**SVEUČILIŠTE U SPLITU**

**FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE**

**TENISKA IGRA U UNITY-JU**

**SEMINARSKI RAD**

Kolegij: **Računalna grafika**

Mentor: **Vladan Papić**

Studenti: **Mia Gudić, Petra Vujević**

Split, siječanj 2021.

Sadržaj:

[1 UNITY POGON ZA IGRE 1](#_Toc61214684)

[1.1 Pregled Unity-ja 1](#_Toc61214685)

[1.2 Pregled korisničkog sučelja Unity softvera 1](#_Toc61214686)

[1.3 Kodiranje u Unity-ju 3](#_Toc61214687)

[1.4 Unity u kontekstu seminarskog rada 3](#_Toc61214688)

[2 ANIMACIJA TENISKE IGRE 4](#_Toc61214689)

[2.1 Opis animacije 4](#_Toc61214690)

[2.2 Razvoj animacije u Unity-ju 4](#_Toc61214691)

[2.3 Postavljanje scene 5](#_Toc61214692)

[2.4 Programiranje putanje teniske loptice 11](#_Toc61214693)

[2.4.1 Skripta *Player* 13](#_Toc61214694)

[*2.4.2* Skripta *Ball* 15](#_Toc61214695)

[2.4.3 Skripta *Bot* 16](#_Toc61214696)

[3 ZAKLJUČAK 18](#_Toc61214697)

[4 LITERATURA 19](#_Toc61214698)

[5 SAŽETAK 20](#_Toc61214699)

# UNITY POGON ZA IGRE

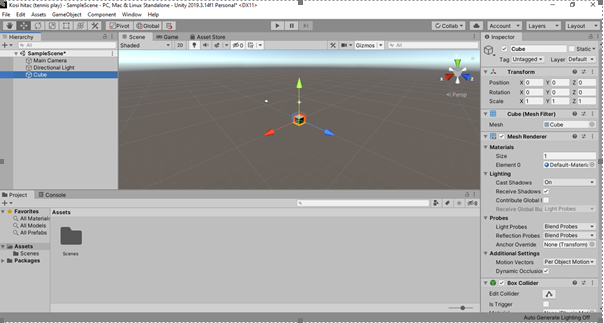
Unity pogon za igre (engl. *Unity game engine*) je svjetska platforma za stvaranje i rad interaktivnih, 3D sadržaja u stvarnom vremenu, pružajući alate za brzu i učinkovitu izradu igara koje se mogu objaviti na širokom rasponu uređaja. Može se koristiti za stvaranje 3D i 2D virtualne stvarnosti i igara proširene stvarnosti, kao i simulacija i drugih doživljaja. U Unity se mogu importirati 2D i 3D resursi napravljeni na nekom drugom softveru te se od njih mogu stvoriti okruženja i prizori dodavanjem rasvjete, zvuka, posebnih efekata itd. Također je moguće animirati importirane objekte. Osim što se najviše koristi za izradu video igara, Unity pogon za igre usvojile su razne industrije kao što su filmska, automobilska, arhitekturalna, inženjerska i slično.

## Pregled Unity-ja

Unity je pogon na više platformi te je podržan na Windows-u i macOS-u, a inačica uređivača je dostupna i za Linux platformu. 2018. godine pogon je proširen tako da podržava izradu igara na više od 25 različitih platformi. Prvih deset godina postojanja Unity softver morao je biti kupljen u potpunosti, dok trenutno postoje dvije opcije licenciranja. Besplatna licenca namijenjena je osobnoj upotrebi ili malim tvrtkama, a pretplate ovise o prihodima koje ostvaruju igre koristeći Unity. Objavljeno je nekoliko glavnih verzija Unity-ja od njegovog pokretanja. Posljednja stabilna verzija, 2020.1.4, objavljena je u rujnu 2020. godine.

## Pregled korisničkog sučelja Unity softvera

Glavni prozor Unity uređivača sačinjen je od prozora s karticama koje se mogu preurediti i grupirati. Stoga se izgled uređivača može razlikovati ovisno o osobnoj sklonosti programera i vrsti posla kojeg obavlja. Zadani raspored prozora omogućuje praktičan pristup najviše korištenim prozorima (Slika 1).



Slika 1 Prikaz korisničkog sučelja Unity softera

Unity korisničko sučelje sastoji se od sljedećih elemenata:

1. Alatna traka omogućuje pristup najvažnijim radnim značajkama. Na lijevoj strani sadrži osnovne alate za upravljanje prikazom scene i objektima unutar nje. U središtu su kontrole reproduciranja, pauziranja i koraka. Desno se nalaze gumbi koji omogućuju pristup Unity Collaborate, *Cloud*-u (za pristup Unity uslugama) i korisničkom Unity računu. U desnom kutu se još nalaze *Layers* za kontrolu objekata na slojevima i *Layout* koji pruža alternativne izglede za prozore uređivača i omogućuje spremanje vlastitih prilagođenih izgleda.
2. Hijerarhijski prozor prikazuje listu svih objekata u trenutnoj sceni hijerarhijskim redoslijedom. Neki od ovih su izravne instance iz *Asset* datoteke, dok su druge instance *Prefabs* (unaprijed prilagođeni objekti koji sačinjavaju većinu scene). Hijerarhija prikazuje strukturu kako su objekti pričvršćeni jedni na druge.
3. U Projekt prozoru može se pristupiti i upravljati svime što pripada vašem projektu. Sadržava modele, materijale, teksture, skripte itd.
4. Prozor Inspektora prikazuje detaljne informacije o trenutno odabranom objektu, što uključuje i sve komponente dodane od strane korisnika, njihova svojstva i mogućnost promjene funkcionalnosti objekta u sceni.
5. Scenski prikaz je interaktivni pogled na svijet koji korisnik stvara. Scenski prikaz upotrebljava se za odabir i pozicioniranje prizora, likova, kamera, svjetla i objekata potrebnih za konstruiranje željene scene.
6. Prikaz igre prikazan je iz kamera u vašoj sceni. Reprezentativan je za konačnu, objavljenu aplikaciju. Služi za pregled igre tijekom procesa razvoja.

## Kodiranje u Unity-ju

Jezik koji se koristi u Unity-ju je C#. Svi jezici s kojima Unity djeluje su objektno orijentirani skriptni jezici. Unity omogućuje primarni API za skriptiranje u C# jeziku, kako za Unity uređivač u obliku dodataka, tako i za same igre, kao i funkciju povlačenja i ispuštanja, *drag* i *drop.* Skripte određuju ponašanje objekata i komponenti pričvršćenih na objekte. Skriptiranje u Unity-ju se razlikuje od čistog programiranja zato što nije potrebno stvarati kod koji pokreće aplikaciju jer Unity to radi umjesto korisnika/programera. Unity čita sve podatke koji su u sceni igre. Unity se usmjerava uputama koje pišu u skriptama te ih izvršava okvir za okvirom što brže može. Skripte su napisane posebnim jezikom koji Unity može razumjeti. Verzije Unity-ja do 2017.3, imaju uređivač teksta pod nazivom MonoDevelop koji pomaže s dovršavanjem koda, dojavom pogrješki unutar koda te omogućuje korištenje prečica. Od izdanja verzije 2018.1, mogu se koristiti Visual Studio for Unity Community ili drugi uređivači teksta kao što su Visual Studio, Notepad ili Sublime text.

## Unity u kontekstu seminarskog rada

Unity je u ovome seminarskom radu okolina u kojoj će se napraviti i prikazati animacija teniske igre.

Služi za:

* Postavljanje scene u kojoj će se prikazati animacija teniske igre što je u ovom slučaju teniski teren.
* Dodavanje događaja (eng. *Event*) na objekte
* Prikaz putanje teniske loptice kojom se dobacuju dva igrača

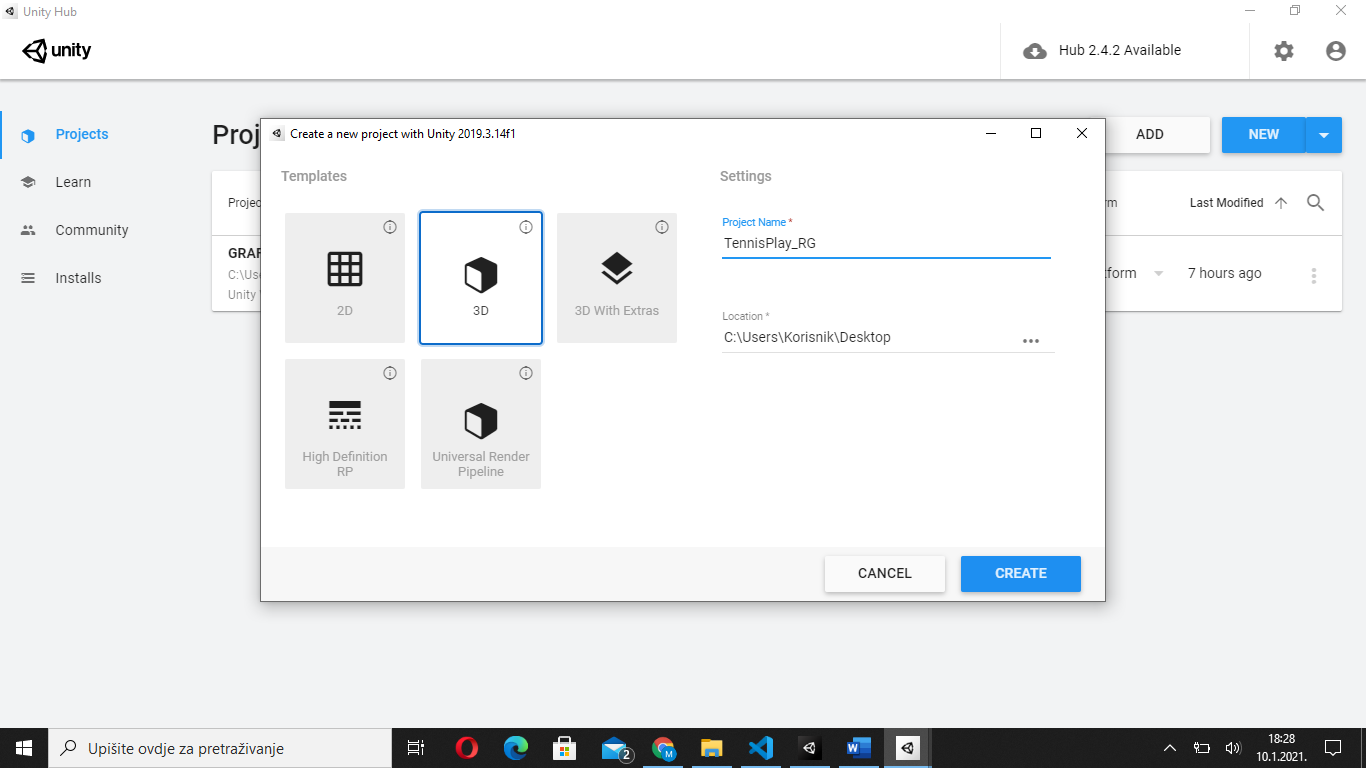
# ANIMACIJA TENISKE IGRE

## Opis animacije

Animacijom će se predstaviti teniska igra. Početna pozicija igre izgleda tako da se na teniskom terenu nalaze dva igrača na suprotnim stranama terena i teniska loptica na određenom položaju ispred igrača kojim upravlja korisnik. Igrač se kreće na način da korisnik njime upravlja tipkama gore, dolje, lijevo i desno. Igra započinje kada igrač dotakne lopticu i time je usmjerava prema protivničkom igraču. Mjesto na koje će pasti loptica određeno je metom koja se može pomicati lijevo ili desno pritiskom na tipku F. Brzina udarca loptice može se povećati pritiskom tipke E. Ispred igrača kojim upravlja korisnik nalaze se tri mete prema kojima, slučajnim odabirom, protivnički igrač upućuje lopticu natrag. Ukoliko jedan od igrača napravi pogrešku, loptica se automatski vraća na početnu poziciju.

## Razvoj animacije u Unity-ju

Sami početak izrade kreće otvaranjem programa Unity. Odabire se opcija *New* za stvaranje novog projekta te se odabire opcija 3D projekta, upisuje se naziv projekta po želji i odabire se konačna lokacija projekta na računalu. Na tom mjestu se kreira mapa koja će sadržavati sve potrebne informacije i dodatke projektu što uključuje scene, C# skripte, materijale, imporirane objekte iz drugih programa i slično (Slika 2).



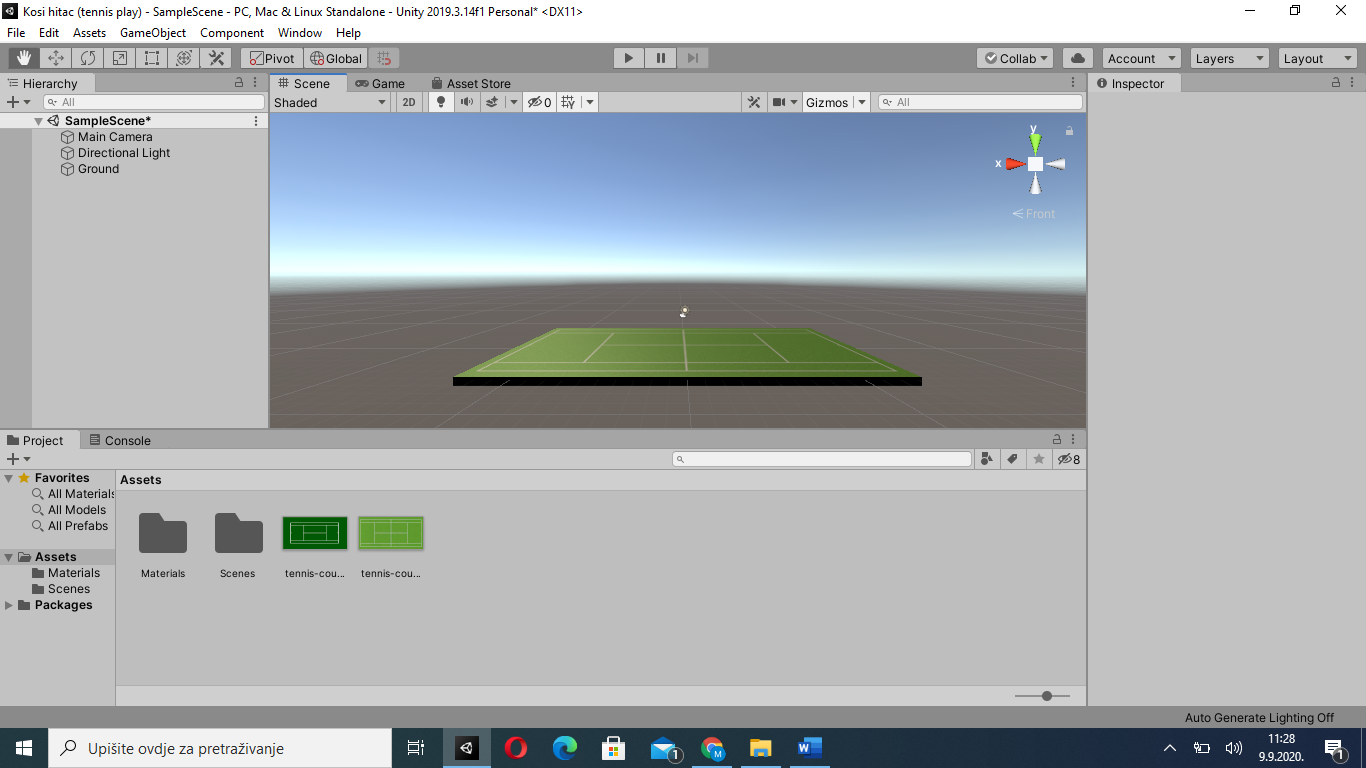
Slika 2 Prikaz kreiranja novog projekta "TennisPlay\_RG“

Nakon što je projekt kreiran, u Projekt prozoru se stvaraju četiri mape s nazivima *Materials*, *Prefabs*, *Scenes*, *Scripts*. Svi modeli koji se koriste nalaziti će se u mapi *Materials*, kod koji će biti korišten za dodavanje događaja na objekte biti će spremljen u mapi *Scripts*, sve scene spremati će se u mapu *Scenes*, a svi objekti koji se naprave, a biti će korišteni uzastopno, nalaziti će se u mapi *Prefabs*. Sljedeći korak biti će postavljanje scene.

## Postavljanje scene

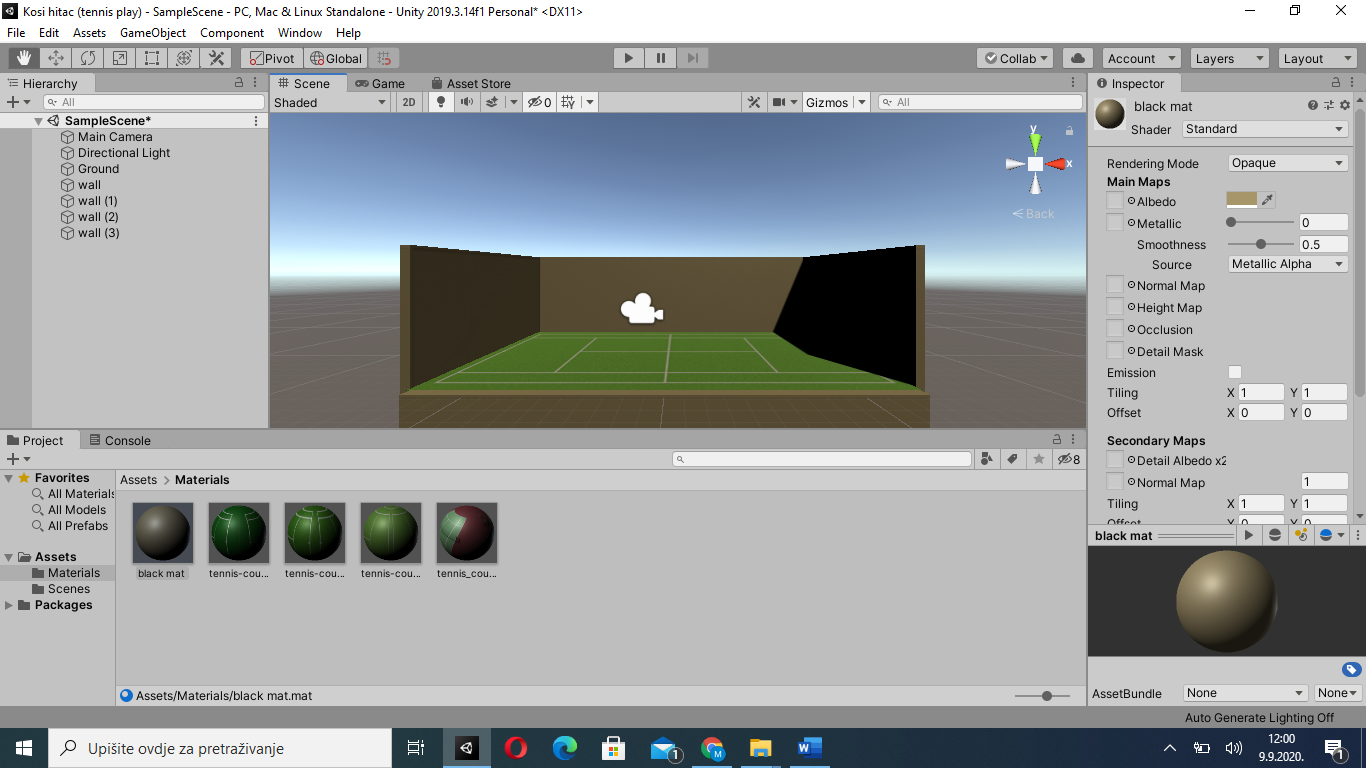
Nakon otvaranja projekta, u hijerarhijskom prozoru, moguće je pronaći već generirane objekte rasvjete i kamere (*Main* *Camera* i *Directional* *Light*) koji se koriste za rasvjetu scene i prikaz.

Kao gotovi produkt cilj je postići realističan prikaz teniskog terena što uključuje kreiranje podloge i modela (zidovi, mreža, rasvjeta, igrači, mete i loptica). Podloga se generira tako što se postavi kursor na hijerarhijski prozor i desnim klikom generira se 3D objekt *Cube*. Veličina objekta se prilagođava na x, y i z osi. Nakon dobivene željene veličine podloge u mapu *Assets* je potrebno uvesti datoteku .jpg formata koja predstavlja teniski teren. Uvoz imovine koja se dodaje objektima se postiže desnim klikom u prozoru *Project* i odabirom „*Import* *New* *Asset*“ gumba. Nakon uvoza, za dodavanje imovine na objekt, koriste se *drag* i *drop* funkcije nakon čega je postignut željeni izgled podloge (Slika 3).



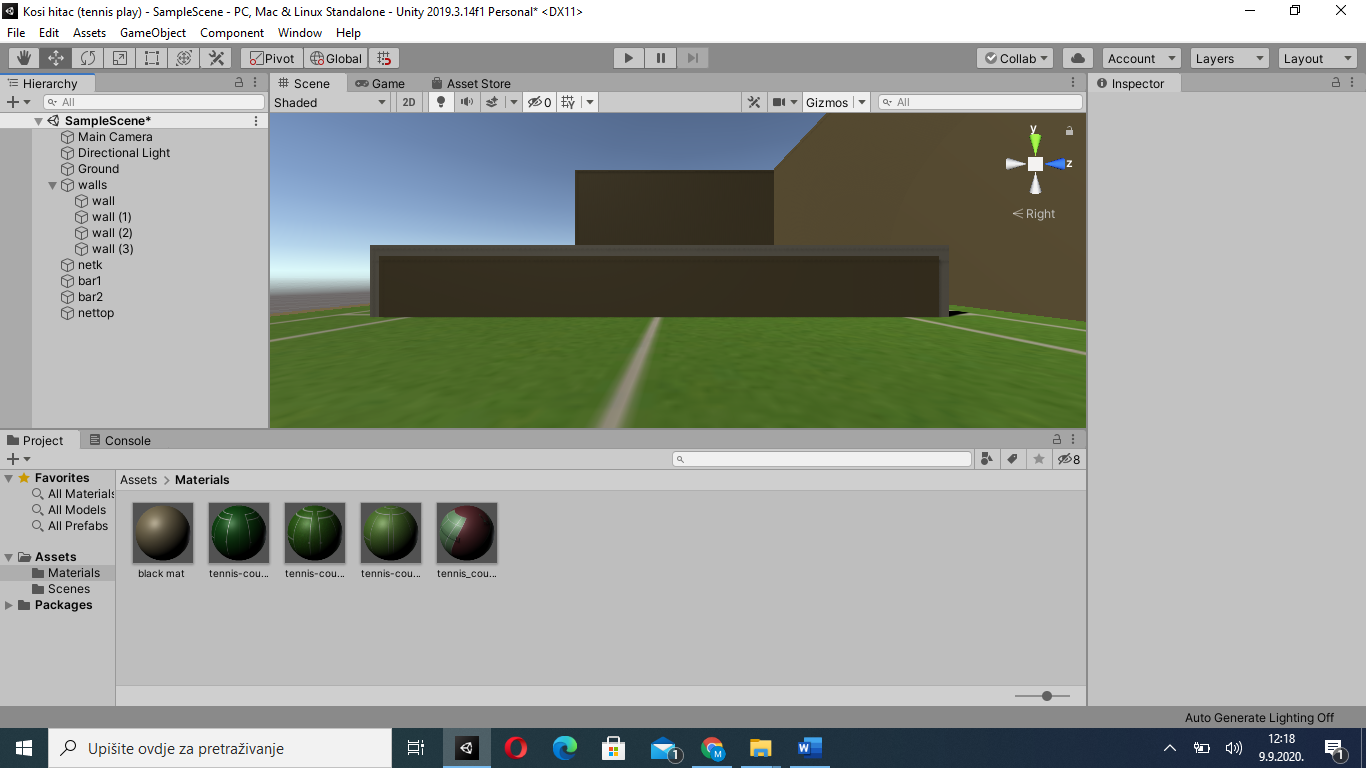
Slika 3 Prikaz podloge

Sada je kreirana podloga na koju je moguće postaviti sve modele. Idući korak u postavljanju scene je kreiranje zidova. Izrada zidova se obavlja na sličan način kao i izrada podloge. Dodaje se novi 3D objekt te se pomoću navigacijskih i rotacijskih alata za pomicanje modela u prostoru dobije željeni izgled zida. Kreira se novi materijal ovisno o želji izgleda objekta i funkcijama *drag* and *drop* se doda objektu. Na taj način je kreiran jedan zid, a cilj je postavljanje zida sa svake strane podloge što se obavlja dupliciranjem napravljenog objekta i opcijama *move*, *rotate* i *scale* se postiže željeni izgled i postavlja se na željenu poziciju (Slika 4). Zid na koji gleda kamera spušta se kako bi scena bila vidljiva. Zbog bolje preglednosti, u prozoru Hierarchy se formira prazni 3D objekt kojem se daje naziv „*Walls*“ te se u njega spremaju svi objekti zidova.



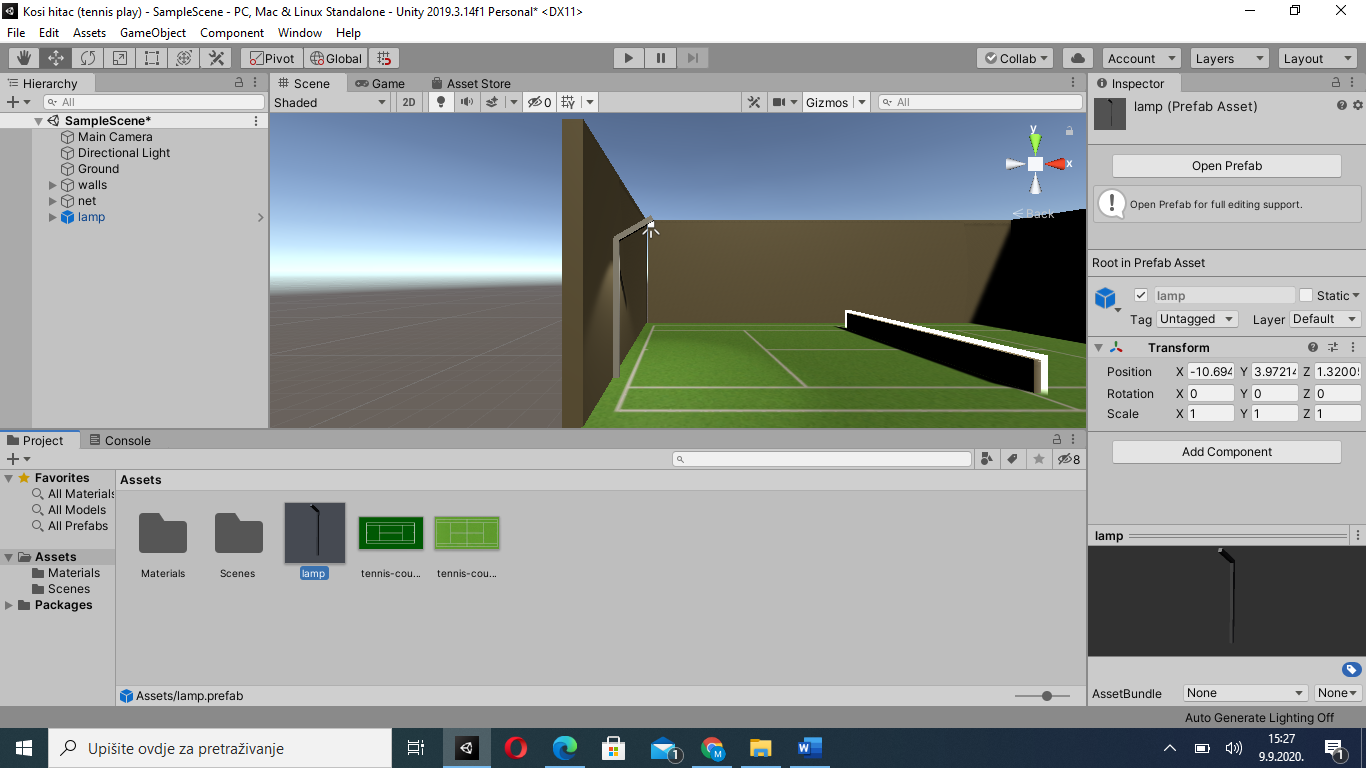
Slika 4 Prikaz scene nakon dodavanja modela zidova

Nakon što su zidovi i tereni dodani potrebno je dodati tenisku mrežu. Dodaju se tri 3D objekta *Cube* od kojih će se napraviti okvir mreže promjenom veličina tih objekata po x, y i z osi. Definira se novi materijal, bijele boje, koji će biti dodan okviru mreže, na isti način kao i do sada, funkcijama *drag* i *drop*. Zatim se kreira još jedan 3D objekt koji će biti sama mreža i njemu je potrebno dodati materijal crne boje (Slika 5). Zbog preglednosti, ali i kako bi se cijelu mrežu s okvirom moglo označiti kao cjelinu, kreira se novi prazni 3D objekt kojem se daje naziv *net* i u njega se spremaju svi objekti mreže.

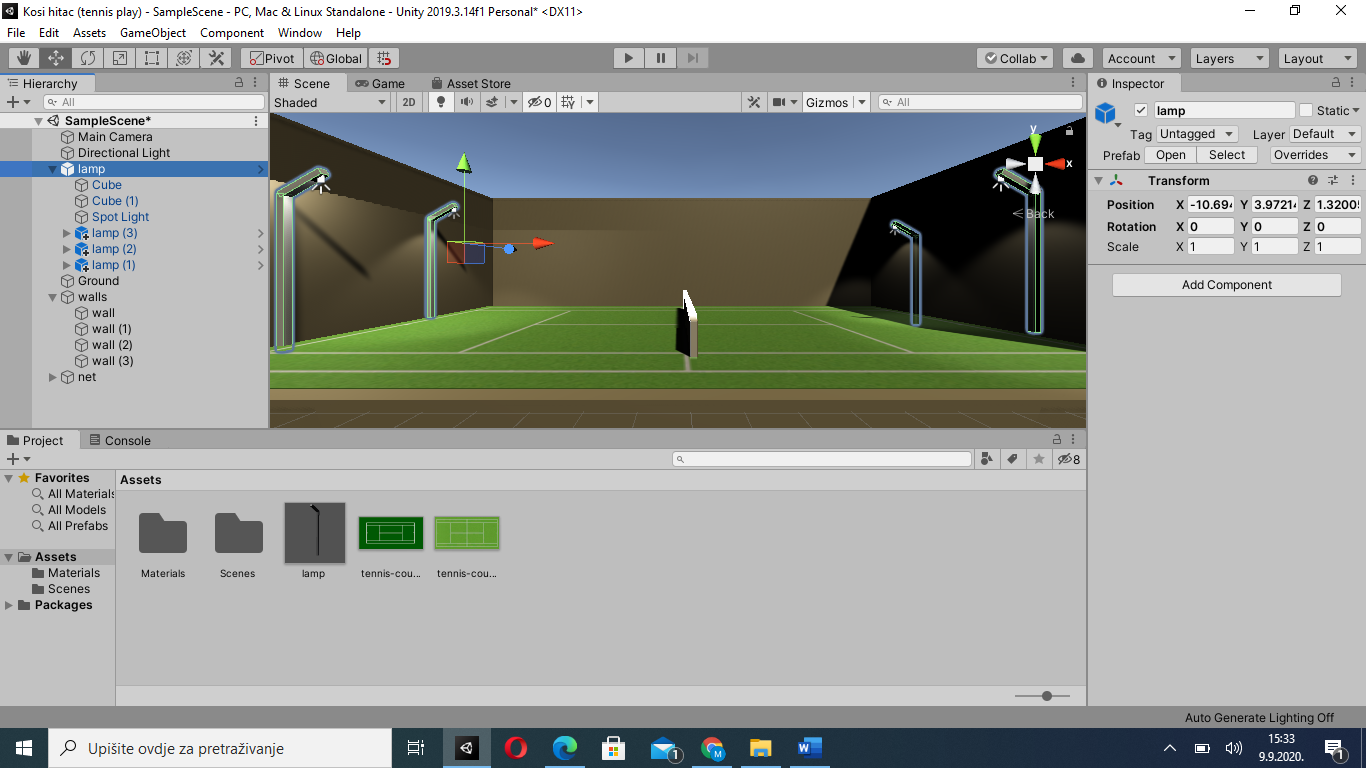


Slika 5 Prikaz scene nakon dodavanja modela mreže

Sljedeće što ćemo dodati sceni su svjetiljke radi bolje rasvjete scene. Želimo ih postaviti naslonjene na zid. Izradu počinjemo dodavanjem novog 3D objekta *Cube*. Korištenjem opcije *Scale* izdužimo objekt koji smo dodali, koji je u obliku kocke, te na taj način napravimo stalak za svjetiljku. Dupliciramo dodani objekt i smanjimo ga kako bismo kreirali „glavu“ svjetiljke koja će stajati u kosom položaju naslonjena na stalak. Svjetlost dodajmo odabirom opcije „*Light*“ i zatim odabirom kategorije „*Spotlight*“ što obavljamo desnim klikom u prozoru *Hierarchy*. Sada smo formirali novi objekt Spotlight koji pozicioniramo ispod „glave“ svjetiljke kako bismo dodali svjetlost te u prozoru *Inspector* povećamo domet i opseg svjetla opcijama *Range* i *Spot* *Angle*. Izgled svjetiljke prikazan je na (Slika 6). Formiramo novi prazni objekt te u njega dodamo sve objekte svjetiljke. Cijeli objekt svjetiljke funkcijama *drag* i *drop* dovedemo u prozor *Project* kako bismo od njega napravili *Prefab* da se jednostavno može umetati nove svjetiljke u scenu kao što je prikazano na slici (Slika 7).



Slika 6 Izgled svjetiljke



Slika 7 Prikaz scene nakon dodavanja svih svjetiljki funkcijama drag i drop

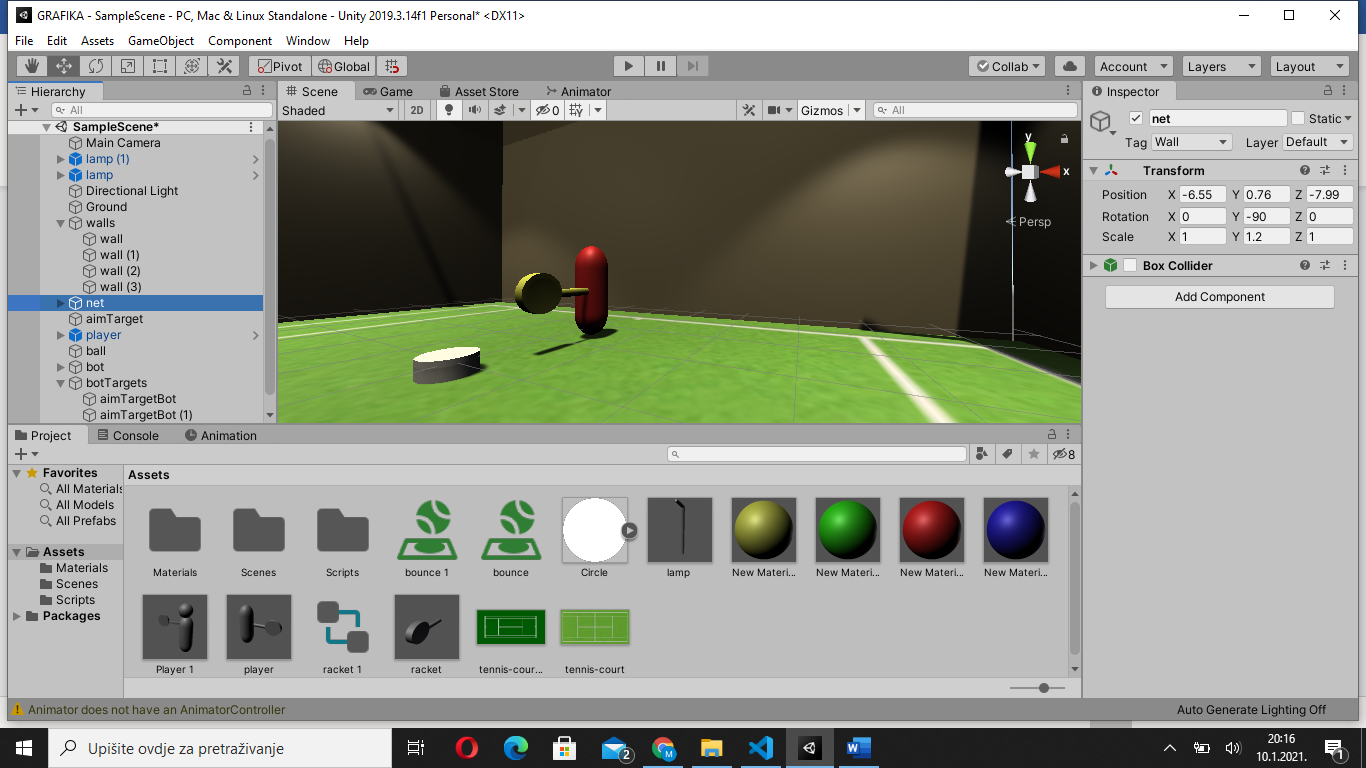
Preostaje još dodati glavne sudionike animacije, igrače i tenisku lopticu.

Za izradu loptice dodaje se novi 3D objekt *Sphere*. Korigira se veličina i pozicija i dodaje potrebni novi materijal da izgled bude što sličniji teniskoj loptici.

Za izradu igrača dodaje se novi 3D objekt *Cylinder*. Korigira se veličina i pozicija i dodaje potrebni novi materijal. Igraču se dodaje teniski reket koji se izradi dodavanjem 3D objekata *Cylinder* te se učini djetetom objekta igrač.

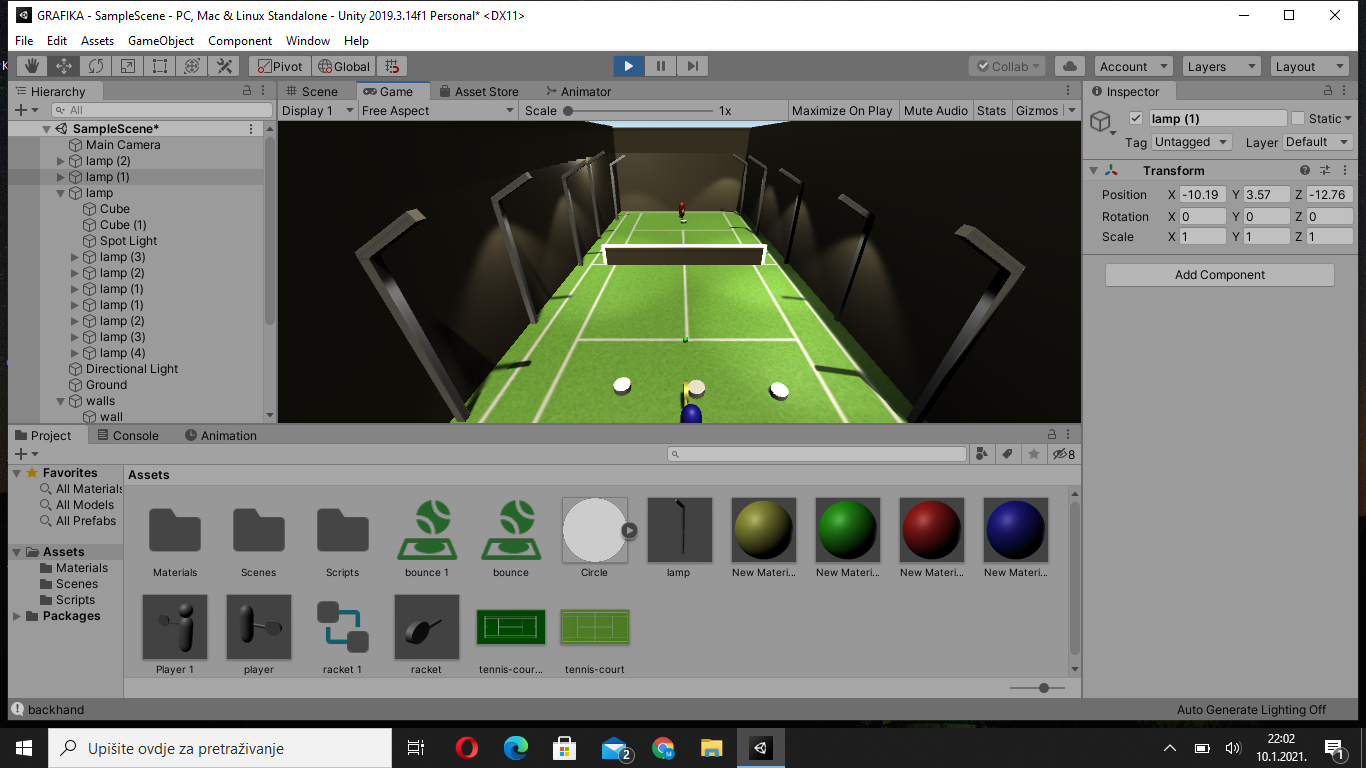


Slika 8 Igrač kojim upravlja korisnik i teniska loptica



Slika 9 Protivnički igrač i meta

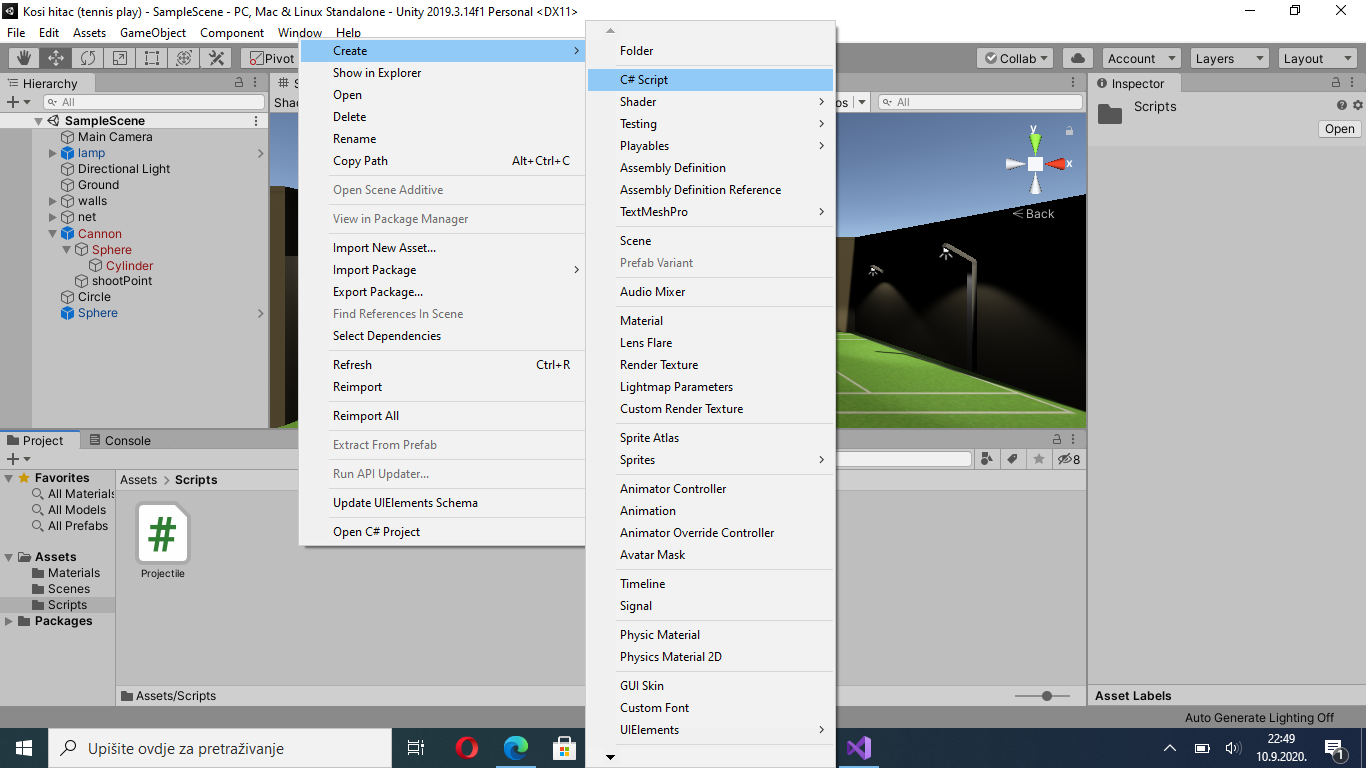
Postavljanje scene time je završilo te još preostaje programirati ponašanje ključnih dijelova animacije: igrača, teniske loptice i meta. Konačna scena prikazana je na slici (Slika 10).



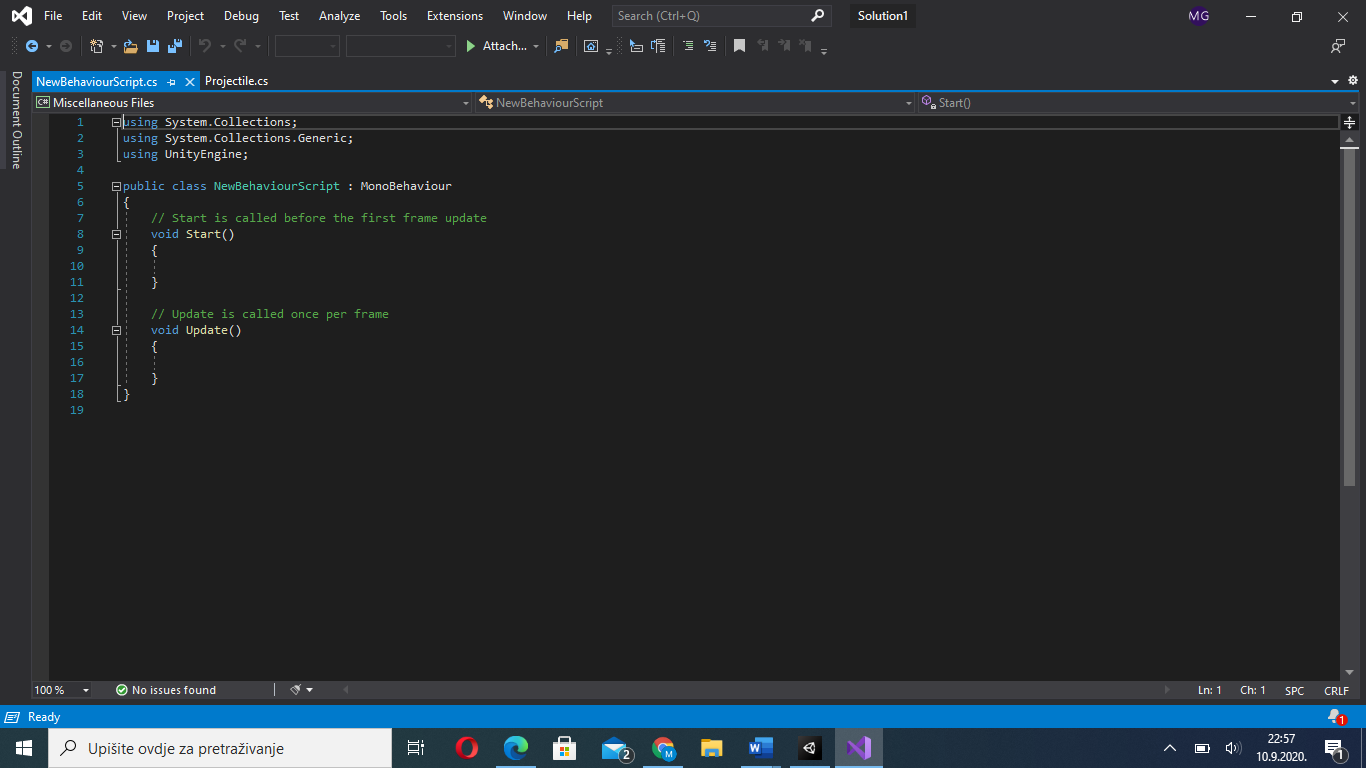
Slika 10 Prikaz konačne scene

## Programiranje putanje teniske loptice

Za dodavanje akcije na jedan od objekata potrebno je napisati C# skriptu koja se doda na objekt funkcijama *drag* i *drop*. Nova skripta dodaje se u mapu *Scripts* u prozoru *Project* desnim klikom i odabirom polja *Create*, a zatim se odabire C# *Script* kao što jeprikazano na slici (Slika 11) . U mapi Scripts se stvara ikona koja označava skriptu. Skriptu se preimenuje po želji. Duplim klikom na nju otvara se datoteka u programu Microsoft Visual Studio ili nekom drugom editoru (Slika 12).



Slika 11 Stvaranje nove skripte



Slika 12 Izgled novostvorene skripte

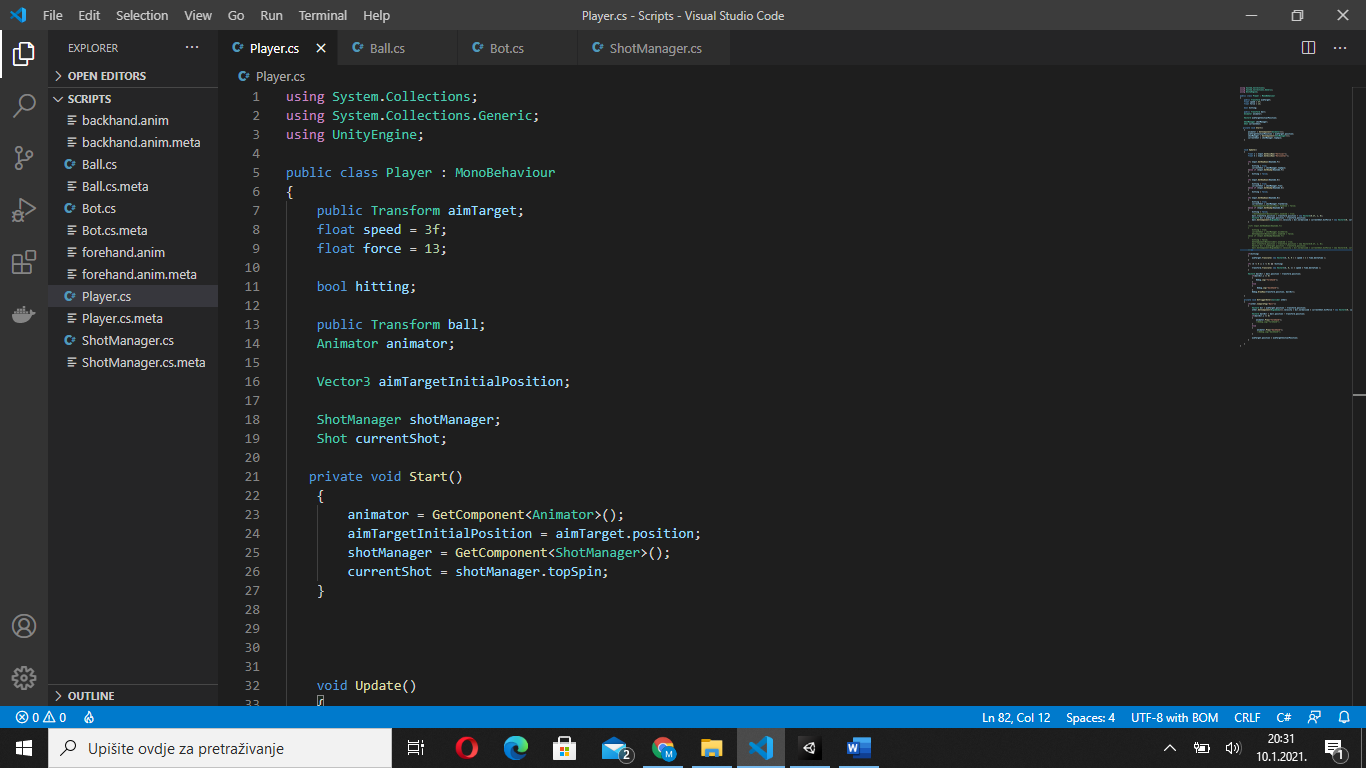
Za simulaciju teniske igre potrebno je napisati tri skripte, *Player*, *Ball* i *Bot* kako bismo dodali kretnje na lopticu i oba igrača.

### Skripta *Player*

Pisanjem skripte *Player* programira se ponašanje korisnikovog igrača. Pisanje skripte počinje dodavanjem varijabli. Dodaju se javne varijable *aimTarget* i *ball*, varijable brzine kretanja i jakosti udaranja loptice koje su tipa *float*, varijabla *hitting* tipa *bool*, koja koja je istinita ako je igrač dotaknuo lopticu, a lažna ako nije te definiramo varijablu *aimTargetInitialPosition* koja je tipa *Vector 3*, a odnosi se na poziciju mete koja određuje smjer kretanja loptice.

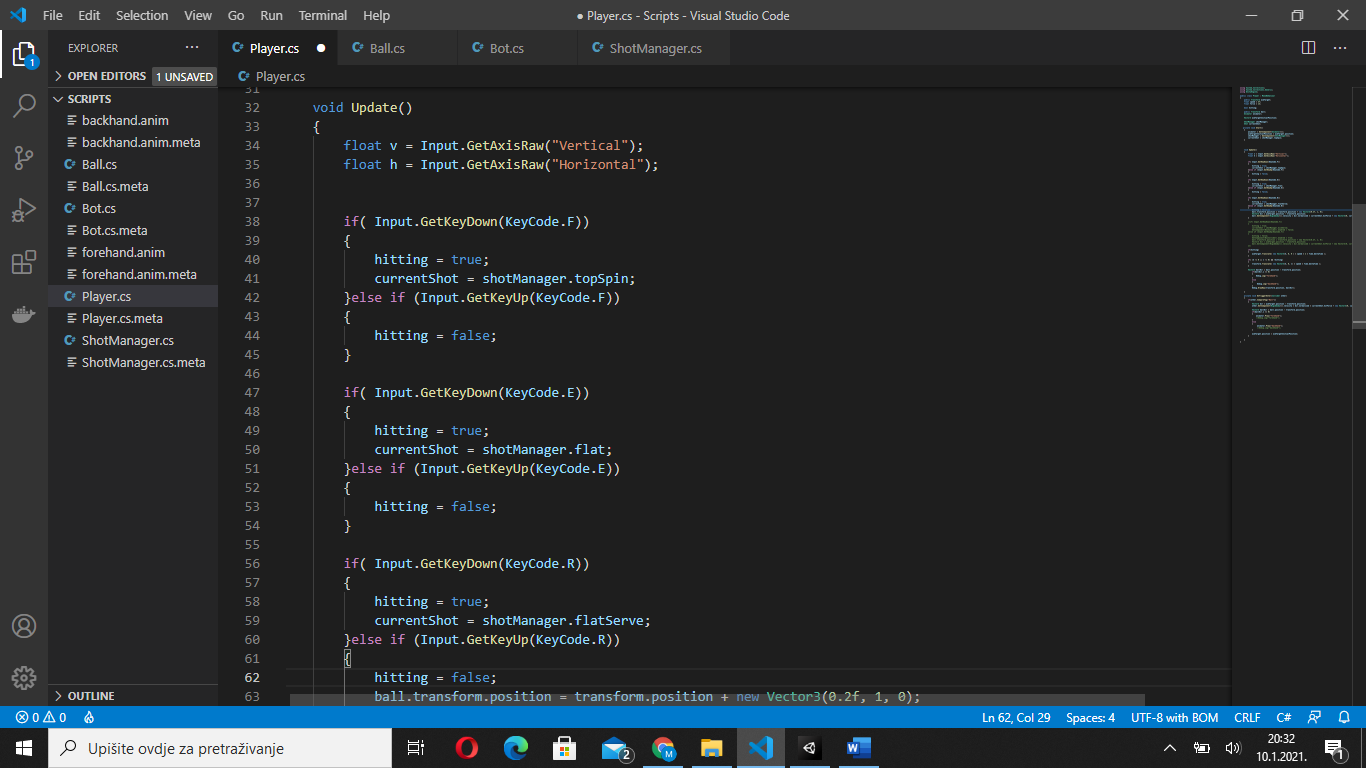
*Vector3* koristi se za prikaz 3D vektora i točaka. Ova se struktura koristi u cijelom Unityju za prolaženje 3D položaja i uputa. Sadrži i funkcije za obavljanje uobičajenih vektorskih operacija.

Unutar skripte napisane su tri funkcije: *Start*, *Update* i *OnTriggerEnter*. Dodane varijable i funkcija *Start* prikazani su na (Slika 13).

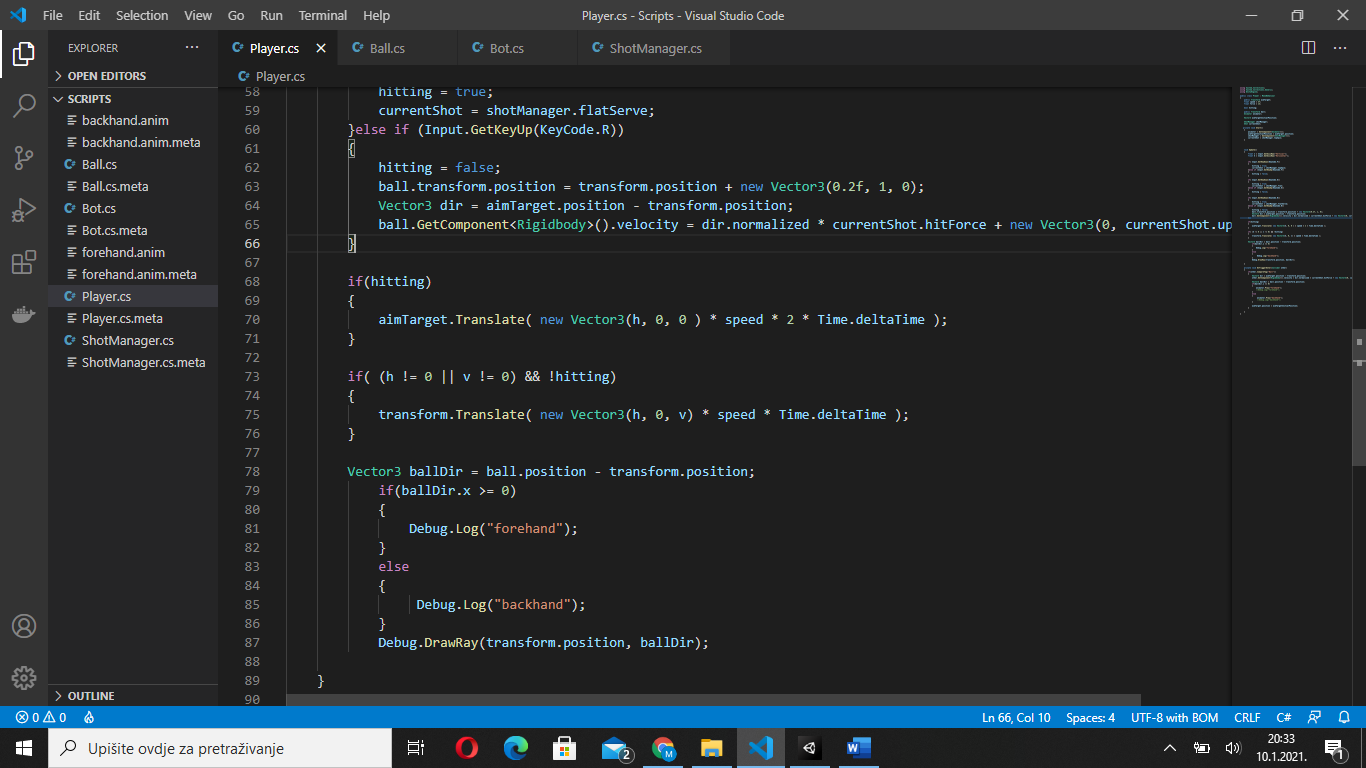


Slika 13 varijable i funkcija Start

Unutar funkcije *Update* deklarirane su dvije varijable h i v tipa *float* koje označavaju vertikalno i horizontalno kretanje korisnikovog igrača. Također, definirano je i ponašanje igrača pritiskom tipki F i E. Pritiskom tipke F pomiče se meta, u horizontalnom smijeru, koja se nalazi ispred protivničkog igrača te označava smjer udarca loptice. Pritiskom tipke E, prilikom udarca loptice, povećava se jačina udarca. Funkcija *Update* prikazana je na (Slika 14 i Slika 15)

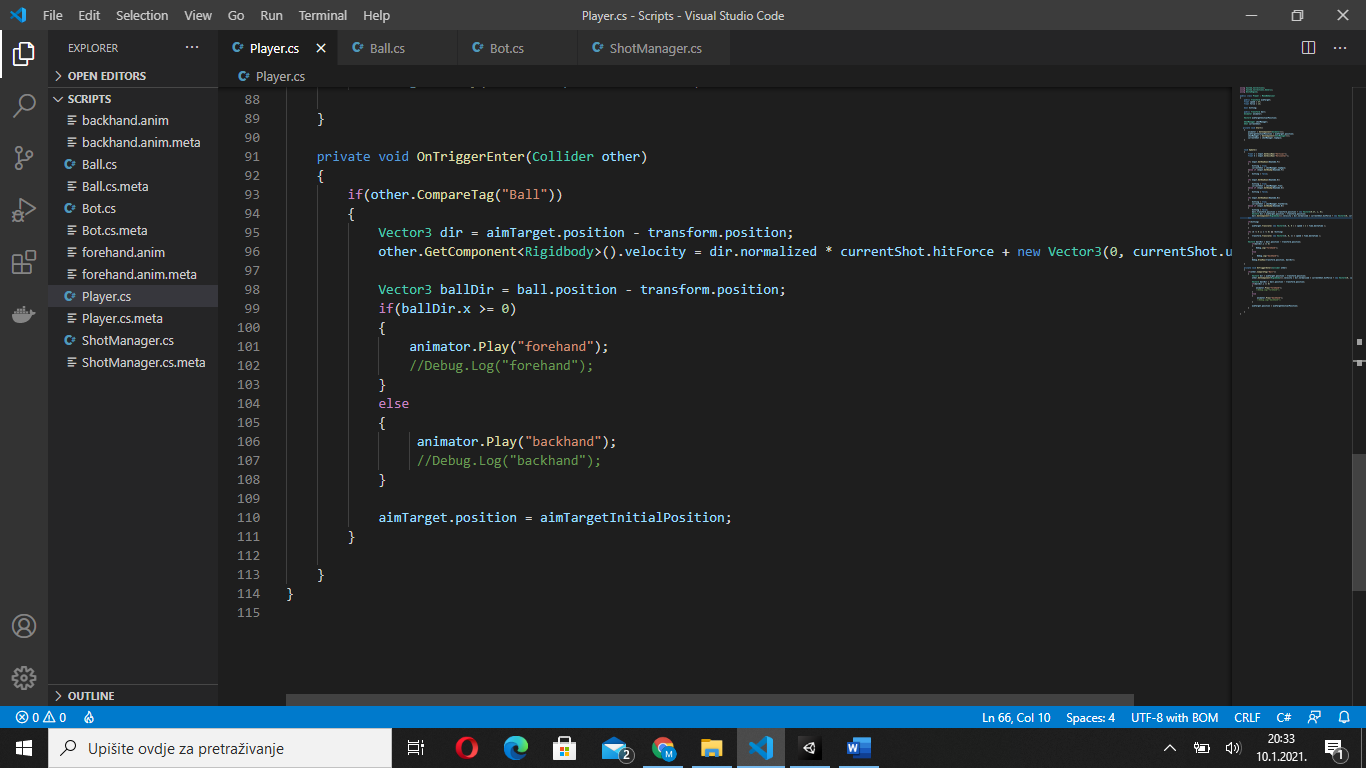


Slika 14 Funkcija Update



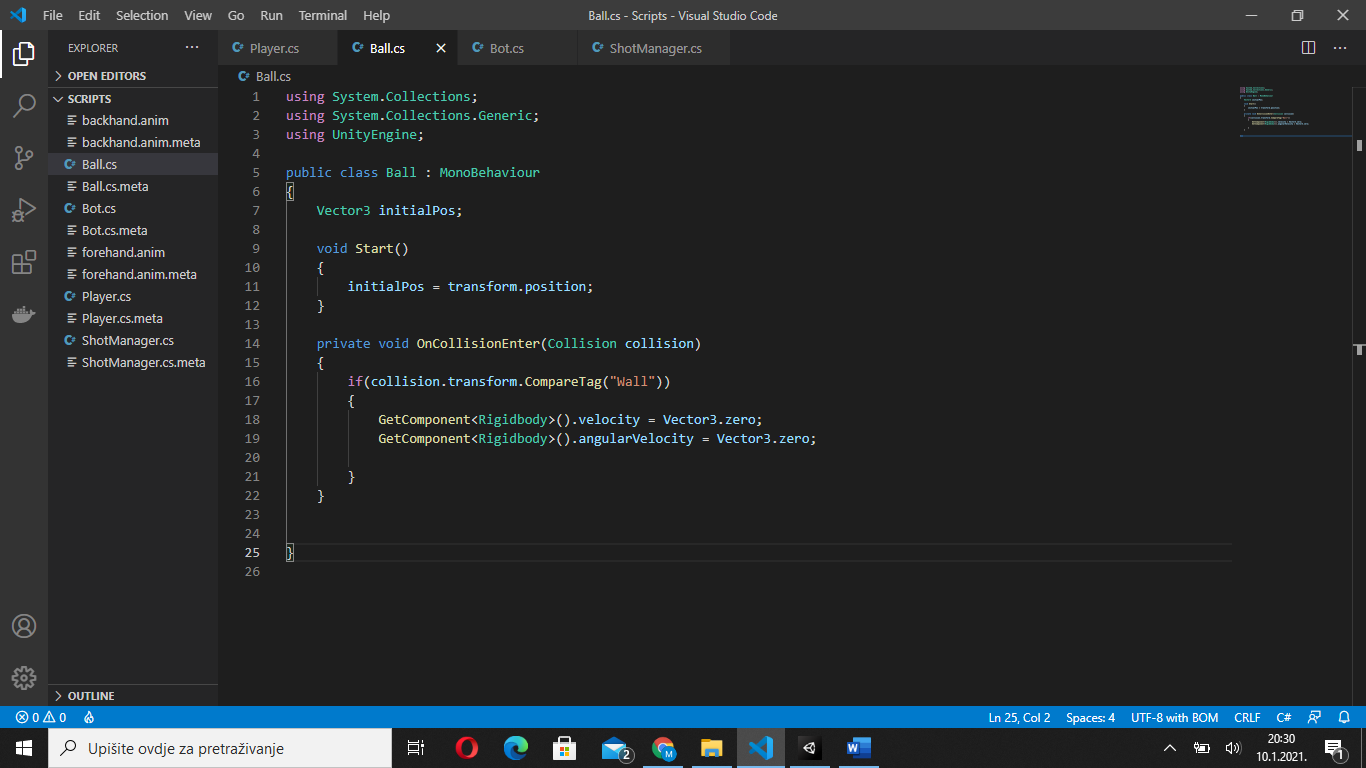
Slika 15 Funkcija Update

Funkcija OnTriggerEnter prikazana je na (Slika 16)



Slika 16 Funkcija OnTriggerEnter

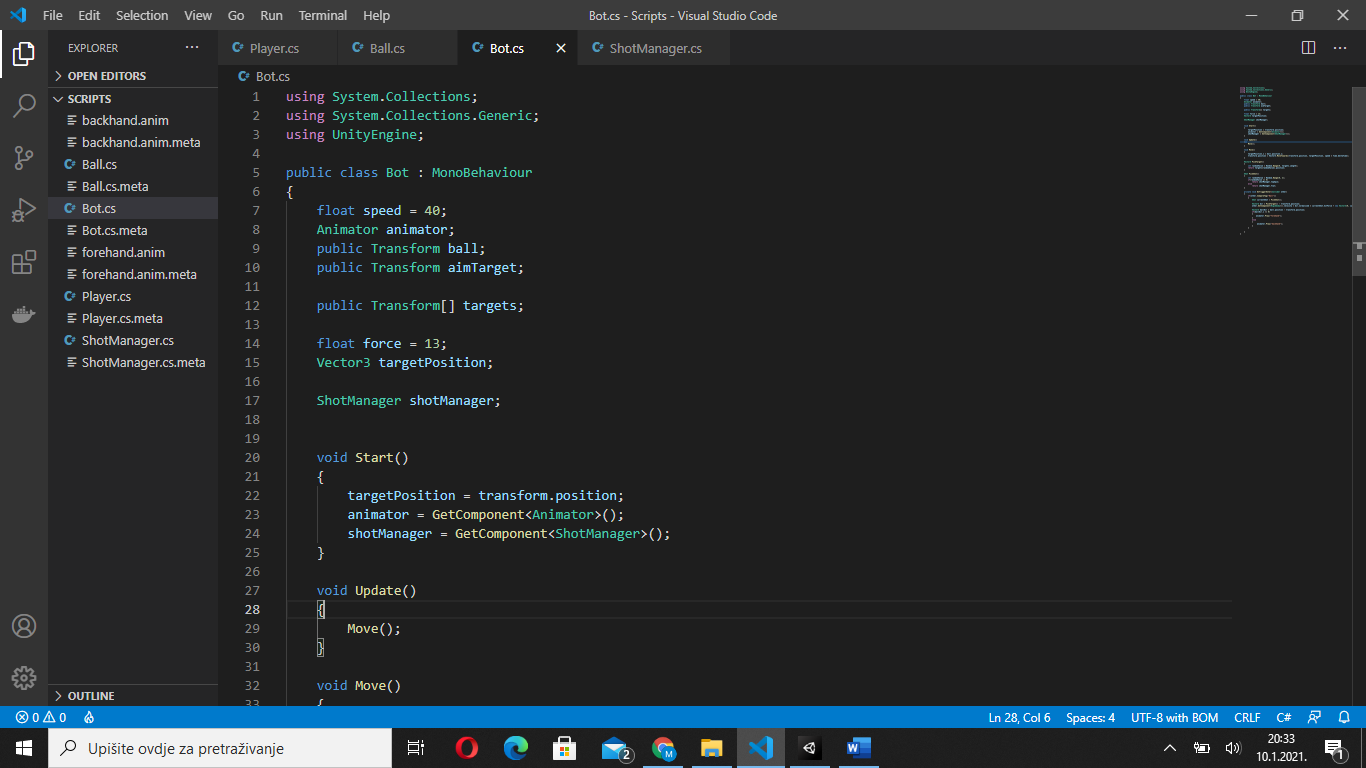
### Skripta *Ball*

Skripta *Ball* (Slika 17) definira kretanje teniske loptice nakon što je igrač dotakne. 

Slika 17 Skripta Ball

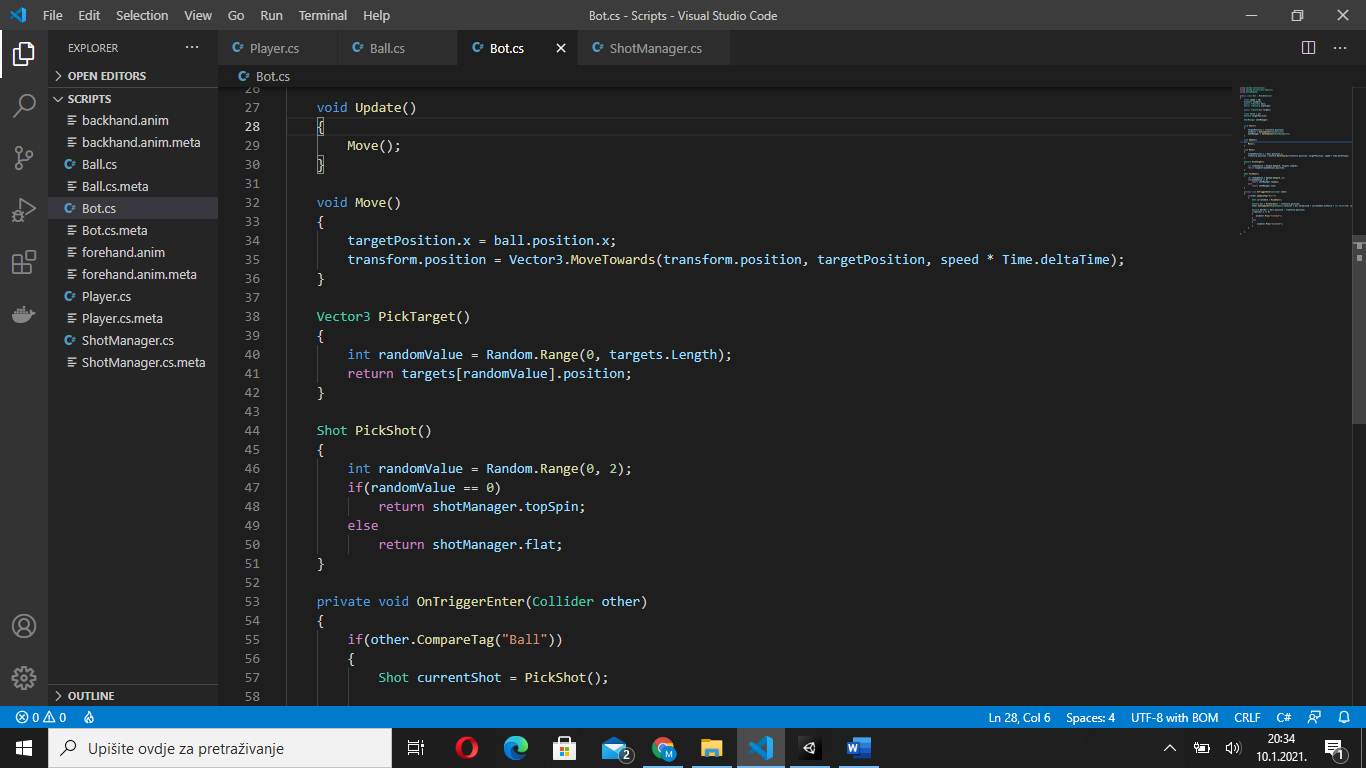
### Skripta *Bot*

Skripta *Bot* odnosi se na kretnje i udaranje loptice suprotnog igrača. Sadrži funkcije *Start*, *Update*, *Move*, *PickTarget*, *PickShot* i *OnTriggerEnter*. Dodane varijable i funkcija *Start* prikazani su na (Slika 18).



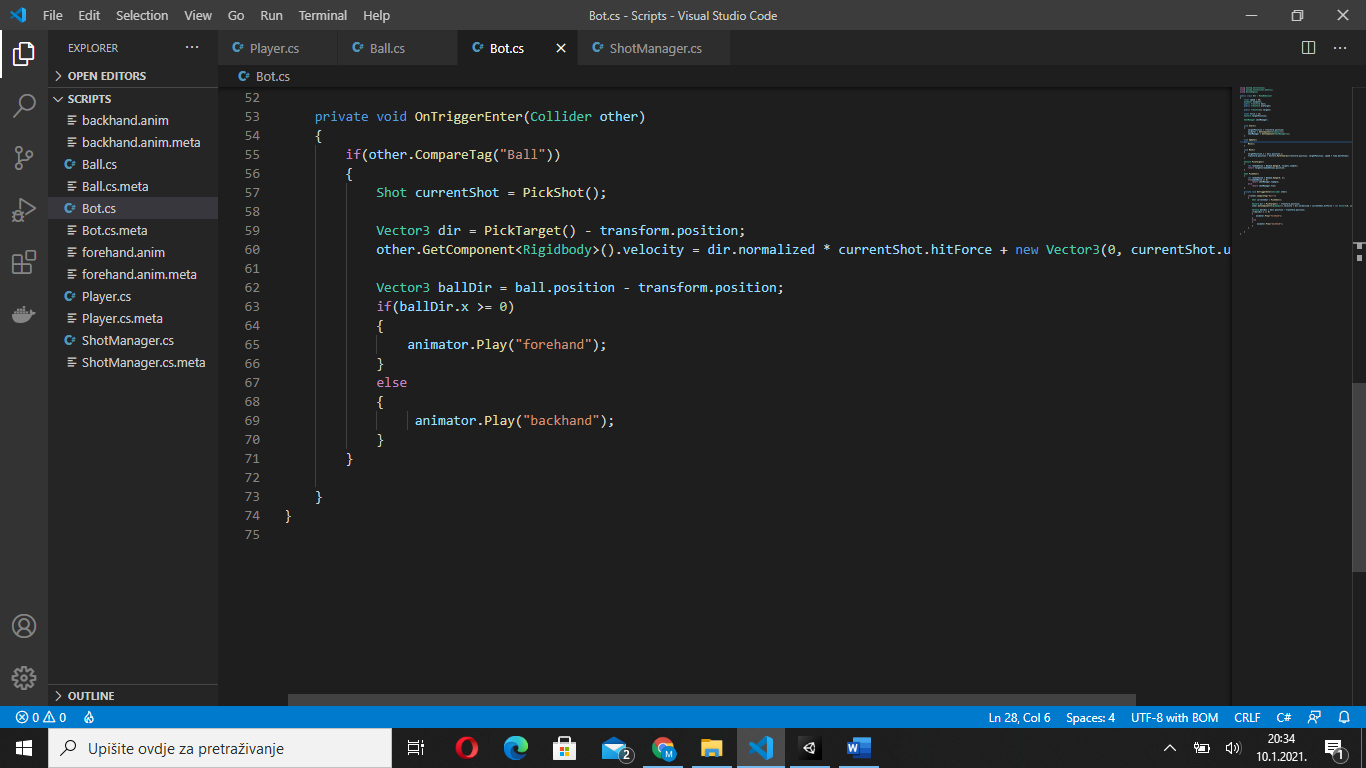
Slika 18 Dodane varijable i funkcija Start

Funkcije *Update* i *Move* zadužene su za kretanje protivničkog igrača prema loptici koja se kreće prema njemu. Funkcije *PickTarget* i *PickShot* određuju *random* smjer i brzinu kojom će protivnički igrač udariti lopticu (Slika 19)



Slika 19 Funkcije Update, Move, PickTarget i PickShot

Funkcija *OnTriggerEnter* prikazana je na slici (Slika 20)



Slika 20 Funkcija OnTriggerEnter

# ZAKLJUČAK

Korištenjem Unity Engine softvera napravljena je animacija teniske igre. U navedenom softveru postavljena je scena s modelima koja predstavlja realističan prikaz teniskog terena, a zatim je objektima igrača, meta i loptice dodan doživljaj (eng. *Event*). Pomoću napisanih skripti omogućena je realistična simulacija teniske igre.

Od 2018. godine Unity Engine softver koristi se za stvaranje približno polovice novih mobilnih igara na tržištu i 60 posto sadržaja proširene stvarnosti i virtualne stvarnosti.

# LITERATURA

[1] <https://en.wikipedia.org/wiki/Unity_(game_engine)>

[2] <https://docs.unity3d.com/Manual/UsingTheEditor.html>

[3] <https://learn.unity.com/tutorial/using-the-unity-interface#5c7f8528edbc2a002053b6c9>

[4] <https://docs.unity3d.com/Manual/LearningtheInterface.html?_ga=2.68317073.1445881095.1599514391-154929265.1596018361>

# SAŽETAK

**RAČUNALNA ANIMACIJA TENISKE IGRE**

Tema ovog seminarskog rada je, kroz animaciju, prikazati tenisku igru. Opisan je softver koji je korišten prilikom izrade animacije, *Unity* *Engine*, te su napisane njegove osnovne karakteristike i svojstva. Također je opisano kodiranje u *Unity* *Engine* softveru. Animacija je zamišljena u vidu teniske igre odnosno dobacivanja dvaju igača teniskom lopticom u ovisnosti o poziciji meta. Detaljno je opisan cijeli postupak izrade animacije: postavljanje scene, izrada modela, dodavanje materijala, pisanje skripti i dodavanje događaja na modele.

**KLJUČNE RIJEČI:** teniska igra, meta, animacija, Unity, model, skripta