Izrada i prikaz 3D modela u Google VR okruženju

Pišonić, Barbara

Undergraduate thesis / Završni rad

2020

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:353077

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-08-04**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli
Fakultet Informatike Pula

Barbara Pišonić

Izrada i prikaz 3D modela u Google VR okruženju

Završni rad

Rujan, 2020. Pula

Sveučilište Jurja Dobrile u Puli
Fakultet Informatike Pula

Barbara Pišonić

Izrada i prikaz 3D modela u Google VR okruženju
Creating and displaying 3D models in Google VR

Završni rad

JMBAG: 0303075413, redovita studentica

Studijski smjer: Informatika

Predmet: Modeliranje i simulacije

Mentor : doc. dr. sc. Darko Etinger

Rujan, 2020. Pula

IZJAVA O AKADEMSKOJ ČESTITOSTI

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

Student

_____

U Puli, _____, _____ godine

IZJAVA

o korištenju autorskog djela

Ja, Barbara Pišonić dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom Izrada i prikaz 3D modela u Google VR okruženju koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis

_____

Sadržaj

1. Uvod 1
2. Postupak kreiranja okoline 2
2.1. Kreiranje novog projekta 2
2.2. Postavljanje za VR 2
2.3 Podloga 5
2.4 Materijali 7
2.5 Zidovi 9
2.6 Postavljanje kamere 10
3. Ubacivanje objekata 13
3.1 3D Warehouse 13
3.2 Uređivanje scene pomoću objekata 14
3.3 Asset Store 15
4. Animacije 18
4.1 Mixamo 18
4.2 AI patrola 23
5. Zaključak 30
6. Literatura 31
Sažetak 32
Abstract 33

1. Uvod

Modeliranje i simulacije su česte pojave u današnjem društvu. Uzimajući u obzir brzinu razvoja tehnologije, nije ni čudo što danas možemo simulirati razna okruženja oko nas. Upravo to modeliranje i simulacije su napravili korak naprijed korištenjem virtualne realnosti. Virtualna realnost je mogućnost prikaza umjetne okoline. Korisnik koristi posebne naočale namijenjene upravo za VR, gdje svako oko ima svoj monitor. Ovisno o vrsti naočala, neke imaju i dodatnu mogućnost korištenja posebnih senzora koji prate pokrete tijela. Neke naočale također uključuju i rukavice koje pružaju mogućnost upravljanja predmetima u virtualnoj okolini. Tako, npr., studenti medicine mogu vježbati operirati putem virtualnog okruženja. Dakle, virtualna realnost ne mora nužno koristiti za zabavu, već nam može poslužiti i kako bismo se mogli obrazovati. Ovaj završni rad bazira se na korištenju virtualnih naočala za mobilne uređaje, koje nemaju nikakve dodatne mogućnosti osim pregleda virtualnog okruženja.

Završni rad napravljen je u sklopu projekta KA2 PROJECT 3D and VR for VET 2019-1-HR01-KA202-061006. Virtualno okruženje prikazuje „tvornicu“ koja sadrži modele robotske ruke koje su dio navedenog projekta, animirane likove i AI patrolu te ostale objekte. Korisnik ima mogućnost upravljanja koristeći pomicanje glave. Upravljanje funkcionira na način, ako korisnik podigne glavu, tada se može kretati, a ako ju spusti, zaustavlja se. U nastavku rada se detaljno opisuje postupak kreiranja ovog virtualnog okruženja, kako smo ubacili 3D modele, kako smo kreirali AI patrolu te kod za praćenje pomicanja glave.

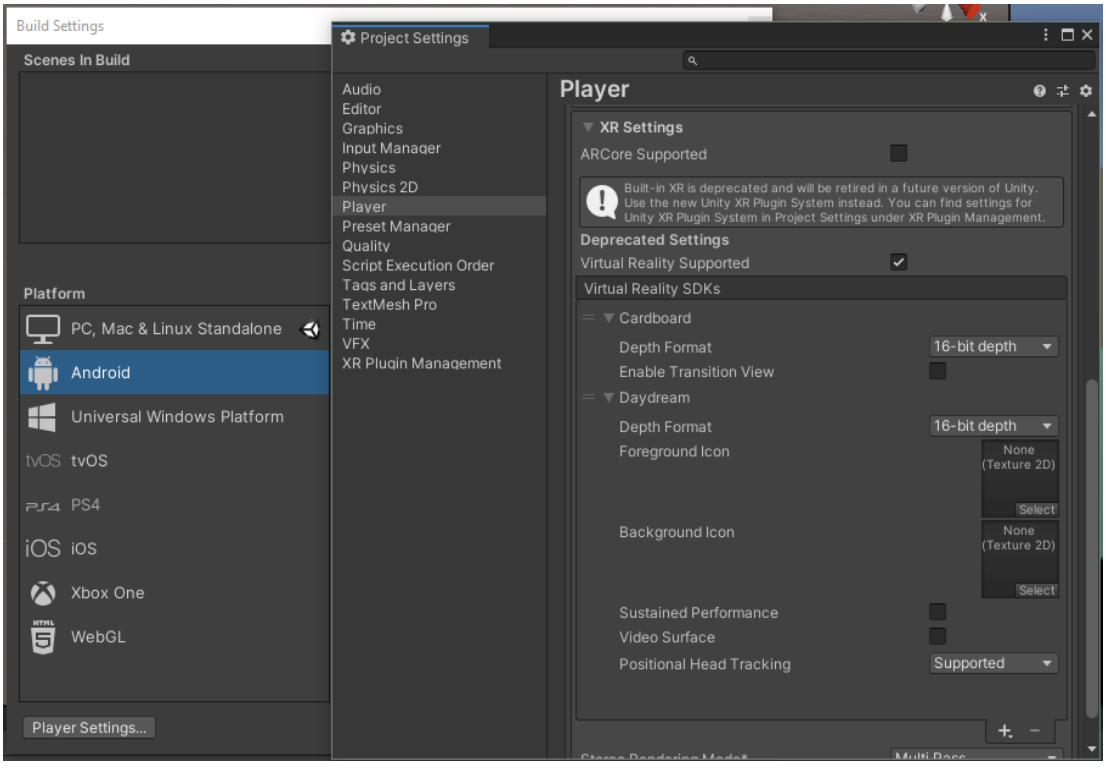
2. Postupak kreiranja okoline

2.1. Kreiranje novog projekta

Kako bismo započeli s radom, prvo trebamo kreirati novi projekt. Novi projekt kreiramo tako da otvorimo Unity, pritisnemo “New” sa postojećim postavkama te pritisnemo “Create”. Na početku dobijemo prazan projekt čiji *layout* možemo prilagoditi ovisno o našim potrebama. Prvo trebamo spremiti scenu kako bismo mogli spremiti sve nastale promjene u projektu. U gornjem lijevom kutu pritisnemo “File” i “Save as”. Bitno je spremiti unutar mape Assets, tj. u mapu Scenes. Nakon toga svaki put kada želimo spremiti promjene, možemo kliknuti ctrl + s ili otići na “File” i “Save”. Ukoliko želimo spremiti još jednu scenu, umjesto „Save“ pritisnemo „Save as“. Također je bitno da i iduća scena bude unutar Scene mape.

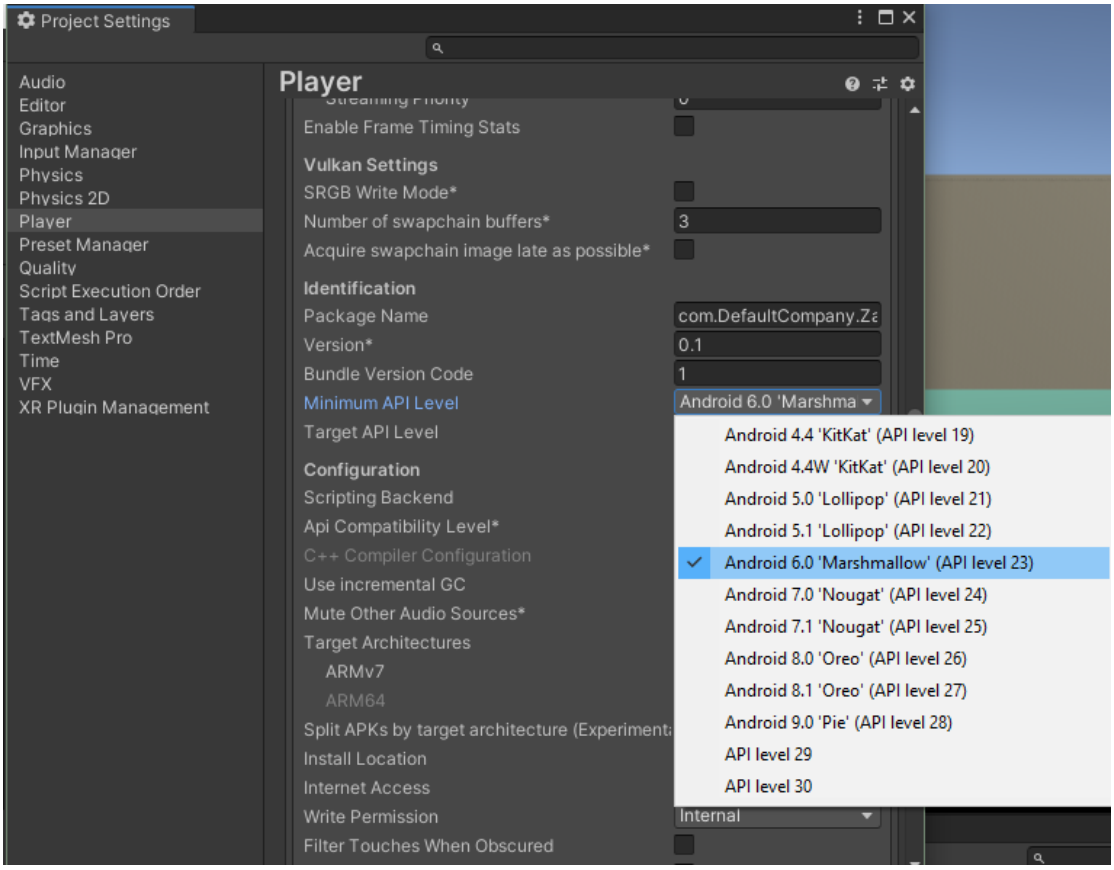
2.2. Postavljanje za VR

Prije nego započnemo s kreiranjem scene, prebacit ćemo build s PC-a na Android. To radimo tako da odemo na 'File' i kliknemo na 'Build Settings'. Tada nam se otvori novi prozor i pritisnemo 'Android'. Nakon toga odaberemo 'Player Settings' gdje idemo na 'Player' i 'XR Settings'. Pod 'Virtual Reality SDKs' dodamo Cardboard i Daydream. Zatvorimo prozor „Player settings“, prisnemo 'Build' i odaberemo gdje želimo spemiti naš .apk.



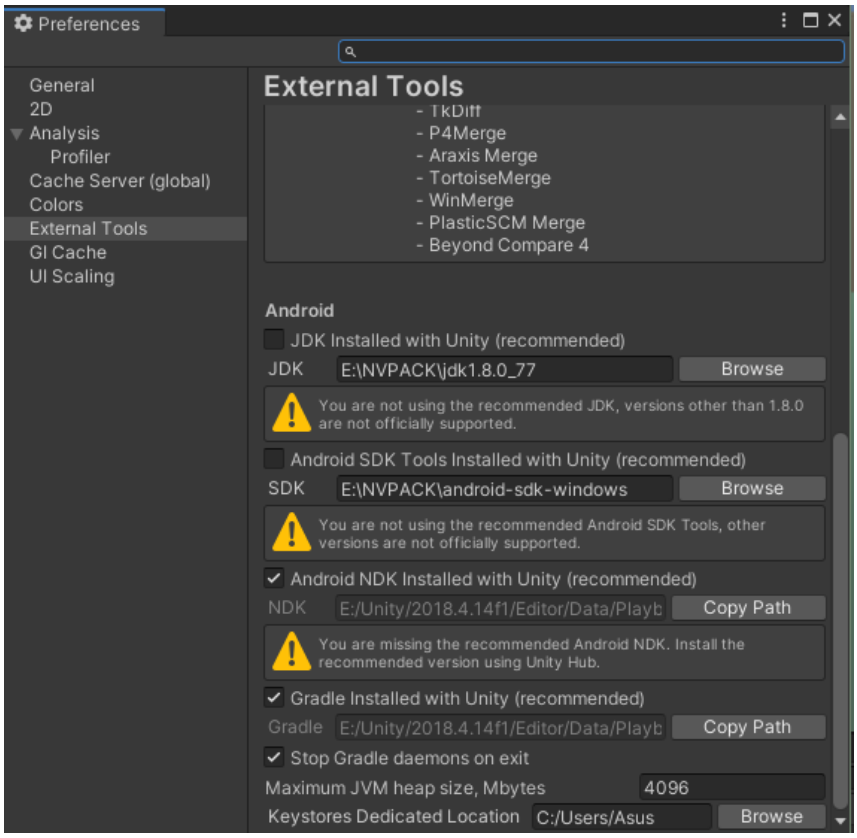
Slika 1 - Player Settings

Kako bi to sve moglo funkcionirati potreban nam je SDK. Sada moramo ići na Google i tražimo *Nvidia codework.* Moramo se registrirati na njihovoj stranici kako bismo mogli koristiti njihov SDK, no registracija je vrlo jednostavna. Nakon što se registriramo, preuzmemo i instaliramo datoteku, možemo dodati SDK našem projektu. Ako želimo buildati aplikaciju za Cardboard, poželjno bi bilo postaviti minimalni API level. To radimo tako da pod „Player Settings“ odemo na „Other Setting“ te pronađemo „Minimum API Level“. Njega možemo postaviti na API level 25. U našem slučaju smo stavili API level 23.



Slika 2 - Odabrani minimalni API lvl

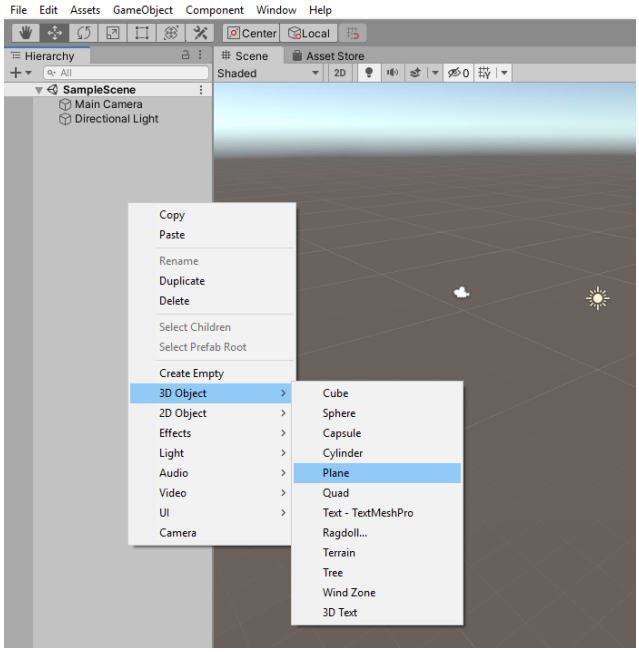
Zatim trebamo postaviti naš android uređaj kako bismo mogli testirati na njemu. Odaberemo postavke te pritisnemo „O uređaju“. Zatim je potrebno odabrati 'Podaci o softveru' i onda pritisnemo cca 7 puta „Broj verzije“. Nakon toga ćemo dobiti poruku da je način programera uključen. Sada kada opet odemo na postavke, možemo uočiti da imamo dodatno polje koje se zove „Opcije razvoja“. Kliknemo na njih te omogućimo „Otklanjanje pogrešaka putem USB-a“. Sljedeće trebamo otići na „Sigurnost“ i omogućiti instalacija aplikacija koje nisu s Play Store. Ako dobijete poruku da nemate SDK, kliknite na „Edit“ i „Preferences“. Pod „External tools“ možemo vidjeti koji SDK je trenutno izabran. Tamo se mora nalaziti onaj koji smo skinuli s Nvidia codework. Sada možemo uspješno testirati našu aplikaciju putem mobitela.



Slika 3 - Prikaz trenutnog SDK

2.3 Podloga

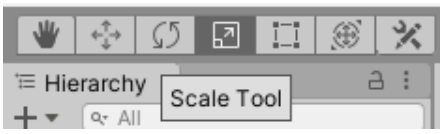
Prije nego dodamo modele, moramo staviti podlogu. S lijeve strane u “Hierarchy” pritisnemo desni klik i odaberemo “3D Object” i zatim “Plane”.



Slika 4 - Kreiranje podloge

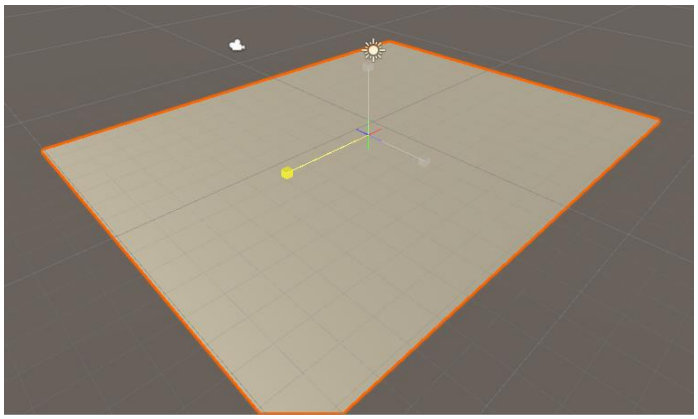
Nakon što smo uspješno kreirali podlogu za naše modele, možemo odabrati njenu veličinu. Veličinu možemo odabrati na 3 načina:

1. Koristeći “Scale tool”



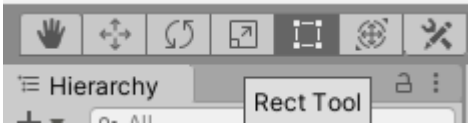
Slika 5 – Scale tool

Zatim vidimo strelice crvene, plave i zelene boje. Crvena i plava strelica upravljaju širinom, odnosno dužinom pozadine. Ako odaberemo jednu strelicu te ju povučemo u jednom smjeru, podloga će se automatski proširiti, tj. produljiti na suprotnoj strani. Isto tako odabirom središnje točke možemo povećati ili smanjiti cjelokupnu podlogu



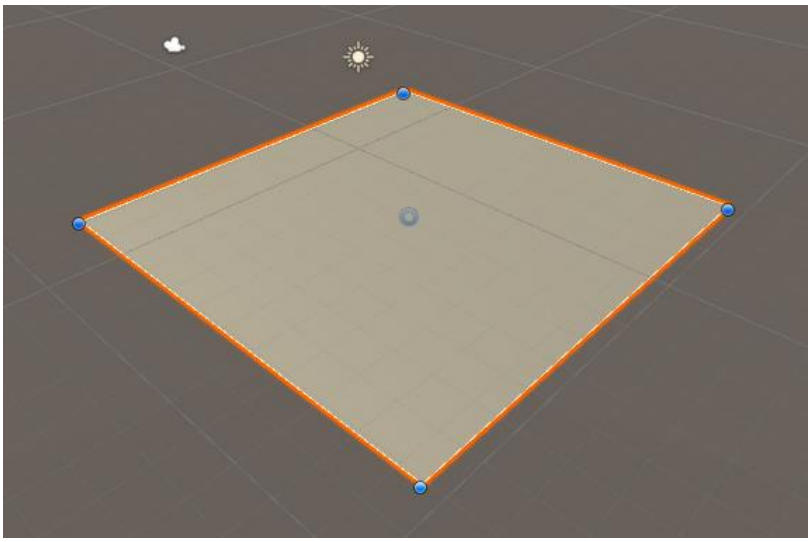
Slika 6 - Povećanje planea putem scale alata

2. Koristeći “Rect tool”



Slika 7 - Rect tool

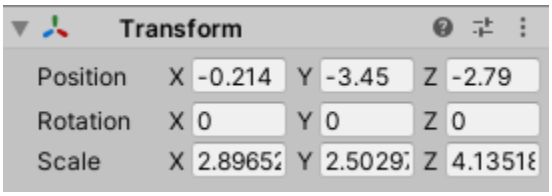
Ovaj način vrlo je sličan prethodnom. Odabirom ovog načina, na našoj podlozi prikažu se točke na svakom kutu putem kojih biramo njenu veličinu.



Slika 8 - Povećanje planea putem rect alata

3. Koristeći x i z

S desne strane projekta nalazi se “Transform” polje gdje možemo vidjeti “Scale”, odnosno veličinu odabranog objekta. Taj je objekt u ovom slučaju podloga, tj. plane. Ručnim unosom x i z mijenjamo veličinu podloge. Možemo primijetiti mogućnost mijenjanja x-a, ali u ovome slučaju ukoliko promjenimo x, visina podloge ostaje ista. Uz “Scale” polje vidimo i “Position” - poziciju i “Rotation” - rotacija. Samo ime kaže što oni rade, no kasnije ćemo ih detaljnije opisati.



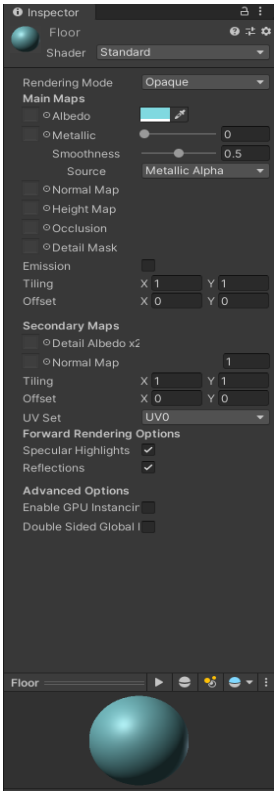
Slika 9 - Transform vrijednosti

2.4 Materijali

Kako bismo kreirali materijal, trebamo kreirati novu mapu radi bolje organizacije. Osim mape „Materials“, možemo odmah kreirati i mape „Models“, „Scripts“ i „Textures“. Zatim uđemo u mapu materijala te pritisnemo desni klik, „create“ i odaberemo materijal. Sve te mape moraju se nalaziti unutar glavne mape „Assets“.

Slika 10 - Hijerarhija mapa

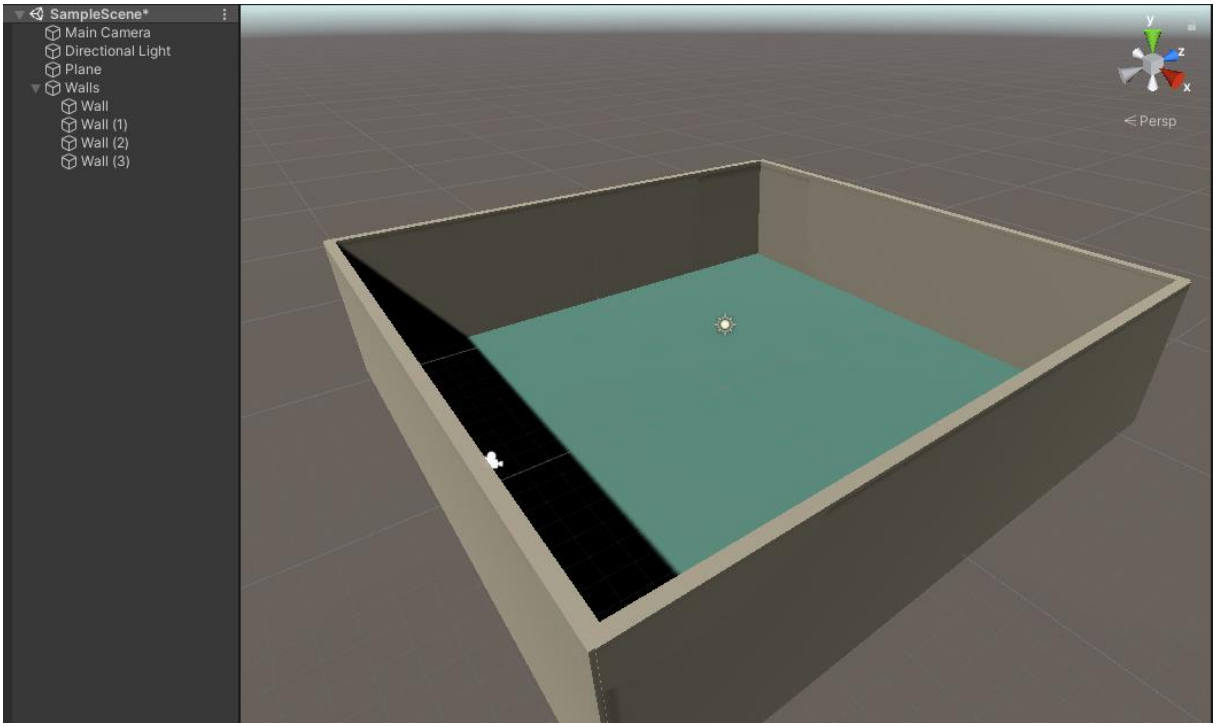
Nakon što kreiramo novi materijal možemo dva puta kliknuti na njega. Primijetimo kako se s desne strane nalazi novi prozor koji se zove „Inspector“. U tom prozoru vidimo detalje o odabranom objektu. U našem je slučaju to materijal koji možemo urediti po našem izboru. Za sada ćemo ga preimenovati u „Floor“ te ćemo sada trenutno odabrati plavu boju, koju ćemo kasnije promijeniti. Kako bismo primijenili taj materijal na podlogu, možemo ga jednostavno odabrati i „odvući“ na podlogu.



Slika 11 - Novi materijal

2.5 Zidovi

Kako naše okruženje neće biti otvorenog tipa, kreirat ćemo zidove. Prvo ćemo u hijerarhiji kreirati prazni objekt kako bi struktura cijelog projekta bila preglednija. Taj prazni objekt ćemo nazvati „Walls“. Zatim kreiramo nove 3d objekte – kocke. Možemo kreirati samo jednu, urediti ju po našoj želji te ju onda duplicirati. U inspektoru kocke, odaberemo željene dimenzije našeg zida te ga postavimo na željeno mjesto.

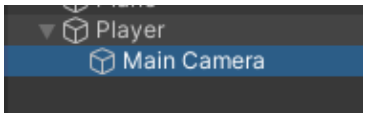


Slika 12 - Kreiranje zidova

Sada ako želimo možemo ili kreirati novi materijal ili staviti postojeći materijal, s podloge, na zidove. Ako želimo, možemo kreirati i strop, ali za sada ćemo ostaviti prostoriju bez stropa zbog boljeg svjetla. Svijetlo dodajemo na isti način kao što kreiramo nove objekte te možemo odrediti jačinu i domet putem inspektora. Sada možemo prijeći na uređenje scene. Na isti način možemo kreirati i dvije dodatne sobe koje će nam poslužiti kao ured i kao soba gdje ćemo prikazati logo sveučilišta.

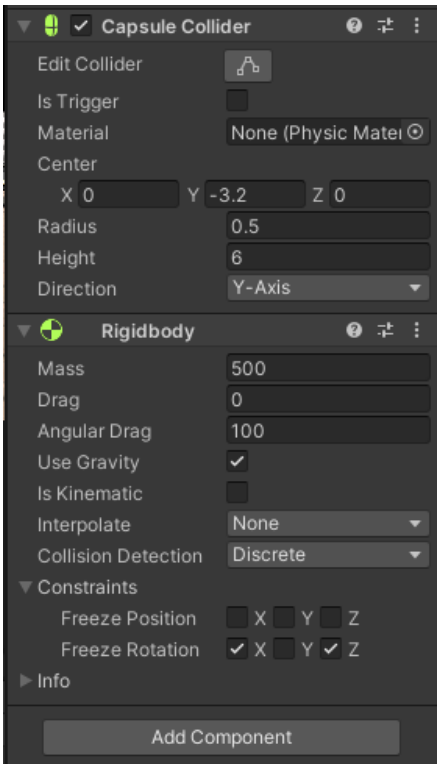
2.6 Postavljanje kamere

S obzirom na to da se ovaj projekt bazira na virtualnoj realnosti, znači da moramo postaviti kameru tako da bude u prvom licu. Kako bismo to postigli treba nam prazni objekt koji ćemo nazvati „Player“. Nakon objekta, povlačimo „Main camera“ pod taj objekt. Tako sada imamo objekt unutar objekta. Pod playera dodajemo nove komponente, *Rigidbodyi Capsule Collider.* Capsule Collider je najbolji izbor u našem slučaju jer je najsličniji oblik ljudskom tijelu. Kasnije ćemo playeru dodati i skriptu koja će hodati u smjeru u kojem se usmjeri mobilni uređaj.



Slika 13 - Glavna kamera

Komponente će imati vrijednosti kao na slici. Kod Capsule Collidera biramo visinu (height) i širinu (radius). Ona zapravo prezentira tijelo našeg igrača, kao njegov kostur. Rigidbody služi za gravitaciju. Biramo koju masu te postavljamo da ćemo koristiti gravitaciju. Znači u slučaju da se dva tijela sudare, reagirali bi na temelju postavki gravitacije koje smo unijeli. Iz tog razloga je masa 500, kako naš igrač ne bi „odletio“ pri sudaru. Također zamrzavamo rotaciju na X i Z kako se igrač ne bi srušio. Isto tako smanjujemo rizik od *motion sicknessa* (bolesti putovanja).



Slika 14 - Komponente Capsule COllider i Rigidbody

Sljedeće dodajemo skriptu igraču. U skripti se nalazi canMove vrijednost preko koje pratimo može li se igrač kretati ili ne. Ako je vrijednost jednaka true, naš igrač će se kretati tako da se transformiraju vrijednosti kamere koja je na igraču (ona predstavlja oči, tj ono što igrač vidi). Također vidimo vrijednost speed pomoću koje postavljamo brzinu kretanja igrača. Ako se igrač može kretati i ako je pogled ispod određenog kuta, igrač staje. Ako se igrač ne može kretati, a pogleda iznad određenog kuta, tada se počne kretati. U slučaju sudara s ostalim objektima, čija je oznaka „stopper“, igrač se prestane kretati.

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

public class FollowPlayer : MonoBehaviour
{
    public float speed = 30f;
    public bool canMove = false;

    void Update()
    {
        float xRot = Camera.main.transform.eulerAngles.x;

        Quaternion rotMin = Quaternion.Euler(new Vector3(-20f, 0, 0));
        Quaternion rotMax = Quaternion.Euler(new Vector3(20f, 0, 0));
        Quaternion rotation = Camera.main.transform.rotation;
        if(!canMove){
            if(rotation.x <= rotMin.x){
                canMove = true;
                Debug.Log("Go");
            }
        }
        if(canMove){
            if(rotation.x >= rotMax.x){
                canMove = false;
                Debug.Log("Stop");
            }
        }
```

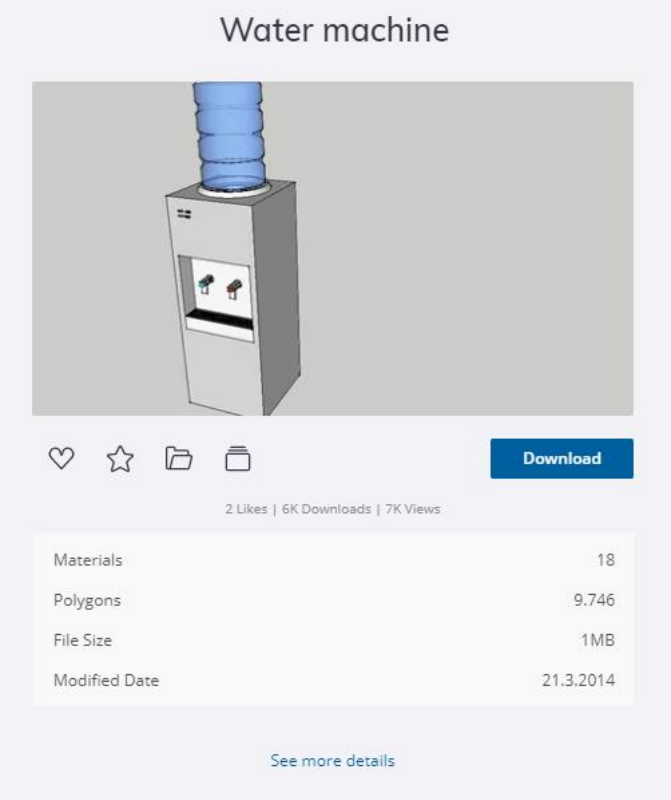
```
        if(canMove){
            Vector3 cameraAngle = Camera.main.transform.forward;
            this.gameObject.transform.Translate(cameraAngle * Time.deltaTime * speed);
            Debug.Log("Walking");
        }
    }

    private void OnCollisionEnter(Collision otherObject)
    {
        if(otherObject.transform.tag == "stopper"){
            canMove = false;
        }
        Debug.Log("Collision detected: " +  otherObject.gameObject.name);
    }
}
```

3. Ubacivanje objekata

Kako smo završili s postavljanjem kamere i igrača, prelazimo na objekte. Objekt robotske ruke smo ubacili ubacivanjem u mapu Models. Ostale objekte ubacujemo preko stranice 3D Warehouse.

3.1 3D Warehouse

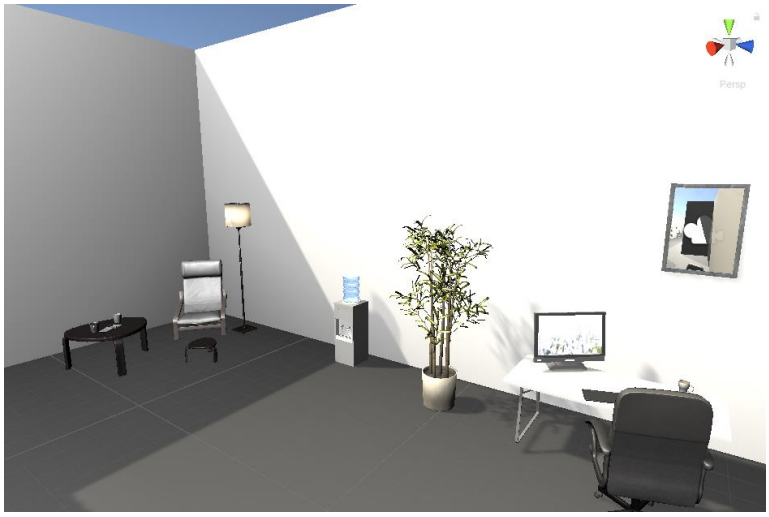
3D Warehouse pruža uslugu korištenja besplatnih modela koje su kreirali drugi korisnici. Pri odlasku na stranicu biramo koje objekte želimo u našem projektu. Odaberemo model koji nam se sviđa te pritisnemo download. Izaberemo koje sve modele želimo, preuzmemo ih te ih ubacimo u mapu Models. Sada s obzirom na to da imamo više modela, možemo početi uređivati scenu.



Slika 15 - Primjer modela iz 3D Warehouse

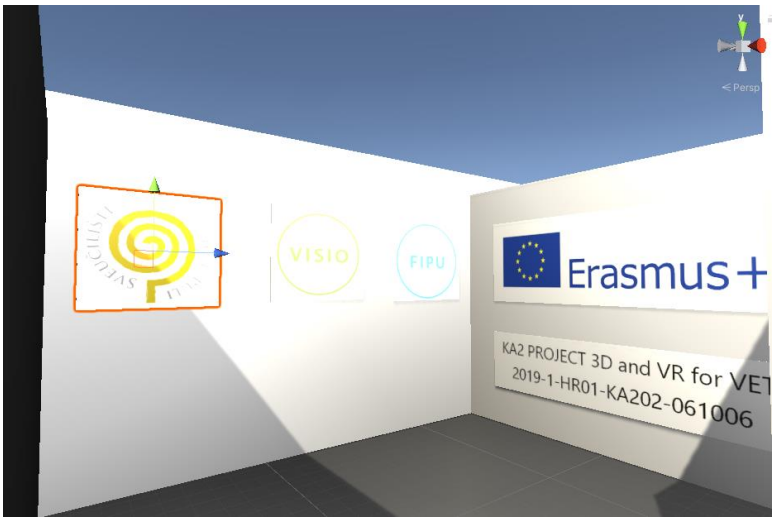
3.2 Uređivanje scene pomoću objekata

S obzirom na to da smo preuzeli sve željene objekte, možemo ih ubaciti na scenu. Tako ćemo kreirati mali ured u koji ćemo kasnije ubaciti i likove s određenim animacijama. Pri unosu objekta na scenu, biramo njegove dimenzije putem inspektora, kao što smo do sada radili sa *planeom* i sa zidovima.



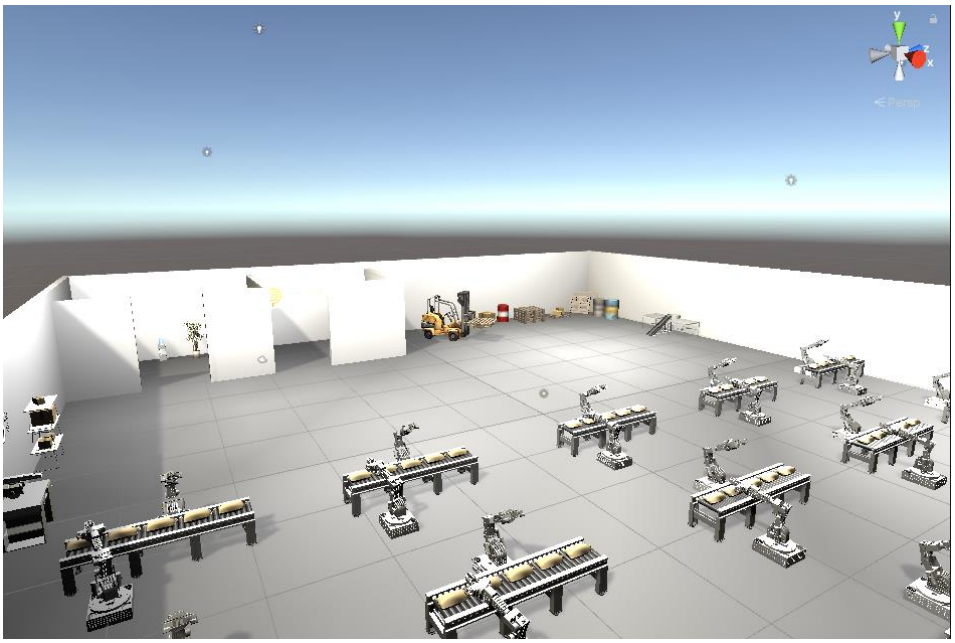
Slika 16 - Prikaz scene sa modelima

Prva prostorija je gotova. Sada ćemo kreirati sljedeću prostoriju u koju ćemo ubaciti logo sveučilišta i fakulteta. Preuzmemo png logoa koje trebamo te ih importamo u mapu Models. Zatim kreiramo novi objekt – kocku. Toj kocki dajemo dimenzije koje odgovaraju slici na zidu. Odaberemo logo koji želimo i samo ga prenesemo na kocku koja se nalazi u sceni. Rezultat toga je sljedeće:



Slika 17 - Prostorija sa logoima

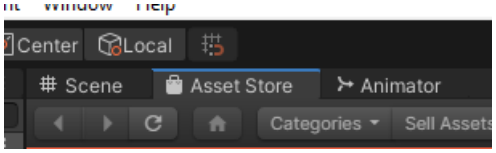
U prostoriji se nalaze logoi Sveučilišta J.J., FIPU, Erasmus +, FIPU te naziv projekta. Na isti način uređujemo ostatak scene. Ubacujemo objekte koje smo preuzeli sa 3D Warehouse te model robotske ruke. Kako smo ubacili sve modele koje smo htjeli te ih postavili na željena mjesta, možemo odmah i kreirati materijale kako bismo ih uredili. Tako dobijemo ispunjenu scenu, ali u kojoj se trenutno ništa ne dešava.



Slika 18 - Prikaz modela cijele scene

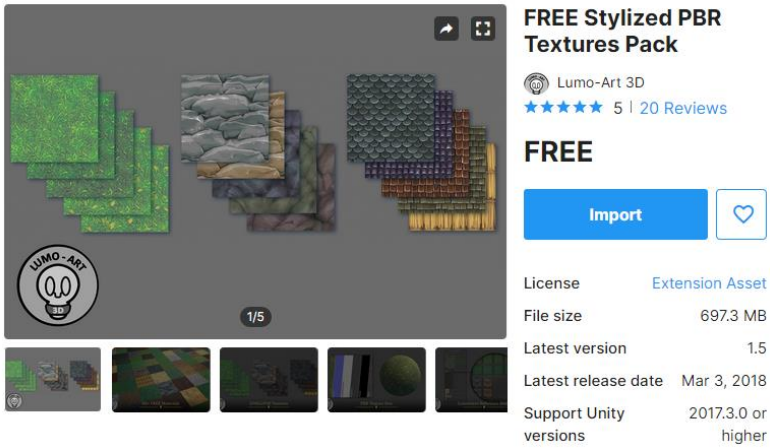
3.3 Asset Store

S obzirom na to da ne želimo jednolični izgled poda i zida, preuzet ćemo gotove proizvode putem Unity Asset Store. Tablica Asset Store se nalazi odmah pokraj scene.



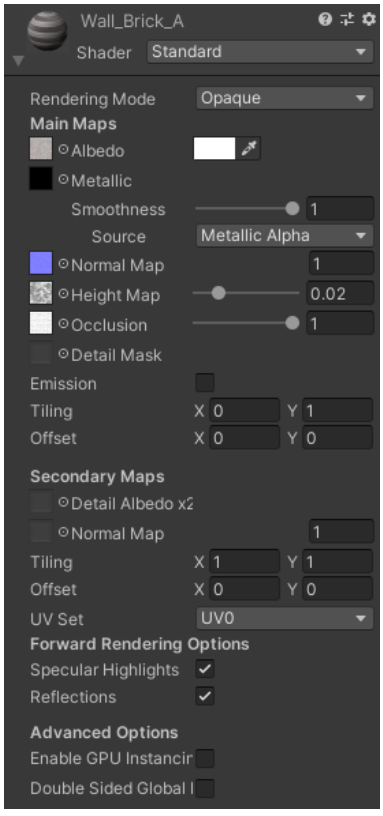
Slika 19 - Asset store

Upišemo što tražimo te odaberemo cijenu. U našem slučaju tražimo materijale za pod i zid te smo pod cijenu stavili besplatno. Pronađemo što nam odgovara te pritisnemo download. Kada preuzimanje završi potrebno je ubaciti materijale u Assets mapu.

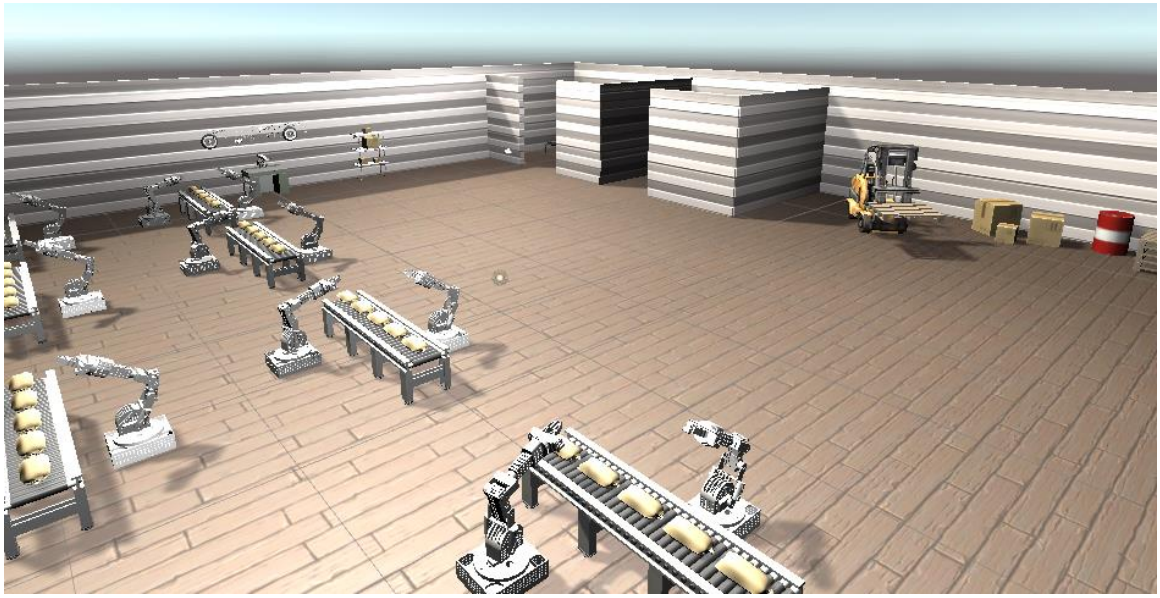


Slika 20 - Materijali sa Asset Store

Izaberemo materijal koji nam se sviđa te ga prenesemo na zidove i pod. Odabrani materijal možemo urediti po našoj želji, npr. koliko će se puta neki uzorak ponavljati te možemo podesiti boju...



Slika 21 - Postavke materijala



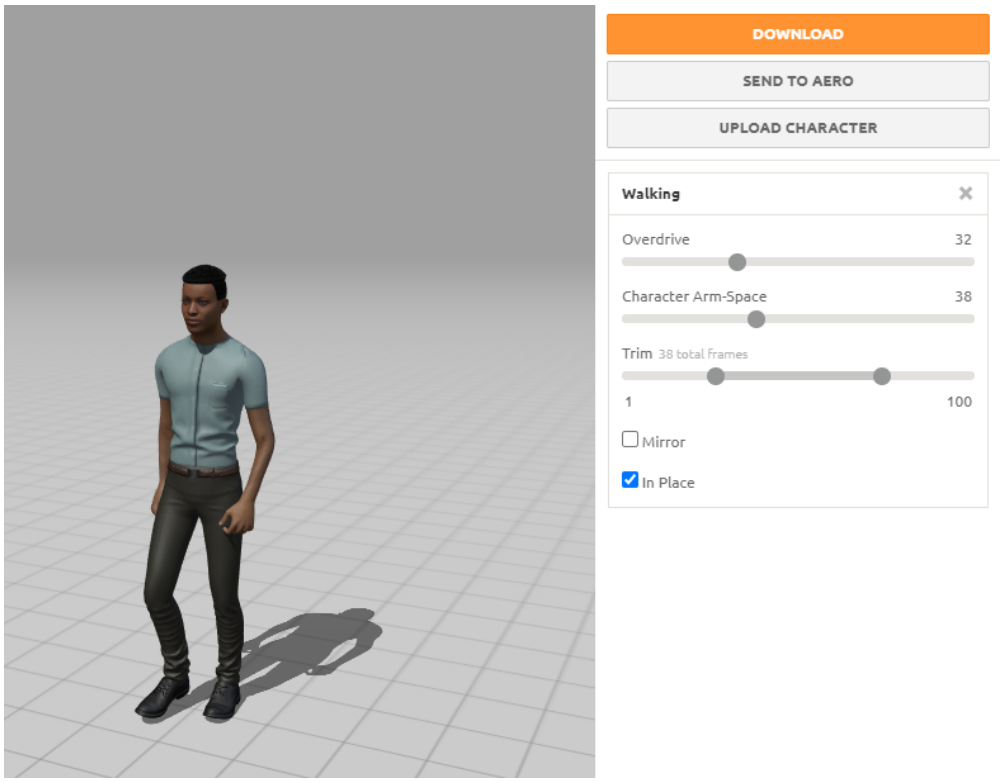
Slika 22 - Prikaz materijala zidova i poda

Uređenje scene je gotovo te se možemo početi baviti kreiranjem AI patrole koja će imati točno određenu rutu po kojoj će se kretati. Također ćemo dodati još radnika kako bi cjelokupno iskustvo bilo življe.

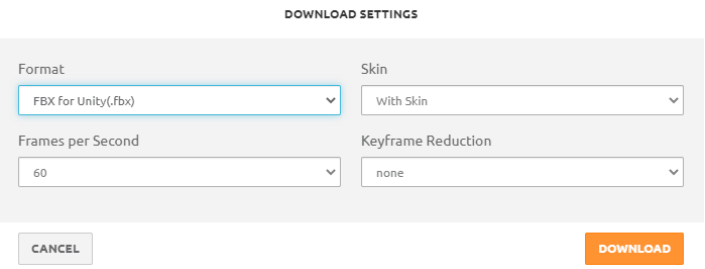
4. Animacije

4.1 Mixamo

Mixamo je tvrtka koja omogućuje besplatno korištenje 3D animacija, ali isto tako i prodaje svoje usluge. Tehnologija kojom se bave koristi metode strojnog učenja za automatizaciju koraka procesa animacije likova, uključujući 3D modeliranje do rigovanja i 3D animaciju. Ako želimo preuzeti njihove likove i animacije, moramo se registrirati. Nakon registracije imamo mogućnost preuzimanja likova i animacija. Izaberemo željenog lika te kliknemo na download. U skočnom prozoru odaberemo FBX format i T-Pose. Zatim biramo koje animacije su nam potrebne. Za ovaj projekt ćemo koristiti animacije razgovora, tipkanja po tipkovnici, hodanja i razmišljanja. Tako možemo redom tražiti animacije te za sve njih vrijede isti uvjeti preuzimanja. Bitno je da na animaciji, pri preuzimanju, odaberemo „In place“ kako bismo lakše upravljali animacijom unutar Unityja. Kod pojave skočnog prozora pod „Format“ izaberemo FBX for Unity.

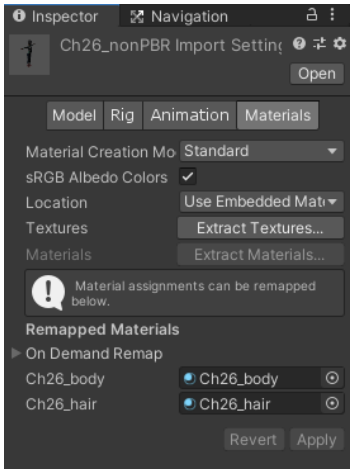


Slika 23 - Mixamo animacija

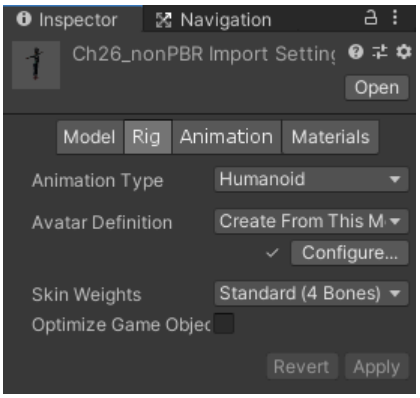


Slika 24 - Postavke preuzimanja animacije

Prije nego ubacimo likove i animacije u Unity, kreiramo nove mape pod nazivom Characters i Animations. Možemo ih jednostavno ubaciti tako da ih dovučemo u određene mape. Pod mapom Characters moramo svim likovima *extractati* materijale i teksture. Pritiskom na lika, otvara se inspektor s desne strane. Pod tablicom Materials vidimo dio gdje se nalaze teksture i materijali. Pritisnemo Extract Textures i izaberemo mapu Textures pod mapom Assets. Isto tako napravimo i za materijale, samo u mapu Materials. Također je bitno pod tablicom Rig za Animation Type odabrati Humanoid.

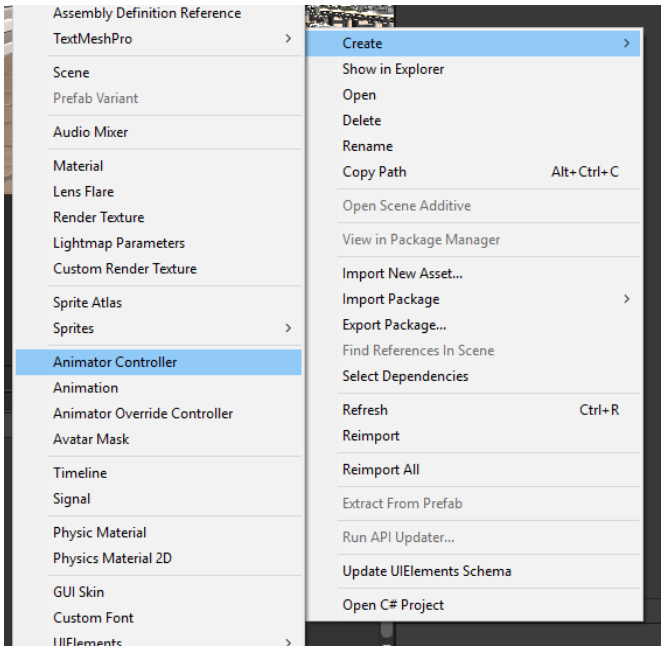


Slika 25 - Extractanje tekstura i materijala



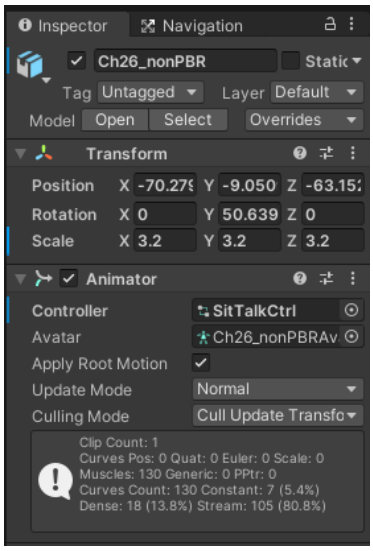
Slika 26 - Humanoid oblik animacije

Nakon uspješnog ubacivanja likova, red je na animacije. No, prije nego krenemo animirati, bitno je dodati Animator Controller na lika koji će imati neku animaciju. U mapi Characters pritisnemo desni klik te odaberemo Create i zatim Animator Controller.

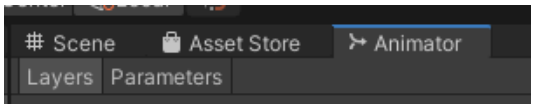


Slika 27 - Kreiranje Animator Controllera

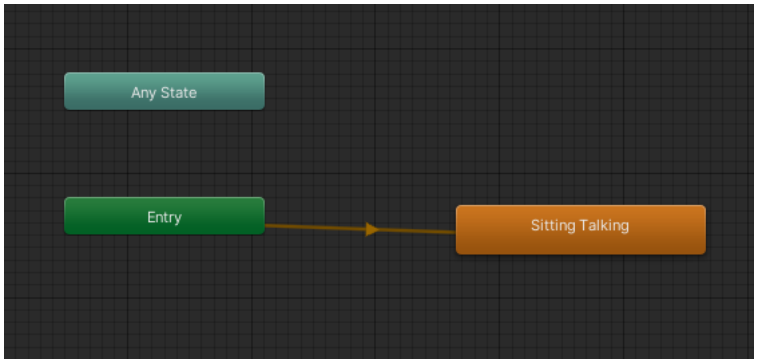
Animator Controller povučemo kod animatora lika gdje piše *Controller*. Sada trebamo povezati animaciju s likom. Pritiskom na *Animator* tablicu pokraj tablice scene, dobijemo uvid u detalje o animatoru i animaciji. Tj, potrebno je ubaciti željenu animaciju u tu tablicu kako bismo ju mogli kontrolirati.



Slika 28 - Animator komponenta na liku

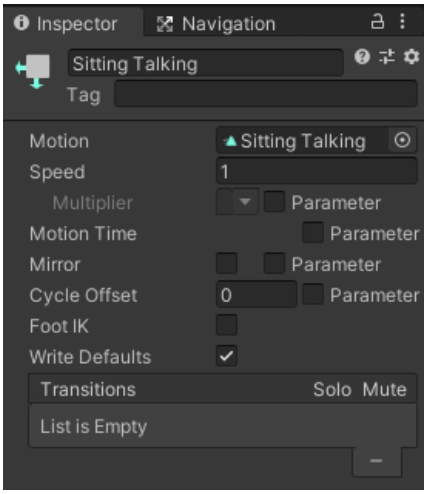


Slika 29 - Animator tablica

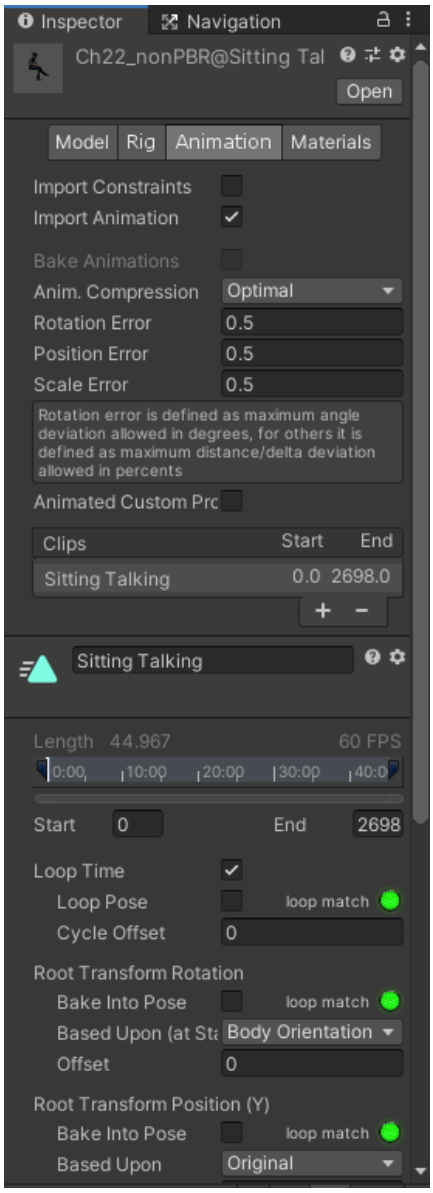


Slika 30 - Prikaz animacija u animatoru

Ubacivanjem animacije automatski povezujemo lika i animaciju. Znači, animacija se povezala s likom putem Animator Controllera. Jedan klik na *Sitting Talking* animaciju dobijemo jedne postavke. Dva klika na *Sitting Talking* daju druge postavke. Prvi klik prikazuje neke opće postavke, tipa brzina odvijanja animacije. Dva klika pokazuju malo više detalja animacije. Ovdje odaberemo da želimo ponavljati animaciju, tj. označimo *Loop Time* sa kvačicom.Tu je također bitno da je pod *Rig* odabran *Humanoid* oblik.



Slika 31 - Postavke animacije nakon jednog klika

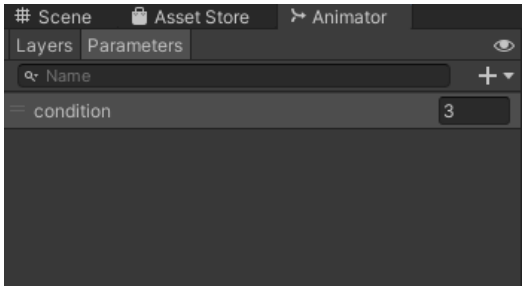


Slika 32 - Postavke animacije nakon dva klika

Položaj likova možemo podesiti dok je scena aktivna. Ako ih ide postaviti na poziciju dok scena nije aktivna, naići ćemo na probleme jer kada je scena aktivna, točno možemo vidjeti gdje i kako se lik kreće. Ako nije aktivna, lik će biti u T-pozi i sve će nam biti otežano. Idući korak je kreiranje AI patrole koristeći likove i animacije, također iz Mixamo.

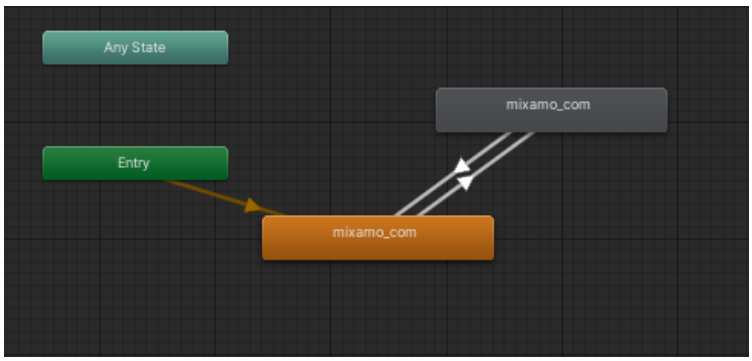
4.2 AI patrola

Koristeći znanje koje smo do sada prikupili, možemo ubaciti željenog lika koji će predstavljati AI na scenu. Na isti način kreiramo Animator Controller, ali ovaj puta ćemo postavljati neke parametre kako bi AI znala kada se lik treba kretati. Postavljamo neku *idle* pozu, no ona nije nužna u ovome slučaju, i postavljamo animaciju kretanja. Postavljamo parametar *condition* kao broj 3. Znači kada lik dobije određeno brzinu, u ovom slučaju 3, on će se početi kretati.



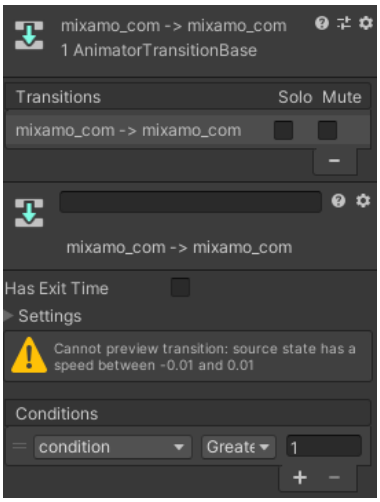
Slika 33 - Animator parametri

Prvo stavljamo *idle* pozu u animator te zatim dodajemo animaciju kretanja. Napravimo da su animacije međusobno povezane tako da znamo kada se prelazi s koje animacije.

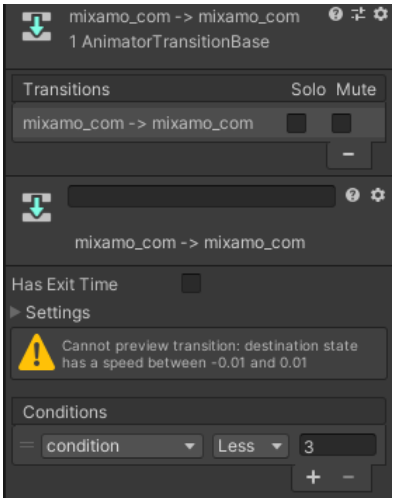


Slika 34 - Prijelaz iz stanja u stanje, odnosno iz animacije u animaciju

Prelazak iz *idle* poze u animaciju kretanja treba imati uvjet da je *condition* veći od 1, a prelazak iz animacije kretanja u *idle* pozu treba imati uvjet *condition* je manji od 3.

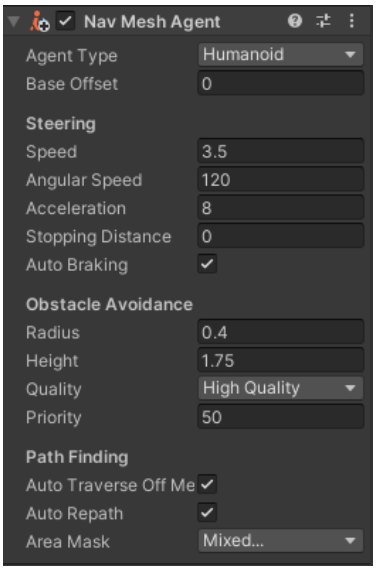


Slika 35 - Uvjet prelaska iz idle animacije u animaciju kretanja



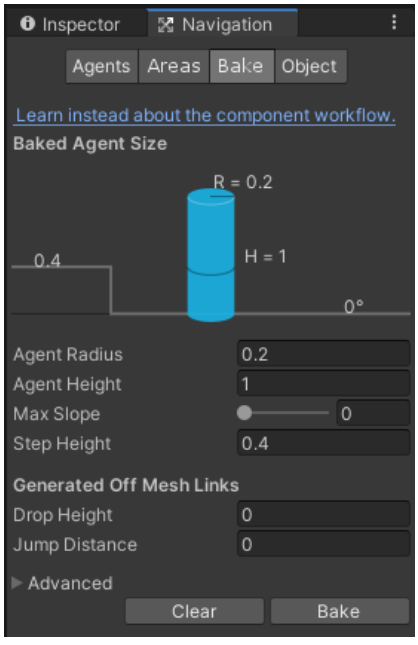
Slika 36 - Uvjet prelaska iz animacije kretanja u idle animaciju

Nakon toga dodajemo nove komponente AI patroli, *Rigidbody*, *Nav Mesh Agent* i *Collider*. Pod *Rigidbody* ćemo označiti da je AI kinematička kako ne bi okolina utjecala na nju. Zatim imamo *Collider* kojem biramo visinu i širinu kako bi odgovarao dimenzijama lika koji predstavlja AI. *Nav Mesh Agent* je nova komponenta koju do sada nismo koristili. Ona je ta koja će navoditi lika gdje se treba kretati. Pod *Agent Type* stavljamo Humanoid te također visina i širina moraju odgovarati dimenzijama lika.



Slika 37 - Nav Mesh Agent

Potrebno je označiti površinu po kojoj će se patrola kretati. U tablici *Navigation* koja se nalazi pokraj inspektora, vidimo postavke koje su potrebne za to. Odaberemo radius i visinu te ostale detalje, ovisno o kojoj površini se radi, ima li površina stepenice ili ide li uzbrdo... Odabrana površina po kojoj se patrola može kretati bit će označena plavom bojom na sceni. Kada dobijemo zadovoljavajuće rezultate, pritisnemo bake.



Slika 38 - Odabir podloge kretanja agenta

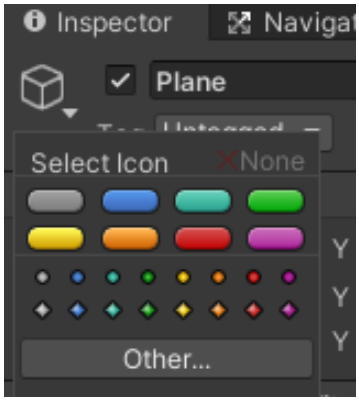
Kako bi AI funkcionirala kako treba, potrebno joj je reći što i kada treba raditi. Iz tog razloga kreiramo skriptu. Možemo koristiti službenu skriptu Unity dokumentacije *(https://docs.unity3d.com/Manual/nav-AgentPatrol.html*).

```
public class Patrol : MonoBehaviour {
        public Transform[] points;
        private int destPoint = 0;
        private NavMeshAgent agent;
        void Start () {
            agent = GetComponent<NavMeshAgent>();
            agent.autoBraking = false;
            GotoNextPoint();
        }
        void GotoNextPoint() {
            if (points.Length == 0)
                return;
            agent.destination = points[destPoint].position;
            destPoint = (destPoint + 1) % points.Length;
        }
        void Update () {
            if (!agent.pathPending && agent.remainingDistance < 0.5f)
                GotoNextPoint();
        }
    }
```

Postavljamo varijablu agent sa svojstvom *Nav Mesh Agent*. Ako onemogućimo autobreaking svojstvo kod agenta, tada agent neće „kočiti“, tj. usporavati kako se približava nekoj točki. Zatim se poziva funkcija kod koje se agent vraća na početnu točku, ako je obišao sve točke. Agent ide prema sljedećoj odabranoj točki u polu, znači redoslijed točki prati putem polja. U polje postavimo sve željene točke, znači polje može sadržavati onoliko točaka koliko mi stavimo. Update funkcija bira sljedeću točku koja ide nakon trenutne. Agent uvijek računa koliko točaka ima ukupno u polju, na kojoj se agent trenutno nalazi, prema kojoj treba ići i koliko ih je preostalo.

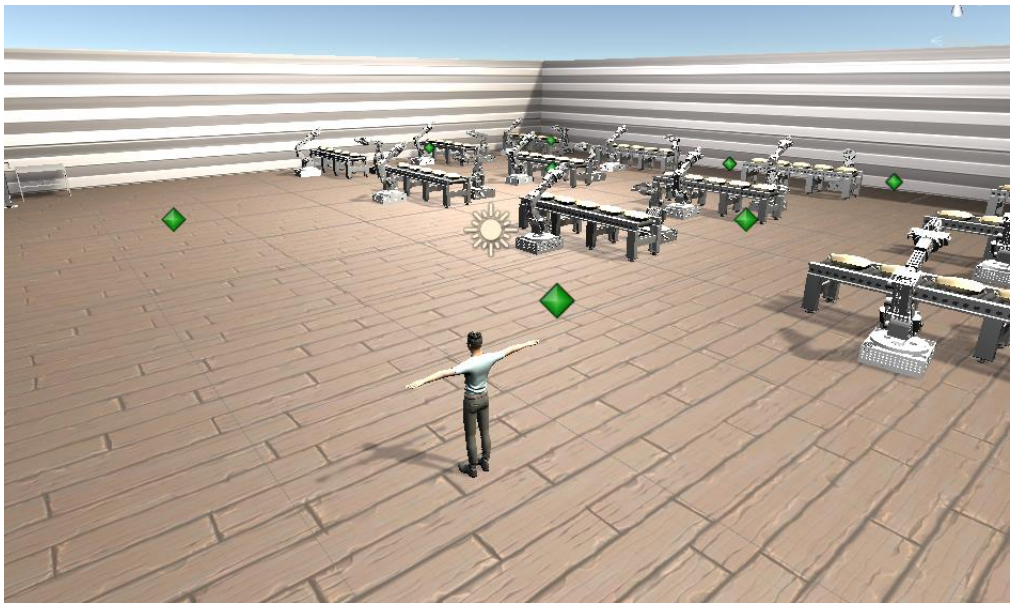
Imamo AI i skriptu, posljednje što nam ostaje jesu upravo te točke iz prethodne skripte. U hijerarhiji kreiramo prazni objekt koji ćemo nazvati „Points AI“. Prazni objekti nam služe kako bismo imali preglednu strukturu svih objekata koji se nalaze na sceni. Zatim unutar tog objekta, pritisnemo desni klik te ponovo kreiramo prazni objekt. Razlika između ovog praznog objekta i ostalih je ta što ćemo njega označiti s ikonom kako bismo ga vidjeli na sceni. Prazne objekte označujemo s ikonicama putem inspektora pritiskom na ikonicu kocke. Njih sada možemo redom poslagati gdje sve želimo da se naša patrola kreće.

Slika 39 - Kreiranje točaka po kojima se agent kreće



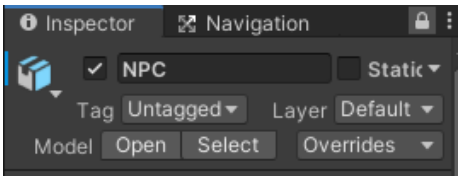
Slika 40 - Odabir izgleda tih točaka

Tako dobijemo sljedeću sliku:



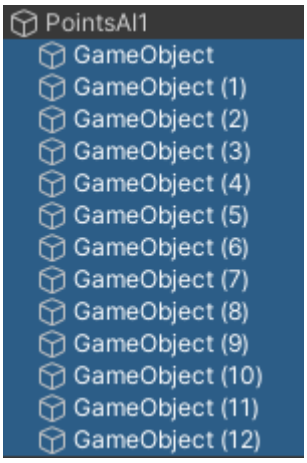
Slika 41 - Prikaz scene sa AI točkama

Kako bismo povezali te objekte sa skriptom, potrebno je zaključati inspektor lika patrole. Odaberemo lika koji predstavlja patrolu te u gornjem desnom kutu pritisnemo lokot. Tako ćemo imati pristup njegovim komponentama, iako smo odabrali neki drugi objekt.

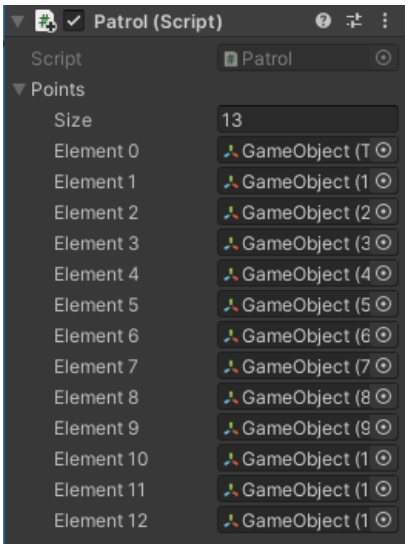


Slika 42 - Zaključani inspektor

Sada, iz hijerarhije, odaberemo sve točke kojima će se patrola kretati te ih povučemo u polje *Points* koje se nalazi u skripti.



Slika 43 - Odabrane točke



Slika 44 - Točke prenesene u skriptu

5. Zaključak

Ovaj završni rad se bazirao na kreiranju i prikazu 3D modela korištenjem VR tehnologije. Kako bismo to postigli koristili smo se Unityjem. Osim njega, kao pomoćni alati poslužile su i stranice Mixamo i 3D Warehouse. Te dvije stranice su nam pružile potrebne objekte te likove i animaciju. Cijeli taj prikaz 3D modela bazira se na virtualnoj realnosti za mobilne uređaje. Kako korištene naočale nisu imale dodatne funkcije, upravljanje cijelom scenom se zapravo bazira na samom pogledu korisnika.Korisnik stavi mobilni uređaj u VR naočale te se ponaša kao da je u prostoriji. Ako se korisnik okrene, okrene se i scena na mobilnom uređaju. Kada korisnik pogleda gore, on će se početi kretati. Ako se zabije u neki drugi predmet ili ako spusti pogled, zaustavit će se. Aplikacija je vrlo jednostavna za korištenje te prikazuje osnovne mogućnosti virtualne realnosti. Osim toga što korisnik upravlja scenom, pojavljuju se i drugi likovi na sceni. Ti likovi predstavljaju radnike od kojih svaki ima svoje zaduženje. Možemo vidjeti kako neki rade za računalom, dok su drugi na pauzi. Osim njih, na neki način pojavljuje se i umjetna inteligencija. Točnije, kreirali smo lika koji funkcionira na bazi umjetne inteligencije. Postavili smo mu zadane točke po kojima se može kretati. Lik sam računa koliko je točki zadano, na kojoj se trenutno nalazi i koliko ih je preostalo. Ako je preostalo 0 točki, on se vraća na početnu i tako u krug.

6. Literatura

Internetski izvori:

1. https://www.udemy.com/course/building-your-first-vr-experience-with-unity/
2. https://docs.unity3d.com/Manual/index.html
3. https://3dwarehouse.sketchup.com/
4. https://www.mixamo.com/

Sažetak

Ovaj završni rad se bazira na prikazu 3D modela robotske ruke. Osim tog modela koristili su se i drugi modeli koji su preuzeti sa stranice 3D Warehouse. Koristili smo ih iz razloga kako bi scena bila bogatija i življa. Kako ne bi imali samo prazne modele bez ikakvih animacija, dodali smo i radnike. Svakom smo liku dodali neku animaciju koja predstavlja simboliku stvarnog rada, dakle na taj smo način simulirali radno okruženje u realnom vremenu. Jedan od radnika je zapravo i AI, tj. bazira se na umjetnoj inteligenciji. To je zapravo AI patrola koja ima zadane točke kretanja. Kako se cijela scena bazira na 3D modelima, potrebno ih je i prikazati tako da ih možemo okružiti. Iz tog razloga je cijela scena zapravo u virtualnoj realnosti. Virtualna realnost nam omogućuje pregled i upravljanje umjetnim okruženjem kojeg smo mi kreirali koristeći određene modele, materijale, animacije i skripte. Korisnik putem mobilnog uređaja upravlja cijelom scenom tako što se zaustavlja i kreće na komandu. Te komande su zapravo kutevi pogleda. Kako korisnik prati scenu, tako se scena prilagođava korisniku.

Ključne riječi: Unity, VR, virtualna realnost, 3D modeli, AI, umjetna inteligencija

Abstract

This work is based on a 3D display of a robotic hand. Besides that model, we used other models that were downloaded from a site called 3D Warehouse. We used them for a reason so that the scene is richer and more vivid. We added even some workers so that we don't have just empty objects. Each worker has a specific animation that represents a symbol of real work, so in that way, we simulated a real-time work environment. One of the workers is actually an AI agent that patrols between a designated set of points. As the whole scene is based on the display of 3D models, we have to display them in a way so that we can go around them. That is why the whole scene is actually made in virtual reality. Virtual reality allows us to display and control the artificial environment that we created using specific models, materials, animations, and scripts. The user has control over the whole scene by using his mobile phone, in a way that he can start and stop moving on command. These commands are actually angles of sight. As the user follows the scene, the scene is adjusting to the user.

Keywords: Unity, VR, virtual reality, 3D models, AI, artificial intelligence