

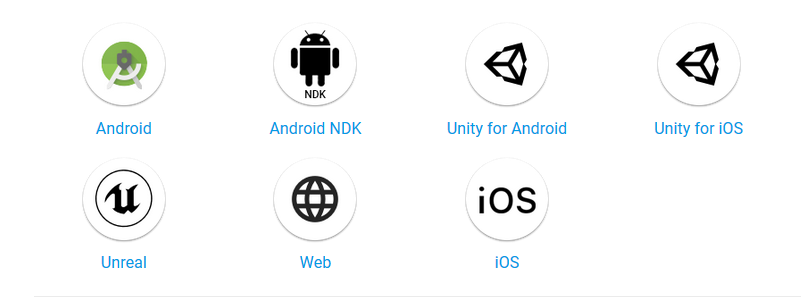
## Dijeljenje

Za stvaranje aplikacija koje dijele podatke ili imaju više učesnika, označeni položaj (sidro), aplikacije mogu dijeliti preko tzv. sidra u oblaku (Cloud anchors) koji čuva položaj sidra i značajke okolnih točaka. Ova se sidra mogu dijeliti s drugim korisnicima koji koriste isto okruženje neovisno je li mu pristupaju preko iOS ili android uređaja. Na taj način, aplikacije mogu prikazivati uvijek iste objekte vezana za sidra pa isti virtualni doživljaj može dijeliti više korisnika istovremeno.

[8].

## Razvojna okruženja

ARCore pruža SDK[[11]](#footnote-11) za brojna popularna razvojna okruženja prikazana na slici (Slika 4‑1). On sadržava izvorne API za sve bitne AR značajke kao što su praćenje pokreta, razumijevanje okoliša i procjena svjetlosti. Pomoću toga mogu se izraditi potpuno nova iskustva s AR ili poboljšati postojeće aplikacije s AR značajkama[8].



Slika 4‑1 ARCore razvojna okruženja

## Podržani uređaji

Arcore je dizajniram da radi na širokom opsegu Android mobitela sa sustavom Android 7.0.0 (Nougat) i više. Lista podržanih uređaja nalazi se na slici (Slika 4‑2)[9]:



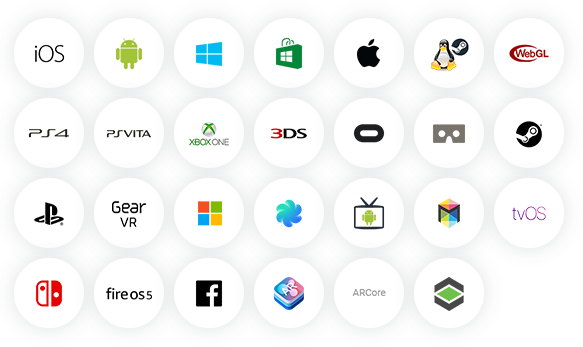
Slika 4‑2 ARCore podržani uređaji

# UNITY

Unity je razvojno okruženje za video igre na različitim platformama koji je razvila tvrtka Unity Technologies. Osnovni fokus je na razvoju 2D i 3D video igara, te simulacija za računala, konzole i mobilne uređaje. Snažna grafika i editor sa svim potrebnim značajkama opravdavaju ulogu najpopularnije platforme za izradu videoigara.

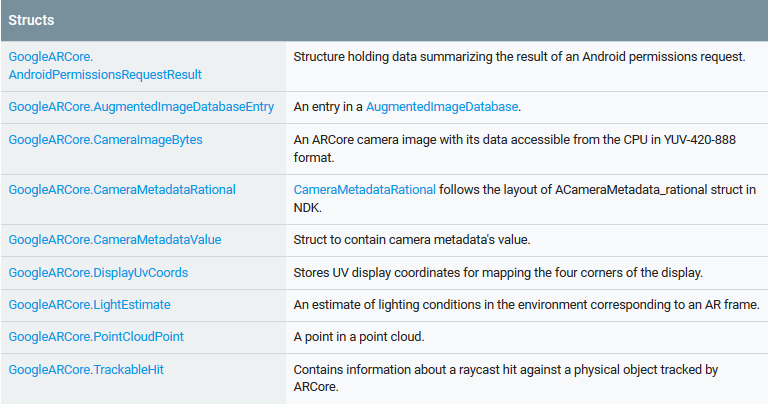
## Općenito u Unity-u

Lako je poveziv s računalima, mobilnim uređajima, kućnim sistemima i sličnim. Unity podržava 27 različitih platformi (Slika 5‑1) kao sto su: Android, ARCore, Windows, IOS ..

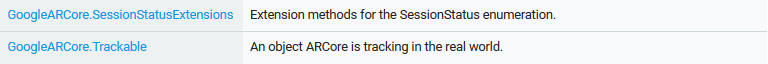
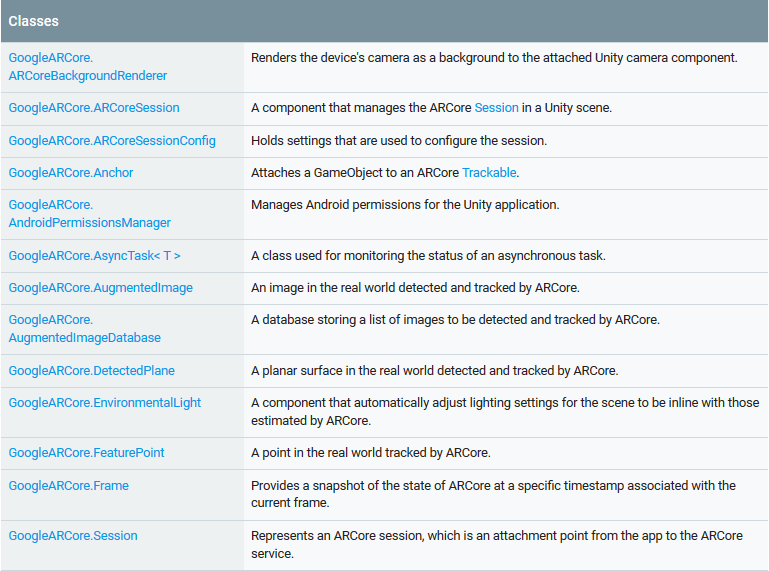


Slika 5‑1 Unity podržane platforme

Unity API reference pokrivaju neke od temeljnih područja Unity sistema te tako korisniku omogućavaju lako korištenje gotovih funkcija i događaja koje se nalaze u sklopu njih. Jednako tako, Google je napravio ARCore podržane verzije istih klasa i struktura prikazane na slikama ispod. (Slika 5‑2,Slika 5‑3)



Slika 5‑2 Unity strukture



Slika 5‑3 Unity klase

## Instalacija

Kako bi koristili ARCore platformu prilikom izrade aplikacije, potrebno je instalirati Unity paket s Android Build Support komponentom koju izabiremo tijekom instalacije. Preporučena verzija Unity-a je Unity 2017.4 LTS ili viša, a minimalna verzija je 2017.3.0f2. Nadalje, potreban je ARCore SDK za Unity koji se može preuzeti sa službene stranice ARCore-a kao arcore-unity-sdk-v\*.unitypackage datoteka[11]. Kako bi mogli povezati mobilni uređaj s aplikacijom, potrebno je preuzeti Android SDK 7.0 ( API level 24) ili noviji.

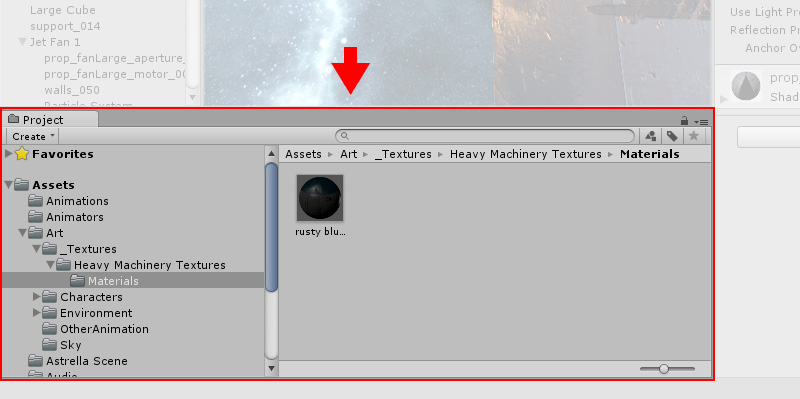
Prilikom izrade aplikacije koristila sam:

* 2017.4.3f1 verziju Unity Engine-a.
* Android SDK platforma 8.0.0
* Android SDK Build-Tools 27.0.3.
* Android SDK Tools 26.1.1.
* ARCore SDK za Unity v1.2.0.

## Sučelje Unity-a

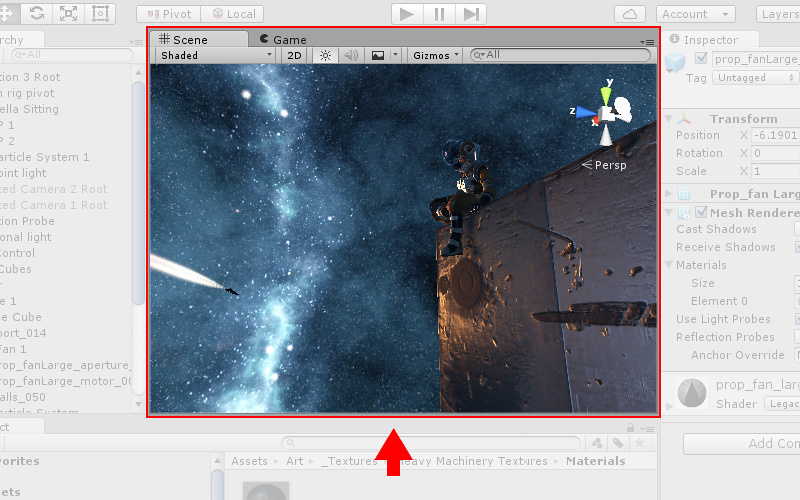
Sučelje Unity-a je vrlo jednostavno i lako za korištenje. Sastoji se od 5 prozora[12]:

* Projektni prozor (Slika 5‑4) prikazuje sve pakete dostupne za daljnju izradu aplikacije. Kad se unese novi paket, on se prikazuje u ovom prozoru.



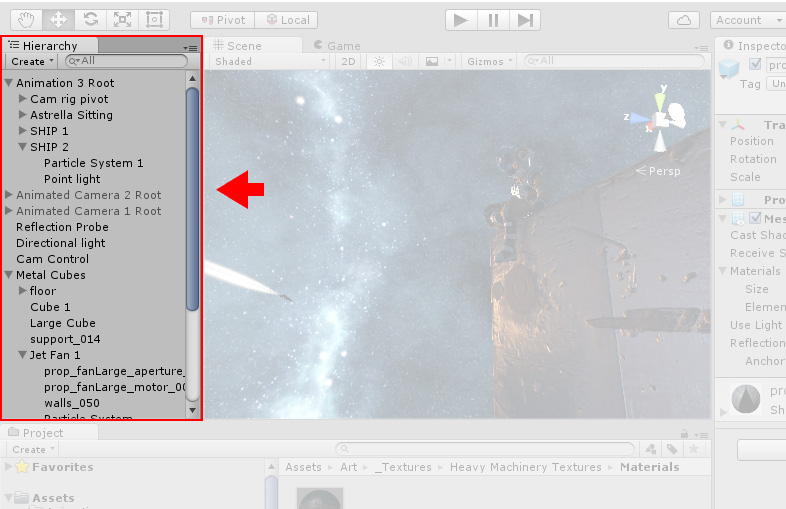
Slika 5‑4 Projektni prozor

* Scenski prozor (Slika 5‑5) dopušta vizualnu navigaciju i uređivanje scene. Scena se može prikazivati u 2D i 3D projektu, ovisno o tome koji je potreban.



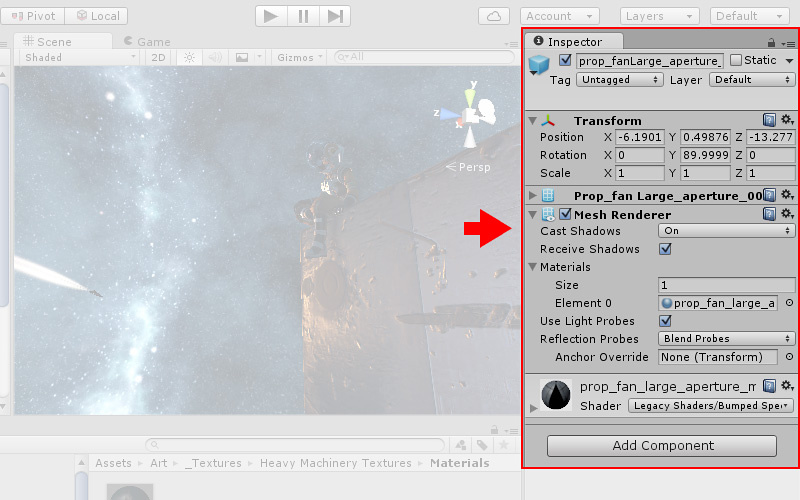
Slika 5‑5 Scenski prozor

* Hijerarhijski prozor (Slika 5‑6) je hijerarhijski tekstualni prikaz svih igraćih objekata na sceni. Svaka stavka na sceni ima unos u hijerarhiji, tako da su dva prozora inherentno povezana. Hijerarhija otkriva strukturu međusobno povezanih objekata.



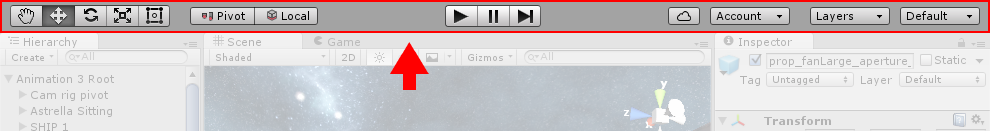
Slika 5‑6 Hijerarhijski prozor

* Inspektorski prozor (Slika 5‑7) omogućava pregled i uređivanje postavki trenutno odabranog igraćeg objekta. S obzirom na to da različiti igraći objekti imaju različite postavke, izgled inspektorskog prozora varira.



Slika 5‑7 Inspektorski prozor

* Alatna traka (Slika 5‑8) sastoji se od 7 osnovnih kontrola koje omogućuju pristup najvažnijim radnim značajkama. Objašnjavajući slijeva na desno to su alati za preoblikovanje igraćih objekata, alati za preoblikovanje gizmo pregleda, alati za upravljanje pokretanja igraće scene, cloud opcija koja omogućava spremanje izrađenih sadržaja, opcija za prijavljivanje na online korisnički račun, Layers kontrola koja određuje koji su objekti prikazani na sceni i Layout opcija koja kontrolira raspored svih prikaza.

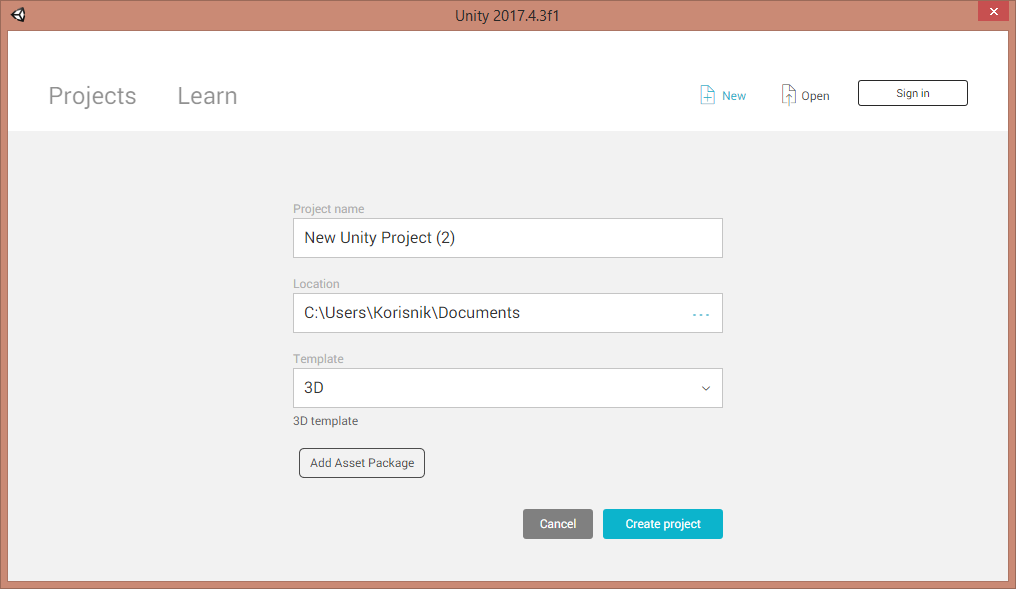


Slika 5‑8 Alatna traka

## Integracija Android SDK za Unity

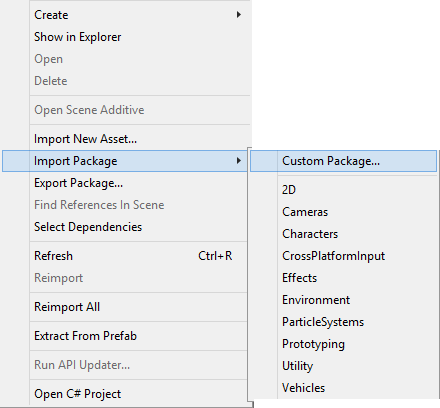
Nakon što je instaliran Unity i preuzeti s interneta svi potrebni SDK za izradu aplikacije proširene stvarnosti, potrebno je integrirati Android SDK u Unity.

Pokretanjem Unity Engine-a otvara se novi prozor za kreiranje novog ili otvaranje već prethodno napravljenog projekta. Prilikom izrade novog projekta pogodnog za MeasureAR aplikaciju bitno je odabrati 3D izradu aplikacije kao što je prikazano na slici (Slika 5‑9)

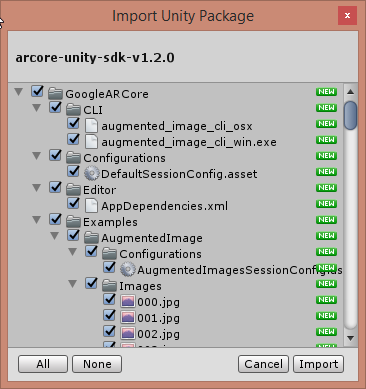


Slika 5‑9 Kreiranje novog Unity projekta

Kreiranjem novog projekta otvara se sučelje Unity-a objašnjeno u prethodnom potpoglavlju. Android SDK paket se unosi u Unity odabirom opcije prikazanom na slici . Moguće je izabrati unos samo nekih komponenata kao što je prikazano na slici (Slika 5‑10,Slika 5‑11)



Slika 5‑10 Unošenje paketa



Slika 5‑11 Unošenje ARCore SDK paketa

## Koordinatni sustav Unity-a

Unity podržava 4 koordinatna sustava: Screen, World, Viewport i GUI.

Screen koordinate predstavljaju 2D prikaz, mjere u pikselima i započinju u donjem lijevom kutu u točki (0,0) i idu do širine i visine ekrana. Screen koordinate se mijenjaju s rezolucijom i orijentacijom uređaja.

**[[12]](#footnote-12)**GUI koordinate se koriste GUI sustavom. Identične su Screen koordinatama u svim komponentama osim u početnoj točki koja počinje u gornjem lijevom kutu u (0,0) i širi se do širine i visine ekrana.

Viewport koordinate se ne mijenjaju bez obzira na rezoluciju. 2d formata su i počinju u toči (0,0) u donjem lijevom kutu i idu do točke (1,1) u gornjem desnom kutu.

World koordinate su 3D koordinate koje koordiniraju sustav i sve objekte u njemu.

# MEASUREAR APLIKACIJA

MeasureAR aplikacija je aplikacija za mjerenje dimenzije prostora. Ona detektira ravne površine pogodne za mjerenje te interakcijom korisnika izvodi daljnje funkcije. Princip rada aplikacije zasniva se na hit testing-u koji uzima zadnje 2 koordinate na kojima je korisnik komunicirao s ekranom i izračunava udaljenost između njih.

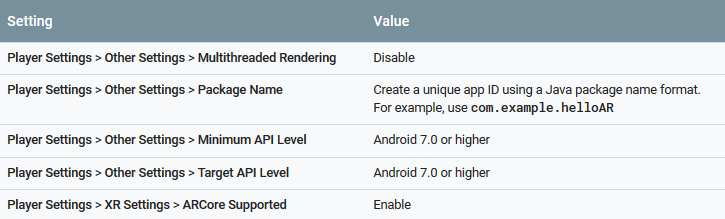
Komponente potrebne za razvoj MeasureAR aplikacije mogu se razdvojiti u dvije kategorije: one hardverske i one softverske. Hardverska kategorija obuhvaća mobitel koji podržava ARCore platformu i sadrži najnoviju verziju ARCore-a. Softverska kategorija obuhvaća Android studio i njegove SDK kao i Unity Engine.

## HelloAR

HelloAR je osnovna ARCore podržana aplikacija koja detektira ravne površine pogodne za usidrivanje Andy androida. Ova aplikacija dolazi u osnovnom SDK paketu Androida za Unity.

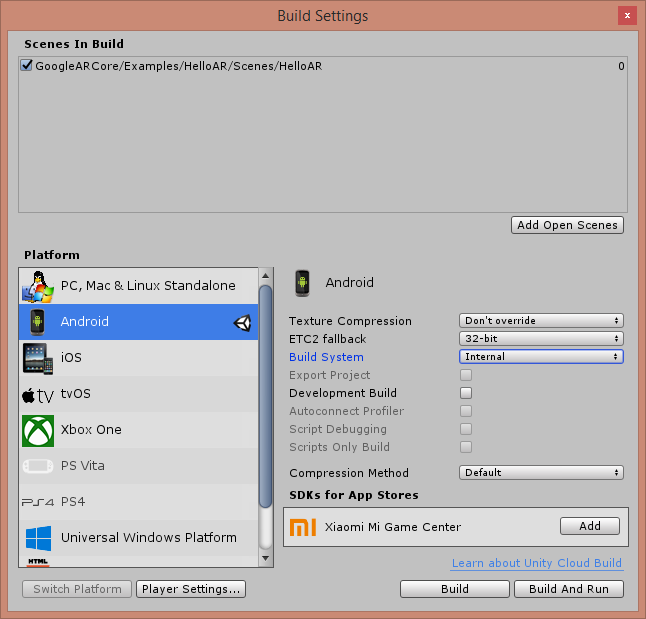
Prije pokretanja same HelloAR aplikacije, potrebno je promijeniti određene postavke u Inspektor prozoru a to su:

* Onemogućavanje višestrukog prikazivanja
* Promjena imena projekta
* Određivanje minimalnog API levela koji u slučaju korištenja ARCore biblioteke mora biti minimalno Android 7.0.0 (Nougat)
* Omogućavanje ARCore podržanih datoteka

Pristup promijenjenim komponentama prikazan je na slici (Slika 6‑1):

Slika 6‑1 Postavljanje komponenti

Kako bi se omogućilo pokretanje aplikacije potrebno je omogućiti programerske opcije na mobilnom uređaju, kao i USB debuggiranje čije opcije se nalaze u postavkama mobilnog uređaja. Uređaj se spaja s računalnim uređajem putem USB kanala. Kako bi mobilni uređaj i Unity bili kompatibilni, u Build postavkama potrebno je postaviti Android kao korišteni uređaj. Dodaje se HelloAR scena i postavlja sustav za pokretanje na unutarnji pogled. Program se pokreće pritiskom na tipku Build and Run kao što je prikazano na slici (Slika 6‑2)



Slika 6‑2 Pokretanje projekta

Unity stvara projekt na Android APK[[13]](#footnote-13), instalira ga na mobitel i pokreće. Kad se program pokrene, pali se kamera na Android uređaju i mobitel, detektirajući ravne površine, počinje prikazivati značajne točke. Ako postoji više susjednih točkica koje su nastale unutar iste ravnine, one se povezuju u novu ravninu na kojoj je moguće raditi (Slika 6‑3). Svaki put kad se nova ravnina detektira, prikazat će se u novoj boji.



Slika 6‑3 Detektiranje ravne površine

Nakon što je detektirana ravna površina moguće je usidravati Andy Androide metodom hit testinga koja je objašnjena u prošlom poglavlju. Korisnik dotiče ekran i na koordinati na kojoj je komunicirao s ekranom se pojavljuje Andy Android (Slika 6‑4) što će biti iskorišteno kao osnova pri izradi MeasureAR aplikacije.



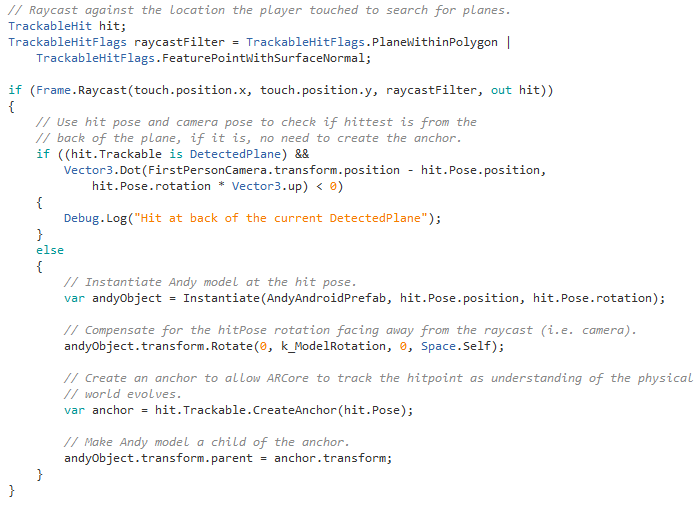
Slika 6‑4 Usidravanje Andy Androida

## HelloAR Controller skripta

HelloAR aplikacija sadrži HelloAR Controller skriptu koja opisuje postupak usmjeravanja zrake u slučaju interakcije korisnika s ekranom kao i usidrivanje Andy Androida na dotaknutu koordinatu.

Funkcija Raycast (Slika 6‑5) klase Frame uzima za parametre (x,y) koordinate dotaknute točke i varijable imena raycastFilter i hit. RaycastFilter varijabla je filter koji se odnosi na bit maskerade gdje svaki set bitova u TrackableHitFlags[[14]](#footnote-14) predstavlja kategoriju usmjerenih hitaca koju metoda poziva treba razmotriti kao valjanu. Hit je lista TrackableHit[[15]](#footnote-15) strukture koja će se postaviti ukoliko je usmjeravanje zrake bilo uspješno. Funkcija vraća *true* ukoliko je usmjeravanje trake uspjelo, inače vraća *false.*

Unutar Raycast funkcije se postavljaju uvjeti da ravna površina bude detektirana i da poziv funkcije Vector3.Dot bude manji od nule. Funkcija Dot strukture Vector3[[16]](#footnote-16) je funkcija koja vraća float vrijednost jednaku umnošku pomnoženih amplituda 2 vektora i kosinusa kuta između njih. Ukoliko je vraćena vrijednost manja od nula, kao što je uvjet u ovom slučaju, vektori su okrenuti u suprotnim smjerovima. U tom slučaju šalje se poruka Unity konzoli da je dotaknuta točka iza trenutno detektirane površine. U suprotnom, inicijalizira se varijabla andyobject koja postavlja Andija na dotaknutu točku, postavlja se njegova orijentacija i usidruje ga se na dotičnu koordinatu.



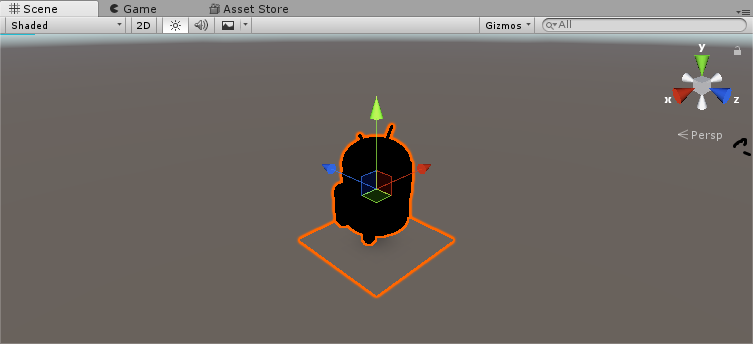
Slika 6‑5 Raycast funkcija

## MeasureAR razvoj aplikacije

Za izradu MeasureAR aplikacije koja mjeri udaljenosti između dvije točke u prostoru, koristi se prethodno objašnjena HelloAR aplikacija. Pomoću te aplikacije moguće je dodirom na ekran uređaja usidriti Andy Androida na koordinate definiranih točaka.

### Sfera

Otvaranjem igraćeg objekta Example Controller u inspektorskom prozoru vidimo da on sadrži varijablu Andy Android Prefab s vrijednosti Andy. Prenošenjem Andija na scenu i klikom na tipku ''f''' ishodište scene vezuje se za položaj Andija (Slika 6‑6).

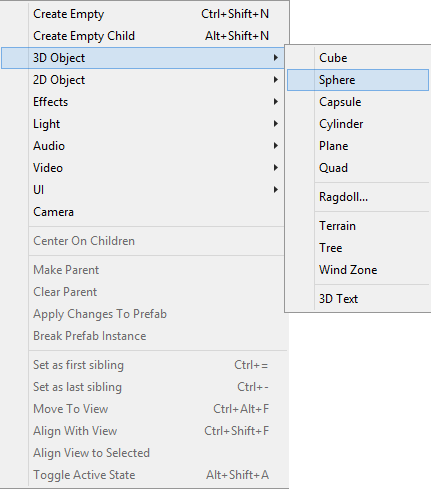


Slika 6‑6 Fokusiranje scene na Andy Androida

S obzirom na to da za potrebe naše aplikacije Andy nije najbolja opcija, potrebno je stvoriti novi 3D objekt veličine slične Andy-u. Novi objekt stvaramo odlaskom na (Slika 6‑7)

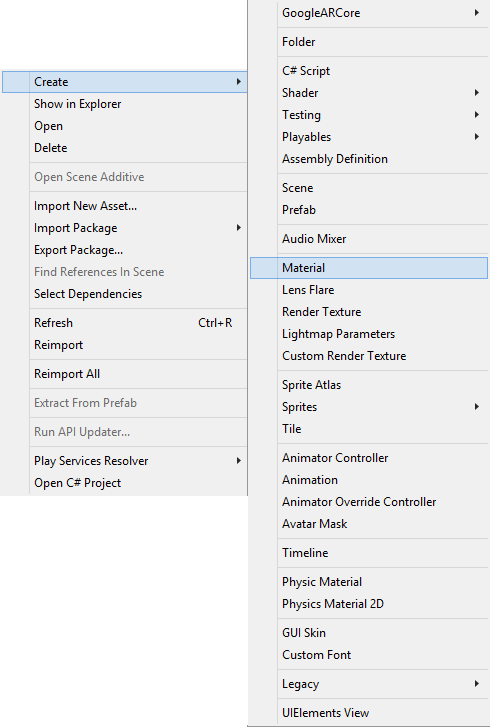
*GameObject > 3D object > Sphere*

Novostvorena sfera veličinom je puno veća od Andy-a i potrebno je smanjiti. Za smanjivanje novog 3D objekta koristi se alatna traka objašnjena u prethodnom poglavlju. Andy-a se briše sa scene.

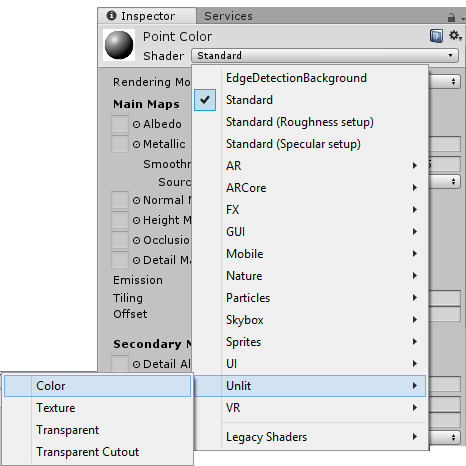


Slika 6‑7 Postupak izrade 3D objekta

Nakon što je izrađen novi 3D objekt, potrebno je izraditi materijal imena 'Point Color' za izgled objekta odlaskom na *Create > Material (*Slika 6‑8*)* i odrediti boju materijala *Shader > Unlit > Color* (Slika 6‑9).



Slika 6‑8 postupak izrade novog materijala



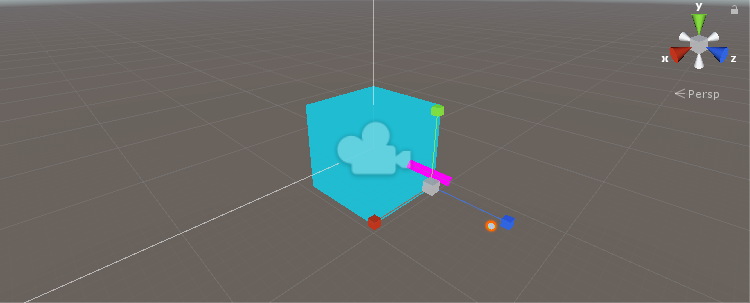
Slika 6‑9 Postupak odabira boje materijala

Nakon što je materijal stvoren, potrebno ga je prenijeti preko našeg 3D objekta kako bi poprimio novoizrađeno svojstvo.

### Linija

Nakon što je napravljena sfera, potrebno je izraditi liniju koja će povezivati dvije sfere. Za izradu spomenutog koristi se već definirana Unity funkcija 'Line Renderer'. Klikom na sferu u Inspector prozoru potrebno je dodati komponentu 'Line Renderer'.

U početku linija neće biti vidljiva zato što počinje na točki koordinate (0,0,0) i završava na točki (0,0,1). Kako bi linija postala vidljiva potrebno je otvoriti Android Device igraći objekt na sceni. Otvara se novi prikaz (Slika 6‑10) u kojoj je ljubičasti sadržaj naša linija. S obzirom na to da je linija predebela u Line Renderer-u u polje 'Width' ( hrv. Širina ) upisuje se proizvoljna mala vrijednost, u našem slučaju to je 0.03.



Slika 6‑10 Linija

Ako se želi, moguće je promijeniti boju linije stvaranjem novog materijala na isti način na koji se stvarala boja za sferu. Nakon što se izradi materijal, potrebno je prenijeti ga u polje *Materials::Element 0* u inspektorskom prozoru sfernog igraćeg objekta.

U inspektorskom prozoru se postavlja pozicija na 0 i prenosi sfera na Andy-a kako bi ubuduće svakim klikom na površinu bila prikazana sfera a ne Andy. Izbriše se sfera iz hijerarhijskog prozora.

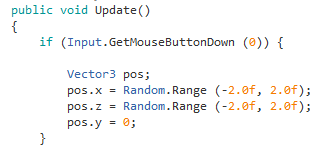
### Example Controller skripta

Nakon napravljene sfere i linije, potrebno je napraviti funkcije koje će one obavljati. Iako naša aplikacija mora biti ARCore podržana, prvo izrađujemo aplikaciju koju ćemo moći pokretati i provjeravati na Unity sceni, a naknadno je pretvaramo u ARCore podržanu aplikaciju. Za početak potrebno je napraviti listu (Slika 6‑11) koja će sadržavati sve objekte:

public.png

Slika 6‑11 Stvaranje liste Points

U 'Update' funkciji (Slika 6‑12) kreira se pozicija i rotacija:



Slika 6‑12 Update funkcija

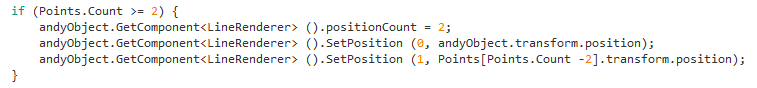
Instancira se igraći objekt na površinu pomoću gotove funkcije Instantiate (Slika 6‑13) koja za varijable prima Andy Android-a, poziciju i rotaciju Andy-a.

Dodaje se Andy na listu:

Screenshot (498).png

Slika 6‑13 Funkcija Instantiate

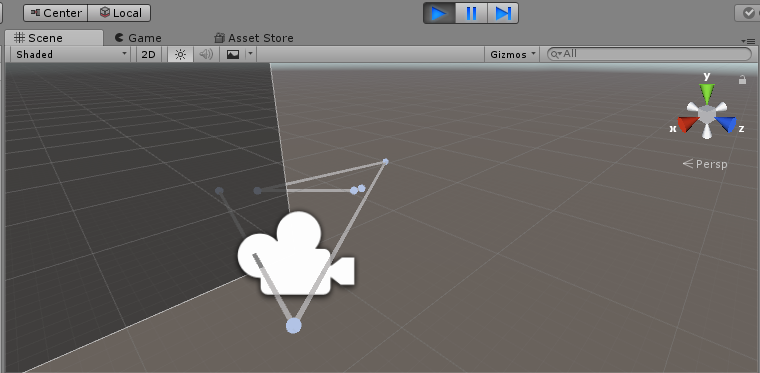
Kako bi se stvorila linija potrebno je imati minimalno 2 točke pa se stvara uvjet (Slika 6‑14) koji, ako je istinit, počinje crtati liniju:



Slika 6‑14 Uvjet za stvaranje linije

U funkciji je potrebno koristiti funkciju set.position koja postavlja pozicije točaka. Prva set.position funkcija za vrijednosti uzima 0 što je prva pozicija i trenutnu vrijednost Andy objekta. Druga set.position funkcija uzima vrijednost 1 što je druga pozicija i zadnji Andy objekt koji je dodan.

Pokretanjem aplikacije na Game sceni i klikanjem miša po njoj, na sceni se počinju pojavljivati linije i točke (Slika 6‑15).



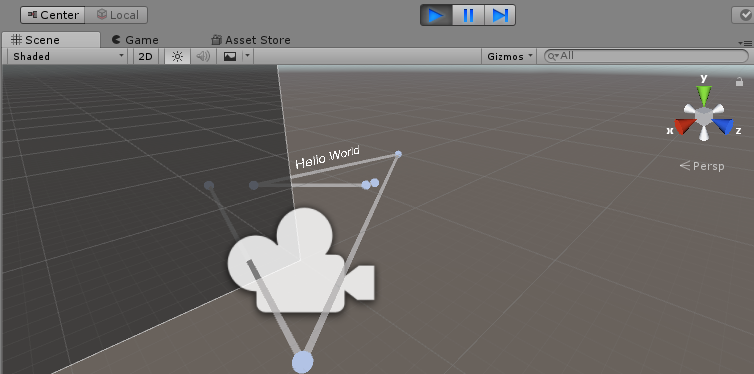
Slika 6‑15 Povezivanje točki linijom

### Tekst

Zbog lakšeg određivanja proporcionalnosti teksta i linija, iako još uvijek Play mode-u, kreiramo novi 3D objekt koji je u ovom slučaju 3D tekst (Slika 6‑16).

*GameObject > 3D Object > 3D Text*

S obzirom na to da je tekst mutan, u inspektor prozoru potrebno je povećati komponentu Font Size za 100 puta veću vrijednost. Nadalje je potrebno smanjiti skalu, tj veličinu teksta na sceni tako da bude pogodan za čitanje, a opet ne prevelik. Postavlja se sidro na centar kako bi novostvoreni tekst bio centriran. Tekstualni objekt se prenosi iz hijerarhijskog prozora u Prefab folder pod imenom 'Text'.



Slika 6‑16 Kreiranje 3D teksta

Nakon što je napravljen 3D tekst potrebno je opet otvoriti ARController skriptu i izraditi funkciju kojom će taj tekst ispisivati udaljenost između dvije točke.

Za početak se stvara varijablu za tekst (Slika 6‑17):

Screenshot (502).png

Slika 6‑17 Varijabla Text

Svaki put kad se povuče linija mora se prikačiti tekst na sredinu te linije. Nadopunjuje se Update funkcija prethodno korištenom funkcijom Instantiate:

*var temp = Instantiate ( Text, (andyObject.transform.position + Points [Points.Count – 2].transform.position)/2, Quaterion.identity);*

Ovom funkcijom omogućavamo povezivanje teksta i linije između dvije točke, međutim potrebno je postaviti tekst da se prikazuje pod čitljivim kutom (Slika 6‑18).

Screenshot (503) - Copy.png

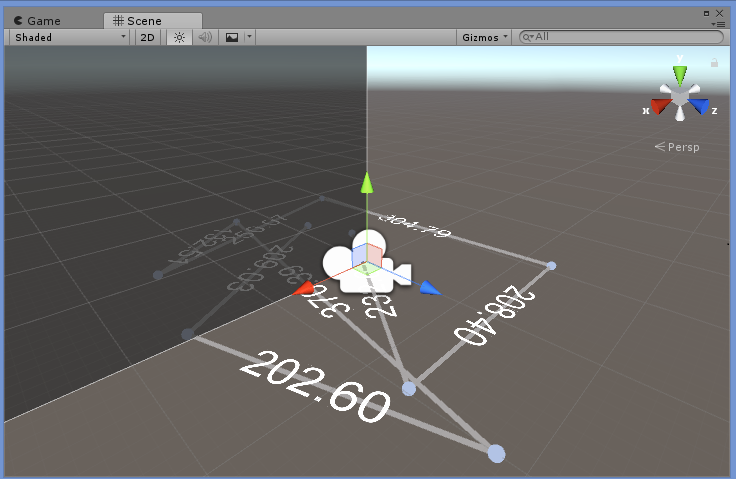
Slika 6‑18 Pozicioniranje teksta

Prva funkcija će učiniti da tekst gleda na zadnju točku koju smo napravili, a druga funkcija da gleda tekst put gore pod kutem od 90 stupnjeva i da bude iste orijentacije kao linija.

Za mjerenje udaljenosti koristi se gotova funkcija *Distance*. Rezultat je potrebno pomnožiti sa 100 kako bi prikaz bio u centimetrima Za čitkiji prikaz, rezultat se postavlja u string varijablu s 2 decimalne znamenke. Nakon što se novi kod kompajlira, tekstualna komponentu prenosi se u ARController igraći objekt.

*temp.GetComponent<TextMesh>().text = (Vector3.Distance(andyObject.transform.position, Points[Points.Count – 2].transform.position)\*100).ToString(''0.00'');*

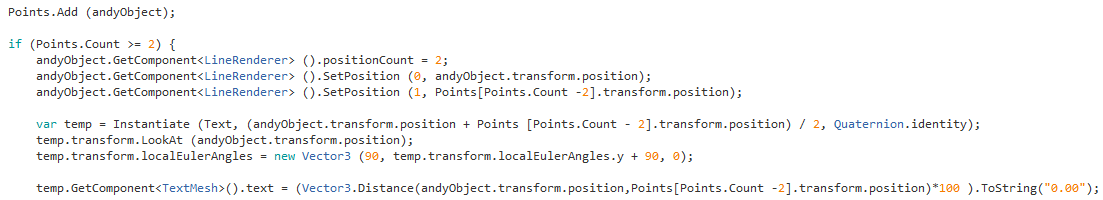
Kad se aplikacija pokrene i korisnik klika po ekranu igraće scene, ispisuju se linije i udaljenosti između krajnjih točki. S ovim korakom smo napravili aplikaciju mjerenja udaljenosti između točaka (Slika 6‑19).



Slika 6‑19 MeasureAR aplikacija u Unity sceni

### ARCore okruženje

Kako bi aplikacija radila u ARCore okruženju potrebno je iznova urediti ARController skriptu. Kopira se dio prethodno napisanog koda i prenosi u dio koda (Slika 6‑20) koji uvjetuje bacanje zrake prilikom dodira ekrana. Izbriše se prethodno napisani kod kako se ne bi miješao s ARCore dijelom koda.



Slika 6‑20 MeasureAR kod u ARCore okruženju

Pokretanjem aplikacije, mobilna kamera se pali i detektira ravne površine. Korisnik dodirom na ekran bira početku i krajnju točku između kojih aplikacija izračuna i ispiše udaljenost (Slika 6‑21).



Slika 6‑21 MeasureAR aplikacija u ARCore okruženju

# ZAKLJUČAK

U posljednjih nekoliko godina razvoj proširene stvarnosti raste u velikim razmjerima čineći integraciju virtualnih objekata u stvarni svijet gotovo neprimjetnom. S obzirom na veliku popularnost, potražnja za alatima potrebnim za izradu aplikacija je sve veća. Korišteno Unity razvojno okruženje jedno je od službenih okruženja korištenih za izradu aplikacija proširene stvarnosti s pomoću ARCore platforme. Prilikom izrade aplikacija potreban je prikazni uređaj što je u ovom slučaju bio ARCore podržan mobitel za koji je bilo potrebno preuzeti aplikaciju ARCore-a. Isto tako instalacijom Android SDK paketa za Unity omogućeno je povezivanje mobitela s aplikacijom. U praktičnom dijelu kreirana je aplikacija za izračunavanje udaljenosti između 2 točke. Korisnik pokreće aplikaciju koja detektira ravne površine. Doticanjem ekrana korisnik pozicionira sidra na koordinate dotaknutih točaka a aplikacija gotovom funkcijom izračunava udaljenost između njih i ispisuje vrijednost na liniju koja povezuje točke. Način realizacije aplikacije omogućava kontinuirani ispis udaljenosti za zadnje dvije dotaknute točke.

Za izradu zadatka ovog završnog rada bilo je potrebno upoznati se s funkcionalnošću i logikom korištenja alata za proširenu stvarnost. Za implementaciju, a u konačnici i za realizaciju zadatka, odabranaGoogleova ARCore platforma. Osim poznavanja funkcija koje nudi ARCore SDK, potrebno je steći i znanja o izradi aplikacije u Unity Engine razvojnom okruženju. Kreiranjem temeljne aplikacije za mjerenje udaljenosti između dvije točke ravnine, osigurani su temelji za nadogradnju aplikacije u kojoj se preciznost može povećavati, spremati rezultate i izračunavati dodatne vrijednosti kao što su površine i volumen. Zadaća završnog rada bila je upoznavanje s radom ARCore-a i izrada aplikacije za mjerenje dimenzija prostora i ona je uspješno ostvarena. Daljnji razvoj aplikacije u navedenim komponentama bit će ostvaren u budućnosti i kao finalni projekt nalazit će se na Github linku[14].

# LITERATURA

#### [1] https://en.wikipedia.org/wiki/Augmented\_reality

#### [2] http://www.realitytechnologies.com/augmented-reality

#### [3] https://www.wikitude.com/

#### [4] https://developer.apple.com/documentation/arkit

#### [5] https://thinkmobiles.com/blog/best-ar-sdk-review/

#### [6] https://en.wikipedia.org/wiki/ARToolKit

#### [7] <https://en.wikipedia.org/wiki/Tango_(platform)>

#### [8] https://developers.google.com/ar/discover/

#### [9] https://developers.google.com/ar/discover/supported-devices

#### [10] https://en.wikipedia.org/wiki/Unity\_(game\_engine)

#### [11] https://github.com/google-ar/arcore-unity-sdk/releases

#### [12]<https://docs.unity3d.com/Manual/LearningtheInterface.html?_ga=2.81325463.1043136766.1531390115-639174347.1526317830>

#### [13] <https://answers.unity.com/questions/781870/understanding-unitys-coordinate-system.html>

#### [14] https://github.com/asundo00

# SAŽETAK

U završnom radu predstavljena je aplikacija proširene stvarnosti koja osigurava prikaz linije mjerenja između dvije točke ravnine i prikaz udaljenosti između te dvije točke u centimetrima. Za realizaciju projekta korištena je ARCore platforma proširene stvarnosti i okruženje Unity za integraciju raspoloživih Alata u odgovarajući prostorni model. Opisana su osnovna obilježja sustava za proširenu stvarnost, dan je kratki prikaz najpopularnijih koje se danas koriste, a zbog jasnije realizacije praktičnog dijela zadatka, detaljnije su opisana obilježja grafičkog enginea Unity, načina instalacije svih potrebnih komponenti i funkcija iz SDK ARCore platforme kojima je zadatak realiziran. Aplikacija je učitana i pokrenuta korištenjem Samsung S8 mobilnog uređaja. Pokazalo se da se mjerenja ravninskih udaljenosti izvode s točnošću od +/- 8%, što aplikaciju čini korisnom u premjeravanju duljina unutrašnjih prostora zbog, primjerice, planiranja rasporeda namještaja. Buduće aktivnosti bit će usmjerene na proširianje funkcionalnosti aplikacije i ugradnji dodatnih prostornih mjerenja.

# SUMMARY

In this thesis, presented augmented reality application provides a display of the measurement line between the two points on the plane and the representation of the distance between those two points in centimeters. For the realization of the project, the ARCore platform has been used and the Unity environment to integrate the available tools into an appropriate 3D model. The basic features of the augmented reality system are described, a brief overview of the most popular platforms are given, and for a clearer understanding of the practical part of the task, the features of the Unity graphic engine are described in more detail, the process of the installation of all the required components and the description of the required functions from the SDK ARCore platform, . The app has been downloaded and launched using the Samsung S8 mobile device. It has been shown that the distance measurements are performed with accuracy of +/- 8% , which makes the application useful in measuring the length of the interior space due to, for example, arranging the furniture layout. Future activities will be aimed at expanding application functionality and installing additional spatial measurements.

# POPIS KRATICA I REFERENCA

Vizualna odometrija – proces određivanja pozicije i orijentacije robota analizirajući sliku kamere

AR – Augmented Reality

HDM – Head Mouted Display

SLAM – Simultaneous Localization and Mapping

VIO – Visual Inner Odometry

Open Source – Otvoren za javnost i slobodno korištenje

VuMark – Kombinacija slike i QR-koda

API – Application Programming Interface

Hit Testing – označavanje zanimljivog mjesta na ekranu dodirom

COM – Concurrent Odometry and Mapping

SDK – Software Developement Kit

GUI – Graphical user Interface

APK – Android PacKage

TrackableHitFlag – Zastave koje predstavljaju kategorije usmjerenih zraka

TrackableHit – Struktura koja sadrži informacije o sudaru usmjerene zrake i objekta kojeg prati ARCore

Vector3 – Struktura koja predstavlja 3D vektore i točke

1. Proces određivanja pozicije i orijentacije robota analizirajući sliku kamere [↑](#footnote-ref-1)
2. Eng. Augmented Reality - Proširena stvarnost [↑](#footnote-ref-2)
3. Head – mouted display [↑](#footnote-ref-3)
4. Simultaneous Localization and Mapping [↑](#footnote-ref-4)
5. Visual Inner Odometry [↑](#footnote-ref-5)
6. Otvoren za javnost i slobodno korištenje [↑](#footnote-ref-6)
7. Kombinacija slike i QR-koda [↑](#footnote-ref-7)
8. Application Programming Interface [↑](#footnote-ref-8)
9. Označavanje zanimljivog mjesta na ekranu dodirom [↑](#footnote-ref-9)
10. Concurrent odometry and mapping [↑](#footnote-ref-10)
11. Software development kit [↑](#footnote-ref-11)
12. Graphical user interface [↑](#footnote-ref-12)
13. Android PacKage [↑](#footnote-ref-13)
14. Zastave koje predstavljaju kategorije usmjerenih zraka [↑](#footnote-ref-14)
15. Struktura koja sadrži informacije o sudaru usmjerene zrake i objekta kojeg prati ARCore [↑](#footnote-ref-15)
16. Struktura koja predstavlja 3D vektore i točke [↑](#footnote-ref-16)