

Virtualna stvarnost - primjena na Androidu i Google Cardboard-u

Bengez, Mario

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Polytechnic of Međimurje in Čakovec / Međimursko veleučilište u Čakovcu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:110:532546>

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

Download date / Datum preuzimanja: **2025-02-05**



Repository / Repozitorij:

[Polytechnic of Međimurje in Čakovec Repository -
Polytechnic of Međimurje Undergraduate and
Graduate Theses Repository](#)



MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

STRUČNI STUDIJ RAČUNARSTVO

MARIO BENGÉZ

VIRTUALNA STVARNOST – PRIMJENA NA ANDROIDU I GOOGLE
CARDBOARDU

ZAVRŠNI RAD

ČAKOVEC, 2021.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI STUDIJ RAČUNARSTVO

MARIO BENGÉZ

VIRTUALNA STVARNOST – PRIMJENA NA ANDROIDU I GOOGLE
CARDBOARD-U

VIRTUAL REALITY – APPLICATION ON ANDROID AND GOOGLE
CARDBOARD

ZAVRŠNI RAD

mentor: Nenad Breslauer, v. pred.

ČAKOVEC, 2021.

Zahvala:

Iskreno zahvaljujem mentoru. Nenadu Breslaueru na pomoći i prenesenom znanju koje mi je pomoglo u izradi ovog rada.

Zahvaljujem i svojoj obitelji te prijateljima na podršci tijekom studiranja.

SAŽETAK:

Industrija videoigara gospodarski je sektor čiji je cilj razvoj, marketing i monetizacija videoigara. Ona obuhvaća milijune korisnika što zahtijeva zapošljavanje velikog broja stručnih ljudi širom svijeta. Najbrže je rastuća industrija, a vrijedi više od 150 milijardi dolara što je gotovo tri puta više nego filmska industrija koja pripada istoj kategoriji „zabavnog“ tipa.

Najveći broj igrača čini muška populacija oko 59% dok 41% čini ženska populacija. Za razliku od muškaraca, koji su podijeljeni na više platformi kao što su konzole „Playstation“, „Xbox“, „Nintendo Switch“ i PC (engl. personal computer) platforme „Steam“ i „Epic Games“, većina ženske populacije usmjerena je na mobilne igre na svojim pametnim telefonima. To čini pametne telefone najvećom platformom za igranje igara. [1]

Virtualna stvarnost je tehnologija razvijena još u 20. stoljeću za potrebe avio i auto industrije. Tek posljednjih desetak godina nakon razvoja prvog prototipa „Oculus Rift“ počela se ubrzano razvijati za potrebe zabave pružajući svojim korisnicima doživljaj virtualnog svijeta i interakciju s njim.

Na tržištu postoji više besplatnih pokretača (engl. game engine) kao što su „Unreal Engine“, „CryEngine“, „Armory“, „Unity“, „Godot“. Za izradu ovog rada korišten je Unity. Ovim radom prikazan je postupak izrade računalne igre namijenjene za platformu Android uz korištenje Google Cardboard naočala za virtualnu stvarnost. Za izradu ovog rada potrebno je korištenje minimalno tri programska paketa. Za pisanje programskog koda koristi se integrirano razvojno okruženje (engl. IDE) Microsoft Visual Studio i Blender koji se koristi za izradu i teksturiranje 3D modela. Za modificiranje audiozapisa u radu korist će se Audacity. Kako bi se sve spojilo u jednu cjelinu koristi se Unity 3D. Unity je game engine za izradu same igre i pomoću njega će svi elementi biti spojeni u konačan proizvod. U radu će se detaljno opisati i prikazati primjer upotrebe svakog od potrebnih programa u svrhu izrade android igre u virtualnoj stvarnosti.

Ključne riječi: *Android, Google Cardboard, Blender, virtualna stvarnost, Microsoft Visual Studio, Audacity, Unity 3D*

Sadržaj

1. UVOD	7
2. PROGRAMSKI PAKET	8
2.1. Unity	8
2.1.1. Unity sučelje	8
2.2. Google Cardboard SDK.....	8
2.2. Blender	9
2.3. Microsoft Visual Studio 2019	9
2.4. Audacity	9
3. PRIMJENA UNITY-A U VIRTUALNOJ STVARNOSTI.....	10
3.1. Izrada i priprema projekta.....	10
3.2. Kreiranje igrača	11
3.3. Izrada početne scene.....	13
3.4. Skybox.....	16
4. PRIMJENA PROGRAMA BLENDER	17
5. PRIMJENA PROGRAMA AUDACITY	20
6. POSTAVLJANJE PROSTORIJE MUZEJA	21
6.1. Kretanje igrača.....	22
6.1.1. Koraci igrača.....	23
6.2. Prikaz opisa slike	23
6.3. Paljenje i gašenje svjetla.....	25
7. IZRADA ZAVRŠNE DATOTEKE	27
8. OBJAVA IGRE NA GOOGLE PLAY TRGOVINI.....	29
7.2. Postavke za objavljivanje	29
9. ZAKLJUČAK	31

10. LITERATURA	32
POPIS SLIKA	33

1. UVOD

Svakodnevno sve više ljudi koristi računalne igre kao najčešći oblik zabave. Svake godine industrija videoigara u sve je većem usponu, a to daje puno mogućnosti za zaposlenje. Uz današnje tehnologije svatko može izraditi jednostavnu igru, ali za razvoj složenije igre potreban je cijeli tim dizajnera, animatora, 3D modelara, menadžera projekta, programera. Od takvog tima može se očekivati da će u nekom normalnom roku napraviti kompletnu igru koja ima početak i kraj.

Virtualna stvarnost (*engl. Virtual Reality*) promijenila je način na koji se igre doživljavaju. Dok su za računala najznačajniji Oculus Rift i Valve Index na pametnim telefonima najveći je pomak napravio Google Cardboard zbog svoje jednostavnosti i pristupačnosti. Virtualna stvarnost je uz zabavu donijela i mogućnost istraživanja lokacija koje nisu svakome nadohvat ruke.

Ovaj rad koristit će game engine Unity, verzija 2008.4.26f1, razvojno okruženje za pisanje programskog koda je Visual Studio 2019, a za 3D modeliranje koristi se Blender 2.83.5. Prikazat će se postupak izrade virtualnog muzeja za Google Cardboard.

Cilj je ovog rada prikazati da pomoću virtualne stvarnosti možemo na brz, jednostavan i jeftin način posjetiti bilo koji muzej u svijetu iz udobnosti svog doma. U igri će igrač imati priliku pogledati neke od najpoznatijih slika koje se nalaze u galerijama diljem svijeta. Korištenjem naočala za virtualnu stvarnost (*engl. Google Cardboard*) igraču će se nastojati što bolje prikazati virtualni muzej kako stvorio dojam da se nalazi u pravom muzeju.

2. PROGRAMSKI PAKET

Programski paket sastoji se od programa potrebnih za izradu neke videoigre u virtualnoj stvarnosti. Unity je glavni program u kojemu se svi elementi kombiniraju u jednu cjelinu. Blender se koristi za izradu i teksturiranje 3D modela koji će se kasnije koristiti u igri. Microsoft Visual Studio 2019 koristi se za pisanje skripti u C#.

2.1. Unity

Unity, uz Unreal Engine, jedan je od najpoznatijih i najraširenijih softvera za izradu igara na svijetu. Razvila ga je tvrtka Unity Technologies. Tvrtku su osnovali Nicolas Francis, Joachim Ante i David Helgason 2004. godine u Danskoj. Svaki mjesec više od tri milijarde ljudi preuzme igre izrađene u Unityu. Prednost Unitya je da nudi mogućnost izrade igre ili simulacije na gotovo svakoj platformi. Unity omogućava izradu 2D, 3D, VR, AR igre ili simulacije na gotovo 30 platformi, a neke od njih su: Android, iOS, Windows, Linux, MacOS, Google Cardboard, Google Daydream, Oculus, SmartTV i još desetak drugih. Programski jezik koji se koristi za skriptiranje je C# zbog svoje raširene i jednostavne primjene. [2]

2.1.1. Unity sučelje

Unity editor je sučelje koje korisnik koristi za izradu igre i ima pristup raznim alatnim trakama i prozorima kojima može sam definirati razmještaj i veličinu. Glavni elementi sučelja su: Inspector, Hijerarhija scene, prozor scene, prozor igre, preglednik projekta, osnovna alatna traka.

2.2. Google Cardboard SDK

Google Cardboard SDK skup je softvera (*engl. Software development kit*) koji omogućava stvaranje aplikacije za Google Cardboard VR platformu. Ovaj SDK omogućava jednostavnu implementaciju praćenja kretanja, orijentacije i unos korisnika te prikaz virtualne stvarnosti na bilo kojem novijem pametnom telefonu (*engl. Smartphone*). [3]

2.2. Blender

Blender je besplatan softverski alat koji se koristi za izradu 3D modela, vizualnih efekata i animacija. Koristi se od 1995. godine i podržava Windows, Linux i MacOS. Blender je najčešće korišten program kada je potrebna izrada nekog objekta za igru. S obzirom da je besplatan, a po svojim mogućnostima nije lošiji od alata koji se naplaćuju, najčešći je odabir animatora i modelatora. Ovim programom izrađuje se potpun dizajn neke 2D ili 3D igre. [4]

2.3. Microsoft Visual Studio 2019

Microsoft Visual Studio integrirano je razvojno okruženje (*engl. Integrated development environment*) u vlasništvu tvrtke Microsoft. Ovaj program omogućuje pisanje C, C++, C#, python, TypeScript, JavaScript, F# i mnogo drugih programskih jezika. U izradi ovog rada koristit će se programski jezik C#. Prednosti ovog programa su brzo i jednostavno pronalaženje sintaktičkih grešaka i mogućnost analize koda koristeći alat za pronalaženje logičkih grešaka (*engl. debugger*). [5]

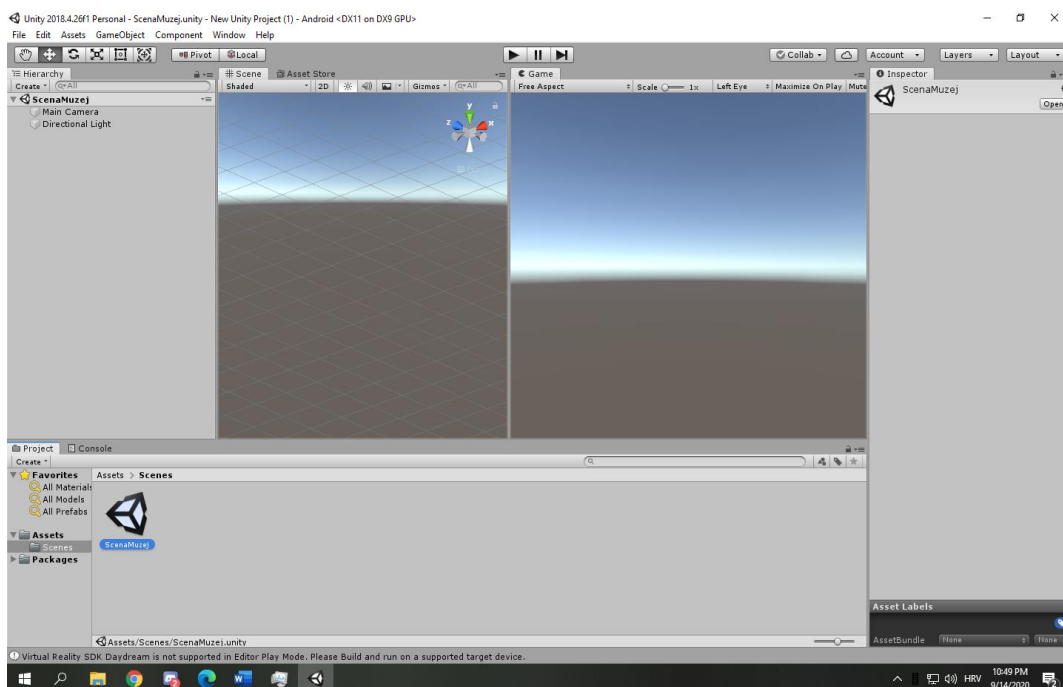
2.4. Audacity

Audacity je besplatni program otvorenog koda (*engl. open source*) za snimanje, uređivanje, rezanje i miksanje audiozapisa. Dostupan je na Windows, macOS i GNU/Linux operativnim sustavima. Programom je moguće isporučiti zvuk u 32 i 64 bitnoj varijanti. S obzirom da samo sučelje programa nije jednostavno, Audacity na svojim službenim stranicama sadrži dokumentaciju i vodiče kako koristiti program. Za potrebe ovog rada Audacity koristit će se za prilagodbu zvučnih efekata koji se koriste u igri. [6]

3. PRIMJENA UNITY-A U VIRTUALNOJ STVARNOSTI

3.1. Izrada i priprema projekta

Otvaranjem Unity programa pojavljuje se prozor koji nudi mogućnost otvaranja postojećih projekata ili kreiranja novog projekta. Ukoliko se stvara novi projekt tada se mora unaprijed znati želimo li 2D ili 3D projekt. Ovaj rad je izrađen u 3D tipu. Nakon izrade projekta uvodi se prethodno preuzet Google Cardboard SDK koji omogućuje primjenu virtualne stvarnosti. Sljedeći je korak napraviti scenu, dati joj smisljeno ime i spremiti je u mapu.[8]

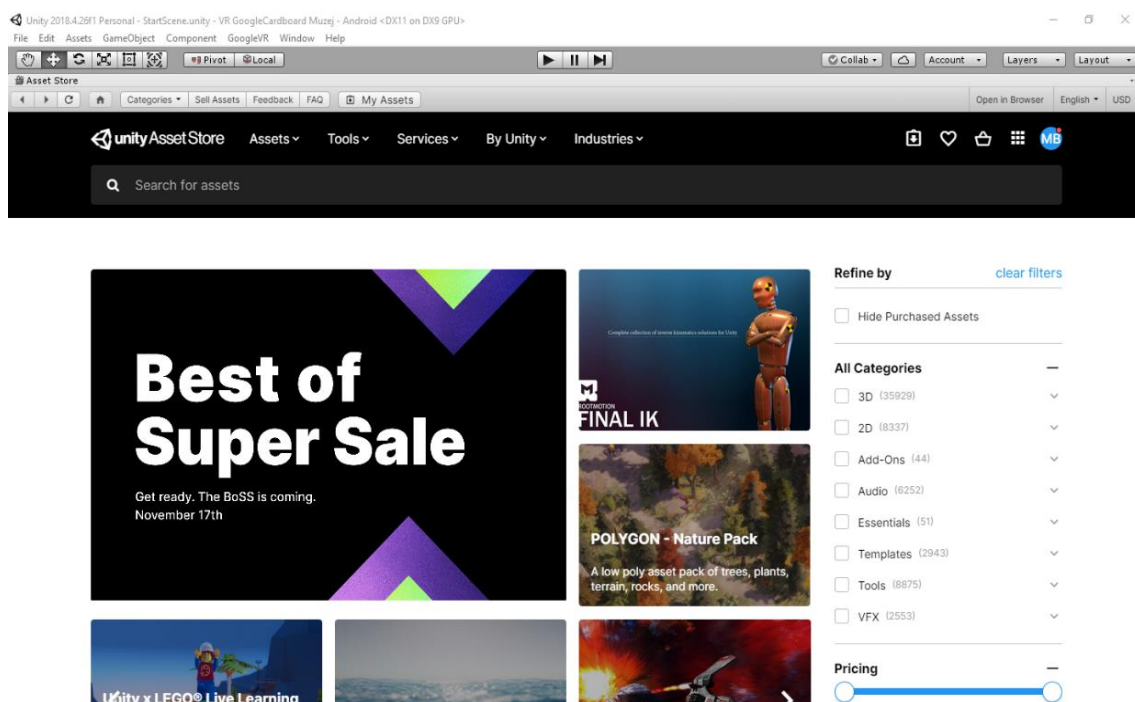


Slika 1. Unity sučelje nakon izrade projekta

Izvor: Autor

„GameObjects“ temeljni su objekti u Unityu koji predstavljaju likove, rekvizite i ambijente. Sami po sebi ne postižu puno, ali djeluju kao spremnici za komponente koje daju objektima stvarnu funkcionalnost. Naprimjer „GameObject“ može biti slika, tipka, 2D ili 3D objekt, kamera. Svaki objekt koji se nalazi u sceni vidljiv je u prozoru „Hierarchy“. Prozor „Scene“ koristi se dizajniranje scene u igri. Korištenje tog prozora pojednostavljuje snalaženje u 3D prostoru i lakše manipuliranje objektima koji se nalaze

u nekoj sceni. Svaki objekt na sceni sadrži komponentu transform koja određuje poziciju, rotaciju i veličinu objekta unutar scene. Prozor „Game“ služi kao prozor za testiranje i prikazuje trenutni izgled igre i ono što će krajnji korisnik vidjeti. Korištenjem ovog prozora uvelike će uštedjeti vrijeme jer ne moramo svaki put izrađivati izvršnu datoteku igre da bi se isprobale implementirane funkcije. Unity omogućuje kreiranje vlastitih jednostavnih modela, ali za neke složenije elemente, materijale i teksture Unity daje mogućnost preuzimanja i uvoženja dodataka sa službenog Asset Storea.



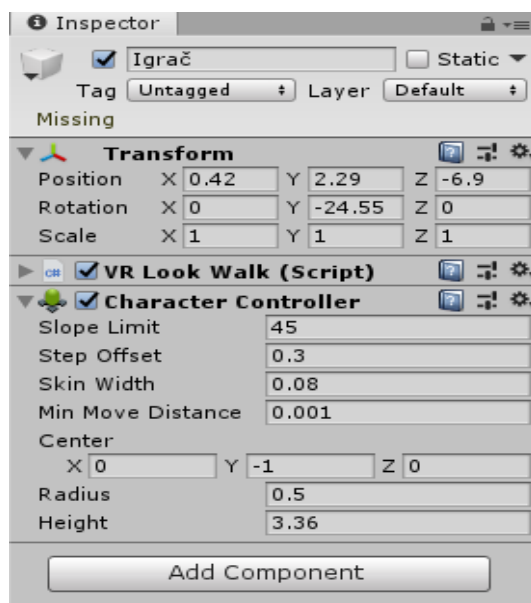
Slika 2. Unity Asset Store

Izvor: Autor

3.2. Kreiranje igrača

Za izradu igrača potrebno je dodati objekt imena Igrač i dodati mu objekte „GvrEditorEmulator“ i „GvrReticlePointer“ iz prethodno uvezenog GoogleVR SDK paketa. Objekt „GvrEditorEmulator“ omogućuje prikaz virtualne stvarnosti na mobilnom uređaju, odnosno podijelit će zaslon na dva dijela i omogućiti kontrolu kamere pomoću senzora ugrađenih u svaki pametni mobilni telefon. Objekt „GvrReticlePointer“ iscrtava točku na sredini zaslona koja olakšava orijentaciju i omogućuje interakciju s drugim

objektima. Dodavanjem ovog objekta automatski se u hijerarhiju dodaje „EventSystem“ objekt bez kojeg „GvrReticlePointer“ ne bi radio. Na objekt „Igrač“ potrebno je dodati objekt „Camera“ koji će biti „child object“ igrača i obrisati „Main Camera“ objekt. Time će kamera moći prikazivati sliku na zaslonu pomoću objekta „GvrEditorEmulator“. Također je potrebno dodati komponentu „Capsule Collider“ koja će omogućiti da igrač ne propada kroz pod ili prolazi kroz zidove i ostale objekte koji se nalaze u igri. I „Capsule Collider“ omogućuju postavljane visine igrača. Da bi se sve komponente spojile u smislenu i funkcionalnu cjelinu, potrebno je dodati komponentu „Script“. Ta će komponenta pomoću programskog koda kontrolirati igrača i omogućiti mu kretanje u prostoru.



Slika 3. Komponente objekta Igrač

Izvor: Autor

3.3. Izrada početne scene

Nakon pripreme podloge, kamere i osvjetljenja na scenu je potrebno dodati korisničko sučelje (*engl. User Interface, UI*) pa se dodaje objekt tipa „Canvas“. „Canvas“ je objekt u kojem se nalaze svi elementi nekog korisničkog sučelja i služi za renderiranje elemenata sučelja. Svi UI elementi objekti su „djeca“ objekta „Canvas“.



Slika 4. Krajnji izgled korisničkog sučelja

Izvor: Autor

Kada se postave tekst i željeni gumbi potrebno je napisati programski kod koji će omogućiti interakciju s tim tipkama. Prvo treba dodati objekt „Gvr Reticle Pointer“ koji će iscrtati točku na sredini zaslona, a ona prikazuje fokus igrača. Kako pojedine „GoogleCardboard“ naočale nemaju gumb, koji bi se koristio kao klik, potrebno je napraviti skriptu koja će to raditi umjesto tipke. Ta će skripta kliknuti tipku ako gledamo u jednu od tipki nekoliko sekundi.[9]

```
public class btnclick : MonoBehaviour
{
    public Image imgcircle;
    public UnityEvent GVRClick;
    public float totalTime = 2;
    bool status = false;
    public float gvrTimer;

    private void Start()
    {
        //postavlja ispunjenost kruga na 0%
        imgcircle.fillAmount = 0;
    }
    // Update is called once per frame
    void Update()
    {
        //provjera statusa i ispunjavanje kruga ako gledamo u zeljenu
tipku
        if(status)
        {
            gvrTimer += Time.deltaTime;
            imgcircle.fillAmount = gvrTimer / totalTime;
        }
        //automatski klik nakon određenog vremena
        if(gvrTimer > totalTime)
        {
            GVRClick.Invoke();
        }
    }
    // "event trigger" pointer je u području tipke
    public void GvrON()
    {
        status = true;
    }

    // "event trigger" pointer nije u području tipke
    public void GvrOFF()
    {

```

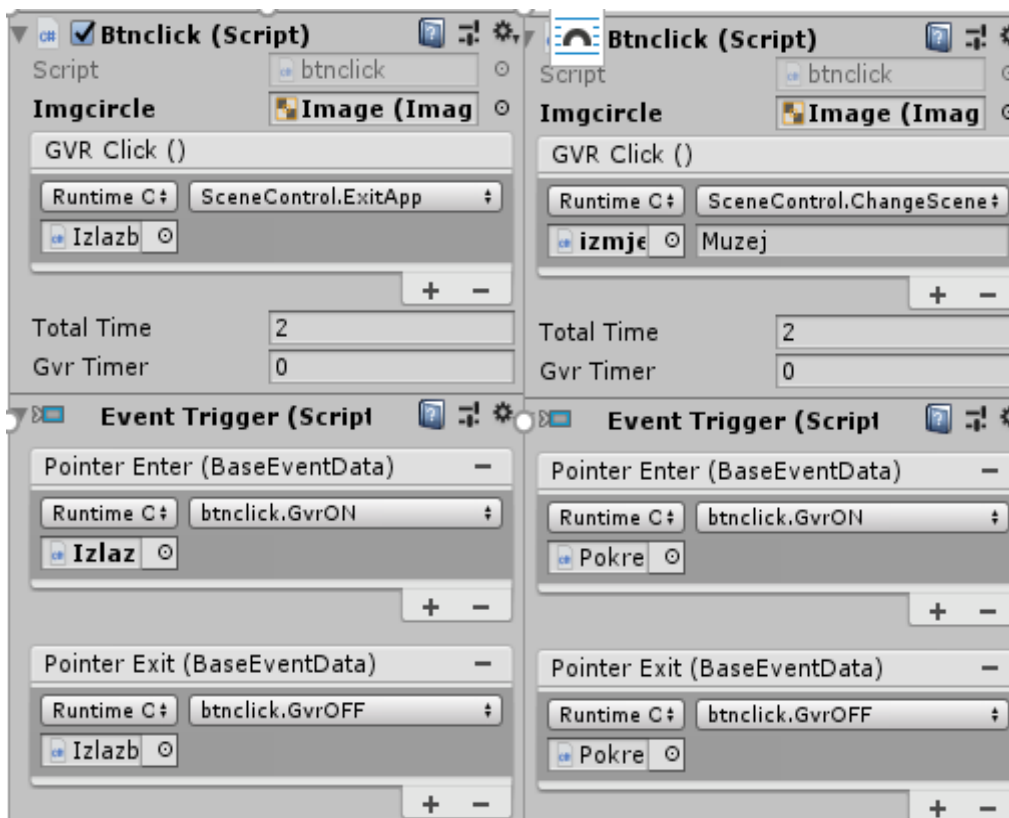
```
status = false;
gvrTimer = 0;
imgcircle.fillAmount = 0;
}
```

Kada je skripta za klik napisana potrebno je dodati novi objekt imena „izmjena_scene“ koji će na sebi imati skriptu koja igraču omogućuje prelazak na iduću scenu ili izlazak iz igre.

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

public class SceneControl : MonoBehaviour
{
    //promjena scene
    public void ChangeScene(string sceneName)
    {
        SceneManager.LoadScene(sceneName);
    }
    //izlaz iz aplikacije
    public void ExitApp()
    {
        Application.Quit();
    }
}
```

Da bi se kreirana skripta dodala na tipke u igru, potrebno je objekt „izmjena scene“ i skriptu za klik dodati na obje tipke. Nakon dodane skripte potrebno je na svakom gumbu dodati potrebne funkcije u „Event Trigger“ komponenti. Na tipki „Pokreni“ postaviti će se naziv scene kako bi skripta znala na koju se scenu igra treba prikazivati. Tipka „Izlaz“ zahtijeva isti postupak, samo će se umjesto funkcije za izmjenu scene odabrati funkcija „ExitApp“ koja će zatvoriti pokrenutu igru. Kako bi se testiralo napravljeno, potrebno je u „Build Settings“ dodati kreirane scene redoslijedom kojim će se prikazivati.

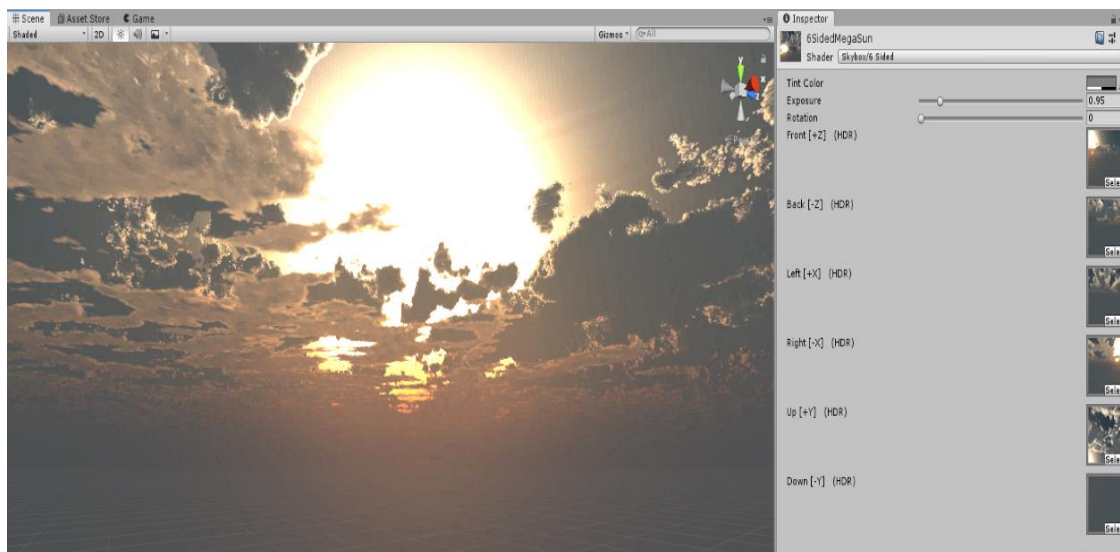


Slika 5. Odabir funkcija na tipkama

Izvor: Autor

3.4. Skybox

Skybox je omot oko cijele scene nacrtan iza svih ostalih grafika u igri. Uloga mu je stvaranje iluzije dalekog horizonta. Može se prikazati na dva načina. Jedan koji se koristi u ovome projektu je prikaz 6 kutija s teksturama ili pomoću teselirane sfere pomoću koje je moguće pretvoriti svaku panoramsku i 360° sliku u skybox. Da bi se skybox promijenio iz onog izvornog slijedi se put: Window → Rendering → Lighting Settings.



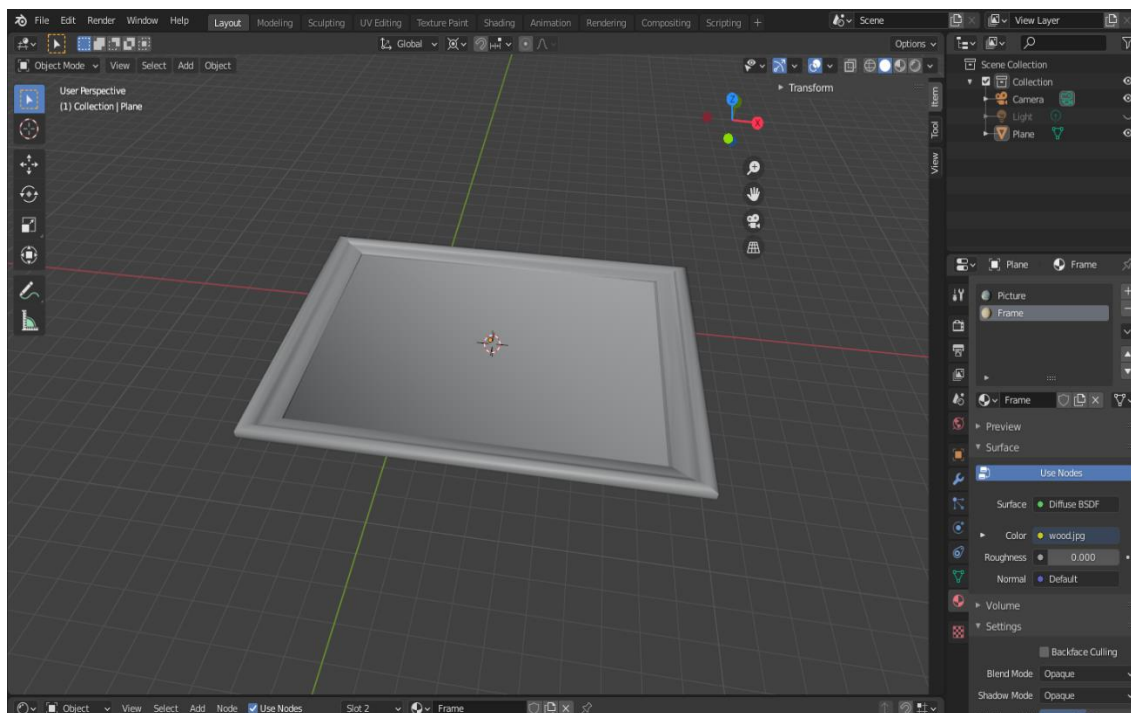
Slika 6. Prikaz Skyboxa

Izvor: Autor

4. PRIMJENA PROGRAMA BLENDER

Blender će se u ovom projektu koristiti za izradu i teksturiranje objekata. Unity ne omogućuje izradu kompleksnih objekata pa se iz tog razloga koristi blender. Za izradu modela korištene su osnovne funkcionalnosti programa poput skaliranja, rotiranja, ispune, istiskivanja, rezanja i drugih, a sve se odvijalo preko objektnog moda (*engl. Object mode*) i moda za uređivanje (*engl. Edit mode*).

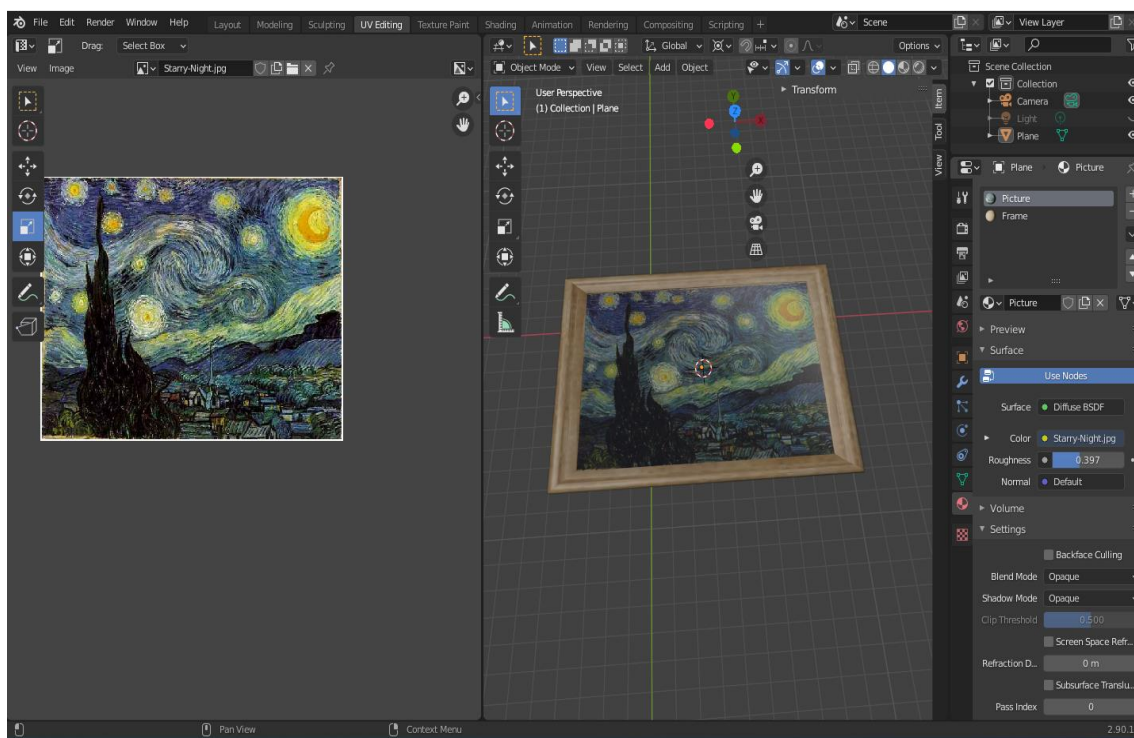
U program je potrebno uvesti datoteku formata (*.blend*) te dodati u scenu kao bilo koji objekt.



Slika 7. Prikaz prije teksturiranja

Izvor: Autor

Teksturiranje (*eng. texturing*) je metoda za dodavanje detalja na površinu objekta. Na površinu se dodaje slika. Detalji sa slike primjenjuju se na cijeloj površini objekta neophodno o reljefu objekta.



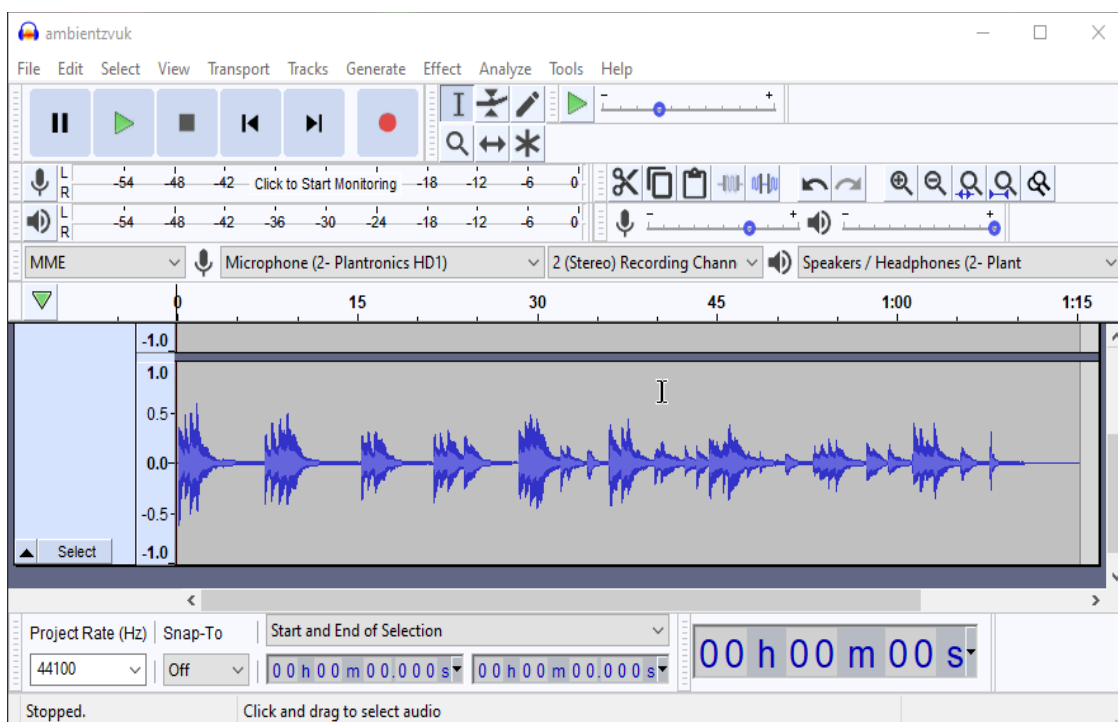
Slika 8. Prikaz nakon tekstuiranja

Izvor: Autor

Kako bi se na neki objekt primjenila određena tekstura potrebno je pod opcijama objekta odabrati „Materials“ odabrati opciju „Image Texture“ i zatim dodati sliku koja će se prikazivati na objektu kao tekstura. Da bi tekstura bila vidljiva potrebno je postaviti scenu na „Rendered“.

5. PRIMJENA PROGRAMA AUDACITY

Sučelje Audacityja je podjeljeno na osamnaest dijelova, a tri glavna dijela su gumbi za editiranje, alatna traka i traka zvuka (*engl. timeline*). Gumbi za editiranje omogućuju reproduciranje, pauziranje, premotavanje, rezanje i skraćivanje zapisa. Alatna traka služi za primjenu određenih zvučnih efekata, dodavanje prethodno generiranih zvukova, pojačavanju i stišavanju završetka audiozapisa, izoštravanju zvuka te nudi još mnoge druge mogućnosti. Najviše se koristi traka zvuka koja prikazuje promjene koje su napravljene na pojedinom audiozapisu i omogućuje zumiranje audiozapisa kako bi se mogle vidjeti i najmanje promjene u nekom djeliću sekunde. Za potrebe ovog rada audiozapis će se izrezati i skratiti kako bi se jednostavnije implementirao u igru.



Slika 9. Sučelje programa Audacity

Izvor: Autor

6. POSTAVLJANJE PROSTORIJE MUZEJA

Prvi je korak izrade scene muzeja bio napraviti prostoriju koristeći GameObject > 3DObject > Plane. Kad se izradi prostorija potrebno je dodati teksture preuzete s trgovine resursa (*engl. Asset Store*). Nakon tekstura postavljamo slike s okvirima na zidove.



Slika 10. Prostorija sa slikama

Izvor: Autor

6.1. Kretanje igrača

GoogleCardboard ne sadrži tipke ili kontroler koji bi se koristio za kretanje pa se mora naći drugi način za kretanje po prostoru. U nastavku je prikazan programski kod za kretanje igrača pogledom prema dolje.

```
public class VRLookWalk : MonoBehaviour
{
    public Transform vrCamera;
    //kut gledanja koji pokrece hodanje
    public float toggleAngle = 30.0f;
    //brzina
    public float speed = 3.0f;
    public bool moveFoward;
    public CharacterController cc;

    // Start is called before the first frame update
    void Start()
    {
        cc = GetComponentInParent<CharacterController>();
    }

    // Update is called once per frame
    void Update()
    {
        //hodanje ako je pogled između zacrtanog kuga i 90°
        if ( vrCamera.eulerAngles.x >= toggleAngle & vrCamera.eulerAngles.x < 90.0f)
        {
            moveFoward = true;
        }
        else
        {
            moveFoward = false;
        }
        //hodanje određenom brzinom
        if(moveFoward == true)
```

```
{  
    Vector3 foward = vrCamera.TransformDirection(Vector3.forward);  
    cc.SimpleMove(foward * speed);  
}  
}
```

6.1.1. Koraci igrača

Za zvučne efekte poput koraka potrebno je na objekt „Igrač“ dodati komponentu „AudioSource“ pomoću koje će se reproducirati zvuk i skriptu „footsteps“ koja brine o tome hoće li se zvučni efekt pokrenuti i zaustaviti u pravo vrijeme.

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class footsteps : MonoBehaviour  
{  
    bool moveforward;  
    // Update is called once per frame  
    void Update()  
    {  
        //referenca na bool varijablu iz VRLookWalk skripte  
        moveforward = GetComponent<VRLookWalk>().moveFoward;  
  
        if (moveforward == true && GetComponent<CharacterController>().velocity.magnitude > 2f &&  
            GetComponent<AudioSource>().isPlaying == false)  
        {  
            GetComponent<AudioSource>().Play();  
        }  
    }  
}
```

6.2. Prikaz opisa slike

Opis slike vrši se pomoću „EventTrigger“ komponente. Pokraj svake slike nalazi se papir na koji se potrebno fokusirati pomoću iscrtane točke na ekranu da bi se prikazao opis slike sa

„Canvasa“ koji je dijete objekta igrač. Skripta će pogledom na određeni papir prikazati samo kanvas koji sadrži opis određene slike.

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

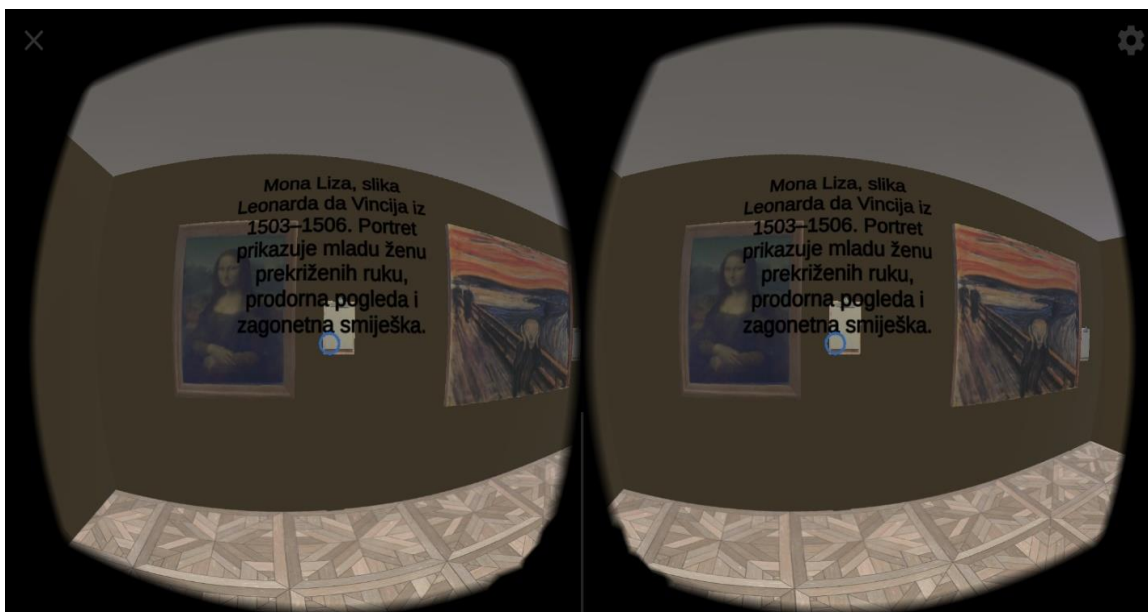
public class Opis : MonoBehaviour
{
    //public Text text;
    public GameObject textmesh;

    void Start()
    {
        // text.enabled = false;
        textmesh.gameObject.SetActive(false);
    }

    //prikaz opisa slike
    public void PokaziOpis()
    {
        // text.enabled = true;
        textmesh.gameObject.SetActive(true); ;
    }

    //sakrivanje opisa slike
    public void MakniOpis()

    {
        //text.enabled = false;
        textmesh.gameObject.SetActive(false);
    }
}
```



Slika 11. Prikaz opisa slike

Izvor: Autor

6.3. Paljenje i gašenje svjetla

Da bi se postiglo paljenje i gašenje svjetla, u scenu je prvo potrebno dodati osvjetljenje. U ovom slučaju na svjetla se dodaje komponenta „Light“ tipa „point“. Točkasta se svjetlost koristi da bi scena bila podjednako osvjetljena od svakog izvora svjetlosti koji se nalazi na sceni. Paljenje i gašenje svjetla kontrolira prazni objekt (engl. empty object) pomoću skripte.

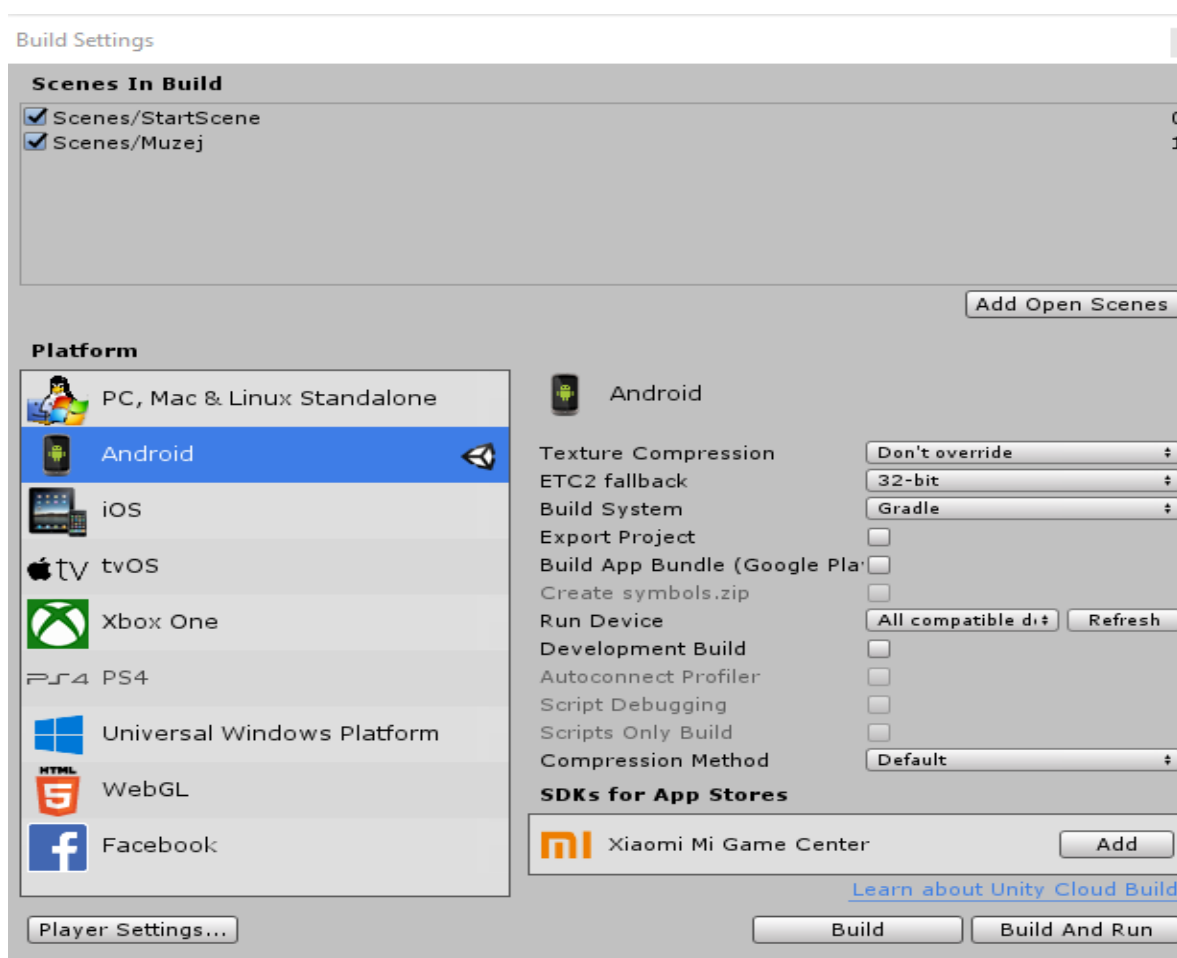
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class prekidac: MonoBehaviour
{
    public Light[] svjetla;
    public bool isOn = false;
    private AudioSource audio;
    void Start()
    {
        audio = GetComponent<AudioSource>();
        for (int i = 0; i < svjetla.Length; i++)
        {
```

```
        svjetla[i].enabled = isOn;
    }
}
public void UpaliUgasi()
{
    isOn = !isOn;
    for(int i = 0; i < svjetla.Length; i++)
    {
        svjetla[i].enabled = isOn;
    }

    audio.Play();
}
```

7. IZRADA ZAVRŠNE DATOTEKE

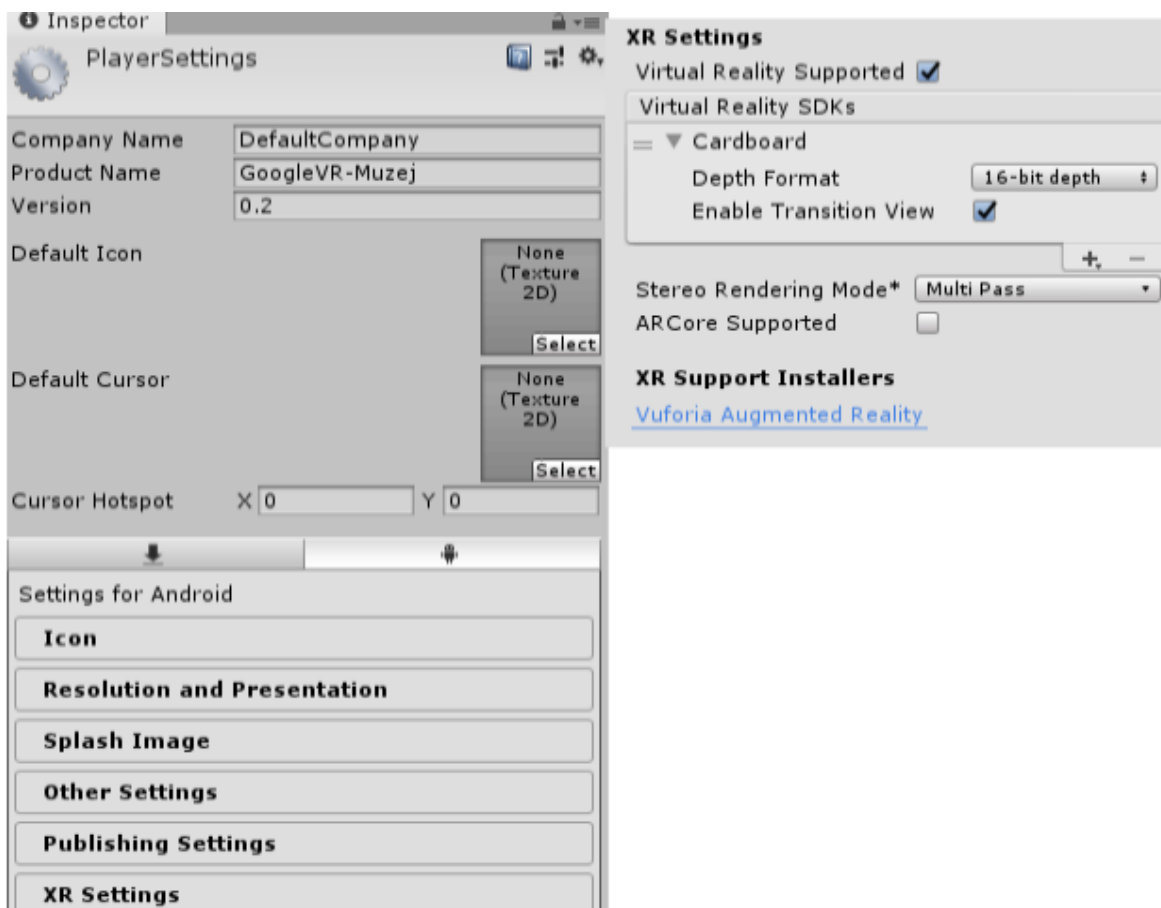
Da bi se završila izrada igre, potrebno je napraviti datoteku kompatibilnu s android uređajima. Prvi korak je dodati sve scene koje igra koristi na listu scena. Poredak scena je veoma važan jer će se prilikom pokretanja uvijek učitati scena koja je prva na listi dok je za pokretanje ostalih scena potrebno napisati programski kod koji će u određenom trenutku pokrenuti određenu scenu. Nakon dodavanja scena potrebno je promijeniti platformu na Android kako bi se izvršna datoteka mogla pokrenuti pomoću android pametnog telefona.



Slika 12. Lista scena i platforma

Izvor: Autor

Sljedeći je korak otvoriti „PlayerSettings“ → „XR Settings“ te dodati paket „Cardboard“ bez kojeg igra neće raditi. Zatim je potrebno odabrati minimalnu razinu aplikacijskog programskog sučelja (*engl. application programming interface, API*) koje mora biti razine 19 ili više. Nakon toga može se proizvoljno dodati ikona i naziv, ali to ne utječe na samu funkcionalnost igre.



Slika 13. "PlayerSettings"

Izvor: Autor

8. OBJAVA IGRE NA GOOGLE PLAY TRGOVINI

Google Play je aplikacija za preuzimanje sadržaja za android uređaje. Predstavljena je 2008. godine pod imenom Android Market, a 2012. godine spojena je s aplikacijama Google glazba i Google e-knjige. Trenutno je na aplikaciji moguće preuzeti oko tri milijuna aplikacija raznih vrsta i žanrova.

7.2. Postavke za objavljivanje

U odjeljku „Publishing Settings“ bit će potrebno podesiti svojstva. Prva i najvažnija stvar je „Keystore“. Stvorit ćemo novu datoteku koja u sebi sadrži ključ aplikacije. Taj je ključ potreban u slučaju da će se aplikacija nadograđivati ili ažurirati. Bez ključa nije moguće objaviti aplikaciju u trgovini.

Publishing Settings

Keystore

Use Existing Keystore Create a new keystore

Browse Keystore C:/Users/Mario/VR Go

Keystore password *****

Confirm keystore passwo *****

Key

Alias mario

Password *****

Build

Custom Gradle Template

User Proguard File

Minify

Release None

Debug None

Split Application Binary

Use legacy SDK tools

Slika 14. Primjer postavki za objavljivanje

Izvor: Autor

Prije objave na „Google Play“ trgovini potrebno je izraditi razvojni korisnički račun, a cijena mu iznosi 25 američkih dolara. Nakon kupnje potrebno se prijaviti na web-stranicu „Google Play Console“ na kojoj će se odvijati proces objave aplikacije. Prije

pokretanja izrade aplikacije potrebno je odabrati „Build App Bundle (GooglePlay)“ opciju kako bi Unity znao da treba izraditi datoteku tipa (.aab) i postaviti „Target API level“ na razinu 30 jer od kolovoza 2021. godine nije moguće objaviti aplikacije čija je API razina manja.[7] Nakon izrade aplikacije potrebno je otvoriti Google Play konzolu i popuniti jezik, naslov aplikacije, kratak opis i puni opis. Od grafičkih je materijala obavezno postaviti minimalno dvije snimke zaslona od maksimalno njih osam. Nakon grafičkih materijala potrebno je učitati ikonu aplikacije i po želji postaviti link na promotivni video. Za kraj je potrebno odrediti kategoriju aplikacije koja je u ovom slučaju simulacija. Kategorizacijom završava prvi dio objave aplikacije u trgovini.

Konačan je dio objave učitati izvršnu datoteku aplikacije tipa (.aab). Prilikom učitavanja datoteke automatski se provjerava ispravnost i sadržaj zlonamjernih softvera. Ako je aplikacija ispravna i prođe sve testove, potrebno je kliknuti tipku „Pokreni uvođenje u produkciju“ kojim će se pokrenuti pregledavanje aplikacije pomoću algoritama koji traje 24 sata. Nakon 24 sata aplikacija će biti dostupna za preuzimanje na trgovini.

9. ZAKLJUČAK

Ovim je radom predstavljen je razvoj 3D mobilne igre uz pomoć Google Cardboarda i Unity 3D game enginea. Virtualna je stvarnost tehnologija koja pokušava korisniku dočarati virtualni svijet kao stvarnost. S obzirom da je razvoj virtualne stvarnosti veoma brz, povećava se i interes za nove i uzbudljive aplikacije i igre koje koriste tu tehnologiju. Unity je danas jedan od najraširenijih game enginea za razvoj računalnih igara, a zbog svoje jednostavnosti namijenjena je svima – od profesionalaca do potpunih početnika u svijetu videoigara.

Nadogradnja i poboljšanje ove igre je brz i jednostavan posao zbog jednostavnosti Google Cardboard platforme. Performanse Google Cardboarda ovise samo o snazi pametnog telefona i njegovim mogućnostima što znači da poboljšanja poput dodavanja tekstura i objekata visoke rezolucije zahtijevaju pametne telefone daleko većih performansi. Dok dodavanje novih značajki (*engl. features*) poput „text-to-speech“ mogućnosti je puno kompliciraniji zadatak jer zahtijeva kompleksan programski kod i Unity dodatke (*engl. add-ons*) koji nisu uvijek podržani u svim verzijama Unity 3D game enginea.

Zadatak izrade igre bio je prikazati proces izrade mobilne igre u virtualnoj stvarnosti. Izrada ove igre složen je proces jer zahtijeva korištenje više različitih alata koje je potrebno spojiti u jednu smislenu cjelinu.

Cilj je ovog rada ispunjen. Prikazan je način izrade igre pomoću alata koji su svima dostupni i besplatni. Opisan je postupak izrade igre s najnovijim tehnologijama i primjenom potrebnih alata u svrhu upotpunjavanja procesa izrade igre u virtualnoj stvarnosti.

10. LITERATURA

- [1] Distribution of video gamers in the United States from 2006 to 2021, by gender. <https://www.statista.com/statistics/232383/gender-split-of-us-computer-and-video-gamers/> (16.9.2021.)
- [2] Unity User Manual 2020.3. <https://docs.unity3d.com/Manual/index.html> (16.9.2021.)
- [3] Google cardboard SDK for Unity <https://developers.google.com/cardboard/develop/unity/quickstart> (16.9.2021.)
- [4] Blender <https://www.blender.org/> (16.9.2021.)
- [5] Visual Studio 2019 <https://visualstudio.microsoft.com/vs/> (16.9.2021.)
- [6] Audacity Documentation <https://www.audacityteam.org/help/documentation/> (16.9.2021.)
- [7] Meet Google Play's target API level requirement <https://developer.android.com/distribute/best-practices/develop/target-sdk?hl=hr> (17.9.2021)
- [8] Google Cardboard SDK documentation <https://developers.google.com/vr/reference/unity> (16.9.2021.)
- [9] Dokumentacija c#, <https://code.msdn.microsoft.com/> (16.9.2021.)

POPIS SLIKA

Slika 1. Unity sučelje nakon izrade projekta	10
Slika 2. Unity Asset Store.....	11
Slika 3. Komponente objekta Igrač.....	12
Slika 4. Krajnji izgled korisničkog sučelja	13
Slika 5. Odabir funkcija na tipkama	16
Slika 6. Prikaz Skybox-a.....	17
Slika 7. Prikaz prije teksturiranja.....	18
Slika 8. Prikaz nakon teksturiranja	19
Slika 9. Sučelje programa Audacity	20
Slika 10. Prostorija sa slikama.....	21
Slika 11. Prikaz opisa slike	25
Slika 12. Lista scena i platforma.....	27
Slika 13. "PlayerSettings"	28
Slika 14. Primjer postavki za objavljivanje	29