

Primjena proširene stvarnosti u nastavi iz prirodnoslovnih predmeta u osnovnim i srednjim školama

Mađerić, Neven

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:150831>

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

Download date / Datum preuzimanja: **2025-04-02**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

Neven Mađerić

PRIMJENA PROŠIRENE STVARNOSTI U
NASTAVI IZ PRIRODOSLOVNIH PREDMETA U
OSNOVNIM I SREDNJIM ŠKOLAMA

Diplomski rad

Zagreb, 2022.

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

INTEGRIRANI PREDDIPLOMSKI I DIPLOMSKI SVEUČILIŠNI STUDIJ
FIZIKA I INFORMATIKA; SMJER NASTAVNIČKI

Neven Mađerić

Diplomski rad

**PRIMJENA PROŠIRENE STVARNOSTI
U NASTAVI IZ PRIRODOSLOVNIH
PREDMETA U OSNOVNIM I
SREDNJIM ŠKOLAMA**

Doc. dr. sc. Andrej Novak

Ocjena diplomskog rada: _____

Povjerenstvo: 1. _____

2. _____

3. _____

Datum polaganja: _____

Zagreb, 2022.

Sažetak

Školski su udžbenici iz prirodoslovnih predmeta često bogati primjerima koji učenicima olakšavaju predodžbu, a onda i usvajanje određenog pojma ili koncepta. Međutim, u takvim udžbenicima (dvodimenzionalnog prikaza), često se prikazuju primjeri koji opisuju trodimenzionalne pojave, procese i modele. Pri tome se javljaju poteškoće u predodžbi odnosno vizualizaciji istih na koje učenici nerijetko nailaze. Te bi se poteškoće mogle smanjiti, a možda čak i izbjeći prikazivanjem tih primjera u trodimenzionalnom prostoru. Cilj je ovog diplomskog rada kroz odabrane primjere iz prirodoslovlja pokazati kako se neki modeli ili procesi mogu prikazati na brz i jednostavan način koristeći proširenu stvarnost. Proširena se stvarnost može implementirati pomoću paketa Vuforia koji se nalazi unutar softvera za razvoj i upravljanje videoigara Unity. Ovaj je programski paket baziran na sintaksi objektno orijentiranog jezika C#, a gotova se aplikacija može pokrenuti na pametnim telefonima. Kada bi kamera pametnog telefona detektirala stacionarnu sliku nekog modela, aplikacija bi prikazala trodimenzionalni prikaz tog istog modela. Cilj praktičnog dijela ovog diplomskog rada je prikazati razne primjere primjene proširene stvarnosti koji su bazirani na temama iz fizike, kemije, matematike i biologije.

Ključne riječi: proširena stvarnost, objektno orijentirano programiranje, Unity, Vuforia, mobilna aplikacija

Application of augmented reality in teaching of science subjects in primary and secondary schools

Abstract

Science textbooks for schools have many examples that can help students understand and adopt certain terms or concepts. However, textbooks with two-dimensional representations, often show examples of three-dimensional phenomena, processes, and models. There can be some difficulties in imagining or visualizing them, which students often encounter. Those difficulties could be reduced, and perhaps even avoided, by showing these examples in three-dimensional space. The objective of this paper is to show how some models or processes can be presented in a quick and simple way using augmented reality. Augmented reality can be implemented using the Vuforia package found within Unity - a software for video game development and management. This software package is based on the syntax of the object-oriented language C#, and the finished application can be run on smartphones. When a smartphone camera would detect a stationary image of a model, the application would show a three-dimensional view of that same model. The goal of the practical part of this paper is to present various examples of application of augmented reality which are based on subjects from physics, chemistry, mathematics, and biology.

Keywords: augmented reality, object oriented programming, Unity, Vuforia, mobile application

Sadržaj

1	Uvod	1
2	Proširena stvarnost	2
3	Tehnologije za izradu mobilne aplikacije	5
3.1	Unity	5
3.2	Microsoft Visual Studio	7
3.3	Vuforia	8
3.3.1	Kreiranje ciljanih slika (eng. <i>Image Targets</i>)	8
3.4	Blender	9
4	Izrada aplikacije	11
4.1	Stacionarni modeli	11
4.2	Nestacionarni modeli	16
4.3	Proširenje modela	19
4.4	Korisničko sučelje (eng. <i>User Interface, UI</i>)	21
4.4.1	Izrada korisničkog sučelja	21
4.5	Priprema aplikacije za izvoz na pametni telefon	23
5	Primjena aplikacije u prirodoslovnim znanostima	25
5.1	Biologija	25
5.2	Kemija	26
5.3	Matematika	27
5.4	Fizika	28
6	Zaključak	30
	Dodaci	32
A	Funkcionalnost aplikacije	32
B	Popis slika	33
	Literatura	35

1 Uvod

Tehnološki napredak znatno je utjecao na izgled današnjih učionica, ali i na izbor i primjenu nastavnih strategija i metoda. Interaktivni pametni uređaji sve su prisutniji, dok istraživački usmjerena nastava sve više zamjenjuje tradicionalne oblike poučavanja. Svrha istraživački usmjerene nastave jest omogućiti učeniku da vlastitom aktivnošću i istraživanjem dođe do spoznaje. U tome mu svakako pomaže opremljenost učionice modernim tehnološkim sredstvima i pomagalicama. Isto tako, školski su udžbenici često popraćeni elektroničkom građom u obliku CD-a ili DVD-a, a u posljednje se vrijeme u udžbenicima mogu pronaći i QR kodovi koji omogućavaju brže i lakše usvajanje informacija, digitalni pristup tiskanom sadržaju te interaktivnost koja učenika potiče na daljnje istraživanje i učenje [1].

Osim toga, u nastavi se sve više koriste mobilne i web aplikacije. Takve se aplikacije često razvijaju pomoću tehnologije proširene stvarnosti čija je primjena tako našla svoje mjesto i u području obrazovanja, odnosno učenja. Jedna od takvih aplikacija je *Augmented Chemical Reactions* [2]. Riječ je o aplikaciji koja prikazuje dinamiku deformacije molekula kada se one međusobno približe. Učenici tako dobivaju bolji uvid u ponašanje molekula u njihovom međusobnom odnosu. S obzirom na intuitivno upravljanje 3D modelima molekula koje omogućava, ova aplikacija itekako doprinosi lakšem razumijevanju i učenju nastavnog gradiva kemije. Isto tako, proširena stvarnost omogućila je razvoj aplikacije za prikazivanje i uspoređivanje Bohrovog modela atoma i kvantno-mehaničkog modela atoma. Aplikacija nastoji potaknuti učenika na kritičko i apstraktno razmišljanje [3]. Proširena stvarnost također omogućava simulacije mehanike fluida, od onih jednostavnijih koje prikazuju hidrostatski tlak, do onih složenijih koje prikazuju gibanja plinova i tekućina [4]. Simulacije koje se temeljene na proširenoj stvarnosti doprinose većoj aktivnosti učenika u nastavi fizike za razliku od standardnih simulacija koje od učenika traže slabu ili gotovo nikakvu interakciju [5]. Nadalje, još učinkovitijim se pokazalo korištenje proširene stvarnosti zajedno s grupnim radom jer bi učenici nakon analiziranja određenog 3D modela ili procesa mogli o njemu zajedno promišljati i međusobno raspravljati [6].

Imajući na umu sve navedeno, cilj ovog diplomskog rada je izraditi aplikaciju koja će učenicima obogatiti iskustvo učenja, omogućiti im interakciju s pojedinim modelima ili procesima te tako doprinijeti bržem i lakšem usvajanju nastavnog gradiva.

2 Proširena stvarnost

Proširena stvarnost (eng. *Augmented Reality*, AR) je tehnologija koja objektima fizikalnog svijeta pridružuje one virtualne koji se onda korisniku čine kao dio tog stvarnog svijeta. Osnovne značajke proširene stvarnosti su dakle spajanje stvarnih i virtualnih objekata u stvarnom okruženju, međusobno usklađivanje istih te omogućavanje interakcije između stvarnog i virtualnog svijeta u stvarnom vremenu [7].

Nastanak termina proširene stvarnosti (eng. *Augmented Reality*) veže se uz 1990. godinu i pripisuje Boeingovom istraživaču Thomasu Caudellu. Zajedno sa kolegom Davidom Mizellom, Caudell je osmislio sustav digitalnog oslikavanja koji se postavljao na glavu, a čija je svrha bila olakšati proces proizvodnje zrakoplova. Sustav je funkcionirao na način da je preklapao shematski prikaz komponenata sa mjestima na koja su ih radnici trebali postaviti [8].

Međutim, početci upotrebe proširene stvarnosti sežu do 1961. godine kada je američki filmski snimatelj Morton Heilig patentirao uređaj poznat pod nazivom *Sensorama*. Riječ je o uređaju prikazanom na slici 2.1 koji je simulirao vožnju motociklom koristeći zaslon za prikaz slike grada, ventilatore za vjetar te određene odašiljače za mirise i buku tako da je korisniku pružao potpun doživljaj vožnje motociklom kroz grad [9].

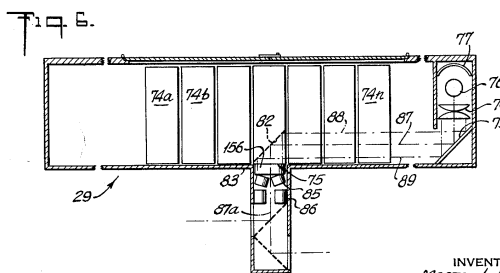
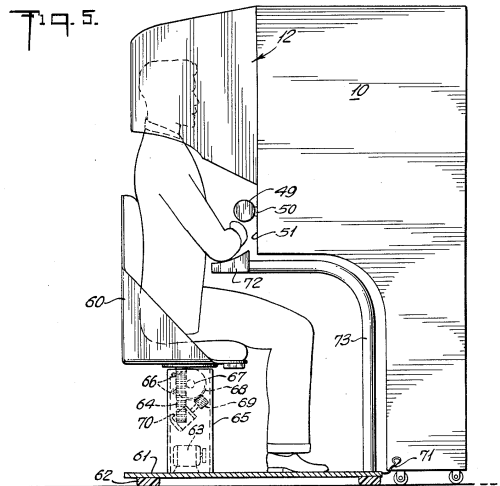
Aug. 28, 1962

M. L. HEILIG
SENSORAMA SIMULATOR

3,050,870

Filed Jan. 10, 1961

8 Sheets-Sheet 3



INVENTOR
MORTON L. HEILIG
BY
Douglas M. Clarkson
ATTORNEY

Slika 2.1: Prikaz *Sensorame*, uređaja koji je patentirao američki filmski snimatelj Morton Heilig [10].

Sedam godina kasnije skupina studenata s Harvarda pod vodstvom Ivana Sutherlanda osmislila je prve naočale sa sustavom proširene stvarnosti pod nazivom *Damoklov mač* (eng. *The Sword of Damocles*). Riječ je o sustavu koji je na stereoskopskom zaslonu prikazivao sliku čija se perspektiva mijenjala ovisno o pokretima glave i očiju korisnika [9]. Razvoj proširene stvarnosti u sljedećim godinama nije posustajao. *Videoplac* Myrona Kruegera (1975), *Virtual Fixtures* Louisa Rosenberga (1992) te *ARToolKit* (2000) samo su neki od izuma koji su obilježili razvoj proširene stvarnosti.

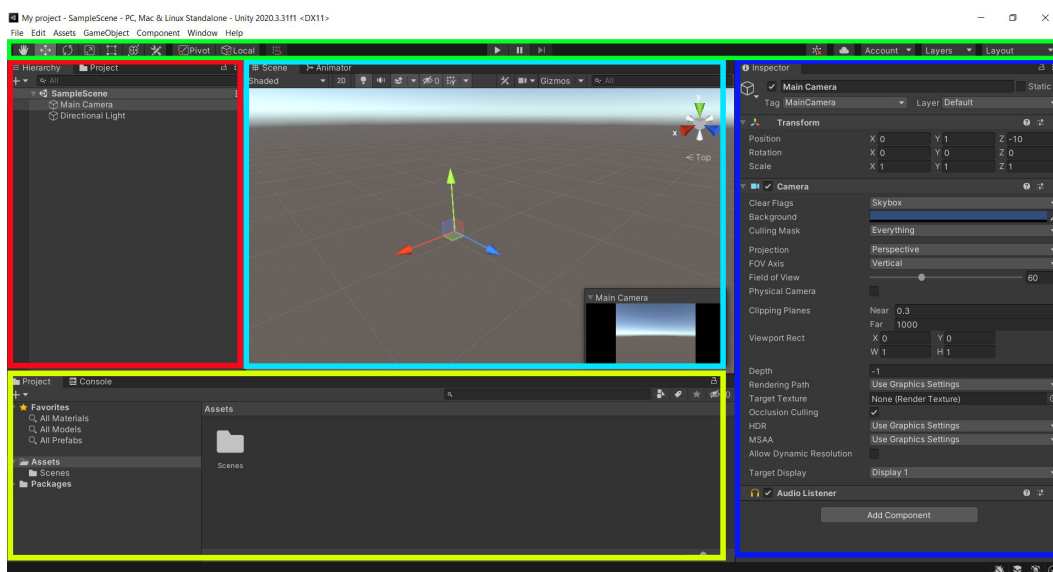
Tehnologija proširene stvarnosti tako je našla svoje mjesto u mnogim područjima kao što su zrakoplovstvo, inženjerstvo, marketing, medicina, a posljednjih godina sve više i u obrazovanju. Mnogi se znanstvenici slažu da uporaba ovakvih tehnologija kao što je proširena stvarnost doprinosi većoj motivaciji učenika, njihovoj međusobnoj interakciji te ugodnijoj atmosferi za učenje jer omogućava predočavanje apstraktnih i složenih sadržaja na zanimljiv i inovativan način, čineći pritom proces učenja smislenijim i učinkovitijim. U prilog tomu idu i određena istraživanja koja ukazuju na poboljšane rezultate učenja kod učenika u čijoj je nastavi bila zastupljena tehnologija proširene znanosti [11].

3 Tehnologije za izradu mobilne aplikacije

U ovome ćemo poglavlju navesti i opisati tehnologije koje smo koristili za izradu mobilne aplikacije.

3.1 Unity

Unity je 3D razvojna platforma koja omogućava programerima, grafičkim umjetnicima i dizajnerima stvaranje interaktivnog iskustva u stvarnom vremenu kroz izradu igara i aplikacija. Unity koristi C# koji je objektno orijentirani jezik. Ono po čemu se Unity razlikuje od ostalih platformi je Unity Asset Store koji korisnicima omogućava kupovanje, prodaju i besplatno dijeljenje 2D i 3D modela i animacija, raznih alata i obrazaca, dodatka i proširenja te video i audio sadržaja [12]. Unity se može instalirati na Windows, Mac i Linux operativnom sistemu te omogućava implementaciju sadržaja na velik broj platformi. To mogu biti operacijski sustavi iOS, Android, Windows i Linux, konzole PlayStation 4 i PlayStation 5, Xbox One, Xbox S, Nintendo Switch, pametni televizori operacijskih sustava Android i OS te platforme ARCore, Oculus Rift i Microsoft HoloLens. Unity je moguće besplatno preuzeti sa službene stranice [13]. Za korištenje dodatnih mogućnosti potrebno je odabrati i platiti određeni plan [14].



Slika 3.1: Prikaz Unity sučelja koje sadrži Alatnu traku (označenu zelenom bojom), Hijerarhijski prozor (označen crvenom bojom), Prikaz scene (označen svjetloplavom bojom), Prozor projekta (označen žutom bojom) te Pregledni prozor (označen tamnoplavom bojom).

Na slici 3.1 prikazano je početno sučelje Unityja na kojemu se nalaze: Prozor projekta (eng. *Project Window*), Hijerarhijski prozor (eng. *Hierarchy Window*), Prikaz scene (eng. *Scene View*), Pregledni prozor (eng. *Inspector Window*) te Alatna traka (eng. *Toolbar*).

Prozor projekta (eng. *Project Window*) odnosi se na donji lijevi dio sučelja koji hijerarhijski prikazuje popis datoteka, objekata i skripti trenutnog projekta. Odaberemo li određenu datoteku, objekt ili skriptu, na desnoj strani prozora prikazati će se njihov sadržaj. Korisnik u prozor projekta dodaje sve datoteke, objekte i skripte koje želi uključiti u svoj projekt [15].

Hijerarhijski prozor (eng. *Hierarchy Window*) odnosi se na gornji lijevi dio sučelja koji sadrži listu svih objekata (eng. *GameObject*) unutar trenutne scene (eng. *Scene*). Objekti mogu biti 2D i 3D modeli, kamera (eng. *Camera*), svjetlo (eng. *Light*), dodatne scene, itd. Objekti su prikazani onim redoslijedom kojim su kreirani, ali ih je moguće pomicati prema vlastitom izboru. Korisnik treba obratiti pozornost na odnos *Roditelj-Dijete* (eng. *Parent-Child*). Svaki objekt unutar scene može biti roditelj drugim objektima koje onda nazivamo djecom. Djeca nasljeđuju određena svojstva roditelja, odnosno ti se objekti pomiču, rotiraju i skaliraju onako kako to čini roditelj. Objekti koji su postavljeni kao djeca i sami mogu postati roditelji, odnosno imati niz drugih objekata koji će biti njihova djeca [16].

Prikaz scene (eng. *Scene View*) nalazi se u samoj sredini sučelja i pruža prikaz scene iz različitih perspektiva. Prikaz scene korisniku omogućava odabir, postavljanje i upravljanje objektima u sceni [17].

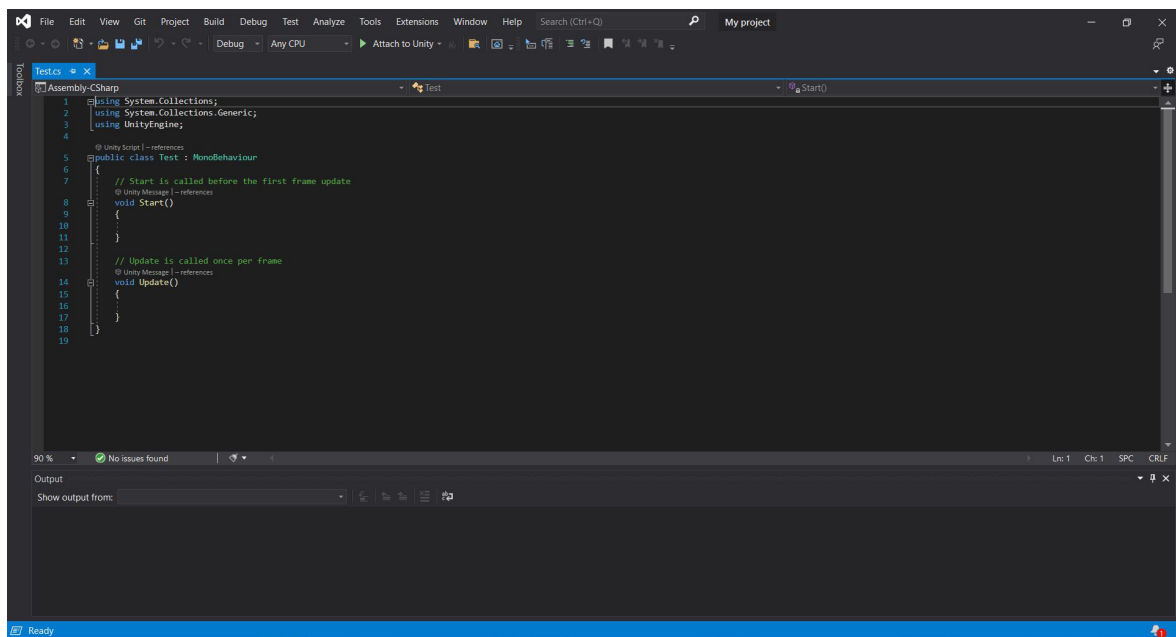
Pregledni prozor (eng. *Inspector Window*) nalazi se na desnoj strani sučelja i korisniku pruža informacije (npr. položaj, rotacija, veličina, itd.) o odabranom objektu. Korisnik u ovom prozoru također može upravljati istima, odnosno modificirati ih. Osim toga, korisnik može dodavati nove skriptne komponente (eng. *Script Components*) i uređivati njihove varijable, tj. mijenjati njihove parametre, a da pritom ne mijenja njihov kod [18].

Alatna traka (eng. *Toolbar*) nalazi se pri vrhu sučelja i sadrži alate za navigaciju i upravljanje projektom. Alatna traka može se podijeliti na tri dijela, pri čemu alati s lijeve strane služe za upravljanje objektima na sceni. Na sredini trake nalaze se alati za pokretanje, pauziranje i kontrolu koraka (eng. *Play, Pause and Step Control*). Alati s desne strane trake obuhvaćaju mogućnost povezivanja s Unity Cloud Servi-

ces, postavke korisničkog računa te mogućnost upravljanja slojevima (eng. *Layers*) i rasporedom sučelja (eng. *Layout*) [19].

3.2 Microsoft Visual Studio

Microsoft Visual Studio je integrirano razvojno okruženje (eng. *Integrated Development Environment*, IDE) koje omogućava izradu raznih računalnih programa i mobilnih aplikacija. Razvoj Visual Studia počinje 1997. godine s namjerom stvaranja razvojne okoline koja će omogućiti korištenje različitih programskih jezika unutar jednog i jedinstvenog alata [20]. U svojoj najnovijoj verziji Visual Studio predstavlja idealni i univerzalni (a zbog toga i najpopularniji) alat za razvoj aplikacija na Windows platformi. Neke od mogućnosti koje Visual Studio nudi su izrada HTML stranica, njihovo dizajniranje korištenjem CSS-a, korištenje JavaScripta te programiranje dinamičkog djela na samom web poslužitelju [20]. Microsoft Visual Studio može se besplatno preuzeti sa svoje službene web stranice [21].



Slika 3.2: Prikaz sučelja Microsoft Visual Studia.

3.3 Vuforia

Vuforia je komplet za razvoj softvera (eng. *Software Development Kit*, SDK) koji omogućava izradu aplikacija s proširenom stvarnosti za mobilne uređaje. Koristeći tehnologije računalnog vida, Vuforia aplikaciji omogućava prepoznavanje slika, objekata i prostora u svrhu konfiguracije aplikacije za interakciju sa stvarnim svijetom [22]. Paket Vuforia moguće je besplatno preuzeti s njihove službene web stranice [23]. Za preuzimanje paketa Vuforia potrebno se prethodno registrirati na istoj stranici. Vuforia nije zaseban program, nego paket koji je potrebno instalirati unutar neke razvojne platforme, npr. unutar Unity platforme. Vuforia se u Unity može jednostavno instalirati pomoću *drag and drop* metode (povuci i ispusti), tj. povlačeći preuzeti paket i ispuštajući ga u prozor projekta u Unityju. Nakon uspješne instalacije Vuforie, potrebno je kreirati određene okidače koji će aktivirati pojavljivanje objekata. Ti okidači mogu se temeljiti na slikama (*Image Targets*) ili objektima (*Cylinder Targets* i *Model Targets*). *Image Targets* ili ciljane slike su slike koje Vuforia može detektirati i pratiti u stvarnom prostoru. Vuforia uspoređuje značajke slike s kamere mobilnog uređaja s bazom podataka ciljanih slika. Detektiranjem određene ciljane slike, na ekranu mobilnog uređaja prikazati će se objekt (stvoren u Unityju) povezan s tom ciljanom slikom. Ciljane slike omogućavaju nam da na jednostavan način sadržaj proširene stvarnosti postavimo na ravne objekte kao što su fotografije, sličice, igraće karte [24] i sl.

3.3.1 Kreiranje ciljanih slika (eng. *Image Targets*)

Kod kreiranja ciljanih slika, korisnik mora uzeti u obzir niz čimbenika koji određuju njihovu kvalitetu. Kvalitetu ciljane slike moguće je očitati na temelju broja zvjezdica (od 1 do 5) iz *Vuforia Target Managera* u koji svaka ciljana slika treba biti prethodno učitana. Skup svih učitanih ciljanih slika u *Vuforia Target Manageru* čini bazu podataka koju je potrebno preuzeti i zatim uvesti u Unity *drag and drop* metodom. Za preuzimanje baze podataka potrebno je preuzeti ključ licence (eng. *License Key*) klikom na opciju *Get Basic* u *License Manageru*. Isti je ključ potrebno kopirati u Unity ako ciljane slike preuzimamo pojedinačno. Ako slike preuzimamo kao bazu podataka, ključ će već biti ugrađen u istoj bazi, pa ga nije potrebno zasebno kopirati. Neki od čimbenika koji će utjecati na kvalitetu (tj. broj zvjezdica) ciljane slike su: bogat-

stvo detalja (na slici), kontrast (koji se odnosi na što veću razliku između svijetlih i tamnih elemenata), uzorci koji se ne ponavljaju (tj. jedinstvene značajke i grafike) i format (koji mora biti 8- ili 24-bitni PNG i JPG format veličine manje od 2 MB) [25]. Dakle, broj zvjezdica biti će veći ako će elementi ciljane slike imati oštre rubove i šiljaste vrhove, visoki kontrast te motive koji se neće ponavljati, u što smo se i sami uvjerali prilikom izrade naše aplikacije. Značajke koje se nalaze na rubovima ciljane slike, Vuforia neće uzimati u obzir. To se područje slike naziva *Feature-Exclusion Buffer* i iznosi oko 8 % širine sa svake strane slike [25].

3.4 Blender

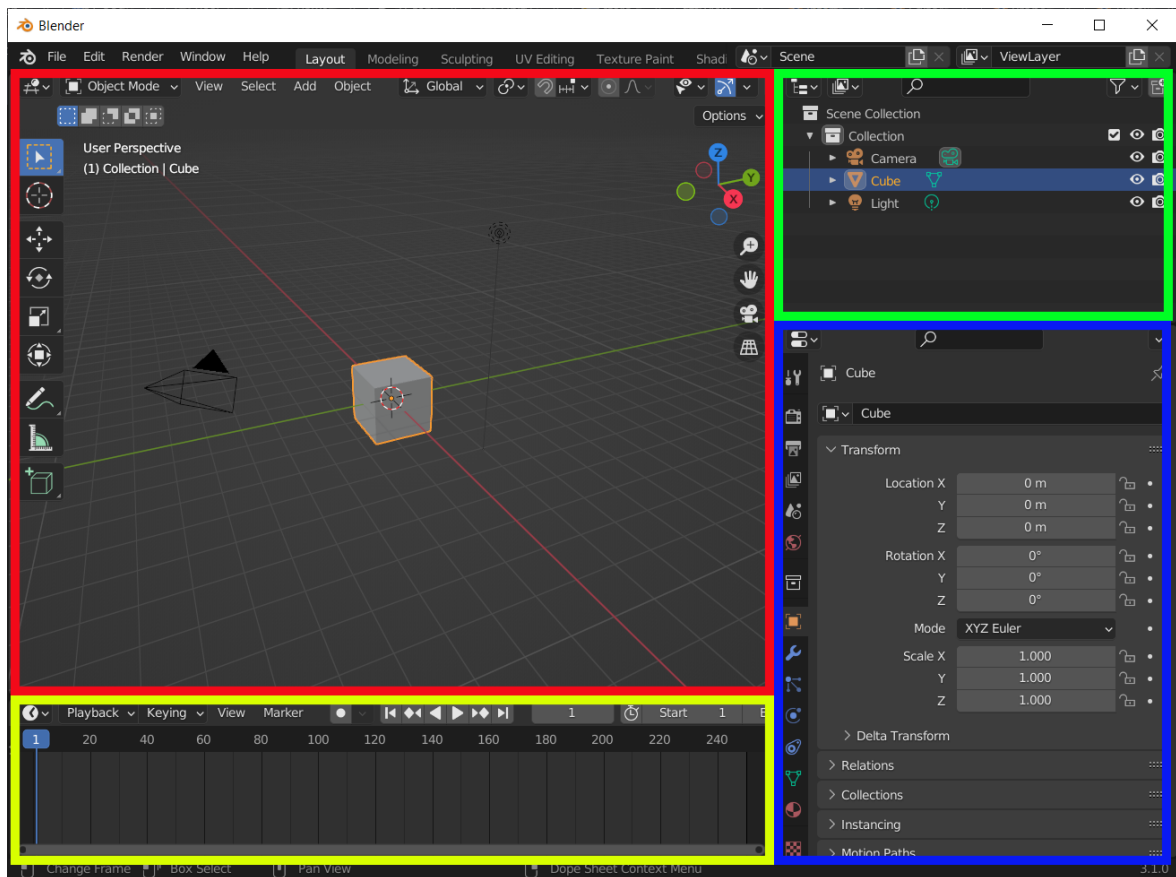
Blender je besplatan softver otvorenog koda (eng. *Open Source Software*) koji omogućava 3D modeliranje, 2D i 3D animiranje, simuliranje, video i audio uređivanje, itd. Riječ je o višeploatformskom (eng. *cross-platform*) softveru koji se može koristiti na Linux, Windows i Mac operativnim sustavima [26]. Blender se može besplatno preuzeti sa službene stranice [27].

Početno stanje sučelja prikazano na slici 3.3 podijeljeno je na nekoliko dijelova iliti uređivača (eng. *Editors*): 3D prikaz (eng. *3D Viewport*), Vremensku crtu (eng. *Timeline*), *Outliner*, Svojstva (eng. *Properties*). Izgled sučelja može se mijenjati što znači da se postojeći uređivači mogu micati i dodavati neki drugi (npr. *Image Editor*, *UV Editor*, *Compositor*, *Movie Clip Editor*, *Text Editor*, itd.) [27].

3D prikaz (eng. 3D Viewport) omogućava interakciju korisnika s trenutnom 3D scenom, odnosno objektima koji se nalaze unutar iste. U 3D prikazu korisnik može modelirati, animirati, stvarati i uređivati teksture, stvarati i uređivati video sadržaje, itd [28].

Vremenska crta (eng. Timeline) je uređivač koji se koristi za upravljanje ključnim kadrovima (eng. *keyframes*) unutar animacije. Vremenska crta korisniku pruža širok pregled animacije, odnosno prikazuje trenutni kadar, početni i završni kadar te ostale ključne kadrove animacije. Vremenska crta također uključuje opcije za pokretanje, pauziranje i preskakanje animacija [29].

Outliner je hijerarhijski popis svih datoteka i objekata koje se nalaze unutar trenutnog projekta. *Outliner* korisniku omogućava pregledavanje i upravljanje objektima (brisanje, omogućavanje ili onemogućavanje odabira u 3D prikazu, grupiranje,



Slika 3.3: Prikaz sučelja Blendera koji sadrži 3D prikaz (označen crvenom bojom), *Outliner* (označen zelenom bojom), *Svojstva* (označen plavom bojom) i *Vremensku* crtu (označenu žutom bojom).

itd.) [30].

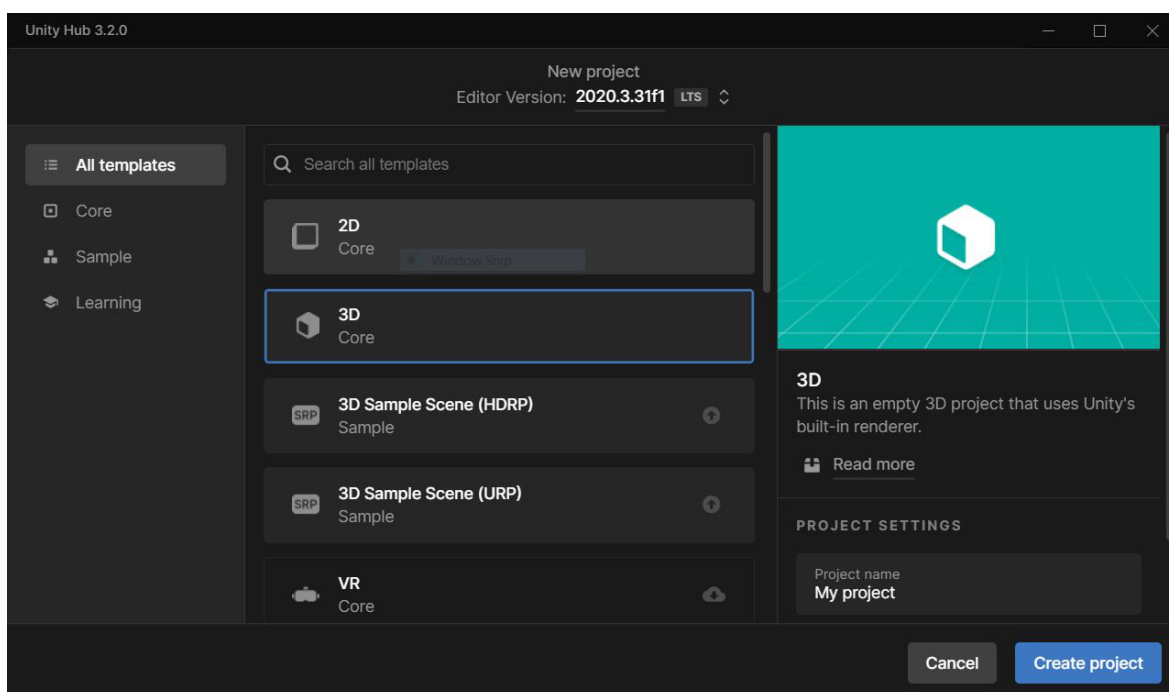
Svojstva (eng. *Properties*) se odnose na uređivač koji korisniku omogućuje uređivanje svojstava odabranih objekata i scene. Neka od svojstava su: svojstva modifikatora (eng. *Modifier Properties*), svojstva materijala (eng. *Material Properties*), svojstva scene (eng. *Scene Properties*), svojstva objekta (eng. *Object Properties*), [31] itd.

4 Izrada aplikacije

Nakon teorijskog djela ovog diplomskog rada u kojemu smo predstavili tehnologije i alate koji se koriste za izradu aplikacija, u praktičnom djelu rada koji slijedi opisati ćemo i objasniti kako smo iste koristili za izradu naše aplikacije.

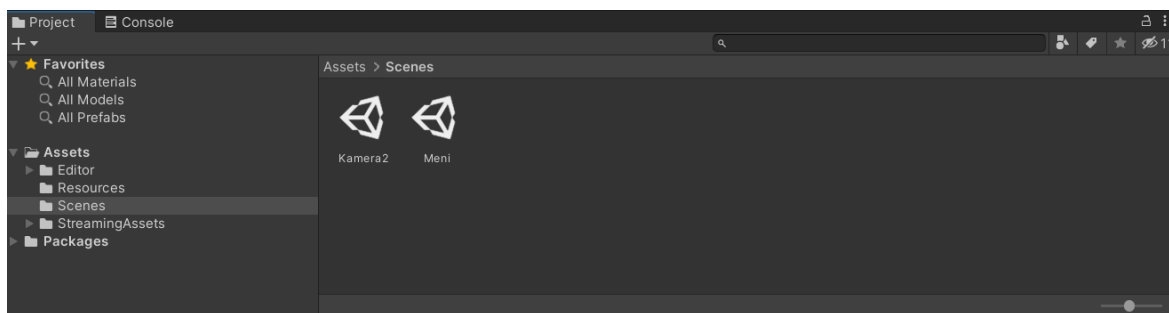
4.1 Stacionarni modeli

Nakon preuzimanja i instalacije navedenih tehnologija, izrada naše aplikacije započinje u Unity programu. Prilikom instalacije Unityja, automatski se instalira i Unity Hub prikazan na slici 4.1. Unity Hub omogućava korisniku da započne s izradom novog projekta (izabere postojeći *template* ili kreira novi) ili da pokrene već postojeći projekt. Iako Unity Hub nudi već stvorene *templates* koji omogućavaju rad s proširenom stvarnosti, za izradu naše aplikacije stvorili smo novi 3D projekt.



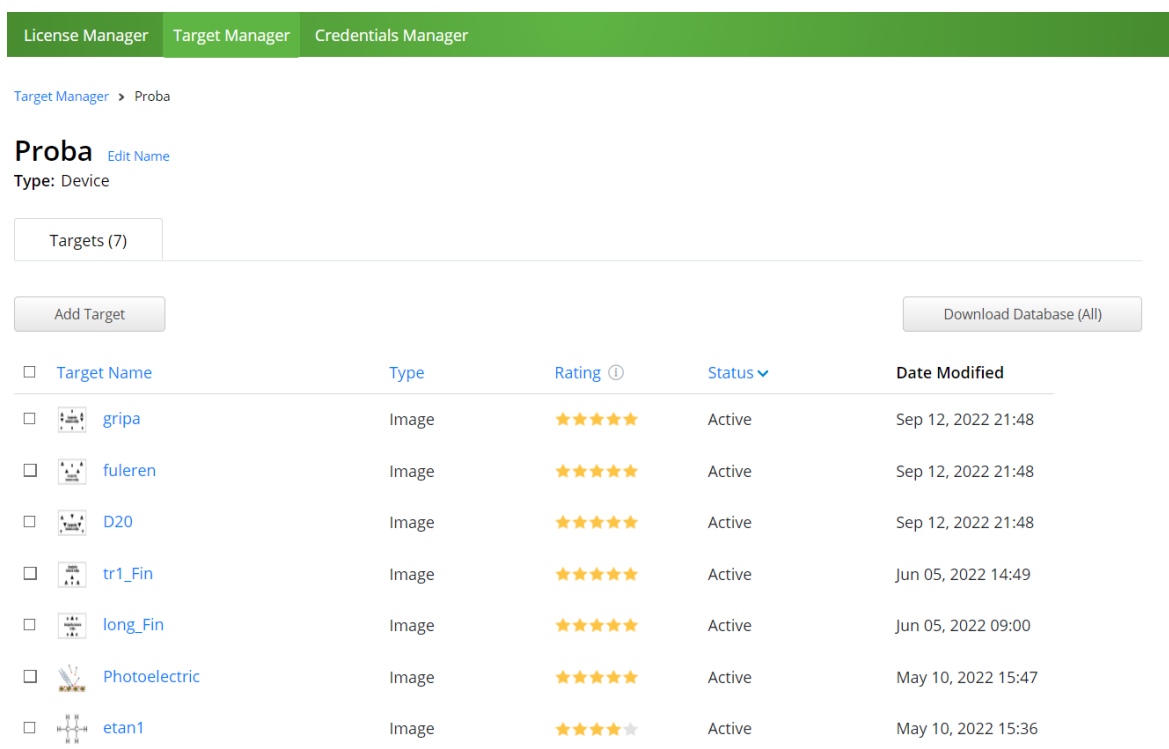
Slika 4.1: Prikaz Unity Huba prilikom prvog pokretanja.

Jedan projekt može sadržavati više scena koje se onda mogu izmjenjivati (uglavnom pomoću skripti). Kod naše aplikacije koristili smo samo dvije scene (prikazane na slici 4.2). Prvu scenu nazvali smo *Meni* i ona nam je predstavljala glavni izbornik. Drugu scenu nazvali smo *Kamera2* i njezina je glavna uloga bila detektirati ciljane slike (eng. *Image Targets*) i pokazati na ekranu objekte povezane s tim slikama.



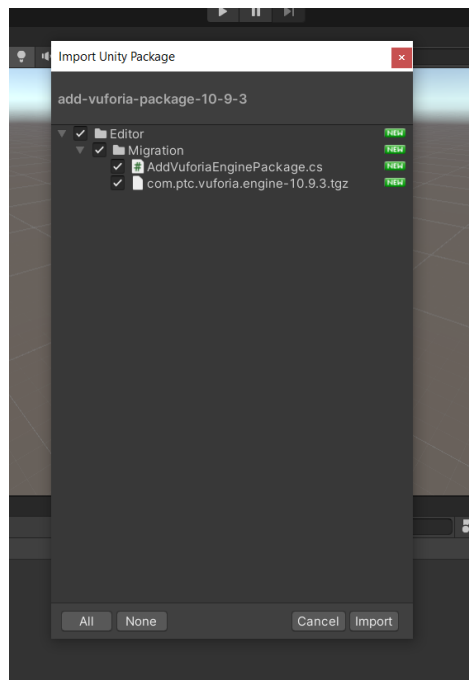
Slika 4.2: Prikaz scena koje sadržava aplikacija.

Ciljane slike (eng. *Image Targets*) preuzeli smo iz baze podataka kreirane u *Vuforia Target Manageru* (prikazana na slici 4.3). Bazu podataka kreirali smo učitavanjem zasebnih ciljanih slika u *Vuforia Target Manager*, birajući pritom one ciljane slike koje su na temelju ranije spomenutih čimbenika imale što veći broj zvjezdica. Kreiranu bazu podataka preuzeli smo i uveli u Unity *drag and drop* metodom.



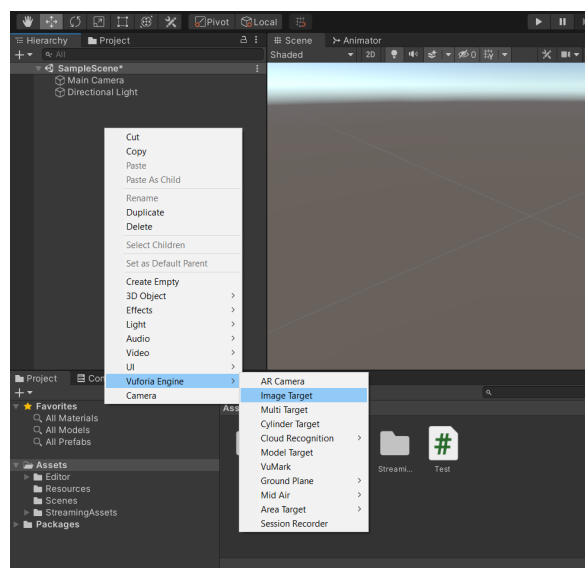
Slika 4.3: Prikaz baze podataka ciljanih slika.

U novokreirani projekt zatim smo uveli (eng. *import*) Vuforia.



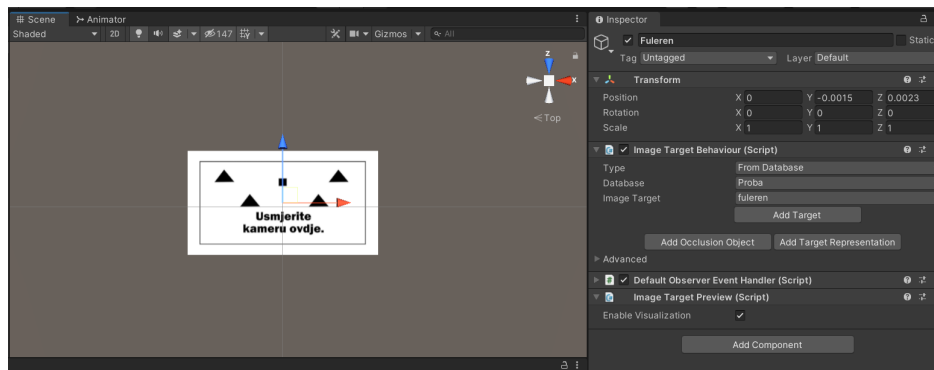
Slika 4.4: Prikaz uvoza paketa Vuforia u Unity.

Koristeći Vuforia unutar Unityja stvorili smo objekte ciljanih slika (eng. *Image Targets*). Za stvorene objekte postoji mogućnost preimenovanja, stoga smo ih preimenovali u ime onog 3D modela kojeg smo željeli da aplikacija prikaže detektiranjem određene ciljane slike.



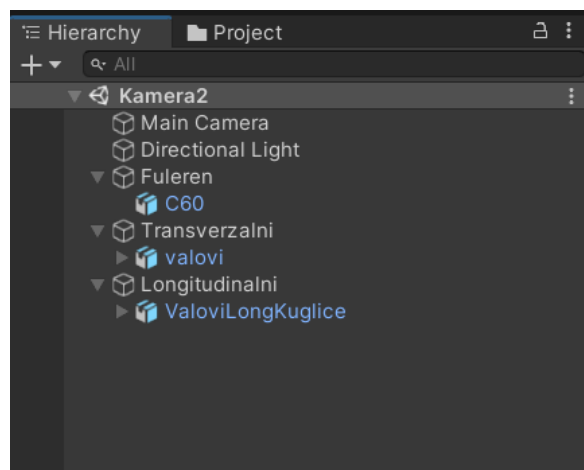
Slika 4.5: Prikaz kreiranja objekta *Image Targets*.

Na primjer, jednog od objekata nazvali smo Fuleren jer će detektiranjem određene ciljane slike aplikacija prikazati 3D model fulerena (objekt ciljane slike Fuleren prikazan na slici 4.6).



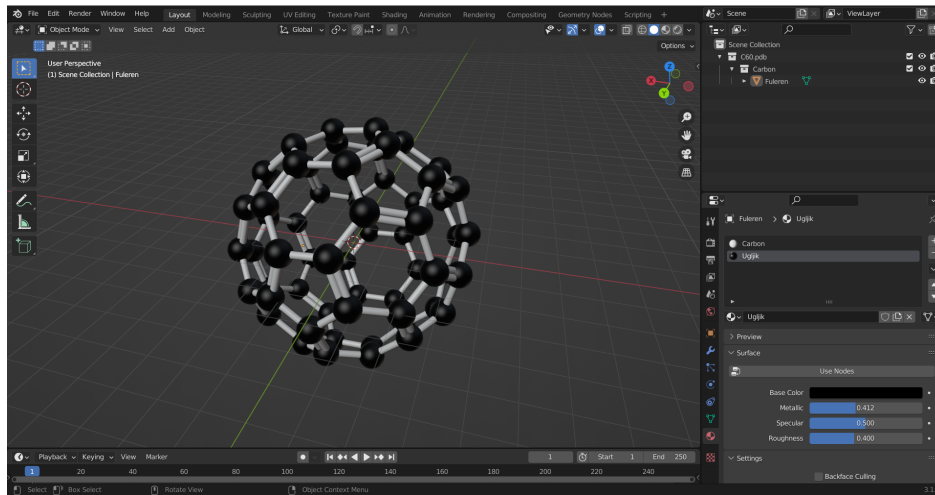
Slika 4.6: Prikaz objekta *Image Targets* preimenovanog u Fuleren.

Objekt 3D modela fulerena (C60) uveli smo u *Assets* (mapu u prozoru projekta), a zatim ga *drag and drop* metodom postavili u scenu kao dijete (eng. *Child* u odnosu *Parent-Child*) objekta ciljane slike Fulerena (eng. *Parent*) kako bi aplikacija detektiranjem te iste ciljane slike Fulerena na ekranu mobilnog uređaja prikazala 3D model fulerena (C60).

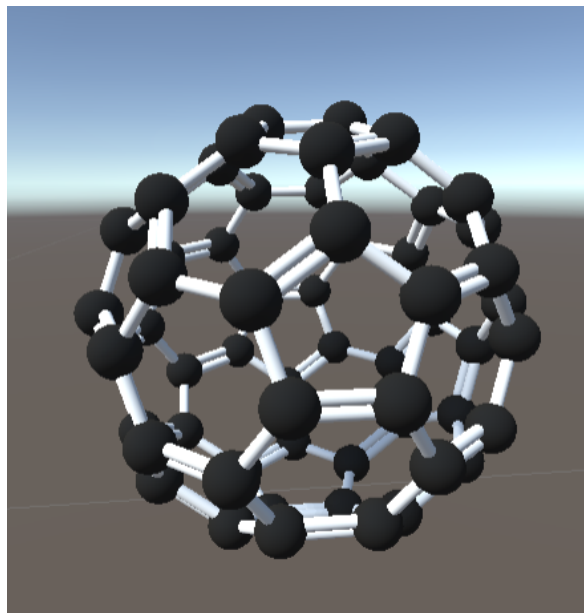


Slika 4.7: Prikaz odnosa *Parent-Child* unutar hijerarhijskog prozora.

3D model fulerena modeliran je u Blenderu (slika 4.8). Iz Blendera smo ga izveli (eng. *export*) kao datoteku koju je Unity mogao prepoznati kao cjeloviti 3D objekt (slika 4.9) kako bi kasnije mogli njime upravljati (mijenjati njegovu veličinu, rotirati ga, pomicati ga, itd.).



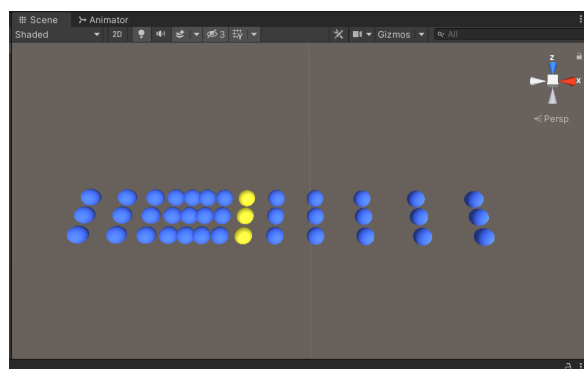
Slika 4.8: Prikaz 3D modela fulerena unutar Blendera.



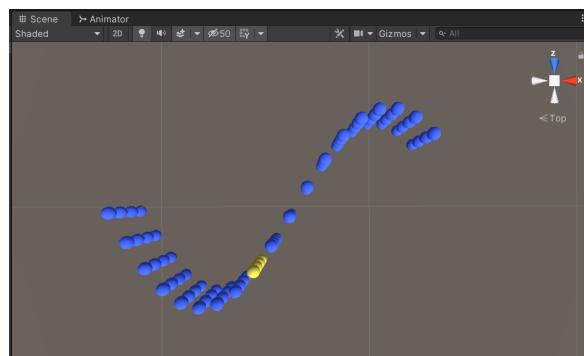
Slika 4.9: Prikaz 3D modela fulerena unutar Unityja.

4.2 Nestacionarni modeli

U prethodnom potpoglavlju objasnili smo postupak kreiranja ciljanih slika (eng. *Image Targets*) za stacionarne modele kao što je fuleren. U ovom ćemo potpoglavlju opisati postupak kreiranja nestacionarnih modela, odnosno pokazati što je još potrebno učiniti kako bi aplikacija kreirala objekte koji nisu stacionarni i time omogućila korisnicima promatranje animiranog 3D modela koji predstavlja neki proces ili pojavu. Za primjer nestacionarnih modela koristili smo valove za koje smo kreirali objekte ciljanih slika pod nazivima Longitudinalni i Transverzalni. O valovima i njihovim modelima više ćemo govoriti u poglavlju kasnije.

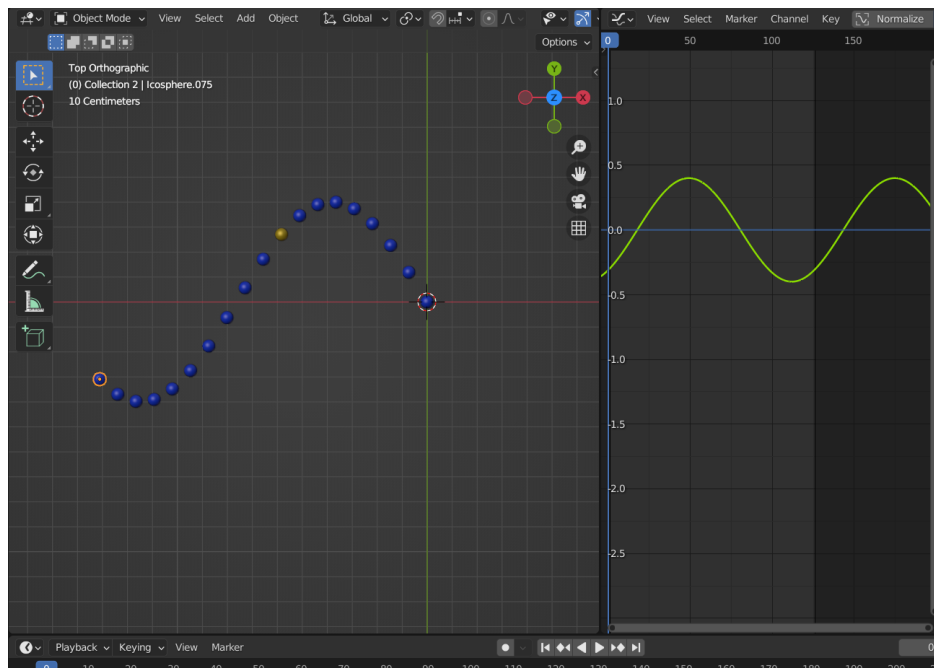


Slika 4.10: Prikaz 3D modela longitudinalnog vala unutar Unityja.



Slika 4.11: Prikaz 3D modela transverznog vala unutar Unityja.

3D modeli valova također su kreirani u Blenderu. Međutim, budući da smo željeli da se kreirani valovi „gibaju“, trebali smo animirati 3D objekte (kuglice) koje sačinjavaju valove. Animacije su rađene u Blenderu (slika 4.12). Svakoj kuglici u modelu bilo je potrebno pridružiti odgovarajuću putanju kojom će se kretati za vrijeme animacije. Blender ima nekoliko *Built-In* funkcija koje omogućavaju oscilirajuća kretanja za vrijeme trajanja animacije. Svakoj od kuglica pridružili smo funkciju sinus i odgovarajuće parametre, amplitudu i fazni pomak.



Slika 4.12: Prikaz animiranja vala u Blenderu.

Zatim smo animacije kreirane u Blenderu izveli (eng. *bake animation*) u Unity. Kako bi Unity mogao koristiti animaciju koju smo kreirali u Blenderu, potrebno je 3D modelu vala u preglednom prozoru objekta (eng. *Inspector Window*) dodati komponentu pod nazivom Animator (eng. *Animator component*). Komponenta Animator upućuje na *Animator Controller* i *Avatar* [32].

Animator Controller omogućava upravljanje animacijama, odnosno određivanje redoslijeda izvođenja animacija [33].

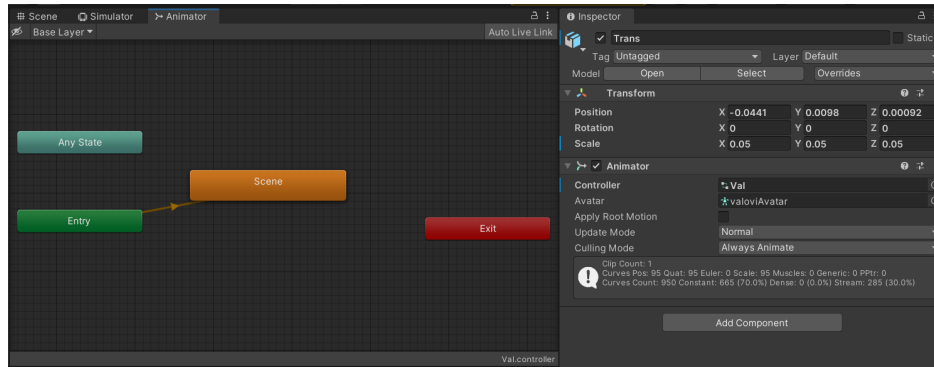
Avatar je prozor u Unityju unutar kojega odabiremo koji će se dijelovi objekta micati (eng. *Rigging*). No, ovaj smo dio već „odradili“ u Blenderu jer smo tamo odredili koji će se dijelovi valova (koje kuglice) i na koji način gibati (harmonijsko titranje) [34]. Ovaj se dio znatno zakomplicira kada je riječ o kompleksnijim animacijama, odnosno o animacijama koje imaju više elemenata.

Budući da smo sav „posao“ oko animacija odradili u Blenderu i izvezli ih u Unity, unutar *Animator Controllera* bilo je potrebno samo dodati gotovu animaciju što je na slici 4.13 prikazano narančastim pravokutnikom *Scene*.

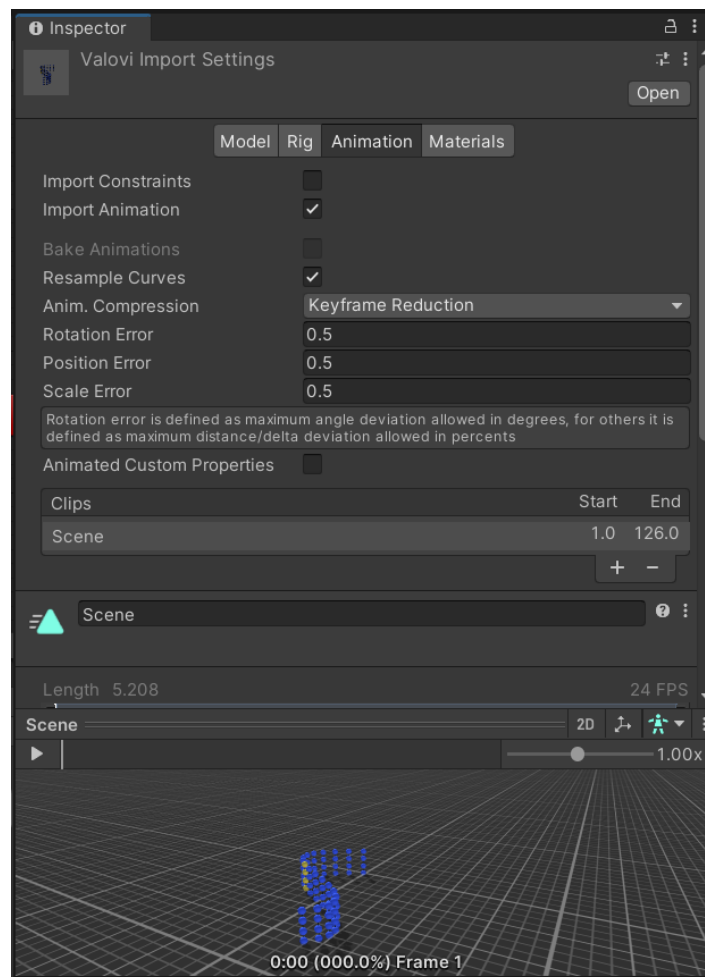
Uređivanjem smo dobili animaciju koja se nakon određenih ključnih kadrova (eng. *keyframes*) ponavlja. Animacija će trajati, tj. val će se gibati svo vrijeme dokle je ciljana slika (kao okidač) detektirana kamerom.

Kao i kod Fulerena, izvezen objekt čini cjelinu kojom možemo upravljati unutar

preglednog prozora (eng. *Inspector Window*), tj. rotirati ga, skalirati, pomicati, itd. Objektima kuglica koji čine valove ne može se zasebno upravljati.



Slika 4.13: Prikaz izgleda prozora *Animator Controller*.

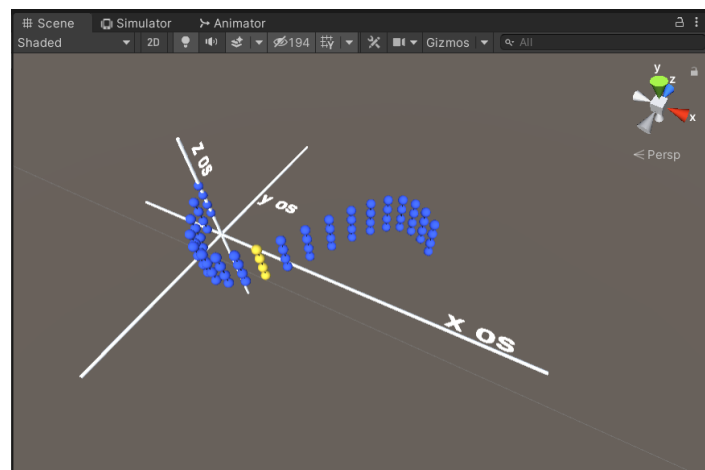


Slika 4.14: Prikaz animacije unutar preglednog prozora.

4.3 Proširenje modela

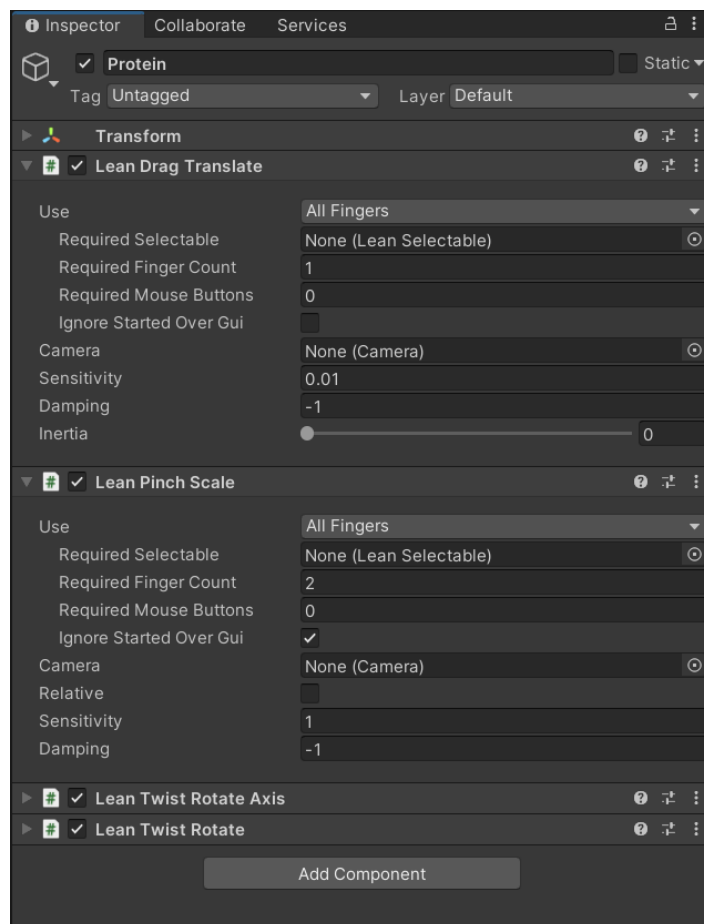
Kako bi korisnik aplikacije bolje razumio u kojem se smjeru val širi, odnosno koji se elementi gibaju u kojim smjerovima, potrebno je napraviti referentni koordinatni sustav. Koordinatni sustav odnosi se na objekt sačinjen od tri cilindra koja predstavljaju x, y i z os.

Koordinatni sustav mora biti dijete objekta ciljane slike kako bi se na ekranu mobilnog uređaja prikazao zajedno s objektom 3D modela, u našem slučaju vala. Kako bi pojednostavili nastavak izrade aplikacije i omogućili uvođenje novih funkcionalnosti, kreirali smo prazan objekt pod nazivom *Val+os* koji je roditelj koordinatnog sustava (osi) i 3D modela vala te dijete objekta ciljane slike.



Slika 4.15: Prikaz modela vala (*Val+os*) nakon dodavanja koordinatnih osi.

Kako bi korisnik imao bolje iskustvo pri korištenju aplikacije, potrebno mu je omogućiti interakciju s objektima koje aplikacija prikazuje, odnosno omogućiti mu da izravnim dodirrom zaslona rotira i skalira objekt. Zbog toga smo preko Unity Asset Storea preuzeli *Lean Touch* [35] paket, koji nam omogućava brz i jednostavan način dodavanja naredbi za upravljanje zaslonom osjetljivim na dodir, i uveli ga u Unity. Komponente (skripte) iz *Lean Touch* paketa dodali smo modelima za koje želimo da ih korisnik može rotirati i skalirati.



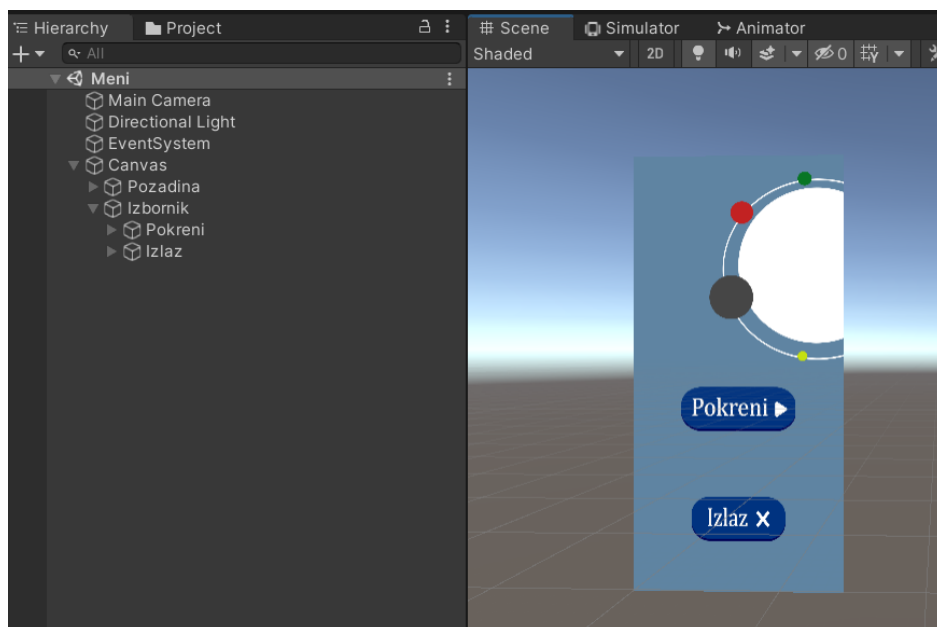
Slika 4.16: Prikaz *Lean Touch* komponenata objekta *Protein* unutar preglednog prozora.

4.4 Korisničko sučelje (eng. User Interface, UI)

Svaka aplikacija posjeduje neki oblik korisničkog sučelja. Korisničko sučelje omogućuje korisniku interakciju s aplikacijom.

4.4.1 Izrada korisničkog sučelja

Unity je znatno pojednostavnio izradu korisničkog sučelja. U hijerarhijskom prozoru (eng. *Hierarchy Window*) potrebno je stvoriti objekt *Canvas*. *Canvas* je područje koje mora sadržavati sve one elemente koje želimo da aplikacija prikazuje. Svaki element koji želimo da se prikazuje mora biti dijete objekta *Canvas* [36].



Slika 4.17: Prikaz odnosa *Parent-Child* u hijerarhijskom prozoru i izgled glavnog izbornika.

Na slici 4.17 prikazano je korisničko sučelje naše aplikacije. Objekti koje ono sadrži su pozadina i gumbi (eng. *Buttons*) *Pokreni* i *Izlaz*. Pozadina glavnog izbornika odnosi se na objekt *Image* koji korisniku prikazuje neinteraktivnu sliku [37].

Objekt *Image* preimenovali smo u objekt *Pozadina*, a slika koju taj objekt prima (ona koju želimo da aplikacija prikazuje) mora biti 2D grafičkog oblika, tj. oblika *Sprite* [15].

Gumbi *Pokreni* i *Izlaz* tipični su interaktivni objekti. Reagirati će jednom kada se pritisnu, bez obzira koliko dugo ostaju pritisnuti, ali se akcija izvršava tek kada se gumb otpusti [16]. Izgled i naziv gumba moguće je promijeniti, pri čemu se za promjenu izgleda mora koristiti *Sprite*.

Kako bi glavni izbornik ispravno funkcionirao, potrebno je dodati prazan objekt koji će biti dijete objekta *Canvas*, a roditelj interaktivnih tipki *Pokreni* i *Izlaz*, te imati komponentu (skriptu) koja će voditi brigu o tome jesu li i koji su gumbi pritisnuti. Nakon stvaranja takvog objekta, preimenovali smo ga u *Izbornik*.

Spomenutu skriptu (slika 4.18) kreirali smo pomoću Microsoft Visual Studia. Jezik koji smo koristili za pisanje skripti je C#. Jezik C# je objektno orijentirani jezik. Pisanje skripti za potrebe ove aplikacije zahtijevalo je poznavanje rada s klasama, objektima [17], itd.

```
Assembly-CSharp | GlavniIzbornik
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6
7  public class GlavniIzbornik : MonoBehaviour
8  {
9
10     public void Pokreni()
11     {
12         SceneManager.LoadScene("Kamera2", LoadSceneMode.Single);
13     }
14
15     public void Izlaz()
16     {
17         Application.Quit();
18     }
19
20
21     public void Izlaz_u_meni()
22     {
23         SceneManager.LoadScene("Meni", LoadSceneMode.Single);
24     }
25 }
```

Slika 4.18: Prikaz skripte *GlavniIzbornik* i njezinih metoda unutar Microsoft Visual Studia.

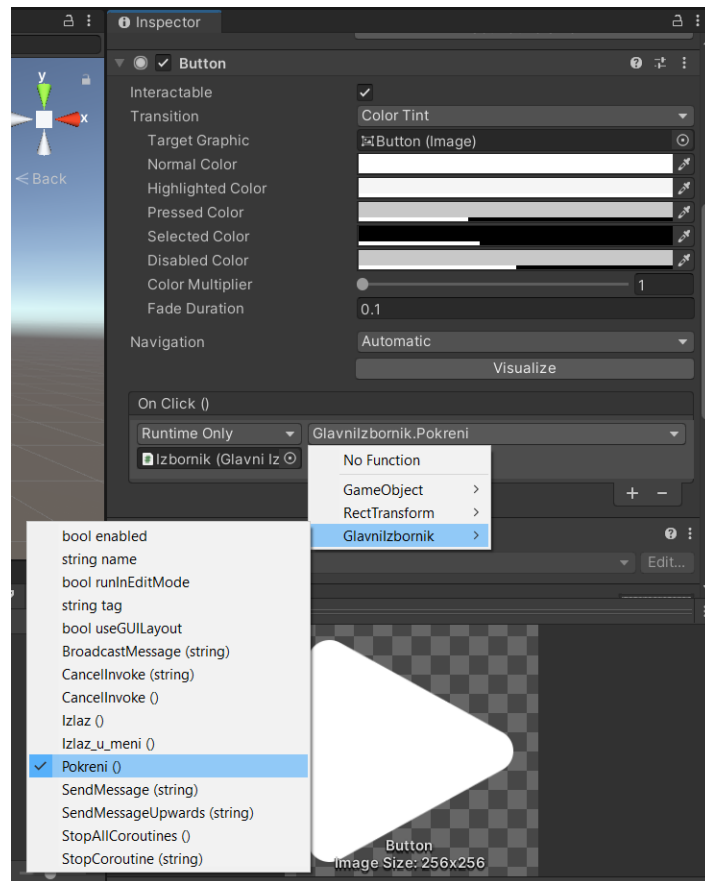
Klasa *GlavniIzbornik* sadrži nekoliko metoda. Svaka od ovih metoda deklarirana je kao javna (eng. *public*) što znači da će toj metodi moći pristupiti svi objekti koji će imati komponentu *GlavniIzbornik*.

Kako bi upravljanje scenama bilo moguće, potrebno je klasu *SceneManager* uključiti u skriptu *GlavniIzbornik* pozivom *using UnityEngine.SceneManagement*. Klasa *SceneManager* ima nekoliko metoda, a za potrebe aplikacije koristili smo metodu *LoadScene*. Ova metoda omogućava pokretanje scene pomoću njezinog imena ili redosljeda scene. Kako bi kod bio pregledniji, koristili smo opciju učitavanja scene pomoću naziva scene.

Metoda *Pokreni()* je metoda koju smo dodali na gumb *Pokreni* kako bi aplikacija otvorila scenu *Kamera2*.

Metoda *Izlaz()* je metoda kojom izlazimo iz aplikacije, a metoda *Izlaz_u_meni()*

koju smo dodali gumbu unutar scene *Kamera2* je metoda pomoću koje izlazimo iz scene *Kamera2* u scenu *Meni* (glavni izbornik). Metoda *Izlaz()* koristi metodu *Application.Quit* koja je dio osnovne klase *UnityEngine* (koju sadrže sve aplikacije rađene u Unityu).

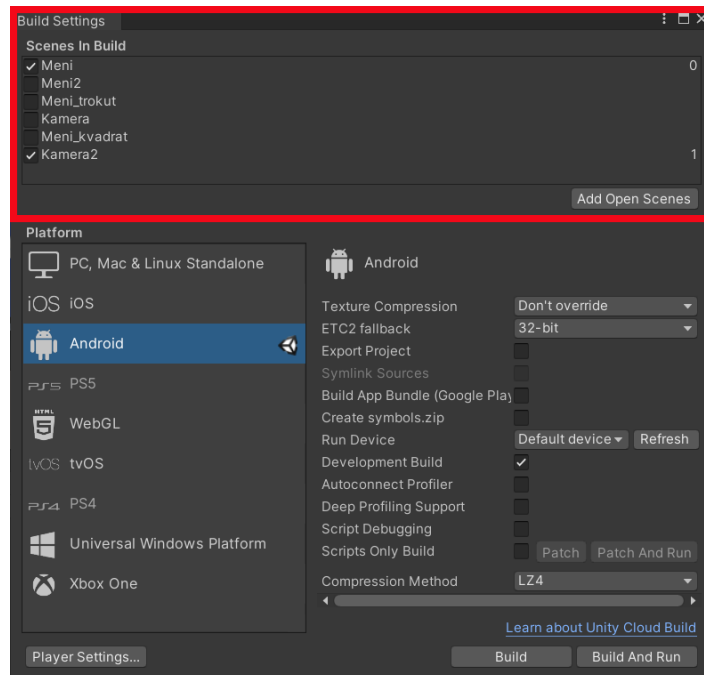


Slika 4.19: Prikaz javnih metoda koje sadrži klasa *GlavniIzbornik*.

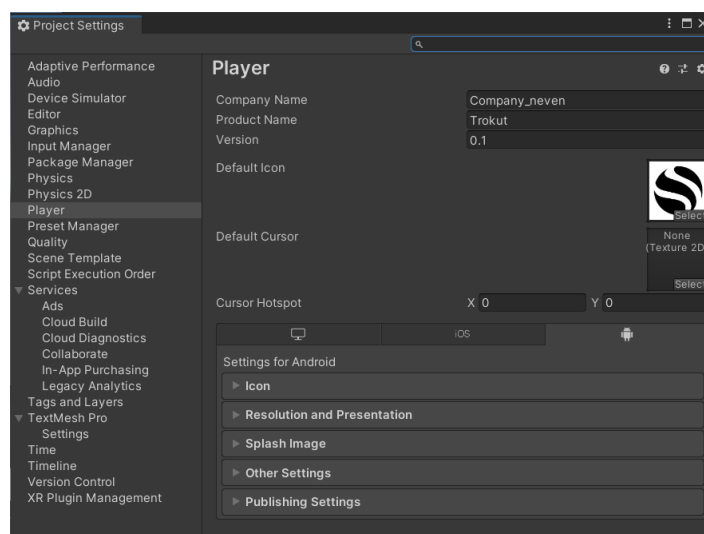
4.5 Priprema aplikacije za izvoz na pametni telefon

Prije nego aplikaciju izvezemo na pametni telefon, potrebno je u izborniku *Build settings* [41] konfigurirati određene funkcije. Prije svega potrebno je odabrati platformu za koju želimo izgraditi (eng. *build*) aplikaciju te odabrati scene koje želimo da aplikacija sadržava. Ono što je bitno kod scena je to da moraju poštovati hijerarhiju. Ona scena koja se nalazi na prvom mjestu popisa scena biti će prva prilikom pokretanja aplikacije. Hijerarhija scena prikazana je na slici 4.20 unutar crvenog okvira.

Jedna od opcija unutar izbornika *Build settings* je opcija *Player Settings* [42] unutar koje možemo definirati razne parametre koji se odnose na aplikaciju, npr. ime aplikacije, urednika aplikacije, ikonu aplikacije itd.



Slika 4.20: Prikaz izbornika *Build settings* u kojem se obavlja odabir platforme i odabir scena (označeno crvenom bojom).



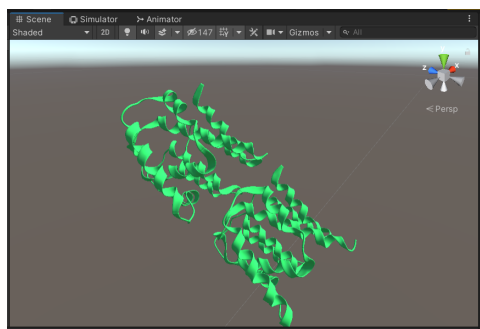
Slika 4.21: Prikaz opcija unutar prozora *Player settings*.

5 Primjena aplikacije u prirodoslovnim znanostima

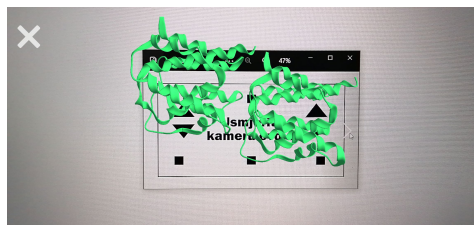
U ovome ćemo poglavlju prikazati nekoliko primjera primjene proširene stvarnosti u području prirodoslovnih znanosti koristeći prethodno opisanu aplikaciju. Navedeni primjeri vezani su za teme koje se obrađuju u školskim udžbenicima iz fizike, kemije, matematike i biologije. Primjeri su ilustrativne prirode.

5.1 Biologija

Za primjer iz biologije odabrali smo virus Influenze poznatiji pod nazivom gripa. Najzastupljeniji protein u virusu gripe je matrični protein (M1) koji tvori matrični sloj neposredno ispod lipidne ovojnice i koji povezuje virusne ribonukleoproteine [43]. S gradivom o virusima učenici se susreću u sedmom razredu osnovnoškolskog obrazovanja (prva godina učenja biologije) te trećem razredu srednjoškolskog obrazovanja (peta godina učenja biologije). Premda su dostupnost interneta i pristup raznim programima znatno pojednostavili prikaz i pregled virusa u virtualnom okruženju, prednost ove aplikacije je što (osim prikaza i pregleda virusa) učeniku omogućava interakciju s 3D modelom u stvarnom vremenu.



Slika 5.1: Prikaz modela M1 proteina unutar Unityja.

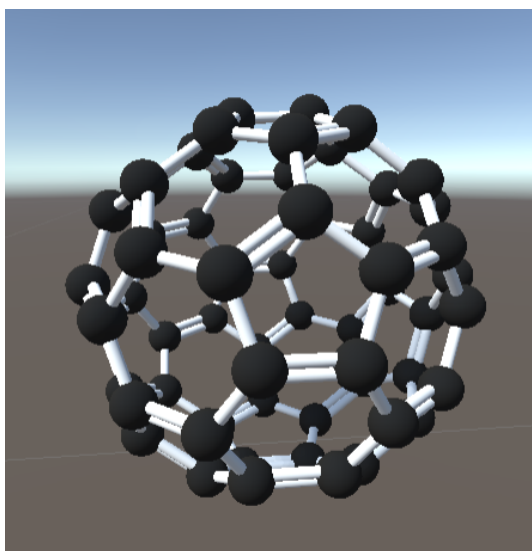


Slika 5.2: Prikaz modela M1 proteina kojeg prikazuje aplikacija.

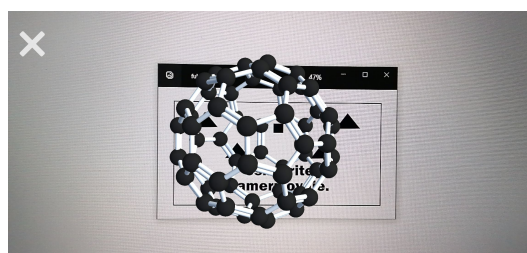
Na slici 5.1 prikazan je 3D model matričnog proteina (M1). S web stranice RCSBA-a (Research Collaboratory for Structural Bioinformatics) preuzeli smo PDB format matričnog proteina. Riječ je o američkom podatkovnom centru za globalnu banku podataka Protein Data Bank (PDB) koja sadrži 3D strukturne podatke bioloških makromolekula (proteina, DNA i RNA) [44]. Koristeći Blender, preuzeti PDB format pretvoren je u 3D model i uvezen u Unity. Primjer smo dovršili slijedeći postupke opisane u poglavlju o izradi aplikacije.

5.2 Kemija

Anorganski spojevi i njihova vizualizacija često učenicima predstavljaju velik problem, a s ovim se gradivom učenici susreću u osmom razredu osnovnoškolskog obrazovanja (druga godina učenja kemije) te četvrtom razredu srednjoškolskog obrazovanja (šesta godina učenja kemije). Alotropske modifikacije ugljika kao što su grafit i dijamant učenici lakše razumiju i vizualiziraju zbog njihove svakodnevne primjene i upotrebe. Međutim, to nije slučaj s modifikacijama poput fulerena i grafena s kojima se učenici ne susreću tako često. Fuleren je molekula sastavljena od 60 atoma ugljika (C₆₀) povezanih jednostrukim i dvostrukim vezama u šuplju sferu s 12 peterokutnih i 20 šesterokutnih stranica. Oblik molekule fulerena, čiji izgled podsjeća na nogometnu loptu, prikazan je na slici 5.3 [45].



Slika 5.3: Prikaz modela fulerena unutar Unityja.

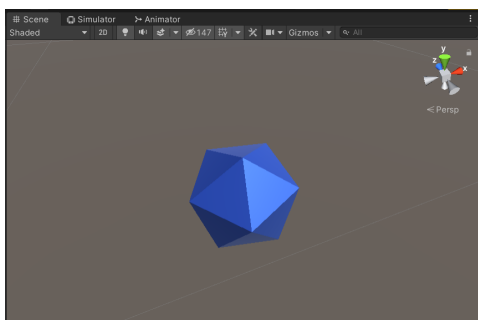


Slika 5.4: Prikaz modela fulerena kojeg prikazuje aplikacija.

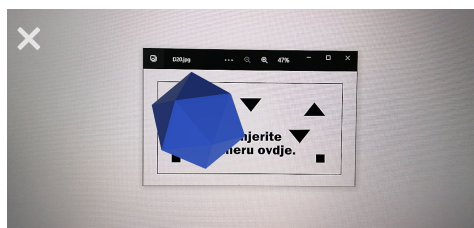
Fuleren smo koristili kao primjer u izradi aplikacije gdje su detaljnije opisani svi koraci.

5.3 Matematika

Iz matematike odabrali smo primjer ikosaedra, poliedra omeđenog s 20 ploha koji je jedan od pet Platonovih tijela. Pravilni ikosaedar čini 20 sukladnih jednakostraničnih trokuta, 12 vrhova i 30 bridova. Geometrijsko gradivo o Platonovim tijelima sastavni je dio nastavnog sadržaja matematike u osmom razredu osnovnoškolskog obrazovanja (osma godina učenja matematike) te drugom razredu srednjoškolskog obrazovanja (deseta godina učenja matematike). Učenici nerijetko nailaze na poteškoće u vizualiziranju konstruiranih tijela. Ova aplikacija učenicima olakšava vizualizaciju složenijih geometrijskih tijela te im omogućava promatranje tijela u bilo kojoj ravnini. Kao i ostali primjeri, 3D ikosaedra (prikazan na slici 5.5) modeliran je u Blenderu, izvezen u Unity, i dovršen na temelju opisanih postupaka za izradu aplikacije.



Slika 5.5: Prikaz modela ikosaedra unutar Unityja.



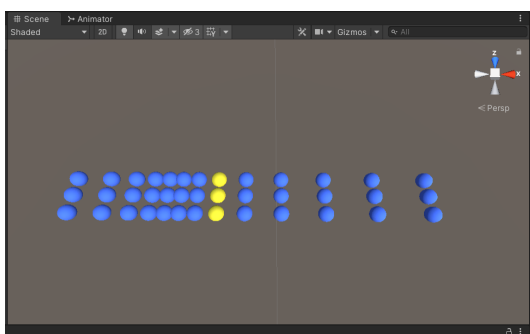
Slika 5.6: Prikaz modela ikosaedra kojeg prikazuje aplikacija.

5.4 Fizika

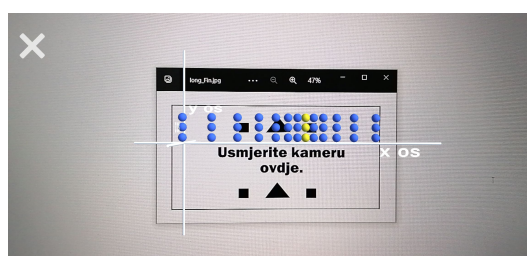
Sam koncept vala učenicima često zadaje probleme. Iako se s valovima susreću svakodnevno (elektromagnetski valovi, zvučni valovi, itd.), učenici ne razumiju u potpunosti što oni zapravo jesu niti kako nastaju. Učenici se često pitaju jesu li valovi objekti ili poremećaji koji se šire u prostoru. Kod definiranja samog koncepta vala učenicima zapravo problem predstavlja nerazumijevanje onoga što kod vala putuje. Učenici pogrešno smatraju da je samo materija ta koja može putovati i time prenositi energiju. Stoga i ne čudi da koncept mehaničkog vala koji prenosi energiju, ali ne i materiju, učenicima stvara velike nejasnoće. Primjerice, učenici kod longitudinalnih valova pogrešno smatraju da jedna čestica vala prenosi zvuk od izvora zvuka do slušatelja.

Premda se u ovom diplomskom radu nećemo detaljnije baviti odgovorima na ova pitanja, cilj nam je pomoću ove aplikacije omogućiti učenicima da uvide što se i kako giba kod mehaničkih valova.

Kreirali smo modele longitudinalnog (slika 5.7) i transverznog (slika 5.9) vala. 3D model longitudinalnog vala sastoji se od 12 plavih kuglica i 1 žute kuglice. U model smo dodali dva dodatna reda kuglica kako bi učenicima zornije predočili gibanje vala. Žute kuglice u valu pomažu učenicima da lakše uoče razliku između vala i gibanja pojedinih čestica vala.

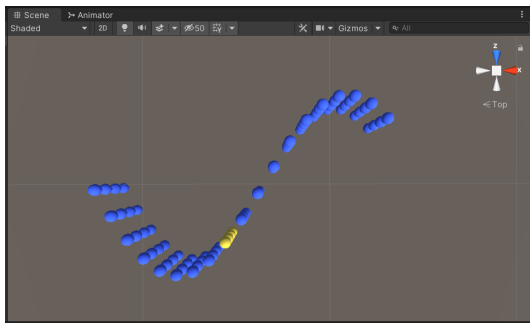


Slika 5.7: Prikaz modela longitudinalnog vala unutar Unityja.

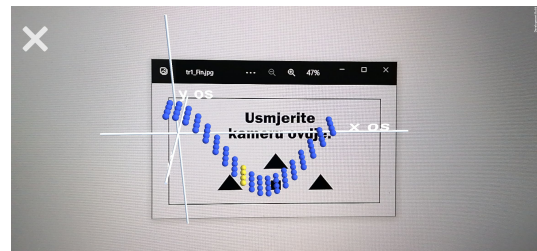


Slika 5.8: Prikaz modela longitudinalnog vala kojeg prikazuje aplikacija.

3D model transverzalnog vala sastoji se od 18 plavih kuglica i 1 žute kuglice. Kao i kod longitudinalnog vala, i kod ovog smo modela dodali još tri dodatna reda kuglica. Kuglice ovog modela osciliraju na način da formiraju sinusoidealni val. Svrha žutih kuglica u modelu ista je kao i kod longitudinalnog vala. Kada učenik usmjeri kameru na ciljanu sliku, aplikacija će prikazati animirani 3D model vala te mu omogućiti interakciju s tim valom u 3D svijetu. Učenik tako može rotirati i skalirati val, promatrati ga u kojoj god ravnini i sl.



Slika 5.9: Prikaz modela transverzalnog vala unutar Unityja.



Slika 5.10: Prikaz modela transverzalnog vala kojeg prikazuje aplikacija.

Aplikacija na taj način omogućava učenicima da uoče i razumiju kako se val širi kroz prostor, odnosno kako se čestice valova ne prenose s jednog mjesta na drugo (tako da bi za vrijeme trajanja vala stigli s početka na kraj), već osciliraju oko određene točke.

6 Zaključak

Mogućnosti koje nudi proširena stvarnost velikih su razmjera, a njena primjena sve više raste, ne samo u komercijalnom sektoru, već i u obrazovanju. Sve brži tehnološki razvoj svakako olakšava i pojednostavljuje korištenje tehnologije proširene stvarnosti u obrazovanju. Aplikacije koje se razvijaju na osnovi ove tehnologije doista imaju veliki potencijal da pomognu učenicima u usvajanju novog te utvrđivanju već naučenog gradiva.

Ovaj diplomski rad stoga opisuje izradu aplikacije za mobilne uređaje pomoću proširene stvarnosti koja će učenicima omogućiti vizualizaciju i interakciju s modelima s kojima se možda ne bi imali prilike susresti u klasičnoj učionici. U samoj izradi aplikacije koristili smo tehnologije Vuforia, Unity, Blender i Microsoft Visual Studio. Nakon instalacije na mobilne uređaje, kreirana aplikacija učenicima pruža mogućnost da „ožive“ kemijske elemente, proteine, fizikalne pojave, geometrijska tijela, i sl. Tako će aplikacija u nastavi fizike učenicima omogućiti interakciju s animiranim 3D valom u stvarnom vremenu i tako im pomoći da uoče i usvoje što je val i koje su njegove glavne značajke. Prednost ove aplikacije očituje se u tome da učenici imaju priliku promatrati val u stvarnom vremenu i iz bilo koje ravnine i uvidjeti na koji se način val giba u tim ravninama. Osim ovog vizualnog aspekta prisutnog u našoj aplikaciji, proširena stvarnost ima mogućnost dodavanja i zvučnih efekata što može doprinijeti još većoj aktivnosti učenika u nastavi.

Važno je spomenuti da svrha izrade ove aplikacije nije zamijeniti ulogu nastavnika. Naprotiv, ova aplikacija može pomoći učiteljima u motiviranju učenika na učenje i aktivno sudjelovanje u nastavi, ali i u odučavanju učenika od pogrešnih znanja, npr. onih koja se odnose na valove. Kao i svaka druga aplikacija i ova aplikacija ima svojih ograničenja. Jedno od ograničenja s kojim smo se susreli je modeliranje procesa. Procesi moraju što preciznije prikazivati stvarne pojave, a pri tome biti vizualno dovoljno jasni da kod učenika ne stvore dodatna nerazumijevanja. Također, upotreba ovakve aplikacije u nastavi pretpostavlja da svaki učenik posjeduje pametni telefon koji ju može pokrenuti.

Uzimajući u obzir i prednosti i ograničenja ove aplikacije, možemo zaključiti da bi njena primjena u nastavi mogla obogatiti i olakšati proces učenja kako učiteljima tako i učenicima. Daljnji razvoj ovakvih tehnologija zasigurno će u budućnosti omogućiti

još bržu i jednostavniju implementaciju proširene stvarnosti u obrazovanje.

Dodaci

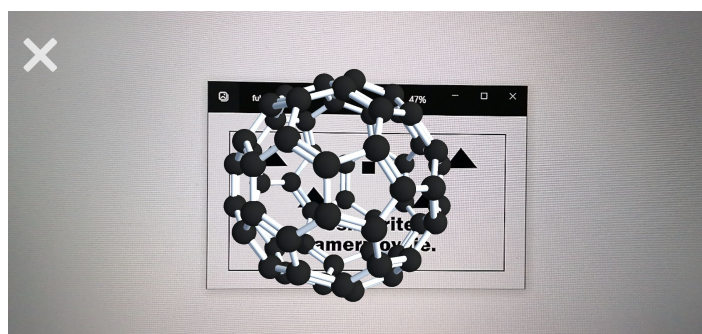
Dodatak A Funkcionalnost aplikacije

Nakon što učenik otvori aplikaciju na pametnom telefonu, prvo što će mu se na zaslonu prikazati jest scena glavnog izbornika. Glavni izbornik aplikacije sadrži dva gumba, gumb *Pokreni* i gumb *Izlaz*. Pritiskom na gumb *Pokreni* učenik ulazi u scenu koja prikazuje ono što vidi kamera pametnog telefona, dok pritiskom na gumb *Izlaz* učenik izlazi iz aplikacija.



Slika A.1: Slika prikazuje scenu glavnog izbornika koji sadrži gumbe *Pokreni* i *Izlaz*.

Nakon što kamera detektira ciljanu sliku, na zaslonu pametnog telefona prikazuje se 3D model koji smo dodijelili toj ciljanoj slici unutar aplikacije. Korisničko sučelje same scene kamere sadrži gumb koji je označen sa X. Pritiskom na gumb X, aplikacija vraća učenika iz scene kamere u glavni izbornik.



Slika A.2: Slika prikazuje scenu kamere koja je detektiranjem ciljane slike fullerena prikazala njegov 3D model.

Dodatak B Popis slika

2.1	Prikaz <i>Sensorame</i> , uređaja koji je patentirao američki filmski snimatelj Morton Heilig [10].	3
3.1	Prikaz Unity sučelja koje sadrži Alatnu traku (označenu zelenom bojom), Hijerarhijski prozor (označen crvenom bojom), Prikaz scene (označen svjetloplavom bojom), Prozor projekta (označen žutom bojom) te Pregledni prozor (označen tamnoplavom bojom).	5
3.2	Prikaz sučelja Microsoft Visual Studia.	7
3.3	Prikaz sučelja Blendera koji sadrži 3D prikaz (označen crvenom bojom), <i>Outliner</i> (označen zelenom bojom), Svojstva (označen plavom bojom) i Vremensku crtu (označenu žutom bojom).	10
4.1	Prikaz Unity Huba prilikom prvog pokretanja.	11
4.2	Prikaz scena koje sadržava aplikacija.	12
4.3	Prikaz baze podataka ciljanih slika.	12
4.4	Prikaz uvoza paketa Vuforia u Unity.	13
4.5	Prikaz kreiranja objekta <i>Image Targets</i>	13
4.6	Prikaz objekta <i>Image Targets</i> preimenovanog u Fuleren.	14
4.7	Prikaz odnosa <i>Parent-Child</i> unutar hijerarhijskog prozora.	14
4.8	Prikaz 3D modela fulerena unutar Blendera.	15
4.9	Prikaz 3D modela fulerena unutar Unityja.	15
4.10	Prikaz 3D modela longitudinalnog vala unutar Unityja.	16
4.11	Prikaz 3D modela transverznog vala unutar Unityja.	16
4.12	Prikaz animiranja vala u Blenderu.	17
4.13	Prikaz izgleda prozora <i>Animator Controller</i>	18
4.14	Prikaz animacije unutar preglednog prozora.	18
4.15	Prikaz modela vala (<i>Val+os</i>) nakon dodavanja koordinatnih osi.	19
4.16	Prikaz <i>Lean Touch</i> komponenata objekta <i>Protein</i> unutar preglednog prozora.	20
4.17	Prikaz odnosa <i>Parent-Child</i> u hijerarhijskom prozoru i izgled glavnog izbornika.	21

4.18 Prikaz skripte <i>GlavniIzbornik</i> i njezinih metoda unutar Microsoft Visual Studia.	22
4.19 Prikaz javnih metoda koje sadrži klasa <i>GlavniIzbornik</i>	23
4.20 Prikaz izbornika <i>Build settings</i> u kojem se obavlja odabir platforme i odabir scena (označeno crvenom bojom).	24
4.21 Prikaz opcija unutar prozora <i>Player settings</i>	24
5.1 Prikaz modela M1 proteina unutar Unityja.	25
5.2 Prikaz modela M1 proteina kojeg prikazuje aplikacija.	25
5.3 Prikaz modela fullerena unutar Unityja.	26
5.4 Prikaz modela fullerena kojeg prikazuje aplikacija.	26
5.5 Prikaz modela ikosaedra unutar Unityja.	27
5.6 Prikaz modela ikosaedra kojeg prikazuje aplikacija.	27
5.7 Prikaz modela longitudinalnog vala unutar Unityja.	28
5.8 Prikaz modela longitudinalnog vala kojeg prikazuje aplikacija.	28
5.9 Prikaz modela transverznog vala unutar Unityja.	29
5.10 Prikaz modela transverznog vala kojeg prikazuje aplikacija.	29
A.1 Slika prikazuje scenu glavnog izbornika koji sadrži gumb <i>Pokreni</i> i <i>Izlaz</i> . 32	
A.2 Slika prikazuje scenu kamere koja je detektiranjem ciljane slike fullere- na prikazala njegov 3D model.	32

Bibliografija

- [1] Jelaš, D. QR kodovi u obrazovanju. Split : Visoka škola za menadžment i dizajn Aspira, 2011. https://cuc.carnet.hr/2011/images/e_3_qr_kodovi389f.pdf?dm_document_id=474&dm_dnl=1, 9.9.2022.
- [2] Maier, P.; Klinker, G.; Tonnis, M. Augmented Reality for teaching spatial relations, 2009. <http://campar-old.in.tum.de/pub/maierp2009ijas/maierp2009ijas.pdf>, 9.9.2022.
- [3] Suprpto, N.; Nandyansah, W.; Mubarak, H. An Evaluation of the “PicsAR” Research Project: An Augmented Reality in Physics Learning. // International Journal of Emerging Technologies in Learning (iJET). Vol. 15 10(2020) <https://www.learntechlib.org/p/217045/>, 9.9.2022.
- [4] Fujisawa, M.; Kato, H. Interactive Fluid Simulation Using Augmented Reality Interface / edited by Shumaker and Randall. Springer : Berlin : Heidelberg, 2009. Str. 431-438. https://link.springer.com/chapter/10.1007/978-3-642-02771-0_48#citeas, 9.9.2022.
- [5] Li, N. ; Gu, Y.; Chang, L.; Duh, H. Influences of AR-Supported Simulation on Learning Effectiveness in Face-to-face Collaborative Learning for Physics. // 11th IEEE International Conference on Advanced Learning Technologies, ICALT 2011. Str. 320-322. https://www.researchgate.net/publication/221424606_Influences_of_AR-Supported_Simulation_on_Learning_Effectiveness_in_Face-to-face_Collaborative_Learning_for_Physics, 10.9.2022.
- [6] Matcha, W.; Awang, R.; Dayang, R. Preliminary Investigation on the Use of Augmented Reality in Collaborative Learning / edited by Abd Manaf; Sahibuddin; Ahmad; Mohd Daud; El-Qawasmeh. Springer : Berlin : Heidelberg, 2011. Str 189-198. https://link.springer.com/chapter/10.1007/978-3-642-25483-3_15, 10.9.2022.
- [7] Azuma, R.; Baillot, Y.; Behringer, R.; Feiner, S.; Julier,S.; MacIntyre, B. Recent advances in augmented reality. // Computer Graphics and Applications.

Vol. 21, 6(2001), str. 34-47., <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=963459>, 20.8.2022.

- [8] Ozdamli, F.; Karagozlu, K. Preschool Teachers' Opinions on the Use of Augmented Reality Application in Preschool Science Education. // Hrvatski časopis za odgoj i obrazovanje. Vol. 20, 1(2018): str. 43-74., <https://hrcak.srce.hr/en/clanak/293889#>, 20.8.2022.
- [9] Pleše, P. Proširena stvarnost. Završni rad. Zagreb : Filozofski fakultet odsjek za informacijske i komunikacijske znanosti, 2019
- [10] Sensorama simulator, <https://patentimages.storage.googleapis.com/e7/49/2e/8113c2ef65cabf/US3050870-drawings-page-3.png/>, 20.8.2022.
- [11] Solak, E.; Cakir, R. Investigating the Role of Augmented Reality Technology in the Language Classroom. // Hrvatski časopis za odgoj i obrazovanje. Vol. 18, 4(2016): str. 1067-1085., <https://hrcak.srce.hr/en/172319#>, 20.8.2022.
- [12] Unity Asset Store, (2022), <https://assetstore.unity.com/>, 12.8.2022.
- [13] Unity, (2022), <https://unity.com/>, 12.8.2022.
- [14] Unity Platform, (2022), <https://unity.com/products/unity-platform/>, 13.8.2022.
- [15] The Project window, (13.8.2022), <https://docs.unity3d.com/Manual/ProjectView.html>, 14.8.2022.
- [16] The Hierarchy window, (13.8.2022), <https://docs.unity3d.com/Manual/Hierarchy.html>, 14.8.2022.
- [17] The Scene view, (13.8.2022), <https://docs.unity3d.com/Manual/UsingTheSceneView.html>, 14.8.2022.
- [18] The Project window, (13.8.2022), <https://docs.unity3d.com/Manual/ProjectView.html>, 14.8.2022.
- [19] The Toolbar, (13.8.2022), <https://docs.unity3d.com/Manual/Toolbar.html>, 14.8.2022.

- [20] Sanjković, V.; Trstenjak, B. Visual Studio - univerzalni alat za razvoj aplikacija // Zbornik radova Međimurskog veleučilišta u Čakovcu. Vol. 2, 1(2011), str. 115-124., <https://hrcak.srce.hr/71334/>, 20.8.2022.
- [21] Microsoft Visual Studio, (2022), <https://visualstudio.microsoft.com/>, 13.8.2022.
- [22] Getting Started with Vuforia Engine in Unity, (2021), <https://library.vuforia.com/getting-started/getting-started-vuforia-engine-unity/>, 15.8.2022.
- [23] Vuforia developer portal, (2022), <https://developer.vuforia.com/downloads/sdk/>, 15.8.2022.
- [24] Image Targets, (2021), <https://library.vuforia.com/objects/image-targets/>, 15.8.2022.
- [25] Best Practices for Designing and Developing Image-Based Targets, (2021), <https://library.vuforia.com/objects/best-practices-designing-and-developing-image-based-targets/>, 15.8.2022.
- [26] About Blender, (2022), <https://www.blender.org/about/>, 16.8.2022.
- [27] Download Blender, (2022), <https://www.blender.org/download/>, 16.8.2022.
- [28] 3D Viewport, (2.5.2022), <https://docs.blender.org/manual/en/3.1/editors/3dview/index.html>, 16.8.2022.
- [29] Timeline, (19.8.2022), <https://docs.blender.org/manual/en/latest/editors/timeline.htm>, 20.8.2022.
- [30] Outliner, (2.5.2022), <https://docs.blender.org/manual/en/3.1/editors/outliner/index.html>, 20.8.2022.
- [31] Properties, (19.8.2022), https://docs.blender.org/manual/en/latest/editors/properties_editor.html, 20.8.2022.
- [32] Animator component, (13.8.2022), <https://docs.unity3d.com/Manual/class-Animator.html>, 14.8.2022.

- [33] Animator Controller, (13.8.2022), <https://docs.unity3d.com/Manual/class-AnimatorController.html>, 14.8.2022.
- [34] Importing a model with humanoid animations, (13.8.2022), <https://docs.unity3d.com/Manual/ConfiguringtheAvatar.html>, 14.8.2022.
- [35] Lean Touch, (4.3.2022), <https://assetstore.unity.com/packages/tools/input-management/lean-touch-30111>, 14.8.2022.
- [36] Canvas, (6.10.2020), <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/UICanvas.html>, 16.8.2022.
- [37] Image, (6.10.2020), <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/script-Image.html>, 17.8.2022.
- [38] Sprites, (24.2.2021), <https://docs.unity3d.com/2020.1/Documentation/Manual/Sprites.html>, 17.8.2022.
- [39] IMGUI Controls, (13.8.2022), <https://docs.unity3d.com/Manual/gui-Controls.html>, 17.8.2022.
- [40] Overview of classes, structs, and records in C#, (12.8.2022), <https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/object-oriented>, 17.8.2022.
- [41] Build Settings, (13.8.2022), <https://docs.unity3d.com/Manual/BuildSettings.html>, 20.8.2022.
- [42] PlayerSettings, (13.8.2022), <https://docs.unity3d.com/ScriptReference/PlayerSettings.html>, 20.8.2022.
- [43] Hilsch, M.; Goldenbogen, B.; Sieben, C.; Höfer, C. T.; Rabe, J. P.; Klipp, E.; Herrmann, A.; Chiantia, S. Influenza A matrix protein M1 multimerizes upon binding to lipid membranes // *Biophysical journal*. Vol. 107, 4(2014), str. 912–923, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4142230/>, 20.8.2022.
- [44] About RCSB PDB: Enabling Breakthroughs in Scientific and Biomedical Research and Education, (2020), <https://www.rcsb.org/pages/about-us/index>, 20.8.2022.

[45] Kroto, H. W.; Walton, D. R. M. Fullerene // Encyclopedia Britannica.
(13.3.2019.) <https://www.britannica.com/science/fullerene>, 12.8.2022.